
**HP Integrity Superdome -
Itanium2/1.6GHz/24MB iL3 - 64p/128c**

using

HP-UX 11i v3

and

**Oracle Database 10g Release 2 Enterprise Edition with
Partitioning**

**TPC Benchmark® C
Full Disclosure Report**

**Second Edition
August 21, 2007**

**Submitted for Review
February 27, 2007**



Second Edition - August 21, 2007

Hewlett-Packard Company believes that the information in this document is accurate as of the publication date. The information in this document is subject to change without notice. Hewlett-Packard Company assumes no responsibility for any errors that may appear in this document.

The pricing information in this document is believed to accurately reflect the current prices as of the publication date. However, Hewlett-Packard Company provides no warranty of the pricing information in this document.

Benchmark results are highly dependent upon workload, specific application requirements, and system design and implementation. Relative system performance will vary as a result of these and other factors. Therefore, TPC Benchmark[®] C should not be used as a substitute for a specific customer application benchmark when critical capacity planning and/or product evaluation decisions are contemplated.

All performance data contained in this report was obtained in a rigorously controlled environment. Results obtained in other operating environments may vary significantly. Hewlett-Packard Company does not warrant or represent that a user can or will achieve similar performance expressed in transactions per minute (tpmC[®]) or normalized price/performance (\$/tpmC[®]). No warranty of system performance or price/performance is expressed or implied in this report.

©Copyright Hewlett-Packard Company 2007

All rights reserved. Permission is hereby granted to reproduce this document in whole or in part provided the copyright notice printed above is set forth in full text on the title page of each item reproduced.

Printed in U.S.A., August 21, 2007.

HP, HP-UX, HP C++/ANSI C Developer's Bundle /HP-UX, HP 9000 are registered trademarks of Hewlett-Packard Company.

are registered trademarks of Oracle Corporation.

TUXEDO 8.1 is a registered trademark of BEA System, Inc.

UNIX is a registered trademark in the United States and other countries, licensed exclusively through X/Open Company Limited.

TPC Benchmark, TPC-C, and tpmC are registered certification marks of the Transaction Processing Performance Council.

All other brand or product names mentioned herein are trademarks or registered trademarks of their respective owners.

Abstract

Overview

This report documents the methodology and results of the TPC Benchmark[®] C test conducted on the HP Integrity Superdome - Itanium2/1.6GHz/24MB iL3 - 64p/128c in a client/server configuration, using Oracle Database 10g Release 2 Enterprise Edition with Partitioning and the TUXEDO 8.1 transaction monitor. The operating system used for the benchmark was Hewlett-Packard's HP-UX 11i v3. The application was written in C and compiled using HP C++/ANSI C Developer's Bundle /HP-UX.

TPC Benchmark C Metrics

The standard TPC Benchmark[®] C metrics, tpmC[®] (transactions per minute), price per tpmC[®] (three year capital cost per measured tpmC[®]), and the availability date are reported as required by the benchmark specification.

Standard and Executive Summary Statements

Page *iii* contains the standard system summary and pages *iv-vi* contain the executive summary of the benchmark results for the HP Integrity Superdome - Itanium2/1.6GHz/24MB iL3 - 64p/128c.

Auditor

The benchmark configuration, environment and methodology used to produce and validate the test results, and the pricing model used to calculate the price/performance, were audited by Lorna Livingtree for Performance Metrics, Inc. to verify compliance with the relevant TPC specifications.

Standard System Summary

Company Name	System Name	Database Software	Operating System Software
Hewlett-Packard Company	HP Integrity Superdome - Itanium2/1.6GHz/24MB iL3 - 64p/128c	Oracle Database 10g Release 2 Enterprise Edition with Partitioning	HP-UX 11i v3
HP H/W Availability Date - Now S/W Availability Date - August 6, 2007			
Total System Cost (USD)	TPC-C [®] Throughput	Price/Performance (USD)	
Hardware Software 3-year maintenance	Sustained maximum throughput of System running TPC-C [®] expressed in transactions per minute	Total system cost/tpmC (\$11,978,134/4092799)	
\$11,978,134	4,092,799.42 tpmC	\$2.93 per tpmC	



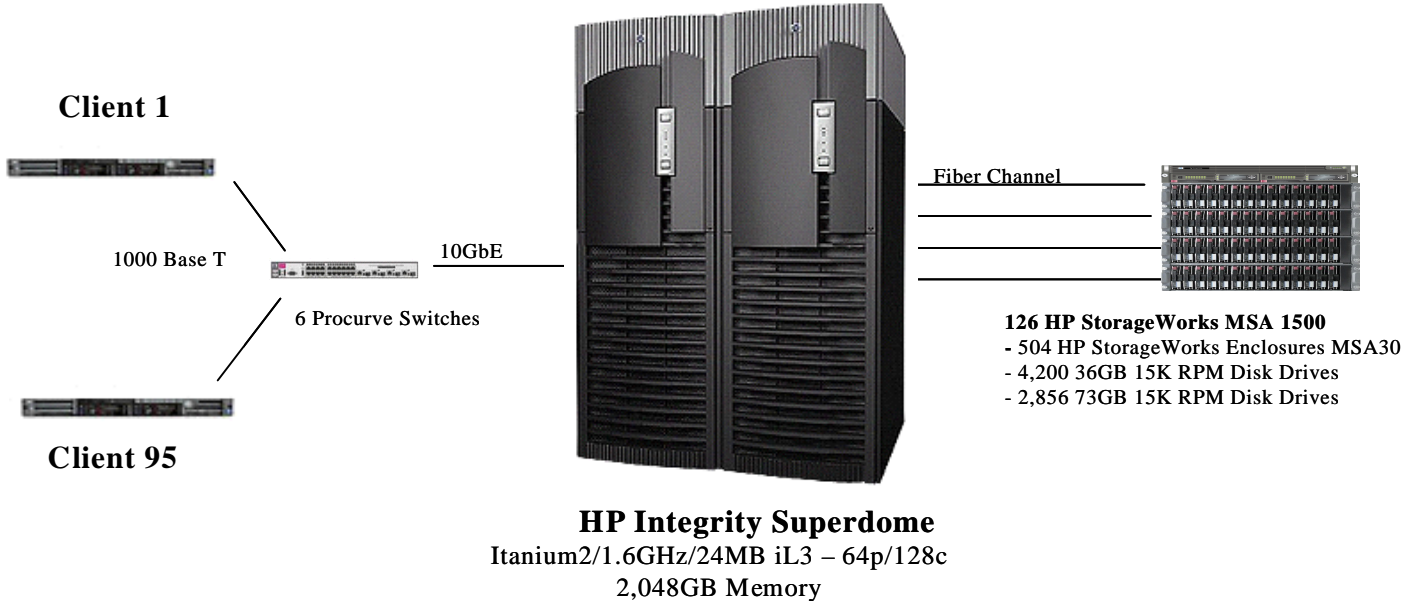
HP Integrity Superdome - Itanium2/1.6GHz/24MB iL3 - 64p/128c

TPC-C Revision 5.8

Report Date:
August 21, 2007

Total System Cost	TPC Throughput	Price/Performance	Availability Date
USD \$11,978,134	4,092,799 tpmC	USD \$2.93/tpmC	August 6, 2007
Server Processors/Cores/Threads	Database Manager	Operating System	Other Software
64/128/256 Intel Itanium2 1.6GHz	Oracle Database 10g Release 2 Enterprise Edition with Partitioning	HP-UX 11i v3	TUXEDO 8.1
			Number of Users
			3,226,200

Server



System Components	Server (HP Integrity Superdome - Itanium2/1.6GHz/24MB iL3 - 64p/128c)		each Client (95 rx1620)	
	Qty	Type	Qty	Type
Processors/Cores/Threads	64/128/256	1.6GHz Intel Itanium2	2/2/2	1.3GHz Intel Itanium2
Cache Memory	2	12MB iL3 Cache	each	3M iL3 cache
Memory	2048	GB	8	GB
Disk Controllers	40	PCI-X 4Gb Fibre Channel Adapter (dual port)	1	Ultra2 SCSI LVD
Disk Drives	126	HP StorageWorks Modular Smart Array 1500 with 4200 36GB 15K RPM and 2856 73GB 15K RPM	1	36 GB
	1	HP Surestore Disk System 2120 with 1 36GB LP 15K HDD		
Total Storage		320974 GB		
Tape Drives	1	DVD ROM		
Terminals	1	HP Server Thin Client		



HP Integrity Superdome - Itanium2/1.6GHz/24MB iL3 - 64p/128c

TPC-C Rev 5.8

Report Date: August 21, 2007

Description	Part Number	Brand	Key	Price US List	Qty	Price	3Year Main.Price
Server Hardware							
Superdome left chassis	A9834A, Opt 429		1	235,950	1	235,950	
Superdome right chassis	A9835A, Opt 429		1	249,950	1	249,950	
Superdome sx2000 Cell Board	A9837A		1	19,250	16	308,000	
3 Year Svc & Support Price (Hardware and Software)							\$1,376,539
Dual-Core Intel Itanium2 9050/1.6GHz/24MB L3	A9840A		1	31,500	64	2,016,000	
512GB Memory Bundle (128x4GB dimms)	A9857A		1	1,599,078	4	6,396,312	
12-Slot PCI-X I/O Chassis	A9836A		1	16,950	16	271,200	
I/O Chassis Enclosure for 12-Slot PCI-X Chassis	A9852A		1	25,750	4	103,000	
Graphite I/O expansion power subsystem	A5861D		1	34,860	2	69,720	
HP Surestore Disk System 2120	A7382A		1	995	1	995	
36GB LP 15K HDD	A7527A		1	966	1	966	
HP Universal Rack10642 G2 Pallet Rack	AF001A		1	1,249	1	1,249	
Modular Power Dist Unit for std racks	A5137AZ		1	145	1	145	
200-240 volts North America	A5137AZ, Opt AW4		1	94	1	94	
PCI-X Dual Ultra320 SCSI Adapter	A7173A		1	795	1	795	
PCI-X 4Gb Fibre Channel Adapter (dual port)	AB379B		1	3,495	40	139,800	
HP DVD + RW Array Field Module	Q1592B		1	649	1	649	
HP rx2620 Server Solution (inc memory, disk, monitor, keyboard/mouse)	AB332A		1	5,315	1	\$5,315	
				Subtotal		9,800,140	1,376,539
Server Software							
Oracle Database 10g Release 2 Enterprise Edition, Unlimited Users, Per Processor, 3 yrs	Runtime		2	20,000	64**	1,280,000	
Partitioning, Unlimited Users, Per Processor, 3 yrs			2	5,000	64**	320,000	
Oracle Database Server Support Package for 3 years			2	2,000	3		6,000
HP-UX 11i v3 Foundation Operating Environment	B9429AC		1	2,370	128	303,360	
HP-UX 11i v3 HP9000 Integrity FOE Media	BA489AA, Opt AJR		1	565	1	565	
				Subtotal		1,903,925	6,000
Storage							
Rack System/E R3000 XR UPS	192186-001		1	1,365	1	1,365	
5 meter Fibre Optic Cable	221691-B22		1	82	346	28,372	
15 meter Fibre Optic Cable	221691-B23		1	103	30	3,090	
HP StorageWorks MSA 1500 w/MSA30 Enc (126+ 13 spares)	AD510A		1	8,995	139	1,250,305	
HP StorageWorks Enclosures MSA30 (378+38 spares)	302969-B21		1	2,829	416	1,176,864	
HP StorageWorks MSA 1000 Controller (126+13 spares)	218231-B22		1	4,290	139	596,310	
MSA1500 Fibre Channel I/O Module 126+13 spares)	AA987A		1	375	139	52,125	
MSA1500 Dual Channel SCSI I/O Module (378 + 38 spares)	AA988A		1	525	416	218,400	
3 Yr Support Price for MSA1000, MSA30, disks, switches							INCLUDED
73GB 15K Ultra320 Hard Drive (2856 + 286 spares)	286778-B22		1	439	3142	1,379,338	
36GB 15K Ultra320 Hard Drive (4200 + 420 spares)	286776-B22		1	319	4620	1,473,780	
HP StorageWorks 4/64 Full SAN Switch	AG457A		1	62,999	2	125,998	
HP StrgWrks4gbSW SnglPK SFP Transcvr	A7446B		1	199	124	24,676	
HP StrgWrks2gbSW SnglPK SFP Transcvr	221470-B21		1	199	296	58,904	
HP StorageWorks SAN Switch 4/8	A8000A		1	4,999	60	299,940	
HP Universal Rack10642 G2 Pallet Rack	AF001A		1	1,249	44	54,956	
				Subtotal		6,744,423	0
Client Hardware							



HP Integrity Superdome - Itanium2/1.6GHz/24MB iL3 - 64p/128c

TPC-C Rev 5.8

Report Date: August 21, 2007

Description	Part Number	Brand	Price US List Key Price	Qty	Price	3Year Main.Price	
Client Hardware							
HP Integrity rx1620 w 1.3GHz Intel Itanium 2 Processor	AB430A		1 3,445	95	327,275		
1.3GHz Intel Itanium 2 Processor	AB481A		1 1,650	95	156,750		
3 Year Support Price (Hardware & Software)						444,192	
36GB 15K RPM Ultra320 SCSI Internal Disk	AB420A		1 389	95	36,955		
2GB Memory Module (2 x 1GB DIMMS)	AB223A		1 1,275	380	484,500		
HP Server Thin Client (monitor, keyboard/mouse, cable inc.)	AB300B		1 1,250	1	1,250		
HP Universal Rack 10642 G2 Shock Rack	AF001A		1 1,249	4	4,944		
Modular Power Dist.	252663-24		1 299	16	4,734		
HP-UX Fndn OE DVD Media	B9106AA, Opt. AJR		1 565	1	565		
HP-UX Fndn OE DVD Media, Factory Integrated	B9106AA, Opt. OD1		1 199	1	199		
HP-UX Fndn OE Integrity PPL max2CPU w/sys	B9430AC		455	190	86,450		
Subtotal					1,103,622	444,192	
Client Software							
HP C++/ANSI C Developer's Bundle	B9007AA, Opt. 2AH		1 966	1	966	170	
BEA Tuxedo 8.1			3 1,200	95	114,000	71,820	
Subtotal					114,966	71,990	
User Connectivity							
HP ProCurve Switch 2724 (24-port)	J4897A		1 1,599	4	6,396		
HP ProCurve Switch 3400cl-24G	J4905A		1 3,795	2	7,590		
ProCurve 10-GbE X2-SC SR Optic transceivers	J8436A		1 2,999	4	11,996		
HP ProCurve 10GbE Meida Flex Module	J8435A		1 2,699	2	5,398		
Subtotal					31,380	0	
Oracle Mandatory E-Business Discount (license and support)			2		(401,500)		
BEA 10% Mandatory Discount			3		(11,400)		
HP's Large configuration Discount and Support Prepayment*					(\$8,279,746)	(\$926,397)	
*** These components are not immediately orderable. See the FDR for more information.					Total	11,005,810	972,324
**2 = 0.50 * 4. Explanation: For the purposes of counting the number of processors which require licensing, an Intel multicore chip with "n" cores shall be determined by multiplying "n" cores by a factor of 0.50.							
*A 46.7% discount was based on the overall value of the specific components from HP (Price Key) in this single quotation. Discounts for similarly sized configurations will be similar to those quoted here, but may vary based on the components in the quotation. Prices are for US dollars and similar quantities.							
1=HP 2= Oracle 3=BEA					Three Year Cost of Ownership (USD):		\$11,978,134
					tpmC Rating:		4,092,799
Audited by Lorna Livingtree of Performance Metrics, Inc.					\$/tpmC (USD):		\$2.93
Prices used in TPC benchmarks reflect actual prices a customer would pay for a one-time purchase of the stated components. Individually negotiated discounts are not permitted. Special prices based on assumptions about past or future purchases are not permitted. All discounts reflect standard pricing policies for the listed components. For complete details, see the pricing sections of the TPC benchmark specifications. If you find that the stated prices are not available according to these terms, please inform the TPC at pricing@tpc.org. Thank you.							

Numerical Quantities Summary for HP Integrity Superdome - Itanium2/1.6GHz/24MB iL3 - 64p/128c

MQTH, Computed Maximum Qualified Throughput

4,092,799 tpmC

Response Times (in seconds)

	90th %-ile	Maximum	Average
New-Order	0.51s	84.83s	0.24s
Payment	0.49s	85.20s	0.23s
Order-Status	0.51s	79.95s	0.23s
Delivery (interactive portion)	0.10s	81.97s	0.14s
Delivery (deferred portion)	0.50s	82.38s	0.21s
Stock-Level	0.61s	81.88s	0.31s
Menu	0.10s	0.28s	0.07s

Transaction Mix, in percent of total transactions

New-Order	44.95%
Payment	43.01%
Order-Status	4.01%
Delivery	4.02%
Stock-Level	4.01%

Keying/Think Times

	Keying Time			Think Time		
	Min	Avg	Max	Min	Avg	Max
New-Order	18.01s	18.02s	18.19s	0.02s	12.02s	251.71s
Payment	3.01s	3.02s	3.19s	0.02s	12.02s	246.09s
Order-Status	2.01s	2.02s	2.18s	0.02s	10.02s	171.08s
Delivery (interactive)	2.01s	2.02s	2.19s	0.02s	5.03s	95.32s
Stock-Level	2.01s	2.02s	2.19s	0.02s	5.02s	89.68s

Test Duration

Ramp up time	42 minutes
Measurement interval	120 minutes
Transactions during measurement interval	1,092,597,205
Ramp down time	23.338 minutes

Checkpointing

Number of checkpoints in measurement interval	4
Checkpoint Interval	29.45 minutes

TPC Benchmark C Overview

This is the full disclosure report for a benchmark test of the HP Integrity Superdome - Itanium2/1.6GHz/24MB iL3 - 64p/128c using Oracle Database 10g Release 2 Enterprise Edition with Partitioning. It meets the requirements of the TPC Benchmark[®] C Standard Specification, Revision 5.8 dated HP Integrity Superdome - Itanium2/1.6GHz/24MB iL3 - 64p/128c.

TPC Benchmark[®] C was developed by the Transaction Processing Performance Council (TPC). It is the intent of this group to develop a suite of benchmarks to measure the performance of computer systems executing a wide range of applications. Hewlett-Packard Company Oracle Corporation are active participants in the TPC.

TPC Benchmark[®] C is an On Line Transaction Processing (OLTP) workload. It is a mixture of read-only and update intensive transactions that simulate the activities found in complex OLTP application environments. It does so by exercising a breadth of system components associated with such environments, which are characterized by:

- The simultaneous execution of multiple transaction types that span a breadth of complexity
- On-line and deferred transaction execution modes
- Multiple on-line terminal sessions
- Moderate system and application execution time
- Significant disk input/output
- Transaction integrity (ACID properties)
- Non-uniform distribution of data access through primary and secondary keys
- Databases consisting of many tables with a wide variety of sizes, attributes, and relationships
- Contention of data access and update

The performance metric reported by TPC-C[®] is a "business throughput" measuring the number of orders processed per minute. Multiple transactions are used to simulate the business activity of processing an order, and each transaction is subject to a response time constraint. The performance metric for this benchmark is expressed in transactions-per-minute-C[®] (tpmC[®]). To be compliant with the TPC-C[®] standard, all references to tpmC[®] results must include the tpmC[®] rate, the associated price-per-tpmC[®], and the availability date of the priced configuration.

Despite the fact that this benchmark offers a rich environment that emulates many OLTP applications, this benchmark does not reflect the entire range of OLTP requirements. In addition, the extent to which a customer can achieve the results reported by a vendor is highly dependent on how closely TPC-C[®] approximates the customer application. The relative performance of systems derived from this benchmark does not necessarily hold for other workloads or environments. Extrapolations to other environments are not recommended.

Hewlett-Packard Company does not warrant or represent that a user can or will achieve performance similar to the benchmark results contained in this report. No warranty of system performance or price/performance is expressed or implied by this report.

Table of Contents

1	GENERAL ITEMS.....	1
1.1	APPLICATION CODE AND DEFINITION STATEMENTS	1
1.2	TEST SPONSOR	1
1.3	PARAMETER SETTINGS	1
1.4	CONFIGURATION DIAGRAMS.....	1
2	CLAUSE 1: LOGICAL DATA BASE DESIGN RELATED ITEMS	4
2.1	TABLE DEFINITIONS	4
2.2	PHYSICAL ORGANIZATION OF DATABASE.....	4
2.3	INSERT AND DELETE OPERATIONS.....	4
2.4	PARTITIONING.....	4
3	CLAUSE 2: TRANSACTION & TERMINAL PROFILES RELATED ITEMS	5
3.1	RANDOM NUMBER GENERATION.....	5
3.2	INPUT/OUTPUT SCREEN LAYOUT.....	5
3.3	PRICED TERMINAL FEATURE VERIFICATION	5
3.4	PRESENTATION MANAGER OR INTELLIGENT TERMINAL	5
3.5	TRANSACTION STATISTICS	6
3.6	QUEUEING MECHANISM	6
4	CLAUSE 3: TRANSACTION AND SYSTEM PROPERTIES RELATED ITEMS	7
4.1	TRANSACTION SYSTEM PROPERTIES (ACID)	7
4.2	ATOMICITY	7
4.2.1	<i>Completed Transaction</i>	7
4.2.2	<i>Aborted Transaction</i>	7
4.3	CONSISTENCY	7
4.4	ISOLATION	8
4.4.1	<i>Isolation Test 1</i>	9
4.4.2	<i>Isolation Test 2</i>	9
4.4.3	<i>Isolation Test 3</i>	9
4.4.4	<i>Isolation Test 4</i>	10
4.4.5	<i>Isolation Test 5</i>	10
4.4.6	<i>Isolation Test 6</i>	11
4.4.7	<i>Isolation Test 7</i>	11
4.4.8	<i>Isolation Test 8</i>	11
4.4.9	<i>Isolation Test 9</i>	12
4.5	DURABILITY.....	12
4.5.1	<i>Loss of Log and Data Disks</i>	13
4.5.2	<i>Instantaneous Interruption and Loss of Memory</i>	14
5	CLAUSE 4: SCALING AND DATA BASE POPULATION RELATED ITEMS.....	15
5.1	INITIAL CARDINALITY OF TABLES	15
5.2	DATABASE AND GROWTH LAYOUT	15
5.3	DATA MODEL & INTERFACES.....	57
5.4	PARTITIONS/REPLICATIONS.....	57
5.5	GROWTH REQUIREMENTS	57
6	CLAUSE 5: PERFORMANCE METRICS AND RESPONSE TIME RELATED ITEMS.....	58
6.1	THROUGHPUT.....	58
6.2	RESPONSE TIME	58
6.3	KEYING AND THINK TIMES	58
6.4	RESPONSE TIME FREQUENCY DISTRIBUTION CURVES AND OTHER GRAPHS	59

6.5	STEADY STATE DETERMINATION	59
6.6	WORK PERFORMED DURING STEADY STATE.....	63
6.6.1	Checkpoint	63
6.6.2	Checkpoint Conditions	63
6.6.3	Checkpoint Implementation	63
6.6.4	Serializable Transactions	63
6.7	MEASUREMENT PERIOD DURATION.....	64
6.8	REGULATION OF TRANSACTION MIX	65
6.9	TRANSACTION MIX.....	65
6.10	TRANSACTION STATISTICS	65
6.11	CHECKPOINT COUNT AND LOCATION	65
7	CLAUSE 6: SUT, DRIVER, AND COMMUNICATION DEFINITION RELATED ITEMS.....	66
7.1	RTE DESCRIPTION.....	66
7.2	LOST CONNECTIONS	68
7.3	EMULATED COMPONENTS.....	68
7.4	FUNCTIONAL DIAGRAMS	68
7.5	NETWORKS	68
7.6	CLIENT SUBSTITUTION.....	68
8	CLAUSE 7: PRICING RELATED ITEMS.....	69
8.1	SYSTEM PRICING	69
8.2	SUPPORT PRICING.....	69
8.2.1	HP Hardware Support	69
8.2.2	HP Software Support	69
8.3	ORACLE CORPORATION STANDARD TECHNICAL SUPPORT.....	69
8.4	AVAILABILITY.....	69
8.5	PRICED SYSTEM CONFIGURATION	70
8.6	THROUGHPUT, PRICE/PERFORMANCE, AND AVAILABILITY DATE.....	70
9	CLAUSE 9: AUDIT RELATED ITEMS.....	71
9.1	AUDITOR'S REPORT	71
10	REPORT AVAILABILITY.....	74
APPENDIX A	CLIENT/SERVER SOURCE	75
A.1	CLIENT FRONT-END	75
CLIENT/CLIENT.C	75	
CLIENT/TUX_TRANSACTION.C.....	84	
CLIENT/MAKEFILE	85	
ORACLE.MK.....	86	
POST_DVRY.C.....	86	
TUX_TRANSACTION.H	87	
A.2	TPC_LIB SOURCE.....	87
LIB/TPCC.H	87	
LIB/KEY_CHARS.H.....	89	
LIB/ERRLOG.C.....	89	
LIB/FMT.C.....	90	
LIB/IOBUF.H.....	92	
LIB/IOBUF.C.....	93	
LIB/RANDOM.C	93	
LIB/MAKEFILE	94	
A.3	TRANSACTION SOURCE.....	94
CLIENT/SERVICE.C	94	
CLIENT/ORACLE/TRANSACTION.C	95	
CLIENT/ORACLE/TPCCPL.C.....	96	

CLIENT/ORACLE/PLNEW.C	101
CLIENT/ORACLE/PLPAY.C	104
CLIENT/ORACLE/PLORD.C	106
CLIENT/ORACLE/PLSTO.C.....	109
CLIENT/ORACLE/PLDEL.C	110
CLIENT/ORACLE/ORA_TPCC.H.....	114
CLIENT/ORACLE/TPCCFLAGS.H.....	117
TPCC_INFO.H	117
TPCCPL.H.....	118
DBWR_ON_INT_CPUS_32LOC.SH	120
INTCTL_32LOC.SH	120
KCTUNE.SH.....	121
ISNR.SH	122
SET_SCHEDNA	122
A.4 SERVER STORED PROCEDURES.....	122
TKVCPDEL.SQL	122
TKVCPNEW.SQL	122
PAYZ.SQL	125
COUNTUSER.....	125
PAYNZ.SQL	125
TPCC.C	126
DELAY.C.....	126
RANDOM.H	126
AUDIT_TABLE.SH	127
COUNTORDERS.SH.....	127
LOGSIZE.SH	127
DN.SH.....	127
UP.SH 127	
TKVCININ.SQL.....	127
/BUILD/TKVININ.SQL	127
APPENDIX B DATABASE DESIGN.....	129
B.1 SCRIPTS	129
ADDFILE.SH	129
ADDS.SH	129
ANALYZE.SH	129
ANALYZE.SQL.....	129
CREATE_CACHE_VIEWS.SQL.....	130
CREATE_SPACESTATS.SQL	130
CREATE_STOREDPROCS.SQL.....	130
CREATEDB.SQL.....	130
CREATEINDEX_ICUST1.SQL	130
CREATEINDEX_ICUST2.SQL	130
CREATEINDEX_IDIST.SQL	131
CREATEINDEX_ITEM.SQL.....	131
CREATEINDEX_INORD.SQL	131
CREATEINDEX_IORDR1.SQL.....	131
CREATEINDEX_IORDR2.SQL.....	131
CREATEINDEX_ISTOK.SQL	131
CREATEINDEX_IWARE.SQL.....	131
CREATEMISC.SH	131
CREATESPACESTATS.SH.....	132
CREATESTATS.SH	132
CREATESTOREDPROCS.SH.....	133
CREATETABLE_CUST.SQL	133
CREATETABLE_DIST.SQL	133

CREATETABLE_HIST.SQL	133
CREATETABLE_ITEM.SQL	134
CREATETABLE_NORD.SQL	134
CREATETABLE_ORDL.SQL	135
CREATETABLE_ORDR.SQL	135
CREATETABLE_STOK.SQL	135
CREATETABLE_WARE.SQL	135
CREATETS.SH	135
CREATEUSER.SH	136
CREATEUSER.SQL	136
DDVIEW.SH	137
DML.SQL	137
DRIVER.SH	137
EXTENT.SQL	138
FREEEXT.SQL	138
INITPAY.SQL	138
LOADCUST.SH	138
LOADDIST.SH	140
LOADHIST.SH	140
LOADITEM.SH	142
LOADNORD.SH	143
LOADORDRORDL.SH	144
LOADSTOK.SH	146
LOADWARE.SH	147
LOCALOPTIONS.SH	148
NEW.SQL	148
P_BUILD.ORA	148
P_CREATE.ORA	149
PAY.SQL	149
PLSQL_MON.SQL	151
PST_C.SQL	151
SHUTDOWNDB.SH	152
SPACE_GET.SQL	152
SPACE_INIT.SQL	153
SPACE_RPT.SQL	153
STARTUPDB.SH	153
STEPENV.SH	153
VIEWS.SQL	155
APPENDIX C TUNABLE PARAMETERS	156
C.1 HP-UX CONFIGURATION - CLIENTS	156
CONFIG/CLIENT2/OSTUNE.VER	156
C.2 HP-UX CONFIGURTION – SERVER	157
CONFIG/SERVER/OSTUNE.VER	157
CONFIG/SERVER/DBTUNE.VER	158
C.3 TUXEDO UBBCONFIG	158
CONFIG/CLIENT2/TMCFG.VER	158
APPENDIX D RTE CONFIGURATION	159
D.1 FIELD VALUE GENERATION	159
SOURCE/SRC/DRIVER/GENERATE.C	159
APPENDIX E DISK STORAGE	160
APPENDIX F PRICE QUOTES.....	162

1 General Items

1.1 Application Code and Definition Statements

The application program (as defined in clause 2.1.7) must be disclosed. This includes, but is not limited to, the code implementing the five transactions and the terminal input output functions.

Appendix A contains the HP C++/ANSI C Developer's Bundle /HP-UX application code used in this TPC-C® test.

1.2 Test Sponsor

A statement identifying the benchmark sponsor(s) and other participating companies must be provided.

The Enterprise Unix Division of Hewlett-Packard Company is the test sponsor of this TPC Benchmark® C.

1.3 Parameter Settings

Settings must be provided for all customer-tunable parameters and options which have been changed from the defaults found in actual products, including but not limited to:

- Database options
- Recover/commit options
- Consistency/locking options
- Operating system and application configuration parameter
- Compilation and linkage options and run-time optimizations used to create/install applications, OS, and/or databases

This requirement can be satisfied by providing a full list of all parameters and options.

The intent of the above clause is that anyone attempting to recreate the benchmark environment has sufficient information to compile, link, optimize, and execute all software used to produce the disclosed benchmark result.

Appendix A contains the application "make" files. Appendix C contains the HP-UX operating system parameters used to generate the kernel for the configuration used in this benchmark. Also included are all of the Oracle Database 10g Release 2 Enterprise Edition with Partitioning database parameters and the TUXEDO 8.1 transaction monitor parameters used.

1.4 Configuration Diagrams

Diagrams of both measured and priced configurations must be provided, accompanied by a description of the differences. This includes, but is not limited to:

- Number and type of processors
- Size of allocated memory, and any specific mapping/partitioning of memory unique to the test
- Number and type of disk units (and controllers, if applicable)
- Number of channels or bus connections to disk units, including the protocol type
- Number of LAN (e.g. Ethernet) connections, including routers, work stations, terminals, etc, that were physically used in the test or are incorporated into the pricing structure (See Clause 8.1.8)
- Type and run-time execution location of software components (e.g. DBMS, client processes, transaction monitors, software drivers, etc)

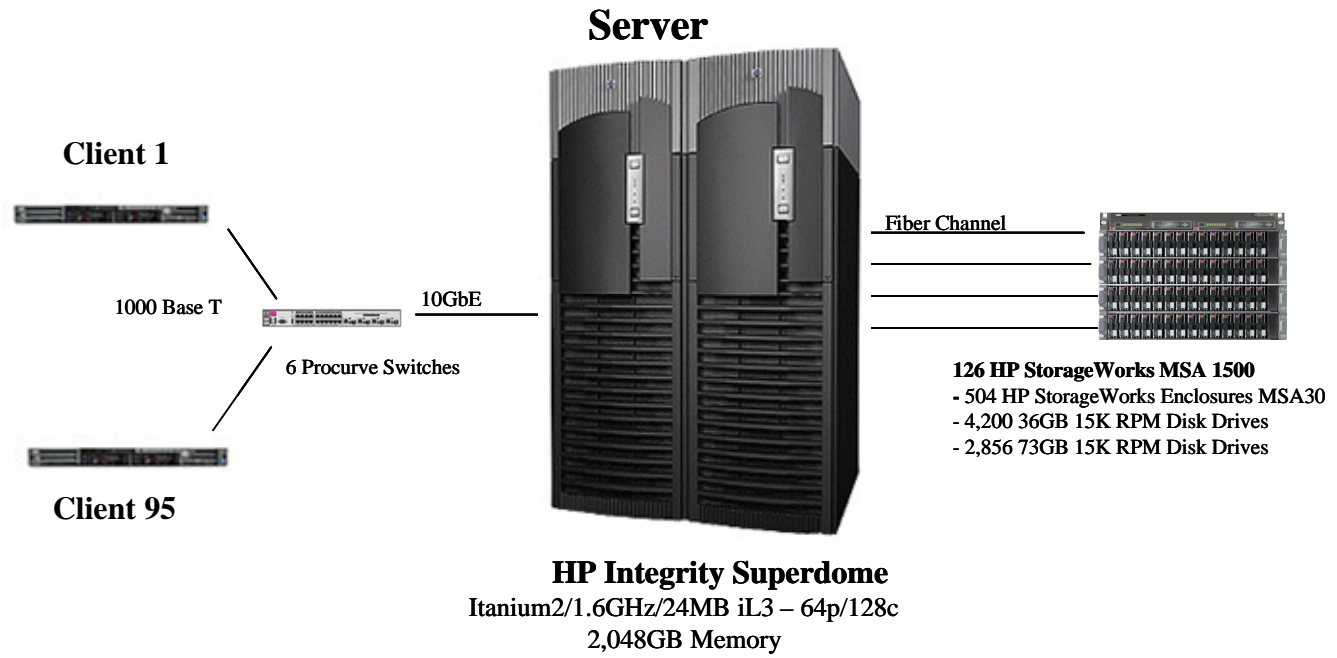
The server System Under Test, an HP Integrity Superdome - Itanium2/1.6GHz/24MB iL3 - 64p/128c depicted in Figure 1.1, consisted of:

- 64 1.6GHz Intel Itanium2 System Processors
- 2048 GB of memory
- 40 PCI-X 4Gb Fibre Channel Adapter (dual port) Adapters
- 126 HP StorageWorks Modular Smart Array 1500 with 4,200 36GB 15K RPM disks and 2,856 73GB 15K RPM disks.
- Five LAN interfaces

As indicated in Figure 1.2, this benchmark configuration used Remote Terminal Emulator (RTE) programs that executed on 72 rx2620 Enterprise Server drivers to emulate TPC-C user sessions. The emulated users on the driver systems were directly connected to the client systems under test via via a 1000 Base-T local area network (LAN) and communicated using TCP/IP. The clients were connected to the SUT either directly through one of the two HP Procurve 3400 switches, or via four HP Procurve 2724 switches that were then connected to two HP Procurve 3400 switches. The HP Procurve 3400 switches were connected to the SUT via 4 10gigabit connections.

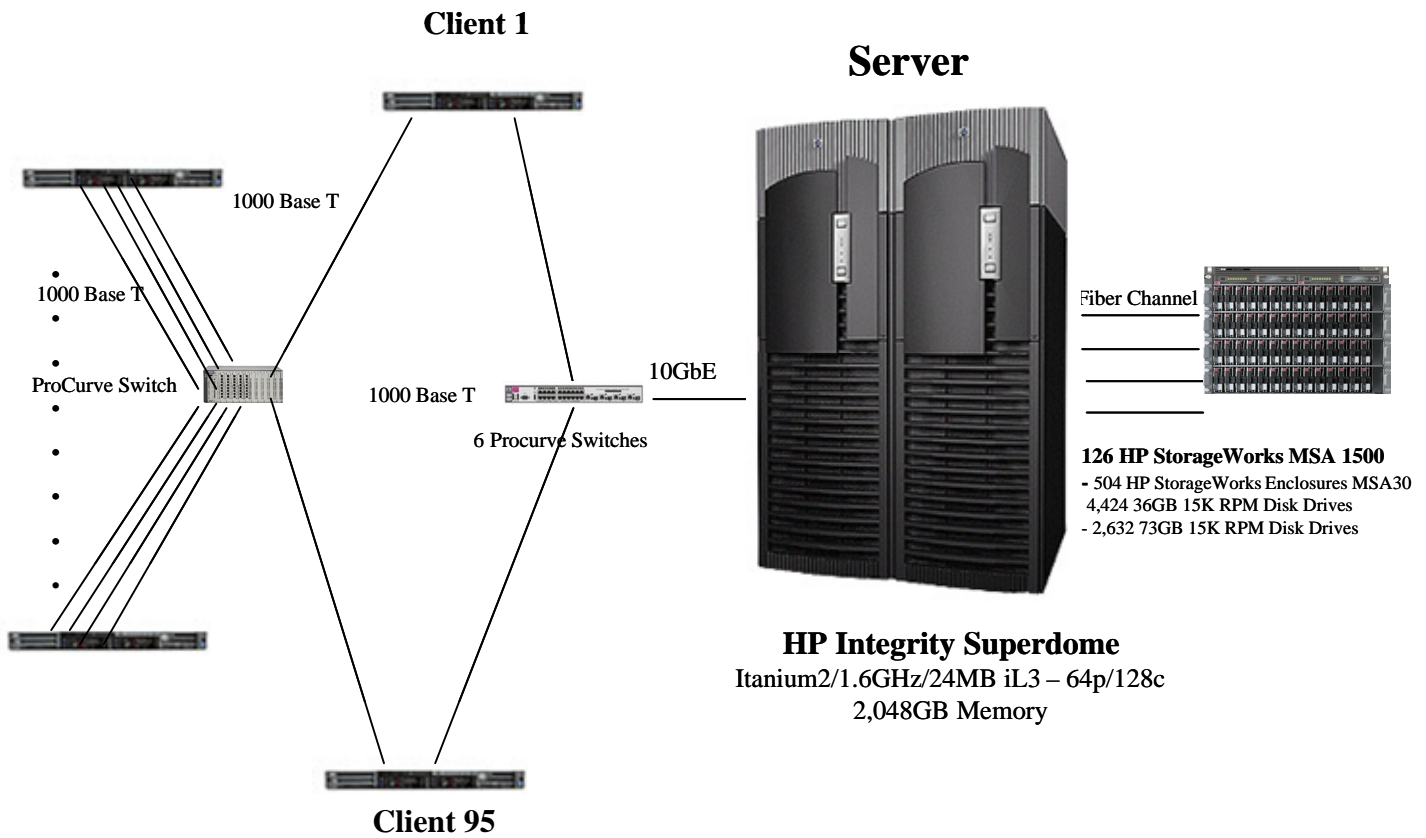
The priced configuration for the HP Integrity Superdome - Itanium2/1.6GHz/24MB iL3 - 64p/128c is shown in Figure 1.1. In the priced configuration, the RTE shown in the benchmark configuration is replaced by the appropriate number of workstations (emulating ANSI terminals) connected to hubs.

Figure 1.1: HP Integrity Superdome Pricing Configuration



Clients – 95 hp rx1620 servers

Figure 1.2: HP Integrity Superdome Benchmark Configuration



Drivers – 72 hp rx2600

Clients – 95 hp rx1620 servers

2 Clause 1: Logical Data Base Design Related Items

2.1 Table Definitions

Listing must be provided for all table definition statements and all other statements used to set up the database.

Appendix B describes the programs that define, create, and populate the Oracle Database 10g Release 2 Enterprise Edition with Partitioning database for TPC-C® testing.

2.2 Physical Organization of Database

The physical organization of tables and indices, within the database, must be disclosed.

Space was allocated to Oracle Database 10g Release 2 Enterprise Edition with Partitioning according to the data in section 5.2. The size of the database table space on each disk drive was calculated to provide even distribution of load across the disk drives.

2.3 Insert and Delete Operations

It must be ascertained that insert and/or delete operations to any of the tables can occur concurrently with the TPC-C® transaction mix. Furthermore, any restrictions in the SUT database implementation that precludes inserts beyond the limits defined in Clause 1.4.11 must be disclosed. This includes the maximum number of rows that can be inserted and the maximum key value for these new rows.

There were no restrictions on insert and delete operations to any tables.

2.4 Partitioning

While there are a few restrictions placed upon horizontal or vertical partitioning of tables and rows in the TPC-C® benchmark, any such partitioning must be disclosed. Replication of tables, if used, must be disclosed. Additional and/or duplicated attributes in any table must be disclosed along with a statement on the impact on performance.

Horizontal partitioning was used for one of the tables and one of indices. The detail of this partitioning can be understood by examining the table and index definition statements in Appendix B.

Replication and additional or duplicated attributes were not used in this implementation.

3 Clause 2: Transaction & Terminal Profiles Related Items

3.1 Random Number Generation

The method of verification for the random number generation must be disclosed.

The random number generator used can be found in the source appendix. It is from the book “The Art of Computer Systems Performance Analysis” by Raj Jain, page 443. The properties of this random number generator are documented in the book. It is a full-period multiplicative linear-congruential random number generator.

3.2 Input/Output Screen Layout

The actual layout of the terminal input/output screens must be disclosed.

The screen layouts corresponded exactly to those in Clauses 2.4.3, 2.5.3, 2.6.3, 2.7.3, and 2.8.3 of the TPC-C[®] Standard Specification.

3.3 Priced Terminal Feature Verification

The method used to verify that the emulated terminals provide all the features described in Clause 2.2.2.4 must be explained. Although not specifically priced, the type and model of the terminals used for the demonstration in 8.1.3.3 must be disclosed and commercially available (including supporting software and maintenance).

The terminal features were verified by manually exercising each specification on an HP 712/80 workstation running an ANSI terminal emulator.

3.4 Presentation Manager or Intelligent Terminal

Any usage of presentation managers or intelligent terminals must be explained.

Application code running on the client implemented the TPC-C user interface. A listing of this code is included in Appendix A. Used capabilities of the terminal beyond basic ASCII entry and display were restricted to cursor positioning.

A presentation manager was not used.

Table 3.1: Transaction Statistics

Type	Item	Value
New Order	Home warehouse items	99.00%
	Remote warehouse items	1.00%
	Rolled back transactions	1.00%
	Average items per order	10.00
Payment	Home warehouse	85.00%
	Remote warehouse	15.00%
	Non primary key access	60.00%
Order Status	Non primary key access	60.00%
Delivery	Skipped transactions	0
Transaction Mix	New Order	44.95%
	Payment	43.01%
	Order Status	4.01%
	Delivery	4.02%
	Stock Level	4.01%

3.5 Transaction Statistics

Table 3.1 lists the numerical quantities that Clauses 8.1.3.5 to 8.1.3.11 require.

3.6 Queuing Mechanism

The queuing mechanism used to defer the execution of the Delivery transaction must be disclosed.

Delivery transactions were submitted to servers using the same TUXEDO mechanism that other transactions used. The only difference was that the call was asynchronous, i.e., control would return to the client process immediately and the deferred delivery part would complete asynchronously.

4 Clause 3: Transaction and System Properties Related Items

4.1 Transaction System Properties (ACID)

The results of the ACID tests must be disclosed along with a description of how the ACID requirements were met. This includes disclosing which case was followed for the execution of Isolation Test 7.

The TPC Benchmark® C Standard Specification defines a set of transaction processing system properties that a system under test (SUT) must support during the execution of the benchmark. Those properties are Atomicity, Consistency, Isolation, and Durability (ACID). This section quotes the specification definition of each of these properties and describes the tests done as specified and monitored by the auditor to demonstrate compliance.

4.2 Atomicity

The system under test must guarantee that transactions are atomic; the system will either perform all individual operations on the data, or will assure that no partially-completed operations leave any effects on the data.

4.2.1 Completed Transaction

Perform the Payment transaction for a randomly selected warehouse, district, and customer (by customer number as specified in Clause 2.5.1.2) and verify that the records in the CUSTOMER, WAREHOUSE, and DISTRICT tables have been changed appropriately.

These tests were performed on a system configured for 322,620 warehouses. The tests were completed using the same OS (HP-UX 11i v3) and RDBMS (Oracle Database 10g Release 2 Enterprise Edition with Partitioning).

The values of w_ytd, d_ytd, c_balance, c_ytd_payment, and c_payment_cnt of a randomly selected warehouse, district, and customer were retrieved. The Payment transaction was executed on the same warehouse, district, and customer. The transaction was committed. The values w_ytd, d_ytd, c_balance, c_ytd_payment, and c_payment_cnt were retrieved again. It was verified that all values had been changed appropriately.

4.2.2 Aborted Transaction

Perform the Payment transaction for a randomly selected warehouse, district, and customer (by customer number as specified in Clause 2.5.1.2) and substitute a ROLLBACK of the transaction for the COMMIT of the transaction. Verify that the records in the CUSTOMER, WAREHOUSE, and DISTRICT tables have NOT been changed

These tests were performed on a system configured for 322,620 warehouses. The tests were completed using the same OS (HP-UX 11i v3) and RDBMS (Oracle Database 10g Release 2 Enterprise Edition with Partitioning).

The values of w_ytd, d_ytd, c_balance, c_ytd_payment and c_payment_cnt of a randomly selected warehouse, district, and customer were retrieved. The Payment transaction was executed on the same warehouse, district, and customer. The transaction was rolled back. The values of w_ytd, d_ytd, c_balance, c_ytd_payment, c_payment_cnt were retrieved again. It was verified that none of the values had changed.

4.3 Consistency

Consistency is the property of the application that requires any execution of a database transaction to take the database from one consistent state to another assuming the database is initially in a consistent state.

These tests were performed on a system configured for 322,620 warehouses. The tests were completed using the same OS (HP-UX 11i v3) and RDBMS (Oracle Database 10g Release 2 Enterprise Edition with Partitioning).

The TPC Benchmark C standard requires the System Under Test to meet the following 12 consistency conditions (c.f. *TPC Standard Specification, Clauses 3.3.2.1 to 3.3.2.12*):

1. the sum of the district balances in a warehouse is equal to the warehouse balance;
2. for each district, the next order-id minus one is equal to maximum order-id in the ORDER table and equal to the maximum new-order-id in the NEW-ORDER table;
3. for each district, the maximum order-id minus minimum order-id in the ORDER table plus one equals the number of rows in the NEW-ORDER table for that district;
4. for each district, the sum of the order-line counts equals the number of rows in the ORDER-LINE table for that district;
5. for each row in the ORDER table, the carrier-id is set to a null value only if there is a corresponding row in the NEW-ORDER table;
6. for each row in the ORDER table, the order-line count must equal the number of rows in the ORDER-LINE table for that order;
7. for any row in the ORDER-LINE table, the delivery date/time is set to a null value only if the corresponding row in the ORDER table has the carrier-id set to a null value;
8. for each warehouse, the year-to-date amount must equal the sum of the amounts in the HISTORY table for that warehouse;
9. for each district, the year-to-date amount must equal the sum of the amounts in the HISTORY table for that district;
10. for each customer, the balance must equal the sum of the order-line amount minus the sum of the history amount for that customer;
11. for each district, the total orders minus the total new-orders must equal the sum of the customer delivery count;
12. for any randomly selected customer, the balance plus the year-to-date payment must equal the sum of the order-line amount.

The TPC Benchmark C Standard Specification requires explicit demonstration that the conditions are satisfied for the first four conditions only.

To demonstrate that consistency is maintained, conditions 1-4 were verified for a sample of warehouses before and after the durability tests.

4.4 Isolation

Operations of concurrent transactions must yield results which are indistinguishable from the results which would be obtained by forcing each transaction to be serially executed to completion in some order.

*This property is commonly called **serializability**. Sufficient conditions must be enabled at either the system or application level to ensure serializability of transactions under any arbitrary mix of TPC-C transactions, unless otherwise specified by the transaction profile. The system or application must have full serializability enabled (i.e., repeated reads of the same rows within any committed transaction must return identical data when run concurrently with any arbitrary mix of TPC-C transactions), except in the case of Stock-Level transaction. For the Stock-Level transaction, the isolation requirement is relaxed to simply require that the transaction see only committed data.*

The TPC Benchmark C Standard (Revision 3.5) defines nine required tests to be performed to demonstrate that the required levels of transaction isolation are met.

These tests were performed on a system configured for 322,620 warehouses. The tests were completed using the same OS (HP-UX 11i v3) and RDBMS (Oracle Database 10g Release 2 Enterprise Edition with Partitioning).

For conventional locking schemes, isolation should be tested as described below. Systems that implement other isolation schemes may require different validation techniques. It is the responsibility of the test sponsor to disclose those techniques and the tests for them. If isolation schemes other than conventional locking are used, it is permissible to implement these tests differently provided full details are disclosed. (Examples of different validation techniques are shown in Isolation Test 7, Clause 3.4.2.7).

4.4.1 Isolation Test 1

This test demonstrates isolation for read-write conflicts of Order-Status and New-Order transactions.

The execution of the above test proceeded as follows:

1. An Order-Status transaction T0 was executed for a randomly selected customer, and the order returned was noted. T0 was committed
2. A New-Order transaction T1 was started for the same customer used in T0. T1 was stopped prior to COMMIT.
3. An Order-Status transaction T2 was started for the same customer used in T1. T2 completed and was committed without being blocked by T1. T2 returned the same order that T0 had returned.
4. T1 was allowed to complete and was committed.
5. An Order-Status transaction T3 was started for the same customer used in T1. T3 returned the order inserted by T1.

This outcome demonstrates serialization of T2 before T1. It has equivalent validity to the outcome specified in the Standard which supposes T1 to be serialized before T2.

4.4.2 Isolation Test 2

This test demonstrates isolation for read-write conflicts of Order-Status and New-Order transactions when the New-Order transaction is ROLLED BACK.

The execution of the above test proceeded as follows:

1. An Order-Status transaction T0 was executed for a randomly selected customer and the order returned was noted. T0 was committed.
2. A New-Order transaction T1 with an invalid item number, was started for the same customer used in T0. T1 was stopped immediately prior to ROLLBACK.
3. An Order-Status transaction T2 was started for the same customer used in T1. T2 completed and was committed without being blocked by T1. T2 returned the same order that T0 had returned.
4. T1 was allowed to ROLLBACK.
5. An Order-Status transaction T3 was started for the same customer used in T1. T3 returned the same order that T0 had returned.

4.4.3 Isolation Test 3

This test demonstrates isolation for write-write conflicts of two New-Order transactions.

The execution of the above test proceeded as follows:

1. The D_NEXT_O_ID of a randomly selected district was retrieved.
2. A New-Order transaction T1 was started for a randomly selected customer within the district used in step 1. T1 was stopped immediately prior to COMMIT.
3. Another New-Order transaction T2 was started for the same customer used in T1. T2 waited.
4. T1 was allowed to complete. T2 completed and was committed.
5. The order number returned by T1 was the same as the D_NEXT_O_ID retrieved in step 1. The order number returned by T2 was one greater than the order number returned by T1.
6. The D_NEXT_O_ID of the same district was retrieved again. It had been incremented by two (i.e. it was one greater than the order number returned by T2).

4.4.4 Isolation Test 4

This test demonstrates isolation for write-write conflicts of two New-Order transactions when one transaction is ROLLED BACK.

The execution of the above test proceeded as follows:

1. The D_NEXT_O_ID of a randomly selected district was retrieved.
2. A New-Order transaction T1, with an invalid item number, was started for a randomly selected customer within the district used in step 1. T1 was stopped immediately prior to ROLLBACK.
3. Another New-Order transaction T2 was started for the same customer used in T1. T2 waited.
4. T1 was allowed to roll back, and T2 completed and was committed.
5. The order number returned by T2 was the same as the D_NEXT_O_ID retrieved in step 1.
6. The D_NEXT_O_ID of the same district was retrieved again. It had been incremented by one (i.e. one greater than the order number returned by T2).

4.4.5 Isolation Test 5

This test demonstrates isolation for write-write conflicts of Payment and Delivery transactions.

The execution of the above test proceeded as follows:

1. A query was executed to find out the customer who would be updated by the next delivery transaction for a randomly selected warehouse and district.
2. The C_BALANCE of the customer found in step 1 was retrieved.
3. A Delivery business transaction T1 was started for the same warehouse used in step 1. T1 was stopped immediately prior to the COMMIT of the database transaction corresponding to the district used in step 1.
4. A Payment transaction T2 was started for the same customer found in step 1. T2 waited.
5. T1 was allowed to complete. T2 completed and was committed.
6. The C_BALANCE of the customer found in step 1 was retrieved again. The C_BALANCE reflected the results of both T1 and T2.

4.4.6 Isolation Test 6

This test demonstrates isolation for write-write conflicts of Payment and Delivery transactions when the Delivery transaction is ROLLED BACK.

The execution of the above test proceeded as follows:

1. A query was executed to find out the customer who would be updated by the next delivery transaction for a randomly selected warehouse and district.
2. The C_BALANCE of the customer found in step 1 was retrieved.
3. A Delivery business transaction T1 was started for the same warehouse used in step 1. T1 was stopped immediately prior to the ROLLBACK of the database transaction corresponding to the district used in step 1.
4. A Payment transaction T2 was started for the same customer found in step 1. T2 waited.
5. T1 was allowed to ROLLBACK. T2 completed and was committed.

The C_BALANCE of the customer found in step 1 was retrieved again. The C_BALANCE reflected the results of only T2.

4.4.7 Isolation Test 7

This test demonstrates repeatable reads for the New-Order transaction while an interactive transaction updates the price of an item.

The execution of the above test proceeded as follows:

1. The I_PRICE of two randomly selected items X and Y were retrieved.
2. A New-Order transaction T2 with a group of items including items X and Y was started. T2 was stopped immediately after retrieving the prices of all items. The prices of items X and Y retrieved matched those retrieved in step 1.
3. A transaction T3 was started to increase the price of items X and Y by 10%.
4. T3 did not stall and no transaction was rolled back. T3 was committed.
5. T2 was resumed, and the prices of all items were retrieved again within T2. The prices of items X and Y matched those retrieved in step 1.
6. T2 was committed.
7. The prices of items X and Y were retrieved again. The values matched the values set by T3.

- Execution followed *Case D* of *Clause 3.4.2.7*.

4.4.8 Isolation Test 8

This test demonstrates isolation for phantom protection between New-Order and Order-Status transactions.

The execution of the above test proceeded as follows:

1. An Order-Status transaction T1 was started for a randomly selected customer.
2. T1 was stopped immediately after reading the order table for the selected customer. The most recent order for that customer was found.

3. A New-Order transaction T2 was started for the same customer. T2 completed and was committed without being blocked by T1.
4. T1 was resumed and the ORDER table was read again to determine the most recent order for the same customer. The order found was the same as the one found in step 2.
5. T1 completed and was committed.

4.4.9 Isolation Test 9

This test demonstrates isolation for phantom protection between New-Order and Delivery transactions.

The execution of the above test proceeded as follows:

1. The NO_D_ID of all new_ORDER rows for a randomly selected warehouse and district was changed to 11. The changes were committed.
2. A Delivery transaction T1 was started for the selected warehouse.
3. T1 was stopped immediately after reading the new_ORDER table for the selected warehouse and district. No qualifying row was found.
4. A New-Order transaction T2 was started for the same warehouse and district. T2 completed and was committed without being blocked by T1.
5. T1 was resumed and the new_ORDER table was read again. No qualifying row was found.
6. T1 completed and was committed.
7. The NO_D_ID of all new_ORDER rows for the selected warehouse and district was restored to the original value. The changes were committed.

4.5 Durability

The tested system must guarantee durability: the ability to preserve the effects of committed transaction and insure database consistency after recovery from any one of the failures listed in Clause 3.5.3.

List of single failures:

- *Permanent irrecoverable failure of any single durable medium containing TPC-C database tables or recovery log data.*
- *Instantaneous interruption (system crash / system hang) in processing which requires system reboot to recover.*
- *Failure of all or part of memory (loss of contents)...*

Specified durability tests were executed to demonstrate satisfaction of the durability requirements for this implementation of TPC Benchmark C. One durability test, described below, covering the following failure situations was performed under the auditor's supervision:

- *Permanent irrecoverable failure of any single durable medium containing TPC-C database tables or recovery log data (Clause 3.5.3.1).*

These tests were performed on a system configured for 322,620 warehouses. The tests were completed using the same OS (HP-UX 11i v3) and RDBMS (Oracle Database 10g Release 2 Enterprise Edition with Partitioning).

Both tests were performed under a load of 3,226,200 users on the full-scale database for the loss of recovery log and loss of data tests. Another durability test, described below, combining the following failure situations was performed under the auditor's supervision:

- *instantaneous interruption which requires system reboot [of processors] to recover. (Clause 3.5.3.2)*
- *failure of all or part of memory. (Clause 3.5.3.3).*

This test was performed under the full performance-measurement load of 3,226,200 users on the full-scale database.

4.5.1 Loss of Log and Data Disks

Because the log devices are Redundant Disk Arrays which each function independently of the rest of the system in ensuring data integrity under loss and/or replacement of any individual disk drive (and other failures as well), integrity under such failure and replacement does not entail any interruption in processing. The data devices are not mirrored and must be restored from backup after failure and subsequent transactions applied from the transaction logs. The tests below validates the durability by demonstrating persistence of the results of transactions processed both before and during these failures, validating the durability upon database recovery (in this instance, forced) of transactions which completed before the failure and the non-effect of transactions which did not complete.

1. The D_NEXT_O_ID fields for all rows in the DISTRICT table were summed up to determine the initial count of the total number of orders (count1).
2. A test was initiated with 3,226,200 terminals. On the driver system, completed/rolled-back transactions (including New-Orders) were recorded in a "success" file.
3. After running at a minimum of 10% of steady state throughput levels for 5 minutes, an individual disk containing recovery log was unplugged from an array.
4. Because of the built-in redundancy in the disk array, the test continued normally.
5. After running again at a minimum of 20% of steady state throughput levels for 5 minutes, an individual disk containing data was unplugged from an array.
6. The test failed with Oracle reporting errors when accessing the files on that array.
7. The disk was replaced.
8. The database files on that array were restored from backup.
9. Oracle was restarted and the transactions in the log were applied to the database.
10. The contents of the "success" file on the driver and the ORDER table were spot-compared to verify that records in the "success" file for completed New-Order transactions had corresponding records in the ORDER table.
11. Step 1 was repeated to determine the current total number of orders (count2). It was verified that count2-count1 was equal to or greater than the number of records for successful New Orders in the RTE "success" file. *This difference would be due only to transactions which were committed on the system under test but for which the output data was not displayed on the [emulated] input/output screen before the failure.*
12. Consistency test 3 was run on the database and the results were verified.

4.5.2 Instantaneous Interruption and Loss of Memory

Instantaneous interruption and loss of memory tests were combined because the loss of power erases the contents of memory. This failure was induced while the benchmark was running by turning off the power supplies to the server.

1. The D_NEXT_O_ID fields for all rows in district table were summed up to determine the initial count of the total number of orders (count1).
2. Transactions were started at full load. On the driver system, completed/rolled-back transactions (including New-Orders) were recorded in a "success" file.
3. After five minutes at steady state throughout, the server system was de-powered.
4. The test was aborted on the driver.
5. The server system was restarted.
6. The database was restarted and a recovery performed using the transaction log.
7. The contents of the "success" file on the driver and the ORDERs table were spot-compared to verify that records in the "success" file for completed New-Order transactions had corresponding records in the ORDERs table.
8. Step 1 was repeated to determine the current total number of orders (count2). It was verified that count2-count1 was equal to or greater than the number of records for successful New Orders in the RTE "success" file. *This difference would be due only to transactions which were committed on the system under test but for which the output data was not displayed on the [emulated] input/output screen before the failure.*
9. Consistency test 3 was run on the database and the results were verified.

5 Clause 4: Scaling and Data Base Population Related Items

5.1 Initial Cardinality of Tables

The cardinality (e.g. number of rows) of each table, as it existed at the start of the benchmark run, must be disclosed. If the database was overscaled and inactive rows of the WAREHOUSE table were deleted the cardinality of the WAREHOUSE table as initially configured and the number of rows deleted must be disclosed.

The TPC-C database for this test was configured with 349,440 warehouses.

Table	Occurrences
Warehouse	349,440
District	3,494,400
Customer	10,483,200,000
History	10,483,200,000
Orders	10,483,200,000
New Orders	3,144,960,000
Order Line	104,834,968,064
Item	100,000
Stock	34,944,000,000

During the measurement all of the warehouses and their associated data were accessed. This was confirmed using D_NEXT_O_ID and W_YTD as described in *Clause 4.2.2 Comment (2)*.

5.2 Database and Growth Layout

The distribution of tables and logs across all media must be explicitly depicted for tested and priced systems.

Table 5.2 indicates the distribution of the database tables over the disks of the tested and priced systems.

I) root, swap, file systems:

=====

The tested system was connected to 1 SCSI disk enclosure with 10 17GB disk drives, one of which is used for root & primary swap and one is for /project. The priced system has 1 36GB drive to hold the operating system root and any file systems.

Device path	Name	Contents	Size	Device
0/0/8/1/0.0.0	c1t0d0	Unused file system	17GB	ST318203LC
0/0/8/1/0.1.0	c1t1d0	Unused file system	17GB	ST318203LC
0/0/8/1/0.2.0	c1t2d0	Unused file system	17GB	ST318203LC
0/0/8/1/0.3.0	c1t3d0	Unused file system	17GB	ST318203LC
0/0/8/1/0.4.0	c1t4d0	Unused file system	17GB	ST318203LC
0/0/8/1/0.8.0	c1t8d0	/project	17GB	ST318203LC
0/0/8/1/0.9.0	c1t9d0	Unused file system	17GB	ST318203LC
0/0/8/1/0.10.0	c1t10d0	Unused file system	17GB	ST318203LC
0/0/8/1/0.11.0	c1t11d0	Unused file system	17GB	ST318203LC
0/0/8/1/0.12.0	c1t12d0	root and primary swap	17GB	ST318203LC

We also use space on the data arrays for secondary swaps.

II) Database files:

=====

1) Connectivity

We use 118 data arrays and 8 log arrays. Each disk array has two controllers, each of which is independently connected to the server either directly in case of log arrays, or through a SAN fabric in case of data arrays. We refer to the two controllers as "left" and "right" controllers.

The server has 64 Fibre Channel ports that connect it to the data arrays. 32 ports are connected to a 64-port "left core" switch, the other 32 ports are connected to the 64-port "right core" switch. The 118 data arrays are divided into 30 quads: 29 quads with 4 data arrays, and a 30th quads with only 2 arrays. Each quad has two 4Gb 8-port Fibre Channel switches: one "left rack" switch and one "right rack" switch. The quad's "left rack" switch connects to the left controllers of the 4 arrays in the quad as well as to the 64-port left core switch. The right rack switch has similar connectivity.

The 8 log arrays were connected directly to the host via 16 4Gb Fibre Channel links.

2) Log files:

4 of the log arrays contain 56 36.4-billion-byte=33.9GB disk drives for a total formatted capacity of 1,899.2GB each. The other 4 log arrays have 56 73-billion-byte=67.8GB disk drives for a total formatted capacity of 3,798.6GB each. That's the capacity before mirroring. We price 8 arrays with 56 67.8GB drives to satisfy the 8 hours log requirement.

The Oracle visible logs are two 650000MB logical volumes striped over 8 arrays. Each array is subdivided into 14 RAID1 LUNs with 65538MB of capacity. Each LUN is introduced to the hp-ux LVM subsystem as a

separate device. So, the vglog8 Volume Group has 112 65538MB devices, over which each of the two 650000MB log file logical volumes is striped.

So, most of the space in vglog8 is left unallocated but is used to satisfy the 8-hour log requirements.

The 112 vglog devices are:

Disk Array	Special device file	Capacity
A.Q10.37	/dev/disk/disk3566	65538MB
A.Q10.37	/dev/disk/disk3578	65538MB
A.Q10.37	/dev/disk/disk3580	65538MB
A.Q10.37	/dev/disk/disk3581	65538MB
A.Q10.37	/dev/disk/disk3582	65538MB
A.Q10.37	/dev/disk/disk3583	65538MB
A.Q10.37	/dev/disk/disk3584	65538MB
A.Q10.37	/dev/disk/disk3585	65538MB
A.Q10.37	/dev/disk/disk3586	65538MB
A.Q10.37	/dev/disk/disk3587	65538MB
A.Q10.37	/dev/disk/disk3588	65538MB
A.Q10.37	/dev/disk/disk3589	65538MB
A.Q10.37	/dev/disk/disk3590	65538MB
A.Q10.37	/dev/disk/disk3591	65538MB
A.Q11.41	/dev/disk/disk3537	65538MB
A.Q11.41	/dev/disk/disk3546	65538MB
A.Q11.41	/dev/disk/disk3551	65538MB
A.Q11.41	/dev/disk/disk3553	65538MB
A.Q11.41	/dev/disk/disk3554	65538MB
A.Q11.41	/dev/disk/disk3555	65538MB
A.Q11.41	/dev/disk/disk3556	65538MB
A.Q11.41	/dev/disk/disk3557	65538MB
A.Q11.41	/dev/disk/disk3558	65538MB
A.Q11.41	/dev/disk/disk3559	65538MB
A.Q11.41	/dev/disk/disk3560	65538MB
A.Q11.41	/dev/disk/disk3561	65538MB
A.Q11.41	/dev/disk/disk3562	65538MB
A.Q11.41	/dev/disk/disk3563	65538MB

A.Q12.45	/dev/disk/disk3510	65538MB
A.Q12.45	/dev/disk/disk3518	65538MB
A.Q12.45	/dev/disk/disk3524	65538MB
A.Q12.45	/dev/disk/disk3525	65538MB
A.Q12.45	/dev/disk/disk3526	65538MB
A.Q12.45	/dev/disk/disk3527	65538MB
A.Q12.45	/dev/disk/disk3528	65538MB
A.Q12.45	/dev/disk/disk3529	65538MB
A.Q12.45	/dev/disk/disk3530	65538MB
A.Q12.45	/dev/disk/disk3531	65538MB
A.Q12.45	/dev/disk/disk3532	65538MB
A.Q12.45	/dev/disk/disk3533	65538MB
A.Q12.45	/dev/disk/disk3534	65538MB
A.Q12.45	/dev/disk/disk3535	65538MB
A.Q13.49	/dev/disk/disk3481	65538MB
A.Q13.49	/dev/disk/disk3490	65538MB
A.Q13.49	/dev/disk/disk3496	65538MB
A.Q13.49	/dev/disk/disk3497	65538MB
A.Q13.49	/dev/disk/disk3498	65538MB
A.Q13.49	/dev/disk/disk3499	65538MB
A.Q13.49	/dev/disk/disk3500	65538MB
A.Q13.49	/dev/disk/disk3501	65538MB
A.Q13.49	/dev/disk/disk3502	65538MB
A.Q13.49	/dev/disk/disk3503	65538MB
A.Q13.49	/dev/disk/disk3504	65538MB
A.Q13.49	/dev/disk/disk3505	65538MB
A.Q13.49	/dev/disk/disk3506	65538MB
A.Q13.49	/dev/disk/disk3507	65538MB
B.Q11.41	/dev/disk/disk3592	65538MB
B.Q11.41	/dev/disk/disk3593	65538MB
B.Q11.41	/dev/disk/disk3594	65538MB
B.Q11.41	/dev/disk/disk3595	65538MB
B.Q11.41	/dev/disk/disk3596	65538MB
B.Q11.41	/dev/disk/disk3597	65538MB
B.Q11.41	/dev/disk/disk3598	65538MB
B.Q11.41	/dev/disk/disk3599	65538MB
B.Q11.41	/dev/disk/disk3600	65538MB
B.Q11.41	/dev/disk/disk3601	65538MB
B.Q11.41	/dev/disk/disk3602	65538MB
B.Q11.41	/dev/disk/disk3603	65538MB
B.Q11.41	/dev/disk/disk3604	65538MB
B.Q11.41	/dev/disk/disk3605	65538MB
B.Q11.42	/dev/disk/disk3466	65538MB
B.Q11.42	/dev/disk/disk3467	65538MB
B.Q11.42	/dev/disk/disk3468	65538MB
B.Q11.42	/dev/disk/disk3469	65538MB
B.Q11.42	/dev/disk/disk3470	65538MB
B.Q11.42	/dev/disk/disk3471	65538MB
B.Q11.42	/dev/disk/disk3472	65538MB
B.Q11.42	/dev/disk/disk3473	65538MB
B.Q11.42	/dev/disk/disk3474	65538MB
B.Q11.42	/dev/disk/disk3475	65538MB

```

B.Q11.42 /dev/disk/disk3476 65538MB
B.Q11.42 /dev/disk/disk3477 65538MB
B.Q11.42 /dev/disk/disk3478 65538MB
B.Q11.42 /dev/disk/disk3479 65538MB
B.Q11.43 /dev/disk/disk3409 65538MB
B.Q11.43 /dev/disk/disk3410 65538MB
B.Q11.43 /dev/disk/disk3411 65538MB
B.Q11.43 /dev/disk/disk3412 65538MB
B.Q11.43 /dev/disk/disk3413 65538MB
B.Q11.43 /dev/disk/disk3414 65538MB
B.Q11.43 /dev/disk/disk3415 65538MB
B.Q11.43 /dev/disk/disk3416 65538MB
B.Q11.43 /dev/disk/disk3417 65538MB
B.Q11.43 /dev/disk/disk3418 65538MB
B.Q11.43 /dev/disk/disk3458 65538MB
B.Q11.43 /dev/disk/disk3459 65538MB
B.Q11.43 /dev/disk/disk3460 65538MB
B.Q11.43 /dev/disk/disk3461 65538MB
B.Q11.44 /dev/disk/disk3394 65538MB
B.Q11.44 /dev/disk/disk3395 65538MB
B.Q11.44 /dev/disk/disk3396 65538MB
B.Q11.44 /dev/disk/disk3397 65538MB
B.Q11.44 /dev/disk/disk3398 65538MB
B.Q11.44 /dev/disk/disk3399 65538MB
B.Q11.44 /dev/disk/disk3400 65538MB
B.Q11.44 /dev/disk/disk3401 65538MB
B.Q11.44 /dev/disk/disk3402 65538MB
B.Q11.44 /dev/disk/disk3403 65538MB
B.Q11.44 /dev/disk/disk3404 65538MB
B.Q11.44 /dev/disk/disk3405 65538MB
B.Q11.44 /dev/disk/disk3406 65538MB
B.Q11.44 /dev/disk/disk3407 65538MB

```

Oracle name	file	Size MB	in Raw/VG	Device(s)
Log1		650,000	VG	8 MSA1500 arrays
Log2		650,000	VG	8 MSA1500 arrays

3) Data files:

75 of the data arrays contain 56 36.4-billion-byte=33.9GB disk drives for a total formatted capacity of 1,899.2GB each. The other 43 data arrays have 56 73-billion-byte=67.8GB disk drives for a total formatted capacity of 3,798.6GB each.

The 118 data arrays were identically configured with 14 RAID0 LUNs visible to Oracle. 12 RAID0 LUNs had a capacity of 16386MB, and were used as "raw" devices for Oracle files. Two other RAID0 LUNs had a capacity of

87039MB each, and contained two volume groups (VGs). A number of the Oracle files were created as logical volumes (LVs), striped across one of these VGs.

The 236 vgdata1 and vgdata2 devices are:

Disk Array	Special device file	Capacity
A.Q1.1	/dev/disk/disk4975	87039MB
A.Q1.1	/dev/disk/disk4976	87039MB
A.Q1.2	/dev/disk/disk4584	87039MB
A.Q1.2	/dev/disk/disk4591	87039MB
A.Q1.3	/dev/disk/disk5467	87039MB
A.Q1.3	/dev/disk/disk5471	87039MB
A.Q1.4	/dev/disk/disk6928	87039MB
A.Q1.4	/dev/disk/disk6929	87039MB
A.Q10.38	/dev/disk/disk5020	87039MB
A.Q10.38	/dev/disk/disk5022	87039MB
A.Q10.39	/dev/disk/disk4981	87039MB
A.Q10.39	/dev/disk/disk5001	87039MB
A.Q10.40	/dev/disk/disk5939	87039MB
A.Q10.40	/dev/disk/disk5942	87039MB
A.Q11.42	/dev/disk/disk6190	87039MB
A.Q11.42	/dev/disk/disk6207	87039MB
A.Q11.43	/dev/disk/disk6210	87039MB
A.Q11.43	/dev/disk/disk6216	87039MB
A.Q11.44	/dev/disk/disk6263	87039MB
A.Q11.44	/dev/disk/disk6266	87039MB
A.Q12.46	/dev/disk/disk5226	87039MB
A.Q12.46	/dev/disk/disk5228	87039MB
A.Q12.47	/dev/disk/disk4222	87039MB
A.Q12.47	/dev/disk/disk4226	87039MB
A.Q12.48	/dev/disk/disk4276	87039MB
A.Q12.48	/dev/disk/disk5396	87039MB
A.Q13.50	/dev/disk/disk5074	87039MB
A.Q13.50	/dev/disk/disk5196	87039MB
A.Q13.51	/dev/disk/disk5191	87039MB
A.Q13.51	/dev/disk/disk5614	87039MB
A.Q13.52	/dev/disk/disk4285	87039MB
A.Q13.52	/dev/disk/disk5343	87039MB
A.Q2.5	/dev/disk/disk3831	87039MB
A.Q2.5	/dev/disk/disk3832	87039MB
A.Q2.6	/dev/disk/disk3613	87039MB
A.Q2.6	/dev/disk/disk3616	87039MB
A.Q2.7	/dev/disk/disk3625	87039MB
A.Q2.7	/dev/disk/disk3627	87039MB
A.Q2.8	/dev/disk/disk3655	87039MB
A.Q2.8	/dev/disk/disk3664	87039MB
A.Q3.10	/dev/disk/disk3818	87039MB
A.Q3.10	/dev/disk/disk3820	87039MB
A.Q3.11	/dev/disk/disk3767	87039MB

A.Q3.11	/dev/disk/disk3772	87039MB
A.Q3.12	/dev/disk/disk3986	87039MB
A.Q3.12	/dev/disk/disk4005	87039MB
A.Q3.9	/dev/disk/disk3895	87039MB
A.Q3.9	/dev/disk/disk3896	87039MB
A.Q4.13	/dev/disk/disk4063	87039MB
A.Q4.13	/dev/disk/disk4067	87039MB
A.Q4.14	/dev/disk/disk4049	87039MB
A.Q4.14	/dev/disk/disk4074	87039MB
A.Q4.15	/dev/disk/disk3934	87039MB
A.Q4.15	/dev/disk/disk3942	87039MB
A.Q4.16	/dev/disk/disk4044	87039MB
A.Q4.16	/dev/disk/disk4060	87039MB
A.Q5.17	/dev/disk/disk4874	87039MB
A.Q5.17	/dev/disk/disk4881	87039MB
A.Q5.18	/dev/disk/disk4512	87039MB
A.Q5.18	/dev/disk/disk4533	87039MB
A.Q5.19	/dev/disk/disk4657	87039MB
A.Q5.19	/dev/disk/disk4671	87039MB
A.Q5.20	/dev/disk/disk4601	87039MB
A.Q5.20	/dev/disk/disk4627	87039MB
A.Q6.21	/dev/disk/disk4729	87039MB
A.Q6.21	/dev/disk/disk4746	87039MB
A.Q6.22	/dev/disk/disk6025	87039MB
A.Q6.22	/dev/disk/disk6027	87039MB
A.Q6.23	/dev/disk/disk5436	87039MB
A.Q6.23	/dev/disk/disk5438	87039MB
A.Q6.24	/dev/disk/disk3732	87039MB
A.Q6.24	/dev/disk/disk3736	87039MB
A.Q7.25	/dev/disk/disk6028	87039MB
A.Q7.25	/dev/disk/disk6030	87039MB
A.Q7.26	/dev/disk/disk4094	87039MB
A.Q7.26	/dev/disk/disk5693	87039MB
A.Q7.27	/dev/disk/disk4201	87039MB
A.Q7.27	/dev/disk/disk4203	87039MB
A.Q7.28	/dev/disk/disk4184	87039MB
A.Q7.28	/dev/disk/disk4191	87039MB
A.Q8.29	/dev/disk/disk5044	87039MB
A.Q8.29	/dev/disk/disk5062	87039MB
A.Q8.30	/dev/disk/disk5702	87039MB
A.Q8.30	/dev/disk/disk5708	87039MB
A.Q8.31	/dev/disk/disk4339	87039MB
A.Q8.31	/dev/disk/disk4993	87039MB
A.Q8.32	/dev/disk/disk5664	87039MB
A.Q8.32	/dev/disk/disk5666	87039MB
A.Q9.33	/dev/disk/disk5366	87039MB
A.Q9.33	/dev/disk/disk5744	87039MB
A.Q9.34	/dev/disk/disk6023	87039MB
A.Q9.34	/dev/disk/disk6024	87039MB
A.Q9.35	/dev/disk/disk6070	87039MB
A.Q9.35	/dev/disk/disk6082	87039MB
A.Q9.36	/dev/disk/disk5656	87039MB

A.O9.36	/dev/disk/disk5706	87039MB
B.O1.1	/dev/disk/disk4408	87039MB
B.Q1.1	/dev/disk/disk4412	87039MB
B.O1.2	/dev/disk/disk5532	87039MB
B.O1.2	/dev/disk/disk5560	87039MB
B.Q1.3	/dev/disk/disk5258	87039MB
B.O1.3	/dev/disk/disk5275	87039MB
B.Q1.4	/dev/disk/disk4911	87039MB
B.O1.4	/dev/disk/disk4918	87039MB
B.O10.37	/dev/disk/disk4377	87039MB
B.Q10.37	/dev/disk/disk4825	87039MB
B.O10.38	/dev/disk/disk5171	87039MB
B.O10.38	/dev/disk/disk5183	87039MB
B.O10.39	/dev/disk/disk4864	87039MB
B.O10.39	/dev/disk/disk5989	87039MB
B.Q10.40	/dev/disk/disk4856	87039MB
B.O10.40	/dev/disk/disk5298	87039MB
B.Q12.45	/dev/disk/disk5546	87039MB
B.Q12.45	/dev/disk/disk5564	87039MB
B.Q12.46	/dev/disk/disk6044	87039MB
B.Q12.46	/dev/disk/disk6045	87039MB
B.Q12.47	/dev/disk/disk4319	87039MB
B.Q12.47	/dev/disk/disk4379	87039MB
B.Q12.48	/dev/disk/disk5003	87039MB
B.Q12.48	/dev/disk/disk5006	87039MB
B.Q13.49	/dev/disk/disk4281	87039MB
B.Q13.49	/dev/disk/disk5835	87039MB
B.O13.50	/dev/disk/disk4582	87039MB
B.O13.50	/dev/disk/disk4707	87039MB
B.Q13.51	/dev/disk/disk4243	87039MB
B.O13.51	/dev/disk/disk4245	87039MB
B.Q13.52	/dev/disk/disk6900	87039MB
B.Q13.52	/dev/disk/disk6901	87039MB
B.O2.5	/dev/disk/disk5949	87039MB
B.Q2.5	/dev/disk/disk5955	87039MB
B.O2.6	/dev/disk/disk4206	87039MB
B.O2.6	/dev/disk/disk4211	87039MB
B.O2.7	/dev/disk/disk5880	87039MB
B.O2.7	/dev/disk/disk5885	87039MB
B.Q2.8	/dev/disk/disk5176	87039MB
B.O2.8	/dev/disk/disk5200	87039MB
B.Q3.10	/dev/disk/disk4840	87039MB
B.Q3.10	/dev/disk/disk4979	87039MB
B.Q3.11	/dev/disk/disk4616	87039MB
B.Q3.11	/dev/disk/disk4737	87039MB
B.Q3.12	/dev/disk/disk4788	87039MB
B.O3.12	/dev/disk/disk4790	87039MB
B.Q3.9	/dev/disk/disk5918	87039MB
B.O3.9	/dev/disk/disk5921	87039MB
B.O4.13	/dev/disk/disk4893	87039MB
B.Q4.13	/dev/disk/disk4909	87039MB
B.O4.14	/dev/disk/disk5425	87039MB

B.Q4.14	/dev/disk/disk5464	87039MB
B.Q4.15	/dev/disk/disk5036	87039MB
B.Q4.15	/dev/disk/disk6176	87039MB
B.Q4.16	/dev/disk/disk4532	87039MB
B.Q4.16	/dev/disk/disk4542	87039MB
B.Q5.17	/dev/disk/disk4435	87039MB
B.Q5.17	/dev/disk/disk4444	87039MB
B.Q5.18	/dev/disk/disk6109	87039MB
B.Q5.18	/dev/disk/disk6111	87039MB
B.Q5.19	/dev/disk/disk4385	87039MB
B.Q5.19	/dev/disk/disk5146	87039MB
B.Q5.20	/dev/disk/disk4401	87039MB
B.Q5.20	/dev/disk/disk4418	87039MB
B.Q6.21	/dev/disk/disk3826	87039MB
B.Q6.21	/dev/disk/disk3833	87039MB
B.Q6.22	/dev/disk/disk3750	87039MB
B.Q6.22	/dev/disk/disk3751	87039MB
B.Q6.23	/dev/disk/disk3978	87039MB
B.Q6.23	/dev/disk/disk3981	87039MB
B.Q6.24	/dev/disk/disk4299	87039MB
B.Q6.24	/dev/disk/disk4300	87039MB
B.Q7.25	/dev/disk/disk5758	87039MB
B.Q7.25	/dev/disk/disk5760	87039MB
B.Q7.26	/dev/disk/disk5492	87039MB
B.Q7.26	/dev/disk/disk5510	87039MB
B.Q7.27	/dev/disk/disk5784	87039MB
B.Q7.27	/dev/disk/disk5785	87039MB
B.Q7.28	/dev/disk/disk5796	87039MB
B.Q7.28	/dev/disk/disk5806	87039MB
B.Q8.29	/dev/disk/disk3937	87039MB
B.Q8.29	/dev/disk/disk3938	87039MB
B.Q8.30	/dev/disk/disk3706	87039MB
B.Q8.30	/dev/disk/disk3709	87039MB
B.Q8.31	/dev/disk/disk5819	87039MB
B.Q8.31	/dev/disk/disk5824	87039MB
B.Q8.32	/dev/disk/disk4164	87039MB
B.Q8.32	/dev/disk/disk4165	87039MB
B.Q9.33	/dev/disk/disk4517	87039MB
B.Q9.33	/dev/disk/disk4518	87039MB
B.Q9.34	/dev/disk/disk4349	87039MB
B.Q9.34	/dev/disk/disk5209	87039MB
C.Q1.1	/dev/disk/disk6622	87039MB
C.Q1.1	/dev/disk/disk6626	87039MB
C.Q1.2	/dev/disk/disk6367	87039MB
C.Q1.2	/dev/disk/disk6378	87039MB
C.Q1.3	/dev/disk/disk6867	87039MB
C.Q1.3	/dev/disk/disk6868	87039MB
C.Q1.4	/dev/disk/disk6413	87039MB
C.Q1.4	/dev/disk/disk6431	87039MB
C.Q2.5	/dev/disk/disk6354	87039MB
C.Q2.5	/dev/disk/disk6652	87039MB
C.Q2.6	/dev/disk/disk6602	87039MB

C.Q2.6	/dev/disk/disk6617	87039MB
C.Q2.7	/dev/disk/disk6746	87039MB
C.Q2.7	/dev/disk/disk6772	87039MB
C.Q2.8	/dev/disk/disk6340	87039MB
C.Q2.8	/dev/disk/disk6416	87039MB
C.Q3.10	/dev/disk/disk6949	87039MB
C.Q3.10	/dev/disk/disk6950	87039MB
C.Q3.11	/dev/disk/disk6283	87039MB
C.Q3.11	/dev/disk/disk6287	87039MB
C.Q3.12	/dev/disk/disk6286	87039MB
C.Q3.12	/dev/disk/disk6290	87039MB
C.Q3.9	/dev/disk/disk6306	87039MB
C.Q3.9	/dev/disk/disk6310	87039MB
C.Q4.13	/dev/disk/disk6718	87039MB
C.Q4.13	/dev/disk/disk6743	87039MB
C.Q4.14	/dev/disk/disk6740	87039MB
C.Q4.14	/dev/disk/disk6744	87039MB
C.Q4.15	/dev/disk/disk6719	87039MB
C.Q4.15	/dev/disk/disk6810	87039MB
C.Q4.16	/dev/disk/disk6665	87039MB
C.Q4.16	/dev/disk/disk6843	87039MB
C.Q5.17	/dev/disk/disk6598	87039MB
C.Q5.17	/dev/disk/disk6782	87039MB
C.Q5.18	/dev/disk/disk6595	87039MB
C.Q5.18	/dev/disk/disk6802	87039MB
C.Q5.19	/dev/disk/disk6391	87039MB
C.Q5.19	/dev/disk/disk6650	87039MB
C.Q5.20	/dev/disk/disk6742	87039MB
C.Q5.20	/dev/disk/disk6751	87039MB
C.Q6.21	/dev/disk/disk6343	87039MB
C.Q6.21	/dev/disk/disk6358	87039MB
C.Q6.22	/dev/disk/disk6334	87039MB
C.Q6.22	/dev/disk/disk6338	87039MB
C.Q6.23	/dev/disk/disk6536	87039MB
C.Q6.23	/dev/disk/disk6550	87039MB
C.Q6.24	/dev/disk/disk6345	87039MB
C.Q6.24	/dev/disk/disk6371	87039MB

4) List of all Oracle datafiles and the corresponding device: (sorted by name)

Each oracle data file is either assigned to a LUN on of the 118 arrays, or to an LV in one of the two VGs on those arrays.

Oracle file	FILE ID	SIZE (MB)	TABLE-SPACE	Disk Array	Special device file	Device Capacity
control_001	N/A	N/A	N/A	All	/dev/vgdata1/rcontrol_001	1888
control_002	N/A	N/A	N/A	All	/dev/vgdata1/rcontrol_002	1888
cust_0_0	5	16100	CUST_0	B.Q5.17	/dev/rdisk/disk4479	16386

cust 0 1	92	16100	CUST 0	B.Q3.11	/dev/rdisk/disk4626	16386
cust 0 10	59	16100	CUST 0	C.Q2.8	/dev/rdisk/disk6555	16386
cust_0_100	114	16100	CUST_0	C.Q4.13	/dev/rdisk/disk6750	16386
cust 0 101	65	16100	CUST 0	A.Q9.36	/dev/rdisk/disk5360	16386
cust 0 102	161	16100	CUST 0	B.Q6.22	/dev/rdisk/disk3763	16386
cust_0_103	72	16100	CUST_0	A.Q6.23	/dev/rdisk/disk5433	16386
cust 0 104	242	16100	CUST 0	B.Q4.16	/dev/rdisk/disk4717	16386
cust_0_105	148	16100	CUST_0	A.Q7.28	/dev/rdisk/disk4188	16386
cust_0_106	81	16100	CUST_0	A.Q6.22	/dev/rdisk/disk6086	16386
cust 0 107	120	16100	CUST 0	A.Q8.32	/dev/rdisk/disk4311	16386
cust_0_108	20	16100	CUST_0	B.Q8.29	/dev/rdisk/disk3898	16386
cust 0 109	57	16100	CUST 0	C.Q2.6	/dev/rdisk/disk6621	16386
cust 0 11	27	16100	CUST 0	B.Q10.37	/dev/rdisk/disk5237	16386
cust 0 110	157	16100	CUST 0	A.Q7.27	/dev/rdisk/disk4197	16386
cust 0 111	256	16100	CUST 0	C.Q1.2	/dev/rdisk/disk6381	16386
cust_0_112	250	16100	CUST_0	C.Q5.19	/dev/rdisk/disk6730	16386
cust 0 113	159	16100	CUST 0	B.Q7.26	/dev/rdisk/disk5480	16386
cust 0 114	130	16100	CUST 0	C.Q5.20	/dev/rdisk/disk6720	16386
cust_0_115	174	16100	CUST_0	B.Q8.30	/dev/rdisk/disk3721	16386
cust 0 116	140	16100	CUST 0	B.Q7.25	/dev/rdisk/disk5717	16386
cust_0_117	87	16100	CUST_0	B.Q12.45	/dev/rdisk/disk5559	16386
cust 0 118	234	16100	CUST 0	B.Q5.17	/dev/rdisk/disk4434	16386
cust 0 119	22	16100	CUST 0	B.Q3.11	/dev/rdisk/disk4673	16386
cust_0_12	255	16100	CUST_0	A.Q13.52	/dev/rdisk/disk5736	16386
cust 0 120	26	16100	CUST 0	A.Q9.35	/dev/rdisk/disk6148	16386
cust 0 121	257	16100	CUST 0	C.Q4.14	/dev/rdisk/disk6662	16386
cust_0_122	259	16100	CUST_0	A.Q5.17	/dev/rdisk/disk4960	16386
cust 0 123	249	16100	CUST 0	A.Q4.15	/dev/rdisk/disk3946	16386
cust 0 124	136	16100	CUST 0	A.Q4.13	/dev/rdisk/disk4082	16386
cust_0_125	46	16100	CUST_0	A.Q1.1	/dev/rdisk/disk5243	16386
cust 0 126	101	16100	CUST 0	A.Q8.29	/dev/rdisk/disk5086	16386
cust_0_127	121	16100	CUST_0	C.Q5.18	/dev/rdisk/disk6814	16386
cust_0_128	171	16100	CUST_0	C.Q2.8	/dev/rdisk/disk6552	16386
cust 0 129	119	16100	CUST 0	B.Q10.37	/dev/rdisk/disk4374	16386
cust_0_13	195	16100	CUST_0	A.Q13.50	/dev/rdisk/disk5665	16386
cust 0 130	216	16100	CUST 0	A.Q13.52	/dev/rdisk/disk5690	16386
cust 0 131	17	16100	CUST 0	A.Q13.50	/dev/rdisk/disk5667	16386
cust 0 132	156	16100	CUST 0	B.Q5.19	/dev/rdisk/disk4804	16386
cust 0 133	8	16100	CUST 0	C.Q6.22	/dev/rdisk/disk6445	16386
cust_0_134	107	16100	CUST_0	B.Q2.8	/dev/rdisk/disk4836	16386
cust 0 135	91	16100	CUST 0	B.Q3.10	/dev/rdisk/disk4949	16386
cust 0 136	211	16100	CUST 0	B.Q9.33	/dev/rdisk/disk4739	16386
cust_0_137	238	16100	CUST_0	A.Q6.24	/dev/rdisk/disk3773	16386
cust 0 138	133	16100	CUST 0	A.Q12.48	/dev/rdisk/disk4294	16386
cust_0_139	112	16100	CUST_0	B.Q4.13	/dev/rdisk/disk4579	16386
cust 0 14	258	16100	CUST 0	B.Q5.19	/dev/rdisk/disk4805	16386
cust 0 140	173	16100	CUST 0	B.Q12.46	/dev/rdisk/disk6112	16386
cust_0_141	169	16100	CUST_0	A.Q9.34	/dev/rdisk/disk6080	16386
cust 0 142	225	16100	CUST 0	A.Q10.38	/dev/rdisk/disk5067	16386
cust 0 143	69	16100	CUST 0	C.Q3.9	/dev/rdisk/disk6374	16386
cust_0_144	124	16100	CUST_0	A.Q11.43	/dev/rdisk/disk6243	16386
cust 0 145	198	16100	CUST 0	C.Q3.10	/dev/rdisk/disk6960	16386

cust 0 146	108	16100	CUST 0	C.Q1.1	/dev/rdisk/disk6583	16386
cust 0 147	9	16100	CUST 0	A.Q12.47	/dev/rdisk/disk5563	16386
cust_0_148	248	16100	CUST_0	C.Q4.16	/dev/rdisk/disk6688	16386
cust 0 149	150	16100	CUST 0	B.Q4.15	/dev/rdisk/disk6179	16386
cust 0 15	74	16100	CUST 0	C.Q6.22	/dev/rdisk/disk6454	16386
cust_0_150	167	16100	CUST_0	B.Q10.39	/dev/rdisk/disk4867	16386
cust 0 151	129	16100	CUST 0	B.Q12.47	/dev/rdisk/disk5621	16386
cust_0_152	233	16100	CUST_0	A.Q11.44	/dev/rdisk/disk6279	16386
cust_0_153	168	16100	CUST_0	A.Q2.5	/dev/rdisk/disk3868	16386
cust 0 154	243	16100	CUST 0	B.Q8.32	/dev/rdisk/disk4136	16386
cust_0_155	201	16100	CUST_0	C.Q3.12	/dev/rdisk/disk6342	16386
cust 0 156	137	16100	CUST 0	B.Q12.48	/dev/rdisk/disk5057	16386
cust 0 157	196	16100	CUST 0	B.Q7.27	/dev/rdisk/disk5263	16386
cust 0 158	221	16100	CUST 0	B.Q1.4	/dev/rdisk/disk4927	16386
cust 0 159	222	16100	CUST 0	B.Q13.51	/dev/rdisk/disk4241	16386
cust_0_16	210	16100	CUST_0	B.Q2.8	/dev/rdisk/disk4833	16386
cust 0 160	254	16100	CUST 0	B.Q13.49	/dev/rdisk/disk5052	16386
cust 0 161	163	16100	CUST 0	C.Q6.24	/dev/rdisk/disk6421	16386
cust_0_162	213	16100	CUST_0	A.Q8.31	/dev/rdisk/disk5317	16386
cust 0 163	141	16100	CUST 0	A.Q1.2	/dev/rdisk/disk4537	16386
cust_0_164	60	16100	CUST_0	B.Q6.23	/dev/rdisk/disk3985	16386
cust 0 165	58	16100	CUST 0	B.Q7.28	/dev/rdisk/disk4270	16386
cust 0 166	37	16100	CUST 0	A.Q5.19	/dev/rdisk/disk4643	16386
cust_0_167	189	16100	CUST_0	B.Q3.9	/dev/rdisk/disk5931	16386
cust 0 168	138	16100	CUST 0	B.Q13.52	/dev/rdisk/disk6898	16386
cust 0 169	98	16100	CUST 0	A.Q12.46	/dev/rdisk/disk5336	16386
cust_0_17	184	16100	CUST_0	B.Q3.10	/dev/rdisk/disk4818	16386
cust 0 170	220	16100	CUST 0	A.Q2.6	/dev/rdisk/disk3638	16386
cust 0 171	115	16100	CUST 0	A.Q9.33	/dev/rdisk/disk5649	16386
cust_0_172	125	16100	CUST_0	B.Q10.40	/dev/rdisk/disk5333	16386
cust 0 173	181	16100	CUST 0	B.Q13.50	/dev/rdisk/disk4600	16386
cust_0_174	230	16100	CUST_0	A.Q13.51	/dev/rdisk/disk4320	16386
cust_0_175	180	16100	CUST_0	B.Q5.20	/dev/rdisk/disk5150	16386
cust 0 176	178	16100	CUST 0	B.Q1.1	/dev/rdisk/disk5522	16386
cust_0_177	103	16100	CUST_0	C.Q2.7	/dev/rdisk/disk6756	16386
cust 0 178	84	16100	CUST 0	C.Q1.3	/dev/rdisk/disk6878	16386
cust 0 179	247	16100	CUST 0	C.Q1.4	/dev/rdisk/disk6524	16386
cust 0 18	191	16100	CUST 0	B.Q9.33	/dev/rdisk/disk4748	16386
cust 0 180	53	16100	CUST 0	C.Q6.21	/dev/rdisk/disk6477	16386
cust_0_181	106	16100	CUST_0	C.Q5.17	/dev/rdisk/disk6584	16386
cust 0 182	139	16100	CUST 0	B.Q4.14	/dev/rdisk/disk5889	16386
cust 0 183	187	16100	CUST 0	A.Q1.4	/dev/rdisk/disk6925	16386
cust_0_184	218	16100	CUST_0	A.Q3.9	/dev/rdisk/disk3894	16386
cust 0 185	205	16100	CUST 0	A.Q2.8	/dev/rdisk/disk3675	16386
cust_0_186	206	16100	CUST_0	A.Q2.7	/dev/rdisk/disk3645	16386
cust 0 187	149	16100	CUST 0	B.Q1.3	/dev/rdisk/disk5611	16386
cust 0 188	227	16100	CUST 0	B.Q5.18	/dev/rdisk/disk6156	16386
cust_0_189	145	16100	CUST_0	B.Q9.34	/dev/rdisk/disk5204	16386
cust 0 19	70	16100	CUST 0	A.Q6.24	/dev/rdisk/disk3778	16386
cust 0 190	200	16100	CUST 0	C.Q4.15	/dev/rdisk/disk6671	16386
cust_0_191	24	16100	CUST_0	B.Q3.12	/dev/rdisk/disk4743	16386
cust 0 192	54	16100	CUST 0	A.Q7.26	/dev/rdisk/disk4096	16386

cust 0 193	109	16100	CUST 0	A.Q3.10	/dev/rdisk/disk3850	16386
cust 0 194	261	16100	CUST 0	A.Q6.21	/dev/rdisk/disk4463	16386
cust_0_195	94	16100	CUST_0	C.Q3.11	/dev/rdisk/disk6322	16386
cust 0 196	134	16100	CUST 0	A.Q7.25	/dev/rdisk/disk6087	16386
cust 0 197	190	16100	CUST 0	A.Q4.16	/dev/rdisk/disk4051	16386
cust_0_198	245	16100	CUST_0	B.Q2.6	/dev/rdisk/disk5873	16386
cust 0 199	186	16100	CUST 0	A.Q3.11	/dev/rdisk/disk3786	16386
cust_0_2	126	16100	CUST_0	A.Q9.35	/dev/rdisk/disk6150	16386
cust_0_20	239	16100	CUST_0	A.Q12.48	/dev/rdisk/disk5705	16386
cust 0 200	164	16100	CUST 0	A.Q5.18	/dev/rdisk/disk4904	16386
cust_0_201	236	16100	CUST_0	A.Q5.20	/dev/rdisk/disk4631	16386
cust 0 202	244	16100	CUST 0	A.Q4.14	/dev/rdisk/disk4029	16386
cust 0 203	23	16100	CUST 0	A.Q1.3	/dev/rdisk/disk5070	16386
cust 0 204	135	16100	CUST 0	A.Q10.39	/dev/rdisk/disk4794	16386
cust 0 205	197	16100	CUST 0	B.Q8.31	/dev/rdisk/disk5429	16386
cust_0_206	21	16100	CUST_0	B.Q10.38	/dev/rdisk/disk5173	16386
cust 0 207	102	16100	CUST 0	B.Q2.5	/dev/rdisk/disk5964	16386
cust 0 208	152	16100	CUST 0	B.Q6.21	/dev/rdisk/disk3851	16386
cust_0_209	144	16100	CUST_0	A.Q11.42	/dev/rdisk/disk6218	16386
cust 0 21	182	16100	CUST 0	B.Q4.13	/dev/rdisk/disk4564	16386
cust_0_210	228	16100	CUST_0	A.Q10.40	/dev/rdisk/disk5553	16386
cust 0 211	56	16100	CUST 0	B.Q1.2	/dev/rdisk/disk5558	16386
cust 0 212	166	16100	CUST 0	C.Q2.5	/dev/rdisk/disk6609	16386
cust_0_213	185	16100	CUST_0	A.Q8.30	/dev/rdisk/disk4267	16386
cust 0 214	68	16100	CUST 0	A.Q3.12	/dev/rdisk/disk4002	16386
cust 0 215	67	16100	CUST 0	B.Q6.24	/dev/rdisk/disk5804	16386
cust_0_216	13	16100	CUST_0	B.Q2.7	/dev/rdisk/disk4119	16386
cust 0 217	30	16100	CUST 0	C.Q6.23	/dev/rdisk/disk6560	16386
cust 0 218	111	16100	CUST 0	C.Q4.13	/dev/rdisk/disk6755	16386
cust_0_219	117	16100	CUST_0	A.Q9.36	/dev/rdisk/disk5362	16386
cust 0 22	179	16100	CUST 0	B.Q12.46	/dev/rdisk/disk6114	16386
cust_0_220	132	16100	CUST_0	B.Q6.22	/dev/rdisk/disk3757	16386
cust_0_221	175	16100	CUST_0	A.Q6.23	/dev/rdisk/disk5430	16386
cust 0 222	235	16100	CUST 0	B.Q4.16	/dev/rdisk/disk4720	16386
cust_0_223	176	16100	CUST_0	A.Q7.28	/dev/rdisk/disk5986	16386
cust 0 224	78	16100	CUST 0	A.Q6.22	/dev/rdisk/disk6083	16386
cust 0 225	232	16100	CUST 0	A.Q8.32	/dev/rdisk/disk4309	16386
cust 0 226	79	16100	CUST 0	B.Q8.29	/dev/rdisk/disk3893	16386
cust 0 227	64	16100	CUST 0	C.Q2.6	/dev/rdisk/disk6633	16386
cust_0_228	224	16100	CUST_0	A.Q7.27	/dev/rdisk/disk4195	16386
cust 0 229	55	16100	CUST 0	C.Q5.19	/dev/rdisk/disk6724	16386
cust 0 23	199	16100	CUST 0	A.Q9.34	/dev/rdisk/disk6081	16386
cust_0_230	158	16100	CUST_0	C.Q1.2	/dev/rdisk/disk6604	16386
cust 0 231	118	16100	CUST 0	B.Q7.26	/dev/rdisk/disk4234	16386
cust_0_232	31	16100	CUST_0	C.Q5.20	/dev/rdisk/disk6721	16386
cust 0 233	193	16100	CUST 0	B.Q8.30	/dev/rdisk/disk3722	16386
cust 0 234	229	16100	CUST 0	B.Q7.25	/dev/rdisk/disk5724	16386
cust_0_235	155	16100	CUST_0	B.Q12.45	/dev/rdisk/disk5552	16386
cust 0 236	212	16100	CUST 0	B.Q5.17	/dev/rdisk/disk4477	16386
cust 0 237	48	16100	CUST 0	B.Q3.11	/dev/rdisk/disk4709	16386
cust_0_238	146	16100	CUST_0	A.Q9.35	/dev/rdisk/disk6142	16386
cust 0 239	19	16100	CUST 0	C.Q4.14	/dev/rdisk/disk6660	16386

cust 0 24	116	16100	CUST 0	A.Q10.38	/dev/rdisk/disk4406	16386
cust 0 240	223	16100	CUST 0	A.Q5.17	/dev/rdisk/disk4961	16386
cust_0_241	29	16100	CUST_0	A.Q4.15	/dev/rdisk/disk3918	16386
cust 0 242	260	16100	CUST 0	A.Q4.13	/dev/rdisk/disk5945	16386
cust 0 243	86	16100	CUST 0	A.Q1.1	/dev/rdisk/disk5321	16386
cust_0_244	217	16100	CUST_0	A.Q8.29	/dev/rdisk/disk5120	16386
cust 0 245	127	16100	CUST 0	C.Q5.18	/dev/rdisk/disk6816	16386
cust_0_246	82	16100	CUST_0	C.Q2.8	/dev/rdisk/disk6548	16386
cust_0_247	49	16100	CUST_0	B.Q10.37	/dev/rdisk/disk5306	16386
cust 0 248	241	16100	CUST 0	A.Q13.52	/dev/rdisk/disk5672	16386
cust_0_249	183	16100	CUST_0	A.Q13.50	/dev/rdisk/disk5675	16386
cust 0 25	11	16100	CUST 0	C.Q3.9	/dev/rdisk/disk6386	16386
cust 0 250	83	16100	CUST 0	B.Q5.19	/dev/rdisk/disk4802	16386
cust 0 251	204	16100	CUST 0	C.Q6.22	/dev/rdisk/disk6428	16386
cust 0 252	38	16100	CUST 0	B.Q2.8	/dev/rdisk/disk4824	16386
cust_0_253	12	16100	CUST_0	B.Q3.10	/dev/rdisk/disk4951	16386
cust 0 254	32	16100	CUST 0	B.Q9.33	/dev/rdisk/disk5142	16386
cust 0 255	251	16100	CUST 0	A.Q6.24	/dev/rdisk/disk3774	16386
cust_0_256	51	16100	CUST_0	A.Q12.48	/dev/rdisk/disk5357	16386
cust 0 257	344	16100	CUST 0	B.Q4.13	/dev/rdisk/disk4577	16386
cust_0_258	419	16100	CUST_0	B.Q12.46	/dev/rdisk/disk6097	16386
cust 0 259	449	16100	CUST 0	A.Q9.34	/dev/rdisk/disk6066	16386
cust 0 26	47	16100	CUST 0	A.Q11.43	/dev/rdisk/disk6254	16386
cust_0_260	347	16100	CUST_0	A.Q10.38	/dev/rdisk/disk5029	16386
cust 0 261	276	16100	CUST 0	C.Q3.9	/dev/rdisk/disk6368	16386
cust 0 262	275	16100	CUST 0	A.Q11.43	/dev/rdisk/disk6244	16386
cust_0_263	270	16100	CUST_0	C.Q3.10	/dev/rdisk/disk6959	16386
cust 0 264	385	16100	CUST 0	C.Q1.1	/dev/rdisk/disk6576	16386
cust 0 265	484	16100	CUST 0	A.Q12.47	/dev/rdisk/disk5557	16386
cust_0_266	266	16100	CUST_0	C.Q4.16	/dev/rdisk/disk6860	16386
cust 0 267	265	16100	CUST 0	B.Q4.15	/dev/rdisk/disk6183	16386
cust_0_268	436	16100	CUST_0	B.Q10.39	/dev/rdisk/disk4360	16386
cust_0_269	488	16100	CUST_0	B.Q12.47	/dev/rdisk/disk5624	16386
cust 0 27	252	16100	CUST 0	C.Q3.10	/dev/rdisk/disk6961	16386
cust_0_270	402	16100	CUST_0	A.Q11.44	/dev/rdisk/disk6250	16386
cust 0 271	356	16100	CUST 0	A.Q2.5	/dev/rdisk/disk3865	16386
cust 0 272	350	16100	CUST 0	B.Q8.32	/dev/rdisk/disk4144	16386
cust 0 273	314	16100	CUST 0	C.Q3.12	/dev/rdisk/disk6321	16386
cust 0 274	407	16100	CUST 0	B.Q12.48	/dev/rdisk/disk5064	16386
cust_0_275	397	16100	CUST_0	B.Q7.27	/dev/rdisk/disk5269	16386
cust 0 276	369	16100	CUST 0	B.Q1.4	/dev/rdisk/disk4924	16386
cust 0 277	399	16100	CUST 0	B.Q13.51	/dev/rdisk/disk5998	16386
cust_0_278	510	16100	CUST_0	B.Q13.49	/dev/rdisk/disk4317	16386
cust 0 279	323	16100	CUST 0	C.Q6.24	/dev/rdisk/disk6355	16386
cust_0_28	123	16100	CUST_0	C.Q1.1	/dev/rdisk/disk6585	16386
cust 0 280	444	16100	CUST 0	A.Q8.31	/dev/rdisk/disk4334	16386
cust 0 281	313	16100	CUST 0	A.Q1.2	/dev/rdisk/disk4540	16386
cust_0_282	503	16100	CUST_0	B.Q6.23	/dev/rdisk/disk3984	16386
cust 0 283	321	16100	CUST 0	B.Q7.28	/dev/rdisk/disk5769	16386
cust 0 284	277	16100	CUST 0	A.Q5.19	/dev/rdisk/disk4700	16386
cust_0_285	395	16100	CUST_0	B.Q3.9	/dev/rdisk/disk5932	16386
cust 0 286	494	16100	CUST 0	B.Q13.52	/dev/rdisk/disk6897	16386

cust 0 287	396	16100	CUST 0	A.Q12.46	/dev/rdisk/disk5342	16386
cust 0 288	306	16100	CUST 0	A.Q2.6	/dev/rdisk/disk3640	16386
cust_0_289	262	16100	CUST_0	A.Q9.33	/dev/rdisk/disk4362	16386
cust 0 29	63	16100	CUST 0	A.Q12.47	/dev/rdisk/disk5561	16386
cust 0 290	335	16100	CUST 0	B.Q10.40	/dev/rdisk/disk5331	16386
cust_0_291	269	16100	CUST_0	B.Q13.50	/dev/rdisk/disk4576	16386
cust 0 292	441	16100	CUST 0	A.Q13.51	/dev/rdisk/disk5051	16386
cust_0_293	354	16100	CUST_0	B.Q5.20	/dev/rdisk/disk4755	16386
cust_0_294	352	16100	CUST_0	B.Q1.1	/dev/rdisk/disk5513	16386
cust 0 295	333	16100	CUST 0	C.Q2.7	/dev/rdisk/disk6758	16386
cust_0_296	389	16100	CUST_0	C.Q1.3	/dev/rdisk/disk6865	16386
cust 0 297	278	16100	CUST 0	C.Q1.4	/dev/rdisk/disk6525	16386
cust 0 298	421	16100	CUST 0	C.Q6.21	/dev/rdisk/disk6385	16386
cust 0 299	287	16100	CUST 0	C.Q5.17	/dev/rdisk/disk6822	16386
cust 0 3	110	16100	CUST 0	C.Q4.14	/dev/rdisk/disk6690	16386
cust_0_30	207	16100	CUST_0	C.Q4.16	/dev/rdisk/disk6851	16386
cust 0 300	376	16100	CUST 0	B.Q4.14	/dev/rdisk/disk5900	16386
cust 0 301	288	16100	CUST 0	A.Q1.4	/dev/rdisk/disk6926	16386
cust_0_302	357	16100	CUST_0	A.Q3.9	/dev/rdisk/disk3892	16386
cust 0 303	322	16100	CUST 0	A.Q2.8	/dev/rdisk/disk3676	16386
cust_0_304	443	16100	CUST_0	A.Q2.7	/dev/rdisk/disk3661	16386
cust 0 305	319	16100	CUST 0	B.Q1.3	/dev/rdisk/disk5273	16386
cust 0 306	432	16100	CUST 0	B.Q5.18	/dev/rdisk/disk6155	16386
cust_0_307	464	16100	CUST_0	B.Q9.34	/dev/rdisk/disk5203	16386
cust 0 308	489	16100	CUST 0	C.Q4.15	/dev/rdisk/disk6672	16386
cust 0 309	418	16100	CUST 0	B.Q3.12	/dev/rdisk/disk4760	16386
cust_0_31	50	16100	CUST_0	B.Q4.15	/dev/rdisk/disk4731	16386
cust 0 310	433	16100	CUST 0	A.Q7.26	/dev/rdisk/disk5639	16386
cust 0 311	316	16100	CUST 0	A.Q3.10	/dev/rdisk/disk3847	16386
cust_0_312	263	16100	CUST_0	A.Q6.21	/dev/rdisk/disk4494	16386
cust 0 313	339	16100	CUST 0	C.Q3.11	/dev/rdisk/disk6317	16386
cust_0_314	437	16100	CUST_0	A.Q7.25	/dev/rdisk/disk6074	16386
cust_0_315	504	16100	CUST_0	A.Q4.16	/dev/rdisk/disk4037	16386
cust 0 316	442	16100	CUST 0	B.Q2.6	/dev/rdisk/disk5868	16386
cust_0_317	485	16100	CUST_0	A.Q3.11	/dev/rdisk/disk3791	16386
cust 0 318	318	16100	CUST 0	A.Q5.18	/dev/rdisk/disk4493	16386
cust 0 319	440	16100	CUST 0	A.Q5.20	/dev/rdisk/disk4710	16386
cust 0 32	240	16100	CUST 0	B.Q10.39	/dev/rdisk/disk4364	16386
cust 0 320	390	16100	CUST 0	A.Q4.14	/dev/rdisk/disk4038	16386
cust_0_321	472	16100	CUST_0	A.Q1.3	/dev/rdisk/disk5069	16386
cust 0 322	337	16100	CUST 0	A.Q10.39	/dev/rdisk/disk5047	16386
cust 0 323	340	16100	CUST 0	B.Q8.31	/dev/rdisk/disk5428	16386
cust_0_324	509	16100	CUST_0	B.Q10.38	/dev/rdisk/disk4380	16386
cust 0 325	303	16100	CUST 0	B.Q2.5	/dev/rdisk/disk5963	16386
cust_0_326	512	16100	CUST_0	B.Q6.21	/dev/rdisk/disk3854	16386
cust 0 327	423	16100	CUST 0	A.Q11.42	/dev/rdisk/disk6217	16386
cust 0 328	304	16100	CUST 0	A.Q10.40	/dev/rdisk/disk6010	16386
cust_0_329	349	16100	CUST_0	B.Q1.2	/dev/rdisk/disk5544	16386
cust 0 33	162	16100	CUST 0	B.Q12.47	/dev/rdisk/disk5622	16386
cust 0 330	404	16100	CUST 0	C.Q2.5	/dev/rdisk/disk6366	16386
cust_0_331	420	16100	CUST_0	A.Q8.30	/dev/rdisk/disk5676	16386
cust 0 332	471	16100	CUST 0	A.Q3.12	/dev/rdisk/disk4000	16386

cust 0 333	470	16100	CUST 0	B.Q6.24	/dev/rdisk/disk4301	16386
cust 0 334	487	16100	CUST 0	B.Q2.7	/dev/rdisk/disk4108	16386
cust_0_335	290	16100	CUST_0	C.Q6.23	/dev/rdisk/disk6580	16386
cust 0 336	295	16100	CUST 0	C.Q4.13	/dev/rdisk/disk6775	16386
cust 0 337	377	16100	CUST 0	A.Q9.36	/dev/rdisk/disk5419	16386
cust_0_338	511	16100	CUST_0	B.Q6.22	/dev/rdisk/disk3755	16386
cust 0 339	365	16100	CUST 0	A.Q6.23	/dev/rdisk/disk5446	16386
cust_0_34	77	16100	CUST_0	A.Q11.44	/dev/rdisk/disk6278	16386
cust_0_340	300	16100	CUST_0	B.Q4.16	/dev/rdisk/disk4565	16386
cust 0 341	448	16100	CUST 0	A.Q7.28	/dev/rdisk/disk5987	16386
cust_0_342	315	16100	CUST_0	A.Q6.22	/dev/rdisk/disk6071	16386
cust 0 343	435	16100	CUST 0	A.Q8.32	/dev/rdisk/disk4308	16386
cust 0 344	296	16100	CUST 0	B.Q8.29	/dev/rdisk/disk3958	16386
cust 0 345	403	16100	CUST 0	C.Q2.6	/dev/rdisk/disk6596	16386
cust 0 346	514	16100	CUST 0	A.Q7.27	/dev/rdisk/disk5856	16386
cust_0_347	305	16100	CUST_0	C.Q1.2	/dev/rdisk/disk6447	16386
cust 0 348	413	16100	CUST 0	C.Q5.19	/dev/rdisk/disk6748	16386
cust 0 349	378	16100	CUST 0	B.Q7.26	/dev/rdisk/disk5478	16386
cust_0_35	18	16100	CUST_0	A.Q2.5	/dev/rdisk/disk3869	16386
cust 0 350	450	16100	CUST 0	C.Q5.20	/dev/rdisk/disk6765	16386
cust_0_351	283	16100	CUST_0	B.Q8.30	/dev/rdisk/disk3716	16386
cust 0 352	267	16100	CUST 0	B.Q7.25	/dev/rdisk/disk5805	16386
cust 0 353	348	16100	CUST 0	B.Q12.45	/dev/rdisk/disk5234	16386
cust_0_354	483	16100	CUST_0	B.Q5.17	/dev/rdisk/disk4488	16386
cust 0 355	329	16100	CUST 0	B.Q3.11	/dev/rdisk/disk4654	16386
cust 0 356	291	16100	CUST 0	A.Q9.35	/dev/rdisk/disk6141	16386
cust_0_357	467	16100	CUST_0	C.Q4.14	/dev/rdisk/disk6698	16386
cust 0 358	495	16100	CUST 0	A.Q5.17	/dev/rdisk/disk4876	16386
cust 0 359	499	16100	CUST 0	A.Q4.15	/dev/rdisk/disk3913	16386
cust_0_36	131	16100	CUST_0	B.Q8.32	/dev/rdisk/disk4142	16386
cust 0 360	299	16100	CUST 0	A.Q4.13	/dev/rdisk/disk5944	16386
cust_0_361	332	16100	CUST_0	A.Q1.1	/dev/rdisk/disk5323	16386
cust_0_362	415	16100	CUST_0	A.Q8.29	/dev/rdisk/disk5114	16386
cust 0 363	431	16100	CUST 0	C.Q5.18	/dev/rdisk/disk6811	16386
cust_0_364	268	16100	CUST_0	C.Q2.8	/dev/rdisk/disk6538	16386
cust 0 365	282	16100	CUST 0	B.Q10.37	/dev/rdisk/disk5220	16386
cust 0 366	462	16100	CUST 0	A.Q13.52	/dev/rdisk/disk5747	16386
cust 0 367	351	16100	CUST 0	A.Q13.50	/dev/rdisk/disk5663	16386
cust 0 368	274	16100	CUST 0	B.Q5.19	/dev/rdisk/disk4801	16386
cust_0_369	466	16100	CUST_0	C.Q6.22	/dev/rdisk/disk6417	16386
cust 0 37	6	16100	CUST 0	C.Q3.12	/dev/rdisk/disk6372	16386
cust 0 370	293	16100	CUST 0	B.Q2.8	/dev/rdisk/disk4822	16386
cust_0_371	310	16100	CUST_0	B.Q3.10	/dev/rdisk/disk4980	16386
cust 0 372	439	16100	CUST 0	B.Q9.33	/dev/rdisk/disk4504	16386
cust_0_373	327	16100	CUST_0	A.Q6.24	/dev/rdisk/disk3770	16386
cust 0 374	447	16100	CUST 0	A.Q12.48	/dev/rdisk/disk5400	16386
cust 0 375	505	16100	CUST 0	B.Q4.13	/dev/rdisk/disk4590	16386
cust_0_376	371	16100	CUST_0	B.Q12.46	/dev/rdisk/disk6096	16386
cust 0 377	490	16100	CUST 0	A.Q9.34	/dev/rdisk/disk6063	16386
cust 0 378	346	16100	CUST 0	A.Q10.38	/dev/rdisk/disk5017	16386
cust_0_379	460	16100	CUST_0	C.Q3.9	/dev/rdisk/disk6337	16386
cust 0 38	15	16100	CUST 0	B.Q12.48	/dev/rdisk/disk4769	16386

cust 0 380	459	16100	CUST 0	A.Q11.43	/dev/rdisk/disk6251	16386
cust 0 381	298	16100	CUST 0	C.Q3.10	/dev/rdisk/disk6958	16386
cust_0_382	501	16100	CUST_0	C.Q1.1	/dev/rdisk/disk6564	16386
cust 0 383	307	16100	CUST 0	A.Q12.47	/dev/rdisk/disk5547	16386
cust 0 384	414	16100	CUST 0	C.Q4.16	/dev/rdisk/disk6840	16386
cust_0_385	408	16100	CUST_0	B.Q4.15	/dev/rdisk/disk4734	16386
cust 0 386	445	16100	CUST 0	B.Q10.39	/dev/rdisk/disk4873	16386
cust_0_387	502	16100	CUST_0	B.Q12.47	/dev/rdisk/disk5600	16386
cust_0_388	373	16100	CUST_0	A.Q11.44	/dev/rdisk/disk6224	16386
cust 0 389	417	16100	CUST 0	A.Q2.5	/dev/rdisk/disk3860	16386
cust_0_39	16	16100	CUST_0	B.Q7.27	/dev/rdisk/disk4307	16386
cust 0 390	317	16100	CUST 0	B.Q8.32	/dev/rdisk/disk4150	16386
cust 0 391	331	16100	CUST 0	C.Q3.12	/dev/rdisk/disk6319	16386
cust 0 392	480	16100	CUST 0	B.Q12.48	/dev/rdisk/disk5053	16386
cust 0 393	272	16100	CUST 0	B.Q7.27	/dev/rdisk/disk5283	16386
cust_0_394	493	16100	CUST_0	B.Q1.4	/dev/rdisk/disk4926	16386
cust 0 395	498	16100	CUST 0	B.Q13.51	/dev/rdisk/disk4257	16386
cust 0 396	391	16100	CUST 0	B.Q13.49	/dev/rdisk/disk5830	16386
cust_0_397	286	16100	CUST_0	C.Q6.24	/dev/rdisk/disk6501	16386
cust 0 398	308	16100	CUST 0	A.Q8.31	/dev/rdisk/disk4898	16386
cust_0_399	324	16100	CUST_0	A.Q1.2	/dev/rdisk/disk4597	16386
cust 0 4	246	16100	CUST 0	A.Q5.17	/dev/rdisk/disk4959	16386
cust 0 40	14	16100	CUST 0	B.Q1.4	/dev/rdisk/disk4925	16386
cust_0_400	302	16100	CUST_0	B.Q6.23	/dev/rdisk/disk3992	16386
cust 0 401	292	16100	CUST 0	B.Q7.28	/dev/rdisk/disk5756	16386
cust 0 402	461	16100	CUST 0	A.Q5.19	/dev/rdisk/disk4703	16386
cust_0_403	334	16100	CUST_0	B.Q3.9	/dev/rdisk/disk5898	16386
cust 0 404	383	16100	CUST 0	B.Q13.52	/dev/rdisk/disk6910	16386
cust 0 405	475	16100	CUST 0	A.Q12.46	/dev/rdisk/disk5307	16386
cust_0_406	279	16100	CUST_0	A.Q2.6	/dev/rdisk/disk3636	16386
cust 0 407	320	16100	CUST 0	A.Q9.33	/dev/rdisk/disk5620	16386
cust_0_408	438	16100	CUST_0	B.Q10.40	/dev/rdisk/disk4866	16386
cust_0_409	411	16100	CUST_0	B.Q13.50	/dev/rdisk/disk4578	16386
cust 0 41	203	16100	CUST 0	B.Q13.51	/dev/rdisk/disk4255	16386
cust_0_410	451	16100	CUST_0	A.Q13.51	/dev/rdisk/disk4330	16386
cust 0 411	325	16100	CUST 0	B.Q5.20	/dev/rdisk/disk4768	16386
cust 0 412	384	16100	CUST 0	B.Q1.1	/dev/rdisk/disk5507	16386
cust 0 413	328	16100	CUST 0	C.Q2.7	/dev/rdisk/disk6793	16386
cust 0 414	496	16100	CUST 0	C.Q1.3	/dev/rdisk/disk6877	16386
cust_0_415	311	16100	CUST_0	C.Q1.4	/dev/rdisk/disk6514	16386
cust 0 416	424	16100	CUST 0	C.Q6.21	/dev/rdisk/disk6402	16386
cust 0 417	422	16100	CUST 0	C.Q5.17	/dev/rdisk/disk6783	16386
cust_0_418	434	16100	CUST_0	B.Q4.14	/dev/rdisk/disk5904	16386
cust 0 419	360	16100	CUST 0	A.Q1.4	/dev/rdisk/disk6933	16386
cust_0_42	208	16100	CUST_0	B.Q13.49	/dev/rdisk/disk4328	16386
cust 0 420	264	16100	CUST 0	A.Q3.9	/dev/rdisk/disk3890	16386
cust 0 421	379	16100	CUST 0	A.Q2.8	/dev/rdisk/disk3674	16386
cust_0_422	361	16100	CUST_0	A.Q2.7	/dev/rdisk/disk3639	16386
cust 0 423	506	16100	CUST 0	B.Q1.3	/dev/rdisk/disk5534	16386
cust 0 424	362	16100	CUST 0	B.Q5.18	/dev/rdisk/disk6153	16386
cust_0_425	294	16100	CUST_0	B.Q9.34	/dev/rdisk/disk5501	16386
cust 0 426	465	16100	CUST 0	C.Q4.15	/dev/rdisk/disk6674	16386

cust 0 427	463	16100	CUST 0	B.Q3.12	/dev/rdisk/disk4764	16386
cust 0 428	478	16100	CUST 0	A.Q7.26	/dev/rdisk/disk4098	16386
cust_0_429	425	16100	CUST_0	A.Q3.10	/dev/rdisk/disk3842	16386
cust 0 43	151	16100	CUST 0	C.Q6.24	/dev/rdisk/disk6467	16386
cust 0 430	458	16100	CUST 0	A.Q6.21	/dev/rdisk/disk4906	16386
cust_0_431	271	16100	CUST_0	C.Q3.11	/dev/rdisk/disk6312	16386
cust 0 432	454	16100	CUST 0	A.Q7.25	/dev/rdisk/disk6072	16386
cust_0_433	497	16100	CUST_0	A.Q4.16	/dev/rdisk/disk4027	16386
cust_0_434	273	16100	CUST_0	B.Q2.6	/dev/rdisk/disk5871	16386
cust 0 435	326	16100	CUST 0	A.Q3.11	/dev/rdisk/disk3784	16386
cust_0_436	508	16100	CUST_0	A.Q5.18	/dev/rdisk/disk4484	16386
cust 0 437	430	16100	CUST 0	A.Q5.20	/dev/rdisk/disk4570	16386
cust 0 438	457	16100	CUST 0	A.Q4.14	/dev/rdisk/disk4057	16386
cust 0 439	388	16100	CUST 0	A.Q1.3	/dev/rdisk/disk4454	16386
cust 0 44	177	16100	CUST 0	A.Q8.31	/dev/rdisk/disk5297	16386
cust_0_440	427	16100	CUST_0	A.Q10.39	/dev/rdisk/disk5031	16386
cust 0 441	473	16100	CUST 0	B.Q8.31	/dev/rdisk/disk5424	16386
cust 0 442	513	16100	CUST 0	B.Q10.38	/dev/rdisk/disk4830	16386
cust_0_443	284	16100	CUST_0	B.Q2.5	/dev/rdisk/disk5962	16386
cust 0 444	412	16100	CUST 0	B.Q6.21	/dev/rdisk/disk3859	16386
cust_0_445	289	16100	CUST_0	A.Q11.42	/dev/rdisk/disk6189	16386
cust 0 446	380	16100	CUST 0	A.Q10.40	/dev/rdisk/disk6011	16386
cust 0 447	400	16100	CUST 0	B.Q1.2	/dev/rdisk/disk5548	16386
cust_0_448	368	16100	CUST_0	C.Q2.5	/dev/rdisk/disk6399	16386
cust 0 449	479	16100	CUST 0	A.Q8.30	/dev/rdisk/disk5684	16386
cust 0 45	172	16100	CUST 0	A.Q1.2	/dev/rdisk/disk4575	16386
cust_0_450	353	16100	CUST_0	A.Q3.12	/dev/rdisk/disk4001	16386
cust 0 451	341	16100	CUST 0	B.Q6.24	/dev/rdisk/disk4302	16386
cust 0 452	406	16100	CUST 0	B.Q2.7	/dev/rdisk/disk4118	16386
cust_0_453	474	16100	CUST_0	C.Q6.23	/dev/rdisk/disk6568	16386
cust 0 454	343	16100	CUST 0	C.Q4.13	/dev/rdisk/disk6809	16386
cust_0_455	401	16100	CUST_0	A.Q9.36	/dev/rdisk/disk5421	16386
cust_0_456	363	16100	CUST_0	B.Q6.22	/dev/rdisk/disk3746	16386
cust 0 457	338	16100	CUST 0	A.Q6.23	/dev/rdisk/disk5426	16386
cust_0_458	446	16100	CUST_0	B.Q4.16	/dev/rdisk/disk4556	16386
cust 0 459	372	16100	CUST 0	A.Q7.28	/dev/rdisk/disk5988	16386
cust 0 46	76	16100	CUST 0	B.Q6.23	/dev/rdisk/disk4137	16386
cust 0 460	359	16100	CUST 0	A.Q6.22	/dev/rdisk/disk6068	16386
cust 0 461	468	16100	CUST 0	A.Q8.32	/dev/rdisk/disk5198	16386
cust_0_462	386	16100	CUST_0	B.Q8.29	/dev/rdisk/disk3960	16386
cust 0 463	394	16100	CUST 0	C.Q2.6	/dev/rdisk/disk6612	16386
cust 0 464	312	16100	CUST 0	A.Q7.27	/dev/rdisk/disk5881	16386
cust_0_465	469	16100	CUST_0	C.Q1.2	/dev/rdisk/disk6471	16386
cust 0 466	280	16100	CUST 0	C.Q5.19	/dev/rdisk/disk6642	16386
cust_0_467	367	16100	CUST_0	B.Q7.26	/dev/rdisk/disk5483	16386
cust 0 468	309	16100	CUST 0	C.Q5.20	/dev/rdisk/disk6645	16386
cust 0 469	410	16100	CUST 0	B.Q8.30	/dev/rdisk/disk3714	16386
cust_0_47	75	16100	CUST_0	B.Q7.28	/dev/rdisk/disk4289	16386
cust 0 470	366	16100	CUST 0	B.Q7.25	/dev/rdisk/disk5750	16386
cust 0 471	429	16100	CUST 0	B.Q12.45	/dev/rdisk/disk5235	16386
cust_0_472	374	16100	CUST_0	B.Q5.17	/dev/rdisk/disk4473	16386
cust 0 473	492	16100	CUST 0	B.Q3.11	/dev/rdisk/disk4667	16386

cust 0 474	355	16100	CUST 0	A.Q9.35	/dev/rdisk/disk6134	16386
cust 0 475	409	16100	CUST 0	C.Q4.14	/dev/rdisk/disk6694	16386
cust_0_476	516	16100	CUST_0	A.Q5.17	/dev/rdisk/disk4920	16386
cust 0 477	330	16100	CUST 0	A.Q4.15	/dev/rdisk/disk3922	16386
cust 0 478	426	16100	CUST 0	A.Q4.13	/dev/rdisk/disk4050	16386
cust_0_479	428	16100	CUST_0	A.Q1.1	/dev/rdisk/disk5324	16386
cust 0 48	89	16100	CUST 0	A.Q5.19	/dev/rdisk/disk4698	16386
cust_0_480	392	16100	CUST_0	A.Q8.29	/dev/rdisk/disk5093	16386
cust_0_481	336	16100	CUST_0	C.Q5.18	/dev/rdisk/disk6819	16386
cust 0 482	370	16100	CUST 0	C.Q2.8	/dev/rdisk/disk6533	16386
cust_0_483	375	16100	CUST_0	B.Q10.37	/dev/rdisk/disk5241	16386
cust 0 484	387	16100	CUST 0	A.Q13.52	/dev/rdisk/disk5725	16386
cust 0 485	481	16100	CUST 0	A.Q13.50	/dev/rdisk/disk5166	16386
cust 0 486	398	16100	CUST 0	B.Q5.19	/dev/rdisk/disk4820	16386
cust 0 487	281	16100	CUST 0	C.Q6.22	/dev/rdisk/disk6379	16386
cust_0_488	486	16100	CUST_0	B.Q2.8	/dev/rdisk/disk5202	16386
cust 0 489	477	16100	CUST 0	B.Q3.10	/dev/rdisk/disk4982	16386
cust 0 49	253	16100	CUST 0	B.Q3.9	/dev/rdisk/disk5930	16386
cust_0_490	476	16100	CUST_0	B.Q9.33	/dev/rdisk/disk4508	16386
cust 0 491	358	16100	CUST 0	A.Q6.24	/dev/rdisk/disk3766	16386
cust_0_492	393	16100	CUST_0	A.Q12.48	/dev/rdisk/disk5398	16386
cust 0 493	416	16100	CUST 0	B.Q4.13	/dev/rdisk/disk4609	16386
cust 0 494	452	16100	CUST 0	B.Q12.46	/dev/rdisk/disk6094	16386
cust_0_495	364	16100	CUST_0	A.Q9.34	/dev/rdisk/disk6062	16386
cust 0 496	381	16100	CUST 0	A.Q10.38	/dev/rdisk/disk4433	16386
cust 0 497	491	16100	CUST 0	C.Q3.9	/dev/rdisk/disk6330	16386
cust_0_498	507	16100	CUST_0	A.Q11.43	/dev/rdisk/disk6246	16386
cust 0 499	297	16100	CUST 0	C.Q3.10	/dev/rdisk/disk6956	16386
cust 0 5	202	16100	CUST 0	A.Q4.15	/dev/rdisk/disk3945	16386
cust_0_50	44	16100	CUST_0	B.Q13.52	/dev/rdisk/disk6919	16386
cust 0 500	345	16100	CUST 0	C.Q1.1	/dev/rdisk/disk6639	16386
cust_0_501	482	16100	CUST_0	A.Q12.47	/dev/rdisk/disk5550	16386
cust_0_502	500	16100	CUST_0	C.Q4.16	/dev/rdisk/disk6848	16386
cust 0 503	405	16100	CUST 0	B.Q4.15	/dev/rdisk/disk6181	16386
cust_0_504	301	16100	CUST_0	B.Q10.39	/dev/rdisk/disk4875	16386
cust 0 505	382	16100	CUST 0	B.Q12.47	/dev/rdisk/disk5610	16386
cust 0 506	342	16100	CUST 0	A.Q11.44	/dev/rdisk/disk6226	16386
cust 0 507	285	16100	CUST 0	A.Q2.5	/dev/rdisk/disk3857	16386
cust 0 508	453	16100	CUST 0	B.Q8.32	/dev/rdisk/disk4148	16386
cust_0_509	455	16100	CUST_0	C.Q3.12	/dev/rdisk/disk6323	16386
cust 0 51	95	16100	CUST 0	A.Q12.46	/dev/rdisk/disk4870	16386
cust 0 510	517	16100	CUST 0	B.Q12.48	/dev/rdisk/disk4407	16386
cust_0_511	456	16100	CUST_0	B.Q7.27	/dev/rdisk/disk5285	16386
cust 0 512	515	16100	CUST 0	B.Q1.4	/dev/rdisk/disk5580	16386
cust_0_513	640	16100	CUST_0	B.Q13.51	/dev/rdisk/disk4249	16386
cust 0 514	648	16100	CUST 0	B.Q13.49	/dev/rdisk/disk4261	16386
cust 0 515	596	16100	CUST 0	C.Q6.24	/dev/rdisk/disk6489	16386
cust_0_516	606	16100	CUST_0	A.Q8.31	/dev/rdisk/disk5363	16386
cust 0 517	548	16100	CUST 0	A.Q1.2	/dev/rdisk/disk4623	16386
cust 0 518	521	16100	CUST 0	B.Q6.23	/dev/rdisk/disk3991	16386
cust_0_519	610	16100	CUST_0	B.Q7.28	/dev/rdisk/disk5755	16386
cust 0 52	113	16100	CUST 0	A.Q2.6	/dev/rdisk/disk3641	16386

cust 0 520	615	16100	CUST 0	A.Q5.19	/dev/rdisk/disk4661	16386
cust 0 521	561	16100	CUST 0	B.Q3.9	/dev/rdisk/disk5896	16386
cust_0_522	662	16100	CUST_0	B.Q13.52	/dev/rdisk/disk6909	16386
cust 0 523	583	16100	CUST 0	A.Q12.46	/dev/rdisk/disk5294	16386
cust 0 524	641	16100	CUST 0	A.Q2.6	/dev/rdisk/disk3633	16386
cust_0_525	539	16100	CUST_0	A.Q9.33	/dev/rdisk/disk5049	16386
cust 0 526	642	16100	CUST 0	B.Q10.40	/dev/rdisk/disk4863	16386
cust_0_527	624	16100	CUST_0	B.Q13.50	/dev/rdisk/disk4581	16386
cust_0_528	653	16100	CUST_0	A.Q13.51	/dev/rdisk/disk4331	16386
cust 0 529	590	16100	CUST 0	B.Q5.20	/dev/rdisk/disk4395	16386
cust_0_53	154	16100	CUST_0	A.Q9.33	/dev/rdisk/disk5681	16386
cust 0 530	585	16100	CUST 0	B.Q1.1	/dev/rdisk/disk5505	16386
cust 0 531	603	16100	CUST 0	C.Q2.7	/dev/rdisk/disk6787	16386
cust 0 532	643	16100	CUST 0	C.Q1.3	/dev/rdisk/disk6876	16386
cust 0 533	559	16100	CUST 0	C.Q1.4	/dev/rdisk/disk6502	16386
cust_0_534	605	16100	CUST_0	C.Q6.21	/dev/rdisk/disk6441	16386
cust 0 535	584	16100	CUST 0	C.Q5.17	/dev/rdisk/disk6812	16386
cust 0 536	663	16100	CUST 0	B.Q4.14	/dev/rdisk/disk5905	16386
cust_0_537	562	16100	CUST_0	A.Q1.4	/dev/rdisk/disk6932	16386
cust 0 538	563	16100	CUST 0	A.Q3.9	/dev/rdisk/disk3889	16386
cust_0_539	554	16100	CUST_0	A.Q2.8	/dev/rdisk/disk3673	16386
cust 0 54	80	16100	CUST 0	B.Q10.40	/dev/rdisk/disk5334	16386
cust 0 540	609	16100	CUST 0	A.Q2.7	/dev/rdisk/disk3637	16386
cust_0_541	557	16100	CUST_0	B.Q1.3	/dev/rdisk/disk5536	16386
cust 0 542	616	16100	CUST 0	B.Q5.18	/dev/rdisk/disk6146	16386
cust 0 543	604	16100	CUST 0	B.Q9.34	/dev/rdisk/disk5500	16386
cust_0_544	654	16100	CUST_0	C.Q4.15	/dev/rdisk/disk6675	16386
cust 0 545	636	16100	CUST 0	B.Q3.12	/dev/rdisk/disk4765	16386
cust 0 546	627	16100	CUST 0	A.Q7.26	/dev/rdisk/disk5657	16386
cust_0_547	635	16100	CUST_0	A.Q3.10	/dev/rdisk/disk3843	16386
cust 0 548	564	16100	CUST 0	A.Q6.21	/dev/rdisk/disk4545	16386
cust_0_549	567	16100	CUST_0	C.Q3.11	/dev/rdisk/disk6304	16386
cust_0_55	143	16100	CUST_0	B.Q13.50	/dev/rdisk/disk4701	16386
cust 0 550	588	16100	CUST 0	A.Q7.25	/dev/rdisk/disk6069	16386
cust_0_551	531	16100	CUST_0	A.Q4.16	/dev/rdisk/disk4045	16386
cust 0 552	655	16100	CUST 0	B.Q2.6	/dev/rdisk/disk4173	16386
cust 0 553	551	16100	CUST 0	A.Q3.11	/dev/rdisk/disk3788	16386
cust 0 554	617	16100	CUST 0	A.Q5.18	/dev/rdisk/disk4489	16386
cust 0 555	614	16100	CUST 0	A.Q5.20	/dev/rdisk/disk4544	16386
cust_0_556	540	16100	CUST_0	A.Q4.14	/dev/rdisk/disk4059	16386
cust 0 557	661	16100	CUST 0	A.Q1.3	/dev/rdisk/disk4457	16386
cust 0 558	572	16100	CUST 0	A.Q10.39	/dev/rdisk/disk4791	16386
cust_0_559	520	16100	CUST_0	B.Q8.31	/dev/rdisk/disk5420	16386
cust 0 56	231	16100	CUST 0	A.Q13.51	/dev/rdisk/disk4323	16386
cust_0_560	578	16100	CUST_0	B.Q10.38	/dev/rdisk/disk4819	16386
cust 0 561	630	16100	CUST 0	B.Q2.5	/dev/rdisk/disk5961	16386
cust 0 562	573	16100	CUST 0	B.Q6.21	/dev/rdisk/disk3849	16386
cust_0_563	533	16100	CUST_0	A.Q11.42	/dev/rdisk/disk6201	16386
cust 0 564	656	16100	CUST 0	A.Q10.40	/dev/rdisk/disk6012	16386
cust 0 565	522	16100	CUST 0	B.Q1.2	/dev/rdisk/disk5555	16386
cust_0_566	644	16100	CUST_0	C.Q2.5	/dev/rdisk/disk6422	16386
cust 0 567	608	16100	CUST 0	A.Q8.30	/dev/rdisk/disk5695	16386

cust 0 568	632	16100	CUST 0	A.Q3.12	/dev/rdisk/disk3999	16386
cust 0 569	545	16100	CUST 0	B.Q6.24	/dev/rdisk/disk4303	16386
cust_0_57	62	16100	CUST_0	B.Q5.20	/dev/rdisk/disk5158	16386
cust 0 570	589	16100	CUST 0	B.Q2.7	/dev/rdisk/disk5617	16386
cust 0 571	582	16100	CUST 0	C.Q6.23	/dev/rdisk/disk6531	16386
cust_0_572	647	16100	CUST_0	C.Q4.13	/dev/rdisk/disk6747	16386
cust 0 573	541	16100	CUST 0	A.Q9.36	/dev/rdisk/disk5457	16386
cust_0_574	569	16100	CUST_0	B.Q6.22	/dev/rdisk/disk3731	16386
cust_0_575	556	16100	CUST_0	A.Q6.23	/dev/rdisk/disk5447	16386
cust 0 576	633	16100	CUST 0	B.Q4.16	/dev/rdisk/disk4587	16386
cust_0_577	593	16100	CUST_0	A.Q7.28	/dev/rdisk/disk5975	16386
cust 0 578	576	16100	CUST 0	A.Q6.22	/dev/rdisk/disk6065	16386
cust 0 579	538	16100	CUST 0	A.Q8.32	/dev/rdisk/disk5107	16386
cust 0 58	35	16100	CUST 0	B.Q1.1	/dev/rdisk/disk5520	16386
cust 0 580	579	16100	CUST 0	B.Q8.29	/dev/rdisk/disk3940	16386
cust_0_581	528	16100	CUST_0	C.Q2.6	/dev/rdisk/disk6579	16386
cust 0 582	664	16100	CUST 0	A.Q7.27	/dev/rdisk/disk4181	16386
cust 0 583	659	16100	CUST 0	C.Q1.2	/dev/rdisk/disk6442	16386
cust_0_584	565	16100	CUST_0	C.Q5.19	/dev/rdisk/disk6768	16386
cust 0 585	543	16100	CUST 0	B.Q7.26	/dev/rdisk/disk5482	16386
cust_0_586	650	16100	CUST_0	C.Q5.20	/dev/rdisk/disk6785	16386
cust 0 587	527	16100	CUST 0	B.Q8.30	/dev/rdisk/disk3713	16386
cust 0 588	657	16100	CUST 0	B.Q7.25	/dev/rdisk/disk5799	16386
cust_0_589	602	16100	CUST_0	B.Q12.45	/dev/rdisk/disk5233	16386
cust 0 59	188	16100	CUST 0	C.Q2.7	/dev/rdisk/disk6757	16386
cust 0 590	586	16100	CUST 0	B.Q5.17	/dev/rdisk/disk4476	16386
cust_0_591	613	16100	CUST_0	B.Q3.11	/dev/rdisk/disk4599	16386
cust 0 592	621	16100	CUST 0	A.Q9.35	/dev/rdisk/disk6125	16386
cust 0 593	529	16100	CUST 0	C.Q4.14	/dev/rdisk/disk6795	16386
cust_0_594	665	16100	CUST_0	A.Q5.17	/dev/rdisk/disk4917	16386
cust 0 595	549	16100	CUST 0	A.Q4.15	/dev/rdisk/disk3933	16386
cust_0_596	649	16100	CUST_0	A.Q4.13	/dev/rdisk/disk4047	16386
cust_0_597	525	16100	CUST_0	A.Q1.1	/dev/rdisk/disk5310	16386
cust 0 598	645	16100	CUST 0	A.Q8.29	/dev/rdisk/disk5124	16386
cust_0_599	566	16100	CUST_0	C.Q5.18	/dev/rdisk/disk6831	16386
cust 0 6	73	16100	CUST 0	A.Q4.13	/dev/rdisk/disk5943	16386
cust 0 60	36	16100	CUST 0	C.Q1.3	/dev/rdisk/disk6879	16386
cust 0 600	574	16100	CUST 0	C.Q2.8	/dev/rdisk/disk6532	16386
cust 0 601	597	16100	CUST 0	B.Q10.37	/dev/rdisk/disk4839	16386
cust_0_602	542	16100	CUST_0	A.Q13.52	/dev/rdisk/disk5710	16386
cust 0 603	601	16100	CUST 0	A.Q13.50	/dev/rdisk/disk4315	16386
cust 0 604	580	16100	CUST 0	B.Q5.19	/dev/rdisk/disk4376	16386
cust_0_605	575	16100	CUST_0	C.Q6.22	/dev/rdisk/disk6400	16386
cust 0 606	607	16100	CUST 0	B.Q2.8	/dev/rdisk/disk4813	16386
cust_0_607	535	16100	CUST_0	B.Q3.10	/dev/rdisk/disk4965	16386
cust 0 608	628	16100	CUST 0	B.Q9.33	/dev/rdisk/disk4724	16386
cust 0 609	631	16100	CUST 0	A.Q6.24	/dev/rdisk/disk3761	16386
cust_0_61	194	16100	CUST_0	C.Q1.4	/dev/rdisk/disk6516	16386
cust 0 610	570	16100	CUST 0	A.Q12.48	/dev/rdisk/disk5800	16386
cust 0 611	558	16100	CUST 0	B.Q4.13	/dev/rdisk/disk4611	16386
cust_0_612	658	16100	CUST_0	B.Q12.46	/dev/rdisk/disk6079	16386
cust 0 613	519	16100	CUST 0	A.Q9.34	/dev/rdisk/disk6050	16386

cust 0 614	523	16100	CUST 0	A.Q10.38	/dev/rdisk/disk4438	16386
cust 0 615	620	16100	CUST 0	C.Q3.9	/dev/rdisk/disk6318	16386
cust_0_616	534	16100	CUST_0	A.Q11.43	/dev/rdisk/disk6242	16386
cust 0 617	568	16100	CUST 0	C.Q3.10	/dev/rdisk/disk6957	16386
cust 0 618	651	16100	CUST 0	C.Q1.1	/dev/rdisk/disk6632	16386
cust_0_619	536	16100	CUST_0	A.Q12.47	/dev/rdisk/disk5543	16386
cust 0 62	43	16100	CUST 0	C.Q6.21	/dev/rdisk/disk6459	16386
cust_0_620	646	16100	CUST_0	C.Q4.16	/dev/rdisk/disk6861	16386
cust_0_621	587	16100	CUST_0	B.Q4.15	/dev/rdisk/disk6178	16386
cust 0 622	544	16100	CUST 0	B.Q10.39	/dev/rdisk/disk4882	16386
cust_0_623	625	16100	CUST_0	B.Q12.47	/dev/rdisk/disk5605	16386
cust 0 624	591	16100	CUST 0	A.Q11.44	/dev/rdisk/disk6229	16386
cust 0 625	594	16100	CUST 0	A.Q2.5	/dev/rdisk/disk3855	16386
cust 0 626	634	16100	CUST 0	B.Q8.32	/dev/rdisk/disk4152	16386
cust 0 627	518	16100	CUST 0	C.Q3.12	/dev/rdisk/disk6316	16386
cust_0_628	592	16100	CUST_0	B.Q12.48	/dev/rdisk/disk4410	16386
cust 0 629	639	16100	CUST 0	B.Q7.27	/dev/rdisk/disk5350	16386
cust 0 63	96	16100	CUST 0	C.Q5.17	/dev/rdisk/disk6590	16386
cust_0_630	666	16100	CUST_0	B.Q1.4	/dev/rdisk/disk5594	16386
cust 0 631	637	16100	CUST 0	B.Q13.51	/dev/rdisk/disk4250	16386
cust_0_632	546	16100	CUST_0	B.Q13.49	/dev/rdisk/disk5822	16386
cust 0 633	626	16100	CUST 0	C.Q6.24	/dev/rdisk/disk6411	16386
cust 0 634	571	16100	CUST 0	A.Q8.31	/dev/rdisk/disk4860	16386
cust_0_635	560	16100	CUST_0	A.Q1.2	/dev/rdisk/disk4628	16386
cust 0 636	618	16100	CUST 0	B.Q6.23	/dev/rdisk/disk3987	16386
cust 0 637	530	16100	CUST 0	B.Q7.28	/dev/rdisk/disk5762	16386
cust_0_638	581	16100	CUST_0	A.Q5.19	/dev/rdisk/disk4679	16386
cust 0 639	524	16100	CUST 0	B.Q3.9	/dev/rdisk/disk5911	16386
cust 0 64	219	16100	CUST 0	B.Q4.14	/dev/rdisk/disk5883	16386
cust_0_640	622	16100	CUST_0	B.Q13.52	/dev/rdisk/disk6899	16386
cust 0 641	623	16100	CUST 0	A.Q12.46	/dev/rdisk/disk4869	16386
cust_0_642	629	16100	CUST_0	A.Q2.6	/dev/rdisk/disk3632	16386
cust_0_643	660	16100	CUST_0	A.Q9.33	/dev/rdisk/disk5566	16386
cust 0 644	552	16100	CUST 0	B.Q10.40	/dev/rdisk/disk4861	16386
cust_0_645	532	16100	CUST_0	B.Q13.50	/dev/rdisk/disk4589	16386
cust 0 646	526	16100	CUST 0	A.Q13.51	/dev/rdisk/disk4332	16386
cust 0 647	598	16100	CUST 0	B.Q5.20	/dev/rdisk/disk5134	16386
cust 0 648	612	16100	CUST 0	B.Q1.1	/dev/rdisk/disk5515	16386
cust 0 649	595	16100	CUST 0	C.Q2.7	/dev/rdisk/disk6790	16386
cust_0_65	93	16100	CUST_0	A.Q1.4	/dev/rdisk/disk6924	16386
cust 0 650	611	16100	CUST 0	C.Q1.3	/dev/rdisk/disk6875	16386
cust 0 651	555	16100	CUST 0	C.Q1.4	/dev/rdisk/disk6498	16386
cust_0_652	577	16100	CUST_0	C.Q6.21	/dev/rdisk/disk6423	16386
cust 0 653	638	16100	CUST 0	C.Q5.17	/dev/rdisk/disk6826	16386
cust_0_654	547	16100	CUST_0	B.Q4.14	/dev/rdisk/disk5280	16386
cust 0 655	537	16100	CUST 0	A.Q1.4	/dev/rdisk/disk6945	16386
cust 0 656	652	16100	CUST 0	A.Q3.9	/dev/rdisk/disk3926	16386
cust_0_657	600	16100	CUST_0	A.Q2.8	/dev/rdisk/disk3671	16386
cust 0 658	553	16100	CUST 0	A.Q2.7	/dev/rdisk/disk3642	16386
cust 0 659	599	16100	CUST 0	B.Q1.3	/dev/rdisk/disk5514	16386
cust_0_66	88	16100	CUST_0	A.Q3.9	/dev/rdisk/disk3917	16386
cust 0 660	619	16100	CUST 0	B.Q5.18	/dev/rdisk/disk6144	16386

cust 0 661	550	16100	CUST 0	B.Q9.34	/dev/rdisk/disk5488	16386
cust 0 67	71	16100	CUST 0	A.Q2.8	/dev/rdisk/disk3677	16386
cust_0_68	45	16100	CUST_0	A.Q2.7	/dev/rdisk/disk3644	16386
cust 0 69	41	16100	CUST 0	B.Q1.3	/dev/rdisk/disk5259	16386
cust 0 7	170	16100	CUST 0	A.Q1.1	/dev/rdisk/disk5320	16386
cust_0_70	226	16100	CUST_0	B.Q5.18	/dev/rdisk/disk6160	16386
cust 0 71	34	16100	CUST 0	B.Q9.34	/dev/rdisk/disk5205	16386
cust_0_72	90	16100	CUST_0	C.Q4.15	/dev/rdisk/disk6676	16386
cust_0_73	100	16100	CUST_0	B.Q3.12	/dev/rdisk/disk4782	16386
cust 0 74	39	16100	CUST 0	A.Q7.26	/dev/rdisk/disk5737	16386
cust_0_75	128	16100	CUST_0	A.Q3.10	/dev/rdisk/disk3853	16386
cust 0 76	25	16100	CUST 0	A.Q6.21	/dev/rdisk/disk4468	16386
cust 0 77	61	16100	CUST 0	C.Q3.11	/dev/rdisk/disk6320	16386
cust 0 78	85	16100	CUST 0	A.Q7.25	/dev/rdisk/disk6090	16386
cust 0 79	142	16100	CUST 0	A.Q4.16	/dev/rdisk/disk4034	16386
cust_0_8	99	16100	CUST_0	A.Q8.29	/dev/rdisk/disk5084	16386
cust 0 80	105	16100	CUST 0	B.Q2.6	/dev/rdisk/disk5875	16386
cust 0 81	153	16100	CUST 0	A.Q3.11	/dev/rdisk/disk3790	16386
cust_0_82	209	16100	CUST_0	A.Q5.18	/dev/rdisk/disk4464	16386
cust 0 83	28	16100	CUST 0	A.Q5.20	/dev/rdisk/disk4497	16386
cust_0_84	42	16100	CUST_0	A.Q4.14	/dev/rdisk/disk4022	16386
cust 0 85	7	16100	CUST 0	A.Q1.3	/dev/rdisk/disk5469	16386
cust 0 86	40	16100	CUST 0	A.Q10.39	/dev/rdisk/disk4795	16386
cust_0_87	10	16100	CUST_0	B.Q8.31	/dev/rdisk/disk5434	16386
cust 0 88	215	16100	CUST 0	B.Q10.38	/dev/rdisk/disk5240	16386
cust 0 89	192	16100	CUST 0	B.Q2.5	/dev/rdisk/disk5965	16386
cust_0_9	160	16100	CUST_0	C.Q5.18	/dev/rdisk/disk6863	16386
cust 0 90	165	16100	CUST 0	B.Q6.21	/dev/rdisk/disk3864	16386
cust 0 91	104	16100	CUST 0	A.Q11.42	/dev/rdisk/disk6219	16386
cust_0_92	66	16100	CUST_0	A.Q10.40	/dev/rdisk/disk6009	16386
cust 0 93	147	16100	CUST 0	B.Q1.2	/dev/rdisk/disk5570	16386
cust_0_94	237	16100	CUST_0	C.Q2.5	/dev/rdisk/disk6605	16386
cust_0_95	122	16100	CUST_0	A.Q8.30	/dev/rdisk/disk5795	16386
cust 0 96	52	16100	CUST 0	A.Q3.12	/dev/rdisk/disk4003	16386
cust_0_97	33	16100	CUST_0	B.Q6.24	/dev/rdisk/disk5404	16386
cust 0 98	214	16100	CUST 0	B.Q2.7	/dev/rdisk/disk4113	16386
cust 0 99	97	16100	CUST 0	C.Q6.23	/dev/rdisk/disk6582	16386
dist 0 0	667	7320	DIST 0	All	/dev/vgdata2/rdist 0 0	7552
hist 0 0	668	16383	HIST 0	All	/dev/vgdata1/rhist 0 0	16992
hist_0_1	691	16383	HIST_0	All	/dev/vgdata2/rhist_0_1	16992
hist 0 10	720	16383	HIST 0	All	/dev/vgdata1/rhist 0 10	16992
hist 0 11	688	16383	HIST 0	All	/dev/vgdata2/rhist 0 11	16992
hist_0_12	676	16383	HIST_0	All	/dev/vgdata1/rhist_0_12	16992
hist 0 13	692	16383	HIST 0	All	/dev/vgdata2/rhist 0 13	16992
hist_0_14	708	16383	HIST_0	All	/dev/vgdata1/rhist_0_14	16992
hist 0 15	714	16383	HIST 0	All	/dev/vgdata2/rhist 0 15	16992
hist 0 16	695	16383	HIST 0	All	/dev/vgdata1/rhist 0 16	16992
hist_0_17	703	16383	HIST_0	All	/dev/vgdata2/rhist_0_17	16992
hist 0 18	672	16383	HIST 0	All	/dev/vgdata1/rhist 0 18	16992
hist 0 19	682	16383	HIST 0	All	/dev/vgdata2/rhist 0 19	16992
hist_0_2	677	16383	HIST_0	All	/dev/vgdata1/rhist_0_2	16992
hist 0 20	685	16383	HIST 0	All	/dev/vgdata1/rhist 0 20	16992

hist 0 21	715	16383	HIST 0	All	/dev/vgdata2/rhist 0 21	16992
hist 0 22	721	16383	HIST 0	All	/dev/vgdata1/rhist 0 22	16992
hist_0_23	718	16383	HIST_0	All	/dev/vgdata2/rhist_0_23	16992
hist 0 24	690	16383	HIST 0	All	/dev/vgdata1/rhist 0 24	16992
hist 0 25	670	16383	HIST 0	All	/dev/vgdata2/rhist 0 25	16992
hist_0_26	679	16383	HIST_0	All	/dev/vgdata1/rhist_0_26	16992
hist 0 27	684	16383	HIST 0	All	/dev/vgdata2/rhist 0 27	16992
hist_0_28	699	16383	HIST_0	All	/dev/vgdata1/rhist_0_28	16992
hist_0_29	709	16383	HIST_0	All	/dev/vgdata2/rhist_0_29	16992
hist 0 3	681	16383	HIST 0	All	/dev/vgdata2/rhist 0 3	16992
hist_0_30	680	16383	HIST_0	All	/dev/vgdata1/rhist_0_30	16992
hist 0 31	696	16383	HIST 0	All	/dev/vgdata2/rhist 0 31	16992
hist 0 32	710	16383	HIST 0	All	/dev/vgdata1/rhist 0 32	16992
hist 0 33	716	16383	HIST 0	All	/dev/vgdata2/rhist 0 33	16992
hist 0 34	711	16383	HIST 0	All	/dev/vgdata1/rhist 0 34	16992
hist_0_35	706	16383	HIST_0	All	/dev/vgdata2/rhist_0_35	16992
hist 0 36	689	16383	HIST 0	All	/dev/vgdata1/rhist 0 36	16992
hist 0 37	722	16383	HIST 0	All	/dev/vgdata2/rhist 0 37	16992
hist_0_38	700	16383	HIST_0	All	/dev/vgdata1/rhist_0_38	16992
hist 0 39	687	16383	HIST 0	All	/dev/vgdata2/rhist 0 39	16992
hist_0_4	698	16383	HIST_0	All	/dev/vgdata1/rhist_0_4	16992
hist 0 40	674	16383	HIST 0	All	/dev/vgdata1/rhist 0 40	16992
hist 0 41	678	16383	HIST 0	All	/dev/vgdata2/rhist 0 41	16992
hist_0_42	686	16383	HIST_0	All	/dev/vgdata1/rhist_0_42	16992
hist 0 43	697	16383	HIST 0	All	/dev/vgdata2/rhist 0 43	16992
hist 0 44	719	16383	HIST 0	All	/dev/vgdata1/rhist 0 44	16992
hist_0_45	712	16383	HIST_0	All	/dev/vgdata2/rhist_0_45	16992
hist 0 46	693	16383	HIST 0	All	/dev/vgdata1/rhist 0 46	16992
hist 0 47	671	16383	HIST 0	All	/dev/vgdata2/rhist 0 47	16992
hist_0_48	707	16383	HIST_0	All	/dev/vgdata1/rhist_0_48	16992
hist 0 49	717	16383	HIST 0	All	/dev/vgdata2/rhist 0 49	16992
hist_0_5	713	16383	HIST_0	All	/dev/vgdata2/rhist_0_5	16992
hist_0_50	705	16383	HIST_0	All	/dev/vgdata1/rhist_0_50	16992
hist 0 51	702	16383	HIST 0	All	/dev/vgdata2/rhist 0 51	16992
hist_0_52	675	16383	HIST_0	All	/dev/vgdata1/rhist_0_52	16992
hist 0 53	704	16383	HIST 0	All	/dev/vgdata2/rhist 0 53	16992
hist 0 6	673	16383	HIST 0	All	/dev/vgdata1/rhist 0 6	16992
hist 0 7	701	16383	HIST 0	All	/dev/vgdata2/rhist 0 7	16992
hist 0 8	694	16383	HIST 0	All	/dev/vgdata1/rhist 0 8	16992
hist_0_9	683	16383	HIST_0	All	/dev/vgdata2/rhist_0_9	16992
icust1 0 0	752	65535	ICUST1	0 All	/dev/vgdata1/ricust1 0 0	66080
icust1 0 1	763	65535	ICUST1	0 All	/dev/vgdata2/ricust1 0 1	66080
icust1_0_2	858	65535	ICUST1_	0 All	/dev/vgdata1/ricust1_0_2	66080
icust1 0 3	893	65535	ICUST1	0 All	/dev/vgdata2/ricust1 0 3	66080
icust2_0_0	856	65535	ICUST2_	0 All	/dev/vgdata1/ricust2_0_0	66080
icust2 0 1	917	65535	ICUST2	0 All	/dev/vgdata2/ricust2 0 1	66080
icust2 0 2	915	65535	ICUST2	0 All	/dev/vgdata1/ricust2 0 2	66080
icust2_0_3	945	65535	ICUST2_	0 All	/dev/vgdata2/ricust2_0_3	66080
icust2 0 4	894	65535	ICUST2	0 All	/dev/vgdata1/ricust2 0 4	66080
icust2 0 5	914	65535	ICUST2	0 All	/dev/vgdata2/ricust2 0 5	66080
icust2_0_6	946	65535	ICUST2_	0 All	/dev/vgdata1/ricust2_0_6	66080
icust2 0 7	892	65535	ICUST2	0 All	/dev/vgdata2/ricust2 0 7	66080

idist 0 0	857	1710	IDIST 0	All	/dev/vgdata1/ridist 0 0	1888
iitem 0 0	891	20	IITEM 0	All	/dev/vgdata2/riitem 0 0	1888
iordr2_0_0	921	65535	IORDR2_	0 All	/dev/vgdata2/riordr2_0_0	66080
iordr2 0 1	937	65535	IORDR2	0 All	/dev/vgdata1/riordr2 0 1	66080
iordr2 0 2	933	65535	IORDR2	0 All	/dev/vgdata2/riordr2 0 2	66080
iordr2_0_3	935	65535	IORDR2_	0 All	/dev/vgdata1/riordr2_0_3	66080
iordr2 0 4	934	65535	IORDR2	0 All	/dev/vgdata2/riordr2 0 4	66080
iordr2_0_5	943	65535	IORDR2_	0 All	/dev/vgdata1/riordr2_0_5	66080
iordr2_0_6	944	65535	IORDR2_	0 All	/dev/vgdata2/riordr2_0_6	66080
iordr2 0 7	936	65535	IORDR2	0 All	/dev/vgdata1/riordr2 0 7	66080
iordr2_0_8	953	65535	IORDR2_	0 All	/dev/vgdata2/riordr2_0_8	66080
istok 0 0	890	65535	ISTOK 0	All	/dev/vgdata1/ristok 0 0	66080
istok 0 1	918	65535	ISTOK 0	All	/dev/vgdata2/ristok 0 1	66080
istok 0 10	916	65535	ISTOK 0	All	/dev/vgdata1/ristok 0 10	66080
istok 0 2	920	65535	ISTOK 0	All	/dev/vgdata1/ristok 0 2	66080
istok_0_3	923	65535	ISTOK_0	All	/dev/vgdata2/ristok_0_3	66080
istok 0 4	931	65535	ISTOK 0	All	/dev/vgdata1/ristok 0 4	66080
istok 0 5	930	65535	ISTOK 0	All	/dev/vgdata2/ristok 0 5	66080
istok_0_6	922	65535	ISTOK_0	All	/dev/vgdata1/ristok_0_6	66080
istok 0 7	932	65535	ISTOK 0	All	/dev/vgdata2/ristok 0 7	66080
istok_0_8	919	65535	ISTOK_0	All	/dev/vgdata1/ristok_0_8	66080
istok 0 9	929	65535	ISTOK 0	All	/dev/vgdata2/ristok 0 9	66080
item 0 0	669	20	ITEM 0	All	/dev/vgdata1/ritem 0 0	1888
iware_0_0	725	430	IWARE_0	All	/dev/vgdata1/riware_0_0	1888
nord 0 0	724	8191	NORD 0	All	/dev/vgdata1/rnord 0 0	9440
nord 0 1	730	8191	NORD 0	All	/dev/vgdata2/rnord 0 1	9440
nord_0_10	738	8191	NORD_0	All	/dev/vgdata1/rnord_0_10	9440
nord 0 11	729	8191	NORD 0	All	/dev/vgdata2/rnord 0 11	9440
nord 0 12	736	8191	NORD 0	All	/dev/vgdata1/rnord 0 12	9440
nord_0_13	737	8191	NORD_0	All	/dev/vgdata2/rnord_0_13	9440
nord 0 14	726	8191	NORD 0	All	/dev/vgdata1/rnord 0 14	9440
nord_0_15	727	8191	NORD_0	All	/dev/vgdata2/rnord_0_15	9440
nord_0_2	741	8191	NORD_0	All	/dev/vgdata1/rnord_0_2	9440
nord 0 3	728	8191	NORD 0	All	/dev/vgdata2/rnord 0 3	9440
nord_0_4	743	8191	NORD_0	All	/dev/vgdata1/rnord_0_4	9440
nord 0 5	733	8191	NORD 0	All	/dev/vgdata2/rnord 0 5	9440
nord 0 6	732	8191	NORD 0	All	/dev/vgdata1/rnord 0 6	9440
nord 0 7	735	8191	NORD 0	All	/dev/vgdata2/rnord 0 7	9440
nord 0 8	734	8191	NORD 0	All	/dev/vgdata1/rnord 0 8	9440
nord_0_9	731	8191	NORD_0	All	/dev/vgdata2/rnord_0_9	9440
ordr 0 0	723	65535	ORDR 0	All	/dev/vgdata1/rordr 0 0	66080
ordr 0 1	924	65535	ORDR 0	All	/dev/vgdata2/rordr 0 1	66080
ordr_0_10	767	65535	ORDR_0	All	/dev/vgdata1/rordr_0_10	66080
ordr 0 100	866	65535	ORDR 0	All	/dev/vgdata1/rordr 0 100	66080
ordr_0_101	777	65535	ORDR_0	All	/dev/vgdata2/rordr_0_101	66080
ordr 0 102	748	65535	ORDR 0	All	/dev/vgdata1/rordr 0 102	66080
ordr 0 103	823	65535	ORDR 0	All	/dev/vgdata2/rordr 0 103	66080
ordr_0_104	805	65535	ORDR_0	All	/dev/vgdata1/rordr_0_104	66080
ordr 0 105	880	65535	ORDR 0	All	/dev/vgdata2/rordr 0 105	66080
ordr 0 106	826	65535	ORDR 0	All	/dev/vgdata1/rordr 0 106	66080
ordr_0_107	899	65535	ORDR_0	All	/dev/vgdata2/rordr_0_107	66080
ordr 0 108	830	65535	ORDR 0	All	/dev/vgdata1/rordr 0 108	66080

ordr 0 109	781	65535	ORDR 0	All	/dev/vgdata2/rordr 0 109	66080
ordr 0 11	833	65535	ORDR 0	All	/dev/vgdata2/rordr 0 11	66080
ordr_0_110	868	65535	ORDR_0	All	/dev/vgdata1/rordr_0_110	66080
ordr 0 111	877	65535	ORDR 0	All	/dev/vgdata2/rordr 0 111	66080
ordr 0 112	816	65535	ORDR 0	All	/dev/vgdata1/rordr 0 112	66080
ordr_0_113	910	65535	ORDR_0	All	/dev/vgdata2/rordr_0_113	66080
ordr 0 114	742	65535	ORDR 0	All	/dev/vgdata1/rordr 0 114	66080
ordr_0_115	806	65535	ORDR_0	All	/dev/vgdata2/rordr_0_115	66080
ordr_0_116	793	65535	ORDR_0	All	/dev/vgdata1/rordr_0_116	66080
ordr 0 117	795	65535	ORDR 0	All	/dev/vgdata2/rordr 0 117	66080
ordr_0_118	950	65535	ORDR_0	All	/dev/vgdata1/rordr_0_118	66080
ordr 0 119	902	65535	ORDR 0	All	/dev/vgdata2/rordr 0 119	66080
ordr 0 12	886	65535	ORDR 0	All	/dev/vgdata1/rordr 0 12	66080
ordr 0 120	865	65535	ORDR 0	All	/dev/vgdata1/rordr 0 120	66080
ordr 0 121	837	65535	ORDR 0	All	/dev/vgdata2/rordr 0 121	66080
ordr_0_122	939	65535	ORDR_0	All	/dev/vgdata1/rordr_0_122	66080
ordr 0 123	827	65535	ORDR 0	All	/dev/vgdata2/rordr 0 123	66080
ordr 0 124	903	65535	ORDR 0	All	/dev/vgdata1/rordr 0 124	66080
ordr_0_125	841	65535	ORDR_0	All	/dev/vgdata2/rordr_0_125	66080
ordr 0 126	818	65535	ORDR 0	All	/dev/vgdata1/rordr 0 126	66080
ordr_0_127	758	65535	ORDR_0	All	/dev/vgdata2/rordr_0_127	66080
ordr 0 128	803	65535	ORDR 0	All	/dev/vgdata1/rordr 0 128	66080
ordr 0 129	855	65535	ORDR 0	All	/dev/vgdata2/rordr 0 129	66080
ordr_0_13	859	65535	ORDR_0	All	/dev/vgdata2/rordr_0_13	66080
ordr 0 130	952	65535	ORDR 0	All	/dev/vgdata1/rordr 0 130	66080
ordr 0 131	791	65535	ORDR 0	All	/dev/vgdata2/rordr 0 131	66080
ordr_0_132	802	65535	ORDR_0	All	/dev/vgdata1/rordr_0_132	66080
ordr 0 133	909	65535	ORDR 0	All	/dev/vgdata2/rordr 0 133	66080
ordr 0 134	780	65535	ORDR 0	All	/dev/vgdata1/rordr 0 134	66080
ordr_0_135	773	65535	ORDR_0	All	/dev/vgdata2/rordr_0_135	66080
ordr 0 136	883	65535	ORDR 0	All	/dev/vgdata1/rordr 0 136	66080
ordr_0_137	809	65535	ORDR_0	All	/dev/vgdata2/rordr_0_137	66080
ordr_0_138	770	65535	ORDR_0	All	/dev/vgdata1/rordr_0_138	66080
ordr 0 139	804	65535	ORDR 0	All	/dev/vgdata2/rordr 0 139	66080
ordr_0_14	940	65535	ORDR_0	All	/dev/vgdata1/rordr_0_14	66080
ordr 0 140	766	65535	ORDR 0	All	/dev/vgdata1/rordr 0 140	66080
ordr 0 141	745	65535	ORDR 0	All	/dev/vgdata2/rordr 0 141	66080
ordr 0 142	850	65535	ORDR 0	All	/dev/vgdata1/rordr 0 142	66080
ordr 0 143	786	65535	ORDR 0	All	/dev/vgdata2/rordr 0 143	66080
ordr_0_144	829	65535	ORDR_0	All	/dev/vgdata1/rordr_0_144	66080
ordr 0 145	895	65535	ORDR 0	All	/dev/vgdata2/rordr 0 145	66080
ordr 0 146	913	65535	ORDR 0	All	/dev/vgdata1/rordr 0 146	66080
ordr_0_147	819	65535	ORDR_0	All	/dev/vgdata2/rordr_0_147	66080
ordr 0 148	755	65535	ORDR 0	All	/dev/vgdata1/rordr 0 148	66080
ordr_0_149	757	65535	ORDR_0	All	/dev/vgdata2/rordr_0_149	66080
ordr 0 15	873	65535	ORDR 0	All	/dev/vgdata2/rordr 0 15	66080
ordr 0 150	836	65535	ORDR 0	All	/dev/vgdata1/rordr 0 150	66080
ordr_0_151	787	65535	ORDR_0	All	/dev/vgdata2/rordr_0_151	66080
ordr 0 152	822	65535	ORDR 0	All	/dev/vgdata1/rordr 0 152	66080
ordr 0 153	882	65535	ORDR 0	All	/dev/vgdata2/rordr 0 153	66080
ordr_0_154	751	65535	ORDR_0	All	/dev/vgdata1/rordr_0_154	66080
ordr 0 155	840	65535	ORDR 0	All	/dev/vgdata2/rordr 0 155	66080

ordr 0 156	884	65535	ORDR 0	All	/dev/vgdata1/rordr 0 156	66080
ordr 0 157	756	65535	ORDR 0	All	/dev/vgdata2/rordr 0 157	66080
ordr_0_158	778	65535	ORDR_0	All	/dev/vgdata1/rordr_0_158	66080
ordr 0 159	951	65535	ORDR 0	All	/dev/vgdata2/rordr 0 159	66080
ordr 0 16	782	65535	ORDR 0	All	/dev/vgdata1/rordr 0 16	66080
ordr_0_160	838	65535	ORDR_0	All	/dev/vgdata1/rordr_0_160	66080
ordr 0 161	779	65535	ORDR 0	All	/dev/vgdata2/rordr 0 161	66080
ordr_0_162	749	65535	ORDR_0	All	/dev/vgdata1/rordr_0_162	66080
ordr_0_163	798	65535	ORDR_0	All	/dev/vgdata2/rordr_0_163	66080
ordr 0 164	897	65535	ORDR 0	All	/dev/vgdata1/rordr 0 164	66080
ordr_0_165	814	65535	ORDR_0	All	/dev/vgdata2/rordr_0_165	66080
ordr 0 166	828	65535	ORDR 0	All	/dev/vgdata1/rordr 0 166	66080
ordr 0 167	790	65535	ORDR 0	All	/dev/vgdata2/rordr 0 167	66080
ordr 0 168	831	65535	ORDR 0	All	/dev/vgdata1/rordr 0 168	66080
ordr 0 169	864	65535	ORDR 0	All	/dev/vgdata2/rordr 0 169	66080
ordr_0_17	900	65535	ORDR_0	All	/dev/vgdata2/rordr_0_17	66080
ordr 0 170	926	65535	ORDR 0	All	/dev/vgdata1/rordr 0 170	66080
ordr 0 171	851	65535	ORDR 0	All	/dev/vgdata2/rordr 0 171	66080
ordr_0_172	811	65535	ORDR_0	All	/dev/vgdata1/rordr_0_172	66080
ordr 0 173	942	65535	ORDR 0	All	/dev/vgdata2/rordr 0 173	66080
ordr_0_174	761	65535	ORDR_0	All	/dev/vgdata1/rordr_0_174	66080
ordr 0 175	759	65535	ORDR 0	All	/dev/vgdata2/rordr 0 175	66080
ordr 0 176	744	65535	ORDR 0	All	/dev/vgdata1/rordr 0 176	66080
ordr_0_177	785	65535	ORDR_0	All	/dev/vgdata2/rordr_0_177	66080
ordr 0 178	774	65535	ORDR 0	All	/dev/vgdata1/rordr 0 178	66080
ordr 0 179	878	65535	ORDR 0	All	/dev/vgdata2/rordr 0 179	66080
ordr_0_18	874	65535	ORDR_0	All	/dev/vgdata1/rordr_0_18	66080
ordr 0 19	907	65535	ORDR 0	All	/dev/vgdata2/rordr 0 19	66080
ordr 0 2	925	65535	ORDR 0	All	/dev/vgdata1/rordr 0 2	66080
ordr_0_20	784	65535	ORDR_0	All	/dev/vgdata1/rordr_0_20	66080
ordr 0 21	764	65535	ORDR 0	All	/dev/vgdata2/rordr 0 21	66080
ordr_0_22	825	65535	ORDR_0	All	/dev/vgdata1/rordr_0_22	66080
ordr_0_23	753	65535	ORDR_0	All	/dev/vgdata2/rordr_0_23	66080
ordr 0 24	754	65535	ORDR 0	All	/dev/vgdata1/rordr 0 24	66080
ordr_0_25	928	65535	ORDR_0	All	/dev/vgdata2/rordr_0_25	66080
ordr 0 26	870	65535	ORDR 0	All	/dev/vgdata1/rordr 0 26	66080
ordr 0 27	847	65535	ORDR 0	All	/dev/vgdata2/rordr 0 27	66080
ordr 0 28	834	65535	ORDR 0	All	/dev/vgdata1/rordr 0 28	66080
ordr 0 29	869	65535	ORDR 0	All	/dev/vgdata2/rordr 0 29	66080
ordr_0_3	832	65535	ORDR_0	All	/dev/vgdata2/rordr_0_3	66080
ordr 0 30	875	65535	ORDR 0	All	/dev/vgdata1/rordr 0 30	66080
ordr 0 31	797	65535	ORDR 0	All	/dev/vgdata2/rordr 0 31	66080
ordr_0_32	820	65535	ORDR_0	All	/dev/vgdata1/rordr_0_32	66080
ordr 0 33	905	65535	ORDR 0	All	/dev/vgdata2/rordr 0 33	66080
ordr_0_34	750	65535	ORDR_0	All	/dev/vgdata1/rordr_0_34	66080
ordr 0 35	872	65535	ORDR 0	All	/dev/vgdata2/rordr 0 35	66080
ordr 0 36	901	65535	ORDR 0	All	/dev/vgdata1/rordr 0 36	66080
ordr_0_37	927	65535	ORDR_0	All	/dev/vgdata2/rordr_0_37	66080
ordr 0 38	938	65535	ORDR 0	All	/dev/vgdata1/rordr 0 38	66080
ordr 0 39	769	65535	ORDR 0	All	/dev/vgdata2/rordr 0 39	66080
ordr_0_4	783	65535	ORDR_0	All	/dev/vgdata1/rordr_0_4	66080
ordr 0 40	885	65535	ORDR 0	All	/dev/vgdata1/rordr 0 40	66080

ordr 0 41	852	65535	ORDR 0	All	/dev/vgdata2/rordr 0 41	66080
ordr 0 42	768	65535	ORDR 0	All	/dev/vgdata1/rordr 0 42	66080
ordr_0_43	908	65535	ORDR_0	All	/dev/vgdata2/rordr_0_43	66080
ordr 0 44	813	65535	ORDR 0	All	/dev/vgdata1/rordr 0 44	66080
ordr 0 45	799	65535	ORDR 0	All	/dev/vgdata2/rordr 0 45	66080
ordr_0_46	817	65535	ORDR_0	All	/dev/vgdata1/rordr_0_46	66080
ordr 0 47	824	65535	ORDR 0	All	/dev/vgdata2/rordr 0 47	66080
ordr_0_48	815	65535	ORDR_0	All	/dev/vgdata1/rordr_0_48	66080
ordr_0_49	911	65535	ORDR_0	All	/dev/vgdata2/rordr_0_49	66080
ordr 0 5	896	65535	ORDR 0	All	/dev/vgdata2/rordr 0 5	66080
ordr_0_50	789	65535	ORDR_0	All	/dev/vgdata1/rordr_0_50	66080
ordr 0 51	775	65535	ORDR 0	All	/dev/vgdata2/rordr 0 51	66080
ordr 0 52	842	65535	ORDR 0	All	/dev/vgdata1/rordr 0 52	66080
ordr 0 53	808	65535	ORDR 0	All	/dev/vgdata2/rordr 0 53	66080
ordr 0 54	947	65535	ORDR 0	All	/dev/vgdata1/rordr 0 54	66080
ordr_0_55	794	65535	ORDR_0	All	/dev/vgdata2/rordr_0_55	66080
ordr 0 56	788	65535	ORDR 0	All	/dev/vgdata1/rordr 0 56	66080
ordr 0 57	854	65535	ORDR 0	All	/dev/vgdata2/rordr 0 57	66080
ordr_0_58	776	65535	ORDR_0	All	/dev/vgdata1/rordr_0_58	66080
ordr 0 59	889	65535	ORDR 0	All	/dev/vgdata2/rordr 0 59	66080
ordr_0_6	760	65535	ORDR_0	All	/dev/vgdata1/rordr_0_6	66080
ordr 0 60	801	65535	ORDR 0	All	/dev/vgdata1/rordr 0 60	66080
ordr 0 61	861	65535	ORDR 0	All	/dev/vgdata2/rordr 0 61	66080
ordr_0_62	867	65535	ORDR_0	All	/dev/vgdata1/rordr_0_62	66080
ordr 0 63	848	65535	ORDR 0	All	/dev/vgdata2/rordr 0 63	66080
ordr 0 64	846	65535	ORDR 0	All	/dev/vgdata1/rordr 0 64	66080
ordr_0_65	863	65535	ORDR_0	All	/dev/vgdata2/rordr_0_65	66080
ordr 0 66	835	65535	ORDR 0	All	/dev/vgdata1/rordr 0 66	66080
ordr 0 67	740	65535	ORDR 0	All	/dev/vgdata2/rordr 0 67	66080
ordr_0_68	871	65535	ORDR_0	All	/dev/vgdata1/rordr_0_68	66080
ordr 0 69	881	65535	ORDR 0	All	/dev/vgdata2/rordr 0 69	66080
ordr_0_7	906	65535	ORDR_0	All	/dev/vgdata2/rordr_0_7	66080
ordr_0_70	771	65535	ORDR_0	All	/dev/vgdata1/rordr_0_70	66080
ordr 0 71	887	65535	ORDR 0	All	/dev/vgdata2/rordr 0 71	66080
ordr_0_72	862	65535	ORDR_0	All	/dev/vgdata1/rordr_0_72	66080
ordr 0 73	912	65535	ORDR 0	All	/dev/vgdata2/rordr 0 73	66080
ordr 0 74	792	65535	ORDR 0	All	/dev/vgdata1/rordr 0 74	66080
ordr 0 75	948	65535	ORDR 0	All	/dev/vgdata2/rordr 0 75	66080
ordr 0 76	762	65535	ORDR 0	All	/dev/vgdata1/rordr 0 76	66080
ordr_0_77	898	65535	ORDR_0	All	/dev/vgdata2/rordr_0_77	66080
ordr 0 78	849	65535	ORDR 0	All	/dev/vgdata1/rordr 0 78	66080
ordr 0 79	747	65535	ORDR 0	All	/dev/vgdata2/rordr 0 79	66080
ordr_0_8	812	65535	ORDR_0	All	/dev/vgdata1/rordr_0_8	66080
ordr 0 80	765	65535	ORDR 0	All	/dev/vgdata1/rordr 0 80	66080
ordr_0_81	821	65535	ORDR_0	All	/dev/vgdata2/rordr_0_81	66080
ordr 0 82	949	65535	ORDR 0	All	/dev/vgdata1/rordr 0 82	66080
ordr 0 83	796	65535	ORDR 0	All	/dev/vgdata2/rordr 0 83	66080
ordr_0_84	879	65535	ORDR_0	All	/dev/vgdata1/rordr_0_84	66080
ordr 0 85	843	65535	ORDR 0	All	/dev/vgdata2/rordr 0 85	66080
ordr 0 86	853	65535	ORDR 0	All	/dev/vgdata1/rordr 0 86	66080
ordr_0_87	739	65535	ORDR_0	All	/dev/vgdata2/rordr_0_87	66080
ordr 0 88	844	65535	ORDR 0	All	/dev/vgdata1/rordr 0 88	66080

ordr 0 89	941	65535	ORDR 0	All	/dev/vgdata2/rordr 0 89	66080
ordr 0 9	904	65535	ORDR 0	All	/dev/vgdata2/rordr 0 9	66080
ordr_0_90	876	65535	ORDR_0	All	/dev/vgdata1/rordr_0_90	66080
ordr 0 91	807	65535	ORDR 0	All	/dev/vgdata2/rordr 0 91	66080
ordr 0 92	845	65535	ORDR 0	All	/dev/vgdata1/rordr 0 92	66080
ordr_0_93	800	65535	ORDR_0	All	/dev/vgdata2/rordr_0_93	66080
ordr 0 94	772	65535	ORDR 0	All	/dev/vgdata1/rordr 0 94	66080
ordr_0_95	888	65535	ORDR_0	All	/dev/vgdata2/rordr_0_95	66080
ordr_0_96	810	65535	ORDR_0	All	/dev/vgdata1/rordr_0_96	66080
ordr 0 97	860	65535	ORDR 0	All	/dev/vgdata2/rordr 0 97	66080
ordr_0_98	839	65535	ORDR_0	All	/dev/vgdata1/rordr_0_98	66080
ordr 0 99	746	65535	ORDR 0	All	/dev/vgdata2/rordr 0 99	66080
roll1	3	8096	UNDO 1	All	/dev/vgdata1/rroll1	9440
sp 0	1615	2048	SP 0	All	/dev/vgdata1/rsp 0	3776
stok 0 0	954	15800	STOK 0	C.Q4.15	/dev/rdisk/disk6673	16386
stok_0_1	983	15800	STOK_0	B.Q3.12	/dev/rdisk/disk4774	16386
stok 0 10	1182	15800	STOK 0	A.Q5.18	/dev/rdisk/disk4485	16386
stok 0 100	1097	15800	STOK 0	B.Q10.40	/dev/rdisk/disk4361	16386
stok_0_101	987	15800	STOK_0	B.Q13.50	/dev/rdisk/disk4606	16386
stok 0 102	1036	15800	STOK 0	A.Q13.51	/dev/rdisk/disk4326	16386
stok_0_103	955	15800	STOK_0	B.Q5.20	/dev/rdisk/disk5113	16386
stok 0 104	1202	15800	STOK 0	B.Q1.1	/dev/rdisk/disk5511	16386
stok 0 105	1109	15800	STOK 0	C.Q2.7	/dev/rdisk/disk6784	16386
stok_0_106	1013	15800	STOK_0	C.Q1.3	/dev/rdisk/disk6874	16386
stok 0 107	1026	15800	STOK 0	C.Q1.4	/dev/rdisk/disk6495	16386
stok 0 108	988	15800	STOK 0	C.Q6.21	/dev/rdisk/disk6414	16386
stok_0_109	1007	15800	STOK_0	C.Q5.17	/dev/rdisk/disk6827	16386
stok 0 11	1095	15800	STOK 0	A.Q5.20	/dev/rdisk/disk4550	16386
stok 0 110	989	15800	STOK 0	B.Q4.14	/dev/rdisk/disk5638	16386
stok_0_111	995	15800	STOK_0	A.Q1.4	/dev/rdisk/disk6943	16386
stok 0 112	998	15800	STOK 0	A.Q3.9	/dev/rdisk/disk3906	16386
stok_0_113	1067	15800	STOK_0	A.Q2.8	/dev/rdisk/disk3672	16386
stok_0_114	1047	15800	STOK_0	A.Q2.7	/dev/rdisk/disk3635	16386
stok 0 115	1088	15800	STOK 0	B.Q1.3	/dev/rdisk/disk5517	16386
stok_0_116	1133	15800	STOK_0	B.Q5.18	/dev/rdisk/disk6143	16386
stok 0 117	1203	15800	STOK 0	B.Q9.34	/dev/rdisk/disk5490	16386
stok 0 118	1084	15800	STOK 0	C.Q4.15	/dev/rdisk/disk6829	16386
stok 0 119	1178	15800	STOK 0	B.Q3.12	/dev/rdisk/disk4775	16386
stok 0 12	984	15800	STOK 0	A.Q4.14	/dev/rdisk/disk4079	16386
stok_0_120	1124	15800	STOK_0	A.Q7.26	/dev/rdisk/disk5731	16386
stok 0 121	1054	15800	STOK 0	A.Q3.10	/dev/rdisk/disk3836	16386
stok 0 122	1050	15800	STOK 0	A.Q6.21	/dev/rdisk/disk4566	16386
stok_0_123	1048	15800	STOK_0	C.Q3.11	/dev/rdisk/disk6301	16386
stok 0 124	1172	15800	STOK 0	A.Q7.25	/dev/rdisk/disk6054	16386
stok_0_125	958	15800	STOK_0	A.Q4.16	/dev/rdisk/disk4056	16386
stok 0 126	1144	15800	STOK 0	B.Q2.6	/dev/rdisk/disk4180	16386
stok 0 127	1104	15800	STOK 0	A.Q3.11	/dev/rdisk/disk3753	16386
stok_0_128	1154	15800	STOK_0	A.Q5.18	/dev/rdisk/disk4483	16386
stok 0 129	976	15800	STOK 0	A.Q5.20	/dev/rdisk/disk4552	16386
stok 0 13	1169	15800	STOK 0	A.Q1.3	/dev/rdisk/disk5072	16386
stok_0_130	1184	15800	STOK_0	A.Q4.14	/dev/rdisk/disk4072	16386
stok 0 131	1002	15800	STOK 0	A.Q1.3	/dev/rdisk/disk4474	16386

stok_0_132	1196	15800	STOK_0	A.Q10.39	/dev/rdisk/disk5015	16386
stok_0_133	1096	15800	STOK_0	B.Q8.31	/dev/rdisk/disk5416	16386
stok_0_134	1034	15800	STOK_0	B.Q10.38	/dev/rdisk/disk4810	16386
stok_0_135	996	15800	STOK_0	B.Q2.5	/dev/rdisk/disk5959	16386
stok_0_136	1173	15800	STOK_0	B.Q6.21	/dev/rdisk/disk3845	16386
stok_0_137	986	15800	STOK_0	A.Q11.42	/dev/rdisk/disk6212	16386
stok_0_138	1022	15800	STOK_0	A.Q10.40	/dev/rdisk/disk5554	16386
stok_0_139	1105	15800	STOK_0	B.Q1.2	/dev/rdisk/disk4932	16386
stok_0_14	991	15800	STOK_0	A.Q10.39	/dev/rdisk/disk4793	16386
stok_0_140	1155	15800	STOK_0	C.Q2.5	/dev/rdisk/disk6517	16386
stok_0_141	1063	15800	STOK_0	A.Q8.30	/dev/rdisk/disk5689	16386
stok_0_142	1049	15800	STOK_0	A.Q3.12	/dev/rdisk/disk3997	16386
stok_0_143	1094	15800	STOK_0	B.Q6.24	/dev/rdisk/disk4297	16386
stok_0_144	959	15800	STOK_0	B.Q2.7	/dev/rdisk/disk5601	16386
stok_0_145	1059	15800	STOK_0	C.Q6.23	/dev/rdisk/disk6575	16386
stok_0_146	1068	15800	STOK_0	C.Q4.13	/dev/rdisk/disk6752	16386
stok_0_147	1042	15800	STOK_0	A.Q9.36	/dev/rdisk/disk5489	16386
stok_0_148	1179	15800	STOK_0	B.Q6.22	/dev/rdisk/disk3737	16386
stok_0_149	1100	15800	STOK_0	A.Q6.23	/dev/rdisk/disk5841	16386
stok_0_15	1046	15800	STOK_0	B.Q8.31	/dev/rdisk/disk5418	16386
stok_0_150	982	15800	STOK_0	B.Q4.16	/dev/rdisk/disk4559	16386
stok_0_151	1134	15800	STOK_0	A.Q7.28	/dev/rdisk/disk5976	16386
stok_0_152	1080	15800	STOK_0	A.Q6.22	/dev/rdisk/disk6051	16386
stok_0_153	1089	15800	STOK_0	A.Q8.32	/dev/rdisk/disk5100	16386
stok_0_154	1044	15800	STOK_0	B.Q8.29	/dev/rdisk/disk3939	16386
stok_0_155	1008	15800	STOK_0	C.Q2.6	/dev/rdisk/disk6539	16386
stok_0_156	1140	15800	STOK_0	A.Q7.27	/dev/rdisk/disk5878	16386
stok_0_157	1003	15800	STOK_0	C.Q1.2	/dev/rdisk/disk6415	16386
stok_0_158	1131	15800	STOK_0	C.Q5.19	/dev/rdisk/disk6646	16386
stok_0_159	1016	15800	STOK_0	B.Q7.26	/dev/rdisk/disk4238	16386
stok_0_16	1160	15800	STOK_0	B.Q10.38	/dev/rdisk/disk4807	16386
stok_0_160	1166	15800	STOK_0	C.Q5.20	/dev/rdisk/disk6737	16386
stok_0_161	1001	15800	STOK_0	B.Q8.30	/dev/rdisk/disk3711	16386
stok_0_162	1043	15800	STOK_0	B.Q7.25	/dev/rdisk/disk5789	16386
stok_0_163	1205	15800	STOK_0	B.Q12.45	/dev/rdisk/disk5232	16386
stok_0_164	1090	15800	STOK_0	B.Q5.17	/dev/rdisk/disk4437	16386
stok_0_165	1204	15800	STOK_0	B.Q3.11	/dev/rdisk/disk4648	16386
stok_0_166	1025	15800	STOK_0	A.Q9.35	/dev/rdisk/disk6107	16386
stok_0_167	1164	15800	STOK_0	C.Q4.14	/dev/rdisk/disk6818	16386
stok_0_168	1123	15800	STOK_0	A.Q5.17	/dev/rdisk/disk4914	16386
stok_0_169	961	15800	STOK_0	A.Q4.15	/dev/rdisk/disk3948	16386
stok_0_17	1135	15800	STOK_0	B.Q2.5	/dev/rdisk/disk5960	16386
stok_0_170	1121	15800	STOK_0	A.Q4.13	/dev/rdisk/disk4076	16386
stok_0_171	1185	15800	STOK_0	A.Q1.1	/dev/rdisk/disk5288	16386
stok_0_172	973	15800	STOK_0	A.Q8.29	/dev/rdisk/disk5152	16386
stok_0_173	1069	15800	STOK_0	C.Q5.18	/dev/rdisk/disk6817	16386
stok_0_174	1041	15800	STOK_0	C.Q2.8	/dev/rdisk/disk6541	16386
stok_0_175	1106	15800	STOK_0	B.Q10.37	/dev/rdisk/disk4846	16386
stok_0_176	956	15800	STOK_0	A.Q13.52	/dev/rdisk/disk5272	16386
stok_0_177	1110	15800	STOK_0	A.Q13.50	/dev/rdisk/disk5106	16386
stok_0_178	1145	15800	STOK_0	B.Q5.19	/dev/rdisk/disk5169	16386
stok_0_179	1027	15800	STOK_0	C.Q6.22	/dev/rdisk/disk6397	16386

stok 0 18	1009	15800	STOK 0	B.Q6.21	/dev/rdisk/disk3846	16386
stok 0 180	990	15800	STOK 0	B.Q2.8	/dev/rdisk/disk4815	16386
stok_0_181	1180	15800	STOK_0	B.Q3.10	/dev/rdisk/disk4827	16386
stok 0 182	1116	15800	STOK 0	B.Q9.33	/dev/rdisk/disk4506	16386
stok 0 183	1085	15800	STOK 0	A.Q6.24	/dev/rdisk/disk3758	16386
stok_0_184	1181	15800	STOK_0	A.Q12.48	/dev/rdisk/disk5790	16386
stok 0 185	1070	15800	STOK 0	B.Q4.13	/dev/rdisk/disk4525	16386
stok_0_186	1037	15800	STOK_0	B.Q12.46	/dev/rdisk/disk6077	16386
stok_0_187	977	15800	STOK_0	A.Q9.34	/dev/rdisk/disk6047	16386
stok 0 188	1064	15800	STOK 0	A.Q10.38	/dev/rdisk/disk5025	16386
stok_0_189	1207	15800	STOK_0	C.Q3.9	/dev/rdisk/disk6315	16386
stok 0 19	975	15800	STOK 0	A.Q11.42	/dev/rdisk/disk6211	16386
stok 0 190	1146	15800	STOK 0	A.Q11.43	/dev/rdisk/disk6186	16386
stok 0 191	1174	15800	STOK 0	C.Q3.10	/dev/rdisk/disk6954	16386
stok 0 192	1052	15800	STOK 0	C.Q1.1	/dev/rdisk/disk6565	16386
stok_0_193	1161	15800	STOK_0	A.Q12.47	/dev/rdisk/disk4230	16386
stok 0 194	960	15800	STOK 0	C.Q4.16	/dev/rdisk/disk6846	16386
stok 0 195	1187	15800	STOK 0	B.Q4.15	/dev/rdisk/disk5046	16386
stok_0_196	1098	15800	STOK_0	B.Q10.39	/dev/rdisk/disk5993	16386
stok 0 197	1101	15800	STOK 0	B.Q12.47	/dev/rdisk/disk4321	16386
stok_0_198	966	15800	STOK_0	A.Q11.44	/dev/rdisk/disk6232	16386
stok 0 199	1138	15800	STOK 0	A.Q2.5	/dev/rdisk/disk3852	16386
stok 0 2	1200	15800	STOK 0	A.Q7.26	/dev/rdisk/disk5729	16386
stok_0_20	1073	15800	STOK_0	A.Q10.40	/dev/rdisk/disk6013	16386
stok 0 200	971	15800	STOK 0	B.Q8.32	/dev/rdisk/disk4154	16386
stok 0 201	1206	15800	STOK 0	C.Q3.12	/dev/rdisk/disk6303	16386
stok_0_202	1186	15800	STOK_0	B.Q12.48	/dev/rdisk/disk4411	16386
stok 0 203	962	15800	STOK 0	B.Q7.27	/dev/rdisk/disk5352	16386
stok 0 204	1035	15800	STOK 0	B.Q1.4	/dev/rdisk/disk5586	16386
stok_0_205	974	15800	STOK_0	B.Q13.51	/dev/rdisk/disk4233	16386
stok 0 206	1117	15800	STOK 0	B.Q13.49	/dev/rdisk/disk5816	16386
stok_0_207	1086	15800	STOK_0	C.Q6.24	/dev/rdisk/disk6480	16386
stok_0_208	1113	15800	STOK_0	A.Q8.31	/dev/rdisk/disk4973	16386
stok 0 209	1102	15800	STOK 0	A.Q1.2	/dev/rdisk/disk4602	16386
stok_0_21	1150	15800	STOK_0	B.Q1.2	/dev/rdisk/disk4930	16386
stok 0 210	1188	15800	STOK 0	B.Q6.23	/dev/rdisk/disk3982	16386
stok 0 211	1156	15800	STOK 0	B.Q7.28	/dev/rdisk/disk5738	16386
stok 0 212	967	15800	STOK 0	A.Q5.19	/dev/rdisk/disk4696	16386
stok 0 213	1136	15800	STOK 0	B.Q3.9	/dev/rdisk/disk5913	16386
stok_0_214	997	15800	STOK_0	B.Q13.52	/dev/rdisk/disk6905	16386
stok 0 215	1000	15800	STOK 0	A.Q12.46	/dev/rdisk/disk4371	16386
stok 0 216	1141	15800	STOK 0	A.Q2.6	/dev/rdisk/disk3629	16386
stok_0_217	1176	15800	STOK_0	A.Q9.33	/dev/rdisk/disk4460	16386
stok 0 218	1142	15800	STOK 0	B.Q10.40	/dev/rdisk/disk5335	16386
stok_0_219	1147	15800	STOK_0	B.Q13.50	/dev/rdisk/disk4629	16386
stok 0 22	1011	15800	STOK 0	C.Q2.5	/dev/rdisk/disk6348	16386
stok 0 220	1093	15800	STOK 0	A.Q13.51	/dev/rdisk/disk4327	16386
stok_0_221	1071	15800	STOK_0	B.Q5.20	/dev/rdisk/disk5121	16386
stok 0 222	981	15800	STOK 0	B.Q1.1	/dev/rdisk/disk5496	16386
stok 0 223	1107	15800	STOK 0	C.Q2.7	/dev/rdisk/disk6778	16386
stok_0_224	1208	15800	STOK_0	C.Q1.3	/dev/rdisk/disk6873	16386
stok 0 225	1137	15800	STOK 0	C.Q1.4	/dev/rdisk/disk6504	16386

stok_0_226	1148	15800	STOK_0	C.Q6.21	/dev/rdisk/disk6390	16386
stok_0_227	1165	15800	STOK_0	C.Q5.17	/dev/rdisk/disk6706	16386
stok_0_228	1118	15800	STOK_0	A.Q1.4	/dev/rdisk/disk6921	16386
stok_0_229	999	15800	STOK_0	B.Q4.14	/dev/rdisk/disk5709	16386
stok_0_23	1024	15800	STOK_0	A.Q8.30	/dev/rdisk/disk5680	16386
stok_0_230	1120	15800	STOK_0	A.Q3.9	/dev/rdisk/disk3903	16386
stok_0_231	1087	15800	STOK_0	A.Q2.8	/dev/rdisk/disk3668	16386
stok_0_232	1056	15800	STOK_0	A.Q2.7	/dev/rdisk/disk3634	16386
stok_0_233	1130	15800	STOK_0	B.Q1.3	/dev/rdisk/disk5512	16386
stok_0_234	1023	15800	STOK_0	B.Q5.18	/dev/rdisk/disk6057	16386
stok_0_235	1210	15800	STOK_0	B.Q9.34	/dev/rdisk/disk5493	16386
stok_0_236	963	15800	STOK_0	C.Q4.15	/dev/rdisk/disk6844	16386
stok_0_237	1157	15800	STOK_0	B.Q3.12	/dev/rdisk/disk4758	16386
stok_0_238	1030	15800	STOK_0	A.Q7.26	/dev/rdisk/disk5733	16386
stok_0_239	1175	15800	STOK_0	A.Q3.10	/dev/rdisk/disk3830	16386
stok_0_24	985	15800	STOK_0	A.Q3.12	/dev/rdisk/disk3998	16386
stok_0_240	1198	15800	STOK_0	A.Q6.21	/dev/rdisk/disk4574	16386
stok_0_241	1114	15800	STOK_0	C.Q3.11	/dev/rdisk/disk6298	16386
stok_0_242	1197	15800	STOK_0	A.Q7.25	/dev/rdisk/disk6052	16386
stok_0_243	1031	15800	STOK_0	A.Q4.16	/dev/rdisk/disk4052	16386
stok_0_244	1167	15800	STOK_0	B.Q2.6	/dev/rdisk/disk5859	16386
stok_0_245	968	15800	STOK_0	A.Q3.11	/dev/rdisk/disk3742	16386
stok_0_246	1149	15800	STOK_0	A.Q5.18	/dev/rdisk/disk4496	16386
stok_0_247	1108	15800	STOK_0	A.Q5.20	/dev/rdisk/disk4573	16386
stok_0_248	978	15800	STOK_0	A.Q4.14	/dev/rdisk/disk4080	16386
stok_0_249	1209	15800	STOK_0	A.Q1.3	/dev/rdisk/disk4467	16386
stok_0_25	1158	15800	STOK_0	B.Q6.24	/dev/rdisk/disk4304	16386
stok_0_250	1189	15800	STOK_0	A.Q10.39	/dev/rdisk/disk5000	16386
stok_0_251	1162	15800	STOK_0	B.Q8.31	/dev/rdisk/disk5410	16386
stok_0_252	1199	15800	STOK_0	B.Q10.38	/dev/rdisk/disk4806	16386
stok_0_253	1060	15800	STOK_0	B.Q2.5	/dev/rdisk/disk5958	16386
stok_0_254	1119	15800	STOK_0	B.Q6.21	/dev/rdisk/disk3794	16386
stok_0_255	1017	15800	STOK_0	A.Q11.42	/dev/rdisk/disk6208	16386
stok_0_256	1168	15800	STOK_0	A.Q10.40	/dev/rdisk/disk5927	16386
stok_0_257	1334	15800	STOK_0	B.Q1.2	/dev/rdisk/disk4931	16386
stok_0_258	1226	15800	STOK_0	C.Q2.5	/dev/rdisk/disk6515	16386
stok_0_259	1216	15800	STOK_0	A.Q8.30	/dev/rdisk/disk4282	16386
stok_0_26	1029	15800	STOK_0	B.Q2.7	/dev/rdisk/disk4124	16386
stok_0_260	1225	15800	STOK_0	A.Q3.12	/dev/rdisk/disk3996	16386
stok_0_261	1281	15800	STOK_0	B.Q6.24	/dev/rdisk/disk4298	16386
stok_0_262	1460	15800	STOK_0	B.Q2.7	/dev/rdisk/disk4122	16386
stok_0_263	1364	15800	STOK_0	C.Q6.23	/dev/rdisk/disk6586	16386
stok_0_264	1261	15800	STOK_0	C.Q4.13	/dev/rdisk/disk6764	16386
stok_0_265	1375	15800	STOK_0	A.Q9.36	/dev/rdisk/disk5485	16386
stok_0_266	1247	15800	STOK_0	B.Q6.22	/dev/rdisk/disk3741	16386
stok_0_267	1459	15800	STOK_0	A.Q6.23	/dev/rdisk/disk4259	16386
stok_0_268	1354	15800	STOK_0	B.Q4.16	/dev/rdisk/disk4535	16386
stok_0_269	1425	15800	STOK_0	A.Q7.28	/dev/rdisk/disk5974	16386
stok_0_27	992	15800	STOK_0	C.Q6.23	/dev/rdisk/disk6572	16386
stok_0_270	1430	15800	STOK_0	A.Q6.22	/dev/rdisk/disk6049	16386
stok_0_271	1369	15800	STOK_0	A.Q8.32	/dev/rdisk/disk5211	16386
stok_0_272	1231	15800	STOK_0	B.Q8.29	/dev/rdisk/disk3962	16386

stok 0 273	1307	15800	STOK 0	C.Q2.6	/dev/rdisk/disk6542	16386
stok 0 274	1431	15800	STOK 0	A.Q7.27	/dev/rdisk/disk5876	16386
stok_0_275	1365	15800	STOK_0	C.Q1.2	/dev/rdisk/disk6396	16386
stok 0 276	1243	15800	STOK 0	C.Q5.19	/dev/rdisk/disk6794	16386
stok 0 277	1336	15800	STOK 0	B.Q7.26	/dev/rdisk/disk5479	16386
stok_0_278	1253	15800	STOK_0	C.Q5.20	/dev/rdisk/disk6733	16386
stok 0 279	1461	15800	STOK 0	B.Q8.30	/dev/rdisk/disk3715	16386
stok_0_28	1170	15800	STOK_0	C.Q4.13	/dev/rdisk/disk6636	16386
stok_0_280	1343	15800	STOK_0	B.Q7.25	/dev/rdisk/disk5779	16386
stok 0 281	1409	15800	STOK 0	B.Q12.45	/dev/rdisk/disk4335	16386
stok_0_282	1221	15800	STOK_0	B.Q5.17	/dev/rdisk/disk4984	16386
stok 0 283	1393	15800	STOK 0	B.Q3.11	/dev/rdisk/disk4656	16386
stok 0 284	1323	15800	STOK 0	A.Q9.35	/dev/rdisk/disk6104	16386
stok 0 285	1232	15800	STOK 0	C.Q4.14	/dev/rdisk/disk6687	16386
stok 0 286	1451	15800	STOK 0	A.Q5.17	/dev/rdisk/disk4903	16386
stok_0_287	1300	15800	STOK_0	A.Q4.15	/dev/rdisk/disk3944	16386
stok 0 288	1319	15800	STOK 0	A.Q4.13	/dev/rdisk/disk4069	16386
stok 0 289	1344	15800	STOK 0	A.Q1.1	/dev/rdisk/disk5289	16386
stok_0_29	1065	15800	STOK_0	A.Q9.36	/dev/rdisk/disk5453	16386
stok 0 290	1419	15800	STOK 0	A.Q8.29	/dev/rdisk/disk5161	16386
stok_0_291	1282	15800	STOK_0	C.Q5.18	/dev/rdisk/disk6797	16386
stok 0 292	1384	15800	STOK 0	C.Q2.8	/dev/rdisk/disk6544	16386
stok 0 293	1338	15800	STOK 0	B.Q10.37	/dev/rdisk/disk5250	16386
stok_0_294	1436	15800	STOK_0	A.Q13.52	/dev/rdisk/disk5264	16386
stok 0 295	1394	15800	STOK 0	A.Q13.50	/dev/rdisk/disk5103	16386
stok 0 296	1327	15800	STOK 0	B.Q5.19	/dev/rdisk/disk5163	16386
stok_0_297	1237	15800	STOK_0	C.Q6.22	/dev/rdisk/disk6349	16386
stok 0 298	1229	15800	STOK 0	B.Q2.8	/dev/rdisk/disk5180	16386
stok 0 299	1366	15800	STOK 0	B.Q3.10	/dev/rdisk/disk4963	16386
stok_0_3	1045	15800	STOK_0	A.Q3.10	/dev/rdisk/disk3840	16386
stok 0 30	1183	15800	STOK 0	B.Q6.22	/dev/rdisk/disk3733	16386
stok_0_300	1285	15800	STOK_0	B.Q9.33	/dev/rdisk/disk5123	16386
stok_0_301	1355	15800	STOK_0	A.Q6.24	/dev/rdisk/disk3754	16386
stok 0 302	1424	15800	STOK 0	A.Q12.48	/dev/rdisk/disk5401	16386
stok_0_303	1443	15800	STOK_0	B.Q4.13	/dev/rdisk/disk4529	16386
stok 0 304	1356	15800	STOK 0	B.Q12.46	/dev/rdisk/disk6061	16386
stok 0 305	1426	15800	STOK 0	A.Q9.34	/dev/rdisk/disk6038	16386
stok 0 306	1440	15800	STOK 0	A.Q10.38	/dev/rdisk/disk5007	16386
stok 0 307	1328	15800	STOK 0	C.Q3.9	/dev/rdisk/disk6313	16386
stok_0_308	1217	15800	STOK_0	A.Q11.43	/dev/rdisk/disk6184	16386
stok 0 309	1381	15800	STOK 0	C.Q3.10	/dev/rdisk/disk6948	16386
stok 0 31	1074	15800	STOK 0	A.Q6.23	/dev/rdisk/disk4240	16386
stok_0_310	1211	15800	STOK_0	C.Q1.1	/dev/rdisk/disk6561	16386
stok 0 311	1345	15800	STOK 0	A.Q12.47	/dev/rdisk/disk4228	16386
stok_0_312	1324	15800	STOK_0	C.Q4.16	/dev/rdisk/disk6847	16386
stok 0 313	1214	15800	STOK 0	B.Q4.15	/dev/rdisk/disk4742	16386
stok 0 314	1305	15800	STOK 0	B.Q10.39	/dev/rdisk/disk5994	16386
stok_0_315	1400	15800	STOK_0	B.Q12.47	/dev/rdisk/disk4318	16386
stok 0 316	1346	15800	STOK 0	A.Q11.44	/dev/rdisk/disk6235	16386
stok 0 317	1246	15800	STOK 0	A.Q2.5	/dev/rdisk/disk3848	16386
stok_0_318	1452	15800	STOK_0	B.Q8.32	/dev/rdisk/disk4160	16386
stok 0 319	1395	15800	STOK 0	C.Q3.12	/dev/rdisk/disk6300	16386

stok 0 32	1038	15800	STOK 0	B.Q4.16	/dev/rdisk/disk4583	16386
stok 0 320	1382	15800	STOK 0	B.Q12.48	/dev/rdisk/disk4415	16386
stok_0_321	1294	15800	STOK_0	B.Q7.27	/dev/rdisk/disk5803	16386
stok 0 322	1433	15800	STOK 0	B.Q1.4	/dev/rdisk/disk5592	16386
stok 0 323	1376	15800	STOK 0	B.Q13.51	/dev/rdisk/disk4252	16386
stok_0_324	1276	15800	STOK_0	B.Q13.49	/dev/rdisk/disk5812	16386
stok 0 325	1298	15800	STOK 0	C.Q6.24	/dev/rdisk/disk6425	16386
stok_0_326	1263	15800	STOK_0	A.Q8.31	/dev/rdisk/disk5650	16386
stok_0_327	1288	15800	STOK_0	A.Q1.2	/dev/rdisk/disk4595	16386
stok 0 328	1337	15800	STOK 0	B.Q6.23	/dev/rdisk/disk3990	16386
stok_0_329	1249	15800	STOK_0	B.Q7.28	/dev/rdisk/disk5745	16386
stok 0 33	1040	15800	STOK 0	A.Q7.28	/dev/rdisk/disk5977	16386
stok 0 330	1453	15800	STOK 0	A.Q5.19	/dev/rdisk/disk4649	16386
stok 0 331	1383	15800	STOK 0	B.Q3.9	/dev/rdisk/disk5910	16386
stok 0 332	1286	15800	STOK 0	B.Q13.52	/dev/rdisk/disk6908	16386
stok_0_333	1385	15800	STOK_0	A.Q12.46	/dev/rdisk/disk4367	16386
stok 0 334	1401	15800	STOK 0	A.Q2.6	/dev/rdisk/disk3622	16386
stok 0 335	1377	15800	STOK 0	A.Q9.33	/dev/rdisk/disk5533	16386
stok_0_336	1306	15800	STOK_0	B.Q10.40	/dev/rdisk/disk5337	16386
stok 0 337	1390	15800	STOK 0	B.Q13.50	/dev/rdisk/disk4624	16386
stok_0_338	1260	15800	STOK_0	A.Q13.51	/dev/rdisk/disk4329	16386
stok 0 339	1432	15800	STOK 0	B.Q5.20	/dev/rdisk/disk5117	16386
stok 0 34	964	15800	STOK 0	A.Q6.22	/dev/rdisk/disk6053	16386
stok_0_340	1367	15800	STOK_0	B.Q1.1	/dev/rdisk/disk5499	16386
stok 0 341	1410	15800	STOK 0	C.Q2.7	/dev/rdisk/disk6781	16386
stok 0 342	1445	15800	STOK 0	C.Q1.3	/dev/rdisk/disk6872	16386
stok_0_343	1370	15800	STOK_0	C.Q1.4	/dev/rdisk/disk6492	16386
stok 0 344	1325	15800	STOK 0	C.Q6.21	/dev/rdisk/disk6377	16386
stok 0 345	1326	15800	STOK 0	C.Q5.17	/dev/rdisk/disk6712	16386
stok_0_346	1434	15800	STOK_0	B.Q4.14	/dev/rdisk/disk5353	16386
stok 0 347	1303	15800	STOK 0	A.Q1.4	/dev/rdisk/disk6922	16386
stok_0_348	1277	15800	STOK_0	A.Q3.9	/dev/rdisk/disk3902	16386
stok_0_349	1357	15800	STOK_0	A.Q2.8	/dev/rdisk/disk3669	16386
stok 0 35	970	15800	STOK 0	A.Q8.32	/dev/rdisk/disk5105	16386
stok_0_350	1254	15800	STOK_0	A.Q2.7	/dev/rdisk/disk3631	16386
stok 0 351	1435	15800	STOK 0	B.Q1.3	/dev/rdisk/disk5503	16386
stok 0 352	1347	15800	STOK 0	B.Q5.18	/dev/rdisk/disk6131	16386
stok 0 353	1340	15800	STOK 0	B.Q9.34	/dev/rdisk/disk5487	16386
stok 0 354	1454	15800	STOK 0	C.Q4.15	/dev/rdisk/disk6836	16386
stok_0_355	1269	15800	STOK_0	B.Q3.12	/dev/rdisk/disk4741	16386
stok 0 356	1386	15800	STOK 0	A.Q7.26	/dev/rdisk/disk5673	16386
stok 0 357	1311	15800	STOK 0	A.Q3.10	/dev/rdisk/disk3829	16386
stok_0_358	1465	15800	STOK_0	A.Q6.21	/dev/rdisk/disk4736	16386
stok 0 359	1273	15800	STOK 0	C.Q3.11	/dev/rdisk/disk6293	16386
stok_0_36	1122	15800	STOK_0	B.Q8.29	/dev/rdisk/disk3941	16386
stok 0 360	1262	15800	STOK 0	A.Q7.25	/dev/rdisk/disk6043	16386
stok 0 361	1275	15800	STOK 0	A.Q4.16	/dev/rdisk/disk4054	16386
stok_0_362	1287	15800	STOK_0	B.Q2.6	/dev/rdisk/disk5865	16386
stok 0 363	1455	15800	STOK 0	A.Q3.11	/dev/rdisk/disk3780	16386
stok 0 364	1378	15800	STOK 0	A.Q5.18	/dev/rdisk/disk4501	16386
stok_0_365	1250	15800	STOK_0	A.Q5.20	/dev/rdisk/disk4538	16386
stok 0 366	1456	15800	STOK 0	A.Q4.14	/dev/rdisk/disk4070	16386

stok 0 367	1233	15800	STOK 0	A.Q1.3	/dev/rdisk/disk4470	16386
stok 0 368	1295	15800	STOK 0	A.Q10.39	/dev/rdisk/disk5002	16386
stok_0_369	1270	15800	STOK_0	B.Q8.31	/dev/rdisk/disk5452	16386
stok 0 37	1091	15800	STOK 0	C.Q2.6	/dev/rdisk/disk6534	16386
stok 0 370	1289	15800	STOK 0	B.Q10.38	/dev/rdisk/disk4803	16386
stok_0_371	1245	15800	STOK_0	B.Q2.5	/dev/rdisk/disk5957	16386
stok 0 372	1321	15800	STOK 0	B.Q6.21	/dev/rdisk/disk3844	16386
stok_0_373	1244	15800	STOK_0	A.Q11.42	/dev/rdisk/disk6209	16386
stok_0_374	1427	15800	STOK_0	A.Q10.40	/dev/rdisk/disk5917	16386
stok 0 375	1447	15800	STOK 0	B.Q1.2	/dev/rdisk/disk4929	16386
stok_0_376	1358	15800	STOK_0	C.Q2.5	/dev/rdisk/disk6628	16386
stok 0 377	1255	15800	STOK 0	A.Q8.30	/dev/rdisk/disk4284	16386
stok 0 378	1444	15800	STOK 0	A.Q3.12	/dev/rdisk/disk3995	16386
stok 0 379	1371	15800	STOK 0	B.Q6.24	/dev/rdisk/disk5821	16386
stok 0 38	1032	15800	STOK 0	A.Q7.27	/dev/rdisk/disk4183	16386
stok_0_380	1331	15800	STOK_0	B.Q2.7	/dev/rdisk/disk5598	16386
stok 0 381	1312	15800	STOK 0	C.Q6.23	/dev/rdisk/disk6578	16386
stok 0 382	1360	15800	STOK 0	C.Q4.13	/dev/rdisk/disk6759	16386
stok_0_383	1266	15800	STOK_0	A.Q9.36	/dev/rdisk/disk5538	16386
stok 0 384	1308	15800	STOK 0	B.Q6.22	/dev/rdisk/disk3738	16386
stok_0_385	1218	15800	STOK_0	A.Q6.23	/dev/rdisk/disk5414	16386
stok 0 386	1421	15800	STOK 0	B.Q4.16	/dev/rdisk/disk4534	16386
stok 0 387	1405	15800	STOK 0	A.Q7.28	/dev/rdisk/disk5978	16386
stok_0_388	1267	15800	STOK_0	A.Q6.22	/dev/rdisk/disk6041	16386
stok 0 389	1291	15800	STOK 0	A.Q8.32	/dev/rdisk/disk4305	16386
stok 0 39	1019	15800	STOK 0	C.Q1.2	/dev/rdisk/disk6404	16386
stok_0_390	1463	15800	STOK_0	B.Q8.29	/dev/rdisk/disk3950	16386
stok 0 391	1238	15800	STOK 0	C.Q2.6	/dev/rdisk/disk6545	16386
stok 0 392	1329	15800	STOK 0	A.Q7.27	/dev/rdisk/disk5867	16386
stok_0_393	1296	15800	STOK_0	C.Q1.2	/dev/rdisk/disk6437	16386
stok 0 394	1406	15800	STOK 0	C.Q5.19	/dev/rdisk/disk6412	16386
stok_0_395	1257	15800	STOK_0	B.Q7.26	/dev/rdisk/disk5484	16386
stok_0_396	1256	15800	STOK_0	C.Q5.20	/dev/rdisk/disk6760	16386
stok 0 397	1339	15800	STOK 0	B.Q8.30	/dev/rdisk/disk3710	16386
stok_0_398	1265	15800	STOK_0	B.Q7.25	/dev/rdisk/disk5774	16386
stok 0 399	1464	15800	STOK 0	B.Q12.45	/dev/rdisk/disk5231	16386
stok 0 4	1055	15800	STOK 0	A.Q6.21	/dev/rdisk/disk4569	16386
stok 0 40	1004	15800	STOK 0	C.Q5.19	/dev/rdisk/disk6766	16386
stok 0 400	1315	15800	STOK 0	B.Q5.17	/dev/rdisk/disk4446	16386
stok_0_401	1411	15800	STOK_0	B.Q3.11	/dev/rdisk/disk4635	16386
stok 0 402	1462	15800	STOK 0	A.Q9.35	/dev/rdisk/disk6101	16386
stok 0 403	1235	15800	STOK 0	C.Q4.14	/dev/rdisk/disk6689	16386
stok_0_404	1236	15800	STOK_0	A.Q5.17	/dev/rdisk/disk4944	16386
stok 0 405	1213	15800	STOK 0	A.Q4.15	/dev/rdisk/disk3951	16386
stok_0_406	1362	15800	STOK_0	A.Q4.13	/dev/rdisk/disk4071	16386
stok 0 407	1368	15800	STOK 0	A.Q1.1	/dev/rdisk/disk5274	16386
stok 0 408	1223	15800	STOK 0	A.Q8.29	/dev/rdisk/disk5162	16386
stok_0_409	1380	15800	STOK_0	C.Q5.18	/dev/rdisk/disk6608	16386
stok 0 41	1125	15800	STOK 0	B.Q7.26	/dev/rdisk/disk4235	16386
stok 0 410	1420	15800	STOK 0	C.Q2.8	/dev/rdisk/disk6547	16386
stok_0_411	1457	15800	STOK_0	B.Q10.37	/dev/rdisk/disk5257	16386
stok 0 412	1349	15800	STOK 0	A.Q13.52	/dev/rdisk/disk5281	16386

stok_0_413	1309	15800	STOK_0	A.Q13.50	/dev/rdisk/disk5099	16386
stok_0_414	1448	15800	STOK_0	B.Q5.19	/dev/rdisk/disk5251	16386
stok_0_415	1290	15800	STOK_0	C.Q6.22	/dev/rdisk/disk6341	16386
stok_0_416	1313	15800	STOK_0	B.Q2.8	/dev/rdisk/disk5179	16386
stok_0_417	1372	15800	STOK_0	B.Q3.10	/dev/rdisk/disk4844	16386
stok_0_418	1222	15800	STOK_0	B.Q9.33	/dev/rdisk/disk5136	16386
stok_0_419	1215	15800	STOK_0	A.Q6.24	/dev/rdisk/disk3747	16386
stok_0_42	980	15800	STOK_0	C.Q5.20	/dev/rdisk/disk6771	16386
stok_0_420	1278	15800	STOK_0	A.Q12.48	/dev/rdisk/disk5382	16386
stok_0_421	1304	15800	STOK_0	B.Q4.13	/dev/rdisk/disk4530	16386
stok_0_422	1342	15800	STOK_0	B.Q12.46	/dev/rdisk/disk6060	16386
stok_0_423	1402	15800	STOK_0	A.Q9.34	/dev/rdisk/disk6036	16386
stok_0_424	1258	15800	STOK_0	A.Q10.38	/dev/rdisk/disk4422	16386
stok_0_425	1292	15800	STOK_0	C.Q3.9	/dev/rdisk/disk6302	16386
stok_0_426	1439	15800	STOK_0	A.Q11.43	/dev/rdisk/disk6188	16386
stok_0_427	1227	15800	STOK_0	C.Q3.10	/dev/rdisk/disk6952	16386
stok_0_428	1297	15800	STOK_0	C.Q1.1	/dev/rdisk/disk6629	16386
stok_0_429	1332	15800	STOK_0	A.Q12.47	/dev/rdisk/disk4223	16386
stok_0_43	1061	15800	STOK_0	B.Q8.30	/dev/rdisk/disk3712	16386
stok_0_430	1458	15800	STOK_0	C.Q4.16	/dev/rdisk/disk6845	16386
stok_0_431	1301	15800	STOK_0	B.Q4.15	/dev/rdisk/disk4728	16386
stok_0_432	1335	15800	STOK_0	B.Q10.39	/dev/rdisk/disk5992	16386
stok_0_433	1379	15800	STOK_0	B.Q12.47	/dev/rdisk/disk4316	16386
stok_0_434	1212	15800	STOK_0	A.Q11.44	/dev/rdisk/disk6248	16386
stok_0_435	1449	15800	STOK_0	A.Q2.5	/dev/rdisk/disk3841	16386
stok_0_436	1316	15800	STOK_0	B.Q8.32	/dev/rdisk/disk4158	16386
stok_0_437	1310	15800	STOK_0	C.Q3.12	/dev/rdisk/disk6309	16386
stok_0_438	1442	15800	STOK_0	B.Q12.48	/dev/rdisk/disk4405	16386
stok_0_439	1239	15800	STOK_0	B.Q7.27	/dev/rdisk/disk5726	16386
stok_0_44	1039	15800	STOK_0	B.Q7.25	/dev/rdisk/disk5791	16386
stok_0_440	1234	15800	STOK_0	B.Q1.4	/dev/rdisk/disk5593	16386
stok_0_441	1373	15800	STOK_0	B.Q13.51	/dev/rdisk/disk4232	16386
stok_0_442	1284	15800	STOK_0	B.Q13.49	/dev/rdisk/disk5809	16386
stok_0_443	1391	15800	STOK_0	C.Q6.24	/dev/rdisk/disk6406	16386
stok_0_444	1224	15800	STOK_0	A.Q8.31	/dev/rdisk/disk4336	16386
stok_0_445	1341	15800	STOK_0	A.Q1.2	/dev/rdisk/disk4558	16386
stok_0_446	1414	15800	STOK_0	B.Q6.23	/dev/rdisk/disk4159	16386
stok_0_447	1361	15800	STOK_0	B.Q7.28	/dev/rdisk/disk5748	16386
stok_0_448	1350	15800	STOK_0	A.Q5.19	/dev/rdisk/disk4645	16386
stok_0_449	1264	15800	STOK_0	B.Q3.9	/dev/rdisk/disk5894	16386
stok_0_45	1075	15800	STOK_0	B.Q12.45	/dev/rdisk/disk5210	16386
stok_0_450	1408	15800	STOK_0	B.Q13.52	/dev/rdisk/disk6907	16386
stok_0_451	1240	15800	STOK_0	A.Q12.46	/dev/rdisk/disk4850	16386
stok_0_452	1333	15800	STOK_0	A.Q2.6	/dev/rdisk/disk3618	16386
stok_0_453	1387	15800	STOK_0	A.Q9.33	/dev/rdisk/disk5497	16386
stok_0_454	1407	15800	STOK_0	B.Q10.40	/dev/rdisk/disk5309	16386
stok_0_455	1351	15800	STOK_0	B.Q13.50	/dev/rdisk/disk4615	16386
stok_0_456	1415	15800	STOK_0	A.Q13.51	/dev/rdisk/disk4325	16386
stok_0_457	1318	15800	STOK_0	B.Q5.20	/dev/rdisk/disk4752	16386
stok_0_458	1413	15800	STOK_0	B.Q1.1	/dev/rdisk/disk4359	16386
stok_0_459	1220	15800	STOK_0	C.Q2.7	/dev/rdisk/disk6780	16386
stok_0_46	1028	15800	STOK_0	B.Q5.17	/dev/rdisk/disk4482	16386

stok 0 460	1299	15800	STOK 0	C.Q1.3	/dev/rdisk/disk6871	16386
stok 0 461	1428	15800	STOK 0	C.Q1.4	/dev/rdisk/disk6472	16386
stok_0_462	1437	15800	STOK_0	C.Q6.21	/dev/rdisk/disk6365	16386
stok 0 463	1396	15800	STOK 0	C.Q5.17	/dev/rdisk/disk6791	16386
stok 0 464	1241	15800	STOK 0	B.Q4.14	/dev/rdisk/disk5359	16386
stok_0_465	1228	15800	STOK_0	A.Q1.4	/dev/rdisk/disk6923	16386
stok 0 466	1403	15800	STOK 0	A.Q3.9	/dev/rdisk/disk3904	16386
stok_0_467	1274	15800	STOK_0	A.Q2.8	/dev/rdisk/disk3660	16386
stok_0_468	1422	15800	STOK_0	A.Q2.7	/dev/rdisk/disk3623	16386
stok 0 469	1302	15800	STOK 0	B.Q1.3	/dev/rdisk/disk5508	16386
stok_0_47	1051	15800	STOK_0	B.Q3.11	/dev/rdisk/disk4685	16386
stok 0 470	1423	15800	STOK 0	B.Q5.18	/dev/rdisk/disk6129	16386
stok 0 471	1441	15800	STOK 0	B.Q9.34	/dev/rdisk/disk5494	16386
stok 0 472	1389	15800	STOK 0	C.Q4.15	/dev/rdisk/disk6669	16386
stok 0 473	1412	15800	STOK 0	B.Q3.12	/dev/rdisk/disk4738	16386
stok_0_474	1283	15800	STOK_0	A.Q7.26	/dev/rdisk/disk4095	16386
stok 0 475	1271	15800	STOK 0	A.Q3.10	/dev/rdisk/disk3827	16386
stok 0 476	1397	15800	STOK 0	A.Q6.21	/dev/rdisk/disk4646	16386
stok_0_477	1293	15800	STOK_0	C.Q3.11	/dev/rdisk/disk6294	16386
stok 0 478	1438	15800	STOK 0	A.Q7.25	/dev/rdisk/disk6042	16386
stok_0_479	1359	15800	STOK_0	A.Q4.16	/dev/rdisk/disk4024	16386
stok 0 48	1192	15800	STOK 0	A.Q9.35	/dev/rdisk/disk6122	16386
stok 0 480	1279	15800	STOK 0	B.Q2.6	/dev/rdisk/disk5862	16386
stok_0_481	1268	15800	STOK_0	A.Q3.11	/dev/rdisk/disk3760	16386
stok 0 482	1416	15800	STOK 0	A.Q5.18	/dev/rdisk/disk4511	16386
stok 0 483	1252	15800	STOK 0	A.Q5.20	/dev/rdisk/disk4543	16386
stok_0_484	1317	15800	STOK_0	A.Q4.14	/dev/rdisk/disk4066	16386
stok 0 485	1429	15800	STOK 0	A.Q10.39	/dev/rdisk/disk4785	16386
stok 0 486	1230	15800	STOK 0	A.Q1.3	/dev/rdisk/disk4426	16386
stok_0_487	1314	15800	STOK_0	B.Q8.31	/dev/rdisk/disk5834	16386
stok 0 488	1374	15800	STOK 0	B.Q10.38	/dev/rdisk/disk4800	16386
stok_0_489	1242	15800	STOK_0	B.Q2.5	/dev/rdisk/disk4088	16386
stok_0_49	1177	15800	STOK_0	C.Q4.14	/dev/rdisk/disk6855	16386
stok 0 490	1363	15800	STOK 0	B.Q6.21	/dev/rdisk/disk3837	16386
stok_0_491	1392	15800	STOK_0	A.Q11.42	/dev/rdisk/disk6204	16386
stok 0 492	1248	15800	STOK 0	A.Q10.40	/dev/rdisk/disk5925	16386
stok 0 493	1219	15800	STOK 0	B.Q1.2	/dev/rdisk/disk4972	16386
stok 0 494	1417	15800	STOK 0	C.Q2.5	/dev/rdisk/disk6360	16386
stok 0 495	1450	15800	STOK 0	A.Q8.30	/dev/rdisk/disk4287	16386
stok_0_496	1348	15800	STOK_0	A.Q3.12	/dev/rdisk/disk3994	16386
stok 0 497	1322	15800	STOK 0	B.Q6.24	/dev/rdisk/disk5817	16386
stok 0 498	1446	15800	STOK 0	B.Q2.7	/dev/rdisk/disk5602	16386
stok_0_499	1388	15800	STOK_0	C.Q6.23	/dev/rdisk/disk6573	16386
stok 0 5	1190	15800	STOK 0	C.Q3.11	/dev/rdisk/disk6307	16386
stok_0_50	1191	15800	STOK_0	A.Q5.17	/dev/rdisk/disk4916	16386
stok 0 500	1398	15800	STOK 0	C.Q4.13	/dev/rdisk/disk6619	16386
stok 0 501	1330	15800	STOK 0	A.Q9.36	/dev/rdisk/disk5535	16386
stok_0_502	1466	15800	STOK_0	B.Q6.22	/dev/rdisk/disk3745	16386
stok 0 503	1352	15800	STOK 0	A.Q6.23	/dev/rdisk/disk5417	16386
stok 0 504	1418	15800	STOK 0	B.Q4.16	/dev/rdisk/disk4539	16386
stok_0_505	1259	15800	STOK_0	A.Q7.28	/dev/rdisk/disk4192	16386
stok 0 506	1320	15800	STOK 0	A.Q6.22	/dev/rdisk/disk6039	16386

stok 0 507	1404	15800	STOK 0	A.Q8.32	/dev/rdisk/disk5662	16386
stok 0 508	1353	15800	STOK 0	B.Q8.29	/dev/rdisk/disk3920	16386
stok_0_509	1280	15800	STOK_0	C.Q2.6	/dev/rdisk/disk6599	16386
stok 0 51	1012	15800	STOK 0	A.Q4.15	/dev/rdisk/disk3931	16386
stok 0 510	1251	15800	STOK 0	A.Q7.27	/dev/rdisk/disk5869	16386
stok_0_511	1272	15800	STOK_0	C.Q1.2	/dev/rdisk/disk6388	16386
stok 0 512	1399	15800	STOK 0	C.Q5.19	/dev/rdisk/disk6346	16386
stok_0_513	1548	15800	STOK_0	B.Q7.26	/dev/rdisk/disk5475	16386
stok_0_514	1580	15800	STOK_0	C.Q5.20	/dev/rdisk/disk6643	16386
stok 0 515	1485	15800	STOK 0	B.Q8.30	/dev/rdisk/disk3707	16386
stok_0_516	1513	15800	STOK_0	B.Q7.25	/dev/rdisk/disk5770	16386
stok 0 517	1501	15800	STOK 0	B.Q12.45	/dev/rdisk/disk5565	16386
stok 0 518	1592	15800	STOK 0	B.Q5.17	/dev/rdisk/disk4439	16386
stok 0 519	1467	15800	STOK 0	B.Q3.11	/dev/rdisk/disk4637	16386
stok 0 52	1057	15800	STOK 0	A.Q4.13	/dev/rdisk/disk4073	16386
stok_0_520	1593	15800	STOK_0	A.Q9.35	/dev/rdisk/disk6088	16386
stok 0 521	1537	15800	STOK 0	C.Q4.14	/dev/rdisk/disk6658	16386
stok 0 522	1603	15800	STOK 0	A.Q5.17	/dev/rdisk/disk4962	16386
stok_0_523	1609	15800	STOK_0	A.Q4.15	/dev/rdisk/disk3943	16386
stok 0 524	1568	15800	STOK 0	A.Q4.13	/dev/rdisk/disk4078	16386
stok_0_525	1567	15800	STOK_0	A.Q1.1	/dev/rdisk/disk5276	16386
stok 0 526	1585	15800	STOK 0	A.Q8.29	/dev/rdisk/disk5061	16386
stok 0 527	1574	15800	STOK 0	C.Q5.18	/dev/rdisk/disk6804	16386
stok_0_528	1514	15800	STOK_0	C.Q2.8	/dev/rdisk/disk6526	16386
stok 0 529	1531	15800	STOK 0	B.Q10.37	/dev/rdisk/disk5247	16386
stok 0 53	1126	15800	STOK 0	A.Q1.1	/dev/rdisk/disk5239	16386
stok_0_530	1477	15800	STOK_0	A.Q13.52	/dev/rdisk/disk5284	16386
stok 0 531	1604	15800	STOK 0	A.Q13.50	/dev/rdisk/disk5077	16386
stok 0 532	1557	15800	STOK 0	B.Q5.19	/dev/rdisk/disk5201	16386
stok_0_533	1503	15800	STOK_0	C.Q6.22	/dev/rdisk/disk6344	16386
stok 0 534	1586	15800	STOK 0	B.Q2.8	/dev/rdisk/disk5178	16386
stok_0_535	1572	15800	STOK_0	B.Q3.10	/dev/rdisk/disk4831	16386
stok_0_536	1610	15800	STOK_0	B.Q9.33	/dev/rdisk/disk5102	16386
stok 0 537	1495	15800	STOK 0	A.Q6.24	/dev/rdisk/disk3744	16386
stok_0_538	1605	15800	STOK_0	A.Q12.48	/dev/rdisk/disk4269	16386
stok 0 539	1594	15800	STOK 0	B.Q4.13	/dev/rdisk/disk4521	16386
stok 0 54	1020	15800	STOK 0	A.Q8.29	/dev/rdisk/disk5159	16386
stok 0 540	1595	15800	STOK 0	B.Q12.46	/dev/rdisk/disk6059	16386
stok 0 541	1517	15800	STOK 0	A.Q9.34	/dev/rdisk/disk6035	16386
stok_0_542	1536	15800	STOK_0	A.Q10.38	/dev/rdisk/disk4424	16386
stok 0 543	1482	15800	STOK 0	C.Q3.9	/dev/rdisk/disk6305	16386
stok 0 544	1518	15800	STOK 0	A.Q11.43	/dev/rdisk/disk6187	16386
stok_0_545	1492	15800	STOK_0	C.Q3.10	/dev/rdisk/disk6951	16386
stok 0 546	1509	15800	STOK 0	C.Q1.1	/dev/rdisk/disk6627	16386
stok_0_547	1564	15800	STOK_0	A.Q12.47	/dev/rdisk/disk4224	16386
stok 0 548	1565	15800	STOK 0	C.Q4.16	/dev/rdisk/disk6668	16386
stok 0 549	1560	15800	STOK 0	B.Q4.15	/dev/rdisk/disk4759	16386
stok_0_55	1103	15800	STOK_0	C.Q5.18	/dev/rdisk/disk6830	16386
stok 0 550	1489	15800	STOK 0	B.Q10.39	/dev/rdisk/disk5991	16386
stok 0 551	1530	15800	STOK 0	B.Q12.47	/dev/rdisk/disk4381	16386
stok_0_552	1515	15800	STOK_0	A.Q11.44	/dev/rdisk/disk6268	16386
stok 0 553	1544	15800	STOK 0	A.Q2.5	/dev/rdisk/disk3839	16386

stok 0 554	1541	15800	STOK 0	B.Q8.32	/dev/rdisk/disk4156	16386
stok 0 555	1587	15800	STOK 0	C.Q3.12	/dev/rdisk/disk6292	16386
stok_0_556	1519	15800	STOK_0	B.Q12.48	/dev/rdisk/disk5066	16386
stok 0 557	1522	15800	STOK 0	B.Q7.27	/dev/rdisk/disk5728	16386
stok 0 558	1510	15800	STOK 0	B.Q1.4	/dev/rdisk/disk5595	16386
stok_0_559	1573	15800	STOK_0	B.Q13.51	/dev/rdisk/disk4244	16386
stok 0 56	1077	15800	STOK 0	C.Q2.8	/dev/rdisk/disk6529	16386
stok_0_560	1497	15800	STOK_0	B.Q13.49	/dev/rdisk/disk4293	16386
stok_0_561	1496	15800	STOK_0	C.Q6.24	/dev/rdisk/disk6436	16386
stok 0 562	1475	15800	STOK 0	A.Q8.31	/dev/rdisk/disk5506	16386
stok_0_563	1576	15800	STOK_0	A.Q1.2	/dev/rdisk/disk4567	16386
stok 0 564	1493	15800	STOK 0	B.Q6.23	/dev/rdisk/disk4157	16386
stok 0 565	1486	15800	STOK 0	B.Q7.28	/dev/rdisk/disk5719	16386
stok 0 566	1596	15800	STOK 0	A.Q5.19	/dev/rdisk/disk4644	16386
stok 0 567	1606	15800	STOK 0	B.Q3.9	/dev/rdisk/disk5914	16386
stok_0_568	1575	15800	STOK_0	B.Q13.52	/dev/rdisk/disk6904	16386
stok 0 569	1505	15800	STOK 0	A.Q12.46	/dev/rdisk/disk4373	16386
stok 0 57	1082	15800	STOK 0	B.Q10.37	/dev/rdisk/disk4843	16386
stok_0_570	1549	15800	STOK_0	A.Q2.6	/dev/rdisk/disk3617	16386
stok 0 571	1528	15800	STOK 0	A.Q9.33	/dev/rdisk/disk4264	16386
stok_0_572	1569	15800	STOK_0	B.Q10.40	/dev/rdisk/disk5313	16386
stok 0 573	1498	15800	STOK 0	B.Q13.50	/dev/rdisk/disk4630	16386
stok 0 574	1589	15800	STOK 0	A.Q13.51	/dev/rdisk/disk4324	16386
stok_0_575	1490	15800	STOK_0	B.Q5.20	/dev/rdisk/disk5153	16386
stok 0 576	1516	15800	STOK 0	B.Q1.1	/dev/rdisk/disk4356	16386
stok 0 577	1545	15800	STOK 0	C.Q2.7	/dev/rdisk/disk6776	16386
stok_0_578	1597	15800	STOK_0	C.Q1.3	/dev/rdisk/disk6870	16386
stok 0 579	1523	15800	STOK 0	C.Q1.4	/dev/rdisk/disk6449	16386
stok 0 58	1018	15800	STOK 0	A.Q13.52	/dev/rdisk/disk4306	16386
stok_0_580	1577	15800	STOK_0	C.Q6.21	/dev/rdisk/disk6394	16386
stok 0 581	1524	15800	STOK 0	C.Q5.17	/dev/rdisk/disk6777	16386
stok_0_582	1588	15800	STOK_0	B.Q4.14	/dev/rdisk/disk5422	16386
stok_0_583	1561	15800	STOK_0	A.Q1.4	/dev/rdisk/disk6930	16386
stok 0 584	1611	15800	STOK 0	A.Q3.9	/dev/rdisk/disk3901	16386
stok_0_585	1554	15800	STOK_0	A.Q2.8	/dev/rdisk/disk3652	16386
stok 0 586	1581	15800	STOK 0	A.Q2.7	/dev/rdisk/disk3628	16386
stok 0 587	1546	15800	STOK 0	B.Q1.3	/dev/rdisk/disk5516	16386
stok 0 588	1525	15800	STOK 0	B.Q5.18	/dev/rdisk/disk6127	16386
stok 0 589	1491	15800	STOK 0	B.Q9.34	/dev/rdisk/disk4347	16386
stok_0_59	1005	15800	STOK_0	A.Q13.50	/dev/rdisk/disk4310	16386
stok 0 590	1506	15800	STOK 0	C.Q4.15	/dev/rdisk/disk6682	16386
stok 0 591	1607	15800	STOK 0	B.Q3.12	/dev/rdisk/disk4792	16386
stok_0_592	1579	15800	STOK_0	A.Q7.26	/dev/rdisk/disk5678	16386
stok 0 593	1542	15800	STOK 0	A.Q3.10	/dev/rdisk/disk3825	16386
stok_0_594	1550	15800	STOK_0	A.Q6.21	/dev/rdisk/disk4681	16386
stok 0 595	1566	15800	STOK 0	C.Q3.11	/dev/rdisk/disk6291	16386
stok 0 596	1562	15800	STOK 0	A.Q7.25	/dev/rdisk/disk6040	16386
stok_0_597	1571	15800	STOK_0	A.Q4.16	/dev/rdisk/disk4021	16386
stok 0 598	1511	15800	STOK 0	B.Q2.6	/dev/rdisk/disk5596	16386
stok 0 599	1578	15800	STOK 0	A.Q3.11	/dev/rdisk/disk3777	16386
stok_0_6	1010	15800	STOK_0	A.Q7.25	/dev/rdisk/disk6055	16386
stok 0 60	1127	15800	STOK 0	B.Q5.19	/dev/rdisk/disk5187	16386

stok 0 600	1507	15800	STOK 0	A.Q05.18	/dev/rdisk/disk4503	16386
stok 0 601	1484	15800	STOK 0	A.Q05.20	/dev/rdisk/disk4592	16386
stok_0_602	1538	15800	STOK_0	A.Q04.14	/dev/rdisk/disk4077	16386
stok 0 603	1582	15800	STOK 0	A.Q01.3	/dev/rdisk/disk4429	16386
stok 0 604	1600	15800	STOK 0	A.Q10.39	/dev/rdisk/disk4787	16386
stok_0_605	1539	15800	STOK_0	B.Q08.31	/dev/rdisk/disk5825	16386
stok 0 606	1512	15800	STOK 0	B.Q10.38	/dev/rdisk/disk4375	16386
stok_0_607	1563	15800	STOK_0	B.Q02.5	/dev/rdisk/disk5953	16386
stok_0_608	1612	15800	STOK_0	B.Q06.21	/dev/rdisk/disk3835	16386
stok 0 609	1570	15800	STOK 0	A.Q11.42	/dev/rdisk/disk6200	16386
stok_0_61	1111	15800	STOK_0	C.Q06.22	/dev/rdisk/disk6369	16386
stok 0 610	1551	15800	STOK 0	A.Q10.40	/dev/rdisk/disk5923	16386
stok 0 611	1558	15800	STOK 0	B.Q01.2	/dev/rdisk/disk4387	16386
stok 0 612	1526	15800	STOK 0	C.Q02.5	/dev/rdisk/disk6440	16386
stok 0 613	1535	15800	STOK 0	A.Q08.30	/dev/rdisk/disk4286	16386
stok_0_614	1508	15800	STOK_0	A.Q03.12	/dev/rdisk/disk3993	16386
stok 0 615	1478	15800	STOK 0	B.Q06.24	/dev/rdisk/disk5814	16386
stok 0 616	1487	15800	STOK 0	B.Q02.7	/dev/rdisk/disk4210	16386
stok_0_617	1479	15800	STOK_0	C.Q06.23	/dev/rdisk/disk6556	16386
stok 0 618	1468	15800	STOK 0	C.Q04.13	/dev/rdisk/disk6618	16386
stok_0_619	1613	15800	STOK_0	A.Q09.36	/dev/rdisk/disk5599	16386
stok 0 62	1066	15800	STOK 0	B.Q02.8	/dev/rdisk/disk4812	16386
stok 0 620	1499	15800	STOK 0	B.Q06.22	/dev/rdisk/disk3748	16386
stok_0_621	1472	15800	STOK_0	A.Q06.23	/dev/rdisk/disk5444	16386
stok 0 622	1527	15800	STOK 0	B.Q04.16	/dev/rdisk/disk4619	16386
stok 0 623	1601	15800	STOK 0	A.Q07.28	/dev/rdisk/disk5973	16386
stok_0_624	1469	15800	STOK_0	A.Q06.22	/dev/rdisk/disk6037	16386
stok 0 625	1473	15800	STOK 0	A.Q08.32	/dev/rdisk/disk5654	16386
stok 0 626	1471	15800	STOK 0	B.Q08.29	/dev/rdisk/disk3935	16386
stok_0_627	1470	15800	STOK_0	C.Q02.6	/dev/rdisk/disk6587	16386
stok 0 628	1483	15800	STOK 0	A.Q07.27	/dev/rdisk/disk5872	16386
stok_0_629	1520	15800	STOK_0	C.Q01.2	/dev/rdisk/disk6373	16386
stok_0_63	1021	15800	STOK_0	B.Q03.10	/dev/rdisk/disk4968	16386
stok 0 630	1476	15800	STOK 0	C.Q05.19	/dev/rdisk/disk6691	16386
stok_0_631	1502	15800	STOK_0	B.Q07.26	/dev/rdisk/disk5477	16386
stok 0 632	1533	15800	STOK 0	C.Q05.20	/dev/rdisk/disk6736	16386
stok 0 633	1529	15800	STOK 0	B.Q08.30	/dev/rdisk/disk3708	16386
stok 0 634	1552	15800	STOK 0	B.Q07.25	/dev/rdisk/disk5743	16386
stok 0 635	1504	15800	STOK 0	B.Q12.45	/dev/rdisk/disk5556	16386
stok_0_636	1540	15800	STOK_0	B.Q05.17	/dev/rdisk/disk4430	16386
stok 0 637	1488	15800	STOK 0	B.Q03.11	/dev/rdisk/disk4633	16386
stok 0 638	1480	15800	STOK 0	A.Q09.35	/dev/rdisk/disk6085	16386
stok_0_639	1583	15800	STOK_0	C.Q04.14	/dev/rdisk/disk6807	16386
stok 0 64	1058	15800	STOK 0	B.Q09.33	/dev/rdisk/disk4505	16386
stok_0_640	1481	15800	STOK_0	A.Q05.17	/dev/rdisk/disk4880	16386
stok 0 641	1543	15800	STOK 0	A.Q04.15	/dev/rdisk/disk3953	16386
stok 0 642	1590	15800	STOK 0	A.Q04.13	/dev/rdisk/disk4065	16386
stok_0_643	1494	15800	STOK_0	A.Q01.1	/dev/rdisk/disk4974	16386
stok 0 644	1534	15800	STOK 0	A.Q08.29	/dev/rdisk/disk5037	16386
stok 0 645	1555	15800	STOK 0	C.Q05.18	/dev/rdisk/disk6601	16386
stok_0_646	1584	15800	STOK_0	C.Q02.8	/dev/rdisk/disk6387	16386
stok 0 647	1547	15800	STOK 0	B.Q10.37	/dev/rdisk/disk5170	16386

stok 0 648	1474	15800	STOK 0	A.Q13.52	/dev/rdisk/disk5349	16386
stok 0 649	1532	15800	STOK 0	A.Q13.50	/dev/rdisk/disk5080	16386
stok_0_65	1193	15800	STOK_0	A.Q6.24	/dev/rdisk/disk3759	16386
stok 0 650	1598	15800	STOK 0	B.Q5.19	/dev/rdisk/disk5194	16386
stok 0 651	1608	15800	STOK 0	C.Q6.22	/dev/rdisk/disk6351	16386
stok_0_652	1559	15800	STOK_0	B.Q2.8	/dev/rdisk/disk5177	16386
stok 0 653	1599	15800	STOK 0	B.Q3.10	/dev/rdisk/disk4835	16386
stok_0_654	1553	15800	STOK_0	B.Q9.33	/dev/rdisk/disk5111	16386
stok_0_655	1614	15800	STOK_0	A.Q6.24	/dev/rdisk/disk3735	16386
stok 0 656	1500	15800	STOK 0	A.Q12.48	/dev/rdisk/disk4273	16386
stok_0_657	1556	15800	STOK_0	B.Q4.13	/dev/rdisk/disk4714	16386
stok 0 658	1591	15800	STOK 0	B.Q12.46	/dev/rdisk/disk6046	16386
stok 0 659	1602	15800	STOK 0	A.Q9.34	/dev/rdisk/disk6026	16386
stok 0 66	993	15800	STOK 0	A.Q12.48	/dev/rdisk/disk5794	16386
stok 0 660	1521	15800	STOK 0	A.Q10.38	/dev/rdisk/disk5012	16386
stok_0_67	1078	15800	STOK_0	B.Q4.13	/dev/rdisk/disk4528	16386
stok 0 68	1151	15800	STOK 0	B.Q12.46	/dev/rdisk/disk6078	16386
stok 0 69	1152	15800	STOK 0	A.Q9.34	/dev/rdisk/disk6048	16386
stok_0_7	969	15800	STOK_0	A.Q4.16	/dev/rdisk/disk4039	16386
stok 0 70	1143	15800	STOK 0	A.Q10.38	/dev/rdisk/disk5023	16386
stok_0_71	1053	15800	STOK_0	C.Q3.9	/dev/rdisk/disk6299	16386
stok 0 72	1099	15800	STOK 0	A.Q11.43	/dev/rdisk/disk6223	16386
stok 0 73	1159	15800	STOK 0	C.Q3.10	/dev/rdisk/disk6955	16386
stok_0_74	1139	15800	STOK_0	C.Q1.1	/dev/rdisk/disk6567	16386
stok 0 75	1033	15800	STOK 0	A.Q12.47	/dev/rdisk/disk4229	16386
stok 0 76	1092	15800	STOK 0	C.Q4.16	/dev/rdisk/disk6684	16386
stok_0_77	1128	15800	STOK_0	B.Q4.15	/dev/rdisk/disk5043	16386
stok 0 78	1014	15800	STOK 0	B.Q10.39	/dev/rdisk/disk4358	16386
stok 0 79	1062	15800	STOK 0	B.Q12.47	/dev/rdisk/disk5607	16386
stok_0_8	1132	15800	STOK_0	B.Q2.6	/dev/rdisk/disk4202	16386
stok 0 80	1201	15800	STOK 0	A.Q11.44	/dev/rdisk/disk6231	16386
stok_0_81	1079	15800	STOK_0	A.Q2.5	/dev/rdisk/disk3856	16386
stok_0_82	979	15800	STOK_0	B.Q8.32	/dev/rdisk/disk4146	16386
stok 0 83	1171	15800	STOK 0	C.Q3.12	/dev/rdisk/disk6314	16386
stok_0_84	1115	15800	STOK_0	B.Q12.48	/dev/rdisk/disk4413	16386
stok 0 85	972	15800	STOK 0	B.Q7.27	/dev/rdisk/disk5344	16386
stok 0 86	1194	15800	STOK 0	B.Q1.4	/dev/rdisk/disk5589	16386
stok 0 87	1015	15800	STOK 0	B.Q13.51	/dev/rdisk/disk4251	16386
stok 0 88	1163	15800	STOK 0	B.Q13.49	/dev/rdisk/disk5820	16386
stok_0_89	957	15800	STOK_0	C.Q6.24	/dev/rdisk/disk6470	16386
stok 0 9	1072	15800	STOK 0	A.Q3.11	/dev/rdisk/disk3756	16386
stok 0 90	1081	15800	STOK 0	A.Q8.31	/dev/rdisk/disk4865	16386
stok_0_91	1076	15800	STOK_0	A.Q1.2	/dev/rdisk/disk4732	16386
stok 0 92	1153	15800	STOK 0	B.Q6.23	/dev/rdisk/disk3983	16386
stok_0_93	965	15800	STOK_0	B.Q7.28	/dev/rdisk/disk5764	16386
stok 0 94	994	15800	STOK 0	A.Q5.19	/dev/rdisk/disk4694	16386
stok 0 95	1006	15800	STOK 0	B.Q3.9	/dev/rdisk/disk5912	16386
stok_0_96	1129	15800	STOK_0	B.Q13.52	/dev/rdisk/disk6906	16386
stok 0 97	1112	15800	STOK 0	A.Q12.46	/dev/rdisk/disk5248	16386
stok 0 98	1195	15800	STOK 0	A.Q2.6	/dev/rdisk/disk3630	16386
stok_0_99	1083	15800	STOK_0	A.Q9.33	/dev/rdisk/disk5545	16386
system 1	1	400	SYSTEM	All	/dev/vgdata1/rsystem 1	1888

temp 0 0	N/A	N/A	N/A	All	/dev/vgdata1/rtemp 0 0	16992
temp 0 1	N/A	N/A	N/A	All	/dev/vgdata2/rtemp 0 1	16992
temp_0_10	N/A	N/A	N/A	All	/dev/vgdata1/rtemp_0_10	16992
temp 0 11	N/A	N/A	N/A	All	/dev/vgdata2/rtemp 0 11	16992
temp 0 12	N/A	N/A	N/A	All	/dev/vgdata1/rtemp 0 12	16992
temp_0_13	N/A	N/A	N/A	All	/dev/vgdata2/rtemp_0_13	16992
temp 0 14	N/A	N/A	N/A	All	/dev/vgdata1/rtemp 0 14	16992
temp_0_15	N/A	N/A	N/A	All	/dev/vgdata2/rtemp_0_15	16992
temp_0_16	N/A	N/A	N/A	All	/dev/vgdata1/rtemp_0_16	16992
temp 0 17	N/A	N/A	N/A	All	/dev/vgdata2/rtemp 0 17	16992
temp_0_18	N/A	N/A	N/A	All	/dev/vgdata1/rtemp_0_18	16992
temp 0 19	N/A	N/A	N/A	All	/dev/vgdata2/rtemp 0 19	16992
temp 0 2	N/A	N/A	N/A	All	/dev/vgdata1/rtemp 0 2	16992
temp 0 20	N/A	N/A	N/A	All	/dev/vgdata1/rtemp 0 20	16992
temp 0 21	N/A	N/A	N/A	All	/dev/vgdata2/rtemp 0 21	16992
temp_0_22	N/A	N/A	N/A	All	/dev/vgdata1/rtemp_0_22	16992
temp 0 23	N/A	N/A	N/A	All	/dev/vgdata2/rtemp 0 23	16992
temp 0 24	N/A	N/A	N/A	All	/dev/vgdata1/rtemp 0 24	16992
temp_0_25	N/A	N/A	N/A	All	/dev/vgdata2/rtemp_0_25	16992
temp 0 26	N/A	N/A	N/A	All	/dev/vgdata1/rtemp 0 26	16992
temp_0_27	N/A	N/A	N/A	All	/dev/vgdata2/rtemp_0_27	16992
temp 0 28	N/A	N/A	N/A	All	/dev/vgdata1/rtemp 0 28	16992
temp 0 29	N/A	N/A	N/A	All	/dev/vgdata2/rtemp 0 29	16992
temp_0_3	N/A	N/A	N/A	All	/dev/vgdata2/rtemp_0_3	16992
temp 0 30	N/A	N/A	N/A	All	/dev/vgdata1/rtemp 0 30	16992
temp 0 31	N/A	N/A	N/A	All	/dev/vgdata2/rtemp 0 31	16992
temp_0_32	N/A	N/A	N/A	All	/dev/vgdata1/rtemp_0_32	16992
temp 0 33	N/A	N/A	N/A	All	/dev/vgdata2/rtemp 0 33	16992
temp 0 34	N/A	N/A	N/A	All	/dev/vgdata1/rtemp 0 34	16992
temp_0_35	N/A	N/A	N/A	All	/dev/vgdata2/rtemp_0_35	16992
temp 0 36	N/A	N/A	N/A	All	/dev/vgdata1/rtemp 0 36	16992
temp_0_37	N/A	N/A	N/A	All	/dev/vgdata2/rtemp_0_37	16992
temp_0_38	N/A	N/A	N/A	All	/dev/vgdata1/rtemp_0_38	16992
temp 0 39	N/A	N/A	N/A	All	/dev/vgdata2/rtemp 0 39	16992
temp_0_4	N/A	N/A	N/A	All	/dev/vgdata1/rtemp_0_4	16992
temp 0 40	N/A	N/A	N/A	All	/dev/vgdata1/rtemp 0 40	16992
temp 0 41	N/A	N/A	N/A	All	/dev/vgdata2/rtemp 0 41	16992
temp 0 42	N/A	N/A	N/A	All	/dev/vgdata1/rtemp 0 42	16992
temp 0 43	N/A	N/A	N/A	All	/dev/vgdata2/rtemp 0 43	16992
temp_0_44	N/A	N/A	N/A	All	/dev/vgdata1/rtemp_0_44	16992
temp 0 45	N/A	N/A	N/A	All	/dev/vgdata2/rtemp 0 45	16992
temp 0 46	N/A	N/A	N/A	All	/dev/vgdata1/rtemp 0 46	16992
temp_0_47	N/A	N/A	N/A	All	/dev/vgdata2/rtemp_0_47	16992
temp 0 48	N/A	N/A	N/A	All	/dev/vgdata1/rtemp 0 48	16992
temp_0_49	N/A	N/A	N/A	All	/dev/vgdata2/rtemp_0_49	16992
temp 0 5	N/A	N/A	N/A	All	/dev/vgdata2/rtemp 0 5	16992
temp 0 6	N/A	N/A	N/A	All	/dev/vgdata1/rtemp 0 6	16992
temp_0_7	N/A	N/A	N/A	All	/dev/vgdata2/rtemp_0_7	16992
temp 0 8	N/A	N/A	N/A	All	/dev/vgdata1/rtemp 0 8	16992
temp 0 9	N/A	N/A	N/A	All	/dev/vgdata2/rtemp 0 9	16992
tpccaux	2	120	SYSAUX	All	/dev/vgdata1/rtpccaux	1888
ware 0 0	4	740	WARE 0	All	/dev/vgdata1/rware 0 0	1888

The distribution of the database tables over the disk arrays of the priced system is an extension of the distribution described in Table 5.2; some ancillary details are mentioned in Appendix E. 60-day storage growth requirements are met with the unused space of this configuration. Figure 1.2 shows the configuration of the priced-system disks.

5.3 Data Model & Interfaces

A statement must be provided that describes:

1. *The data model implemented by the DBMS used (e.g. relational, network, hierarchical)*
2. *The database interface used (e.g. embedded, call-level) and access language (e.g. SQL, DL/I, COBOL, read/write) used to implement the TPC-C transactions. If more than one interface/access language is used to implement TPC-C, each interface/access language must be described and a list of which interface/access language is used with which transaction type must be disclosed.*

Oracle Database 10g Release 2 Enterprise Edition with Partitioning is a relational DBMS. SQL stored procedures were used, invoked through the Oracle Call Interface (OCI); the application code appears in Appendix A.

5.4 Partitions/Replications

The mapping of database partitions/replications must be explicitly described.

Horizontal partitioning was used for one of the tables and one of indices. The detail of this partitioning can be understood by examining the table and index definition statements in Appendix B.

5.5 Growth Requirements

Details of the 60 day space computations along with proof that the database is configured to sustain 8 hours for the dynamic tables (Order, Order-Line, and History) must be disclosed.

See Appendix E.

6 Clause 5: Performance Metrics and Response Time Related Items

6.1 Throughput

Measured tpmC must be reported.

Table 6.1: Measured tpmC

tpmC [®]	4,092,799.42
-------------------	--------------

6.2 Response Time

Ninetieth percentile, maximum and average response times must be reported for all transaction types as well as for the menu response time.

Table 6.2: Response Times

Response Times	Average	90th %-ile	Maximum
New-Order	0.24s	0.51s	84.83s
Payment	0.23s	0.49s	85.20s
Order-Status	0.23s	0.51s	79.95s
Delivery (interactive portion)	0.14s	0.10s	81.97s
Delivery (deferred portion)	0.21s	0.50s	82.38s
Stock-Level	0.31s	0.61s	81.88s
Menu	2.470s	0.10s	0.28s

6.3 Keying and Think Times

The minimum, the average, and the maximum keying and think times must be reported for each transaction type.

Table 6.3: Keying Times

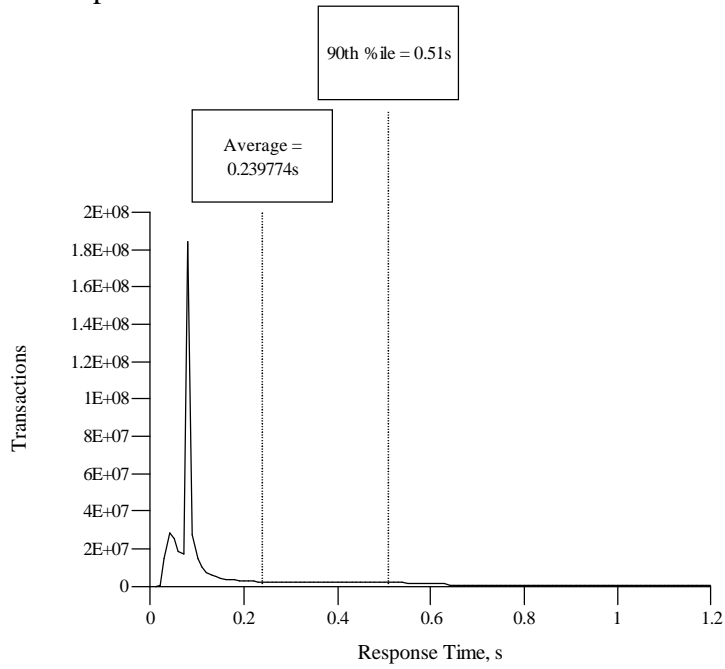
Keying Times	Minimum	Average	Maximum
New Order	18.01s	18.02s	18.19s
Payment	3.01s	3.02s	3.19s
Order Status	2.01s	2.02s	2.18s
Interactive Delivery	2.01s	2.02s	2.19s
Stock Level	2.01s	2.02s	2.19s

Table 6.4: Think Times

Think Times	Minimum	Average	Maximum
New Order	0.02s	12.02s	251.71s
Payment	0.02s	12.02s	246.09s
Order Status	0.02s	10.02s	171.08s
Interactive Delivery	0.02s	5.03s	95.32s
Stock Level	0.02s	5.02s	89.68s

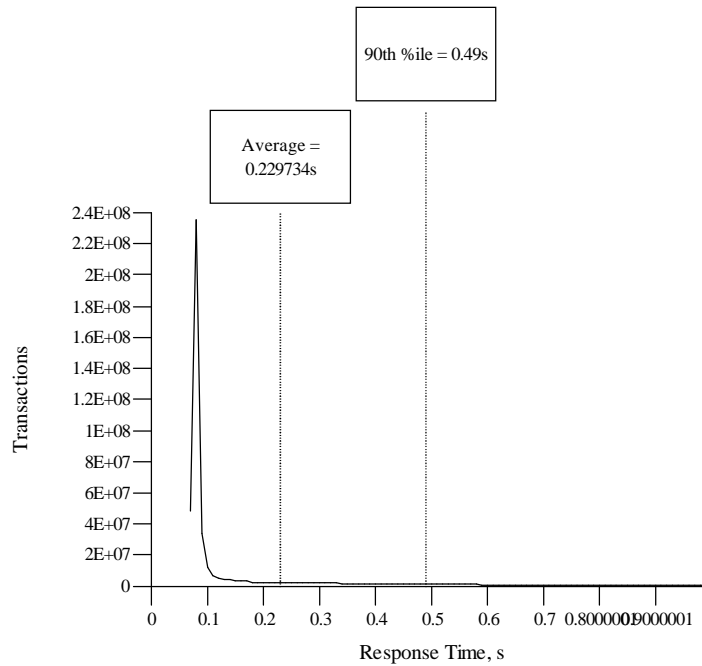
6.4 Response Time Frequency Distribution Curves and Other Graphs

Figure 6.1: New Order Response Time Distribution



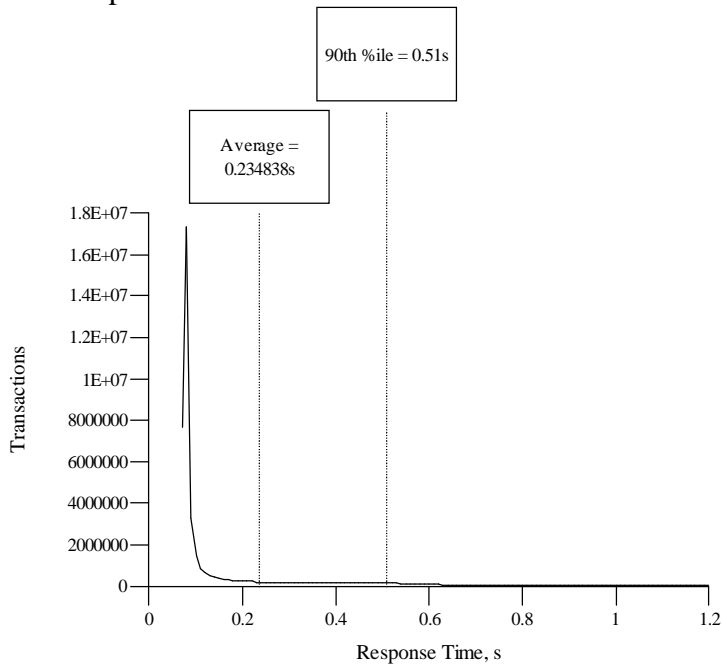
Response time frequency distribution for New Order transaction

Figure 6.2: Payment Response Time Distribution



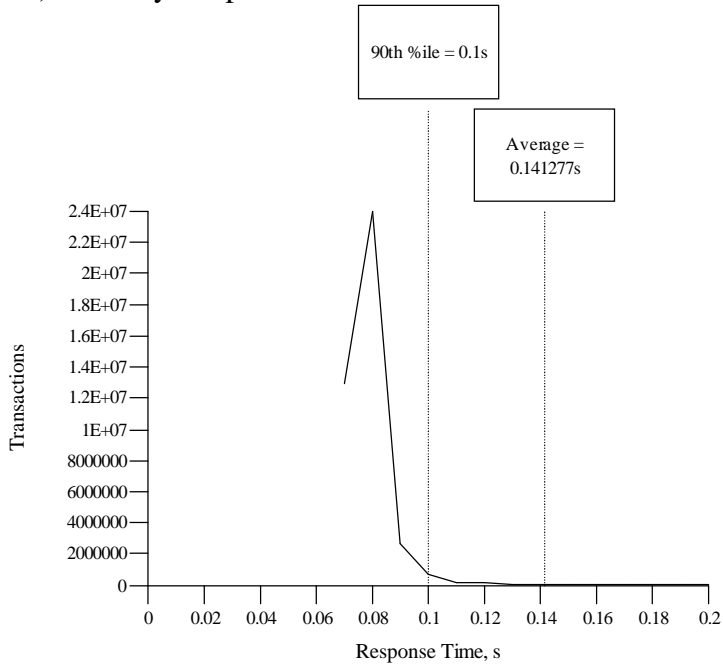
Response time frequency distribution for Payment transaction

Figure 6.3: Order Status Response Time Distribution



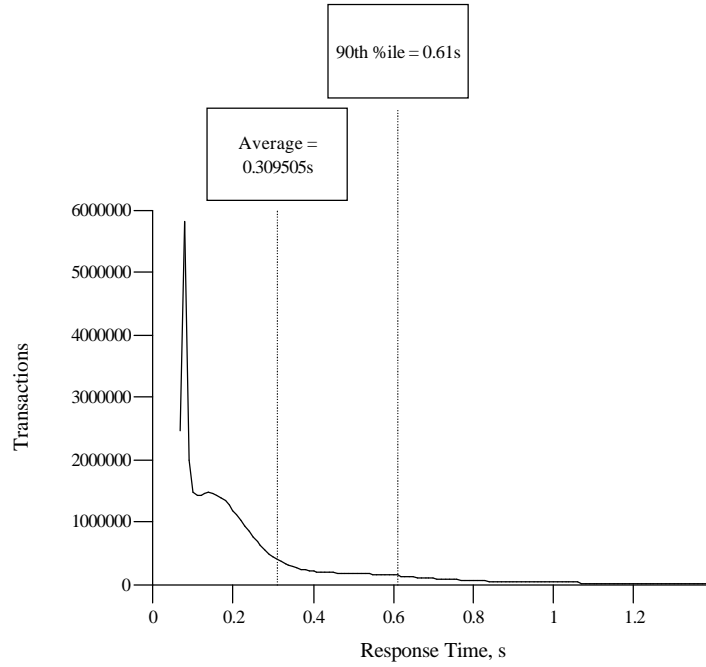
Response time frequency distribution for Order Status transaction

Figure 6.4: (Interactive) Delivery Response Time Distribution



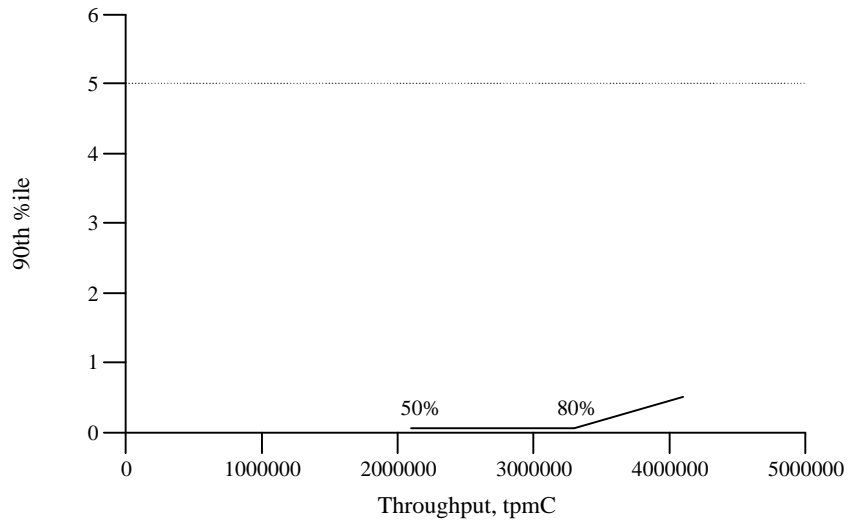
Response time frequency distribution for Delivery transaction

Figure 6.5: Stock Level Response Time Distribution



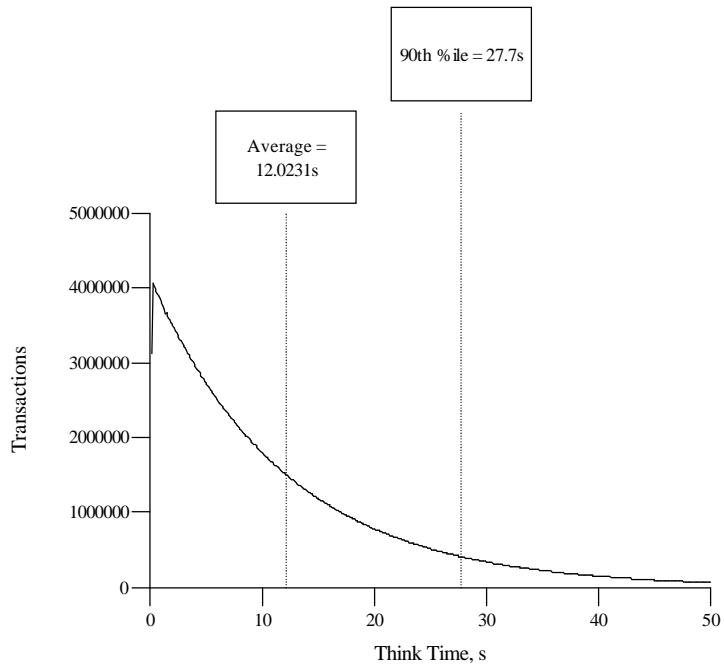
Response time frequency distribution for Stock Level transaction

Figure 6.6: Response Time Versus Throughput



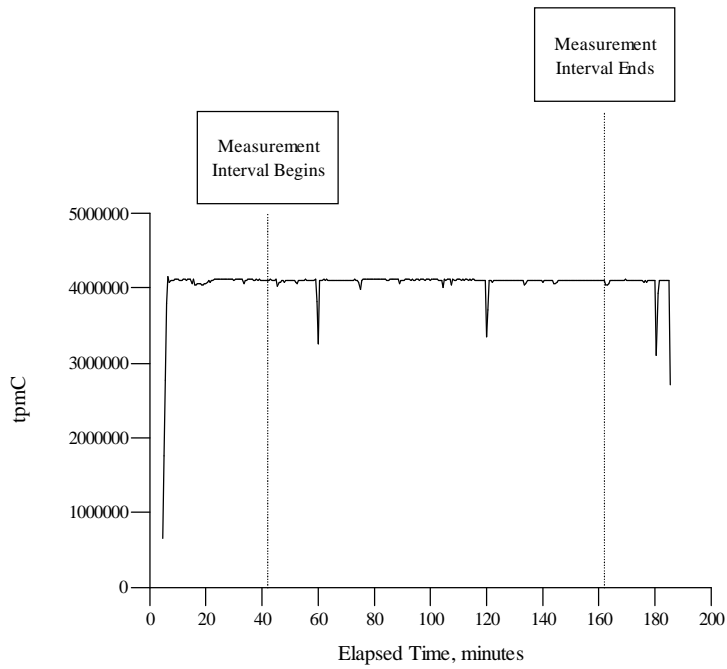
New Order response time versus Throughput

Figure 6.7: New Order Think Time Distribution



Think time frequency distribution for New Order transaction

Figure 6.8: Throughput Versus Elapsed Time



Throughput of the New-Order transaction versus elapsed time

6.5 Steady State Determination

The method used to determine that the SUT had reached a steady state prior to commencing the measurement interval must be disclosed.

Synchronization techniques employed in the benchmark process ensure that all emulated users are logged into the client and have opened the application before submitting transactions. Once all users are connected, each pauses a random amount of time before submitting transactions. The pause time distribution is controlled by a benchmark input parameter. The ramp-up interval is discernible in the graph of throughput over time. The data reduction also tracks the user load and indicates the point in time at which all users have submitted at least one transaction. The throughput is observed to be steady within the systematic and statistical variability of the measurement after all users are submitting transactions. A checkpoint is initiated upon the end of ramp-up.

6.6 Work Performed During Steady State

A description of how the work normally performed during a sustained test (for example checkpointing, writing redo/undo log records, etc.), actually occurred during the measurement interval must be reported.

Modified database buffers migrated to disk on a least-recently-used basis independent of transaction commits. In addition, every block modification was protected by redo log records. These redo log records were written to the redo log buffer (in memory) and were flushed to a redo log file on disk either when the transaction committed or when the redo log buffer became full. However, due to the rapid commit during this benchmark, the redo log buffer was always flushed by a commit before it became full. Also, because many transactions were committing in a short period of time, a single flush of the redo log buffer resulted in many transactions' redo log data being written to disk. This is called group commit.

6.6.1 Checkpoint

During an Oracle Database 10g Release 2 Enterprise Edition with Partitioning checkpoint, all modified blocks in the shared buffer cache which had not been written to disk since the last checkpoint are written to disk.

6.6.2 Checkpoint Conditions

Oracle Database 10g Release 2 Enterprise Edition with Partitioning performs a checkpoint for the following conditions:

1. A redo log switch occurs.
2. The amount of data written to a redo log reaches the `log_checkpoint_interval`
3. The amount of time since the last checkpoint reaches the `log_checkpoint_timeout`.

6.6.3 Checkpoint Implementation

The first method listed above, i.e., a log switch when the redo log file filled up, was used to cause checkpoints. After the initial checkpoint, a log switch was performed every 29.45 minutes in average. All checkpoint intervals were less than 30 minutes.

6.6.4 Serializable Transactions

Oracle supports serializable transaction isolation in full compliance with the SQL92 and TPC-C requirements. This is implemented by extending the multiversion concurrency control mechanism long supported by Oracle.

Oracle queries take no read locks and see only data committed as of the beginning of the query's execution. This means that readers and writers coexist without blocking one another, providing a high degree of concurrency and

consistency. While this mode does prevent reading dirty data, Oracle's default isolation level also permits a transaction that issues a query twice to see non-repeatable reads and phantoms, as defined in SQL92 and TPC-C. Beginning with Oracle7 release 7.3, a transaction may request a high degree of isolation with the command SET TRANSACTION ISOLATION LEVEL SERIALIZABLE, as defined in SQL92. This transaction mode prevents read/write and write/write conflicts that would cause serializability failures. A session can establish this mode as its default mode, so the SET TRANSACTION command need not be issued in each transaction.

Oracle implements SERIALIZABLE mode by extending of the scope of read consistency from the individual query to the entire transaction. Instead of limiting a query to data committed at the time a query begins, in a serializable transaction all queries see data as of the beginning of the transaction. Thus, a serializable transaction sees a fixed snapshot of the database, established as of the beginning of the transaction.

All reads within a serializable transaction see only committed data as of the start of that transaction, plus new updates done by the transaction itself. All reads by a serializable transaction are therefore repeatable, as the transaction will access prior versions of data changed (or deleted) by other transactions after the start of the serializable transaction. This behavior also results in phantom protection since new rows created by other transactions will be invisible to the serializable transaction.

To ensure proper isolation, a serializable transaction cannot modify rows that were changed by other transactions after the beginning of the serializable transaction. If a serializable transaction attempts to update (or delete) a row previously changed by another transaction (serializable or not) since the beginning of the serializable transaction, the update (or delete) statement will fail with error ORA-08177: "Can't serialize access", and the statement will rollback.

SET TRANSACTION ISOLATION

LEVEL SERIALIZABLE;

SELECT ...

SELECT...

UPDATE...

IF "Can't serialize access"

THEN ROLLBACK; LOOP and retry

ELSE COMMIT;

When a serializable transaction fails with this error, the application may either commit the work executed to that point, execute additional (but different) statements, or rollback the entire transaction. Repeated attempts to execute the same statement will always fail with the error "Can't serialize access", unless the other transaction has rolled back and released its lock. This error and these recovery options are similar to the treatment of deadlocks in systems that use read locks to ensure serializable execution. In both cases, conflicts between transactions cannot be resolved unless one of the transactions rolls back and restarts or commits without re-executing the statement receiving the error.

6.7 Measurement Period Duration

A statement of the duration of the measurement interval for the reported Maximum Qualified Throughput (tpmC_®) must be included.

The measurement interval was 120 minutes.

6.8 Regulation of Transaction Mix

The method of regulation of the transaction mix (e.g., card decks or weighted random distribution) must be described. If weighted distribution is used and the RTE adjusts the weights associated with each transaction type, the maximum adjustments to the weight from the initial value must be disclosed.

The weighted selection method of *Clause 5.2.4.1* was used. The weights were not adjusted during the run.

6.9 Transaction Mix

The percentage of the total mix for each transaction type must be disclosed.

Table 6.5: Transaction Mix

Type	Percentage
New Order	44.95%
Payment	43.01%
Order Status	4.01%
Delivery	4.02%
Stock Level	4.010%

6.10 Transaction Statistics

The percentage of New-Order transactions rolled back as a result of invalid item number must be disclosed. The average number of order-lines entered per New-Order transaction must be disclosed. The percentage of remote order-lines entered per New-Order transaction must be disclosed. The percentage of remote Payment transactions must be disclosed. The percentage of customer selections by customer last name in the Payment and Order-Status transactions must be disclosed. The percentage of Delivery transactions skipped due to there being fewer than necessary orders in the New-Order table must be disclosed.

See Table 3.1

6.11 Checkpoint Count and Location

The number of checkpoints in the measurement interval, the time in seconds from the start of the measurement interval to the first checkpoint, and the Checkpoint Interval must be disclosed.

Two checkpoints were completed before the measurement interval began. There were four checkpoints completed within the measurement interval. The first of those checkpoints (#2) starts approximately 2 minutes 45 seconds into the measurement interval. The average checkpoint interval is 29 minutes, 27 seconds and a checkpoint during the MI lasts approximately 26 minutes, 31 seconds.

The run started at 11:22:12. The Measurement Interval was 12:04:12 to 14:04:12

The checkpoints during this run were:

	Start	End		
Checkpoint	time	time	Duration	
	11:22:12			run starts
#0		11:34:37		
#1	11:37:33	12:04:03	26:30:00	
	12:04:12			measurement starts
#2	12:06:57	12:33:26	26:29:00	
#3	12:36:22	13:02:56	26:34:00	
#4	13:05:52	13:32:23	26:31:00	
#5	13:35:18	14:01:48	26:30:00	
		14:04:12		measurement ends
#6	14:04:45			

7 Clause 6: SUT, Driver, and Communication Definition Related Items

7.1 RTE Description

If the RTE is commercially available, then its inputs must be specified. Otherwise, a description must be supplied of what inputs (e.g., scripts) to the RTE had been used. The RTE input parameters, code fragments, functions, et cetera used to generate each transaction input field must be disclosed. Comment: The intent is to demonstrate the RTE was configured to generate transaction input data as specified in Clause 2.

The RTE (Remote Terminal Emulator) on the driver system was developed at Hewlett-Packard and is not commercially available. Appendix D lists RTE input parameters and code fragments used to generate each transaction input field.

For this instance of the TPC-C benchmark, 72 drivers and 95 clients were used. The drivers emulated users logged in to the clients. An overview of the benchmark software on the drivers, clients and server is shown in Figure 7.1.

The benchmark is started with the **run** command on the driver system. **Run** controls the overall execution of the benchmark. After reading a configuration file, **run** starts TUXEDO on the client, collects pre-benchmark audit information and inserts a timestamp into a database audit table. When all the initial steps are completed, **run** invokes another program, **driver**, to start the benchmark. As the benchmark completes, **run** shuts down TUXEDO and collects the benchmark results into a single location.

Driver is the heart of the benchmark software. It simulates users as they log in, execute transactions and view results. **Driver** collects response times for each transaction and saves them in a file for future analysis.

Qualify is the post-processing analysis program. It produces the numerical summaries and histograms needed for the disclosure report.

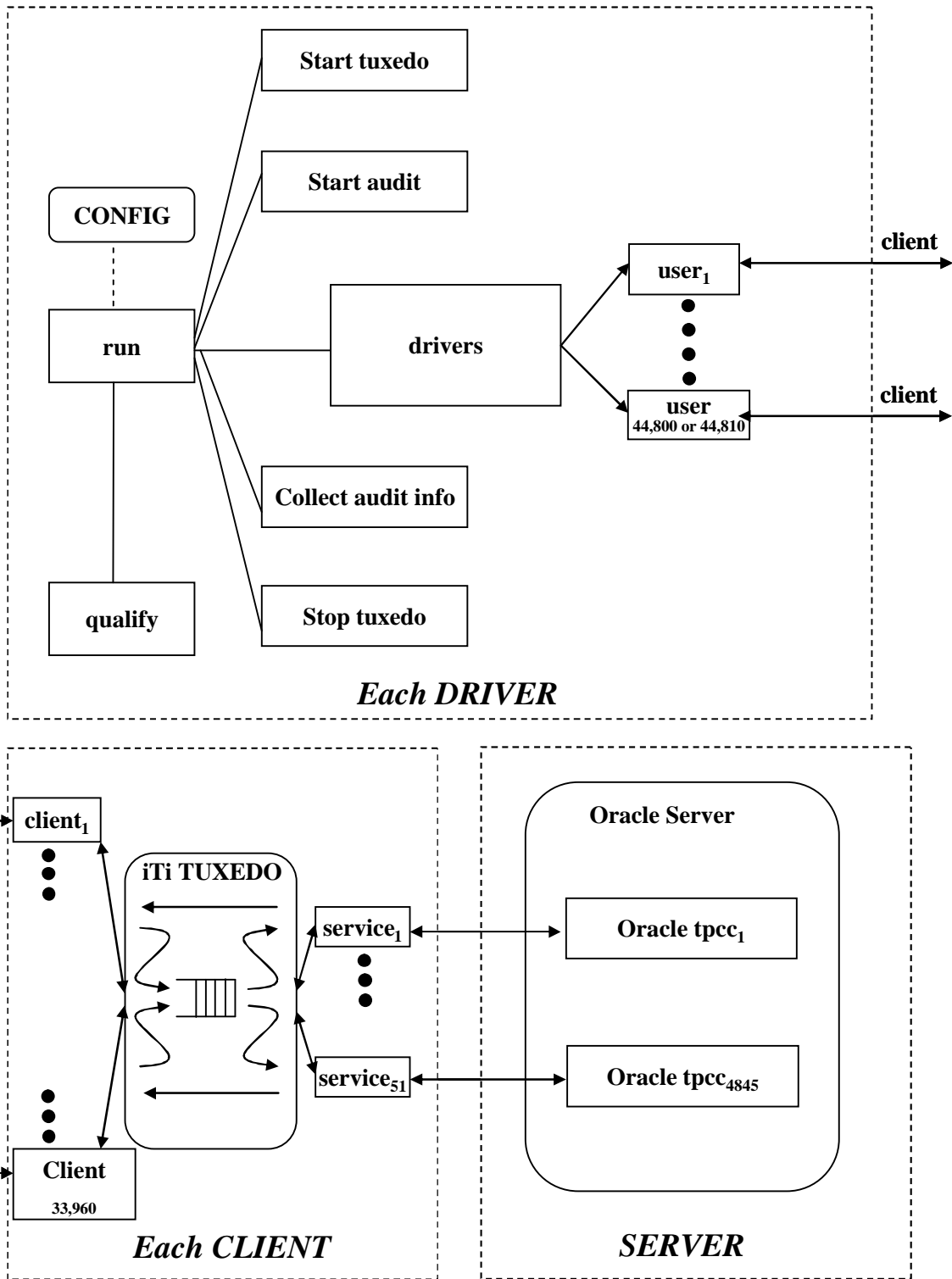


Figure 7.1: Benchmark Software

7.2 Lost Connections

No terminal connections were lost during the measurement interval.

7.3 Emulated Components

It must be demonstrated that the functionality and performance of the components being emulated in the Driver System are equivalent to the priced system.

In the benchmark configuration, the 3,226,200 simulated workstations connected to the clients over 1 1000BT lan through one hp procurve switch. In the priced configuration, the 3,226,200 workstations would connect to the clients via a combination of hubs and switches which eventually mutipexed down to 1 1000BT lan.

7.4 Functional Diagrams

A complete functional diagram of both the benchmark and the configuration of the proposed (target) system must be disclosed. A detailed list of all hardware and software functionality being performed on the Driver System and its interface to the SUT must be disclosed.

Figures 1.1 and 1.2 (in Chapter 1) show functional diagrams of the benchmark and configured systems. A description of the RTE and benchmark software is provided above.

7.5 Networks

The network configuration of both the tested and proposed services which are being represented and a thorough explanation of exactly which parts are being replaced with the Driver System must be disclosed.

In both the tested and priced configurations, each client is either connected to one of 4 ProCurve 2724 switches, or directly connected a ProCurve CL3400 switch. The 2724 switches are then connected to the two CL3400 switches. The CL3400 switches connect to the SUT.

The bandwidth of the networks used in the tested/priced configurations must be disclosed.

The 1000 Base-T and 10gigabit ethernet local area networks (LAN) used with bandwidths of 1000 megabits per second and 10,000 megabits per second.

7.6 Client Substitution

No client substitution was used.

8 Clause 7: Pricing Related Items

8.1 System Pricing

A detailed list of hardware and software used in the priced system must be reported. Each item must have vendor part number, description, and release/revision level, and either general availability status or committed delivery data. If package-pricing is used contents of the package must be disclosed. Pricing source(s) and effective date(s) of price(s) must also be reported.

The total 3-year price of the entire configuration must be reported including: hardware, software, and maintenance charges. Separate component pricing is recommended. The basis of all discounts used must be disclosed.

Each priced configuration consists of an integrated system package, additional options, and components. Prices for all Hewlett-Packard products that are not provided by a third party quote are HP's US list prices. A one (1) year warranty is standard with all Hewlett-Packard products.

8.2 Support Pricing

The three year support pricing for Hewlett-Packard products is based on twenty-four (24) months of monthly support costs; thirty-six (36) months minus the twelve month warranty period. The Oracle Corporation support pricing is based on thirty-six (36) months of monthly support costs. The following support products were priced in the benchmark:

- HP four-hour on-site repair hardware support,
- HP telephone support for software and updates
- Oracle Corporation Standard Technical Support and,
- BEA TUXEDO Standard Technical Support

8.2.1 HP Hardware Support

HP's on-site support for hardware provides service 24 hour, seven day support.

8.2.2 HP Software Support

HP Software Support provides the following:

- Access to the HP Response Centers for fault isolation and problem solving assistance,
- Guaranteed two (2) hour call return, immediate response for critical calls,
- Electronic access to product and support information,
- Electronic access to software patches,
- Right-to-use and copy software updates.

8.3 Oracle Corporation Standard Technical Support

Oracle Corporation Standard Technical Support includes:

Product updates,

- A regular technical publication,
- Unlimited, toll-free telephone service to assist in product installation, syntax, and usage that is available 24 hours, seven days a week.

8.4 Availability

The committed delivery date for general availability (availability date) of products used in the price calculation must be reported. When the priced system includes products with different availability dates, the reported availability date for the priced system must be the date at which all components are committed to be available.

see below

8.5 Priced System Configuration

The hardware, software, and support/maintenance products priced in this benchmark are detailed on page v. Patch 5932060 needs to be applied. The patch is available at metalink.oracle.com.

8.6 Throughput, Price/Performance, and Availability Date

A statement of the measured tpmC[®] as well as the respective calculations for the 3-year pricing, price/performance (price/tpmC[®]).

For Throughput and Price/Performance, please see page iv and v. The Price/Performance calculation spreadsheet appears on page v.

Description	Source	Orderability Date	Availability Date
HP-UX 11i v3 Firmware for HP StorageWorks MSA*		N/A	April 1, 2007
HP-UX 11i, v3 SW-Patches for Foundation Operating Environment*		N/A	August 6, 2007
Oracle Database 10g Release 2 (10.2.0.3) Enterprise Edition with Partitioning	http://www.oracle.com	August 6, 2007	August 6, 2007

For HP pricing verification, please contact HP Unix Sales Development at 408-447-2320

For Oracle pricing verification, please contact Ravi Rajamani, Ravi.Rajamani@oracle.com, 650-506-5776

*Firmware and patches can be downloaded free of charge from HP's website after specified date

9 Clause 9: Audit Related Items

9.1 Auditor's Report

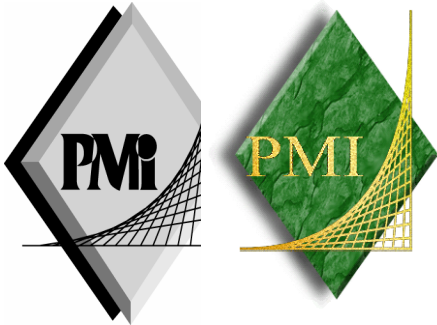
If the benchmark has been independently audited, then the auditor's name, address, phone number, and a brief audit summary report indicating compliance must be included in the Full Disclosure Report. A statement should be included, specifying when the complete audit report will become available and who to contact in order to obtain a copy.

If audited, the auditor's attestation letter must be made readily available to the public as part of the Full Disclosure Report, but a detailed report from the auditor is not required.

This implementation of the TPC Benchmark[®] C on the HP Integrity Superdome - Itanium2/1.6GHz/24MB iL3 - 64p/128c was audited by Lorna Livingtree for Performance Metrics, Inc..

Lorna Livingtree
Performance Metrics, Inc.
P.O. Box 984
Klamath, CA 95548
U.S.A.
Phone: (707) 482-0523
Fax:(707) 482-1352

The attestation letter is shown on the following pages.



PERFORMANCE METRICS INC.
TPC Certified Auditors

August 13, 2007

Mr. Curt G. Thiem,
 HP-UX Performance Manager
 Mailstop 4220
 Hewlett-Packard Company
 19447 Pruneridge Avenue
 Cupertino, CA 95014-0683

I have verified by both on-site and by remote the TPC Benchmark™ C for the following configuration:

Platform: HP Integrity Superdome
Database Manager: Oracle Database 10g Release 2 Enterprise Edition with Partitioning
Operating System: HP-UX 11i v3
Transaction Monitor: BEA Tuxedo 8.1

System Under Test: HP Integrity Superdome with:				
CPU's	Memory	Disks (total)	90% Response	TpmC
64 Itanium2 @1.6GHz	2,048 GB	4424 @36GB 2632@ 72GB	0.51	4,092,799
95 clients: HP Integrity rx1620 each with:				
2 Itanium2 @1.3 GHz	8 GB	1 @ 36GB	Na	Na

In my opinion, these performance results were produced in compliance with the TPC requirements for the benchmark. The following attributes of the benchmark were given special attention:

- **The transactions were correctly implemented.**
- **The database files were properly sized.**
- **The database was properly scaled with 349,440 warehouses, 322,620 of which were active during the measured interval.**
- **The ACID properties were successfully demonstrated.**
- **Data loss durability was demonstrated on the full SUT configuration with a database properly populated for 322,620 warehouses.**
- **Input data was generated according to the specified percentages.**
- **Eight hours of mirrored log space was present on the tested system.**
- **Eight hours of growth space for the dynamic tables was present on the tested system.**
- **The data for the 60 days space calculation was verified.**
- **The steady state portion of the test was 120 minutes.**
- **One checkpoint was taken in steady state before the measured interval opened.**
- **Four checkpoints were completed inside the measured interval.**
- **The system pricing was checked for major components and maintenance.**
- **Third party quotes were verified for compliance.**

Auditor Notes:

Additional disks were attached to the server and used for backup. I verified that these disks were unused during the performance run. There were 224 disks of size 72GB priced as a substitute for 224 disks of size 36GB. Performance data demonstrated that the substituted disks perform as good or better than the 36GB drives.

Sincerely,



**Lorna Livingtree
Auditor**

10 Report Availability

Requests for this TPC Benchmark C Full Disclosure Report should be sent to:

Transaction Processing Performance Council
c/o Tpc Administrator
P.O. Box 29920
San Francisco, CA 94129-0920

or your local Hewlett-Packard sales office.

Appendix A Client/Server Source

This appendix contains the source and makefiles for all client and server programs.

A.1 Client Front-End

client/client.c

```
*****
@(#) Version: A.10.10 $Date: 2005/04/11 10:10:29 $

(c) Copyright 1996, Hewlett-Packard Company, all rights reserved.
*****
History
941101 JVM Fixed login screen to detect broken connection (used to loop)
941013 JVM Added audit strings to the login form
941013 VM modified the getfield procedure to add digit and char check
according to the field type.
941014 VM added the status_msg routine to display transaction results.
941015 VM added zip routine to format zip codes and phone routine
to format phone numbers.
*****
#include <stdlib.h>
#include <stdio.h>
#include <signal.h>
#include <sys/types.h>
#include <sys/socket.h>
#include <netinet/tcp.h>
#include <netinet/in.h>
#include <netdb.h>
#include <fcntl.h>
#include <errno.h>
#include <pthread.h>
#include <sched.h>
#include <math.h>

#include "key_chars.h"
#include "tpcc.h"
#include "tux_transaction.h"
#include "iobuf.h"

/* Macro to define an I/O buffer of x characters, initialized to empty */
#define define_iobuf(name, size) \
char name##_data[size]; \
iobuf name[1]

#define init_iobuf(name, size, _ifid, _ofd) \
name->ifid = _ifid; \
name->ofd = _ofd; \
name->beg = name##_data; \
name->end = name##_data + size; \
name->max = name##_data + size; \
name->cur = name##_data;

/*
 * Input/Output buffers + screen buffers
 */

#define INPUT_BUF_SIZE 1024
#define OUTPUT_BUF_SIZE 4096
#define NEWORDER_FORM_SIZE 900
#define PAYMENT_FORM_SIZE 400
#define ORDSTAT_FORM_SIZE 300
#define DELIVERY_FORM_SIZE 300
#define STOCKLEV_FORM_SIZE 300

/* Number of Threads per Tuxedo Context */
#define MAX_THREADS_PER_CONTEXT 16

/* Maximum number of threads per server */
#ifdef _hpux
#define MAX_USERS_PER_PROCESS 1000
int number_of_servers = 15; /* default number of servers to spawn */
#else
#define MAX_USERS_PER_PROCESS 240 /* make it less than threads-per-process on
Linux:250 */
int number_of_servers = 64;
```

```
#endif

/* Process local only */
long tux_context; /* Tuxedo context to use */
int port_number = 11000; /* address to listen on */
int user_connections = 0; /* number of current connections */
pthread_t user_ids[MAX_USERS_PER_PROCESS] = {0}; /* thread ids spawned per
server */

struct thread_data {
int fd; /* Stream file descriptor */
long tux_context; /* Tuxedo context to use */
};
typedef struct thread_data thread_data;

/*
 * Prototype definitions
 */

static int next_field(int current, int key, int max);
static int neworder(iobuf *in_buf, iobuf *out_buf, iobuf *neworder_form, void **data_ptr,
neworder_trans *t, ID warehouse);
static int neworder_read(iobuf *in_buf, iobuf *out_buf, neworder_trans *t, ID warehouse);
static void neworder_write(iobuf *out_buf, neworder_trans *t);
static void neworder_setup(iobuf *neworder_form, ID warehouse);
static int payment(iobuf *in_buf, iobuf *out_buf, iobuf *payment_form, void **data_ptr,
payment_trans *t, ID warehouse);
static void payment_setup(iobuf *payment_form, ID warehouse);
static int payment_read(iobuf *in_buf, iobuf *out_buf, payment_trans *t, ID warehouse);
static void payment_write(iobuf *out_buf, payment_trans *t);
static int ordstat(iobuf *in_buf, iobuf *out_buf, iobuf *ordstat_form, void **data_ptr,
ordstat_trans *t, ID warehouse);
static void ordstat_setup(iobuf *ordstat_form, ID warehouse);
static int ordstat_read(iobuf *in_buf, iobuf *out_buf, ordstat_trans *t, ID warehouse);
static void ordstat_write(iobuf *out_buf, ordstat_trans *t);
static int delivery(iobuf *in_buf, iobuf *out_buf, iobuf *delivery_form, void *data_ptr,
delivery_trans *t, ID warehouse);
static void delivery_setup(iobuf *delivery_form, ID warehouse);
static int delivery_read(iobuf *in_buf, iobuf *out_buf, delivery_trans *t, ID warehouse);
static void delivery_write(iobuf *out_buf, delivery_trans *t);
static int stocklev(iobuf *in_buf, iobuf *out_buf, iobuf *stocklev_form, void **data_ptr,
stocklev_trans *t, ID warehouse, ID district);
static void stocklev_setup(iobuf *stocklev_form, ID warehouse, ID district);
static int stocklev_read(iobuf *in_buf, iobuf *out_buf, stocklev_trans *t, ID warehouse, ID
district);
static void stocklev_write(iobuf *out_buf, stocklev_trans *t);
static int valid_char(int key, FIELD_TYPE ftype);
static int getfield(int row, int col, char buf[], int width, FIELD_TYPE ftype,
iobuf *in_buf, iobuf *out_buf);
static int read_text(int row, int col, char *s, int width, iobuf *in_buf,
iobuf *out_buf);
static int read_money(int row, int col, double *m, int width, iobuf *in_buf,
iobuf *out_buf);
static int read_number(int row, int col, int *n, int width, iobuf *in_buf,
iobuf *out_buf);

static void clear_screen(iobuf *out_buf);
static void position(iobuf *out_buf, int row, int col);
static void trigger(iobuf *out_buf);
static void trigger2(iobuf *out_buf);
static void status(iobuf *out_buf, int row, int col, int status);
static void blanks(iobuf *out_buf, int row, int col, int len);
static void empty(iobuf *out_buf, int row, int col, int len);
static void zip(iobuf *out_buf, int row, int col, char *str);
static void phone(iobuf *out_buf, int row, int col, char *str);
static void text(iobuf *out_buf, int row, int col, char str[]);
static void long_text(iobuf *out_buf, int row, int col, char *str, int width);
static void money(iobuf *out_buf, int row, int col, double x, int width);
static void date_only(iobuf *out_buf, int row, int col, char *date_str);
static void date(iobuf *out_buf, int row, int col, char *date_str);
static void real(iobuf *out_buf, int row, int col, double x, int width, int dec);
static void number(iobuf *out_buf, int row, int col, int n, int width);
static void cleanup(iobuf *out_buf);
static void msgline(iobuf *out_buf, char *str);
static int menu_read(iobuf *in_buf, iobuf *out_buf);
static void menu_setup(iobuf *out_buf);
static int login(iobuf *in_buf, iobuf *out_buf, ID *warehouse, ID *district);

void *
client_main(void *arg)
{
/*
 * variables set up during initialization
 */
iobuf *in_buf;
iobuf *out_buf;
void **data_ptr; /* where data gets copied to Tuxedo */
ID warehouse;
ID district;
int key;
int user;

define_iobuf(output_stuff, OUTPUT_BUF_SIZE);
define_iobuf(input_stuff, INPUT_BUF_SIZE);
define_iobuf(payment_form, PAYMENT_FORM_SIZE);
define_iobuf(neworder_form, NEWORDER_FORM_SIZE);
define_iobuf(ordstat_form, ORDSTAT_FORM_SIZE);
```



```

define_iobuf(delivery_form, DELIVERY_FORM_SIZE);
define_iobuf(stocklev_form, STOCKLEV_FORM_SIZE);

/* a generic transaction variable. */
generic_trans generic_transaction;

thread_data *td = (thread_data *)arg;
generic_trans *trans=&generic_transaction;

/* setup Tuxedo Context */
data_ptr = (void *)thread_transaction_begin(td->tux_context);

/* Initialize the forms */
init_iobuf(neworder_form, NEWORDER_FORM_SIZE, td->fd, td->fd);
init_iobuf(payment_form, PAYMENT_FORM_SIZE, td->fd, td->fd);
init_iobuf(ordstat_form, ORDSTAT_FORM_SIZE, td->fd, td->fd);
init_iobuf(delivery_form, DELIVERY_FORM_SIZE, td->fd, td->fd);
init_iobuf(stocklev_form, STOCKLEV_FORM_SIZE, td->fd, td->fd);

/* Initialize input/output */
init_iobuf(output_stuff, OUTPUT_BUF_SIZE, td->fd, td->fd);
init_iobuf(input_stuff, INPUT_BUF_SIZE, td->fd, td->fd);

/* Setup Input and Output buffers */
in_buf = input_stuff;
out_buf = output_stuff;

/* get the user, warehouse and district numbers */
warehouse = EMPTY_NUM;
district = EMPTY_NUM;
key = login(in_buf, out_buf, &warehouse, &district);
user = warehouse*DIST_PER_WARE + district + 1;

/* set up the forms */
menu_setup(out_buf);
neworder_setup(neworder_form, warehouse);
payment_setup(payment_form, warehouse);
ordstat_setup(ordstat_form, warehouse);
delivery_setup(delivery_form, warehouse);
stocklev_setup(stocklev_form, warehouse, district);

/* connect to the delivery queue */
delivery_init(user);

/* repeat until done */
while (key != '9' && key != EOF) {

    /* get the menu choice */
    key = menu_read(in_buf, out_buf);

    /* process according to the choice */
    switch(key) {
        case '1':
            key = neworder(in_buf, out_buf,
                &data_ptr,
                &trans->neworder, warehouse);
            break;
        case '2':
            key = payment(in_buf, out_buf,
                &data_ptr,
                &trans->payment, warehouse);
            break;
        case '3':
            key = ordstat(in_buf, out_buf, ordstat_form,
                &data_ptr,
                &trans->ordstat, warehouse);
            break;
        case '4':
            /* Note delivery does not modify data_ptr, thus
            address is not taken */
            key = delivery(in_buf, out_buf, delivery_form,
                data_ptr, &trans-
                >delivery, warehouse);
            break;
        case '5':
            key = stocklev(in_buf, out_buf, stocklev_form,
                &data_ptr,
                &trans->stocklev, warehouse, district);
            break;
        case EOF:
            break;
        case '9':
            break;
        default:
            msgline(out_buf, "Please enter a valid menu
            choice");
            break;
    }
}

/* done */
cleanup(out_buf);

/* Close socket */
close(td->fd);

/* Exit Thread */
pthread_exit(NULL);
}

/*****
*****
Neworder form processing
*****
*****
static int
neworder(iobuf *in_buf, iobuf *out_buf, iobuf *neworder_form,
        void **data_ptr, neworder_trans *t, ID warehouse)
{
    int key;
    display(neworder_form);
    key = neworder_read(in_buf, out_buf, t, warehouse);
    if (key != ENTER) return key;
    neworder_transaction(data_ptr, t);
    neworder_write(out_buf, t);
    return key;
}

static int
neworder_read(iobuf *in_buf, iobuf *out_buf, neworder_trans *t, ID warehouse)
{
    int i;
    int field;
    int key;
    int ol;
        int ol_count;
        int all_local;
        int move_slot;

    /* Our warehouse number is fixed */
    t->W_ID = warehouse;
    t->D_ID = EMPTY_NUM;

    /* assume nothing set yet */
    t->C_ID = EMPTY_NUM;
    for (i=0; i<15; i++) {
        t->item[i].OL_I_ID = EMPTY_NUM;
        t->item[i].OL_QUANTITY = EMPTY_NUM;
        t->item[i].OL_SUPPLY_W_ID = EMPTY_NUM;
    }

    /* Process fields until done */
    for (field = 1; field > 0; field = next_field(field, key, 47)) {
        retry: switch (field) {
            case 1: key = read_number(4, 29, &t->D_ID, 2, in_buf, out_buf);
                    break;
            case 2: key = read_number(5, 12, &t->C_ID, 4, in_buf, out_buf);
                    break;
            case 3: case 6: case 9: case 12: case 15:
            case 18: case 21: case 24: case 27: case 30:
            case 33: case 36: case 39: case 42: case 45:
                    ol = (field - 3) / 3;
                    key = read_number(9+ol, 2, &t->item[ol].OL_SUPPLY_W_ID, 6, in_buf,
                    out_buf);
                    break;
            case 4: case 7: case 10: case 13: case 16:
            case 19: case 22: case 25: case 28: case 31:
            case 34: case 37: case 40: case 43: case 46:
                    ol = (field - 3) / 3;
                    key = read_number(9+ol, 10, &t->item[ol].OL_I_ID, 6, in_buf, out_buf);
                    break;
            case 5: case 8: case 11: case 14: case 17:
            case 20: case 23: case 26: case 29: case 32:
            case 35: case 38: case 41: case 44: case 47:
                    ol = (field - 3) / 3;
                    key = read_number(9+ol, 45, &t->item[ol].OL_QUANTITY, 2, in_buf, out_buf);
                    break;
        }
    }

    /* abort the screen if requested */
    if (key != ENTER) {
        return key;
    }
}

```

```

/* make sure all necessary fields are filled in */
if (t->D_ID == EMPTY_NUM)
    {field=1; msgline(out_buf, "Please specify district"); goto retry;}
if (t->C_ID == EMPTY_NUM)
    {field=2; msgline(out_buf, "Please specify customer id"); goto retry;}

/* calculate how many items were entered */
    ol_count = 0;
    all_local = 1;
    move_slot = -1;
for (i=0; i < 15; i++) {
    if ((t->item[i].OL_I_ID == EMPTY_NUM) &&
        (t->item[i].OL_SUPPLY_W_ID == EMPTY_NUM) &&
        (t->item[i].OL_QUANTITY == EMPTY_NUM)) {
        /* All are clear, so no item */
        if (move_slot == -1) {
            move_slot = i;
        } else {
            /* this is potentially an order line, so check it
out */
            if (t->item[i].OL_SUPPLY_W_ID == EMPTY_NUM) {
                field=i*3+3;
                msgline(out_buf, "Please enter
supply warehouse");
                goto retry;
            }
            if (t->item[i].OL_I_ID == EMPTY_NUM) {
                field=i*3+4;
                msgline(out_buf, "Please enter
Item id");
                goto retry;
            }
            if (t->item[i].OL_QUANTITY == EMPTY_NUM ||
                t->item[i].OL_QUANTITY <= 0) {
                field=i*3+5;
                msgline(out_buf, "Please enter
quantity > 0");
                goto retry;
            }
            /* It is a complete orderline, so count it */
            ol_count++;

            /* decide if they were all local */
            if (t->item[i].OL_SUPPLY_W_ID != t->W_ID) {
                all_local = 0;
            }

            if (move_slot != -1) {
                /* Move the item up to fill in a
hole */
                t->item[move_slot] = t->item[i];
                move_slot++; /* bump up to the
next slot */
            }
        }
    }

    if (ol_count == 0)
        {field=3; msgline(out_buf, "Please enter at least one orderline"); goto retry;}

    t->O_OL_CNT = ol_count;
    t->all_local = all_local;

    /* display number of order lines */
    number(out_buf, 6, 42, t->O_OL_CNT, 2);

    msgline(out_buf, "");
    flush(out_buf);
    return key;
}

static void
neworder_write(iobuf *out_buf, neworder_trans *t)
{
    int i;
    MONEY amount, total_amount, cost;

    /* Rev. 3.3 error checking: both of the following branches are
    * skipped. We'll go to status and print an error message.
    */

    if (t->status == E_INVALID_ITEM) {
        /* CASE: invalid item, display only these values */
        text(out_buf, 5, 25, t->C_LAST);
        text(out_buf, 5, 52, t->C_CREDIT);
        number(out_buf, 6, 15, t->O_ID, 8);
    } else if (t->status == OK) {
        /* CASE: everything OK, display everything */
        text(out_buf, 5, 25, t->C_LAST);
        text(out_buf, 5, 52, t->C_CREDIT);
        number(out_buf, 6, 15, t->O_ID, 8);
        date(out_buf, 4, 61, t->O_ENTRY_D);

```

```

real(out_buf, 5, 64, t->C_DISCOUNT * 100, 5, 2);
real(out_buf, 6, 59, t->W_TAX*100, 5, 2);
real(out_buf, 6, 74, t->D_TAX*100, 5, 2);

total_amount = 0;
for (i=0; i < t->O_OL_CNT; i++) {

    /* keep track of amount of each line and total */
    amount = t->item[i].L_PRICE * t->item[i].OL_QUANTITY;
    total_amount += amount;

    /* display the item line */
    number(out_buf, 9+i, 2, t->item[i].OL_SUPPLY_W_ID, 6);
    number(out_buf, 9+i, 10, t->item[i].OL_I_ID, 6);
    text(out_buf, 9+i, 19, t->item[i].L_NAME);
    number(out_buf, 9+i, 45, t->item[i].OL_QUANTITY, 2);
    number(out_buf, 9+i, 51, t->item[i].S_QUANTITY, 3);
    position(out_buf, 9+i, 58); pushc(out_buf, t->item[i].brand_generic);
    money(out_buf, 9+i, 62, t->item[i].L_PRICE, 7);
    money(out_buf, 9+i, 71, amount, 8);
}

/* Clear the screen of any empty input fields */
clear_screen(out_buf);

/* display the total cost */
text(out_buf, 24, 63, "Total:");
cost = total_amount * (1 - t->C_DISCOUNT) * (1 + t->W_TAX + t->D_TAX);
money(out_buf, 24, 71, cost, 9);
}

/* display the status message */
status(out_buf, 24, 1, t->status);
}

static void
neworder_setup(iobuf *neworder_form, ID warehouse)
{
    int item;
    iobuf *old;

    /* start with an empty form */
    reset(neworder_form);

    /* clear the iobuf below the menu */
    position(neworder_form, 3, 1);
    clear_screen(neworder_form);

    /* set up all the field labels */
    text(neworder_form, 3, 36, "New Order");
    text(neworder_form, 4, 1, "Warehouse:");
    number(neworder_form, 4, 12, warehouse, 6);
    text(neworder_form, 4, 19, "District:");
    empty(neworder_form, 4, 29, 2);
    text(neworder_form, 4, 55, "Date:");
    text(neworder_form, 5, 1, "Customer:");
    empty(neworder_form, 5, 12, 4);
    text(neworder_form, 5, 19, "Name:");
    text(neworder_form, 5, 44, "Credit:");
    text(neworder_form, 5, 57, "Disc:");
    text(neworder_form, 6, 1, "Order Number:");
    text(neworder_form, 6, 25, "Number of Lines:");
    text(neworder_form, 6, 52, "W_Tax:");
    text(neworder_form, 6, 67, "D_Tax:");
    text(neworder_form, 8, 2, "Supp_W Item_Num Item_Name");
    text(neworder_form, 8, 45, "Qty Stock B/G Price Amount");

    /* display blank fields for each item */
    for (item = 1; item <= 15; item++) {
        empty(neworder_form, 8+item, 2, 6);
        empty(neworder_form, 8+item, 10, 6);
        empty(neworder_form, 8+item, 45, 2);
    }
}

Payment form processing

static int
payment(iobuf *in_buf, iobuf *out_buf, iobuf *payment_form,
        void **data_ptr, payment_trans *t, ID warehouse)
{
    int key;
    display(payment_form);
    key = payment_read(in_buf, out_buf, t, warehouse);
    if (key != ENTER) return key;
    payment_transaction(data_ptr, t);
    payment_write(out_buf, t);
}

```

```

return key;
}

static void
payment_setup(iobuf *payment_form, ID warehouse)
{
    /* start with an empty form */
    reset(payment_form);

    /* clear the iobuf below the menu */
    position(payment_form, 3,1);
    clear_screen(payment_form);

    /* set up all the field labels */
    text(payment_form, 3, 38, "Payment");
    text(payment_form, 4, 1, "Date:");
    text(payment_form, 6, 1, "Warehouse:");
    number(payment_form, 6, 12, warehouse, 6);
    text(payment_form, 6, 42, "District:");
    empty(payment_form, 6, 52, 2);
    text(payment_form, 11, 1, "Customer:");
    empty(payment_form, 11, 11, 4);
    text(payment_form, 11, 17, "Cust-Warehouse:");
    empty(payment_form, 11, 33, 6);
    text(payment_form, 11, 40, "Cust-District:");
    empty(payment_form, 11, 54, 2);
    text(payment_form, 12, 1, "Name:");
    empty(payment_form, 12, 29, 16);
    text(payment_form, 12, 50, "Since:");
    text(payment_form, 13, 50, "Credit:");
    text(payment_form, 14, 50, "%Disc:");
    text(payment_form, 15, 50, "Phone:");
    text(payment_form, 17, 1, "Amount Paid:");
    empty(payment_form, 17, 23, 8);
    text(payment_form, 17, 37, "New Cust-Balance:");
    text(payment_form, 18, 1, "Credit Limit:");
    text(payment_form, 20, 1, "Cust-Data:");
}

static int
payment_read(iobuf *in_buf, iobuf *out_buf, payment_trans *t, ID warehouse)
{
    int i;
    int field;
    int key;

    /* Our warehouse number is fixed */
    t->W_ID = warehouse;
    t->C_ID = EMPTY_NUM;
    t->D_ID = EMPTY_NUM;
    t->C_W_ID = EMPTY_NUM;
    t->C_D_ID = EMPTY_NUM;
    t->H_AMOUNT = EMPTY_FLT;
    t->C_LAST[0] = '\0';

    /* Process fields until done */
    for (field = 1; field > 0; field = next_field(field, key, 6)) {
        retry: switch (field) {

            case 1: key = read_number(6, 52, &t->D_ID, 2, in_buf, out_buf);
                    break;

            case 2:
                /* if last name specified, skip this field */
                if (t->C_LAST[0] != '\0')
                    break;

                /* read in the customer id */
                key = read_number(11, 11, &t->C_ID, 4, in_buf, out_buf);

                /* if specified, don't allow last name to be entered */
                if (t->C_ID != EMPTY_NUM)
                {
                    blanks(out_buf, 12, 29, 16);
                    t->C_LAST[0] = '\0';
                }

                /* refresh the C_LAST underlines, if possibly needed */
                else if (t->C_LAST[0] == '\0')
                    empty(out_buf, 12, 29, 16);
                break;

            case 3: key = read_number(11, 33, &t->C_W_ID, 6, in_buf, out_buf);
                    break;

            case 4: key = read_number(11, 55, &t->C_D_ID, 2, in_buf, out_buf);
                    break;

            case 5:
                /* skip this field if C_ID was already specified */
                if (t->C_ID != EMPTY_NUM)
                    break;

                /* read in the customer last name */
                key = read_text(12, 29, t->C_LAST, 16, in_buf, out_buf);

                /* if specified, don't allow c_id to be entered */
                if (t->C_LAST[0] != '\0')
                {
                    blanks(out_buf, 11, 11, 4);
                    t->C_ID = EMPTY_NUM;
                }

                /* refresh the C_ID underlines, if possibly needed */
                else if (t->C_ID == EMPTY_NUM)
                    empty(out_buf, 11, 11, 4);
                break;

            case 6: key = read_money(17, 23, &t->H_AMOUNT, 8, in_buf, out_buf);
                    break;
                }

        /* if Aborted, then done */
        if (key != ENTER) {
            return key;
        }

        /* Make sure all the fields were entered */
        if (t->D_ID == EMPTY_NUM)
            {field=1; msgline(out_buf, "Please enter district id"); goto retry;}
        if (t->C_ID == EMPTY_NUM && t->C_LAST[0] == '\0')
            {field=2; msgline(out_buf, "C_ID or C_LAST must be entered"); goto retry;}
        if (t->C_W_ID == EMPTY_NUM)
            {field=3; msgline(out_buf, "Please enter customer's warehouse"); goto retry;}
        if (t->C_D_ID == EMPTY_NUM)
            {field=4; msgline(out_buf, "please enter customer's district"); goto retry;}
        if (t->H_AMOUNT == EMPTY_FLT)
            {field=6; msgline(out_buf, "Please enter payment amount"); goto retry;}
        if (t->H_AMOUNT <= 0)
            {field=6; msgline(out_buf, "Please enter a positive payment"); goto retry;}

        t->byname = (t->C_ID == EMPTY_NUM);
        msgline(out_buf, "");
        flush(out_buf);
        return key;
    }

    static void
    payment_write(iobuf *out_buf, payment_trans *t)
    {
        /* if errors, display a message and quit */
        if (t->status != OK) {
            status(out_buf, 24, 1, t->status);
            return;
        }

        /* display the screen */
        date(out_buf, 4, 7, t->H_DATE);
        text(out_buf, 7, 1, t->W_STREET_1);
        text(out_buf, 7, 42, t->D_STREET_1);
        text(out_buf, 8, 1, t->W_STREET_2);
        text(out_buf, 8, 42, t->D_STREET_2);
        text(out_buf, 9, 1, t->W_CITY);
        text(out_buf, 9, 22, t->W_STATE);
        zip(out_buf, 9, 25, t->W_ZIP);
        text(out_buf, 9, 42, t->D_CITY);
        text(out_buf, 9, 63, t->D_STATE);
        zip(out_buf, 9, 66, t->D_ZIP);
        number(out_buf, 11, 11, t->C_ID, 4);
        text(out_buf, 12, 9, t->C_FIRST);
        text(out_buf, 12, 26, t->C_MIDDLE);
        text(out_buf, 12, 29, t->C_LAST);
        date_only(out_buf, 12, 58, t->C_SINCE);
        text(out_buf, 13, 9, t->C_STREET_1);
        text(out_buf, 13, 58, t->C_CREDIT);
        text(out_buf, 14, 9, t->C_STREET_2);
        real(out_buf, 14, 58, t->C_DISCOUNT*100, 5, 2); /* percentage or fraction? */
        text(out_buf, 15, 9, t->C_CITY);
        text(out_buf, 15, 30, t->C_STATE);
        zip(out_buf, 15, 33, t->C_ZIP);
        phone(out_buf, 15, 58, t->C_PHONE);
        money(out_buf, 17, 17, t->H_AMOUNT, 14);
        money(out_buf, 17, 55, t->C_BALANCE, 15);
        money(out_buf, 18, 17, t->C_CREDIT_LIM, 14);

        /* Display cust data if bad credit. */
        if (t->C_CREDIT[0] == 'B' && t->C_CREDIT[1] == 'C') {
            long_text(out_buf, 20, 12, t->C_DATA, 50);
        }

        trigger2(out_buf);
    }

    /******

```

```

/* ORDDAT form processing */
/*****

static int
ordstat(iobuf *in_buf, iobuf *out_buf, iobuf *ordstat_form,
        void **data_ptr, ordstat_trans *t, ID warehouse)
{
    int key;
    display(ordstat_form);
    key = ordstat_read(in_buf, out_buf, t, warehouse);
    if (key != ENTER) return key;
    ordstat_transaction(data_ptr, t);
    ordstat_write(out_buf, t);
    return key;
}

static void
ordstat_setup(iobuf *ordstat_form, ID warehouse)
{
    /* start with an empty form */
    reset(ordstat_form);

    /* clear the iobuf below the menu */
    position(ordstat_form, 3,1);
    clear_screen(ordstat_form);

    /* set up all the field labels */
    text(ordstat_form, 3, 35, "Order-Status");
    text(ordstat_form, 4, 1, "Warehouse:");
    number(ordstat_form, 4, 12, warehouse, 6);
    text(ordstat_form, 4, 19, "District:");
    empty(ordstat_form, 4, 29, 2);
    text(ordstat_form, 5, 1, "Customer:");
    empty(ordstat_form, 5, 11, 4);
    text(ordstat_form, 5, 18, "Name:");
    empty(ordstat_form, 5, 44, 16);
    text(ordstat_form, 6, 1, "Cust-Balance:");
    text(ordstat_form, 8, 1, "Order-Number");
    text(ordstat_form, 8, 26, "Entry-Date:");
    text(ordstat_form, 8, 60, "Carrier-Number:");
    text(ordstat_form, 9, 1, "Supply-W");
    text(ordstat_form, 9, 14, "Item-Num");
    text(ordstat_form, 9, 25, "Qty");
    text(ordstat_form, 9, 33, "Amount");
    text(ordstat_form, 9, 45, "Delivery-Date");
}

static int
ordstat_read(iobuf *in_buf, iobuf *out_buf, ordstat_trans *t, ID warehouse)
{
    int i;
    int field;
    int key;

    /* Our warehouse number is fixed */
    t->W_ID = warehouse;
    t->C_ID = EMPTY_NUM;
    t->D_ID = EMPTY_NUM;
    t->C_LAST[0] = '\0';

    /* Process fields until done */
    for (field = 1; field > 0; field = next_field(field, key, 3)) {
        retry: switch (field) {

            case 1: key = read_number(4, 29, &t->D_ID, 2, in_buf, out_buf);
                    break;

            case 2:
                /* if last name specified, skip this field */
                if (t->C_LAST[0] != '\0')
                    break;

                /* read in the customer id */
                key = read_number(5, 11, &t->C_ID, 4, in_buf, out_buf);

                /* if specified, don't allow last name to be entered */
                if (t->C_ID != EMPTY_NUM)
                {
                    blanks(out_buf, 5, 44, 16);
                    t->C_LAST[0] = '\0';
                }

                /* refresh the C_LAST underlines, if possibly needed */
                else if (t->C_LAST[0] == '\0')
                    empty(out_buf, 5, 44, 16);
                break;

            case 3:
                /* skip this field if C_ID was already specified */
                if (t->C_ID != EMPTY_NUM)
                    break;

                /* read in the customer last name */
                key = read_text(5, 44, t->C_LAST, 16, in_buf, out_buf);

                /* if specified, don't allow c_id to be entered */
                if (t->C_LAST[0] != '\0')
                {
                    blanks(out_buf, 5, 11, 4);
                    t->C_ID = EMPTY_NUM;
                }

                /* refresh the C_ID underlines, if possibly needed */
                else if (t->C_ID == EMPTY_NUM)
                    empty(out_buf, 5, 11, 4);
                break;
            }
        }

        /* ensure all the necessary fields were entered */
        if (t->D_ID == EMPTY_NUM)
            {field=1; msgline(out_buf, "Please enter district id"); goto retry;}
        if (t->C_ID == EMPTY_NUM && t->C_LAST[0] == '\0')
            {field=2; msgline(out_buf, "C_ID or C_LAST must be entered"); goto retry;}

        t->byname = (t->C_ID == EMPTY_NUM);
        msgline(out_buf, "");
        flush(out_buf);
        return key;
    }

    static void
    ordstat_write(iobuf *out_buf, ordstat_trans *t)
    {
        int i;

        /* if errors, display a status message and quit */
        if (t->status != OK) {
            status(out_buf, 24, 1, t->status);
            return;
        }

        /* display the results */
        number(out_buf, 5, 11, t->C_ID, 4);
        text(out_buf, 5, 24, t->C_FIRST);
        text(out_buf, 5, 41, t->C_MIDDLE);
        text(out_buf, 5, 44, t->C_LAST);
        money(out_buf, 6, 15, t->C_BALANCE, 10);
        number(out_buf, 8, 15, t->O_ID, 8);
        date(out_buf, 8, 38, t->O_ENTRY_DATE);
        if (t->O_CARRIER_ID > 0) {
            number(out_buf, 8, 76, t->O_CARRIER_ID, 2);
        }

        for (i=0; i<t->o_cnt; i++) {
            number(out_buf, i+10, 3, t->item[i].OL_SUPPLY_W_ID, 6);
            number(out_buf, i+10, 14, t->item[i].OL_I_ID, 6);
            number(out_buf, i+10, 25, t->item[i].OL_QUANTITY, 2);
            money(out_buf, i+10, 32, t->item[i].OL_AMOUNT, 9);
            date_only(out_buf, i+10, 47, t->item[i].OL_DELIVERY_DATE);
        }
        trigger2(out_buf);
    }

    /* delivery form processing */

    static int
    delivery(iobuf *in_buf, iobuf *out_buf, iobuf *delivery_form,
            void **data_ptr, delivery_trans *t, ID warehouse)
    {
        int key;
        display(delivery_form);
        key = delivery_read(in_buf, out_buf, t, warehouse);
        if (key != ENTER) return key;
        delivery_enqueue(data_ptr, t);
        delivery_write(out_buf, t);
        return key;
    }

    static void
    delivery_setup(iobuf *delivery_form, ID warehouse)
    {
        /* start with an empty form */
        reset(delivery_form);

        /* clear the iobuf below the menu */
        position(delivery_form, 3,1);
        clear_screen(delivery_form);

        /* set up all the field labels */
        text(delivery_form, 3, 38, "Delivery");
    }
}

```

```

text(delivery_form, 4, 1, "Warehouse:");
number(delivery_form, 4, 12, warehouse, 6);
text(delivery_form, 6, 1, "Carrier Number:");
empty(delivery_form, 6, 17, 2);
}

static int
delivery_read(iobuf *in_buf, iobuf *out_buf, delivery_trans *t, ID warehouse)
{
    int i;
    int field;
    int key;

    /* Our warehouse number is fixed */
    t->del.W_ID = warehouse;
    t->del.O_CARRIER_ID = EMPTY_NUM;

    /* Process fields until done */
    for (field = 1; field > 0; field = next_field(field, key, 1)) {
        retry: switch (field) {
            case 1: key = read_number(6, 17, &t->del.O_CARRIER_ID, 2, in_buf, out_buf);
                break;
        }
    }

    /* if Aborted, then done */
    if (key != ENTER) {
        return key;
    }

    /* Must enter the carrier id */
    if ((t->del.O_CARRIER_ID == EMPTY_NUM) ||
        (t->del.O_CARRIER_ID < 1) ||
        (t->del.O_CARRIER_ID > 10))
        {field=1; msgline(out_buf, "Please enter a Carrier Number within 1 and 10"); goto
        retry; }

    /* clear the message line */
    msgline(out_buf, "");
    flush(out_buf);
    return key;
}

static void
delivery_write(iobuf *out_buf, delivery_trans *t)
{
    if (t->del.status == OK) {
        text(out_buf, 8, 1, "Execution Status: Delivery has been queued");
        trigger2(out_buf);
    } else {
        status(out_buf, 8, 1, t->del.status);
    }
}

/*****
*/
/* stocklev form processing
*/
/*****

static int
stocklev(iobuf *in_buf, iobuf *out_buf, iobuf *stocklev_form,
        void **data_ptr, stocklev_trans *t, ID warehouse, ID district)
{
    int key;
    display(stocklev_form);
    key = stocklev_read(in_buf, out_buf, t, warehouse, district);
    if (key != ENTER) return key;
    stocklev_transaction(data_ptr, t);
    stocklev_write(out_buf, t);
    return key;
}

static void
stocklev_setup(iobuf *stocklev_form, ID warehouse, ID district)
{
    /* start with an empty form */
    reset(stocklev_form);

    /* clear the iobuf below the menu */
    position(stocklev_form, 3,1);
    clear_screen(stocklev_form);

    /* set up all the field labels */
    text(stocklev_form, 3, 35, "Stock-Level");
    text(stocklev_form, 4, 1, "Warehouse:");
    number(stocklev_form, 4, 12, warehouse, 6);
    text(stocklev_form, 4, 19, "District:");
    number(stocklev_form, 4, 29, district, 2);
    text(stocklev_form, 6, 1, "Stock Level Threshold:");
    empty(stocklev_form, 6, 24, 2);
    text(stocklev_form, 8, 1, "low stock");
}

```

```

static int
stocklev_read(iobuf *in_buf, iobuf *out_buf, stocklev_trans *t,
             ID warehouse, ID district)
{
    int field;
    int key;

    t->W_ID = warehouse;
    t->D_ID = district;
    t->threshold = EMPTY_NUM;

    /* Process fields until done */
    for (field = 1; field > 0; field = next_field(field, key, 1)) {
        retry: switch (field) {
            case 1: key = read_number(6, 24, &t->threshold, 2, in_buf, out_buf);
                break;
        }
    }

    /* if Aborted, then done */
    if (key != ENTER) {
        return key;
    }

    /* make sure the necessary fields were entered */
    if ((t->threshold == EMPTY_NUM) ||
        (t->threshold < 10) ||
        (t->threshold > 20))

        {field=1; msgline(out_buf, "Please enter a threshold within 10 and 20"); goto retry; }

    /* clear the message line */
    msgline(out_buf, "");
    flush(out_buf);
    return key;
}

static void
stocklev_write(iobuf *out_buf, stocklev_trans *t)
{
    if (t->status == OK) {
        number(out_buf, 8, 12, t->low_stock, 3);
        trigger2(out_buf);
    } else {
        status(out_buf, 10, 1, t->status);
    }
}

/*****
*****
login form processing
*****
*****

static int
login(iobuf *in_buf, iobuf *out_buf, ID *warehouse, ID *district)
{
    int field;
    int key;
    char auditstr[21];
    int w_id, d_id;

    /* assume the default values */
    w_id = *warehouse;
    d_id = *district;
    auditstr[0] = '\0';

    /* display the login menu */
    position(out_buf, 1,1); clear_screen(out_buf);
    text(out_buf, 3, 30, "Please login.");
    text(out_buf, 5,5,"Warehouse:");
    number(out_buf, 5, 16, w_id, 6);
    text(out_buf, 5, 24, "District:");
    number(out_buf, 5, 34, d_id, 2);
    text(out_buf, 15, 5, "Audit String:");
    text(out_buf, 15, 19, CLIENT_AUDIT_STRING);
    empty(out_buf, 16, 19, 20);

    /* Get values until done */
    for (field = 1; field > 0; field = next_field(field, key, 3)) {
        retry: switch (field) {
            case 1:
                key = read_number(5, 16, &w_id, 6, in_buf, out_buf);
                break;

            case 2:
                key = read_number(5, 34, &d_id, 2, in_buf, out_buf);
                break;

            case 3:

```

```

        key = read_text(16, 19, auditstr, 20, in_buf, out_buf);
        break;
    }
}

if (key != ENTER) {
    return EOF;
}

if (w_id == EMPTY_NUM && *warehouse == EMPTY_NUM) {
    msgline(out_buf, "You must enter a warehouse id");
    field = 1;
    goto retry;
}

if (d_id == EMPTY_NUM && *district == EMPTY_NUM) {
    msgline(out_buf, "You must enter a district id");
    field = 2;
    goto retry;
}

if (w_id != EMPTY_NUM) {
    *warehouse = w_id;
}
if (d_id != EMPTY_NUM) {
    *district = d_id;
}

/* done */
return key;
}

/*****
*****

menu form processing

*****
*****

static void
menu_setup(iobuf *out_buf)
{

    /* display the menu on the iobuf -- never erased */
    position(out_buf, 1, 1);
    clear_screen(out_buf);
    string(out_buf, "(1)New-Order (2)Payment (3)Order-Status ");
    string(out_buf, "(4)Delivery (5)StockLevel (9)Exit");
}

static int
menu_read(iobuf *in_buf, iobuf *out_buf)
{
    position(out_buf, 1, 1);
    trigger(out_buf);
    return getkey(in_buf, out_buf);
}

static int
next_field(int current, int key, int max)
{
    if (key == BACKTAB)
        if (current == 1) return max;
        else return current-1;
    else if (key == TAB)
        if (current == max) return 1;
        else return current+1;
    else
        return 0;
}

static void
msgline(iobuf *out_buf, char *str)
{
    position(out_buf, 24, 1);
    clear_screen(out_buf);
    string(out_buf, str);
}

static void
cleanup(iobuf *out_buf)
{
    /* detach from the delivery queue */
    delivery_done();

    /* clear the screen */
    position(out_buf, 1, 1);
    clear_screen(out_buf);
    trigger(out_buf);
    flush(out_buf);
}

```

```

/*****
*****

```

Screen Output Routines

```

*****
*****

```

```

static void
number(iobuf *out_buf, int row, int col, int n, int width)
{
    char str[81];
    fmt_num(str, n, width);
    text(out_buf, row, col, str);
}

```

```

static void
real(iobuf *out_buf, int row, int col, double x, int width, int dec)
{
    char str[81];
    fmt_flt(str, x, width, dec);
    text(out_buf, row, col, str);
}

```

```

static void
date(iobuf *out_buf, int row, int col, char *date_str)
{
    text(out_buf, row, col, date_str);
}

```

```

static void
date_only(iobuf *out_buf, int row, int col, char *date_str)
{
    date_str[10] = '\0';
    text(out_buf, row, col, date_str);
}

```

```

static void
money(iobuf *out_buf, int row, int col, double x, int width)
{
    char str[81];
    fmt_money(str, x, width);
    text(out_buf, row, col, str);
}

```

```

static void
long_text(iobuf *out_buf, int row, int col, char *str, int width)
{
    int pos;

```

```

    /* repeat until the entire string is written out */
    for (pos = width; *str != '\0'; str++, pos++)
    {

```

```

        /* if at end of line, position the cursor to next line */
        if (pos >= width)
        {
            position(out_buf, row, col);
            pos = 0;
            row++;
        }

```

```

        /* output the next character */
        pushc(out_buf, *str);
    }
}

```

```

static void
text(iobuf *out_buf, int row, int col, char str[])
{
    position(out_buf, row, col);
    string(out_buf, str);
}

```

```

static void
phone(iobuf *out_buf, int row, int col, char *str)
{
    char temp[30];

    fmt_phone(temp, str);
    text(out_buf, row, col, temp);
}

```

```

static void
zip(iobuf *out_buf, int row, int col, char *str)
{
    char temp[30];
}

```

```

    fmt_zip(temp, str);
    text(out_buf, row, col, temp);
}

static void
empty(iobuf *out_buf, int row, int col, int len)
{
    position(out_buf, row, col);
    while (len-- > 0)
        pushc(out_buf, '_');
}

static void
blanks(iobuf *out_buf, int row, int col, int len)
{
    position(out_buf, row, col);
    while (len-- > 0)
        pushc(out_buf, ' ');
}

static void
status(iobuf *out_buf, int row, int col, int status)
/******
status displays the transaction status
Note: must correspond to 'get_status' in driver/keystroke.c
*****
{
    text(out_buf, row, col, "Execution Status: ");

    if (status == OK)
        string(out_buf, "Transaction Committed");
    else if (status == E_INVALID_ITEM)
        string(out_buf, "Item number is not valid");
    /* Do the rev. 3.3 error checking here. */
    else if (status == E_INVALID_INPUT)
        string(out_buf, "Invalid input, transaction not executed");
    else
    {
        string(out_buf, "Rollback -- ");
        number(out_buf, row, col+30, status, 5);
    }
    trigger2(out_buf);
}

/******
ASCII terminal control
*****
*****

static void
trigger(iobuf *out_buf)
/******
trigger sends a turnaround sequence to let the driver know to send input
*****
{
    pushc(out_buf, TRIGGER);
}

static void
trigger2(iobuf *out_buf)
/******
trigger2 sends another turnaround sequence to let the driver know what
is going on.
*****
{
    pushc(out_buf, TRIGGER2);
}

static void
position(iobuf *out_buf, int row, int col)
/******
position positions the cursor at the given row and column
*****
{
    pushc(out_buf, ESCAPE);
    pushc(out_buf, '[');
    if (row >= 10)
        pushc(out_buf, '0' + row/10);
    pushc(out_buf, '0' + row%10);
    pushc(out_buf, ';');
    if (col >= 10)
        pushc(out_buf, '0' + col/10);
    pushc(out_buf, '0' + col%10);
    pushc(out_buf, 'H');
}

static void
clear_screen(iobuf *out_buf)
/******

```

```

clear_screen clears the iobuf from cursor position to end of iobuf
*****
{
    pushc(out_buf, ESCAPE);
    pushc(out_buf, '[');
    pushc(out_buf, 'J');
}

/******
Screen Input Routines
*****
#define funny(key) (key != ENTER && key != TAB && key != BACKTAB)

static int
read_number(int row, int col, int *n, int width, iobuf *in_buf, iobuf *out_buf)
/******
read_number reads an integer field
*****
{
    char temp[81];
    int key;
    int err;
    debug("read_number: row=%d col=%d width=%d n=%d\n", row, col, width, *n);

    /* generate the current characters */
    fmt_num(temp, *n, width);
    err = NO;

    /* repeat until a valid number or a funny key is pressed */
    for (;;)
    {
        /* Let the user edit the field */
        key = getfield(row, col, temp, width, Num, in_buf, out_buf);
        if (funny(key)) return key;

        /* convert the field to a number */
        *n = cvt_num(temp);
        if (*n != INVALID_NUM) break;

        msgline(out_buf, "Invalid digit entered");
        pushc(out_buf, BELL);
        err = YES;
    }

    /* display the new number */
    number(out_buf, row, col, *n, width);
    if (err) msgline(out_buf, "");
    debug("read_number: n=%d key=%d\n", *n, key);
    return key;
}

static int
read_money(int row, int col, double *m, int width, iobuf *in_buf, iobuf *out_buf)
{
    char temp[81];
    int key;
    int err;

    err = NO;
    fmt_money(temp, *m, width);

    /* repeat until a valid number or a funny key is pressed */
    for (;;)
    {
        key = getfield(row, col, temp, width, Money, in_buf, out_buf);
        if (funny(key)) return key;

        *m = cvt_money(temp);
        if (*m != INVALID_FLT) break;

        msgline(out_buf, "Please enter amount $99999.99");
        pushc(out_buf, BELL);
        err = YES;
    }

    money(out_buf, row, col, *m, width);
    if (err) msgline(out_buf, "");
    return key;
}

static int
read_text(int row, int col, char *s, int width, iobuf *in_buf, iobuf *out_buf)
{
    char temp[81];
    int key;
    int i;

    /* generate the current characters */
    fmt_text(temp, s, width);

```

```

/* let the user edit the field */
key = getfield(row, col, temp, width, Text, in_buf, out_buf);
if (funny(key)) return key;

/* Strip off leading and trailing space characters */
cvt_text(temp, s);

/* redisplay the current text */
fmt_text(temp, s, width);
text(out_buf, row, col, temp);

return key;
}

static int
getfield(int row, int col, char buf[], int width, FIELD_TYPE ftype,
         iobuf *in_buf, iobuf *out_buf)
{
    int pos, key;

    debug("getfield: width=%d buf=%s\n", width, width, buf);

    /* go to the beginning of the field */
    position(out_buf, row, col);
    trigger(out_buf);
    pos = 0;

    /* repeat until a special control character is pressed */
    for (;;)
    {
        /* get the next character */
        key = getkey(in_buf, out_buf);

        /* CASE: Add to buf if it fits and Is it a valid character ? */
        if (pos < width && valid_char(key, ftype))
        {
            buf[pos] = key;
            pos++;
            pushc(out_buf, key);
        }

        /* CASE: char is BACKSPACE. Erase last character. */
        else if (key == BACKSPACE && pos > 0)
        {
            pos--;
            buf[pos] = '_';
            pushc(out_buf, BACKSPACE);
            pushc(out_buf, '_');
            pushc(out_buf, BACKSPACE);
        }

        /* CASE: enter, tab, backtab, ^c. Exit loop */
        else if (key == ENTER || key == TAB || key == BACKTAB || key == CNTRL_C
                || key == EOF)
            break;

        else if (key == '\031') /* for debugging, let ^X == ENTER */
            {key = ENTER; break;}

        /* Otherwise, ignore the character and beep */
        else
            pushc(out_buf, BELL);
    }

    debug("getfield: final key: %d buf=%s\n", key, width, buf);
    return key;
}

static int
valid_char(int key, FIELD_TYPE ftype)
/******
valid_char is true if the key is valid for this type of field
*****
{
    int valid;
    switch(ftype)
    {
        case Num : valid = (isdigit(key) || key == '-' || key == '.');
                    break;

        case Text : valid = (isprint(key) || key == ' ');
                    break;

        case Money : valid = (isdigit(key) || key == '-' || key == '.'
                             || key == '$' || key == ' ');
                    break;

        default : valid = NO;
    }
}

return valid;
}

```

```

static pthread_t
spawn_user(int c_fd, long tc)
{
    int pid;
    int ret;
    pthread_t t;
    thread_data *td;

    td = (thread_data *)malloc(sizeof(thread_data));
    if (td == NULL) {
        perror("Can't create thread argument data\n");
    }
    td->fd = c_fd;
    td->tux_context = tc;
    ret = pthread_create(&t, NULL, client_main, (void *)td);
    if (ret != 0) {
        perror("Can't create client thread\n");
    }
    return t;
}

int prepare_socket(int fd)
{
    int yes = 1;
    int level;

#ifdef __hpux
        level = SOL_SOCKET;
#else
        level = IPPROTO_TCP;
#endif
    if (setsockopt(fd, SOL_SOCKET, SO_KEEPALIVE, &yes, sizeof(yes)) < 0)
        return -1;
    if (setsockopt(fd, SOL_SOCKET, SO_REUSEADDR, &yes, sizeof(yes)) < 0)
        return -1;
    if (setsockopt(fd, level, TCP_NODELAY, &yes, sizeof(yes)) < 0)
        return -1;
    return 0;
}

int
connect_client(int server_fd)
/******
connect_client connects the clients who are waiting
*****
{
    int fd, vfd;
    struct sockaddr dummy_addr;
    int dummy_size = sizeof(dummy_addr);
    int try_count = 0;

    /* accept a connection to a new client. Exit if no more */
    while(1) {
        fd = accept(server_fd, &dummy_addr, &dummy_size);
        if (fd < 0) {
            if (errno != ENOBUFS || ++try_count > 10) {
                /* syserror exits */
                perror("Can't accept new client; try # %d\n", try_count);
            }
            continue;
        } else
            break;
    }

    /* set the socket parameters */
    if (prepare_socket(fd) < 0)
        perror("Can't set socket parameters\n");

    return fd;
}

int
server_socket(int port)
/******
server_socket creates a socket for a server with the given name
*****
{
    int fd;
    struct sockaddr_in address;
    int retval;

    /* create a socket */
    fd = socket(AF_INET, SOCK_STREAM, 0);
    if (fd < 0)
        perror("Can't create a socket\n");
    if ((retval = prepare_socket(fd)) < 0)
        perror("Can't configure the socket, retval=%d\n", retval);

    /* build up an internet style address */
    address.sin_family = AF_INET;
    address.sin_port = htons(port);
    address.sin_addr.s_addr = INADDR_ANY;

    /* set up the socket to listen at the given address */
}

```



```

if (bind(fd, (struct sockaddr *)&address, sizeof(struct sockaddr)) < 0) {
    syslog("Can't bind the socket to address\n");
}

if (listen(fd, SOMAXCONN) < 0) {
    syslog("Can't listen\n");
}

return fd;
}

static void
GetArgs(int argc, char **argv)
{
    extern char *optarg;
    extern int optind;
    char ch;
    int total_users=0;
    int nr_client=1; /* minimum nr_client */
    int req_servers=0;

    while((ch = getopt(argc, argv, "u:c:p:")) != EOF) {
        switch (ch) {
            case 'u':
                total_users = atoi(optarg);
                break;

            case 'c':
                nr_client = ((nr_client = atoi(optarg)) > 0 ? nr_client : 1);
                break;

            case 'p':
                port_number = atoi(optarg);
                break;

            default:
                printf("Usage: %s -u total_users -c nr_client -p port_number\n", argv[0]);
                exit(1);
        }
    }
    req_servers = (int)ceil((double)total_users/nr_client/MAX_USERS_PER_PROCESS);
    number_of_servers = (req_servers > number_of_servers) ?
        req_servers : number_of_servers;
}

int
main(int argc, char **argv)
{
    int server_fd;
    int client_fd;
    int i;
    int pid;
    int policy;
    long tux_context; /* Tuxedo context to use */
    struct sched_param param;

    /* We don't want zombie children */
    signal(SIGCHLD, SIG_IGN);

    /* Ignore SIGPIPE, since they occur normally */
    signal(SIGPIPE, SIG_IGN);

#ifdef __hpux
    policy = SCHED_NOAGE;
    param.sched_priority = PRI_HPUX_TO_POSIX(180);
#else
    policy = SCHED_OTHER;
    param.sched_priority = 0;
#endif

    if ((sched_setscheduler(0, policy, &param)) < 0) {
        perror("Server can't run sched_noage");
    }

    GetArgs(argc, argv);

    /* create a socket to accept new requests */
    server_fd = server_socket(port_number);
    if (server_fd < 0) {
        syslog("Can't create a listening socket\n");
    }

    /* Create more servers if requested */
    for(i = 0; i < (number_of_servers-1); i++) {
        if (pid = fork() == -1) {
            syslog("Could not fork a new helper process\n");
        } else if (pid == 0) {
            /* Child */
            break;
        } else {
            /* Parent */
        }
    }

    /* repeat forever in each child */
    while (user_connections < MAX_USERS_PER_PROCESS) {
        client_fd = connect_client(server_fd);

```

```

        if ((user_connections % MAX_THREADS_PER_CONTEXT) == 0) {
            /* connect to the transaction processor */
            tux_context = transaction_begin();
        }
        user_ids[user_connections] = spawn_user(client_fd, tux_context);
        user_connections++;
    }

    /* Close listening socket */
    close(server_fd);

    for(i = 0; i < user_connections; i++) {
        if (pthread_join(user_ids[i], NULL) != 0) {
            message("Pthread message, error = %d, thread_id = %d, id = %d\n",
                errno, user_ids[i], i);
            syslog("Pthread_join error\n");
        }
    }

    /* detach from transaction engine */
    transaction_done();

    return 0;
}

```

client/tux_transaction.c

```

*****
@(#) Version: A.10.10 $Date: 2005/04/11 10:11:33 $
*****
(c) Copyright 1996, Hewlett-Packard Company, all rights reserved.
*****
*****/

#include <varargs.h>
#include <errno.h>
#include "tux_transaction.h"

#define MYMAX(a, b) (a > b) ? a : b

static void
tux_error(format, va_alist)
char *format;
va_dcl
{
    va_list argptr;

    va_start(argptr);
    vmessage(format, argptr);

    message("Tuxedo error %d\n", tpermo);

    errno = Unixerr;
    if (tpermo == TPEOS) {
        syslog("Tuxedo encountered O/S error\n");
    }

    if (tpermo == TPESVCERR || tpermo == TPETIME) {
        message("Retrying transaction\n");
        if (tpermo == TPETIME)
            /*JW
            sleep (1);
            */
            delay(1.0);
    } else {
        error("EXITING !!!\n");
    }
}

TPCONTEXT_T
transaction_begin(void)
{
    static TPINIT *initialization_buffer = NULL;
    TPCONTEXT_T ctx = 0; /*JW: was NULL, Linux compiled with a warning message.
        // TPCONTEXT_T is a typedef long.

    /* Create buffer needed to indicate MultiContexts operation */
    if (initialization_buffer == NULL) {
        initialization_buffer = (TPINIT *)calloc("TPINIT", NULL,
            TPINITNEED(0));

        if (initialization_buffer == NULL) {
            tux_error("Unable to allocate Tuxedo TPINIT memory\n");
        }

        initialization_buffer->flags = TPMULTICONTEXTS;
    }

    /* attach to Tuxedo */
    if (tpinit(initialization_buffer) == -1) {

```

```

        tux_error("Failed to attach to Tuxedo\n");
    }

    /* get the context */
    if (tpgetctx(&ctx, 0) == -1) {
        tux_error("Failed to get Tuxedo context\n");
    }
    return ctx;
}

void *
thread_transaction_begin(TPCONTEXT_T ctx)
{
    unsigned long alloc_size;
    void *data_ptr;

    if (tpsetctx(ctx, 0) == -1) {
        tux_error("Could not set Tuxedo context\n");
    }

    /* allocate structures for each transaction */
    alloc_size = MYMAX(sizeof(neworder_trans), sizeof(payment_trans));
    alloc_size = MYMAX(alloc_size, sizeof(ordstat_trans));
    alloc_size = MYMAX(alloc_size, sizeof(stocklev_trans));
    alloc_size = MYMAX(alloc_size, sizeof(delivery_trans));
    data_ptr = (void *)tpalloc("CARRAY", NULL, alloc_size);

    if (data_ptr == NULL) {
        tux_error("Unable to allocate Tuxedo memory\n");
    }
    return data_ptr;
}

void
transaction_done(void)
{
    if (tpterm() == -1) {
        tux_error("Unable to detach from Tuxedo\n");
    }
}

void
neworder_transaction(void **data_ptr, neworder_trans *t)
{
    long result;
    *((neworder_trans *)*data_ptr) = *t;
    while (tpcall("NEWO_SVC", (char *)*data_ptr, sizeof(neworder_trans),
        (char **)(&data_ptr), &result, TPSIGRSTRT|TPNOTIME) == -1) {
        tux_error("Tuxedo failed for neworder transaction\n");
        *((neworder_trans *)*data_ptr) = *t;
    }
    *t = *((neworder_trans *)*data_ptr);
}

void
payment_transaction(void **data_ptr, payment_trans *t)
{
    long result;
    *((payment_trans *)*data_ptr) = *t;
    while (tpcall("PMT_SVC", (char *)*data_ptr, sizeof(payment_trans),
        (char **)(&data_ptr), &result, TPSIGRSTRT|TPNOTIME) == -1) {
        tux_error("Tuxedo failed for payment transaction\n");
        *((payment_trans *)*data_ptr) = *t;
    }
    *t = *((payment_trans *)*data_ptr);
}

void
ordstat_transaction(void **data_ptr, ordstat_trans *t)
{
    long result;
    *((ordstat_trans *)*data_ptr) = *t;
    while (tpcall("ORDS_SVC", (char *)*data_ptr, sizeof(ordstat_trans),
        (char **)(&data_ptr), &result, TPSIGRSTRT|TPNOTIME) == -1) {
        tux_error("Tuxedo failed for ordstat transaction\n");
        *((ordstat_trans *)*data_ptr) = *t;
    }
    *t = *((ordstat_trans *)*data_ptr);
}

void
stocklev_transaction(void **data_ptr, stocklev_trans *t)
{
    long result;
    *((stocklev_trans *)*data_ptr) = *t;
    while (tpcall("STKL_SVC", (char *)*data_ptr, sizeof(stocklev_trans),
        (char **)(&data_ptr), &result, TPSIGRSTRT|TPNOTIME) == -1) {
        tux_error("Tuxedo failed for stocklev transaction\n");
        *((stocklev_trans *)*data_ptr) = *t;
    }
    *t = *((stocklev_trans *)*data_ptr);
}

```

```

void
delivery_init(int u)
{
}

void
delivery_enqueue(void **data_ptr, delivery_trans *t)
{
    gettimeofday(&t->del.enqueue[0], NULL);
    t->del.status = OK;

    *((delivery_trans *)*data_ptr) = *t;
    while (tpacall("DVRV_SVC", (char *)*data_ptr, sizeof(delivery_trans),
        TPNOREPLY) == -1) {
        tux_error("Tuxedo failed enqueueing delivery transaction\n");
        *((delivery_trans *)*data_ptr) = *t;
    }
}

void
delivery_done(void)
{
}

```

client/Makefile

```

#####
#####
#(##) Version: A.10.10 $Date: 2003/06/26 16:03:06 $
#####
#(c) Copyright 1996, Hewlett-Packard Company, all rights reserved.
#####

#
# Common makefile definitions that are used on all platforms.
#

P = ${WORK_DIR}/src
L = $(P)/lib
D = $(P)/driver
S = $(P)/client

include ./database.mk

SH_OPT =
OPT = -Wl,-E
LDOPTS = -E -ldld

TUX_INCLUDE= -I${TUXDIR}/include

COMMON_FLAGS = $(OPTIMIZE) $(DD_DS_FLAGS) $(CINCLUDES)
SH_CFLAGS = $(COMMON_FLAGS) $(SH_OPT) ${TUX_INCLUDE}
CFLAGS = $(COMMON_FLAGS) $(OPT) ${TUX_INCLUDE}

PROGRAMS = client client_batch client_batch_dumb service_real service_dumb raw

all: ${PROGRAMS}

${S}/dummy/transaction.o: ${S}/dummy/transaction.c
    cc $(CFLAGS) -o ${S}/dummy/transaction.o -c
    ${S}/dummy/transaction.c

raw: raw.o $(D)/socket.o
    cc $(CFLAGS) raw.o $(D)/socket.o $(L)/tpc_lib.a -o raw

client.o: client.c
    cc $(CFLAGS) -D_REENTRANT -c client.c

tux_transaction.o: tux_transaction.c
    cc $(CFLAGS) -D_REENTRANT -c tux_transaction.c

dummy_queue.o: dummy_queue.c
    cc $(CFLAGS) -D_REENTRANT -c dummy_queue.c

post_dvrv.o: post_dvrv.c
    cc $(CFLAGS) -D_REENTRANT -c post_dvrv.c

client: client.o tux_transaction.o
    ${TUXDIR}/bin/buildclient -v -o client \
    -f "${COMMON_FLAGS} -D_REENTRANT $(OPT) $(VMPSZ) \
    client.o tux_transaction.o $(L)/tpc_lib.a" -l -lnsl -lm -lc"

service_dumb: service.o post_dvrv.o ${S}/dummy/transaction.o $(L)/tpc_lib.a
    ${TUXDIR}/bin/buildserver -v -b shm -s NEWO_SVC -s PMT_SVC \

```

```

-s ORDS_SVC -s STKL_SVC -s DVRY_SVC -o service_dumb \
-f "$(SVC_VMPSZ) service.o post_dvry.o \
${S}/dummy/transaction.o $(L)/tpc_lib.a $(L)/load_lib.a" -l "-lnsl -lm"

service_real: service.o post_dvry.o $(DB_OBJS) $(L)/tpc_lib.a
$(TUXDIR)/bin/buildserver -v -b shm -s NEWO_SVC -s PMT_SVC \
-s ORDS_SVC -s STKL_SVC -s DVRY_SVC -o service_real \
-f "$(SVC_VMPSZ) service.o post_dvry.o \
$(DB_OBJS) $(L)/tpc_lib.a $(LDFLAGS)" -l "-lnsl"

client_batch: $(D)/driver.o $(D)/keystroke_batch.o $(D)/screen.o \
$(D)/generate.o $(D)/shm.o $(D)/tpcc.o \
post_dvry.o $(DB_OBJS) dummy_que.o $(L)/tpc_lib.a
$(CC) $(D)/driver.o $(D)/keystroke_batch.o $(D)/screen.o \
$(D)/generate.o $(D)/shm.o $(D)/tpcc.o post_dvry.o \
$(DB_OBJS) dummy_que.o $(L)/tpc_lib.a \
$(LDFLAGS) -o client_batch;

client_batch_dumb: $(D)/driver.o $(D)/keystroke_batch.o $(D)/screen.o \
$(D)/generate.o $(D)/shm.o $(D)/tpcc.o \
post_dvry.o $(S)/dummy/transaction.o dummy_que.o \
$(L)/tpc_lib.a $(L)/load_lib.a
$(CC) $(CFLAGS) $(D)/driver.o $(D)/keystroke_batch.o $(D)/screen.o \
$(D)/generate.o $(D)/shm.o $(D)/tpcc.o post_dvry.o \
$(S)/dummy/transaction.o dummy_que.o $(L)/tpc_lib.a \
$(L)/load_lib.a \
-lnsl -lm -lc -lpthread \
-o client_batch_dumb;

#lnsl -lm -lc -lcl

clean:
rm -f *.o
rm -f */*.o

clobber: clean
rm -f ${PROGRAMS}
rm -f ${WORK_DIR}/bin/service

install: ${PROGRAMS}
cp ${PROGRAMS} ${WORK_DIR}/bin
ln -s ./service_real ${WORK_DIR}/bin/service

```

oracle.mk

```

*****
#@(#) Version: A.10.10 $Date: 2003/06/26 16:03:06 $
#
#(c) Copyright 1996, Hewlett-Packard Company, all rights reserved.
*****

#
# Oracle specific build definitions
#
#include $(ORACLE_HOME)/rdbs/lib/env_rdbms.mk
LIBDIR=lib$(BITSZ)

#
# Oracle specific definitions
#
ORA_DEF_OPT = $(ORACLE_HOME)/rdbs/lib/defopt.o
ORA_CFLAGS = $(COMMON_FLAGS) $(DB_INCLUDE) $(OPT)
${TUX_INCLUDE}
ORA_LDFLAGS = $(DD_DS_FLAGS) -L$(ORACLE_HOME)/rdbs/lib/\
-L$(ORACLE_HOME)/lib/\
-L$(ORACLE_HOME)/rdbs/lib/\
-lclntsh `cat $(ORACLE_HOME)/lib/sysliblist` -lm -o $@ \
$(ORA_DEF_OPT) -lclntsh

DB_LDFLAGS = $(ORA_LDFLAGS$(BITSZ))
#Old INCLUDE for Oracle labeled prior to 0612-2003.
#DB_INCLUDE =-I. -I$(ORACLE_HOME)/rdbs/demo -
-I$(ORACLE_HOME)/rdbs/public \
-I$(ORACLE_HOME)/rdbs/include -I$(ORACLE_HOME)/plsql/public \
-I$(ORACLE_HOME)/network/public

DB_INCLUDE =-I. -I$(ORACLE_HOME)/rdbs/public \
-I$(ORACLE_HOME)/network/public

DB_OBJS=plnew.o plord.o plpay.o pldel.o plsto.o tpccpl.o ${S}/oracle/transaction.o

```

```

plnew.o: ${S}/oracle/plnew.c
$(CC) $(ORA_CFLAGS) $(L)/tpc_lib.a -c ${S}/oracle/plnew.c;
plord.o: ${S}/oracle/plord.c
$(CC) $(ORA_CFLAGS) $(L)/tpc_lib.a -c ${S}/oracle/plord.c;
plpay.o: ${S}/oracle/plpay.c
$(CC) $(ORA_CFLAGS) $(L)/tpc_lib.a -c ${S}/oracle/plpay.c;
pldel.o: ${S}/oracle/pldel.c
$(CC) $(ORA_CFLAGS) $(L)/tpc_lib.a -c ${S}/oracle/pldel.c;
plsto.o: ${S}/oracle/plsto.c
$(CC) $(ORA_CFLAGS) $(L)/tpc_lib.a -c ${S}/oracle/plsto.c;
tpccpl.o: ${S}/oracle/tpccpl.c
$(CC) $(ORA_CFLAGS) $(L)/tpc_lib.a -c ${S}/oracle/tpccpl.c;

${S}/oracle/transaction.o: ${S}/oracle/transaction.c
$(CC) $(ORA_CFLAGS) -o ${S}/oracle/transaction.o -c
${S}/oracle/transaction.c

```

post_dvry.c

```

/*****
****
#@(#) Version: A.10.10 $Date: 2005/04/11 10:10:45 $

(c) Copyright 1996, Hewlett-Packard Company, all rights reserved.
*****
****
#include <unistd.h>
#include <stdio.h>
#include <stdlib.h>
#include <netinet/in.h>
#include "tpcc.h"

```

```

static FILE *rfiler;

void
dvry_open(int id)
{
    char fullname[128];
    char *basename;

    /* get the base file name for the deferred dvry results */
    /*
    * Make it a directory under /tmp so at least we can set it to a
    * symbolic link in case /tmp doesn't have enough room.
    */
    basename = getenv("TPCC_DVRY_FILE");
    if (basename == NULL) {
        basename = "/tmp/TPCC_DVRY_FILE";
    }

    /* create the full file name */
    sprintf(fullname, "%s.%d", basename, id);

    /* open the file */
    rfiler = fopen(fullname, "ab");
    if (rfiler == NULL) {
        syslog("Delivery server %d can't open file %s\n", id, fullname);
    }
}

void
post_dvry(delivery_trans *t)
{
    int d;
    delivery_rec r[1];

    r->status= htonl( t->del.status );
    r->W_ID = htonl( t->del.W_ID );
    r->O_CARRIER_ID = htonl( t->del.O_CARRIER_ID );
    for (d=0; d<10; d++) {
        r->order[d].O_ID= htonl( t->del.order[d].O_ID);
        r->order[d].status = htonl( t->del.order[d].status);
    }
    r->enqueue[0].tv_usec = (int)htonl( t->del.enqueue[0].tv_usec );
    r->enqueue[0].tv_nsec = (int)htonl( t->del.enqueue[0].tv_nsec );
    r->dequeue[0].tv_usec = (int)htonl( t->del.dequeue[0].tv_usec );
    r->dequeue[0].tv_nsec = (int)htonl( t->del.dequeue[0].tv_nsec );
    r->complete[0].tv_usec = (int)htonl( t->del.complete[0].tv_usec );
    r->complete[0].tv_nsec = (int)htonl( t->del.complete[0].tv_nsec );

    /* ACID_STUFF */
    r->acid_timing = htonl( t->del.acid_timing );
    r->acid_action = htonl( t->del.acid_action );
}

```

```

        if (fwrite(r, sizeof(*r), 1, rfile) != 1)
            syserror("Delivery server: Can't post dvry results\n");
    }

void
dvry_close(void)
{
    if (fclose(rfile) < 0)
        syserror("Delivery server can't close file\n");
}

```

tux_transaction.h

```

/*****
@(#) Version: A.10.10 $Date: 2005/04/11 10:11:42 $

(c) Copyright 1996, Hewlett-Packard Company, all rights reserved.
*****/

#ifndef __tux_transaction_header__
#define __tux_transaction_header__

#include <varargs.h>
#include <errno.h>

#include "tpcc.h"
#include "atmi.h"
#include "Unix.h"

void *thread_transaction_begin(TPCONTEXT_t ctx);
void transaction_done(void);
void neworder_transaction(void **data_ptr, neworder_trans *t);
void payment_transaction(void **data_ptr, payment_trans *t);
void ordstat_transaction(void **data_ptr, ordstat_trans *t);
void stocklev_transaction(void **data_ptr, stocklev_trans *t);
void delivery_init(int u);
void delivery_enqueue(void **data_ptr, delivery_trans *t);
void delivery_done(void);

#endif

```

A.2 Tpc_lib Source

lib/tpcc.h

```

/*****
@(#) Version: A.10.10 $Date: 2005/04/11 09:59:56 $

(c) Copyright 1996, Hewlett-Packard Company, all rights reserved.

History
@022801 ML Added Client Substitution Report for TPC-C TAB ID 334.

*****/

#ifndef TPCC_INCLUDED
#define TPCC_INCLUDED
#include <values.h>
#include <sys/time.h>
#include <sys/resource.h>
#include <errno.h>

/* The auditor can define these 20 char strings to be anything */
#define DRIVER_AUDIT_STRING "driver audit string"
#define CLIENT_AUDIT_STRING "client audit string"

#define BATCH_MODE ( tpcc_mode == 1 || tpcc_mode == 5 )

#ifndef DEBUG

```

```

#define debug printf
#else
#define debug (void)
#endif

#include <stdio.h>

typedef int ID; /* All id's */
typedef double MONEY; /* Large integer number of cents */
typedef char TEXT; /* Add an extra byte for null terminator */
typedef double TIME; /* Elapsed seconds from start of run (float?) */
typedef int COUNT; /* integer numbers of things */
typedef long long BIN_TYPE;
typedef double REAL; /* real numbers */
typedef int LOGICAL; /* YES or NO */
typedef struct { /* days and seconds since Jan 1, 1900 */
    int day; /* NULL represented by negative day */
    int sec;
} DATE;

/* Macro to convert time of day to TIME */
/* tpcc_start_time is declared in "delay.c" */
#include <time.h>
extern struct timeval tpcc_start_time;
#define elapsed_time(t) ( ((t)->tv_sec - tpcc_start_time.tv_sec) + \
    ((t)->tv_usec - tpcc_start_time.tv_usec) / 1000000.0 )

typedef enum {Num,Money,Text,Time,Real,Date} FIELD_TYPE; /* screen field types */

/* Various TPCC constants */
#define W_ID_LEN 4
#define D_ID_LEN 2
#define C_ID_LEN 4
#define I_ID_LEN 6
#define OL_QTY_LEN 2
#define PMT_LEN 7
#define C_ID_LEN 4
#define C_LAST_LEN 16
#define CARRIER_LEN 2
#define THRESHOLD_LEN 2
#define DIST_PER_WARE 10
#define CUST_PER_DIST 3000
#define ORD_PER_DIST 3000
#define MAXITEMS 100000

/* database identifiers and populations */
#define no_item MAXITEMS /* 100000 */
#define no_dist_pw DIST_PER_WARE
#define no_cust_pd CUST_PER_DIST /* 3000 */
#define no_ord_pd ORD_PER_DIST /* 3000 */

/* fields to add to each transaction for acid testing */
#define ACID_STUFF \
    char acid_txn[2]; \
    int acid_timing; \
    int acid_action

typedef struct
{
    int tv_sec; /* use 4-byte int, 8-byte timeval across 32/64b machines */
    int tv_usec;
} timeval_32b;

typedef struct {
    ID OL_SUPPLY_W_ID;
    ID OL_I_ID;
    TEXT I_NAME[24+1];
    COUNT OL_QUANTITY;
    COUNT S_QUANTITY;
    MONEY I_PRICE;
    char brand_generie;
} neworder_item;

typedef struct {
    int status;
    LOGICAL all_local;
    ID W_ID;
    ID D_ID;
    ID C_ID;
    TEXT C_LAST[C_LAST_LEN+1];
    TEXT C_CREDIT[2+1];
    REAL C_DISCOUNT;
    COUNT O_OL_CNT;
    ID O_ID;
    TEXT O_ENTRY_D[20]; /* dates as text fields */
    REAL W_TAX;
    REAL D_TAX;
    neworder_item item[15];
    ACID_STUFF;
} neworder_trans;

```

```

typedef struct {
    int status;
    LOGICAL byname;
    ID W_ID;
    ID D_ID;
    ID C_ID;
    ID C_D_ID;
    ID C_W_ID;
    MONEY H_AMOUNT;
    TEXT H_DATE[20]; /* date as text field */
    TEXT W_STREET_1[20+1];
    TEXT W_STREET_2[20+1];
    TEXT W_CITY[20+1];
    TEXT W_STATE[2+1];
    TEXT W_ZIP[9+1];
    TEXT D_STREET_1[20+1];
    TEXT D_STREET_2[20+1];
    TEXT D_CITY[20+1];
    TEXT D_STATE[2+1];
    TEXT D_ZIP[9+1];
    TEXT C_FIRST[16+1];
    TEXT C_MIDDLE[2+1];
    TEXT C_LAST[16+1];
    TEXT C_STREET_1[20+1];
    TEXT C_STREET_2[20+1];
    TEXT C_CITY[20+1];
    TEXT C_STATE[2+1];
    TEXT C_ZIP[9+1];
    TEXT C_PHONE[16+1];
    TEXT C_SINCE[20]; /* date as text field */
    TEXT C_CREDIT[2+1];
    MONEY C_CREDIT_LIM;
    REAL C_DISCOUNT;
    REAL C_BALANCE;
    TEXT C_DATA[200+1];
    ACID_STUFF;
} payment_trans;

typedef struct {
    int status;
    LOGICAL byname;
    ID W_ID;
    ID D_ID;
    ID C_ID;
    TEXT C_FIRST[16+1];
    TEXT C_MIDDLE[2+1];
    TEXT C_LAST[16+1];
    MONEY C_BALANCE;
    ID O_ID;
    TEXT O_ENTRY_DATE[20]; /* date as text field */
    ID O_CARRIER_ID;
    COUNT ol_cnt;
    struct {
        ID OL_SUPPLY_W_ID;
        ID OL_I_ID;
        COUNT OL_QUANTITY;
        MONEY OL_AMOUNT;
        TEXT OL_DELIVERY_DATE[20]; /* date as text field */
    } item[15];
    ACID_STUFF;
} ordstat_trans;

typedef struct {
    int status;
    ID W_ID;
    ID D_ID;
    COUNT threshold;
    COUNT low_stock;
    ACID_STUFF;
} stocklev_trans;

typedef struct {
    int status;
    ID W_ID;
    ID O_CARRIER_ID;
    struct {
        ID O_ID;
        int status;
    } order[10];
    struct timeval enqueue[1];
    struct timeval deque[1];
    struct timeval complete[1];
    ACID_STUFF;
} delivery_trans1;

/*
 * Delivery structure needs to be padded to a multiple of 512-bytes
 * for things to work properly with a sigle delivery file.
 */
typedef struct {
    delivery_trans1 del;
    char pad_data[512-sizeof(delivery_trans1)]; /* pad out to 512 bytes */
}

```

```

} delivery_trans;

typedef struct {
    int status;
    ID W_ID;
    ID O_CARRIER_ID;
    struct {
        ID O_ID;
        int status;
    } order[10];
    timeval_32b enqueue[1];
    timeval_32b deque[1];
    timeval_32b complete[1];
    ACID_STUFF;
} delivery_rec; /* use delivery_rec to write/read to/from result files */

typedef union {
    neworder_trans neworder;
    payment_trans payment;
    ordstat_trans ordstat;
    delivery_trans delivery;
    stocklev_trans stocklev;
    int status;
} generic_trans;

/*****
Record formats for results
*****/
typedef struct {
    TIME t1, t2, t3, t4, t5;
    int status;
    unsigned int type :3;
    unsigned int ol_cnt :4;
    unsigned int remote_ol_cnt :4;
    unsigned int byname :1;
    unsigned int remote :1;
    unsigned int skipped :4;
    int clientnum;
    int userid;
} success_t;

/*****
Record formats for loading routines. (DB's have own internal formats
*****/
typedef struct {
    ID W_ID;
    TEXT W_NAME[10+1];
    TEXT W_STREET_1[20+1];
    TEXT W_STREET_2[20+1];
    TEXT W_CITY[20+1];
    TEXT W_STATE[2+1];
    TEXT W_ZIP[9+1];
    REAL W_TAX;
    MONEY W_YTD;
} warehouse_row;

typedef struct {
    ID D_ID;
    ID D_W_ID;
    TEXT D_NAME[10+1];
    TEXT D_STREET_1[20+1];
    TEXT D_STREET_2[20+1];
    TEXT D_CITY[20+1];
    TEXT D_STATE[2+1];
    TEXT D_ZIP[9+1];
    REAL D_TAX;
    MONEY D_YTD;
    ID D_NEXT_O_ID;
} district_row;

typedef struct {
    ID C_ID;
    ID C_D_ID;
    ID C_W_ID;
    TEXT C_FIRST[16+1];
    TEXT C_MIDDLE[2+1];
    TEXT C_LAST[16+1];
    TEXT C_STREET_1[20+1];
    TEXT C_STREET_2[20+1];
    TEXT C_CITY[20+1];
    TEXT C_STATE[2+1];
    TEXT C_ZIP[9+1];
    TEXT C_PHONE[16+1];
    DATE C_SINCE;
    TEXT C_CREDIT[2+1];
    MONEY C_CREDIT_LIM;
    REAL C_DISCOUNT;
    MONEY C_BALANCE;
    MONEY C_YTD_PAYMENT;
}

```

```

COUNT_C_PAYMENT_CNT;
COUNT_C_DELIVERY_CNT;
TEXT_C_DATA[500+1];
} customer_row;

```

```

typedef struct
{
    ID_H_C_ID;
    ID_H_C_D_ID;
    ID_H_C_W_ID;
    ID_H_D_ID;
    ID_H_W_ID;
    DATE_H_DATE;
    MONEY_H_AMOUNT;
    TEXT_H_DATA[24+1];
} history_row;

```

```

typedef struct
{
    ID_NO_O_ID;
    ID_NO_D_ID;
    ID_NO_W_ID;
} neworder_row;

```

```

typedef struct
{
    ID_O_ID;
    ID_O_D_ID;
    ID_O_W_ID;
    ID_O_C_ID;
    DATE_O_ENTRY_D;
    ID_O_CARRIER_ID;
    COUNT_O_OL_CNT;
    LOGICAL_O_ALL_LOCAL;
} order_row;

```

```

typedef struct
{
    ID_OL_O_ID;
    ID_OL_D_ID;
    ID_OL_W_ID;
    ID_OL_NUMBER;
    ID_OL_I_ID;
    ID_OL_SUPPLY_W_ID;
    DATE_OL_DELIVERY_D;
    COUNT_OL_QUANTITY;
    MONEY_OL_AMOUNT;
    TEXT_OL_DIST_INFO[24+1];
} orderline_row;

```

```

typedef struct
{
    ID_I_ID;
    ID_I_IM_ID;
    TEXT_I_NAME[24+1];
    MONEY_I_PRICE;
    TEXT_I_DATA[50+1];
} item_row;

```

```

typedef struct
{
    ID_S_I_ID;
    ID_S_W_ID;
    COUNT_S_QUANTITY;
    TEXT_S_DIST_01[24+1];
    TEXT_S_DIST_02[24+1];
    TEXT_S_DIST_03[24+1];
    TEXT_S_DIST_04[24+1];
    TEXT_S_DIST_05[24+1];
    TEXT_S_DIST_06[24+1];
    TEXT_S_DIST_07[24+1];
    TEXT_S_DIST_08[24+1];
    TEXT_S_DIST_09[24+1];
    TEXT_S_DIST_10[24+1];
    COUNT_S_YTD;
    COUNT_S_ORDER_CNT;
    COUNT_S_REMOTE_CNT;
    TEXT_S_DATA[50+1];
} stock_row;

```

```

/* Empty field values */
#define EMPTY_NUM (MAXINT-1)
#define INVALID_NUM (MAXINT)
#define EMPTY_FLT (MAXDOUBLE)
#define INVALID_FLT (MINDOUBLE)

```

```

/* Status conditions */
#define OK 0
#define E 1
#define E_INVALID_ITEM 2
#define E_NOT_ENOUGH_ORDERS 3
#define E_DB_ERROR 4
#define E_INVALID_INPUT 5
#define E_DB_IRRECERR 6

```

```

/* Error message strings */
extern const char *e_msg[];

```

```

#define YES 1
#define NO 0

```

```

double cvt_ftl();
double cvt_money();
TIME getclock();
TIME getlocalclock();

```

```

#define TPC_MSG_QUE 150

```

```

/*****
Transaction specific stuff
*****/

```

```

/* types of transactions */
#define NEWORDER 1
#define PAYMENT 2
#define ORDSTAT 3
#define DELIVERY 4
#define STOCKLEV 5
#define DEFERRED 6 /* deferred portion of delivery */

```

```

/* the name of each transaction */
extern const char *transaction_name[];

```

```

#endif /* TPCC_INCLUDED */

```

lib/key_chars.h

```

#ifndef __TPCC_KEY_CHARS__
#define __TPCC_KEY_CHARS__

```

```

/* Standard characters used for screen control */
#define ENTER '\015'
#define TAB '\t'
#define BACKTAB '\02' /* ^B */
#define CNTRL '\03'
#define BACKSPACE '\010'
#define BELL '\07'
#define BLANK ' '
#define UNDERLINE '_'
#define ESCAPE '\033'
#define TRIGGER '\021' /* dc1 */
#define TRIGGER2 '\022' /* dc2 */
#endif

```

lib/errlog.c

```

/*****
*****/

```

```

@(#) Version: A.10.10 $Date: 2005/04/11 10:00:24 $

```

```

(c) Copyright 1996, Hewlett-Packard Company, all rights reserved.

```

```

*****/

```

```

#include <stdio.h>
#include <varargs.h>
#include <unistd.h>
#include <errno.h>
#include <string.h>
#include <stdlib.h>
#include <fcntl.h>
#include "errlog.h"

```

```

#define MSG_BUF_SIZE 3*1024

```

```

static int msgfile_fd = -1;

```

```

static void
msg_buf(char *buf, int size)

```

```

{
    char writebuf[MSG_BUF_SIZE+66];
    int ltimestamp;
    time_t teepoch = time(NULL);

    ltimestamp = strftime(writebuf, 64, "%m/%d %T ", localtime(&teepoch));
    strncpy(writebuf+ltimestamp, buf, size);
    write(msgfile_fd, writebuf, ltimestamp + size);
}

void
vmessage(format, argptr)
/*****
*****
char *format;
va_list argptr;

char buf[MSG_BUF_SIZE];

/* format a message id */
sprintf(buf, "Host %-8s Pid %-6d ", getenv("HOST_NAME"), getpid());

/* format the string and print it */
vsprintf(buf+ltimestamp, format, argptr);
if (getenv("NO_ERROR_LOG") == NULL) {
    msg_buf(buf, strlen(buf));
}
if (getenv("NO_STDERR") == NULL) {
    write(2, buf, strlen(buf));
}
}

void
error(format, va_alist)
/*****
*****
error formats a message and outputs it to a standard location (stderr for now)
*****
char *format;
va_dcl
va_list argptr;

msg_buf("error \n", strlen("error \n"));

/* point to the list of arguments */
va_start(argptr);

/* format and print to stderr */
vmessage(format, argptr);

/* done */
va_end(argptr);

/* take an error exit */
exit(1);
}

void
message(format, va_alist)
/*****
*****
message formats a message and outputs it to a standard location (stderr for now)
*****
char *format;
va_dcl
va_list argptr;

msg_buf("message \n", strlen("message \n"));
/* point to the list of arguments */
va_start(argptr);

/* format and print to stderr */
vmessage(format, argptr);

/* done */
va_end(argptr);
}

void
syserror( format, va_alist )
/*****
*****
syserror logs a message with the system error code
*****
char *format;
va_dcl
va_list argptr;
int save_errno = errno;

msg_buf("syserror \n", strlen("syserror \n"));
/* point to the list of arguments */
va_start(argptr);

```

```

/* format and print to stderr */
vmessage(format, argptr);

/* done */
va_end(argptr);

/* display the system error message */
message(" System error message: %d %s\n", save_errno,
        strerror(save_errno));

/* take an error exit */
exit(1);
}

void
console_error(char *str)
{
    int fd = open("/dev/tty", O_WRONLY);
    write(fd, str, strlen(str));
    close(fd);
    exit(1);
}

/*
 * Configure the file descriptor for the message subsystem.
 * This is not multithreaded, so it must be done by the master
 * process before threads are spawned.
 */
void
configure_error_log(void)
{
    char *fname;

    /* get the file name to use */
    fname = getenv("ERROR_LOG");
    if (fname == NULL) {
        fname = "/tmp/ERROR_LOG";
    }

    if (msgfile_fd == -1) {
        msgfile_fd = open(fname, O_WRONLY | O_CREAT |
O_APPEND, 0666);
        if (msgfile_fd < 0) {
            console_error("Can't open tpc error log file
ERROR_LOG\n");
        }
    }
}

```

lib/fmt.c

```

/*****
*****
@(#) Version: A.10.10 $Date: 2005/04/11 10:00:28 $

(c) Copyright 1996, Hewlett-Packard Company, all rights reserved.
*****
*****/
#include "tpcc.h"
#include <math.h> /* needed for ceil (VM) */
#include <strings.h>
#include <ctype.h>

/* formatting routines. */

/* Note: Currently use integer routines to format and convert. Need to
modify the code for cases when integers don't work. */
void fmt_flt(char *str, double x, int width, int dec);
void cvt_text(char *s, char *text);
void fmtint(char *field, int value, int size, char fill);

void
fmt_money(char *str, MONEY m, int width)
{
    if (m == EMPTY_FLT)
    {
        memset(str, '_', width);
        str[width] = '\0';
        return;
    }

    /* format it as a number with a leading blank */
    *str = ' ';
    fmt_flt(str+1, m/100, width-1, 2);

    /* fill in a leading dollar */

```

```

while (*str+1) == ' ')
    str++;
*str = '$';
}

double
cvt_money(char *str)
{
    char temp[81], *t, *s;
    double cvtflt(), f;

    /* skip leading and trailing blanks */
    cvt_text(str, temp);

    /* remove leading $ */
    if (*temp == '$') t = temp + 1;
    else t = temp;

    /* start scan at current character */
    s = t;

    /* allow leading minus sign */
    if (*s == '-')
        s++;

    /* allow leading digits */
    while (isdigit(*s))
        s++;

    /* allow decimal pt and two decimal digits */
    if (*s == '.') s++;
    if (isdigit(*s)) s++;
    if (isdigit(*s)) s++;

    /* There should be no more characters */
    if (*s != '\0') return INVALID_FLT;

    /* convert the floating pt number */
    f = cvtflt(t);
    if (f == EMPTY_FLT) return EMPTY_FLT;
    else if (f == INVALID_FLT) return INVALID_FLT;
    else return rint(f*100);
}

```

```

void
fmt_num(char *str, int n, int width)
{
    /* mark the end of the string */
    str[width] = '\0';

    /* if empty number, return the empty field */
    if (n == EMPTY_NUM)
        memset(str, '_', width);

    /* otherwise, convert the integer */
    else
        fmtint(str, n, width, ' ');

    debug("fmt_num: n=%d str=%s\n", n, str);
}

```

```

int
cvt_num(char *str)
{
    char text[81];
    cvt_text(str, text);
    if (*text == '\0')
        return EMPTY_NUM;
    else
        return cvtint(text);
}

```

```

void
fmtflt(char *str, double x, int width, int dec)
/*****
fmtflt converts a floating pt number to a string "999999.9999"
*****/
{
    int negative;
    int integer, fract;
    double absolute;

    static const double pow10[] =
    {1., 10., 100., 1000., 10000., 100000., 1000000., 10000000.};

    /* mark the end of string */
    str[width] = '\0';

    /* if empty value, make it be an empty field */

```

```

if (x == EMPTY_FLT)
{
    memset(str, '_', width);
    return;
}

absolute = (x < 0)? -x: x;

/* separate into integer and fractional parts */
integer = (int) absolute;
fract = (absolute - integer) * pow10[dec] + .5;

/* let the integer portion contain the sign */
if (x < 0) integer = -integer;

/* Format integer and fraction separately */
fmtint(str, integer, width-dec-1, ' ');
str[width-dec-1] = '.';
fmtint(str+width-dec, fract, dec, '0');
}

```

```

double
cvtflt(char *str)
{
    char text[81];
    char *t;
    double value;
    int div;
    int fract;
    int negative;
    int i;

    /* normalize the text */
    cvt_text(str, text);
    if (*text == '\0')
        return EMPTY_FLT;

    negative = NO;
    fract = NO;
    value = 0;
    div = 1.0;

    negative = (text[0] == '-');
    if (negative) t = text+1;
    else t = text;

    for (; *t != '\0'; t++)
    {
        if (*t == '.')
            if (fract) return INVALID_FLT;
            else fract = YES;

        else if (isdigit(*t))
            {
                value = value*10 + (int)*t - (int)'0';
                if (fract) div *= 10;
            }

        else
            return INVALID_FLT;
    }

    if (fract)
        value /= div;

    if (negative)
        value = -value;

    return value;
}

```

```

void
fmt_text(char *s, char *text, int width)
{
    /* if an empty string, then all underscores */
    if (*text == '\0')
        for (; width > 0; width--)
            *s++ = '_';

    /* otherwise, blank fill it */
    else
    {
        /* copy the text into the new buffer */
        for (; *text != '\0'; width--)
            *s++ = *text++;
    }
}

```



```

    /* fill in the rest with blanks */
    for (; width > 0; width--)
        *s++ = ' ';
    }

/* and finally, terminate the string */
*s = '\0';
}

void
cvt_text(char *s, char *text)
{
    char *lastnb;

    /* skip leading blanks and underscores */
    for (; *s == ' ' || *s == '_'; s++)
        ;

    /* copy the characters, keeping track of last blank or underscore */
    lastnb = text - 1;
    for (; *s != '\0'; *text++ = *s++)
        if (*s != ' ' && *s != '_')
            lastnb = text;

    /* truncate the text string to last nonblank character */
    *(lastnb + 1) = '\0';
}

void
fmtint(char *field, int value, int size, char fill)
/******
fmtint formats an integer value into a character field to make the integer
right-justified within the character field, padded with leading fill
characters (e.g. leading blanks if a blank is passed in for the fill argument
*****
*/
{
    int negative;
    int dividend;
    int remainder;
    char *p;

    /* create characters from right to left */
    p = field + size - 1;

    /* make note if this is a negative number */
    negative = value < 0;
    if (negative)
        value = -value;

    /* Case: Null field. Can't do anything */
    if (p < field)
        ;

    /* Case: value is zero. Print a leading '0' */
    else if (value == 0)
        *p-- = '0';

    /* Otherwise, convert each digit in turn */
    else do
    {
        dividend = value / 10;
        remainder = value - dividend * 10;
        value = dividend;

        *p-- = (char) ( (int)'0' + remainder );

    } while (p >= field && value > 0);

    /* insert a minus sign if appropriate */
    if (negative && p >= field)
        *p-- = '-';

    /* fill in leading characters */
    while (p >= field)
        *p-- = fill;
    }

int
cvtint(char *str)
/******
getint extracts an integer value from the given character field
(ex: turns the string "123" into the integer 123)
*****
*/
{
    int value;
    char c;
    int negative;
    debug("cvtint: str=%s\n", str);

```

```

negative = (*str == '-');
if (negative) str++;

/* convert the integer */
for (value = 0; isdigit(*str); str++)
    value = value * 10 + (int)(*str) - (int)'0';

/* if any non-digit characters, error */
if (*str != '\0')
    return INVALID_NUM;

/* make negative if there was a minus sign */
if (negative)
    value = -value;

debug("cvtint: value=%d\n", value);
return value;
}

```

```

void
fmt_phone(char str[20], char *phone)
{
    /* copy phone number and insert dashes 999999-999-999-9999 */
    str[0] = phone[0]; str[1] = phone[1]; str[2] = phone[2];
    str[3] = phone[3]; str[4] = phone[4]; str[5] = phone[5];
    str[6] = '-';
    str[7] = phone[6]; str[8] = phone[7]; str[9] = phone[8];
    str[10] = '-';
    str[11] = phone[9]; str[12] = phone[10]; str[13] = phone[11];
    str[14] = '-';
    str[15] = phone[12]; str[16] = phone[13]; str[17] = phone[14];
    str[18] = phone[15];
    str[19] = '\0';
}

```

```

void
fmt_zip(char str[20], char *zip)
{
    /* copy zip code and insert dashes 99999-9999 */
    str[0] = zip[0]; str[1] = zip[1]; str[2] = zip[2];
    str[3] = zip[3]; str[4] = zip[4];
    str[5] = '-';
    str[6] = zip[5]; str[7] = zip[6]; str[8] = zip[7]; str[9] = zip[8];
    str[10] = '\0';
}

```

lib/iobuf.h

```

#ifndef __TPCC_IOBUF__
#define __TPCC_IOBUF__

#include <stdlib.h>
#include <stdio.h>

/*
 * Input/Output Buffer management
 */
typedef struct {
    int ifd; /* input file descriptor */
    int ofd; /* output file descriptor */
    char *beg; /* Start of the buffer */
    char *end; /* for output buffers */
    char *max; /* Last address of the buffer */
    char *cur; /* for input buffers */
    char *data; /* data for the buffer */
} iobuf;

#define reset(b) \
    if (1) { \
        (b)->cur = (b)->end = (b)->beg; \
        *(b)->beg = '\0'; \
    } else (void)0

#define flush(b) \
    if (1) { \
        display(b); \
        reset(b); \
    } else (void)0

#define pushc(b,c) \

```

```

        if (1) {
            if ((b)->end >= (b)->max) {
                error("out_buf overflow: beg=0x%x end=%d
max=%d\n", \
                    (b)->beg, (b)->end-(b)->beg, (b)->max-(b)->beg); \
                *((b)->end++) = (c);
                *((b)->end) = '\0'; /* debug */
            }
        } else (void)0

#define popc(b) (*(b)->cur++)

extern void string(iobuf *out_buf, char *str);
extern void display(iobuf *scr);
extern void input(iobuf *scr);
extern int getkey(iobuf *in_buf, iobuf *out_buf);

#endif

```

lib/iobuf.c

```

#include <unistd.h>
#include <stdlib.h>
#include <errno.h>
#include "iobuf.h"
#include "errlog.h"

void
string(iobuf *out_buf, char str[])
{
    for (; *str != '\0'; str++) {
        pushc(out_buf, *str);
    }
}

void
display(iobuf *scr)
{
    char *p;
    int len;
    for (p = scr->beg; p < scr->end; p += len) {
        len = write(scr->ofd, p, scr->end - p);
        if (len <= 0) break;
    }
}

void
input(iobuf *scr)
{
    int len;

    /* read in as many characters as are available */
    len = read(scr->ifd, scr->end, scr->max - scr->end);

    /* if end of input, then pretend we read an END character */
    if (len == 0 || (len == -1 && errno == ECONNRESET)) {
        *scr->end = EOF;
        len = 1;
    } else if (len == -1) {
        /* Check for errors */
        perror("input(scr): unable to read stdin\n");
    }

    /* update the pointers to reflect the new data */
    scr->end += len;
    *scr->end = '\0'; /* for debugging */
}

int
getkey(iobuf *in_buf, iobuf *out_buf)
{
    if (in_buf->cur == in_buf->end) {
        flush(out_buf);
        reset(in_buf);
        input(in_buf);
    }

    return popc(in_buf);
}

```

lib/random.c

```

/*****
 * Version: A.10.10 $Date: 2005/11/14 15:34:50 $
 * Copyright 1996, Hewlett-Packard Company, all rights reserved.
 *****/
#include <stdlib.h>
#include <math.h>
#include <string.h>
#include <unistd.h>
#include "tpcc.h"
#include "random.h"

/*
 * The are all write-once, read-many variables.
 * In a multithreaded environment, they need to
 * be setup in the main process before creating
 * threads. This is done by calling InitRandomStrings().
 */
char lastNames[1000][16];

#define THREAD_LOCAL

#ifdef _REENTRANT
# ifdef HPUX
# undef THREAD_LOCAL
# define THREAD_LOCAL __thread
# endif
#endif

/* The seed value needs to be per-thread given each call to
 * the "randy" function modifies the seed. If it were not per
 * thread, then corruption could occur as multiple threads
 * access the randy function simultaneously.
 */
static THREAD_LOCAL int RandySeedIter;

static double exponential(double mean);

void
GenerateLastNames(void)
{
    int i;
    char *name;
    static const char *n[] = {"BAR", "OUGHT", "ABLE", "PRI", "PRES",
                             "ESE", "ANTI", "CALLY", "ATION", "EING"};

    for(i = 0; i < 1000; i++) {
        name = lastNames[i];
        strcpy(name, n[(i/100)%10]);
        strcat(name, n[(i/10)%10]);
        strcat(name, n[(i/1)%10]);
    }
}

ID
RandomWarehouse(ID local, ID scale, int percent)
{
    ID w_id;

    /* For the given percent of the time, pick the local warehouse */
    if (RandomNumber(1, 100) > percent || scale == 1) {
        w_id = local;
    } else {
        /* Otherwise, pick a non-local warehouse */
        w_id = RandomNumber(2, scale);
        if (w_id == local) {
            w_id = 1;
        }
    }
    return w_id;
}

void
RandomDelay(double mean, double adjust)
/*****
 * random_sleep sleeps according to the TPC specification
 *****/
{
    double secs;

    secs = exponential(mean);

    delay(secs+adjust);
}

static double

```

```

exponential(double mean)
/*****
exponential generates a reverse exponential distribution
*****/
{
    double x;

    x = -log(1.0-RandomValue()) * mean;
    return x;
}

void
SetRandomSeed(int val)
{
#ifdef USE_DRAND48
    srand48(val);
#else
    /*
     * Seed must be between 1 and 2147483646 inclusive
     * In particular, it can't be 0 or 2147483647
     */
    if (val < 1) {
        val = 1;
    } else if (val > 2147483646) {
        val = 2147483646;
    }
    RandySeedIter = val;
    randy();
#endif
}

/* Random number generator from Proceeding of the ACM */
#define RANDY_A_VAL 16807
/* 2^31 - 1 */
#define RANDY_M_VAL 2147483647
/* m / a */
#define RANDY_Q_VAL 127773
/* m % a */
#define RANDY_R_VAL 2836

double
randy(void)
{
    int hi, lo, test;

    /*
     * Make sure the seed is not zero. It could be zero if someone calls this
     * function without first initializing the seed.
     */
    if (RandySeedIter == 0) RandySeedIter = 1;

    hi = RandySeedIter / RANDY_Q_VAL;
    lo = RandySeedIter % RANDY_Q_VAL;

    test = (RANDY_A_VAL * lo) - (RANDY_R_VAL * hi);
    RandySeedIter = (test > 0) ? test + RANDY_M_VAL;

    return((double)RandySeedIter / (double)RANDY_M_VAL);
} /* end of fn randy */

/*****
*****/
/* double RandomValue(void) */
/*
/* return a random value in the range [0.0 .. 1.0) */
/*
*****/
double
RandomValue(void)
{
#ifdef USE_DRAND48
    return drand48();
#else
    return randy();
#endif
}

```

lib/Makefile

```

****
# @(#) Version: A.10.10 $Date: 2002/12/10 14:23:24 $
#
#(c) Copyright 1996, Hewlett-Packard Company, all rights reserved.
****

```

```

CFLAGS= -DHOTWARE -D_REENTRANT $(BUILDFLAGS)

utils=delay.o errlog.o fmt.o random.o tas.o date.o spinlock.o iobuf.o
load_files=random_load.o

all: tpc_lib.a load_lib.a

tpc_lib.a: ${utils}
    rm -f tpc_lib.a
    ar -r tpc_lib.a ${utils}

load_lib.a: ${load_files}
    rm -f load_lib.a
    ar -r load_lib.a ${load_files}

clean:
    rm -f *.o
    rm -f *.a

clobber: clean

.s.o:
    cc $(BUILDFLAGS) -c $*.s

```

A.3 Transaction Source

client/service.c

```

/*****
****
@(#) Version: A.10.10 $Date: 2005/04/11 10:11:31 $

(c) Copyright 1996, Hewlett-Packard Company, all rights reserved.
****
****/

#include <unistd.h>
#include <sys/types.h>
#include "tpcc.h"
#include "atmi.h"

int userid;
char *cmd = NULL;

int tpsvritn(argc, argv)
int argc;
char **argv;
{
    char c;
    int ret;
    time_t t;

    /* Configure the error log */
    configure_error_log();

    t = time((time_t *) NULL);
    userlog("starting up at time %s", ctime(&t));
    /*
     * search for the options
     * "-S" server program
     * purpose: to get svr_id & progname for DVRY_LOG files
     */
    while ((c = getopt(argc, argv, "S:h:")) != EOF) {
        switch(c) {
            case 'S':
                cmd = optarg;
                break;
        }
    }

    /* We just use a single file for delivery */
    userid=0;
    ret = transaction_begin(userid);
    dvry_open(userid);

    return 0;
}

void NEWO_SVC(svcinfo)
TPSVCINFO *svcinfo;
{

```

```

neworder_transaction((neworder_trans *)svcinfo->data);
treturn(TPSUCCESS, 0, svcinfo->data, svcinfo->len, 0);
}

void PMT_SVC(svcinfo)
TPSVINFO *svcinfo;
{
    payment_transaction((payment_trans *)svcinfo->data);
    treturn(TPSUCCESS, 0, svcinfo->data, svcinfo->len, 0);
}

void ORDS_SVC(svcinfo)
TPSVINFO *svcinfo;
{
    ordstat_transaction((ordstat_trans *)svcinfo->data);
    treturn(TPSUCCESS, 0, svcinfo->data, svcinfo->len, 0);
}

void STKL_SVC(svcinfo)
TPSVINFO *svcinfo;
{
    stocklev_transaction((stocklev_trans *)svcinfo->data);
    treturn(TPSUCCESS, 0, svcinfo->data, svcinfo->len, 0);
}

void DVRY_SVC(svcinfo)
TPSVINFO *svcinfo;
{
    delivery_trans *t = (delivery_trans *)svcinfo->data;
    gettimeofday(t->del.deque, NULL);
    delivery_transaction(t);
    gettimeofday(t->del.complete, NULL);
    post_dvry(t);

    /* Why do we return things ? */
    treturn(TPSUCCESS, 0, svcinfo->data, svcinfo->len, 0);
}

/*****
tpsrdone cleans up after the TPC transaction service
*****/
void tpsrdone()
{
    transaction_done();
    dvry_close();

    /* Log a message saying we are done */
    userlog("TUXEDO service %s has shutdown\n", cmd);
}

```

client/oracle/transaction.c

```

#include "ora_tpcc.h"
#include <time.h>
#include "tpcc.h"

/* Always use plsql for delivery. */
#define PLSQLEDEL

#ifdef __STDC__
#include "ociapr.h"
#else
#include "ocikpr.h"
#endif

extern TPCinit(int, char*, char*);

int numtrans = 0;

void
transaction_done (void)
{
    /* fprintf(stderr, "About to call TPCexit\n"); fflush(stderr); */
    TPCexit();
    /* fprintf(stderr, "TPCexit after %d transcatons \n",
numtrans); fflush(stderr); */
}

```

```

/* void */
int
transaction_begin(int id)
{
    int ret;

    if ((ret=TPCinit(id, "tpcc", "tpcc")) == -1)
    {
        fprintf(stderr, "TPCinit failure!\n"); fflush(stderr);
        /* Error */
    }
    numtrans = 0;
    return ret;
}

void
neworder_transaction(neworder_trans *str)
{
    int i;
    struct newstruct ora_str;

    ora_str.newin.w_id = str->W_ID;
    ora_str.newin.d_id = str->D_ID;
    ora_str.newin.c_id = str->C_ID;
    for (i = 0; i < str->o_ol_cnt; i++) {
        ora_str.newin.ol_i_id[i] = str->item[i].OL_I_ID;
        ora_str.newin.ol_supply_w_id[i] = str->item[i].OL_SUPPLY_W_ID;
        ora_str.newin.ol_quantity[i] = str->item[i].OL_QUANTITY;
    }
    for (i = str->o_ol_cnt; i < 15; i++) {
        ora_str.newin.ol_i_id[i] = 0;
        ora_str.newin.ol_supply_w_id[i] = 0;
        ora_str.newin.ol_quantity[i] = 0;
    }

    numtrans++;
    if (TPCnew(&ora_str) == -1) {
        str->status = E_DB_ERROR;
        return;
    } else {
        str->status = OK;
    }

    str->o_ID = ora_str.newout.o_id;
    str->o_ol_cnt = ora_str.newout.o_ol_cnt;
    strncpy(str->c_last, ora_str.newout.c_last, 17);
    strncpy(str->c_credit, ora_str.newout.c_credit, 3);
    str->c_discount = (REAL) ora_str.newout.c_discount;
    str->w_tax = (REAL) ora_str.newout.w_tax;
    str->d_tax = (REAL) ora_str.newout.d_tax;
    strncpy(str->o_entry_d, ora_str.newout.o_entry_d, 20);
    for (i = 0; i < ora_str.newout.o_ol_cnt; i++) {
        strncpy(str->item[i].l_name, ora_str.newout.l_name[i], 25);
        str->item[i].s_quantity = ora_str.newout.s_quantity[i];
        str->item[i].brand_generic = ora_str.newout.brand_generic[i];
        str->item[i].l_price = (MONEY) ora_str.newout.l_price[i]*100.0; /* needs to be in
cents */
    }
    str->status = ((ora_str.newout.status[0] != '\0') ?
E_INVALID_ITEM : OK);
}

/*****
***
* Payment Query
*****/

void
payment_transaction(payment_trans *str)
{
    int i;

    struct paystruct ora_str;

    ora_str.payin.w_id = str->W_ID;
    ora_str.payin.d_id = str->D_ID;
    ora_str.payin.c_w_id = str->C_W_ID;
    ora_str.payin.c_d_id = str->C_D_ID;
    ora_str.payin.h_amount = (float) str->h_amount; /* Amount in cents */
    ora_str.payin.bylastname = str->byname;
    if (ora_str.payin.bylastname) {
        ora_str.payin.c_id = 0;
        strncpy(ora_str.payin.c_last, str->c_last, 17);
        ora_str.payin.c_last[16] = '\0';
        for (i = 15; (i >= 0) && (ora_str.payin.c_last[i] == ' '); i--)
            ora_str.payin.c_last[i] = '\0';
    }
    else {
        ora_str.payin.c_id = str->c_id;
        strcpy(ora_str.payin.c_last, "");
    }
}

```

```

retries = 0;

numtrans++;
if (TPCpay (&ora_str)) {
    str->status = E_DB_ERROR;
    return;
} else {
    str->status = OK;
}

strcpy (str->W_STREET_1, ora_str.payout.w_street_1, 21);
strcpy (str->W_STREET_2, ora_str.payout.w_street_2, 21);
strcpy (str->W_CITY, ora_str.payout.w_city, 21);
strcpy (str->W_STATE, ora_str.payout.w_state, 3);
strcpy (str->W_ZIP, ora_str.payout.w_zip, 10);
strcpy (str->D_STREET_1, ora_str.payout.d_street_1, 21);
strcpy (str->D_STREET_2, ora_str.payout.d_street_2, 21);
strcpy (str->D_CITY, ora_str.payout.d_city, 21);
strcpy (str->D_STATE, ora_str.payout.d_state, 3);
strcpy (str->D_ZIP, ora_str.payout.d_zip, 10);
str->C_ID = ora_str.payout.c_id;
strcpy (str->C_FIRST, ora_str.payout.c_first, 17);
strcpy (str->C_MIDDLE, ora_str.payout.c_middle, 3);
strcpy (str->C_LAST, ora_str.payout.c_last, 17);
strcpy (str->C_STREET_1, ora_str.payout.c_street_1, 21);
strcpy (str->C_STREET_2, ora_str.payout.c_street_2, 21);
strcpy (str->C_CITY, ora_str.payout.c_city, 21);
strcpy (str->C_STATE, ora_str.payout.c_state, 3);
strcpy (str->C_ZIP, ora_str.payout.c_zip, 10);
strcpy (str->C_PHONE, ora_str.payout.c_phone, 17);
strcpy (str->C_SINCE, ora_str.payout.c_since, 11);

strcpy (str->C_CREDIT, ora_str.payout.c_credit, 3);
str->C_CREDIT_LIM = (MONEY)
ora_str.payout.c_credit_lim*100.0; /* needs to be in cents */
str->C_DISCOUNT = (REAL) ora_str.payout.c_discount;
str->C_BALANCE = (REAL) ora_str.payout.c_balance*100.0; /*
needs to be in cents */
/* Oracle passes 201 characters, we copy 200 and terminate on
201.
*/
strcpy (str->C_DATA, ora_str.payout.c_data, 200);
str->C_DATA[200] = '\0';
strcpy (str->H_DATE, ora_str.payout.h_date, 20);
}

void
ordstat_transaction(ordstat_trans *str)
{
    int i;

    struct ordstruct ora_str;

    ora_str.ordin.w_id = str->W_ID;
    ora_str.ordin.d_id = str->D_ID;
    ora_str.ordin.bylastname = str->byname;
    if (ora_str.ordin.bylastname) {
        ora_str.ordin.c_id = 0;
        strcpy (ora_str.ordin.c_last, str->C_LAST, 17);
        ora_str.ordin.c_last[16] = '\0';
        for (i = 15; (i >= 0) && (ora_str.ordin.c_last[i] == ' '));
        i--
        ora_str.ordin.c_last[i] = '\0';
    }
    else {
        ora_str.ordin.c_id = str->C_ID;
        strcpy (ora_str.ordin.c_last, " ");
    }
    retries = 0;

    numtrans++;
    if (TPCord (&ora_str)) {
        str->status = ora_str.ordout.terror;
        if (ora_str.ordin.bylastname) {
            message("Order status error: wid = %d, did = %d, name =
%s\n", str->W_ID, str->D_ID, ora_str.ordin.c_last);
        } else {
            message("Order status error: wid = %d, did = %d, ID =
%d\n", str->W_ID, str->D_ID, str->C_ID);
        }
        return;
    } else {
        str->status = OK;
    }

    str->C_ID = ora_str.ordout.c_id;
    strcpy (str->C_LAST, ora_str.ordout.c_last, 17);
    strcpy (str->C_FIRST, ora_str.ordout.c_first, 17);
    strcpy (str->C_MIDDLE, ora_str.ordout.c_middle, 3);
    str->C_BALANCE = (MONEY) ora_str.ordout.c_balance*100.0; /*
needs to be in cents */
    str->O_ID = ora_str.ordout.o_id;
    strcpy (str->O_ENTRY_DATE, ora_str.ordout.o_entry_d, 20);
    str->O_CARRIER_ID = ora_str.ordout.o_carrier_id;

    str->o_cnt = ora_str.ordout.o_cnt;
    for (i = 0; i < ora_str.ordout.o_cnt; i++) {
        str->item[i].OL_SUPPLY_W_ID = ora_str.ordout.ol_supply_w_id[i];
        str->item[i].OL_I_ID = ora_str.ordout.ol_i_id[i];
        str->item[i].OL_QUANTITY = ora_str.ordout.ol_quantity[i];
        str->item[i].OL_AMOUNT = (MONEY)
ora_str.ordout.ol_amount[i]*100.0; /* needs to be in cents
*/
        strcpy (str->item[i].OL_DELIVERY_DATE,
ora_str.ordout.ol_delivery_d[i], 11);
    }

    /*
    * Delivery Query
    */
    /*
    *****
    *****/

    void
    delivery_transaction(delivery_trans *str)
    {
        double tr_end;
        int i;

        struct delstruct ora_str;

        /* set plsql or OCI delivery */
        #ifdef PLSQDEL
        ora_str.delin.plsqlflag=1;
        #else
        ora_str.delin.plsqlflag=0;
        #endif

        ora_str.delin.w_id = str->del.W_ID;
        ora_str.delin.o_carrier_id = str->del.O_CARRIER_ID;
        retries = 0;

        numtrans++;
        if (TPCdel (&ora_str)) {
            str->del.status = E_DB_ERROR;
            return;
        } else {
            str->del.status = OK;
        }

        for (i = 0; i < 10; i++) {
            if (del_o_id[i] <= 0) {
                str->del.order[i].status = E_NOT_ENOUGH_ORDERS;
            } else {
                str->del.order[i].status = OK;
                str->del.order[i].O_ID = del_o_id[i];
            }
        }
    }

    /*
    *****
    *****/

    * Stock Level Query

    void
    stocklev_transaction(stocklev_trans *str)
    {

        struct stostruct ora_str;
        ora_str.stoin.w_id = str->W_ID;
        ora_str.stoin.d_id = str->D_ID;
        ora_str.stoin.threshold = str->threshold;
        retries = 0;

        numtrans++;
        if (TPCsto (&ora_str)) {
            str->status = E_DB_ERROR;
            return;
        } else {
            str->status = OK;
        }
        str->low_stock = ora_str.stoout.low_stock;
    }
}

```

client/oracle/tpccpl.c

```

#ifdef RCSID
static char *RCSid =
"$Header: tpccpl.c,v 1.4 2003/07/01 15:42:13 mliu Exp $ Copyr (c) 1994 Oracle";
#endif /* RCSID */

```

```

/*=====
==+
| Copyright (c) 1994 Oracle Corp, Redwood Shores, CA |
| OPEN SYSTEMS PERFORMANCE GROUP |
| All Rights Reserved |
=====
==+
| FILENAME
| tpccpl.c
| DESCRIPTION
| TPC-C transactions in PL/SQL.
=====
==*/

#include <stdio.h>
#include <time.h>
#include "ora_tpcc.h"
#ifdef TUX
#include <userlog.h>
#else
#include <stdarg.h>
#endif

#define SQLTXT "alter session set isolation_level = serializable"
#define SQLTXTTRC "alter session set sql_trace = true"
#define SQLTXTTIM "alter session set timed_statistics = true"

FILE *fopen ();
#ifdef ORA_NT
#undef boolean
#include "dpbcore.h"
#define gettime dpbtimef
#else
extern double gettime ();
#endif
int proc_no = 0;
static int logon = 0;
static int new_init = 0;
static int pay_init = 0;
static int ord_init = 0;
static int del_init_oci = 0;
static int del_init_plsql = 0;
static int sto_init = 0;
static int res_init = 0;

int execstatus;
int errcode;

OCIEnv *tpcenv;
OCIServer *tpcsrv;
OCIError *errhp;
OCISvcCtx *tpcsvc;
OCISession *tpcsusr;
OCISmt *curi;

/* for stock-level transaction */

int w_id;
int d_id;
int c_id;
int threshold;
int low_stock;

/* for delivery transaction */

int del_o_id[10];
int retries;

/* for order-status transaction */

int bylastname;
char c_last[17];
char c_first[17];
char c_middle[3];
double c_balance;
int o_id;
text o_entry_d[20];
ub4 datelen;
int o_carrier_id;
int o_ol_cnt;
int ol_supply_w_id[15];
int ol_i_id[15];
int ol_quantity[15];
int ol_amount[15];
ub4 ol_del_len[15];
text ol_delivery_d[15][11];
/* xnie - begin */
OCIRowid *o_rowid;
/* xnie - end */

/* for payment transaction */

int c_w_id;
int c_d_id;
int h_amount;
char w_street_1[21];
char w_street_2[21];
char w_city[21];
char w_state[3];
char w_zip[10];
char d_street_1[21];
char d_street_2[21];
char d_city[21];
char d_state[3];
char d_zip[10];
char c_street_1[21];
char c_street_2[21];
char c_city[21];
char c_state[3];
char c_zip[10];
char c_phone[17];
ub4 sincelen;
text c_since_d[11];
float c_discount;
char c_credit[3];
int c_credit_lim;
char c_data[201];
ub4 hlen;
text h_date[20];

/* for new order transaction */

int nol_i_id[15];
int nol_supply_w_id[15];
int nol_quantity[15];
int nol_quantity10[15];
int nol_quantity91[15];
int nol_ytdqty[15];
int nol_amount[15];
int o_all_local;
float w_tax;
float d_tax;
float total_amount;
char i_name[15][25];
int s_quantity[15];
char brand_gen[15];
int i_price[15];
char brand_generic[15][1];
int status;
int tracelevel = 0;

OCIDate cr_date;
OCIDate c_since;
OCIDate o_entry_d_base;
OCIDate ol_d_base[15];
dvoid *xmem;

#ifdef AVOID_DEADLOCK
int indx[NITEMS], ordl_cnt;
void swap(struct newstruct *str, int i, int j);
void q_sort(int *arr, struct newstruct *str, int left, int right);
#endif

/*
extern char oracle_home[256];
*/

/* NewOrder Binding stuff */

#ifdef TUX
void userlog (char* fntp, ...)
{
va_list va;
va_start(va, fntp);
vfprintf(stderr, fntp, va);
va_end(va);
}
#endif

/* vmm313 void ocierror(fname, lineno, errhp, status) */
int ocierror(fname, lineno, errhp, status)
char *fname;
int lineno;
OCIError *errhp;
sword status;
{
text errbuf[512];
sb4 errcode;
sb4 lstat;
ub4 recno=2;

switch (status) {
case OCI_SUCCESS:

```

```

break;
case OCI_SUCCESS_WITH_INFO:
    fprintf(stderr,"Module %s Line %d\n", fname, lineno);
    fprintf(stderr,"Error - OCI_SUCCESS_WITH_INFO\n");
    lstat = OCIErrorGet (errhp, recno++, (text *) NULL, &errcode, errbuf,
        (ub4) sizeof(errbuf), OCI_HTYPE_ERROR);
    fprintf(stderr,"Error - %s\n", errbuf);
    break;
case OCI_NEED_DATA:
    fprintf(stderr,"Module %s Line %d\n", fname, lineno);
    fprintf(stderr,"Error - OCI_NEED_DATA\n");
    return (IRRECERR);
case OCI_NO_DATA:
    fprintf(stderr,"Module %s Line %d\n", fname, lineno);
    fprintf(stderr,"Error - OCI_NO_DATA\n");
    return (IRRECERR);
case OCI_ERROR:
    lstat = OCIErrorGet (errhp, (ub4) 1,
        (text *) NULL, &errcode, errbuf,
        (ub4) sizeof(errbuf), OCI_HTYPE_ERROR);

if (errcode == NOT_SERIALIZABLE) return (errcode);
if (errcode == SNAPSHOT_TOO_OLD) return (errcode);
while (lstat != OCI_NO_DATA)
{
    fprintf(stderr,"Module %s Line %d\n", fname, lineno);
    fprintf(stderr,"Error - %s\n", errbuf);
    lstat = OCIErrorGet (errhp, recno++, (text *) NULL, &errcode, errbuf,
        (ub4) sizeof(errbuf), OCI_HTYPE_ERROR);
}
return (errcode);
/* vmm313   TPCexit(1); */
/* vmm313   exit(1); */
case OCI_INVALID_HANDLE:
    fprintf(stderr,"Module %s Line %d\n", fname, lineno);
    fprintf(stderr,"Error - OCI_INVALID_HANDLE\n");
    TPCexit(1);
    exit(-1);
case OCI_STILL_EXECUTING:
    fprintf(stderr,"Module %s Line %d\n", fname, lineno);
    fprintf(stderr,"Error - OCI_STILL_EXECUTE\n");
    return (IRRECERR);
case OCI_CONTINUE:
    fprintf(stderr,"Module %s Line %d\n", fname, lineno);
    fprintf(stderr,"Error - OCI_CONTINUE\n");
    return (IRRECERR);
default:
    fprintf(stderr,"Module %s Line %d\n", fname, lineno);
    fprintf(stderr,"Status - %s\n", status);
    return (IRRECERR);
}
return (RECOVERR);
}

FILE *vopen(fnam,mode)
char *fnam;
char *mode;
{
FILE *fd;

#ifdef DEBUG
fprintf(stderr, "tkvuopen() fnam: %s, mode: %s\n", fnam, mode);
#endif

fd = fopen((char *)fnam,(char *)mode);
if (!fd){
    fprintf(stderr, "fopen on %s failed %d\n",fnam,fd);
    exit(-1);
}
return(fd);
}

int sqlfile(fnam,linebuf)
char *fnam;
text *linebuf;
{
FILE *fd;
int nulpt = 0;
char realfile[512];

#ifdef DEBUG
fprintf(stderr, "sqlfile() fnam: %s, linebuf: %#x\n", fnam, linebuf);
#endif

/*
printf(realfile,"%s/bench/tpc/tpcc/blocks/%s",oracle_home,fnam);
*/
sprintf(realfile,"project/tpcc/blocks/%s",fnam);
/* sprintf(realfile,"%s",fnam); */
fd = vopen(realfile,"r");
while (fgets((char *)linebuf+nulpt, SQL_BUF_SIZE,fd))
{
    nulpt = strlen((char *)linebuf);
}
return(nulpt);
}

```

```

}

#ifdef NOT
void vgetdate (unsigned char *oradt)
{
    struct tm *loctime;
    time_t int_time;

    struct ORADATE {
        unsigned char century;
        unsigned char year;
        unsigned char month;
        unsigned char day;
        unsigned char hour;
        unsigned char minute;
        unsigned char second;
    } Date;
    int century;
    int cnvrtOK;

    /* assume convert is successful */
    cnvrtOK = 1;

    /* get the current date and time as an integer */
    time(&int_time);

    /* Convert the current date and time into local time */
    loctime = localtime(&int_time);

    century = (1900+loctime->tm_year) / 100;

    Date.century = (unsigned char)(century + 100);
    if (Date.century < 119 || Date.century > 120) cnvrtOK = 0;
    Date.year = (unsigned char)(loctime->tm_year+100);
    if (Date.year < 100 || Date.year > 199) cnvrtOK = 0;
    Date.month = (unsigned char)(loctime->tm_mon + 1);
    if (Date.month < 1 || Date.month > 12) cnvrtOK = 0;
    Date.day = (unsigned char)loctime->tm_mday;
    if (Date.day < 1 || Date.day > 31) cnvrtOK = 0;
    Date.hour = (unsigned char)(loctime->tm_hour + 1);
    if (Date.hour < 1 || Date.hour > 24) cnvrtOK = 0;
    Date.minute = (unsigned char)(loctime->tm_min + 1);
    if (Date.minute < 1 || Date.minute > 60) cnvrtOK = 0;
    Date.second = (unsigned char)(loctime->tm_sec + 1);
    if (Date.second < 1 || Date.second > 60) cnvrtOK = 0;

    if (cnvrtOK)
        memcpy(oradt,&Date,7);
    else
        *oradt = '\0';

    return;
}

void cvtdmy (unsigned char *oradt, char *outdate)
{
    struct ORADATE {
        unsigned char century;
        unsigned char year;
        unsigned char month;
        unsigned char day;
        unsigned char hour;
        unsigned char minute;
        unsigned char second;
    } Date;

    int day,month,year;

    memcpy(&Date,oradt,7);

    year = (Date.century-100)*100 + Date.year-100;
    month = Date.month;
    day = Date.day;
    sprintf(outdate,"%02d-%02d-%04d",day,month,year);

    return;
}

void cvtdmyhms (unsigned char *oradt, char *outdate)
{
    struct ORADATE {
        unsigned char century;
        unsigned char year;
        unsigned char month;
        unsigned char day;
        unsigned char hour;
        unsigned char minute;
        unsigned char second;
    } Date;
}

```

```

int day,month,year;
int hour,min,sec;

memcpy(&Date,oradt,7);

year = (Date.century-100)*100 + Date.year-100;
month = Date.month;
day = Date.day;
hour = Date.hour - 1;
min = Date.minute - 1;
sec = Date.second - 1;

sprintf(outdate,"%02d-%02d-%4d %02d:%02d:%02d\0",
        day,month,year,hour,min,sec);

return;
}
#endif

void TPCexit (void)
{
if (new_init) {
tkvcndone();
new_init = 0;
}
if (pay_init) {
tkvcpdone();
pay_init = 0;
}
if (ord_init) {
tkvcodone();
ord_init = 0;
}
if (del_init_oci) {
tkvcddone(0);
del_init_oci = 0;
}
if (del_init_plsql) {
tkvcddone(1);
del_init_plsql = 0;
}
if (sto_init) {
tkvcddone();
sto_init = 0;
}

OCIHandleFree((dvoid *)tpcusr, OCI_HTYPE_SESSION);
OCIHandleFree((dvoid *)tpcscv, OCI_HTYPE_SVCCTX);
OCIHandleFree((dvoid *)errhp, OCI_HTYPE_ERROR);
OCIHandleFree((dvoid *)tpcsrv, OCI_HTYPE_SERVER);
OCIHandleFree((dvoid *)tpcenv, OCI_HTYPE_ENV);
}

TPCinit (id, uid, pwd)

int id;
char *uid;
char *pwd;

{

char filename[40];
text stmbuf[100];

OCIInitialize(OCI_DEFAULT|OCI_OBJECT,(dvoid *)0,0,0,0);
OCIEnvInit(&tpcenv, OCI_DEFAULT, 0, (dvoid **)0);
OCIHandleAlloc((dvoid *)tpcenv, (dvoid **)&tpcsrv, OCI_HTYPE_SERVER, 0, (dvoid
**)0);
OCIHandleAlloc((dvoid *)tpcenv, (dvoid **)&errhp, OCI_HTYPE_ERROR, 0, (dvoid
**)0);
OCIHandleAlloc((dvoid *)tpcenv, (dvoid **)&tpcscv, OCI_HTYPE_SVCCTX, 0, (dvoid
**)0);
OCIServerAttach(tpcsrv, errhp, (text *)0,0,OCI_DEFAULT);
OCIAttrSet((dvoid *)tpcscv, OCI_HTYPE_SVCCTX, (dvoid *)tpcsrv,
(ub4)0,OCI_ATTR_SERVER, errhp);
OCIHandleAlloc((dvoid *)tpcenv, (dvoid **)&tpcusr, OCI_HTYPE_SESSION, 0, (dvoid
**)0);
OCIAttrSet((dvoid *)tpcusr, OCI_HTYPE_SESSION, (dvoid *)uid,
(ub4)strlen(uid),OCI_ATTR_USERNAME, errhp);
OCIAttrSet((dvoid *)tpcusr, OCI_HTYPE_SESSION, (dvoid *)pwd, (ub4)strlen(pwd),
OCI_ATTR_PASSWORD, errhp);
OCIERROR(errhp, OCISessionBegin(tpcscv, errhp, tpcusr, OCI_CRED_RDBMS,
OCI_DEFAULT));

OCIAttrSet(tpcscv, OCI_HTYPE_SVCCTX, tpcusr, 0, OCI_ATTR_SESSION, errhp);

/* run all transaction in serializable mode */

```

```

OCIHandleAlloc(tpcenv, (dvoid **)&curi, OCI_HTYPE_STMT, 0, (dvoid**)0);
sprintf ((char *) stmbuf, SQLTXT);
OCIStmtPrepare(curi, errhp, stmbuf, strlen((char *)stmbuf), OCI_NTV_SYNTAX,
OCI_DEFAULT);
OCIERROR(errhp,OCIStmtExecute(tpcscv, curi, errhp,1,0,0,OCI_DEFAULT));
OCIHandleFree(curi, OCI_HTYPE_STMT);

/*
This is done in cvdrv.c
if (tracelevel == 2) {
OCIHandleAlloc(tpcenv, (dvoid **)&curi, OCI_HTYPE_STMT, 0, (dvoid**)0);
memset(stmbuf,0,100);
sprintf ((char *) stmbuf, SQLTXTTRC);
OCIStmtPrepare(curi, errhp, stmbuf, strlen((char *)stmbuf),
OCI_NTV_SYNTAX, OCI_DEFAULT);
OCIERROR(errhp, OCIStmtExecute(tpcscv, curi, errhp,1,0,0,OCI_DEFAULT));
OCIHandleFree((dvoid *)curi, OCI_HTYPE_STMT);
}
*/
if (tracelevel == 3) {
OCIHandleAlloc(tpcenv, (dvoid **)&curi, OCI_HTYPE_STMT, 0, (dvoid**)0);
memset(stmbuf,0,100);
sprintf ((char *) stmbuf, SQLTXTTIM);
OCIStmtPrepare(curi, errhp, stmbuf, strlen((char *)stmbuf),
OCI_NTV_SYNTAX, OCI_DEFAULT);
OCIERROR(errhp, OCIStmtExecute(tpcscv, curi, errhp,1,0,0,OCI_DEFAULT));
OCIHandleFree((dvoid *)curi, OCI_HTYPE_STMT);
}

logon = 1;

OCIERROR(errhp,OCIDateSysDate(errhp,&cr_date));

if (tkvcninit ()) { /* new order */
TPCexit ();
return (-1);
}
else
new_init = 1;

if (tkvcpinit ()) { /* payment */
TPCexit ();
return (-1);
}
else
pay_init = 1;

if (tkvcoinit ()) { /* order status */
TPCexit ();
return (-1);
}
else
ord_init = 1;

if (tkvcddinit (0)) { /* delivery */
TPCexit ();
return (-1);
}
else
del_init_oci = 1;

if (tkvcddinit (1)) { /* delivery */
TPCexit ();
return (-1);
}
else
del_init_plsql = 1;

if (tkvcsinit ()) { /* stock level */
TPCexit ();
return (-1);
}
else
sto_init = 1;

return (0);
}

TPCnew (str)
struct newstruct *str;
{
int i;

w_id = str->newin.w_id;
d_id = str->newin.d_id;
c_id = str->newin.c_id;
for (i = 0; i < 15; i++) {
nol_i_id[i] = str->newin.ol_i_id[i];

```



```

    nol_supply_w_id[i] = str->newin.ol_supply_w_id[i];
    nol_quantity[i] = str->newin.ol_quantity[i];
}
retries = 0;
#endifdef AVOID_DEADLOCK
for (i = NITEMS; i > 0; i--) {
    if (nol_i_id[i-1] > 0) {
        ordl_cnt = i;
        break;
    }
}

for (i = 0; i < NITEMS; i++) indx[i] = i;

q_sort(nol_i_id, str, 0, ordl_cnt-1);

#endif
/*
vgetdate(cr_date); */

OCIERROR(errhp,OCIDateSysDate(errhp,&cr_date));

if (str->newout.terror = tkvcn ()) {
    if (str->newout.terror != RECOVERR)
        str->newout.terror = IRRECERR;
    return (-1);
}

/* fill in date for o_entry_d from time in beginning of txn*/
/*
cvtdmyhms(cr_date,o_entry_d);
*/
datelen = sizeof(o_entry_d);
OCIERROR(errhp,
    OCIDateToText(errhp,&cr_date,(text*)FULLDATE,SIZ(FULLDATE),(text*)0,0,
        &datelen,o_entry_d));

str->newout.terror = NOERR;
str->newout.o_id = o_id;
str->newout.o_ol_cnt = o_ol_cnt;
strncpy (str->newout.c_last, c_last, 17);
strncpy (str->newout.c_credit, c_credit, 3);
str->newout.c_discount = c_discount;
str->newout.w_tax = (float)(w_tax);
str->newout.d_tax = (float)(d_tax);
strncpy (str->newout.o_entry_d, (char*)o_entry_d, 20);
str->newout.total_amount = total_amount;
for (i = 0; i < o_ol_cnt; i++) {
    strncpy (str->newout.i_name[i], i_name[i], 25);
    str->newout.s_quantity[i] = s_quantity[i];
    str->newout.brand_generic[i] = brand_generic[i][0];
    str->newout.i_price[i] = (float)i_price[i]/100;
    str->newout.ol_amount[i] = (float)nol_amount[i]/100;
}

#endifdef AVOID_DEADLOCK
    q_sort(indx, str, 0, ordl_cnt-1);
#endif

if (status)
    strcpy (str->newout.status, "Item number is not valid");
else
    str->newout.status[0] = '\0';
str->newout.retry = retries;
#if defined(TOP) || defined(TUX) /* changed mjb 17 feb for tuxedo */
return(1);
#else
return (0);
#endif
}

TPCpay (str)

struct paystruct *str;

{
    w_id = str->payin.w_id;
    d_id = str->payin.d_id;
    c_w_id = str->payin.c_w_id;
    c_d_id = str->payin.c_d_id;
    h_amount = str->payin.h_amount;
    bylastname = str->payin.bylastname;

/*
vgetdate(cr_date); */
OCIERROR(errhp,OCIDateSysDate(errhp,&cr_date));

if (bylastname) {
        c_id = 0;
        strcpy (c_last, str->payin.c_last, 17);
    }
    else {
        c_id = str->payin.c_id;
        strcpy (c_last, " ");
    }
    retries = 0;

    if (str->ordout.terror = tkvco ()) {
        if (str->ordout.terror != RECOVERR)
            str->ordout.terror = IRRECERR;
        return (-1);
    }
}

TPCord (str)

struct ordstruct *str;

{
    int i;
    w_id = str->ordin.w_id;
    d_id = str->ordin.d_id;
    bylastname = str->ordin.bylastname;
    if (bylastname) {
        c_id = 0;
        strcpy (c_last, str->ordin.c_last, 17);
    }
    else {
        c_id = str->ordin.c_id;
        strcpy (c_last, " ");
    }
    retries = 0;

    if (str->ordout.terror = tkvco ()) {
        if (str->ordout.terror != RECOVERR)
            str->ordout.terror = IRRECERR;
        return (-1);
    }
}

```

```

datelen = sizeof(o_entry_d);
OCIERROR(errhp,
OCIDateToText(errhp,&o_entry_d_base,(text*)FULLDATE,SIZ(FULLDATE),(text*)0,0,
&datelen,o_entry_d));

```

```

str->ordout.terror = NOERR;
str->ordout.c_id = c_id;
strncpy (str->ordout.c_last, c_last, 17);
strncpy (str->ordout.c_first, c_first, 17);
strncpy (str->ordout.c_middle, c_middle, 3);
str->ordout.c_balance = c_balance/100;
str->ordout.o_id = o_id;
strncpy (str->ordout.o_entry_d, (char*)o_entry_d, 20);
if ( o_carrier_id == 11 )
str->ordout.o_carrier_id = 0;
else
str->ordout.o_carrier_id = o_carrier_id;
str->ordout.o_ol_cnt = o_ol_cnt;
for (i = 0; i < o_ol_cnt; i++) {
ol_delivery_d[i][10] = '\0';
if ( !strcmp(char*ol_delivery_d[i],"15-09-1911") )
strncpy(char*ol_delivery_d[i],"NOT DELIVR",10);
str->ordout.ol_supply_w_id[i] = ol_supply_w_id[i];
str->ordout.ol_i_id[i] = ol_i_id[i];
str->ordout.ol_quantity[i] = ol_quantity[i];
str->ordout.ol_amount[i] = (float)ol_amount[i]/100;
strncpy (str->ordout.ol_delivery_d[i], (char*)ol_delivery_d[i], 11);
}
str->ordout.retry = retries;
#if defined(TOP) || defined(TUX)
return(1);
#else
return (0);
#endif
}

```

TPCdcl (str)

```

struct delstruct *str;

```

```

{
double tr_end;
int i;

w_id = str->delin.w_id;
o_carrier_id = str->delin.o_carrier_id;
retries = 0;
/*
vgetdate(cr_date);
OCIERROR(errhp,OCIDateSysDate(errhp,&cr_date));

if (str->delout.terror = tkvcd (str->delin.psqlflag)) {
if(str->delout.terror == DEL_ERROR)
return DEL_ERROR;
if (str->delout.terror != RECOVERR)
str->delout.terror = IRRECERR;
return (-1);
}

str->delout.terror = NOERR;
str->delout.retry = retries;
#if defined(TOP) || defined(TUX) /* changed mjb 17 feb */
return(1);
#else
return (0);
#endif
}

```

TPCsto (str)

```

struct stostruct *str;

```

```

{
w_id = str->stoin.w_id;
d_id = str->stoin.d_id;
threshold = str->stoin.threshold;
retries = 0;

if (str->stoout.terror = tkvcs ()) {
if (str->stoout.terror != RECOVERR)
str->stoout.terror = IRRECERR;
return (-1);
}

str->stoout.terror = NOERR;

```

```

str->stoout.low_stock = low_stock;
str->stoout.retry = retries;
#if defined(TOP) || defined(TUX) /* changed mjb 17 feb */
return(1);
#else
return (0);
#endif
}

```

```

#endif AVOID_DEADLOCK

```

```

void q_sort(int *arr,struct newstruct *str,int left, int right)
{
int i, last;

```

```

if(left >= right)
return;
swap(str,left,(left+right)/2);
last = left;
for(i=left+1;i<=right;i++)
if(arr[i] < arr[left])
swap(str,last,i);
swap(str,last,i);
q_sort(arr,str,left,last-1);
q_sort(arr,str,last+1,right);
}

```

```

void swap(struct newstruct *str, int i, int j)
{

```

```

int temp;
char tmpstr[25];
char tmpch;

```

```

temp = indx[i];
indx[i] = indx[j];
indx[j] = temp;

```

```

temp = nol_i_id[i];
nol_i_id[i] = nol_i_id[j];
nol_i_id[j] = temp;

```

```

temp = nol_supply_w_id[i];
nol_supply_w_id[i] = nol_supply_w_id[j];
nol_supply_w_id[j] = temp;

```

```

temp = nol_quantity[i];
nol_quantity[i] = nol_quantity[j];
nol_quantity[j] = temp;

```

```

temp = str->newout.i_price[i];
str->newout.i_price[i] = str->newout.i_price[j];
str->newout.i_price[j] = temp;

```

```

temp = str->newout.ol_amount[i];
str->newout.ol_amount[i] = str->newout.ol_amount[j];
str->newout.ol_amount[j] = temp;

```

```

temp = str->newout.s_quantity[i];
str->newout.s_quantity[i] = str->newout.s_quantity[j];
str->newout.s_quantity[j] = temp;
strncpy(tmpstr,str->newout.i_name[i], 25);
strncpy(str->newout.i_name[i],str->newout.i_name[j], 25);
strncpy(str->newout.i_name[j],tmpstr, 25);

```

```

tmpch = str->newout.brand_generic[i];
str->newout.brand_generic[i] = str->newout.brand_generic[j];
str->newout.brand_generic[j] = tmpch;

```

```

}
#endif

```

client/oracle/plnew.c

```

#ifndef RCSID
static char *RCSid =
"$Header: plnew.c,v 1.4 2003/08/05 14:10:58 root Exp $ Copyr (c) 1994 Oracle";
#endif /* RCSID */

```

```

/*=====
====
| Copyright (c) 1996 , 1997, 1998 Oracle Corp. Redwood Shores, CA |
| OPEN SYSTEMS PERFORMANCE GROUP |

```

```

=====
==+
| FILENAME
| plnew.c
| DESCRIPTION
| OCI version (using PL/SQL stored procedure) of
| NEW ORDER transaction in TPC-C benchmark.
=====
==*/

```

```

#ifndef ORA_TPCC
#define ORA_TPCC
#include "ora_tpcc.h"
#endif

```

```

#include <userlog.h>

```

```

#define SQLTXT2 "BEGIN initpcc.init_no(idx1arr); END;"

```

```

#define NITEMS 15
#define ROWIDLEN 20
#define OCIROWLEN 20

```

```

struct newctx {
    ub2 nol_i_id_len[NITEMS];
    ub2 nol_supply_w_id_len[NITEMS];
    ub2 nol_quantity_len[NITEMS];
    ub2 nol_amount_len[NITEMS];
    ub2 s_quantity_len[NITEMS];
    ub2 i_name_len[NITEMS];
    ub2 i_price_len[NITEMS];
    ub2 s_dist_info_len[NITEMS];
    ub2 ol_o_id_len[NITEMS];
    ub2 ol_number_len[NITEMS];
    ub2 s_remote_len[NITEMS];
    ub2 s_quant_len[NITEMS];
    ub2 ol_dist_info_len[NITEMS];
    ub2 s_bg_len[NITEMS];

```

```

    int ol_o_id[NITEMS];
    int ol_number[NITEMS];

```

```

    int s_remote[NITEMS];
    char s_dist_info[NITEMS][25];
    OCISmt *curn1;
    OCIBind *ol_i_id_bp;
    OCIBind *ol_supply_w_id_bp;
    OCIBind *i_price_bp;
    OCIBind *i_name_bp;
    OCIBind *s_bg_bp;
    ub4 nol_i_count;
    ub4 nol_s_count;
    ub4 nol_q_count;
    ub4 nol_item_count;
    ub4 nol_name_count;
    ub4 nol_qty_count;
    ub4 nol_bg_count;
    ub4 nol_am_count;
    ub4 s_remote_count;
    OCISmt *curn2;
    OCIBind *ol_quantity_bp;
    OCIBind *s_remote_bp;
    OCIBind *s_quantity_bp;
    OCIBind *w_id_bp;
    OCIBind *d_id_bp;
    OCIBind *c_id_bp;
    OCIBind *o_all_local_bp;
    OCIBind *o_all_cnt_bp;
    OCIBind *w_tax_bp;
    OCIBind *d_tax_bp;
    OCIBind *o_id_bp;
    OCIBind *c_discount_bp;
    OCIBind *c_credit_bp;
    OCIBind *c_last_bp;
    OCIBind *retries_bp;
    OCIBind *cr_date_bp;
    OCIBind *ol_o_id_bp;
    OCIBind *ol_amount_bp;

```

```

    sb2 w_id_len;
    ub2 d_id_len;
    ub2 c_id_len;
    ub2 o_all_local_len;
    ub2 o_all_cnt_len;
    ub2 w_tax_len;
    ub2 d_tax_len;
    ub2 o_id_len;
    ub2 c_discount_len;

```

```

    ub2 c_credit_len;
    ub2 c_last_len;
    ub2 retries_len;
    ub2 cr_date_len;
};

```

```

typedef struct newctx newctx;

```

```

static newctx *nctx;

```

```

tkvcninit ()
{

```

```

    int i;
    text stmbuf[32*1024];

```

```

    nctx = (newctx *) malloc (sizeof(newctx));
    DISCARD memset(nctx,(char)0,sizeof(newctx));
    nctx->w_id_len = sizeof(w_id);
    nctx->d_id_len = sizeof(d_id);
    nctx->c_id_len = sizeof(c_id);
    nctx->o_all_local_len = sizeof(o_all_local);
    nctx->o_all_cnt_len = sizeof(o_all_cnt);
    nctx->w_tax_len = 0;
    nctx->d_tax_len = 0;
    nctx->o_id_len = sizeof(o_id);
    nctx->c_discount_len = 0;
    nctx->c_credit_len = 0;
    nctx->c_last_len = 0;
    nctx->retries_len = sizeof(retries);
    nctx->cr_date_len = sizeof(cr_date);

```

```

    /* open first cursor */
    DISCARD OCIERROR(errhp,OCIHandleAlloc(tpcenv,(dvoid **)&nctx->curn1,
        OCI_HTYPE_STMT, 0, (dvoid**)0));
    #if defined(ISO)
    sqlfile("../blocks/tkvcnpnew_iso.sql",stmbuf);
    #else
    #if defined(ISO7)
    sqlfile("../blocks/tkvcnpnew_iso7.sql",stmbuf);
    #else
    sqlfile("../blocks/tkvcnpnew.sql",stmbuf);
    #endif
    #endif

```

```

    DISCARD OCIERROR(errhp,OCISmtPrepare(nctx->curn1, errhp, stmbuf,
        strlen((char *)stmbuf), OCI_NTV_SYNTAX, OCI_DEFAULT));

```

```

    /* bind variables */

```

```

    OCIBNDPL(nctx->curn1, nctx->w_id_bp, errhp, ":w_id",ADR(w_id),SIZ(w_id),
        SQLT_INT, &nctx->w_id_len);
    OCIBNDPL(nctx->curn1, nctx->d_id_bp, errhp, ":d_id",ADR(d_id),SIZ(d_id),
        SQLT_INT, &nctx->d_id_len);
    OCIBNDPL(nctx->curn1, nctx->c_id_bp, errhp, ":c_id",ADR(c_id),SIZ(c_id),
        SQLT_INT, &nctx->c_id_len);
    OCIBNDPL(nctx->curn1, nctx->o_all_local_bp, errhp, ":o_all_local",
        ADR(o_all_local), SIZ(o_all_local),SQLT_INT, &nctx->o_all_local_len);
    OCIBNDPL(nctx->curn1, nctx->o_all_cnt_bp, errhp, ":o_all_cnt",ADR(o_all_cnt),
        SIZ(o_all_cnt),SQLT_INT, &nctx->o_all_cnt_len);
    OCIBNDPL(nctx->curn1, nctx->w_tax_bp, errhp, ":w_tax",ADR(w_tax),SIZ(w_tax),
        SQLT_FLT, &nctx->w_tax_len);
    OCIBNDPL(nctx->curn1, nctx->d_tax_bp, errhp, ":d_tax",ADR(d_tax),SIZ(d_tax),
        SQLT_FLT, &nctx->d_tax_len);
    OCIBNDPL(nctx->curn1, nctx->o_id_bp, errhp, ":o_id",ADR(o_id),SIZ(o_id),
        SQLT_INT, &nctx->o_id_len);
    OCIBNDPL(nctx->curn1, nctx->c_discount_bp, errhp, ":c_discount",
        ADR(c_discount), SIZ(c_discount),SQLT_FLT, &nctx->c_discount_len);
    OCIBNDPL(nctx->curn1, nctx->c_credit_bp, errhp, ":c_credit",c_credit,
        SIZ(c_credit),SQLT_CHR, &nctx->c_credit_len);
    OCIBNDPL(nctx->curn1, nctx->c_last_bp, errhp, ":c_last",c_last,SIZ(c_last),
        SQLT_STR, &nctx->c_last_len);
    OCIBNDPL(nctx->curn1, nctx->retries_bp, errhp, ":retry",ADR(retries),
        SIZ(retries),SQLT_INT, &nctx->retries_len);
    OCIBNDPL(nctx->curn1, nctx->cr_date_bp, errhp, ":cr_date",&cr_date,
        SIZ(OCIDate), SQLT_ODT, &nctx->cr_date_len);

```

```

    OCIBNDPL(nctx->curn1, nctx->ol_i_id_bp, errhp, ":ol_i_id",nol_i_id,
        SIZ(int), SQLT_INT, nctx->nol_i_id_len,NITEMS,&nctx->nol_i_count);
    OCIBNDPL(nctx->curn1, nctx->ol_supply_w_id_bp, errhp, ":ol_supply_w_id",
        nol_supply_w_id,SIZ(int),SQLT_INT, nctx->nol_supply_w_id_len,
        NITEMS, &nctx->nol_s_count);
    OCIBNDPL(nctx->curn1, nctx->ol_quantity_bp, errhp, ":ol_quantity",
        nol_quantity, SIZ(int),SQLT_INT,nctx->nol_quantity_len,
        NITEMS,&nctx->nol_q_count);
    OCIBNDPL(nctx->curn1, nctx->i_price_bp, errhp, ":i_price",i_price,SIZ(int),
        SQLT_INT, nctx->i_price_len,NITEMS, &nctx->nol_item_count);
    OCIBNDPL(nctx->curn1, nctx->i_name_bp, errhp, ":i_name",i_name,
        SIZ(i_name[0]),SQLT_STR, nctx->i_name_len,NITEMS,
        &nctx->nol_name_count);
    OCIBNDPL(nctx->curn1, nctx->s_quantity_bp, errhp, ":s_quantity",s_quantity,
        SIZ(int), SQLT_INT,nctx->s_quant_len,NITEMS,&nctx->nol_qty_count);
    OCIBNDPL(nctx->curn1, nctx->s_bg_bp, errhp, ":brand_generic",brand_generic,
        SIZ(char), SQLT_CHR,nctx->s_bg_len,NITEMS,&nctx->nol_bg_count);
    OCIBNDPL(nctx->curn1, nctx->ol_amount_bp, errhp, ":ol_amount",nol_amount,

```

```

        SIZ(int),SQLT_INT, nctx->no_l_amount_len,NITEMS,&nctx->no_l_am_count);
OCIBNDPLA(nctx->cur1, nctx->s_remote_bp,errhp,"s_remote",nctx->s_remote,
        SIZ(int),SQLT_INT, nctx->s_remote_len,NITEMS,&nctx->s_remote_count);

/* open second cursor */
DISCARD OCIERROR(errhp,OCIHandleAlloc(tpcenv, (dvoid **)&nctx->cur2),
        OCI_HTYPE_STMT, 0, (dvoid**)0);
DISCARD sprintf((char *) stmbuf, SQLTXT2);
DISCARD OCIERROR(errhp,OCIStmtPrepare(nctx->cur2, errhp, stmbuf,
        strlen(char *)stmbuf), OCI_NTV_SYNTAX, OCI_DEFAULT);

/* execute second cursor to init newinit package */
{
    int idx1arr[NITEMS];
    OCIBind *idx1arr_bp;
    ub2 idx1arr_len[NITEMS];
    ub2 idx1arr_rcode[NITEMS];
    sb2 idx1arr_ind[NITEMS];
    ub4 idx1arr_count;
    ub2 idx;

    for (idx = 0; idx < NITEMS; idx++) {
        idx1arr[idx] = idx + 1;
        idx1arr_ind[idx] = TRUE;
        idx1arr_len[idx] = sizeof(int);
    }
    idx1arr_count = NITEMS;
    o_ol_cnt = NITEMS;

/* Bind array */
OCIBNDPLA(nctx->cur2, idx1arr_bp, errhp, "idx1arr", idx1arr,
        SIZ(int), SQLT_INT, idx1arr_len, NITEMS, &idx1arr_count);

    execstatus = OCIStmtExecute(tpcsvc, nctx->cur2, errhp, 1, 0,
        NULLP(CONST OCISnapshot), NULLP(CONST OCISnapshot), OCI_DEFAULT);
    if(execstatus != OCI_SUCCESS) {
        OCITransRollback(tpcsvc, errhp, OCI_DEFAULT);
        errcode = OCIERROR(errhp, execstatus);
        return -1;
    }
}

return (0);
}

tkvcn ()
{
    int i;
    int rcount;

retry:

    status = 0;          /* number of invalid items */

/* get number of order lines, and check if all are local */

    o_ol_cnt = NITEMS;
    o_all_local = 1;
    for (i = 0; i < NITEMS; i++) {
        if (no_l_id[i] == 0) {
            o_ol_cnt = i;
            break;
        }
        if (no_l_supply_w_id[i] != w_id) {
            nctx->s_remote[i] = 1;
            o_all_local = 0;
        }
        else
            nctx->s_remote[i] = 0;
    }

    nctx->w_id_len = sizeof(w_id);
    nctx->d_id_len = sizeof(d_id);
    nctx->c_id_len = sizeof(c_id);
    nctx->o_all_local_len = sizeof(o_all_local);
    nctx->o_ol_cnt_len = sizeof(o_ol_cnt);
    nctx->w_tax_len = 0;
    nctx->d_tax_len = 0;
    nctx->o_id_len = sizeof(o_id);
    nctx->c_discount_len = 0;
    nctx->c_credit_len = 0;
    nctx->c_last_len = 0;
    nctx->retries_len = sizeof(retries);
    nctx->cr_date_len = sizeof(cr_date);
    /* this is the row count */
    rcount = o_ol_cnt;
    nctx->no_l_i_count = o_ol_cnt;
    nctx->no_l_q_count = o_ol_cnt;
    nctx->no_l_s_count = o_ol_cnt;

```

```

    nctx->s_remote_count = o_ol_cnt;

    nctx->no_l_qty_count = 0;
    nctx->no_l_bg_count = 0;
    nctx->no_l_item_count = 0;
    nctx->no_l_name_count = 0;
    nctx->no_l_am_count = 0;

/* initialization for array operations */
for (i = 0; i < o_ol_cnt; i++) {
    nctx->o_l_number[i] = i + 1;
    nctx->no_l_i_id_len[i] = sizeof(int);
    nctx->no_l_supply_w_id_len[i] = sizeof(int);
    nctx->no_l_quantity_len[i] = sizeof(int);
    nctx->no_l_amount_len[i] = sizeof(int);
    nctx->o_l_o_id_len[i] = sizeof(int);
    nctx->o_l_number_len[i] = sizeof(int);
    nctx->o_l_dist_info_len[i] = nctx->s_dist_info_len[i];
    nctx->s_remote_len[i] = sizeof(int);
    nctx->s_quant_len[i] = sizeof(int);
    nctx->i_name_len[i] = 0;
    nctx->s_bg_len[i] = 0;
}
for (i = o_ol_cnt; i < NITEMS; i++) {

    nctx->no_l_i_id_len[i] = 0;
    nctx->no_l_supply_w_id_len[i] = 0;
    nctx->no_l_quantity_len[i] = 0;
    nctx->no_l_amount_len[i] = 0;
    nctx->o_l_o_id_len[i] = 0;
    nctx->o_l_number_len[i] = 0;
    nctx->o_l_dist_info_len[i] = 0;
    nctx->s_remote_len[i] = 0;
    nctx->s_quant_len[i] = 0;
    nctx->i_name_len[i] = 0;
    nctx->s_bg_len[i] = 0;
}

    execstatus = OCIStmtExecute(tpcsvc, nctx->cur1, errhp, 1, 0, 0,
        OCI_DEFAULT |
OCI_COMMIT_ON_SUCCESS);

    if(execstatus != OCI_SUCCESS) {
        OCITransRollback(tpcsvc, errhp, OCI_DEFAULT);
        errcode = OCIERROR(errhp, execstatus);
        if(errcode == NOT_SERIALIZABLE) {
            retries++;
            goto retry;
        } else if (errcode == RECOVER) {
            retries++;
            goto retry;
        } else if (errcode == SNAPSHOT_TOO_OLD) {
            retries++;
            goto retry;
        } else {
            return -1;
        }
    }

/* did the txn succeed ? */
if (rcount != o_ol_cnt)
{
    status = rcount - o_ol_cnt;
    o_ol_cnt = rcount;
}

#ifdef DEBUG
    printf("w_id = %d, d_id = %d, c_id = %d\n", w_id, d_id, c_id);
#endif

    total_amount = 0;
    for (i = 0; i < o_ol_cnt; i++) total_amount += no_l_amount[i];
    total_amount *= ((float)(1.0 - c_discount)) * (float)(1.0 + ((float)(d_tax)) + ((float)
(w_tax)));
    total_amount = total_amount/100;

    return (0);
}

void tkvcndone ()
{
    int i;

    if (nctx)
    {
        DISCARD OCIHandleFree((dvoid *)nctx->cur1, OCI_HTYPE_STMT);
        DISCARD OCIHandleFree((dvoid *)nctx->cur2, OCI_HTYPE_STMT);
        free (nctx);
    }
}

```

```
}
```

client/oracle/plpay.c

```
#ifndef RCSID
static char *RCSid =
"$Header: plpay.c,v 1.3 2003/07/01 15:44:03 mliu Exp $ Copyr (c) 1994 Oracle";
#endif /* RCSID */

/*=====
++
| Copyright (c) 1995 Oracle Corp, Redwood Shores, CA |
| OPEN SYSTEMS PERFORMANCE GROUP |
| All Rights Reserved |
+=====
++
| FILENAME
| plpay.c
| DESCRIPTION
| OCI version (using PL/SQL stored procedure) of
| PAYMENT transaction in TPC-C benchmark.
+=====
==*/

#include "ora_tpcc.h"

#ifdef TUX
#include <userlog.h>
#endif

#define SQLTXT_INIT "BEGIN inittpc.init_pay; END;"

struct payctx {
OCIStmt *curpi;
OCIStmt *curp0;
OCIStmt *curp1;
OCIBind *w_id_bp[2];
ub2 w_id_len;

OCIBind *d_id_bp[2];
ub2 d_id_len;

OCIBind *c_w_id_bp[2];
ub2 c_w_id_len;

OCIBind *c_d_id_bp[2];
ub2 c_d_id_len;

OCIBind *c_id_bp[2];
ub2 c_id_len;

OCIBind *h_amount_bp[2];
ub2 h_amount_len;

OCIBind *c_last_bp[2];
ub2 c_last_len;

OCIBind *w_street_1_bp[2];
ub2 w_street_1_len;

OCIBind *w_street_2_bp[2];
ub2 w_street_2_len;

OCIBind *w_city_bp[2];
ub2 w_city_len;

OCIBind *w_state_bp[2];
ub2 w_state_len;

OCIBind *w_zip_bp[2];
ub2 w_zip_len;

OCIBind *d_street_1_bp[2];
ub2 d_street_1_len;

OCIBind *d_street_2_bp[2];
ub2 d_street_2_len;

OCIBind *d_city_bp[2];
ub2 d_city_len;
};
```

```
ub2 d_city_len;

OCIBind *d_state_bp[2];
ub2 d_state_len;

OCIBind *d_zip_bp[2];
ub2 d_zip_len;

OCIBind *c_first_bp[2];
ub2 c_first_len;

OCIBind *c_middle_bp[2];
ub2 c_middle_len;

OCIBind *c_street_1_bp[2];
ub2 c_street_1_len;

OCIBind *c_street_2_bp[2];
ub2 c_street_2_len;

OCIBind *c_city_bp[2];
ub2 c_city_len;

OCIBind *c_state_bp[2];
ub2 c_state_len;

OCIBind *c_zip_bp[2];
ub2 c_zip_len;

OCIBind *c_phone_bp[2];
ub2 c_phone_len;

OCIBind *c_since_bp[2];
ub2 c_since_len;

OCIBind *c_credit_bp[2];
ub2 c_credit_len;

OCIBind *c_credit_lim_bp[2];
ub2 c_credit_lim_len;

OCIBind *c_discount_bp[2];
ub2 c_discount_len;

OCIBind *c_balance_bp[2];
ub2 c_balance_len;

OCIBind *c_data_bp[2];
ub2 c_data_len;

OCIBind *h_date_bp[2];
ub2 h_date_len;

OCIBind *retries_bp[2];
ub2 retries_len;

OCIBind *cr_date_bp[2];
ub2 cr_date_len;

OCIBind *byln_bp[2];
ub2 byln_len;
};

typedef struct payctx payctx;

payctx *pctx;

int tkvcpin (void)
{
text stmbuf[SQL_BUF_SIZE];
pctx = (payctx *)malloc(sizeof(payctx));
memset(pctx,(char)0,sizeof(payctx));

/* cursor for init */
DISCARD OCIERROR(errhp,OCIHandleAlloc(tpcenv, (dvoid **)&(pctx->curpi),
OCI_HTYPE_STMT,0,(dvoid**)0));

DISCARD OCIERROR(errhp,OCIHandleAlloc(tpcenv, (dvoid **)&(pctx->curp0),
OCI_HTYPE_STMT,0,(dvoid**)0));
DISCARD OCIERROR(errhp,OCIHandleAlloc(tpcenv, (dvoid **)&(pctx->curp1),
OCI_HTYPE_STMT,0,(dvoid**)0));

/* build the init statement and execute it */
sprintf ((char*)stmbuf, SQLTXT_INIT);
DISCARD OCIERROR(errhp,OCIStmtPrepare(pctx->curpi, errhp, stmbuf,
strlen((char *)stmbuf), OCI_NTV_SYNTAX, OCI_DEFAULT));
DISCARD OCIERROR(errhp,OCIStmtExecute(tpcenv,pctx->curpi,errhp,1,0,
NULLP(CONST OCISnapshot),NULLP(OCISnapshot),OCI_DEFAULT));
};
```



```

OCIBNDR(pctx->curp1, pctx->h_date_bp1, errhp, "h_date", h_date, SIZ(h_date),
        SQLT_STR, &pctx->h_date_ind, &pctx->h_date_len, &pctx->h_date_rc);
*/
OCIBNDPL(pctx->curp1, pctx->retries_bp[1], errhp, "retries", ADR(retries),
        SIZ(int), SIZ(int), &pctx->retries_len);
OCIBNDPL(pctx->curp1, pctx->cr_date_bp[1], errhp, "cr_date", ADR(cr_date),
        SIZ(OCIDate), SIZ(OCIDate), &pctx->cr_date_len);

return (0);
}

tkvcvp ()
{
retry:

pctx->w_id_len = SIZ(w_id);
pctx->d_id_len = SIZ(d_id);
pctx->c_w_id_len = 0;
pctx->c_d_id_len = 0;
pctx->c_id_len = 0;
pctx->h_amount_len = SIZ(h_amount);
pctx->c_last_len = SIZ(c_last);
pctx->w_street_1_len = 0;
pctx->w_street_2_len = 0;
pctx->w_city_len = 0;
pctx->w_state_len = 0;
pctx->w_zip_len = 0;
pctx->d_street_1_len = 0;
pctx->d_street_2_len = 0;
pctx->d_city_len = 0;
pctx->d_state_len = 0;
pctx->d_zip_len = 0;
pctx->c_first_len = 0;
pctx->c_middle_len = 0;
pctx->c_street_1_len = 0;
pctx->c_street_2_len = 0;
pctx->c_city_len = 0;
pctx->c_state_len = 0;
pctx->c_zip_len = 0;
pctx->c_phone_len = 0;
pctx->c_since_len = 0;
pctx->c_credit_len = 0;
pctx->c_credit_lim_len = 0;
pctx->c_discount_len = 0;
pctx->c_balance_len = sizeof(double);
pctx->c_data_len = 0;
pctx->h_date_len = 0;
pctx->retries_len = SIZ(retries);
pctx->cr_date_len = 7;

if(bylastname) {
execstatus=OCIStmtExecute(tpscvc, pctx->curp1, errhp, 1, 0,
        NULLP(CONST OCI_Snapshot), NULLP(OCI_Snapshot),
        OCI_DEFAULT|OCI_COMMIT_ON_SUCCESS);
} else {
execstatus=OCIStmtExecute(tpscvc, pctx->curp0, errhp, 1, 0,
        NULLP(CONST OCI_Snapshot), NULLP(OCI_Snapshot),
        OCI_DEFAULT|OCI_COMMIT_ON_SUCCESS);
}

if(execstatus != OCI_SUCCESS) {
OCITransRollback(tpscvc, errhp, OCI_DEFAULT);
errcode = OCIERROR(errhp, execstatus);
if(errcode == NOT_SERIALIZABLE) {
retries++;
goto retry;
} else if (errcode == RECOVER) {
retries++;
goto retry;
} else if (errcode == SNAPSHOT_TOO_OLD) {
retries++;
goto retry;
} else {
return -1;
}
}
return 0;
}

void tkvcvpdone ()
{
if(pctx) {
free(pctx);
}
}

```

client/oracle/plord.c

```

/* Copyright (c) 2002, Oracle Corporation. All rights reserved. */

*/

NAME
tkvcordq.c - OCI version using queues of ORDER STATUS
transaction in TPC-C benchmark.

DESCRIPTION
<short description of facility this file declares/defines>

EXPORT FUNCTION(S)

INTERNAL FUNCTION(S)
<other external functions defined - one-line descriptions>

STATIC FUNCTION(S)
<static functions defined - one-line descriptions>

NOTES
<other useful comments, qualifications, etc.>

MODIFIED (MM/DD/YY)
xnie 06/25/02 - queue open cluster join.
heri 05/07/02 - Fix error in cursor.
heri 02/01/02 - Cleanup, remove indicator values and return codes.
lwang 07/25/01 - Merged lwang_tpcitrc
lwang 07/23/01 - fix include
lwang 07/23/01 - Creation

*/

#include "ora_tpc.h"

/*-----
PRIVATE TYPES AND CONSTANTS
-----*/

/*-----
STATIC FUNCTION DECLARATIONS
-----*/

#define SQLCUR0 "SELECT rowid FROM cust \
WHERE c_d_id = :d_id AND c_w_id = :w_id AND c_last = :c_last \
ORDER BY c_last, c_d_id, c_w_id, c_first"

#define SQLCUR1 "SELECT /*+ USE_NL(cust) INDEX_DESC(ordr iordr2) */ \
c_id, c_balance, c_first, c_middle, c_last, \
o_id, o_entry_d, o_carrier_id, o_ol_cnt, ordr.rowid \
FROM cust, ordr \
WHERE cust.rowid = :cust_rowid \
AND o_d_id = c_d_id AND o_w_id = c_w_id AND o_c_id = c_id \
ORDER BY o_c_id DESC, o_d_id DESC, o_w_id DESC, o_id DESC"

#define SQLCUR2 "SELECT /*+ USE_NL(cust) INDEX_DESC(ordr iordr2) */ \
c_balance, c_first, c_middle, c_last, \
o_id, o_entry_d, o_carrier_id, o_ol_cnt, ordr.rowid \
FROM cust, ordr \
WHERE c_id = :c_id AND c_d_id = :d_id AND c_w_id = :w_id \
AND o_d_id = c_d_id AND o_w_id = c_w_id AND o_c_id = c_id \
ORDER BY o_c_id DESC, o_d_id DESC, o_w_id DESC, o_id DESC"

#define SQLCUR3 "SELECT /*+ ORDERED USE_NL(ordl) CLUSTER(ordl) */ \
ol_i_id, ol_supply_w_id, ol_quantity, ol_amount, ol_delivery_d \
FROM ordr, ordl \
WHERE ordr.rowid = :ordr_rowid \
AND o_id = ol_o_id AND ol_d_id = o_d_id AND ol_w_id = o_w_id"

#define SQLCUR4 "SELECT count(c_last) FROM cust \
WHERE c_d_id = :d_id AND c_w_id = :w_id AND c_last = :c_last"

struct ordctx {
ub2 c_rowid_len[100];
ub2 ol_supply_w_id_len[NITEMS];
ub2 ol_i_id_len[NITEMS];
ub2 ol_quantity_len[NITEMS];
ub2 ol_amount_len[NITEMS];
ub2 ol_delivery_d_len[NITEMS];
}

```

```

ub2 ol_w_id_len;
ub2 ol_d_id_len;
ub2 ol_o_id_len;

ub4 ol_supply_w_id_csize;
ub4 ol_i_id_csize;
ub4 ol_quantity_csize;
ub4 ol_amount_csize;
ub4 ol_delivery_d_csize;
ub4 ol_w_id_csize;
ub4 ol_d_id_csize;
ub4 ol_o_id_csize;

OCIStmt *curo0;
OCIStmt *curo1;
OCIStmt *curo2;
OCIStmt *curo3;
OCIStmt *curo4;
OCIBind *c_id_bp;
OCIBind *w_id_bp[4];
OCIBind *d_id_bp[4];
OCIBind *c_last_bp[2];
OCIBind *o_id_bp;
OCIBind *c_rowid_bp;
OCIBind *o_rowid_bp;
OCIDefine *c_rowid_dp;
OCIDefine *c_last_dp[2];
OCIDefine *c_id_dp;
OCIDefine *c_first_dp[2];
OCIDefine *c_middle_dp[2];
OCIDefine *c_balance_dp[2];
OCIDefine *o_rowid_dp[2];
OCIDefine *o_id_dp[2];
OCIDefine *o_entry_d_dp[2];
OCIDefine *o_cr_id_dp[2];
OCIDefine *o_ol_cnt_dp[2];
OCIDefine *ol_d_d_dp;
OCIDefine *ol_i_id_dp;
OCIDefine *ol_supply_w_id_dp;
OCIDefine *ol_quantity_dp;
OCIDefine *ol_amount_dp;
OCIDefine *ol_d_base_dp;
OCIDefine *c_count_dp;
OCIRowid *c_rowid_ptr[100];
OCIRowid *c_rowid_cust;
OCIRowid *o_rowid;
int cs;
int cust_idx;
int norow;
int rcount;
int somerows;
};

typedef struct ordctx ordctx;

struct defctx
{
boolean reexec;
ub4 count;
};
typedef struct defctx defctx;

static ordctx *octx;

static defctx cbctx;

tkvcoint ()
{
int i;
text stmbuf[SQL_BUF_SIZE];

octx = (ordctx *) malloc (sizeof(ordctx));
DISCARD memset(octx,(char)0,sizeof(ordctx));
octx->cs = 1;
octx->norow = 0;
octx->somerows = 10;
/* get the rowid handles */
OCIERROR(errhp, OCIDescriptorAlloc((dvoid *)tpcenv,(dvoid **)&octx->o_rowid,
(ub4)OCI_DTYPE_ROWID, (size_t) 0, (dvoid **)0));
for(i=0;i<100;i++) {
DISCARD OCIERROR( errhp, OCIDescriptorAlloc(tpcenv,
(dvoid **)&octx->c_rowid_ptr[i], OCI_DTYPE_ROWID,0,(dvoid **)0);
}

DISCARD OCIERROR( errhp,
OCIHandleAlloc(tpcenv,(dvoid **)&octx->curo0,OCI_HTYPE_STMT,0,(dvoid **)0));
DISCARD OCIERROR( errhp,
OCIHandleAlloc(tpcenv,(dvoid **)&octx->curo1,OCI_HTYPE_STMT,0,(dvoid **)0));
DISCARD OCIERROR( errhp,
OCIHandleAlloc(tpcenv,(dvoid **)&octx->curo2,OCI_HTYPE_STMT,0,(dvoid **)0));
DISCARD OCIERROR( errhp,
OCIHandleAlloc(tpcenv,(dvoid **)&octx->curo3,OCI_HTYPE_STMT,0,(dvoid **)0));
DISCARD OCIERROR( errhp,
OCIHandleAlloc(tpcenv,(dvoid **)&octx->curo4,OCI_HTYPE_STMT,0,(dvoid **)0));

/* c_id = 0, use find customer by lastname. Get an array or rowid's back*/
DISCARD sprintf((char *) stmbuf, SQLCUR0);
DISCARD OCIERROR( errhp,
OCIStmtPrepare(octx->curo0,errhp,stmbuf,(ub4)strlen((char *)stmbuf),
OCI_NT_V_SYNTAX,OCI_DEFAULT));
DISCARD OCIERROR( errhp,
OCIAttrSet(octx->curo0,OCI_HTYPE_STMT,&octx->norow,0,
OCI_ATTR_PREFETCH_ROWS,errhp));
/* get order/customer info back based on rowid */
DISCARD sprintf((char *) stmbuf, SQLCUR1);
DISCARD OCIERROR( errhp,
OCIStmtPrepare(octx->curo1,errhp,stmbuf,(ub4)strlen((char *)stmbuf),
OCI_NT_V_SYNTAX,OCI_DEFAULT));
DISCARD OCIERROR( errhp,
OCIAttrSet(octx->curo1,OCI_HTYPE_STMT,&octx->norow,0,
OCI_ATTR_PREFETCH_ROWS,errhp));

/* c_id == 0, use lastname to find customer */
DISCARD sprintf((char *) stmbuf, SQLCUR2);
DISCARD OCIERROR( errhp,
OCIStmtPrepare(octx->curo2,errhp,stmbuf,(ub4)strlen((char *)stmbuf),
OCI_NT_V_SYNTAX,OCI_DEFAULT));
DISCARD OCIERROR( errhp,
OCIAttrSet(octx->curo2,OCI_HTYPE_STMT,&octx->norow,0,
OCI_ATTR_PREFETCH_ROWS,errhp));

DISCARD sprintf((char *) stmbuf, SQLCUR3);
DISCARD OCIERROR( errhp,
OCIStmtPrepare(octx->curo3,errhp,stmbuf,(ub4)strlen((char *)stmbuf),
OCI_NT_V_SYNTAX,OCI_DEFAULT));
DISCARD OCIERROR( errhp,
OCIAttrSet(octx->curo3,OCI_HTYPE_STMT,&octx->norow,0,
OCI_ATTR_PREFETCH_ROWS,errhp));

DISCARD sprintf((char *) stmbuf, SQLCUR4);
DISCARD OCIERROR( errhp,
OCIStmtPrepare(octx->curo4,errhp,stmbuf,(ub4)strlen((char *)stmbuf),
OCI_NT_V_SYNTAX,OCI_DEFAULT));
DISCARD OCIERROR( errhp,
OCIAttrSet(octx->curo4,OCI_HTYPE_STMT,&octx->norow,0,
OCI_ATTR_PREFETCH_ROWS,errhp));

for (i = 0; i < NITEMS; i++) {

octx->ol_supply_w_id_len[i] = sizeof(int);
octx->ol_i_id_len[i] = sizeof(int);
octx->ol_quantity_len[i] = sizeof(int);
octx->ol_amount_len[i] = sizeof(int);
octx->ol_delivery_d_len[i] = sizeof(ol_d_base[0]);
}

octx->ol_supply_w_id_csize = NITEMS;
octx->ol_i_id_csize = NITEMS;
octx->ol_quantity_csize = NITEMS;
octx->ol_amount_csize = NITEMS;
octx->ol_delivery_d_csize = NITEMS;
octx->ol_w_id_csize = NITEMS;
octx->ol_o_id_csize = NITEMS;
octx->ol_d_id_csize = NITEMS;
octx->ol_w_id_len = sizeof(int);
octx->ol_d_id_len = sizeof(int);
octx->ol_o_id_len = sizeof(int);

/* bind variables */

/* c_id (customer id) is not known */
OCIBND(octx->curo0,octx->w_id_bp[0],errhp,":w_id",ADR(w_id),
SIZ(int),SQLT_INT);
OCIBND(octx->curo0,octx->d_id_bp[0],errhp,":d_id",ADR(d_id),
SIZ(int),SQLT_INT);
OCIBND(octx->curo0,octx->c_last_bp[0],errhp,":c_last",c_last,
SIZ(c_last),SQLT_STR);
OCIDFNRA(octx->curo0,octx->c_rowid_dp,errhp,1,octx->c_rowid_ptr,
SIZ(OCIRowid*),SQLT_RDD,NULL,octx->c_rowid_len,NULL);

OCIBND(octx->curo1,octx->c_rowid_bp,errhp,":cust_rowid", &octx->c_rowid_cust,
sizeof(octx->c_rowid_ptr[0]),SQLT_RDD);
OCIDEF(octx->curo1,octx->c_id_dp,errhp,1,ADR(c_id),SIZ(int),SQLT_INT);
OCIDEF(octx->curo1,octx->c_balance_dp[0],errhp,2,ADR(c_balance),
SIZ(double),SQLT_FLT);
OCIDEF(octx->curo1,octx->c_first_dp[0],errhp,3,c_first,SIZ(c_first)-1,
SQLT_CHR);
OCIDEF(octx->curo1,octx->c_middle_dp[0],errhp,4,c_middle,
SIZ(c_middle)-1,SQLT_AFC);
OCIDEF(octx->curo1,octx->c_last_dp[0],errhp,5,c_last,SIZ(c_last)-1,
SQLT_CHR);
OCIDEF(octx->curo1,octx->o_id_dp[0],errhp,6,ADR(o_id),SIZ(int),SQLT_INT);
OCIDEF(octx->curo1,octx->o_entry_d_dp[0],errhp,7,
&o_entry_d_base,SIZ(OCIDate),SQLT_ODT);

```



```

OCIDEF(octx->curo1,octx->o_cr_id_dp[0],errhp,8,ADR(o_carrier_id),
    SIZ(int),SQLT_INT);
OCIDEF(octx->curo1,octx->o_ol_cnt_dp[0],errhp,9,ADR(o_ol_cnt),
    SIZ(int),SQLT_INT);
OCIDEF(octx->curo1,octx->o_rowid_dp[0],errhp,10,ADR(octx->o_rowid),
    SIZ(OCIRowid*),SQLT_RDD);

/* Bind for third cursor , no-zero customer id */
OCIBND(octx->curo2,octx->w_id_bp[1],errhp,"w_id",ADR(w_id),
    SIZ(int),SQLT_INT);
OCIBND(octx->curo2,octx->d_id_bp[1],errhp,"d_id",ADR(d_id),
    SIZ(int),SQLT_INT);
OCIBND(octx->curo2,octx->c_id_bp,errhp,"c_id",ADR(c_id),
    SIZ(int),SQLT_INT);
OCIDEF(octx->curo2,octx->c_balance_dp[1],errhp,1,ADR(c_balance),
    SIZ(double),SQLT_FLT);
OCIDEF(octx->curo2,octx->c_first_dp[1],errhp,2,c_first,SIZ(c_first)-1,
    SQLT_CHR);
OCIDEF(octx->curo2,octx->c_middle_dp[1],errhp,3,c_middle,
    SIZ(c_middle)-1,SQLT_AFC);
OCIDEF(octx->curo2,octx->c_last_dp[1],errhp,4,c_last,SIZ(c_last)-1,
    SQLT_CHR);
OCIDEF(octx->curo2,octx->o_id_dp[1],errhp,5,ADR(o_id),SIZ(int),SQLT_INT);
OCIDEF(octx->curo2,octx->o_entry_d_dp[1],errhp,6, &o_entry_d_base,
    SIZ(OCIDate),SQLT_ODT);
OCIDEF(octx->curo2, octx->o_cr_id_dp[1],errhp,7,ADR(o_carrier_id),
    SIZ(int), SQLT_INT);
OCIDEF(octx->curo2,octx->o_ol_cnt_dp[1],errhp,8,ADR(o_ol_cnt),
    SIZ(int),SQLT_INT);
OCIDEF(octx->curo2,octx->o_rowid_dp[1],errhp,9,ADR(octx->o_rowid),
    SIZ(OCIRowid*),SQLT_RDD);

/* Bind for last cursor */

/*
OCIBND(octx->curo3,octx->w_id_bp[2],errhp,"w_id",ADR(w_id),
SIZ(int),SQLT_INT);
OCIBND(octx->curo3,octx->d_id_bp[2],errhp,"d_id",ADR(d_id), SIZ(int),SQLT_INT);
OCIBND(octx->curo3,octx->o_id_bp,errhp,"o_id",ADR(o_id), SIZ(int),SQLT_INT);
OCIBND(octx->curo3,octx->c_id_bp,errhp,"c_id",ADR(c_id), SIZ(int),SQLT_INT);
*/
OCIBND(octx->curo3,octx->o_rowid_bp,errhp,"ordr_rowid",
    &octx->o_rowid, SIZ(OCIRowid*),SQLT_RDD);

OCIDFNRA(octx->curo3, octx->ol_i_id_dp, errhp, 1, ol_i_id,SIZ(int),SQLT_INT,
    NULL,octx->ol_i_id_len, NULL);
OCIDFNRA(octx->curo3,octx->ol_supply_w_id_dp,errhp,2, ol_supply_w_id,
    SIZ(int),SQLT_INT, NULL,
    octx->ol_supply_w_id_len, NULL);
OCIDFNRA(octx->curo3, octx->ol_quantity_dp,errhp,3, ol_quantity,SIZ(int),
    SQLT_INT, NULL,octx->ol_quantity_len, NULL);
OCIDFNRA(octx->curo3,octx->ol_amount_dp,errhp,4,ol_amount, SIZ(int),
    SQLT_INT,NULL,octx->ol_amount_len, NULL);
OCIDFNRA(octx->curo3,octx->ol_d_base_dp,errhp,5,ol_d_base,SIZ(OCIDate),
    SQLT_ODT, NULL,octx->ol_delivery_d_len,NULL);

OCIBND(octx->curo4,octx->w_id_bp[3],errhp,"w_id",ADR(w_id),
    SIZ(int),SQLT_INT);
OCIBND(octx->curo4,octx->d_id_bp[3],errhp,"d_id",ADR(d_id),
    SIZ(int),SQLT_INT);
OCIBND(octx->curo4,octx->c_last_bp[1],errhp,"c_last",c_last,
    SIZ(c_last), SQLT_STR);
OCIDEF(octx->curo4,octx->c_count_dp,errhp,1,ADR(octx->rcount),SIZ(int),
    SQLT_INT);

return (0);
}

tkvco ()
{
    int i;
    int rcount;

    #if defined(ISO9)
    int secondread = 0;
    char sdate[30];
    ub4 datelen;
    sysdate(sdate);
    printf("Order Status started at: %s\n", sdate);
    #endif

    for (i = 0; i < NITEMS; i++) {
        octx->ol_supply_w_id_len[i] = sizeof(int);
        octx->ol_i_id_len[i] = sizeof(int);
        octx->ol_quantity_len[i] = sizeof(int);
        octx->ol_amount_len[i] = sizeof(int);
        octx->ol_delivery_d_len[i] = sizeof(OCIDate);
    }
}

```

```

octx->ol_supply_w_id_csize = NITEMS;
octx->ol_i_id_csize = NITEMS;
octx->ol_quantity_csize = NITEMS;
octx->ol_amount_csize = NITEMS;
octx->ol_delivery_d_csize = NITEMS;
retry:
if (bylastname)
{
    cbctx.reexec = FALSE;
    execstatus=OCISmtExecute(tpscvc,octx->curo0,errhp,100,0,
        NULLP(CONST OCISnapshot),NULLP(OCISnapshot),OCI_DEFAULT);
    /* will get OCI_NO_DATA if <100 found */
    if ((execstatus != OCI_NO_DATA) && (execstatus != OCI_SUCCESS))
    {
        errcode=OCIERROR(errhp, execstatus);
        if((errcode == NOT_SERIALIZABLE) || (errcode == RECOVER))
        {
            DISCARD OCITransCommit(tpscvc,errhp,OCI_DEFAULT);
            retries++;
            goto retry;
        } else {
            return -1;
        }
    }
}
if (execstatus == OCI_NO_DATA) /* there are no more rows */
{
    /* get rowcount, find middle one */
    DISCARD OCISmtExecute(octx->curo0,OCI_HTYPE_STMT,&rcount,NULL,
        OCI_ATTR_ROW_COUNT,errhp);
    if (rcount < 1)
    {
        userlog("ORDERSTATUS rcount=%d\n",rcount);
        return (-1);
    }
    octx->cust_idx=(rcount)/2 ;
}
else
{
    /* count the number of rows */
    execstatus=OCISmtExecute(tpscvc,octx->curo4,errhp,1,0,
        NULLP(CONST OCISnapshot),NULLP(OCISnapshot),OCI_DEFAULT);
    if ((execstatus != OCI_NO_DATA) && (execstatus != OCI_SUCCESS))
    {
        errcode=OCIERROR(errhp, execstatus);
        if ((errcode == NOT_SERIALIZABLE) || (errcode == RECOVER))
        {
            DISCARD OCITransCommit(tpscvc,errhp,OCI_DEFAULT);
            retries++;
            goto retry;
        } else {
            return -1;
        }
    }
}
if (octx->rcount+1 < 2*10 )
octx->cust_idx=(octx->rcount+1)/2 ;
else
    /* */
{
    cbctx.reexec = TRUE;
    cbctx.count = (octx->rcount+1)/2 ;
    execstatus=OCISmtExecute(tpscvc,octx->curo0,errhp,cbctx.count,
        0,NULLP(CONST OCISnapshot),
        NULLP(OCISnapshot),OCI_DEFAULT);
    /* will get OCI_NO_DATA if <100 found */
    if (cbctx.count > 0)
    {
        userlog ("did not get all rows ");
        return (-1);
    }
}

if ((execstatus != OCI_NO_DATA) && (execstatus != OCI_SUCCESS))
{
    errcode=OCIERROR(errhp, execstatus);
    if((errcode == NOT_SERIALIZABLE) || (errcode == RECOVER))
    {
        DISCARD OCITransCommit(tpscvc,errhp,OCI_DEFAULT);
        retries++;
        goto retry;
    } else {
        return -1;
    }
}
octx->cust_idx=0 ;
}
}

octx->c_rowid_cust = octx->c_rowid_ptr[octx->cust_idx];
execstatus=OCISmtExecute(tpscvc,octx->curo1,errhp,1,0,
    NULLP(CONST OCISnapshot),NULLP(OCISnapshot),OCI_DEFAULT);
if (execstatus != OCI_SUCCESS)
{
    errcode=OCIERROR(errhp,execstatus);
    DISCARD OCITransCommit(tpscvc,errhp,OCI_DEFAULT);
    if((errcode == NOT_SERIALIZABLE) || (errcode == RECOVER))
    || (errcode == SNAPSHOT_TOO_OLD))
    {

```

```

    retries++;
    goto retry;
} else {
    return -1;
}
}
} else
{
    execstatus=OCISmtExecute(tpcsvc,octx->curo2,errhp,1,0,
        NULLP(CONST OCISnapshot),NULLP(OCISnapshot),
        OCI_DEFAULT);
    if (execstatus != OCI_SUCCESS)
    {
        errcode=OCIERROR(errhp,execstatus);
        DISCARD OCITransCommit(tpcsvc,errhp,OCI_DEFAULT);
        if((errcode == NOT_SERIALIZABLE) || (errcode == RECOVERR)
            || (errcode == SNAPSHOT_TOO_OLD))
        {
            retries++;
            goto retry;
        }
        else
        {
            return -1;
        }
    }
}
#endif ISO9
sysdate (sdate);
if (!secondread)
    printf ("----- FIRST READ RESULT (out) %s -----\n", sdate);
else
    printf ("----- SECOND READ RESULT (out) %s -----\n", sdate);

    printf ("c_id = %d\n", c_id);
    printf ("c_last = %s\n", c_last);
    printf ("c_first = %s\n", c_first);
    printf ("c_middle = %s\n", c_middle);
    printf ("c_balance = %7.2f\n", (float)c_balance/100);
    printf ("o_id = %d\n", o_id);
    datelen = sizeof(o_entry_d);

OCIERROR(errhp,OCIDateToText(errhp,&o_entry_d_base,(text*)FULLDATE.SIZ(FULL
DATE),(text*
)0,0,&datelen,o_entry_d);
    printf ("o_entry_d = %s\n", o_entry_d);
    printf ("o_carrier_id = %d\n", o_carrier_id);
    printf ("o_ol_cnt = %d\n", o_ol_cnt);
    printf ("-----\n\n", sdate);

if (!secondread) {
    printf ("Sleep before re-read order at: %s\n", sdate);
    sleep (30);
    sysdate (sdate);
    printf ("Wake up and reread at: %s\n", sdate);
    secondread = 1;
    goto retry;
}
#endif /* ISO9 */
}
octx->o_l_w_id_len = sizeof(int);
octx->o_l_d_id_len = sizeof(int);
octx->o_l_o_id_len = sizeof(int);

execstatus = OCISmtExecute(tpcsvc,octx->curo3,errhp,o_ol_cnt,0,
    NULLP(CONST OCISnapshot),NULLP(OCISnapshot),
    OCI_DEFAULT | OCI_COMMIT_ON_SUCCESS);
if (execstatus != OCI_SUCCESS )
{
    errcode=OCIERROR(errhp,execstatus);
    DISCARD OCITransCommit(tpcsvc,errhp,OCI_DEFAULT);
    if((errcode == NOT_SERIALIZABLE) || (errcode == RECOVERR)
        || (errcode == SNAPSHOT_TOO_OLD))
    {
        retries++;
        goto retry;
    }
    else
    {
        return -1;
    }
}
}
/* clean up and convert the delivery dates */
for (i = 0; i < o_ol_cnt; i++)
{
    ol_del_len[i]=sizeof(ol_delivery_d[i]);
    DISCARD OCIERROR(errhp,OCIDateToText(errhp,&ol_d_base[i],
        (const text*)SHORTDATE,(ub1)strlen(SHORTDATE),(text*)0,0,
        &ol_del_len[i], ol_delivery_d[i]);
}
/*
    cvtdmy(ol_d_base[i],ol_delivery_d[i]);
*/
}

```

```

    return (0);
}

void tkvcodone ()
{
    if (octx)
        free (octx);
}

/* end of file tkvcord.c */

client/oracle/plsto.c

#ifdef RCSID
static char *RCSid =
    "$Header: plsto.c,v 1.3 2003/07/01 15:40:19 mliu Exp $ Copyr (c) 1994 Oracle";
#endif /* RCSID */

/*=====
====+
| Copyright (c) 1994 Oracle Corp. Redwood Shores, CA |
| OPEN SYSTEMS PERFORMANCE GROUP |
| All Rights Reserved |
=====
====+
| FILENAME
| plsto.c
| DESCRIPTION
| OCI version of STOCK LEVEL transaction in TPC-C benchmark.
=====
====*/

#include "ora_tpc.h"

#ifdef PLSQLSTO
#define SQLTXT "BEGIN stocklevel.getstocklevel (:w_id, :d_id, :threshold, \
:low_stock); END;"
#else
#define SQLTXT "SELECT /*+ nocache(stok) */ count (DISTINCT s_i_id) \
FROM ordl, stok, dist \
WHERE d_id = :d_id AND d_w_id = :w_id AND \
d_id = ol_d_id AND d_w_id = ol_w_id AND \
ol_i_id = s_i_id AND ol_w_id = s_w_id AND \
s_quantity < :threshold AND \
ol_o_id BETWEEN (d_next_o_id - 20) AND (d_next_o_id - 1) \
order by ol_o_id desc"

#endif

struct stoctx {
    OCISmt *curs;
    OCIBind *w_id_bp;
    OCIBind *d_id_bp;
    OCIBind *threshold_bp;
#ifdef PLSQLSTO
    OCIBind *low_stock_bp;
#else
    OCIDefine *low_stock_bp;
#endif
    int norow;
};

typedef struct stoctx stoctx;

stoctx *sctx;

tkvcinit ()
{
    text stmbuf[SQL_BUF_SIZE];
    sctx = (stoctx *)malloc(sizeof(stoctx));
    memset(sctx, char0, sizeof(stoctx));

    sctx->norow=0;

    OCIERROR(errhp,
        OCHandleAlloc(tpcenv,(dvoid**)&sctx->curs,OCI_HTYPE_STMT,0,(dvoid**)0));
    sprintf ((char *) stmbuf, SQLTXT);
    OCIERROR(errhp,OCISmtPrepare(sctx->curs,errhp,stmbuf,strlen((char *)stmbuf),

```

```

                OCLNTV_SYNTAX,OCL_DEFAULT));
#endifdef PLSQLSTO
    OCIERROR(errhp,
        OCIAttrSet(sctx->curs,OCL_HTYPE_STMT,(dvoid*)&sctx->norow,0,
            OCI_ATTR_PREFETCH_ROWS,errhp));
#endif

/* bind variables */

OCIBND(sctx->curs,sctx->w_id_bp,errhp,":w_id",ADR(w_id),sizeof(int),
    SQLT_INT);
OCIBND(sctx->curs,sctx->d_id_bp,errhp,":d_id",ADR(d_id),sizeof(int),
    SQLT_INT);
OCIBND(sctx->curs,sctx->threshold_bp,errhp,":threshold",ADR(threshold),
    sizeof(int),SQLT_INT);
#endifdef PLSQLSTO
OCIBND(sctx->curs,sctx->low_stock_bp,errhp,":low_stock",ADR(low_stock),
    sizeof(int),SQLT_INT);
#else
OCIDEFINE(sctx->curs,sctx->low_stock_bp,errhp,1,ADR(low_stock),
    sizeof(int),SQLT_INT);
#endif

return (0);
}

tkvcs ()
{
retry:
    execstatus= OCISmtExecute(tpcsc,sctx->curs,errhp,1,0,0,0,
        OCL_COMMIT_ON_SUCCESS | OCL_DEFAULT);
    if (execstatus != OCL_SUCCESS)
    {
        errcode=OCIERROR(errhp,execstatus);
        OCITransCommit(tpcsc,errhp,OCL_DEFAULT);
        if((errcode == NOT_SERIALIZABLE) || (errcode == RECOVER)
            || (errcode == SNAPSHOT_TOO_OLD))
        {
            retries++;
            goto retry;
        }
        else {
            return -1;
        }
    }

return (0);
}

void tkvcsdone ()
{
    if(sctx) free(sctx);
}

```

client/oracle/pldel.c

```

#endifdef RCSID
static char *RCSid =
    "$Header: pldel.c,v 1.3 2003/07/01 15:33:25 mliu Exp $ Copyr (c) 1994 Oracle";
#endifdef /* RCSID */

/*=====
|   Copyright (c) 1996 Oracle Corp. Redwood Shores, CA   |
|   OPEN SYSTEMS PERFORMANCE GROUP                       |
|   All Rights Reserved                                   |
=====
+=====
| FILENAME
| pldel.c
| DESCRIPTION
| OCI version of DELIVERY transaction in TPC-C benchmark.
=====
+=====
*/

#include "ora_tpcc.h"
#endifdef TUX
#include <userlog.h>

```

```

#endif

/*
extern int userlog();
*/

#define DMLRETDDEL

#define SQLTXT "BEGIN inittpc.init_del ; END;"

#define SQLTXT1 "DELETE FROM nord WHERE no_d_id = :d_id \
    AND no_w_id = :w_id and rownum <= 1 \
    RETURNING no_o_id into :o_id"

#define SQLTXT3 "UPDATE ordr SET o_carrier_id = :carrier_id \
    WHERE o_id = :o_id and o_d_id = :d_id and o_w_id = :w_id \
    returning o_c_id into :o_c_id"

#define SQLTXT4 "UPDATE ordl \
    SET ol_delivery_d = :cr_date \
    WHERE ol_w_id = :w_id AND ol_d_id = :d_id AND ol_o_id = :o_id \
    RETURNING sum(ol_amount) into :ol_amount"

#define SQLTXT6 "UPDATE cust SET c_balance = c_balance + :amt, \
    c_delivery_cnt = c_delivery_cnt + 1 WHERE c_w_id = :w_id AND \
    c_d_id = :d_id AND c_id = :c_id"

#define NDISTS 10
#define ROWIDLEN 20

struct delctx {
    sb2 del_o_id_ind[NDISTS];
    sb2 del_d_id_ind[NDISTS];
    sb2 c_id_ind[NDISTS];
    sb2 del_date_ind[NDISTS];
    sb2 carrier_id_ind[NDISTS];
    sb2 amt_ind[NDISTS];

    ub4 del_o_id_len[NDISTS];
    ub4 c_id_len[NDISTS];
    int oid_ctx;
    int cid_ctx;
    OCIBind *olamt_bp;

    ub2 w_id_len[NDISTS];
    ub2 d_id_len[NDISTS];
    ub2 del_date_len[NDISTS];
    ub2 carrier_id_len[NDISTS];
    ub2 amt_len[NDISTS];

    ub2 del_o_id_rcode[NDISTS];
    ub2 cons_rcode[NDISTS];
    ub2 w_id_rcode[NDISTS];
    ub2 d_id_rcode[NDISTS];
    ub2 c_id_rcode[NDISTS];
    ub2 del_date_rcode[NDISTS];
    ub2 carrier_id_rcode[NDISTS];
    ub2 amt_rcode[NDISTS];

    int del_o_id[NDISTS];
    int del_d_id[NDISTS];
    int cons[NDISTS];
    int w_id[NDISTS];
    int d_id[NDISTS];
    int c_id[NDISTS];
    int carrier_id[NDISTS];
    int amt[NDISTS];
    ub4 del_o_id_rcnt;
    int retry;
    OCIRowid *no_rowid_ptr[NDISTS];
    OCIRowid *o_rowid_ptr[NDISTS];
    OCIDate del_date[NDISTS];
    OCISmt *curd0;
    OCISmt *curd1;
    OCISmt *curd2;
    OCISmt *curd3;
    OCISmt *curd4;
    OCISmt *curd5;
    OCISmt *curd6;
    OCISmt *curdtest;

    OCIBind *w_id_bp;
    OCIBind *w_id_bp3;
    OCIBind *w_id_bp4;
    OCIBind *w_id_bp5;
    OCIBind *w_id_bp6;
    OCIBind *d_id_bp;
    OCIBind *d_id_bp3;
    OCIBind *d_id_bp4;
    OCIBind *d_id_bp6;
    OCIBind *o_id_bp;
    OCIBind *cr_date_bp;
    OCIBind *c_id_bp;
    OCIBind *c_id_bp3;
}

```

```

OCIBind *no_rowid_bp;
OCIBind *carrier_id_bp;
OCIBind *o_rowid_bp;
OCIBind *del_o_id_bp;
OCIBind *del_o_id_bp3;
OCIBind *amt_bp;
OCIBind *bstr1_bp[10];
OCIBind *bstr2_bp[10];
OCIBind *retry_bp;
OCIDefine *inum_dp;
OCIDefine *d_id_dp;
OCIDefine *del_o_id_dp;
OCIDefine *no_rowid_dp;
OCIDefine *c_id_dp;
OCIDefine *o_rowid_dp;
OCIDefine *cons_dp;
OCIDefine *amt_dp;

int norow;
};

typedef struct delctx delctx;
struct pldelctx {

ub2 del_d_id_len[NDISTS];
ub2 del_o_id_len[NDISTS];

ub2 w_id_len;
ub2 d_id_len[NDISTS];
ub2 o_c_id_len[NDISTS];
ub2 sums_len[NDISTS];
ub2 carrier_id_len;
ub2 ordcnt_len;
ub2 del_date_len;

int del_o_id[NDISTS];
int del_d_id[NDISTS];
int o_c_id[NDISTS];
int sums[NDISTS];
OCIDate del_date;
int carrier_id;
int ordcnt;

ub4 del_o_id_rcnt;
ub4 del_d_id_rcnt;
ub4 o_c_id_rcnt;
ub4 sums_rcnt;

int retry;
OCISmt *curp1;
OCISmt *curp2;
OCIBind *w_id_bp;
OCIBind *d_id_bp;
OCIBind *o_id_bp;
OCIBind *o_c_id_bp;
OCIBind *ordcnt_bp;
OCIBind *sums_bp;
OCIBind *del_date_bp;
OCIBind *carrier_id_bp;
OCIBind *retry_bp;

int norow;
};

```

```

typedef struct pldelctx pldelctx;

```

```

static pldelctx *pldelctx;

```

```

static delctx *detx;

```

```

#ifdef DMLRETDL

```

```

struct amtctx {
int ol_amt[NITEMS];
sb2 ol_amt_ind[NITEMS];
ub4 ol_amt_len[NITEMS];
ub2 ol_amt_rcode[NITEMS];
int ol_cnt;
};

```

```

typedef struct amtctx amtctx;
amtctx *actx;

```

```

#endif

```

```

#ifdef DMLRETDL

```

```

sb4 no_data(dvoid *ctxp, OCIBind *bp, ub4 iter, ub4 index,
dvoid **bufpp, ub4 *alenp, ub1 *piecep,
dvoid **indpp)

```

```

{
*bufpp = (dvoid*)0;
*alenp = 0;
*indpp = (dvoid*)0;
*piecep = OCL_ONE_PIECE;

```

```

return (OCL_CONTINUE);
}

```

```

sb4 TPC_oid_data(dvoid *ctxp, OCIBind *bp, ub4 iter, ub4 index,
dvoid **bufpp, ub4 *alenp, ub1 *piecep,
dvoid **indpp, ub2 **rcodepp)

```

```

{
*bufpp = &dctx->del_o_id[iter];
*indpp = &dctx->del_o_id_ind[iter];
dctx->del_o_id_len[iter]=sizeof(dctx->del_o_id[0]);
*alenp = &dctx->del_o_id_len[iter];
*rcodepp = &dctx->del_o_id_rcode[iter];
*piecep = OCL_ONE_PIECE;
return (OCL_CONTINUE);
}

```

```

sb4 cid_data(dvoid *ctxp, OCIBind *bp, ub4 iter, ub4 index,
dvoid **bufpp, ub4 *alenp, ub1 *piecep,
dvoid **indpp, ub2 **rcodepp)

```

```

{
*bufpp = &dctx->c_id[iter];
*indpp = &dctx->c_id_ind[iter];
dctx->c_id_len[iter]=sizeof(dctx->c_id[0]);
*alenp = &dctx->c_id_len[iter];
*rcodepp = &dctx->c_id_rcode[iter];
*piecep = OCL_ONE_PIECE;
return (OCL_CONTINUE);
}

```

```

#ifdef OLD

```

```

sb4 amt_data(dvoid *ctxp, OCIBind *bp, ub4 iter, ub4 index,
dvoid **bufpp, ub4 *alenp, ub1 *piecep,
dvoid **indpp, ub2 **rcodepp)

```

```

{
amtctx *actx;
actx =(amtctx*)ctxp;
actx->ol_cnt=actx->ol_cnt+1;
*bufpp = &actx->ol_amt[index];
*indpp = &actx->ol_amt_ind[index];
actx->ol_amt_len[index]=sizeof(actx->ol_amt[0]);
*alenp = &actx->ol_amt_len[index];
*rcodepp = &actx->ol_amt_rcode[index];
*piecep = OCL_ONE_PIECE;
if (iter == 1)
return (OCL_CONTINUE);
else
return (OCL_ERROR);
}

```

```

#else

```

```

sb4 amt_data(dvoid *ctxp, OCIBind *bp, ub4 iter, ub4 index,
dvoid **bufpp, ub4 *alenp, ub1 *piecep,
dvoid **indpp, ub2 **rcodepp)

```

```

{
amtctx *actx;
actx =(amtctx*)ctxp;
*bufpp = &actx->ol_amt[index];
*indpp = &actx->ol_amt_ind[index];
actx->ol_amt_len[index]=sizeof(actx->ol_amt[0]);
*alenp = &actx->ol_amt_len[index];
*rcodepp = &actx->ol_amt_rcode[index];
*piecep = OCL_ONE_PIECE;
return (OCL_CONTINUE);
}

```

```

#endif

```

```

#endif

```

```

tkvcldinit (int plsqliflag)

```

```

{

```

```

text stmbuf[SQL_BUF_SIZE];

```

```

if (plsqliflag)

```

```

{
pldelctx = (pldelctx *) malloc (sizeof(pldelctx));
DISCARD memset(pldelctx,(char)0,(ub4)sizeof(pldelctx));
/* Initialize */
DISCARD OCIHandleAlloc((tpeenv, (dvoid**)&pldelctx->curp1, OCI_HTYPE_STMT, 0,
(dvoid**)0);
DISCARD sprintf ((char *) stmbuf, SQLTXT);
DISCARD OCISmtPrepare(pldelctx->curp1, errhp, stmbuf,
(ub4) strlen((char *)stmbuf),
OCI_NTV_SYNTAX, OCI_DEFAULT);
DISCARD OCIERROR(errhp,
OCISmtExecute((tpevc,pldelctx->curp1,errhp,1,0,NULL,(OCISnapshot),
NULL,(OCISnapshot), OCI_DEFAULT));

```

```

DISCARD OCIHandleAlloc((tpeenv,(dvoid**)&pldelctx->curp2, OCI_HTYPE_STMT,
0, (dvoid**)0);
#ifdef ISO5 || defined(ISO6) || defined(ISO8)
#ifdef ISO5

```

```

    sqlfile("../blocks/tkvcpedel_iso5.sql",stmbuf);
#endif
#if defined(ISO6)
    sqlfile("../blocks/tkvcpedel_iso6.sql",stmbuf);
#endif
#if defined(ISO8)
    sqlfile("../blocks/tkvcpedel_iso8.sql",stmbuf);
#endif
#else
    sqlfile("../blocks/tkvcpedel.sql",stmbuf);
#endif
DISCARD OCISmtPrepare(pldctx->curp2, errhp, stmbuf,
    (ub4)strlen((char *)stmbuf), OCI_NTV_SYNTAX, OCI_DEFAULT);
OCIBNDPL(pldctx->curp2, pldctx->w_id_bp, errhp, "w_id",
    ADR(w_id), SIZ(int), SQT_INT, &pldctx->w_id_len);
OCIBNDPL(pldctx->curp2, pldctx->ordcnt_bp, errhp, "ordcnt",
    ADR(pldctx->ordcnt), SIZ(int), SQT_INT, &pldctx->ordcnt_len);
OCIBNDPL(pldctx->curp2, pldctx->del_date_bp, errhp, "now",
    ADR(pldctx->del_date), SIZ(OCIDate), SQT_ODT, &pldctx->del_date_len);
OCIBNDPL(pldctx->curp2, pldctx->carrier_id_bp, errhp,
    "carrier_id", ADR(o_carrier_id), SIZ(int),
    SQT_INT, &pldctx->carrier_id_len);

OCIBNDPLA(pldctx->curp2, pldctx->d_id_bp, errhp, "d_id",
    pldctx->del_d_id, SIZ(int), SQT_INT, pldctx->del_d_id_len,
    NDISTS, &pldctx->del_d_id_rcnt);
OCIBNDPLA(pldctx->curp2, pldctx->o_id_bp, errhp, "order_id",
    pldctx->del_o_id, SIZ(int), SQT_INT, pldctx->del_o_id_len, NDISTS,
    &pldctx->del_o_id_rcnt);
OCIBNDPLA(pldctx->curp2, pldctx->sums_bp, errhp, "sums",
    pldctx->sums, SIZ(int), SQT_INT, pldctx->sums_len, NDISTS,
    &pldctx->sums_rcnt);
OCIBNDPLA(pldctx->curp2, pldctx->o_c_id_bp, errhp, "o_c_id",
    pldctx->o_c_id, SIZ(int), SQT_INT, pldctx->o_c_id_len, NDISTS,
    &pldctx->o_c_id_rcnt);
OCIBND(pldctx->curp2, pldctx->retry_bp, errhp, "retry",
    ADR(pldctx->retry), SIZ(int), SQT_INT);
}
else
{
    dctx = (dctx *) malloc (sizeof(dctx));
    memset(dctx, (char)0, sizeof(dctx));
    dctx->norow = 0;
    actx = (amtctx *) malloc (sizeof(amtctx));
    memset(actx, (char)0, sizeof(amtctx));

    OCIHandleAlloc(tpcenv, (dvoid **)&dctx->curd1, OCI_HTYPE_STMT, 0,
        (dvoid**)0);
    DISCARD sprintf ((char *) stmbuf, "%s", SQT_TXT1);
    DISCARD OCISmtPrepare(dctx->curd1, errhp, stmbuf,
        strlen((char *)stmbuf), OCI_NTV_SYNTAX, OCI_DEFAULT);

    OCIBND(dctx->curd1, dctx->w_id_bp, errhp, "w_id", dctx->w_id, SIZ(int),
        SQT_INT);
    OCIBNDRA(dctx->curd1, dctx->d_id_bp, errhp, "d_id", dctx->d_id, SIZ(int),
        SQT_INT, NULL, NULL, NULL);

    OCIBNDRAD(dctx->curd1, dctx->del_o_id_bp, errhp, "o_id",
        SIZ(int), SQT_INT, NULL,
        &dctx->oid_ctx.no_data.TPC_oid_data);

    /* open third cursor */

    DISCARD OCIHandleAlloc(tpcenv, (dvoid **)&dctx->curd3, OCI_HTYPE_STMT,
        0, (dvoid**)0);
    DISCARD sprintf ((char *) stmbuf, SQT_TXT3);
    DISCARD OCISmtPrepare(dctx->curd3, errhp, stmbuf, strlen((char *)stmbuf),
        OCI_NTV_SYNTAX, OCI_DEFAULT);

    /* bind variables */

    OCIBNDRA(dctx->curd3, dctx->carrier_id_bp, errhp, "carrier_id",
        dctx->carrier_id, SIZ(dctx->carrier_id[0]), SQT_INT,
        dctx->carrier_id_ind, dctx->carrier_id_len, dctx->carrier_id_rcode);

    OCIBNDRA(dctx->curd3, dctx->w_id_bp3, errhp, "w_id", dctx->w_id, SIZ(int),
        SQT_INT, NULL, NULL, NULL);
    OCIBNDRA(dctx->curd3, dctx->d_id_bp3, errhp, "d_id", dctx->d_id, SIZ(int),
        SQT_INT, NULL, NULL, NULL);
    OCIBNDRA(dctx->curd3, dctx->del_o_id_bp3, errhp, "o_id", dctx->del_o_id,
        SIZ(int), SQT_INT, NULL, NULL, NULL);
    OCIBNDRAD(dctx->curd3, dctx->c_id_bp3, errhp, "o_c_id", SIZ(int),
        SQT_INT, NULL, &dctx->cid_ctx.no_data, cid_data);

    /* open fourth cursor */

    DISCARD OCIHandleAlloc(tpcenv, (dvoid **)&dctx->curd4, OCI_HTYPE_STMT,
        0,
        (dvoid**)0);
    DISCARD sprintf ((char *) stmbuf, SQT_TXT4);
    DISCARD OCISmtPrepare(dctx->curd4, errhp, stmbuf, strlen((char *)stmbuf),
        OCI_NTV_SYNTAX, OCI_DEFAULT);
}

}

void shiftdata(from)
int from;
{
    int i;
    for (i=from; i<NDISTS-1; i++)
    {
        dctx->del_o_id_ind[i] = dctx->del_o_id_ind[i+1];
        dctx->del_o_id[i] = dctx->del_o_id[i+1];
        dctx->w_id[i] = dctx->w_id[i+1];
        dctx->d_id[i] = dctx->d_id[i+1];
        dctx->carrier_id[i] = dctx->carrier_id[i+1];
    }
}

tkvcd (int plsqliflag)
{
    int i, j;
    int rpe_count, count;
    int invalid;

    if (plsqliflag)
    {
        pldctx->w_id_len = sizeof (int);
        pldctx->carrier_id_len = sizeof (int);
        for (i = 0; i < NDISTS; i++)
        {
            pldctx->del_o_id_len[i] = sizeof(int);
            del_o_id[i] = 0;
        }
        pldctx->del_date_len = DEL_DATE_LEN;
        DISCARD memcpy(&pldctx->del_date, &cr_date, sizeof(OCIDate));

        pldctx->retry=0;

        DISCARD OCIErrror(errhp,
            OCISmtExecute(tpcvc, pldctx->curp2, errhp, 1, 0, NULL, (CONST OCISnapshot),
                NULL, (OCISnapshot), OCI_DEFAULT));
        for (i = 0; i < NDISTS; i++)
        {
            del_o_id[i] = 0;
        }
        for (i = 0; i < pldctx->del_o_id_rcnt; i++)
            del_o_id[pldctx->del_d_id[i] - 1] = pldctx->del_o_id[i];
    }
    else
    {
        retry:
    }
}

```

```

invalid = 0;

/* initialization for array operations */
for (i = 0; i < NDISTS; i++)
{
    dctx->del_o_id_ind[i] = TRUE;
    dctx->d_id_ind[i] = TRUE;
    dctx->c_id_ind[i] = TRUE;
    dctx->del_date_ind[i] = TRUE;
    dctx->carrier_id_ind[i] = TRUE;
    dctx->amt_ind[i] = TRUE;

    dctx->del_o_id_len[i] = SIZ(dctx->del_o_id[0]);
    dctx->w_id_len[i] = SIZ(dctx->w_id[0]);
    dctx->d_id_len[i] = SIZ(dctx->d_id[0]);
    dctx->c_id_len[i] = SIZ(dctx->c_id[0]);
    dctx->del_date_len[i] = DEL_DATE_LEN;
    dctx->carrier_id_len[i] = SIZ(dctx->carrier_id[0]);
    dctx->amt_len[i] = SIZ(dctx->amt[0]);

    dctx->w_id[i] = w_id;
    dctx->d_id[i] = i+1;
    dctx->carrier_id[i] = o_carrier_id;
    memcpy(&dctx->del_date[i], &cr_date, sizeof(OCIDate));
}

memset(actx, (char)0, sizeof(amtctx));

/* array select from new_order and orders tables */
execstatus=OCISmtExecute(tpcsvc, dctx->curd1, errhp, NDISTS, 0,
    NULLP(CONST OCISnapshot), NULLP(OCISnapshot), OCI_DEFAULT);
if((execstatus != OCI_SUCCESS) && (execstatus != OCI_NO_DATA))
{
    DISCARD OCITransRollback(tpcsvc, errhp, OCI_DEFAULT);
    errcode = OCIERROR(errhp, execstatus);
    if(errcode == NOT_SERIALIZABLE)
    {
        retries++;
        goto retry;
    }
    else if (errcode == RECOVERR)
    {
        retries++;
        goto retry;
    }
    else if (errcode == SNAPSHOT_TOO_OLD)
    {
        retries++;
        goto retry;
    }
    else
    {
        return -1;
    }
}
/* mark districts with no new order */
DISCARD OCIAAttrGet(dctx->curd1, OCI_HTYPE_STMT, &rcount, NULLP(ub4),
    OCI_ATTR_ROW_COUNT, errhp);
rpc = rcount;
if (rcount != NDISTS)
{
    int j = 0;
    for (i=0; i < NDISTS; i++)
    {
        if (dctx->del_o_id_ind[j] == 0) /* there is data here */
            j++;
        else
            shiftdata(j);
    }
}

execstatus=OCISmtExecute(tpcsvc, dctx->curd3, errhp, rpc, 0,
    NULLP(CONST OCISnapshot), NULLP(OCISnapshot), OCI_DEFAULT);
if(execstatus != OCI_SUCCESS)
{
    DISCARD OCITransRollback(tpcsvc, errhp, OCI_DEFAULT);
    errcode = OCIERROR(errhp, execstatus);
    if(errcode == NOT_SERIALIZABLE)
    {
        retries++;
        goto retry;
    }
    else if (errcode == RECOVERR)
    {
        retries++;
        goto retry;
    }
    else if (errcode == SNAPSHOT_TOO_OLD)
    {
        retries++;
        goto retry;
    }
    else
}

```

```

{
    return -1;
}

DISCARD OCIAAttrGet(dctx->curd3, OCI_HTYPE_STMT, &rcount, NULLP(ub4),
    OCI_ATTR_ROW_COUNT, errhp);

if (rcount != rpc)
{
#ifdef TUX
    userlog ("Error in TPC-C server %d: %d rows selected, %d ords updated\n",
        proc_no, rpc, rcount);
#else
    DISCARD fprintf (stderr,
        "Error in TPC-C server %d: %d rows selected, %d ords updated\n",
        proc_no, rpc, rcount);
#endif
    DISCARD OCITransRollback(tpcsvc, errhp, OCI_DEFAULT);
    return (-1);
}

/* array update of order_line table */
execstatus=OCISmtExecute(tpcsvc, dctx->curd4, errhp, rpc, 0,
    NULLP(CONST OCISnapshot), NULLP(OCISnapshot), OCI_DEFAULT);
if(execstatus != OCI_SUCCESS)
{
    DISCARD OCITransRollback(tpcsvc, errhp, OCI_DEFAULT);
    errcode = OCIERROR(errhp, execstatus);
    if(errcode == NOT_SERIALIZABLE)
    {
        retries++;
        goto retry;
    }
    else if (errcode == RECOVERR)
    {
        retries++;
        goto retry;
    }
    else if (errcode == SNAPSHOT_TOO_OLD)
    {
        retries++;
        goto retry;
    }
    else
    {
        return -1;
    }
}
DISCARD OCIAAttrGet(dctx->curd4, OCI_HTYPE_STMT, &rcount, NULLP(ub4),
    OCI_ATTR_ROW_COUNT, errhp);
/* transfer amounts */
for (i=0; i < rpc; i++)
{
    dctx->amt[i]=0;
    if (actx->ol_amt_rcode[i] == 0)
    {
        dctx->amt[i] = actx->ol_amt[i];
    }
}
#ifdef OLD
if (rcount > rpc) {
    userlog
        ("Error in TPC-C server %d: %d ordnrs updated, %d ordl updated\n",
        proc_no, rpc, rcount);
}
#endif

/* array update of customer table */
execstatus=OCISmtExecute(tpcsvc, dctx->curd6, errhp, rpc, 0,
    NULLP(CONST OCISnapshot), NULLP(OCISnapshot),
    OCI_COMMIT_ON_SUCCESS | OCI_DEFAULT);

if(execstatus != OCI_SUCCESS)
{
    OCITransRollback(tpcsvc, errhp, OCI_DEFAULT);
    errcode = OCIERROR(errhp, execstatus);
    if(errcode == NOT_SERIALIZABLE)
    {
        retries++;
        goto retry;
    }
    else if (errcode == RECOVERR)
    {
        retries++;
        goto retry;
    }
    else if (errcode == SNAPSHOT_TOO_OLD)
    {
        retries++;
        goto retry;
    }
    else
    {
        return -1;
    }
}

```

```

    }
}

DISCARD OCIAAttrGet(dctx->curd0,OCI_HTYPE_STMT,&rcount,NULLP(ub4),
    OCI_ATTR_ROW_COUNT,errhp);

if (rcount != rpc) {
#ifdef TUX
    userlog ("Error in TPC-C server %d: %d rows selected, %d cust updated\n",
        proc_no, rpc, rcount);
#else
    DISCARD fprintf (stderr,
        "Error in TPC-C server %d: %d rows selected, %d cust updated\n",
        proc_no, rpc, rcount);
#endif
DISCARD OCITransRollback(tpsvcs, errhp, OCI_DEFAULT);
return (-1);
}

/* return o_id's in district id order */

for (i = 0; i < NDISTS; i++)
    del_o_id[i] = 0;
for (i = 0; i < rpc; i++)
    del_o_id[dctx->d_id[i] - 1] = dctx->del_o_id[i];
}
return (0);
}

void tkvcdone (int plsqflag)
{
if (plsqflag)
{
if (pldctx)
{
DISCARD OCIHandleFree((dvoid *)dctx->curd0,OCI_HTYPE_STMT);
DISCARD free(pldctx);
}
else
{
(dctx)
{
OCIHandleFree((dvoid *)dctx->curd1,OCI_HTYPE_STMT);
OCIHandleFree((dvoid *)dctx->curd2,OCI_HTYPE_STMT);
OCIHandleFree((dvoid *)dctx->curd3,OCI_HTYPE_STMT);
OCIHandleFree((dvoid *)dctx->curd4,OCI_HTYPE_STMT);
OCIHandleFree((dvoid *)dctx->curd5,OCI_HTYPE_STMT);
OCIHandleFree((dvoid *)dctx->curd6,OCI_HTYPE_STMT);
DISCARD free (dctx);
}
}
}
}
}

```

client/oracle/ora_tpcc.h

```

/*
 * $Header: ora_tpcc.h,v 1.3 2003/07/01 15:31:08 mliu Exp $ Copyr (c) 1993 Oracle
 */
/*=====
====+
| Copyright (c) 1995 Oracle Corp, Redwood Shores, CA |
| OPEN SYSTEMS PERFORMANCE GROUP |
| All Rights Reserved |
+=====
====+
| FILENAME
| tpcc.h
| DESCRIPTION
| Include file for TPC-C benchmark programs.
+=====
====*/

#ifdef TPCC_H
#define TPCC_H

#ifdef FALSE
# define FALSE 0
#endif

#ifdef TRUE

```

```

# define TRUE 1
#endif

#include <stdio.h>
#include <stdlib.h>
#include <ctype.h>
#include <string.h>

#ifdef boolean
#define boolean int
#endif

#include <oratypes.h>
#include <oci.h>
#include <ocidfn.h>
/*
#ifdef __STDC__
#include "ociapr.h"
#else
#include "ocikpr.h"
#endif
*/

typedef struct cda_def csrddef;
typedef struct cda_def ldadef;

/* TPC-C transaction functions */

extern int TPCcinit ();
extern int TPCcnew ();
extern int TPCcpay ();
extern int TPCcord ();
extern int TPCcdel ();
extern int TPCcsto ();
extern void TPCcexit ();
extern int TPCcdumpinit ();
extern void TPCcdumpnew ();
extern void TPCcdumppay ();
extern void TPCcdumpord ();
extern void TPCcdumpdel ();
extern void TPCcdumpsto ();
extern void TPCcdumpexit ();
extern void userlog(char* fmt, ...);

/* Error codes */

#define RECOVERR -10
#define IRRECERR -20
#define NOERR 111
#define DEL_ERROR -666
#define DEL_DATE_LEN 7
#define NDISTS 10
#define NITEMS 15
#define SQL_BUF_SIZE 8192

#define FULLDATE "dd-mon-yy.hh24:mi:ss"
#define SHORTDATE "dd-mm-yyyy"

#define DELRT 80.0

extern int tkvcninit ();
extern int tkvcpinit ();
extern int tkvcinit ();
extern int tkvedinit ();
extern int tkvesinit ();

extern int tkvcn ();
extern int tkvcp ();
extern int tkvco ();
extern int tkvcd ();
extern int tkvcs ();

extern void tkvcdone ();
extern void tkvcpdone ();
extern void tkvcodone ();
extern void tkvcdone ();
extern void tkvcsdone ();

extern int tkvcss (); /* for alter session to get memory size and trace */
extern boolean multitrans;
extern int ord_init;

extern void errprt ();
extern int ocierror(char *fname, int lineno,OCIError *errhp, sword status);
extern int sqlfile(char *fname, text *linebuf);

extern FILE *lfp;
extern FILE *fopen ();
extern int proc_no;
extern int doid[];

```

```

extern int execstatus;
extern int errcode;

extern OCIEnv *tpcenv;
extern OCIServer *tpcsrv;
extern OCIError *errhp;
extern OCISvcCtx *tpscvc;
extern OCISession *tpcsur;
extern OCISmt *curntest;
/* The bind and define handles for each transaction are
   included in their respective header files. */

/* for stock-level transaction */

extern int w_id;
extern int d_id;
extern int c_id;
extern int threshold;
extern int low_stock;

/* for delivery transaction */

extern int del_o_id[10];
extern int carrier_id;
extern int retries;

/* for order-status transaction */

extern int bylastname;
extern char c_last[17];
extern char c_first[17];
extern char c_middle[3];
extern double c_balance;
extern int o_id;
extern text o_entry_d[20];
extern int o_carrier_id;
extern int o_ol_cnt;
extern int ol_supply_w_id[15];
extern int ol_i_id[15];
extern int ol_quantity[15];
extern int ol_amount[15];
ub4 ol_del_len[15];
extern text ol_delivery_d[15][11];
/* xnie - begin */
extern OCIRowid *o_rowid;
/* xnie - end */

/* for payment transaction */

extern int c_w_id;
extern int c_d_id;
extern int h_amount;
extern char w_street_1[21];
extern char w_street_2[21];
extern char w_city[21];
extern char w_state[3];
extern char w_zip[10];
extern char d_street_1[21];
extern char d_street_2[21];
extern char d_city[21];
extern char d_state[3];
extern char d_zip[10];
extern char c_street_1[21];
extern char c_street_2[21];
extern char c_city[21];
extern char c_state[3];
extern char c_zip[10];
extern char c_phone[17];
extern text c_since_d[11];
extern char c_credit[3];
extern int c_credit_lim;
extern float c_discount;
extern char c_data[201];
extern text h_date[20];

/* for new order transaction */

extern int nol_i_id[15];
extern int nol_supply_w_id[15];
extern int nol_quantity[15];
extern int nol_quant10[15];
extern int nol_quant91[15];
extern int nol_ytdqty[15];
extern int nol_amount[15];
extern int o_all_local;
extern float w_tax;
extern float d_tax;
extern float total_amount;
extern char i_name[15][25];
extern int i_name_strlen[15];
extern ub2 i_name_strlen_len[15];
extern ub2 i_name_strlen_rcode[15];

extern ub4 i_name_strlen_csize;
extern int s_quantity[15];
extern char brand_gen[15];
extern ub2 brand_gen_len[15];
extern ub2 brand_gen_rcode[15];
extern ub4 brand_gen_csize;
extern int i_price[15];
extern char brand_generic[15][1];
extern int status;
extern int tracelevel;

/* Miscellaneous */
extern OCIDate cr_date;
extern OCIDate c_since;
extern OCIDate o_entry_d_base;
extern OCIDate ol_d_base[15];

#ifdef DISCARD
#define DISCARD (void)
#endif

#ifdef sword
#define sword int
#endif

#define VER7 2

#define NA -1 /* ANSI SQL NULL */
#define NLT 1 /* length for string null terminator */
#define DEADLOCK 60 /* ORA-00060: deadlock */
#define NO_DATA_FOUND 1403 /* ORA-01403: no data found */
#define NOT_SERIALIZABLE 8177 /* ORA-08177: transaction not serializable */
#define SNAPSHOT_TOO_OLD 1555 /* ORA-01555: snapshot too old */

#ifdef NULLP
#define NULLP(x) (x * NULL)
#endif /* NULLP */

#define ADR(object) ((ub1 *)&(object))
#define SIZ(object) ((sword)sizeof(object))

typedef char date[24+NLT];
typedef char varchar2;

#define min(x,y) (((x) < (y)) ? (x) : (y))

#define OCIERROR(errp,function)\
ocierror(__FILE__,__LINE__,(errp),(function));

#define OCIBND(stmp, bndp, errp, sqlvar, progvl, ftype)\
ocierror(__FILE__,__LINE__,(errp),\
OCIHandleAlloc((stmp),(dvoid*)&(bndp),OCI_HTYPE_BIND,0,(dvoid**0));\
ocierror(__FILE__,__LINE__,(errp),\
OCIBindByName((stmp), &(bndp), (errp),\
(text*)(sqlvar), strlen((sqlvar)),\
(progvl), (progvl), (ftype),0,0,0,0,OCI_DEFAULT));

/* bind arrays for sql */
#define OCIBNDRA(stmp,bndp,errp,sqlvar,progvl,ftype,indp,alen,arcode)\
DISCARD ocierror(__FILE__,__LINE__,(errp),\
OCIHandleAlloc((stmp),(dvoid*)&(bndp),OCI_HTYPE_BIND,0,(dvoid**0));\
DISCARD ocierror(__FILE__,__LINE__,(errp),\
OCIBindByName((stmp),&(bndp),(errp),(text*)(sqlvar),strlen((sqlvar)),\
(progvl),(progvl),(ftype),(indp),(alen),(arcode),0,0,OCI_DEFAULT));

/* use with callback data */
#define OCIBNDRAD(stmp,bndp,errp,sqlvar,progvl,ftype,indp,ctxp,\
cbf_nodata,cbf_data)\
DISCARD ocierror(__FILE__,__LINE__,(errp),\
OCIHandleAlloc((stmp),(dvoid*)&(bndp),OCI_HTYPE_BIND,0,(dvoid**0));\
DISCARD ocierror(__FILE__,__LINE__,(errp),\
OCIBindByName((stmp),&(bndp),(errp),(text*)(sqlvar),\
strlen((sqlvar),0,(progvl),(ftype),\
indp,0,0,0,OCI_DATA_AT_EXEC));\
DISCARD ocierror(__FILE__,__LINE__,(errp),\
OCIBindDynamic((bndp),(errp),(ctxp),(cbf_nodata),(ctxp),(cbf_data));

/* bind in/out for plsql without indicator and rcode */
#define OCIBNDPL(stmp,bndp,errp,sqlvar,progvl,ftype,alen)\
DISCARD ocierror(__FILE__,__LINE__,(errp),\
OCIHandleAlloc((stmp),(dvoid*)&(bndp),OCI_HTYPE_BIND,0,(dvoid**0));\
DISCARD ocierror(__FILE__,__LINE__,(errp),\
OCIBindByName((stmp),&(bndp),(errp),(CONST text*)(sqlvar),\
(sb4)strlen((CONST char*)(sqlvar)), (dvoid*)(progvl),(progvl),(ftype),\
NULLP(dvoid),(alen), NULLP(ub2), 0,NULLP(ub4),OCI_DEFAULT));

/* bind in values for plsql with indicator and rcode */
#define OCIBNDR(stmp,bndp,errp,sqlvar,progvl,ftype,indp,alen,arcode)\
DISCARD ocierror(__FILE__,__LINE__,(errp),\

```



```

OCIHandleAlloc((stmp),(dvoid*)&(bndp),OCI_HTYPE_BIND,0,(dvoid**0)); \
DISCARD ocierror(__FILE__,__LINE__,(errp), \
OCIBindByName((stmp),&(bndp),(errp),(text*)(sqlvar),strlen(sqlvar)), \
                (progv),(progv),(ftype),(indp),(alen),(arcod),0,0, \
                OCI_DEFAULT));

/* bind in/out for plsql arrays without indicator and rcode */
#define OCIBNDPLA(stmp,bndp,errp,sqlvar,progv,ftype,alen,ms,cu) \
DISCARD ocierror(__FILE__,__LINE__,(errp), \
OCIHandleAlloc((stmp),(dvoid*)&(bndp),OCI_HTYPE_BIND,0,(dvoid**0)); \
DISCARD ocierror(__FILE__,__LINE__,(errp), \
OCIBindByName((stmp),&(bndp),(errp),(CONST text*)(sqlvar), \
                (sb4)strlen((CONST char*)(sqlvar)),(void*)(progv), \
                (progv),(ftype),NULL,(alen),NULL,(ms),(cu),OCI_DEFAULT));

/* bind in/out values for plsql with indicator and rcode */
#define OCIBNDRAA(stmp,bndp,errp,sqlvar,progv,ftype,indp,alen,arcod, \
ms,cu) \
ocierror(__FILE__,__LINE__,(errp), \
OCIHandleAlloc((stmp),(dvoid*)&(bndp),OCI_HTYPE_BIND,0,(dvoid**0)); \
ocierror(__FILE__,__LINE__,(errp), \
OCIBindByName((stmp),&(bndp),(errp),(text*)(sqlvar),strlen(sqlvar)), \
                (progv),(progv),(ftype),(indp),(alen),(arcod),(ms),(cu),OCI_DEFAULT));

#define OCIDEFINE(stmp,dfnp,errp,pos,progv,progv,ftype) \
OCIDefineByPos((stmp),&(dfnp),(errp),(pos),(progv),(progv),(ftype), \
                0,0,0,OCI_DEFAULT);

#define OCIDDEF(stmp,dfnp,errp,pos,progv,progv,ftype) \
OCIHandleAlloc((stmp),(dvoid*)&(dfnp),OCI_HTYPE_DEFINE,0, \
                (dvoid**0)); \
OCIDefineByPos((stmp),&(dfnp),(errp),(pos),(progv),(progv), \
                (ftype),NULL,NULL,NULL,OCI_DEFAULT); \

#define OCIDFNRA(stmp,dfnp,errp,pos,progv,progv,ftype,indp,alen,arcod) \
OCIHandleAlloc((stmp),(dvoid*)&(dfnp),OCI_HTYPE_DEFINE,0, \
                (dvoid**0)); \
OCIDefineByPos((stmp),&(dfnp),(errp),(pos),(progv), \
                (progv),(ftype),(indp),(alen), \
                (arcod),OCI_DEFAULT);

#define OCIDFNDYN(stmp,dfnp,errp,pos,progv,progv,ftype,indp,ctxp,cbf_data) \
ocierror(__FILE__,__LINE__,(errp), \
OCIHandleAlloc((stmp),(dvoid*)&(dfnp),OCI_HTYPE_DEFINE,0, \
                (dvoid**0)); \
ocierror(__FILE__,__LINE__,(errp), \
OCIDefineByPos((stmp),&(dfnp),(errp),(pos),(progv),(progv),(ftype), \
                (indp),NULL,NULL,OCI_DYNAMIC_FETCH)); \
ocierror(__FILE__,__LINE__,(errp), \
OCIDefineDynamic((dfnp),(errp),(ctxp),(cbf_data)));

/* New order */

struct newinstruct {
int w_id;
int d_id;
int c_id;
int ol_i_id[15];
int ol_supply_w_id[15];
int ol_quantity[15];
};

struct newoutstruct {
int terror;
int o_id;
int o_ol_cnt;
char c_last[17];
char c_credit[3];
float c_discount;
float w_tax;
float d_tax;
char o_entry_d[20];
float total_amount;
char i_name[15][25];
int s_quantity[15];
char brand_generic[15];
float i_price[15];
float ol_amount[15];
char status[26];
int retry;
};

struct newstruct {
struct newinstruct newin;
struct newoutstruct newout;
};

/* Payment */

```

```

struct payinstruct {
int w_id;
int d_id;
int c_w_id;
int c_d_id;
int c_id;
int bylastname;
float h_amount;
char c_last[17];
};

struct payoutstruct {
int terror;
char w_street_1[21];
char w_street_2[21];
char w_city[21];
char w_state[3];
char w_zip[10];
char d_street_1[21];
char d_street_2[21];
char d_city[21];
char d_state[3];
char d_zip[10];
int c_id;
char c_first[17];
char c_middle[3];
char c_last[17];
char c_street_1[21];
char c_street_2[21];
char c_city[21];
char c_state[3];
char c_zip[10];
char c_phone[17];
char c_since[11];
char c_credit[3];
double c_credit_lim;
float c_discount;
double c_balance;
char c_data[201];
char h_data[20];
int retry;
};

struct paystruct {
struct payinstruct payin;
struct payoutstruct payout;
};

/* Order status */

struct ordinstr {
int w_id;
int d_id;
int c_id;
int bylastname;
char c_last[17];
};

struct ordoutstruct {
int terror;
int c_id;
char c_last[17];
char c_first[17];
char c_middle[3];
double c_balance;
int o_id;
char o_entry_d[20];
int o_carrier_id;
int o_ol_cnt;
int ol_supply_w_id[15];
int ol_i_id[15];
int ol_quantity[15];
float ol_amount[15];
char ol_delivery_d[15][11];
int retry;
};

struct ordstruct {
struct ordinstr ordin;
struct ordoutstruct ordout;
};

/* Delivery */

struct delinstr {
int w_id;
int o_carrier_id;
double qtime;
int in_timing_int;
int plsqlflag;
};

```

```

struct deloutstruct {
    int terror;
    int retry;
};

struct delstruct {
    struct delinstruct delin;
    struct deloutstruct delout;
};

/* Stock level */

struct stoinstruct {
    int w_id;
    int d_id;
    int threshold;
};

struct stooutstruct {
    int terror;
    int low_stock;
    int retry;
};

struct stostruc {
    struct stoinstruct stoin;
    struct stooutstruct stoout;
};

#endif

```

client/oracle/tpccflags.h

```

#define PLSQLNO
#define DMLRETDL

```

tpcc_info.h

```

/*
 * $Header: tpcc_info.h 7030100.1 95/07/19 15:11:37 plai Generic<base> $ Copyr (c) 1995
 * Oracle
 */
/*=====
====+
|      Copyright (c) 1995 Oracle Corp. Redwood Shores, CA      |
|      OPEN SYSTEMS PERFORMANCE GROUP                          |
|      All Rights Reserved                                     |
|=====
====+
| FILENAME
| tpcc_info.h
| DESCRIPTION
| Include file for TPC-C benchmark programs.
|=====
====*/

#ifndef TPCC_INFO_H
#define TPCC_INFO_H

/* this set is duplicated in c_Defs.h, c_Defs.h is used for batch driver */
#define MENTXN    0    /* menu txn */
#define NEWTXN    1    /* new order transaction */
#define PAYTXN    2    /* payment transaction */
#define ORDTXN    3    /* order status transaction */
#define DELTXN    4    /* delivery transaction */
#define STOTXN    5    /* stock level transaction */
#define ALLTXN    6    /* for processing all txns */
#define ALLTXNNODEL 7    /* for processing all txns except delivery */
/* New order */

struct newinstruct {
    int w_id;
    int d_id;
    int c_id;
    int ol_i_id[15];
    int ol_supply_w_id[15];
    int ol_quantity[15];
};

struct newoutstruct {
    int terror;
    int o_id;

```

```

    int o_ol_cnt;
    char c_last[17];
    char c_credit[3];
    float c_discount;
    float w_tax;
    float d_tax;
    char o_entry_d[20];
    float total_amount;
    char i_name[15][25];
    int s_quantity[15];
    char brand_generic[15];
    float i_price[15];
    float ol_amount[15];
    char status[26];
    int retry;
};

struct newstruct {
    int retval;
    int old_quantity[15];
    struct newinstruct newin;
    struct newoutstruct newout;
};

/* Payment */

struct payinstruct {
    int w_id;
    int d_id;
    int c_w_id;
    int c_d_id;
    int c_id;
    int bylastname;
    int h_amount;
    char c_last[17];
};

struct payoutstruct {
    int terror;
    char w_street_1[21];
    char w_street_2[21];
    char w_city[21];
    char w_state[3];
    char w_zip[10];
    char d_street_1[21];
    char d_street_2[21];
    char d_city[21];
    char d_state[3];
    char d_zip[10];
    int c_id;
    char c_first[17];
    char c_middle[3];
    char c_last[17];
    char c_street_1[21];
    char c_street_2[21];
    char c_city[21];
    char c_state[3];
    char c_zip[10];
    char c_phone[17];
    char c_since[11];
    char c_credit[3];
    double c_credit_lim;
    float c_discount;
    double c_balance;
    char c_data[201];
    char h_date[20];
    int retry;
};

struct paystruct {
    int retval;
    struct payinstruct payin;
    struct payoutstruct payout;
};

/* Order status */

struct ordinstruct {
    int w_id;
    int d_id;
    int c_id;
    int bylastname;
    char c_last[17];
};

struct ordoutstruct {
    int terror;
    int c_id;
    char c_last[17];
    char c_first[17];
    char c_middle[3];
    double c_balance;
    int o_id;

```

```

char o_entry_d[20];
int o_carrier_id;
int o_ol_cnt;
int ol_supply_w_id[15];
int ol_i_id[15];
int ol_quantity[15];
float ol_amount[15];
char ol_delivery_d[15][11];
int retry;
};

```

```

struct ordstruct {
    int retval;
    struct ordinstruct ordin;
    struct ordoutstruct ordout;
};

```

/* Delivery */

```

struct delinstruct {
    int w_id;
    int o_carrier_id;
    double qtime;
    int in_timing_int;
};

```

```

struct deloutstruct {
    int terror;
    int retry;
};

```

```

struct delstruct {
    int retval;
    struct delinstruct delin;
    struct deloutstruct delout;
};

```

/* Stock level */

```

struct stoinstruct {
    int w_id;
    int d_id;
    int threshold;
};

```

```

struct stooutstruct {
    int terror;
    int low_stock;
    int retry;
};

```

```

struct stostruct {
    int retval;
    struct stoinstruct stoin;
    struct stooutstruct stoout;
};

```

/* used these definitions in client code only */

```

typedef struct delstruct DeliveryData, *pDeliveryData;
typedef struct newstruct NewOrderData, *pNewOrderData;
typedef struct paystruct PaymentData, *pPaymentData;
typedef struct ordstruct OrderStatusData, *pOrderStatusData;
typedef struct stostruct StockLevelData, *pStockLevelData;

```

#endif

tpccpl.h

```

/*
 * $Header: tpccpl.h 7030100.1 96/04/02 18:03:35 plai Generic<base> $ Copyr (c) 1994
Oracle
 */

```

```

/*=====
====+
| Copyright (c) 1994 Oracle Corp., Redwood Shores, CA |
| OPEN SYSTEMS PERFORMANCE GROUP |
| All Rights Reserved |
=====
====+
| FILENAME
| tpccpl.h

```

```

| DESCRIPTION
| Header file for TPC-C transactions in PL/SQL.

```

```

+=====
==*/

```

```

#ifndef TPCCPL_H
#define TPCCPL_H

```

```

#include <stdio.h>
#include "tpcc.h"

```

```

#ifdef TUX
#define DELRT 5.0
#else
#define DELRT 80.0
#endif

```

```

extern int plnewinit ();
extern int plpayinit ();
extern int plordinit ();
extern int pldelinit ();
extern int plstoinit ();

```

```

extern int plnew ();
extern int plpay ();
extern int plord ();
extern int pldel ();
extern int plsto ();

```

```

extern void plnewdone ();
extern void plpaydone ();
extern void plorddone ();
extern void pldeldone ();
extern void plstodone ();

```

```

extern errprt ();
extern void logerr();
extern int ocierror(char *fname, int lineno, OCIError *errhp, sword status);

```

```

extern int sqlfile(char *fname, text *linebuf);

```

```

extern void cvtdmy ( unsigned char *orady, char *outdate);
extern void cvtdmyhms (unsigned char *orady, char *outdate);

```

```

extern FILE *Ifp;
extern FILE *fopen ();
extern int proc_no;
extern int doid[];
extern int execstatus;
extern int errcode;

```

```

extern OCIEnv *tpcenv;
extern OCIServer *tpcsrv;
extern OCIError *errhp;
extern OCISvcCtx *tpcsvc;
extern OCISession *tpcusr;
extern OCISmt *curn, *curn1, *curn2, *curn3[10], *curn4;
extern OCISmt *curntest;
/* The bind and define handles for each transaction are
included in their respective header files. */

```

```

extern lodef tpclda;
extern csrdef curs;
extern csrdef curd;
extern csrdef cur0;
extern csrdef cur01;
extern csrdef cur02;
/*extern csrdef curp0;
extern csrdef curp1;
*/
/*extern csrdef curn, curn1, curn2, curn3[10], curn4;*/

```

```

extern unsigned long tpchda[];

```

/* for stock-level transaction */

```

extern int w_id;
extern int d_id;
extern int c_id;
extern int threshold;
extern int low_stock;

```

/* for delivery transaction */

```

extern int del_o_id[10];
extern int carrier_id;
extern int retries;

```

/* for order-status transaction */

```

extern int bylastname;

```

```

extern char c_last[17];
extern char c_first[17];
extern char c_middle[3];
extern double c_balance;
extern int o_id;
extern char o_entry_d[20];
extern int o_carrier_id;
extern int o_cnt;
extern int ol_supply_w_id[15];
extern int ol_i_id[15];
extern int ol_quantity[15];
extern int ol_amount[15];
extern char ol_delivery_d[15][11];

```

```
/* for payment transaction */
```

```

extern int c_w_id;
extern int c_d_id;
extern int h_amount;
extern char w_street_1[21];
extern char w_street_2[21];
extern char w_city[21];
extern char w_state[3];
extern char w_zip[10];
extern char d_street_1[21];
extern char d_street_2[21];
extern char d_city[21];
extern char d_state[3];
extern char d_zip[10];
extern char c_street_1[21];
extern char c_street_2[21];
extern char c_city[21];
extern char c_state[3];
extern char c_zip[10];
extern char c_phone[17];
extern char c_since_d[11];
extern char c_credit[3];
extern int c_credit_lim;
extern int c_discount;
extern char c_data[201];
extern char h_date[20];
extern char bad_credit[33];

```

```
/* for new order transaction */
```

```

extern int nol_i_id[15];
extern int nol_supply_w_id[15];
extern int nol_quantity[15];
extern int nol_quant10[15];
extern int nol_quant9[15];
extern int nol_ytdqty[15];
extern int nol_amount[15];
extern int o_all_local;
extern int w_tax;
extern int d_tax;
extern float total_amount;
extern char i_name[15][25];
extern int i_name_strlen[15];
extern ub2 i_name_strlen_len[15];
extern ub2 i_name_strlen_rcode[15];
extern ub4 i_name_strlen_csize;
extern int s_quantity[15];
extern char brand_gen[15];
extern ub2 brand_gen_len[15];
extern ub2 brand_gen_rcode[15];
extern ub4 brand_gen_csize;
extern int i_price[15];
extern int status;

```

```
/* Miscellaneous */
```

```

extern unsigned char cr_date[7];
extern unsigned char c_since[7];
extern unsigned char o_entry_d_base[7];
extern unsigned char ol_d_base[15][7];

```

```

#ifndef DISCARD
# define DISCARD (void)
#endif

```

```

#ifndef sword
# define sword int
#endif

```

```
#define VER7 2
```

```

#define NA -1 /* ANSI SQL NULL */
#define NLT 1 /* length for string null terminator */
#define DEADLOCK 60 /* ORA-00060: deadlock */
#define NO_DATA_FOUND 1403 /* ORA-01403: no data found */
#define NOT_SERIALIZABLE 8177 /* ORA-08177: transaction not serializable */
#define SNAPSHOT_TOO_OLD 1555 /* ORA-01555: snapshot too old */

```

```

#ifndef NULLP
# define NULLP (void *)NULL

```

```
#endif /* NULLP */
```

```

#define ADR(object) ((ub1 *) &(object))
#define SIZ(object) ((sword) sizeof(object))

```

```

typedef char date[24+NLT];
typedef char varchar2;

```

```

#define OCIERROR(errp,function)\
ocierror(__FILE__,__LINE__,(errp),(function));

```

```

#define OCIBND(stmp, bndp, errp, sqlvar, progvl, progvl, ftype)\
ocierror(__FILE__,__LINE__,(errp), \

```

```

OCIHandleAlloc((stmp),(dvoid **)&(bndp),OCI_HTYPE_BIND,0,(dvoid **)0); \
ocierror(__FILE__,__LINE__,(errp), \
OCIBindByName((stmp), &(bndp), (errp), \
(text *) (sqlvar), strlen((sqlvar)), \
(progvl), (progvl), (ftype),0,0,0,0,OCI_DEFAULT));

```

```

#define OCIBNDRA(stmp,bndp,errp,sqlvar,progvl,progvl,ftype,indp,alen,arcode) \
ocierror(__FILE__,__LINE__,(errp), \

```

```

OCIHandleAlloc((stmp),(dvoid **)&(bndp),OCI_HTYPE_BIND,0,(dvoid **)0); \
ocierror(__FILE__,__LINE__,(errp), \
OCIBindByName((stmp),&(bndp),(errp),(text *) (sqlvar),strlen((sqlvar)), \
(progvl),(progvl),(ftype),(indp),(alen),(arcode),0,0,OCI_DEFAULT));

```

```

#define OCIBNDRAD(stmp,bndp,errp,sqlvar,progvl,ftype,indp,ctxp,cbf_nodata,cbf_data) \
ocierror(__FILE__,__LINE__,(errp), \

```

```

OCIHandleAlloc((stmp),(dvoid **)&(bndp),OCI_HTYPE_BIND,0,(dvoid **)0); \
ocierror(__FILE__,__LINE__,(errp), \
OCIBindByName((stmp),&(bndp),(errp),(text *) (sqlvar), \
strlen((sqlvar)),0,(progvl),(ftype), \
indp,0,0,0,OCI_DATA_AT_EXEC)); \
ocierror(__FILE__,__LINE__,(errp), \
OCIBindDynamic((bndp),(errp),(ctxp),(cbf_nodata),(ctxp),(cbf_data));

```

```

#define OCIBNDR(stmp,bndp,errp,sqlvar,progvl,progvl,ftype,indp,alen,arcode) \
ocierror(__FILE__,__LINE__,(errp), \

```

```

OCIHandleAlloc((stmp),(dvoid **)&(bndp),OCI_HTYPE_BIND,0,(dvoid **)0); \
ocierror(__FILE__,__LINE__,(errp), \
OCIBindByName((stmp),&(bndp),(errp),(text *) (sqlvar),strlen((sqlvar)), \
(progvl),(progvl),(ftype),(indp),(alen),(arcode),0,0,OCI_DEFAULT));

```

```

#define OCIBNDRAA(stmp,bndp,errp,sqlvar,progvl,progvl,ftype,indp,alen,arcode,ms,cu) \
ocierror(__FILE__,__LINE__,(errp), \
OCIHandleAlloc((stmp),&(bndp),OCI_HTYPE_BIND,0,(dvoid **)0); \
ocierror(__FILE__,__LINE__,(errp), \
OCIBindByName((stmp),&(bndp),(errp),(text *) (sqlvar),strlen((sqlvar)), \
(progvl),(progvl),(ftype),(indp),(alen),(arcode),(ms),(cu),OCI_DEFAULT));

```

```

#define OCIDEFINE(stmp,dfnp,errp,pos,progvl,progvl,ftype)\
OCIDefineByPos((stmp),&(dfnp),(errp),(pos),(progvl),(progvl),(ftype), \
0,0,0,OCI_DEFAULT);

```

```

#define OCIDEF(stmp,dfnp,errp,pos,progvl,progvl,ftype) \
OCIHandleAlloc((stmp),(dvoid **)&(dfnp),OCI_HTYPE_DEFINE,0, \
(dvoid **)0); \
OCIDefineByPos((stmp),&(dfnp),(errp),(pos),(progvl),(progvl), \
(ftype),NULL,NULL,NULL,OCI_DEFAULT); \

```

```

#define OCIDFNRA(stmp,dfnp,errp,pos,progvl,progvl,ftype,indp,alen,arcode) \
OCIHandleAlloc((stmp),(dvoid **)&(dfnp),OCI_HTYPE_DEFINE,0, \
(dvoid **)0); \
OCIDefineByPos((stmp),&(dfnp),(errp),(pos),(progvl), \

```

```

(progvl),(ftype),(indp),(alen), \
(arcode),OCI_DEFAULT); \

```

```

/* OCIDefineArrayOfStruct((dfnp),(errp),(progvl), \

```

```
sizeof((indp)[0]), \

```

```
sizeof((alen)[0]), \

```

```
sizeof((arcode)[0])); \

```

```
*/
```

```

/* OCIDFNRA(stmp,dfnp,errp,pos,progvl,progvl,ftype,indp,alen,arcode) \
ocierror(__FILE__,__LINE__,(errp), \

```

```

OCIHandleAlloc(tpcenv,&(dfnp),OCI_HTYPE_DEFINE,0, \
(dvoid **)0); \
ocierror(__FILE__,__LINE__,(errp), \
OCIDefineByPos((stmp),&(dfnp),(errp),(pos),(progvl), \

```

```

(progvl),(ftype),(indp),(alen), \

```

```

(arcode),OCI_DEFAULT));\
ocierror(_FILE_,_LINE_,(errp),\
        OCIDefineArrayOfStruct((dfnp),(errp),(progv1),\

sizeof((indp)[0]),\

sizeof((alen)[0]),\

sizeof((arcode)[0]));\
*/

#define OBNDRV(lda,cursor,sqlvar,progv,progv1,ftype)\
if (obndrv((cursor),(text*)(sqlvar),NA,(ub1*)(progv),(progv1),(ftype),NA,\
          (sb2 *)0, (text *)0, NA, NA))\
    {errprt(lda,cursor);return(-1);}\
else\
    DISCARD 0

#define OBNDRA(lda,cursor,sqlvar,progv,progv1,ftype,indp,alen,arcode)\
if (obndra((cursor),(text*)(sqlvar),NA,(ub1*)(progv),(progv1),(ftype),NA,\
          (indp),(alen),(arcode),(ub4)0,(ub4*)0,(text*)0,NA,NA))\
    {errprt(lda,cursor);return(-1);}\
else\
    DISCARD 0

#define OBNDRAA(lda,cursor,sqlvar,progv,progv1,ftype,indp,alen,arcode,ms,cs)\
if (obndraa((cursor),(text*)(sqlvar),NA,(ub1*)(progv),(progv1),(ftype),NA,\
          (indp),(alen),(arcode),(ub4)(ms),(ub4*)(cs),(text*)0,NA,NA))\
    {errprt(lda,cursor);return(-1);}\
else\
    DISCARD 0

#define ODEFIN(lda,cursor,pos,buf,buf1,ftype,scal,indp,fmt,fmtl,fmtr,rlen,rcode)\
if (odefint((cursor),(pos),(ub1*)(buf),(buf1),(ftype),(scal),(indp),\
          (text*)(fmt),(fmtl),(fmtr),(rlen),(rcode))\
    {errprt(lda,cursor);return(-1);}\
else\
    DISCARD 0

#define OEXFET(lda,cursor,nrows,cancel,exact)\
if (oexfet((cursor),(nrows),(cancel),(exact))\
    {if ((cursor)->rc == 1403)\
        {i=errprt(lda,cursor); orol(lda); return(-1);}\
        else if (errprt(lda,cursor)==RECOVERR)\
            {orol(lda);return(RECOVERR);}\
        else {orol(lda);return(-1);}\
    }

else\
    DISCARD 0

#define OOPEN(lda,cursor)\
if (oopen((cursor),(lda),(text*)0,NA,NA,(text*)0,NA))\
    {errprt(lda,cursor);return(-1);}\
else\
    DISCARD 0

#define OPARSE(lda,cursor,sqlstm,sql,defflg,lngflg)\
if (oparse((cursor),(sqlstm),(sb4)(sql),(defflg),(ub4)(lngflg))\
    {errprt(lda,cursor);return(-1);}\
else\
    DISCARD 0

#define OFEN(lda,cursor,nrows)\
if (ofen((cursor),(nrows))\
    {if (errprt(lda,cursor)==RECOVERR)\
        {orol(lda);return(RECOVERR);}\
        else {orol(lda);return(-1);}\
    }

else\
    DISCARD 0

#define OEXEC(lda,cursor)\
if (oexec((cursor))\
    {if (errprt(lda,cursor)==RECOVERR)\
        {orol(lda);return(RECOVERR);}\
        else {orol(lda);return(-1);}\
    }

else\
    DISCARD 0

#define OCOM(lda,cursor)\
if (ocom((lda))\
    {errprt(lda,cursor);orol(lda);return(-1);}\
else\
    DISCARD 0

#define OEXN(lda,cursor,itors,rowoff)\
if (oexn((cursor),(itors),(rowoff))\
    {if (errprt(lda,cursor)==RECOVERR)\
        {orol(lda);return(RECOVERR);}\
        else {orol(lda);return(-1);}\
    }

else\
    DISCARD 0

#endif

```

dbwr_on_int_cpus_32loc.sh

```

#!/usr/bin/ksh

function usage {
    echo "Usage: $0 [-h] [-b]"
    echo "t-h: Turn hyperthreading on for the dbwriters pset"
    echo "t-b: Bind the dbwriters to disk interrupt CPUs, presumably CPU 7s"
}

LCPU=OFF
BIND=0
while getopts hb name
do
    case $name in
        h) LCPU=ON
           ;;
        b) BIND=1
           ;;
        ?) usage
           exit 1
           ;;
        esac
    done

# Hard-coded assumption: the disk interrupt CPU is the last core in the cell
INT_CPU=mpsched -s|tail -1|awk '{print $NF}'^
INT_PSET=psrset -p ${INT_CPU}|head -1|awk '{print $NF}'^
psrset -t ${INT_PSET} LCPU=${LCPU}

LOC=0
SECOND_CPU=0
for PID in `ps -ef|grep "ora_dbw"|grep -v grep | awk '{print $2}'|sort -n`
do
    if test $BIND -eq 1; then
        LINE=expr 16 - $LOC
        if test $SECOND_CPU -eq 1; then
            INT_CPU=mpsched -s|tail -${LINE}|head -1|awk '{print $9}'^
        else
            INT_CPU=mpsched -s|tail -${LINE}|head -1|awk '{print $5}'^
        fi
        INT_PSET=`psrset -p ${INT_CPU}|awk '{print $(NF)}'^
        psrset -b ${INT_PSET} ${PID}
        mpsched -c ${INT_CPU} -p $PID
    else
        mpsched -l $LOC -p $PID
    fi
    # psrset -q $PID
    # mpsched -q -p $PID

    LOC=`expr $LOC + 1`
    if test $LOC -eq 16; then
        LOC=0
        SECOND_CPU=1
    fi
done

if test $LOC -ne 0; then
    echo "Number of dbwriters was not a multiple of 16"
fi

intctl_32loc.sh

#!/usr/bin/ksh

function usage {
    echo "Usage: $0 [-0] [-2] [-1] [-d] [-h] [-s] [-3] [-n]"
    echo "t-0: Start by returning all CPUs to pset 0"
    echo "t-2: Use two whole cores for interrupts"
    echo "t-1: Put interrupts on the last core of each cell, presumably CPU 7"
    echo "t-d: The disk interrupts CPUs are in their own pset"
    echo "t-h: Turn hyperthreading on for the interrupts pset"
    echo "t-p: Create two psets, one on each front-side bus"
    echo "t-s: Separate CPU for networking interrupts, presumably CPU 6"
    echo "t-3: Use CPU 3 of each cell CPU for networking interrupts"
    echo "t-n: The networking interrupt CPUs are in their own pset"
}

LCPU=OFF
NET_INT_PSET=0
DISK_INT_PSET=0
SEPARATE_NET_INT_CPUS=0
TWO_CORES=0
LAST_CPU=0

```

```

NET_INT_CPU_FIELD=8
TWO_PSETS=0
while getopts dhnlps023 name
do
    case $name in
        0) prsrt -d all
            ;;
        2) TWO_CORES=1
            ;;
        p) TWO_PSETS=1
            ;;
        s) SEPARATE_NET_INT_CPUS=1
            ;;
        l) LAST_CPU=1
            ;;
        h) LCPU=ON
            ;;
        d) DISK_INT_PSET=1
            ;;
        n) NET_INT_PSET=1
            ;;
        3) NET_INT_CPU_FIELD=5
            ;;
        ?) usage
            exit 1
            ;;
        esac
done

function loop_intctl {
    TRIES=0
    while test $TRIES -le 100
    do
        intctl -w -M -H $1 -I $2 -c $3
        if test $? -eq 0; then
            break
        fi
        TRIES=`expr $TRIES + 1`
    done
}

CPUS_PER_CELL=`mpsched -s|tail -1|wc|awk '{print $2 - 1}'`
CELL=0
ALL_INT_CPUS=""
while test $CELL -le 15; do
    LINE=`expr $CELL + 1`
    if test $LAST_CPU -eq 1; then
        INT_CPU=`mpsched -s|tail -16|head -${LINE}|tail -1|awk '{print $9}'`
    else
        INT_CPU=`mpsched -s|tail -16|head -${LINE}|tail -1|awk '{print $2}'`
    fi
    ALL_INT_CPUS="$ALL_INT_CPUS ${INT_CPU}"
    loop_intctl ${CELL}/0/4/1/0 1 ${INT_CPU}
    loop_intctl ${CELL}/0/6/1/0 1 ${INT_CPU}
    if test $LAST_CPU -eq 1 -a $TWO_CORES -eq 1; then
        INT_CPU=`mpsched -s|tail -16|head -${LINE}|tail -1|awk '{print $5}'`
        ALL_INT_CPUS="$ALL_INT_CPUS ${INT_CPU}"
    fi
    loop_intctl ${CELL}/0/5/1/0 1 ${INT_CPU}
    loop_intctl ${CELL}/0/6/1/1 1 ${INT_CPU}
    CELL=`expr $CELL + 1`
done

if test ${DISK_INT_PSET} -eq 1; then
    INT_PSET=`prsrset -c ${ALL_INT_CPUS}|head -1|awk '{print $NF}'`
    prsrset -t ${INT_PSET} LCPU=${LCPU}
fi

# Move the interrupts of the log devices to the logwriter CPU
INT_CPU=`mpsched -s|tail -15|head -1|awk '{print $2}'`
INT_CPU_SIBLING=`tools/bin/print_spu_mapping | sed -e "s/:/ /" | awk -v
int_cpu=${INT_CPU} '{if ($2==int_cpu) {sibling_next=1}}/My sibling is/{if
(sibling_next=1){sibling_next=0; print $NF; exit}}`
LGWR_CPU=`mpsched -s|tail -15|head -1|awk -v int_cpu=${INT_CPU} -v
int_cpu_sibling=${INT_CPU_SIBLING} '{for (i=2; i<=NF; i++) {if ($i) != int_cpu &&
$(i) !=int_cpu_sibling) {print $(i); exit}} print "Did not find a proper SPU!!!"`

for LBA in 0 1 8 9 10 12 13 14
do
    loop_intctl 1/0/${LBA}/1/0 1 ${LGWR_CPU}
    loop_intctl 1/0/${LBA}/1/1 1 ${LGWR_CPU}
done

if test `prsrset -p ${LGWR_CPU} | awk '{print $NF}'` -eq 0; then
    prsrset -c ${LGWR_CPU}
    LGWR_PSET=`prsrset -p ${LGWR_CPU} | awk '{print $NF}'`
    prsrset -t ${LGWR_PSET} LCPU=OFF
fi

PORT=1522
for CELL in 2 6 10 14; do
    PPA=`lanscan|egrep "AS${CELL}/0/2/1/0"|sed -e "s/*lan/*" -e "s/*/*/"`
    lanadmin -X drv_dp_on $PPA
    lanadmin -X drv_pr_on $PPA
    lanadmin -X drv_mq 8 $PPA

```

```

lanadmin -X dps_tm_off $PPA
for Q in `lanadmin -x card_info $PPA | grep "Queue #" | awk '{print $3}'`; do
    INT_ID=`lanadmin -x card_info $PPA | grep "Queue # $Q" | awk '{print $4}'`
    INT_CELL=`expr \( ${CELL} \) \ / 4 \) * 4 + {Q} \ / 2`
    LINE=`expr $INT_CELL + 1`
    if test ${SEPARATE_NET_INT_CPUS} -eq 1; then
        INT_CPU=`mpsched -s|tail -16|head -${LINE}|tail -1|awk -v Q=${Q} -v
NET_INT_CPU_FIELD=${NET_INT_CPU_FIELD} '{if (int(Q/2)*2==Q)print
$(NET_INT_CPU_FIELD - 4); else print $(NET_INT_CPU_FIELD)}'`
        if test $LAST_CPU -eq 1; then
            INT_CPU=`mpsched -s|tail -16|head -${LINE}|tail -1|awk -v
Q=${Q} '{if (int(Q/2)*2==Q)print $4; else print $8}'`
        else
            INT_CPU=`mpsched -s|tail -16|head -${LINE}|tail -1|awk -v
Q=${Q} '{if (int(Q/2)*2==Q)print $1; else print $5}'`
        fi
        # LAST_CPU=1 and TWO_CORES=1 are hard-coded assumptions in the else case
        INT_CPU=`mpsched -s|tail -16|head -${LINE}|tail -1|awk -v Q=${Q} -v
NET_INT_CPU_FIELD=9 '{if (int(Q/2)*2==Q)print $(NET_INT_CPU_FIELD - 4); else
print $(NET_INT_CPU_FIELD)}'`
        if test $LAST_CPU -eq 1; then
            INT_CPU=`mpsched -s|tail -16|head -${LINE}|tail -1|awk -v
Q=${Q} '{if (int(Q/2)*2==Q)print $5; else print $9}'`
        else
            INT_CPU=`mpsched -s|tail -16|head -${LINE}|tail -1|awk -v
Q=${Q} '{if (int(Q/2)*2==Q)print $2; else print $6}'`
        fi
    fi
    INDEX=""
    INDEX=`lanadmin -x dps_pqm $PPA | grep ${PORT} | awk '{print $1}'`
    if test ${INDEX}XXX != XXX; then
        lanadmin -X dp_del_pqm $INDEX $PPA
    fi
    lanadmin -X dp_add_pqm ${PORT} ${Q} ${PPA}
    loop_intctl ${CELL}/0/2/1/0 ${INT_ID} ${INT_CPU}
    ALL_INT_CPUS="$ALL_INT_CPUS ${INT_CPU}"
    PORT=`expr $PORT + 1`
done
lanadmin -x card_info $PPA
lanadmin -x dps_map $PPA
done
if test ${SEPARATE_NET_INT_CPUS} -eq 1 -a ${NET_INT_PSET} -eq 1; then
    INT_PSET=`prsrset -c ${ALL_INT_CPUS}|head -1|awk '{print $NF}'`
    prsrset -t ${INT_PSET} LCPU=${LCPU}
fi

if test ${TWO_PSETS} -eq 1; then
    ALL_INT_CPUS1=""
    ALL_INT_CPUS2=""
for SPU in `tools/bin/print_spu_mapping | grep "[^1][0X][01],|sed -e "s/:/ /" -e "s/*/*/"`
do
    if test ${SPU} -eq ${LGWR_CPU}; then
        continue
    fi
    ALL_INT_CPUS1="$ALL_INT_CPUS1 $SPU"
done
for SPU in `tools/bin/print_spu_mapping | grep "1[0X][01],|sed -e "s/:/ /" -e "s/*/*/"`
do
    if test ${SPU} -eq ${LGWR_CPU}; then
        continue
    fi
    ALL_INT_CPUS2="$ALL_INT_CPUS2 $SPU"
done
# Assume SPU 0 is in ALL_INT_CPUS1, so we have to use "-a 0" instead of "-c"
INT_PSET=`prsrset -a 0 ${ALL_INT_CPUS1}|head -1|awk '{print $NF}'`
prsrset -t ${INT_PSET} LCPU=${LCPU}
INT_PSET=`prsrset -c ${ALL_INT_CPUS2}|head -1|awk '{print $NF}'`
prsrset -t ${INT_PSET} LCPU=${LCPU}
fi

intctl

#!/usr/bin/ksh
if test $# -gt 1; then
    echo "Usage: `basename $0` [config]"
    exit 1
fi

if test $# -eq 1; then
    kconfig $1 > /dev/null || (echo ""`basename $0`: $1 is not a valid configuration"; exit 1)
    CONFIG=$1
else
    CONFIG="current"
fi

```

kctune.sh

```

#!/usr/bin/ksh
if test $# -gt 1; then
    echo "Usage: `basename $0` [config]"
    exit 1
fi

if test $# -eq 1; then
    kconfig $1 > /dev/null || (echo ""`basename $0`: $1 is not a valid configuration"; exit 1)
    CONFIG=$1
else
    CONFIG="current"
fi

```

```

kctune -b yes -c ${CONFIG} lcpu_attr=1
kctune -b yes -c ${CONFIG} nproc=6700
kctune -b yes -c ${CONFIG} nkthread=20000
kctune -b yes -c ${CONFIG} max_thread_proc=2048
kctune -b yes -c ${CONFIG} maxuprc=6000
kctune -b yes -c ${CONFIG} maxfiles_lim=8192
kctune -b yes -c ${CONFIG} maxfiles=8192
kctune -b yes -c ${CONFIG} process_id_max=99999
kctune -b yes -c ${CONFIG} process_id_min=0
kctune -b yes -c ${CONFIG} filecache_min=640MB filecache_max=640MB
kctune -b yes -c ${CONFIG} default_disk_ir=1
kctune -b yes -c ${CONFIG} vps_chattr_ceiling=4194304
kctune -b yes -c ${CONFIG} nfile=10000000
kctune -b yes -c ${CONFIG} lockable_mem_pct=99%
kctune -b yes -c ${CONFIG} shmmax=0x20000000000
# kctune -b yes -c ${CONFIG} msgctl=1024

```

isnr.sh

```

#!/usr/bin/ksh
QUEUE=0
while [ ${QUEUE} -lt 32 ]; do
  CELL=`expr ${QUEUE} \ / 2 `
  if test $-eq 1 && test $1 = "stop"; then
    Isnrcrl stop listener${QUEUE}
  else
    mpsched -l $CELL -P PACKED Isnrcrl start listener${QUEUE}
    if test `expr ${QUEUE} \ / 2 \ * 2 ` -eq $QUEUE; then
      PSET=`psrset -p 0|awk '{print $(NF)}'`
    else
      PSET=`psrset -p 4|awk '{print $(NF)}'`
    fi
    PID=`ps -ef|grep listener${QUEUE}|grep -v grep|awk '{print $2}'`
    psrset -b ${PSET} ${PID}
  fi
  QUEUE=`expr ${QUEUE} + 1 `
done

```

set_schedna

```

#!/bin/ksh -x
#
ps -ef > ps.$$

if [ `uname` = "HP-UX" ]
then
for PID in `grep "ora_" ps.$$ | awk '{print $2}'`
do
  rtsched -s SCHED_NOAGE -p 180 -P $PID
  mpsched -q -p $PID
done

for PID in `grep "ora_dbw" ps.$$ | awk '{print $2}'`
do
  rtsched -s SCHED_NOAGE -p 180 -P $PID
  mpsched -q -p $PID
done

for PID in `grep "ora_lg" ps.$$ | awk '{print $2}'`
do
  rtsched -s SCHED_NOAGE -p 178 -P $PID

  INT_CPU=`mpsched -s|tail -15|head -1|awk '{print $2}'`
  INT_CPU_SIBLING=`print_spu_mapping |sed -e "s/:/ /" |awk -v int_cpu=${INT_CPU}
  '{if ($2==int_cpu) {sibling_next=1} /My sibling is/ {if (sibling_next==1){sibling_next=0;
  print $NF; exit}}'`
  LGWR_CPU=`mpsched -s|tail -15|head -1|awk -v int_cpu=${INT_CPU} -v
  int_cpu_sibling=${INT_CPU_SIBLING} '{for (i=2; i<=NF; i++) {if ($i) != int_cpu &&
  $(i) !=int_cpu_sibling) {print $(i); exit} } print "Did not find a proper SPU!!!"}'`
  if test `psrset -p ${LGWR_CPU} | awk '{print $NF}'` -eq 0; then
    psrset -c ${LGWR_CPU}
    LGWR_PSET=`psrset -p ${LGWR_CPU} | awk '{print $NF}'`
    psrset -t ${LGWR_PSET} LCPU=OFF
  fi

  LGWR_PSET=`psrset -p ${LGWR_CPU} | awk '{print $NF}'`
  psrset -b ${LGWR_PSET} ${PID}
  mpsched -c ${LGWR_CPU} -p $PID
  mpsched -q -p $PID
done

for PID in `egrep 'oraletpc|tpcc.*client|tpcc.*service|lsnr|BBL' ps.$$ | awk '{print $2}'`
do
  rtsched -s SCHED_NOAGE -p 180 -P $PID
  mpsched -q -p $PID

```

```

done
fi
mv ps.$$/tmp/TPCC_STATS.PS

```

A.4 Server Stored Procedures

tkvcpdel.sql

```

declare
TYPE numarray IS TABLE OF NUMBER INDEX BY BINARY_INTEGER;
TYPE numlist is varray (10) of number;
dist numarray;
amt numarray;
cnt pls_integer;

not_serializable EXCEPTION;
PRAGMA EXCEPTION_INIT(not_serializable, -8177);
deadlock EXCEPTION;
PRAGMA EXCEPTION_INIT(deadlock, -60);
snapshot_too_old EXCEPTION;
PRAGMA EXCEPTION_INIT(snapshot_too_old, -1555);

BEGIN
LOOP BEGIN
FORALL d IN 1..10
DELETE FROM nord N
WHERE no_d_id = inittpc.dist(d)
AND no_w_id = :w_id
AND no_o_id = (select min (no_o_id)
from nord
where no_d_id = N.no_d_id
and no_w_id = N.no_w_id)
RETURNING no_d_id, no_o_id BULK COLLECT INTO :d_id, :order_id;

:ordcnt := SQL%ROWCOUNT;

```

```

FORALL o in 1.. :ordcnt
UPDATE ordr SET o_carrier_id = :carrier_id
WHERE o_id = :order_id (o)
AND o_d_id = :d_id(o)
AND o_w_id = :w_id
RETURNING o_c_id BULK COLLECT INTO :o_c_id;

```

```

FORALL o in 1.. :ordcnt
UPDATE ordl SET ol_delivery_d = :now
WHERE ol_w_id = :w_id
AND ol_d_id = :d_id(o)
AND ol_o_id = :order_id(o)
RETURNING sum(ol_amount) BULK COLLECT INTO :sums;

```

```

FORALL c IN 1.. :ordcnt
UPDATE cust
SET c_balance = c_balance + :sums(c) ,
c_delivery_cnt = c_delivery_cnt + 1
WHERE c_w_id = :w_id
AND c_d_id = :d_id(c)
AND c_id = :o_c_id(c);

```

```

COMMIT;
EXIT;
EXCEPTION
WHEN not_serializable OR deadlock OR snapshot_too_old
THEN
ROLLBACK;
:retry := :retry + 1;
END;

```

```

END LOOP; -- for retry
END;

```

tkvcpnew.sql

-- New Order Anonymous block

```

DECLARE
idx          PLS_INTEGER;
dummy_local  PLS_INTEGER;
cache_ol_cnt PLS_INTEGER;
not_serializable EXCEPTION;
PRAGMA EXCEPTION_INIT(not_serializable,-8177);
deadlock     EXCEPTION;

```

```

PRAGMA EXCEPTION_INIT(deadlock,-60);
snapshot_too_old EXCEPTION;
PRAGMA EXCEPTION_INIT(snapshot_too_old,-1555);

PROCEDURE u1 IS
BEGIN
  FORALL idx IN 1 .. cache_ol_cnt
    UPDATE stock_item
      SET s_order_cnt = s_order_cnt + 1,
          s_ytd = s_ytd + :ol_quantity(idx),
          s_remote_cnt = s_remote_cnt + :s_remote(idx),
          s_quantity = (CASE WHEN s_quantity < :ol_quantity (idx) + 10
                          THEN s_quantity +91
                          ELSE s_quantity
                          END) - :ol_quantity(idx)
    WHERE i_id = :ol_i_id(idx)
    AND s_w_id = :ol_supply_w_id(idx)
    RETURNING i_price, i_name, s_quantity, s_dist_01,
              i_price*ol_quantity(idx),
              CASE WHEN i_data NOT LIKE '%ORIGINAL%'
                THEN 'G'
                ELSE (CASE WHEN s_data NOT LIKE '%ORIGINAL%'
                          THEN 'G'
                          ELSE 'B'
                          END)
                END)
    END
    BULK COLLECT INTO :i_price, :i_name, :s_quantity, inittpc.s_dist,
                     :ol_amount,:brand_generic;
END u1;

PROCEDURE u2 IS
BEGIN
  FORALL idx IN 1 .. cache_ol_cnt
    UPDATE stock_item
      SET s_order_cnt = s_order_cnt + 1,
          s_ytd = s_ytd + :ol_quantity(idx),
          s_remote_cnt = s_remote_cnt + :s_remote(idx),
          s_quantity = (CASE WHEN s_quantity < :ol_quantity (idx) + 10
                          THEN s_quantity +91
                          ELSE s_quantity
                          END) - :ol_quantity(idx)
    WHERE i_id = :ol_i_id(idx)
    AND s_w_id = :ol_supply_w_id(idx)
    RETURNING i_price, i_name, s_quantity, s_dist_02,
              i_price*ol_quantity(idx),
              CASE WHEN i_data NOT LIKE '%ORIGINAL%'
                THEN 'G'
                ELSE (CASE WHEN s_data NOT LIKE '%ORIGINAL%'
                          THEN 'G'
                          ELSE 'B'
                          END)
                END)
    END
    BULK COLLECT INTO :i_price, :i_name, :s_quantity, inittpc.s_dist,
                     :ol_amount,:brand_generic;
END u2;

PROCEDURE u3 IS
BEGIN
  FORALL idx IN 1 .. cache_ol_cnt
    UPDATE stock_item
      SET s_order_cnt = s_order_cnt + 1,
          s_ytd = s_ytd + :ol_quantity(idx),
          s_remote_cnt = s_remote_cnt + :s_remote(idx),
          s_quantity = (CASE WHEN s_quantity < :ol_quantity (idx) + 10
                          THEN s_quantity +91
                          ELSE s_quantity
                          END) - :ol_quantity(idx)
    WHERE i_id = :ol_i_id(idx)
    AND s_w_id = :ol_supply_w_id(idx)
    RETURNING i_price, i_name, s_quantity, s_dist_03,
              i_price*ol_quantity(idx),
              CASE WHEN i_data NOT LIKE '%ORIGINAL%'
                THEN 'G'
                ELSE (CASE WHEN s_data NOT LIKE '%ORIGINAL%'
                          THEN 'G'
                          ELSE 'B'
                          END)
                END)
    END
    BULK COLLECT INTO :i_price, :i_name, :s_quantity, inittpc.s_dist,
                     :ol_amount,:brand_generic;
END u3;

PROCEDURE u4 IS
BEGIN
  FORALL idx IN 1 .. cache_ol_cnt
    UPDATE stock_item
      SET s_order_cnt = s_order_cnt + 1,
          s_ytd = s_ytd + :ol_quantity(idx),
          s_remote_cnt = s_remote_cnt + :s_remote(idx),
          s_quantity = (CASE WHEN s_quantity < :ol_quantity (idx) + 10
                          THEN s_quantity +91
                          ELSE s_quantity
                          END) - :ol_quantity(idx)
    WHERE i_id = :ol_i_id(idx)
    AND s_w_id = :ol_supply_w_id(idx)
    RETURNING i_price, i_name, s_quantity, s_dist_04,
              i_price*ol_quantity(idx),
              CASE WHEN i_data NOT LIKE '%ORIGINAL%'
                THEN 'G'
                ELSE (CASE WHEN s_data NOT LIKE '%ORIGINAL%'
                          THEN 'G'
                          ELSE 'B'
                          END)
                END)
    END
    BULK COLLECT INTO :i_price, :i_name, :s_quantity, inittpc.s_dist,
                     :ol_amount,:brand_generic;
END u4;

PROCEDURE u5 IS
BEGIN
  FORALL idx IN 1 .. cache_ol_cnt
    UPDATE stock_item
      SET s_order_cnt = s_order_cnt + 1,
          s_ytd = s_ytd + :ol_quantity(idx),
          s_remote_cnt = s_remote_cnt + :s_remote(idx),
          s_quantity = (CASE WHEN s_quantity < :ol_quantity (idx) + 10
                          THEN s_quantity +91
                          ELSE s_quantity
                          END) - :ol_quantity(idx)
    WHERE i_id = :ol_i_id(idx)
    AND s_w_id = :ol_supply_w_id(idx)
    RETURNING i_price, i_name, s_quantity, s_dist_05,
              i_price*ol_quantity(idx),
              CASE WHEN i_data NOT LIKE '%ORIGINAL%'
                THEN 'G'
                ELSE (CASE WHEN s_data NOT LIKE '%ORIGINAL%'
                          THEN 'G'
                          ELSE 'B'
                          END)
                END)
    END
    BULK COLLECT INTO :i_price, :i_name, :s_quantity, inittpc.s_dist,
                     :ol_amount,:brand_generic;
END u5;

PROCEDURE u6 IS
BEGIN
  FORALL idx IN 1 .. cache_ol_cnt
    UPDATE stock_item
      SET s_order_cnt = s_order_cnt + 1,
          s_ytd = s_ytd + :ol_quantity(idx),
          s_remote_cnt = s_remote_cnt + :s_remote(idx),
          s_quantity = (CASE WHEN s_quantity < :ol_quantity (idx) + 10
                          THEN s_quantity +91
                          ELSE s_quantity
                          END) - :ol_quantity(idx)
    WHERE i_id = :ol_i_id(idx)
    AND s_w_id = :ol_supply_w_id(idx)
    RETURNING i_price, i_name, s_quantity, s_dist_06,
              i_price*ol_quantity(idx),
              CASE WHEN i_data NOT LIKE '%ORIGINAL%'
                THEN 'G'
                ELSE (CASE WHEN s_data NOT LIKE '%ORIGINAL%'
                          THEN 'G'
                          ELSE 'B'
                          END)
                END)
    END
    BULK COLLECT INTO :i_price, :i_name, :s_quantity, inittpc.s_dist,
                     :ol_amount,:brand_generic;
END u6;

PROCEDURE u7 IS
BEGIN
  FORALL idx IN 1 .. cache_ol_cnt
    UPDATE stock_item
      SET s_order_cnt = s_order_cnt + 1,
          s_ytd = s_ytd + :ol_quantity(idx),
          s_remote_cnt = s_remote_cnt + :s_remote(idx),
          s_quantity = (CASE WHEN s_quantity < :ol_quantity (idx) + 10
                          THEN s_quantity +91
                          ELSE s_quantity
                          END) - :ol_quantity(idx)
    WHERE i_id = :ol_i_id(idx)
    AND s_w_id = :ol_supply_w_id(idx)
    RETURNING i_price, i_name, s_quantity, s_dist_07,
              i_price*ol_quantity(idx),
              CASE WHEN i_data NOT LIKE '%ORIGINAL%'
                THEN 'G'
                ELSE (CASE WHEN s_data NOT LIKE '%ORIGINAL%'
                          THEN 'G'
                          ELSE 'B'
                          END)
                END)
    END
    BULK COLLECT INTO :i_price, :i_name, :s_quantity, inittpc.s_dist,
                     :ol_amount,:brand_generic;
END u7;

PROCEDURE u8 IS
BEGIN
  FORALL idx IN 1 .. cache_ol_cnt
    UPDATE stock_item

```



```

SET s_order_cnt = s_order_cnt + 1,
s_ytd = s_ytd + :ol_quantity(idx),
s_remote_cnt = s_remote_cnt + :s_remote(idx),
s_quantity = (CASE WHEN s_quantity < :ol_quantity (idx) + 10
THEN s_quantity +91
ELSE s_quantity
END) - :ol_quantity(idx)
WHERE i_id = :ol_i_id(idx)
AND s_w_id = :ol_supply_w_id(idx)
RETURNING i_price, i_name, s_quantity, s_dist_08,
i_price*:ol_quantity(idx),
CASE WHEN i_data NOT LIKE '%ORIGINAL%'
THEN 'G'
ELSE (CASE WHEN s_data NOT LIKE '%ORIGINAL%'
THEN 'G'
ELSE 'B'
END)
END
BULK COLLECT INTO :i_price, :i_name, :s_quantity, inittpc.s_dist,
:ol_amount,:brand_generic;
END u8;

```

```

PROCEDURE u9 IS
BEGIN
FORALL idx IN 1 .. cache_ol_cnt
UPDATE stock_item
SET s_order_cnt = s_order_cnt + 1,
s_ytd = s_ytd + :ol_quantity(idx),
s_remote_cnt = s_remote_cnt + :s_remote(idx),
s_quantity = (CASE WHEN s_quantity < :ol_quantity (idx) + 10
THEN s_quantity +91
ELSE s_quantity
END) - :ol_quantity(idx)
WHERE i_id = :ol_i_id(idx)
AND s_w_id = :ol_supply_w_id(idx)
RETURNING i_price, i_name, s_quantity, s_dist_09,
i_price*:ol_quantity(idx),
CASE WHEN i_data NOT LIKE '%ORIGINAL%'
THEN 'G'
ELSE (CASE WHEN s_data NOT LIKE '%ORIGINAL%'
THEN 'G'
ELSE 'B'
END)
END
BULK COLLECT INTO :i_price, :i_name, :s_quantity, inittpc.s_dist,
:ol_amount,:brand_generic;
END u9;

```

```

PROCEDURE u10 IS
BEGIN
FORALL idx IN 1 .. cache_ol_cnt
UPDATE stock_item
SET s_order_cnt = s_order_cnt + 1,
s_ytd = s_ytd + :ol_quantity(idx),
s_remote_cnt = s_remote_cnt + :s_remote(idx),
s_quantity = (CASE WHEN s_quantity < :ol_quantity (idx) + 10
THEN s_quantity +91
ELSE s_quantity
END) - :ol_quantity(idx)
WHERE i_id = :ol_i_id(idx)
AND s_w_id = :ol_supply_w_id(idx)
RETURNING i_price, i_name, s_quantity, s_dist_10,
i_price*:ol_quantity(idx),
CASE WHEN i_data NOT LIKE '%ORIGINAL%'
THEN 'G'
ELSE (CASE WHEN s_data NOT LIKE '%ORIGINAL%'
THEN 'G'
ELSE 'B'
END)
END
BULK COLLECT INTO :i_price, :i_name, :s_quantity, inittpc.s_dist,
:ol_amount,:brand_generic;
END u10;

```

```

PROCEDURE fix_items IS
rows_lost PLS_INTEGER;
max_index PLS_INTEGER;
temp_index PLS_INTEGER;
BEGIN
idx := 1;
rows_lost := 0;
max_index := dummy_local;

WHILE (max_index != cache_ol_cnt) LOOP

WHILE (idx <= sql%rowcount AND
sql%bulk_rowcount(idx + rows_lost) = 1)
LOOP
idx := idx + 1;
END LOOP;

temp_index := max_index;
WHILE (temp_index >= idx + rows_lost) LOOP
:ol_amount(temp_index + 1) := :ol_amount(temp_index);
:i_price(temp_index + 1) := :i_price(temp_index);

```

```

:i_name(temp_index + 1) := :i_name(temp_index);
:s_quantity(temp_index + 1) := :s_quantity(temp_index);
inittpc.s_dist(temp_index + 1) := inittpc.s_dist(temp_index);
:brand_generic(temp_index + 1) := :brand_generic(temp_index);
temp_index := temp_index - 1;
END LOOP;

```

```

IF (idx + rows_lost <= cache_ol_cnt) THEN
:i_price(idx + rows_lost) := 0;
:i_name(idx + rows_lost) := 'NO ITEM';
:s_quantity(idx + rows_lost) := 0;
inittpc.s_dist(idx + rows_lost) := NULL;
:brand_generic(idx + rows_lost) := '';
:ol_amount(idx + rows_lost) := 0;
rows_lost := rows_lost + 1;
max_index := max_index + 1;
END IF;

```

```

END LOOP;
END fix_items;

BEGIN
LOOP BEGIN
cache_ol_cnt := :ol_cnt;

```

```

UPDATE dist SET d_next_o_id = d_next_o_id + 1
WHERE d_id = :d_id AND d_w_id = :w_id
RETURNING d_tax, d_next_o_id-1
INTO :d_tax, :o_id;

```

```

SELECT c_discount, c_last, c_credit
INTO :c_discount, :c_last, :c_credit
FROM cust
WHERE c_id=:c_id AND c_d_id = :d_id AND c_w_id = :w_id;

```

```

SELECT w_tax
INTO :w_tax
FROM ware
WHERE w_id = :w_id;

```

```

INSERT INTO nord (no_o_id, no_d_id, no_w_id)
VALUES (:o_id, :d_id, :w_id);

```

```

INSERT INTO ordr (o_id, o_d_id, o_w_id, o_c_id, o_entry_d,
o_carrier_id, o_ol_cnt, o_all_local)
VALUES (:o_id, :d_id, :w_id, :c_id,
:cr_date, 11, :o_ol_cnt, :o_all_local);

```

```

dummy_local := :d_id;

```

```

IF (dummy_local < 6) THEN
IF (dummy_local < 3) THEN
IF (dummy_local = 1) THEN
u1;
ELSE
u2;
END IF;
ELSE
IF (dummy_local = 3) THEN
u3;
ELSIF (dummy_local = 4) then
u4;
ELSE
u5;
END IF;
END IF;
ELSE
IF (dummy_local < 8) THEN
IF (dummy_local = 6) THEN
u6;
ELSE
u7;
END IF;
ELSE
IF (dummy_local = 8) THEN
u8;
ELSIF (dummy_local = 9) then
u9;
ELSE
u10;
END IF;
END IF;
END IF;

```

```

ELSE
IF (dummy_local < 8) THEN
IF (dummy_local = 6) THEN
u6;
ELSE
u7;
END IF;
ELSE
IF (dummy_local = 8) THEN
u8;
ELSIF (dummy_local = 9) then
u9;
ELSE
u10;
END IF;
END IF;
END IF;

```

```

dummy_local := sql%rowcount;

```

```

IF (dummy_local != cache_ol_cnt ) THEN fix_items; END IF;

```

```

FORALL idx IN 1..dummy_local
INSERT INTO ordl
(ol_o_id, ol_d_id, ol_w_id, ol_number, ol_delivery_d,
ol_i_id,

```

```

        ol_supply_w_id, ol_quantity, ol_amount, ol_dist_info)
VALUES (:o_id, :d_id, :w_id, inittppc.idx1arr(idx),
inittppc.nulldate,
        :ol_i_id(idx), :ol_supply_w_id(idx),
        :ol_quantity(idx), :ol_amount(idx), inittppc.s_dist(idx));

IF (dummy_local != :o_ol_cnt) THEN
    :o_ol_cnt := dummy_local;
    ROLLBACK;
END IF;

EXIT;

EXCEPTION
    WHEN not_serializable OR deadlock OR snapshot_too_old
THEN
    ROLLBACK;
    :retry := :retry + 1;
END;
END LOOP;
END;

```

```

        THEN
            raise NO_DATA_FOUND;
        END IF;
    INSERT INTO hist (h_c_id, h_c_d_id, h_c_w_id, h_d_id, h_w_id,
        h_amount, h_date, h_data)
    VALUES (:c_id, :c_d_id, :c_w_id, :d_id, :w_id, :h_amount,
        :cr_date, inittppc.ware_name || ' ' || inittppc.dist_name);
    EXIT;
EXCEPTION
    WHEN not_serializable OR deadlock OR snapshot_too_old THEN
        ROLLBACK;
        :retry := :retry + 1;
    END;
END LOOP;
END;

```

payz.sql

```

DECLARE /* payz */
not_serializable EXCEPTION;
PRAGMA EXCEPTION_INIT(not_serializable,-8177);
deadlock EXCEPTION;
PRAGMA EXCEPTION_INIT(deadlock,-60);
snapshot_too_old EXCEPTION;
PRAGMA EXCEPTION_INIT(snapshot_too_old,-1555);
BEGIN
    LOOP BEGIN
        UPDATE ware
        SET w_ytd = w_ytd+h_amount
        WHERE w_id = :w_id
        RETURNING w_name,
            w_street_1, w_street_2, w_city, w_state, w_zip
        INTO inittppc.ware_name,
            :w_street_1, :w_street_2, :w_city, :w_state, :w_zip;

        SELECT rowid
        BULK COLLECT INTO inittppc.row_id
        FROM cust
        WHERE c_d_id = :c_d_id AND c_w_id = :c_w_id AND c_last = :c_last
        ORDER BY c_last, c_d_id, c_w_id, c_first;

        inittppc.c_num := sql%rowcount;
        inittppc.cust_rowid := inittppc.row_id((inittppc.c_num) / 2);

        UPDATE cust
        SET c_balance = c_balance - :h_amount,
            c_ytd_payment = c_ytd_payment + :h_amount,
            c_payment_cnt = c_payment_cnt + 1
        WHERE rowid = inittppc.cust_rowid
        RETURNING
            c_id, c_first, c_middle, c_last, c_street_1, c_street_2,
            c_city, c_state, c_zip, c_phone,
            c_since, c_credit, c_credit_lim,
            c_discount, c_balance
        INTO :c_id, :c_first, :c_middle, :c_last,
            :c_street_1, :c_street_2, :c_city, :c_state,
            :c_zip, :c_phone, :c_since, :c_credit,
            :c_credit_lim, :c_discount, :c_balance;

        :c_data := '';
        IF :c_credit = 'BC' THEN
            UPDATE cust
            SET c_data = substr((to_char (:c_id) || '' ||
                to_char (:c_d_id) || '' ||
                to_char (:c_w_id) || '' ||
                to_char (:d_id) || '' ||
                to_char (:w_id) || '' ||
                to_char (:h_amount/100, '9999.99') || '' )
                || c_data, 1, 500)
            WHERE rowid = inittppc.cust_rowid
            RETURNING substr(c_data, 1, 200)
            INTO :c_data;

        END IF;

        UPDATE dist
        SET d_ytd = d_ytd+h_amount
        WHERE d_id = :d_id
        AND d_w_id = :w_id
        RETURNING d_name, d_street_1, d_street_2, d_city,
            d_state, d_zip
        INTO inittppc.dist_name, :d_street_1, :d_street_2, :d_city,
            :d_state, :d_zip;

        IF SQL%NOTFOUND

```

countuser

```

#!/usr/bin/ksh

#####
#@(#) Version: A.10.10 $Date: 2002/07/18 21:14:54 $
#
#(c) Copyright 1996, Hewlett-Packard Company, all rights reserved.
#####

count()
{
    if [ $OSTYPE = "HP-UX" ]
    then
        export UNIX95=1
        ps ux -L -ef #~/bin/ps.ux
    else
        ps -ef #/bin/ps
    fi
}

sleep $1
echo "Number of clients on `hostname` at `date`:" > /tmp/countusers.before
count >> /tmp/countusers.before

sleep $2
echo "Number of clients on `hostname` at `date`:" > /tmp/countusers.after
count >> /tmp/countusers.after

```

paynz.sql

```

DECLARE /* paynz */
not_serializable EXCEPTION;
PRAGMA EXCEPTION_INIT(not_serializable,-8177);
deadlock EXCEPTION;
PRAGMA EXCEPTION_INIT(deadlock,-60);
snapshot_too_old EXCEPTION;
PRAGMA EXCEPTION_INIT(snapshot_too_old,-1555);
BEGIN
    LOOP BEGIN
        UPDATE ware
        SET w_ytd = w_ytd + :h_amount
        WHERE w_id = :w_id
        RETURNING w_name, w_street_1, w_street_2, w_city, w_state, w_zip
        INTO inittppc.ware_name, :w_street_1, :w_street_2, :w_city,
            :w_state, :w_zip;

        UPDATE cust
        SET c_balance = c_balance - :h_amount,
            c_ytd_payment = c_ytd_payment + :h_amount,
            c_payment_cnt = c_payment_cnt + 1
        WHERE c_id = :c_id AND c_d_id = :c_d_id AND
            c_w_id = :c_w_id
        RETURNING rowid, c_first, c_middle, c_last, c_street_1,
            c_street_2, c_city, c_state, c_zip, c_phone,
            c_since, c_credit, c_credit_lim,
            c_discount, c_balance
        INTO inittppc.cust_rowid, :c_first, :c_middle, :c_last, :c_street_1,
            :c_street_2, :c_city, :c_state, :c_zip, :c_phone,
            :c_since, :c_credit, :c_credit_lim,
            :c_discount, :c_balance;
        IF SQL%NOTFOUND THEN
            raise NO_DATA_FOUND;
        END IF;
    END LOOP;
END;

```

```

IF :c_credit = 'BC' THEN
  UPDATE cust
    SET c_data = substr((to_char(:c_id) || ' ' ||
      to_char(:c_d_id) || ' ' ||
      to_char(:c_w_id) || ' ' ||
      to_char(:d_id) || ' ' ||
      to_char(:w_id) || ' ' ||
      to_char(:h_amount/100, '9999.99') || ' ');
      || c_data, 1, 500)
    WHERE rowid = inittpcc.cust_rowid
  RETURNING substr(c_data,1, 200)
  INTO :c_data;

END IF;

UPDATE dist
  SET d_ytd = d_ytd + :h_amount
  WHERE d_id = :d_id
  AND d_w_id = :w_id
  RETURNING d_name, d_street_1, d_street_2, d_city, d_state, d_zip
  INTO inittpcc.dist_name,:d_street_1,:d_street_2,:d_city,:d_state,
  :d_zip;
IF SQL%NOTFOUND THEN
  raise NO_DATA_FOUND;
END IF;

INSERT INTO hist (h_c_id, h_c_d_id, h_c_w_id, h_d_id, h_w_id,
  h_amount, h_date, h_data)
VALUES
(:c_id, :c_d_id, :c_w_id, :d_id, :w_id, :h_amount,
:cr_date, inittpcc.ware_name || ' ' || inittpcc.dist_name);
EXIT;

EXCEPTION
  WHEN not_serializable OR deadlock OR snapshot_too_old THEN
    ROLLBACK;
    :retry := :retry + 1;
END;

END LOOP;
END;

```

tpcc.c

```

#include "tpcc.h"

/* Error message strings */
const char *e_mesg[]={ "Transaction complete.", "Error", "Invalid item number.",
  "Not enough orders.", "Database ERROR !!!!" };

/* the name of each transaction */
const char *transaction_name[] =
{ "", "New_Order", "Payment", "Order-Status",
  "Delivery", "Stock-Level", "Deferred-Delivery" };

```

delay.c

```

*****
@(#) Version: A.10.10 $Date: 2005/04/11 10:00:23 $

(c) Copyright 1996, Hewlett-Packard Company, all rights reserved.
*****
#include <sys/time.h>
#include <errno.h>
#include <time.h>
#include "tpcc.h"

void
delay(double sec)
*****
delay sleeps for the specified number of seconds. (to closest 1/100th second)
*****
{
  struct timespec delay;

  /* if no delay, done */
  if (sec <= 0.0) return;

  /* add a portion of a clock tick to keep averages correct */
  sec += 1.0 / CLK_TCK;

  /* convert the delay to seconds and nanoseconds */
  delay.tv_sec = sec;
  delay.tv_nsec = (sec - delay.tv_sec) * 1000000000;

  if (nanosleep(&delay, NULL) == -1) {

```

```

if (errno != EINTR) {
  if (errno == ENOSYS) {
    struct timeval delay_select;
    /* Use select instead */
    delay_select.tv_sec = sec;
    delay_select.tv_usec = (sec -
delay_select.tv_sec)*1000000;
    if (select(0, NULL, NULL, NULL, &delay_select) < 0) {
      perror("select");
    } else {
      perror("nanosleep");
    }
  }
}
}

```

```

struct timeval tpcc_start_time;

void
initclock(void)
{
  gettimeofday(&tpcc_start_time, NULL);
}

```

```

TIME getclock(void)
*****
getclock returns the current time, expressed in seconds from start of run
*****
{
  struct timeval current;
  gettimeofday(&current, NULL);

  return elapsed_time(&current);
}

```

random.h

```

*****
@(#) Version: A.10.10 $Date: 2005/04/11 09:59:50 $

(c) Copyright 1996, Hewlett-Packard Company, all rights reserved.
*****
#ifndef TPCC_RANDOM
#define TPCC_RANDOM
#include "tpcc.h"

/*
 * Linux does not have efficient thread local data like HP-UX, so
 * just use the built in drand48 library which will provide a
 * thread-safe implementation probably by a mutex.
 */
#ifdef _linux
#define USE_DRAND48
#endif

#ifdef USE_DRAND48
double randy(void);
#endif

extern void GenerateLastNames(void);
extern ID RandomWarehouse(ID local, ID scale, int percent);
extern void RandomDelay(double mean, double adjust);
extern void Randomize(void);
extern void SetRandomSeed(int val);

/* Return a random value in the range [0.0..1.0) */
extern double RandomValue(void);

extern char lastNames[1000][16];

*****
/* RandomNumber selects a uniform random number from min to max inclusive */
*****
#ifdef USE_DRAND48
#define RandomNumber(min,max) \
  ((int)(drand48() * ((int)(max) - (int)(min) + 1)) + (int)(min))
#else
#define RandomNumber(min,max) \
  ((int)(randy() * ((int)(max) - (int)(min) + 1)) + (int)(min))
#endif

*****
/* NURandomNumber selects a non-uniform random number */
*****
#define NURandomNumber(a, min, max, c) \

```

```

((RandomNumber(0, a) | RandomNumber(min, max)) + (c)) % \
((max) - (min) + 1) + (min)

/*****
*****/
/* LastName selects a random TPC-C style last name. */
/*****
*****/
#define LastName(num, name) strcpy(name, lastNames[(num)])

#endif

```

audit_table.sh

```

#!/usr/bin/sh -f
#
#=====
==+
# Copyright (c) 1995 Oracle Corp, Belmont, CA |
# OPEN SYSTEMS PERFORMANCE GROUP |
# All Rights Reserved |
#=====
==+
# FILENAME
# upd_date.sh
# DESCRIPTION
# Insert start time into tpcc_audit_tab.
# USAGE
# upd_date.sh
#

sqlplus -s tpcc/tpcc <<!
set echo on;
delete from tpcc_audit_tab;
insert into tpcc_audit_tab (starttime)
select sysdate from dual;
commit;
select to_char(starttime, 'MM/DD/YY HH24:MI:SS') RUN_START_DATE
from tpcc_audit_tab;
quit;
!

```

countorders.sh

```

#!/usr/bin/sh
#*****
*****/
#@(#) Version: A.10.10 $Date: 2001/12/06 11:26:59 $
#
#(c) Copyright 1996, Hewlett-Packard Company, all rights reserved.
#*****
*****/

echo "`date`: counting orders in dist table"

sqlplus -s tpcc/tpcc <<!
column sum(d_next_o_id) format 9999999999;
select sum(d_next_o_id) from dist;
exit;
!

```

logsize.sh

```

#!/usr/bin/sh
#*****
*****/
#@(#) Version: A.10.10 $Date: 2001/08/24 15:20:45 $
#
#(c) Copyright 1996, Hewlett-Packard Company, all rights reserved.
#*****
*****/

echo "Log segment size at `date`"

```

dn.sh

```

sqlplus 'sys/change_on_install as sysdba' <<!
shutdown $1
exit;
!

# lsnr.sh stop # This will stop all the listeners

```

up.sh

```

#!/usr/bin/ksh

export ASYNC_BUF_CONF=256

./BUILD/stepenv.sh
if [ "$S1" != "" ]
then
  PFILE="pfile=$S1"
fi

sqlplus 'sys/change_on_install as sysdba' <<!
startup SPFILE
exit;
!

```

tkvcin.sql

-- The initnew package for storing variables used in the
-- New Order anonymous block

```

CREATE OR REPLACE PACKAGE initpcc
AS
  TYPE intarray IS TABLE OF INTEGER INDEX BY BINARY_INTEGER;
  TYPE distarray IS TABLE OF VARCHAR(24) INDEX BY BINARY_INTEGER;
  nulldate DATE;
  TYPE rowidarray IS TABLE OF ROWID INDEX BY PLS_INTEGER;
  s_dist intarray;
  idx_larr intarray;
  s_remote intarray;
  dist intarray;
  row_id rowidarray;
  cust_rowid rowid;
  dist_name VARCHAR2(11);
  ware_name VARCHAR2(11);
  c_num PLS_INTEGER;

  PROCEDURE init_no(idxarr intarray);
  PROCEDURE init_del;
  PROCEDURE init_pay;
END initpcc;
/
show errors;

```

```

CREATE OR REPLACE PACKAGE BODY initpcc AS
  PROCEDURE init_no (idxarr intarray)
  IS
  BEGIN
    -- initialize null date
    nulldate := TO_DATE('01-01-1811', 'MM-DD-YYYY');
    idx_larr := idxarr;
  END init_no;

  PROCEDURE init_del
  IS
  BEGIN
    FOR i IN 1 .. 10 LOOP
      dist(i) := i;
    END LOOP;
  END init_del;

  PROCEDURE init_pay IS
  BEGIN
    NULL;
  END init_pay;

END initpcc;
/
show errors
exit

```

/build/tkvin.sql

-- The initnew package for storing variables used in the
-- New Order anonymous block

```

CREATE OR REPLACE PACKAGE initpcc
AS
  TYPE intarray IS TABLE OF INTEGER INDEX BY BINARY_INTEGER;
  TYPE distarray IS TABLE OF VARCHAR(24) INDEX BY BINARY_INTEGER;
  nulldate DATE;
  TYPE rowidarray IS TABLE OF ROWID INDEX BY PLS_INTEGER;
  s_dist intarray;
  idx_larr intarray;
  s_remote intarray;
  dist intarray;
  row_id rowidarray;
  cust_rowid rowid;
  dist_name VARCHAR2(11);

```

```

ware_name      VARCHAR2(11);
c_num          PLS_INTEGER;

PROCEDURE init_no(idxarr intarray);
PROCEDURE init_del;
PROCEDURE init_pay;
END initpcc;
/
show errors;

CREATE OR REPLACE PACKAGE BODY initpcc AS
PROCEDURE init_no (idxarr intarray)
IS
BEGIN
-- initialize null date
nulldate := TO_DATE('01-01-1811', 'MM-DD-YYYY');
idxlarr := idxarr;
END init_no;

PROCEDURE init_del
IS
BEGIN
FOR i IN 1 .. 10 LOOP
dist(i) := i;
END LOOP;
END init_del;

PROCEDURE init_pay IS
BEGIN
NULL;
END init_pay;

END initpcc;
/
show errors
exit

```

Appendix B Database Design

The source code for the process to define, create and populate the **Oracle Database 10g Enterprise Edition Release2 TPC-C** database is included in this appendix.

B.1 Scripts

addfile.sh

```
#!/bin/sh
# $1 = tablespace name
# $2 = filename
# $3 = size
# $4 = temporary ts (1) or not (0)
# global variable $tpcc_listfiles, does not execute sql

if expr x$tpcc_listfiles = xt > /dev/null; then
echo $2 $3 >> $tpcc_bench/files.dat
exit 0
fi

if expr $4 = 1 > /dev/null; then
altersql="alter tablespace $1 add tempfile '$2' size $3 reuse;"
else
altersql="alter tablespace $1 add datafile '$2' size $3 reuse autoextend on;"
fi

$tpcc_sqlplus $tpcc_user_pass <<!
spool addfile_$1.log
set echo on
$altersql
set echo off
spool off
exit ;
!
```

addts.sh

```
#!/bin/sh
# $1 = tablespace name
# $2 = filename
# $3 = size
# $4 = uniform size
# $5 = block size
# $6 = temporary ts (1) or not (0)
# $7 = bitmapped manage (t) or not (f) or (d) for dictionary
# global variable $tpcc_listfiles, does not execute sql

if expr x$tpcc_listfiles = xt > /dev/null; then
echo $2 $3 >> $tpcc_bench/files.dat
exit 0
fi

if expr $5 = auto > /dev/null; then
bssql=
else
bssql="blocksize $5"
fi

if expr $6 = 1 > /dev/null; then
createsql="create temporary tablespace $1 tempfile '$2' size $3 reuse extent management
local uniform size $4;"
else
if expr x$7 = xt > /dev/null; then
createsql="create tablespace $1 datafile '$2' size $3 reuse extent management local
uniform size $4 segment space management auto $bssql nologging ;"
else
if expr x$7 = xd > /dev/null; then
createsql="create tablespace $1 datafile '$2' size $3 reuse extent management dictionary
nologging $bssql;"
else
createsql="create tablespace $1 datafile '$2' size $3 reuse extent management local
uniform size $4 segment space management manual $bssql nologging ;"
fi
fi
fi

$tpcc_sqlplus $tpcc_user_pass <<!
spool createts_$1.log
set echo on
```

```
drop tablespace $1 including contents;
$createsql
set echo off
spool off
exit ;
!
```

analyze.sh

```
#!/bin/sh
$tpcc_sqlplus $tpcc_user_pass @$ {tpcc_sql_dir}/analyze > $tpcc_log_dir/junk 2>&1

if test $? -ne 0
then
exit 1;
else
exit 0;
fi
```

analyze.sql

```
spool analyze.log;
set echo on;

connect tpcc/tpcc

execute dbms_stats.GATHER_TABLE_STATS (OWNNNAME=>'TPCC', -
TABNAME=>'STOK', -
PARTNAME=>NULL, -
ESTIMATE_PERCENT=>1, -
BLOCK_SAMPLE=>TRUE, -
METHOD_OPT=>'FOR ALL COLUMNS SIZE 1', -
DEGREE=>200, -
GRANULARITY=>'DEFAULT', -
CASCADE=>TRUE);

execute dbms_stats.GATHER_TABLE_STATS (OWNNNAME=>'TPCC', -
TABNAME=>'CUST', -
PARTNAME=>NULL, -
ESTIMATE_PERCENT=>1, -
BLOCK_SAMPLE=>TRUE, -
METHOD_OPT=>'FOR ALL COLUMNS SIZE 1', -
DEGREE=>200, -
GRANULARITY=>'DEFAULT', -
CASCADE=>TRUE);

execute dbms_stats.GATHER_TABLE_STATS (OWNNNAME=>'TPCC', -
TABNAME=>'ORDR', -
PARTNAME=>NULL, -
ESTIMATE_PERCENT=>1, -
BLOCK_SAMPLE=>TRUE, -
METHOD_OPT=>'FOR ALL COLUMNS SIZE 1', -
DEGREE=>200, -
GRANULARITY=>'DEFAULT', -
CASCADE=>TRUE);

execute dbms_stats.GATHER_TABLE_STATS (OWNNNAME=>'TPCC', -
TABNAME=>'ORDL', -
PARTNAME=>NULL, -
ESTIMATE_PERCENT=>1, -
BLOCK_SAMPLE=>TRUE, -
METHOD_OPT=>'FOR ALL COLUMNS SIZE 1', -
DEGREE=>200, -
GRANULARITY=>'DEFAULT', -
CASCADE=>TRUE);

execute dbms_stats.GATHER_TABLE_STATS (OWNNNAME=>'TPCC', -
TABNAME=>'NORD', -
PARTNAME=>NULL, -
ESTIMATE_PERCENT=>1, -
BLOCK_SAMPLE=>TRUE, -
METHOD_OPT=>'FOR ALL COLUMNS SIZE 1', -
DEGREE=>100, -
GRANULARITY=>'DEFAULT', -
CASCADE=>TRUE);

execute dbms_stats.GATHER_TABLE_STATS(OWNNNAME=>'TPCC', -
TABNAME=>'HIST', -
PARTNAME=>NULL, -
ESTIMATE_PERCENT=>1, -
BLOCK_SAMPLE=>TRUE, -
METHOD_OPT=>'FOR ALL COLUMNS SIZE 1', -
DEGREE=>100, -
GRANULARITY=>'DEFAULT', -
CASCADE=>TRUE);

execute dbms_stats.GATHER_TABLE_STATS(OWNNNAME=>'TPCC', -
TABNAME=>'DIST', -
```

```

PARTNAME=>NULL, -
ESTIMATE_PERCENT=>1, -
BLOCK_SAMPLE=>TRUE, -
METHOD_OPT=>'FOR ALL COLUMNS SIZE 1', -
DEGREE=>10, -
GRANULARITY=>'DEFAULT', -
CASCADE=>TRUE);

execute dbms_stats.GATHER_TABLE_STATS(OWNNAME=>'TPCC', -
TABNAME=>'ITEM', -
PARTNAME=>NULL, -
ESTIMATE_PERCENT=>10, -
BLOCK_SAMPLE=>TRUE, -
METHOD_OPT=>'FOR ALL COLUMNS SIZE 1', -
DEGREE=>1, -
GRANULARITY=>'DEFAULT', -
CASCADE=>TRUE);

execute dbms_stats.GATHER_TABLE_STATS(OWNNAME=>'TPCC', -
TABNAME=>'WARE', -
PARTNAME=>NULL, -
ESTIMATE_PERCENT=>10, -
BLOCK_SAMPLE=>TRUE, -
METHOD_OPT=>'FOR ALL COLUMNS SIZE 1', -
DEGREE=>10, -
GRANULARITY=>'DEFAULT', -
CASCADE=>TRUE);

set echo off;
spool off;

exit sql.sqlcode;

```

create_cache_views.sql

```

rem This script creates four views that when queried will return
rem the total number of buffers in the buffer cache and the total
rem number of cloned buffers from each of the database's tablespaces.
rem
rem This assumes that each table and index is in its own tablespace.
rem If this is not the case, another query can be used which uses the
rem database's object tables to decipher the different objects. However,
rem this query is slower.
rem
rem This script assumes 7.3.x. If you are using V7.2.x or below, please
rem replace svrmgrl with sqlldr lmode=y.
rem
rem Modification History:
rem
rem wbattist 16-Jun-1996 Create two additional views to keep
rem track of the number of clones in each
rem tablespace.
rem
rem wbattist 24-May-1995 Add the state check for the cbf view
rem to ensure that cloned blocks are not
rem counted.
rem

connect $oracle_dba/$oracle_dba_password;
set echo on;
drop view cbf;
create view cbf as
select distinct(dbarfil) file#, count(1) blocks
from x$bhh
where dbarfil > 0 and state <> 3
group by dbarfil;
drop view cbt;
create view cbt as
select ts.name name,sum(cbf.blocks) buffers
from cbf, file$, ts$
where cbf.file#=file$.file# and file$.ts#=ts$.ts#
group by file$.ts#, ts$.name;
drop view cbfcln;
create view cbfcln as
select distinct(dbarfil) file#, count(1) blocks
from x$bhh
where dbarfil > 0
group by dbarfil;
drop view cbtn;
create view cbtn as
select ts.name name,sum(cbfcln.blocks) buffers
from cbfcln, file$, ts$
where cbfcln.file#=file$.file# and file$.ts#=ts$.ts#
group by file$.ts#, ts$.name;

set echo off;

```

Create_spacestats.sql

```

@space_init
@space_get 349440 4000000
@space_rpt
spool off
exit sql.sqlcode;

```

Create_storedprocs.sql

```

spool createstoreprocs.log
@tkvcin.sql
spool off
exit sql.sqlcode;

```

Createdb.sql

```

/* created automatically by /BUILD/scripts/buildcreatedb.sh Wed Dec 6 08:27:41 PST
2006 */
spool createdb.log

set echo on

shutdown abort

startup pfile=p_create.ora nomount
create database tpcc
controlfile reuse
maxinstances 1
datafile
'/BUILD/dbs/tpcc_disks/system_1' size 400M reuse
logfile '/BUILD/dbs/tpcc_disks/log_1_1' size 8191M reuse,
'/BUILD/dbs/tpcc_disks/log_1_2' size 8191M reuse
sysaux datafile '/BUILD/dbs/tpcc_disks/tpccaux' size 120M reuse ;

```

```

create undo tablespace undo_1 datafile
'/BUILD/dbs/tpcc_disks/roll1' size 8096M reuse blocksize 8K;

set echo off
exit sql.sqlcode

```

createindex_icust1.sql

```

/* created automatically by /BUILD/scripts/buildcreateindex.sh Wed Dec 6 08:29:08 PST
2006 */
set timing on
set sqlblanklines on
spool createindex_icust1.log ;
set echo on ;
drop index icust1 ;
create unique index icust1 on cust ( c_w_id
, c_d_id
, c_id )
pctfree 1 initrans 3
storage ( buffer_pool default )
parallel 128
compute statistics
tablespace icust1_0 ;
set echo off
spool off
exit sql.sqlcode;

```

createindex_icust2.sql

```

/* created automatically by /BUILD/scripts/buildcreateindex.sh Wed Dec 6 08:29:10 PST
2006 */
set timing on
set sqlblanklines on
spool createindex_icust2.log ;
set echo on ;
drop index icust2 ;
create unique index icust2 on cust ( c_last
, c_w_id
, c_d_id
, c_first
, c_id )
pctfree 1 initrans 3
storage ( buffer_pool default )
parallel 128
compute statistics
tablespace icust2_0 ;
set echo off
spool off

```

```
exit sql.sqlcode;
```

createindex_idist.sql

```
/* created automatically by /BUILD/scripts/buildcreateindex.sh Wed Dec 6 08:29:11 PST
2006 */
set timing on
set sqlblanklines on
spool createindex_idist.log ;
set echo on ;
drop index idist ;
create unique index idist on dist ( d_w_id
, d_id )
pctfree 5 intrans 3
storage ( buffer_pool default )
parallel 1
compute statistics
tablespace idist_0 ;
set echo off
spool off
exit sql.sqlcode;
```

createindex_iitem.sql

```
/* created automatically by /BUILD/scripts/buildcreateindex.sh Wed Dec 6 08:29:14 PST
2006 */
set timing on
set sqlblanklines on
spool createindex_iitem.log ;
set echo on ;
drop index iitem ;
create unique index iitem on item ( i_id )
pctfree 5 intrans 4
storage ( buffer_pool default )

compute statistics
tablespace iitem_0 ;
set echo off
spool off
exit sql.sqlcode;
```

createindex_inordr.sql

```
/* created automatically by /BUILD/scripts/buildcreateindex.sh Wed Dec 6 08:29:20 PST
2006 */
set timing on
exit 0;
```

createindex_iordr1.sql

```
/* created automatically by /BUILD/scripts/buildcreateindex.sh Wed Dec 6 08:29:15 PST
2006 */
set timing on
exit 0;
```

createindex_iordr2.sql

```
/* created automatically by /BUILD/scripts/buildcreateindex.sh Wed Dec 6 08:29:17 PST
2006 */
set timing on
set sqlblanklines on
spool createindex_iordr2.log ;
set echo on ;
drop index iordr2 ;
create unique index iordr2 on ord ( o_w_id
, o_d_id
, o_c_id
, o_id )
global partition by range (o_w_id) (
partition iordr2_1 values less than ( 21840 ) tablespace iordr2_0,
partition iordr2_2 values less than ( 43680 ) tablespace iordr2_0,
partition iordr2_3 values less than ( 65520 ) tablespace iordr2_0,
partition iordr2_4 values less than ( 87360 ) tablespace iordr2_0,
partition iordr2_5 values less than ( 109200 ) tablespace iordr2_0,
partition iordr2_6 values less than ( 131040 ) tablespace iordr2_0,
partition iordr2_7 values less than ( 152880 ) tablespace iordr2_0,
partition iordr2_8 values less than ( 174720 ) tablespace iordr2_0,
partition iordr2_9 values less than ( 196560 ) tablespace iordr2_0,
partition iordr2_10 values less than ( 218400 ) tablespace iordr2_0,
partition iordr2_11 values less than ( 240240 ) tablespace iordr2_0,
partition iordr2_12 values less than ( 262080 ) tablespace iordr2_0,
partition iordr2_13 values less than ( 283920 ) tablespace iordr2_0,
partition iordr2_14 values less than ( 305760 ) tablespace iordr2_0,
partition iordr2_15 values less than ( 327600 ) tablespace iordr2_0,
```

```
partition iordr2_16 values less than ( MAXVALUE ) tablespace iordr2_0
)
pctfree 25 intrans 4
storage ( buffer_pool default )
parallel 128
compute statistics;
set echo off
spool off
exit sql.sqlcode;
```

createindex_istok.sql

```
/* created automatically by /BUILD/scripts/buildcreateindex.sh Wed Dec 6 08:29:13 PST
2006 */
set timing on
set sqlblanklines on
spool createindex_istok.log ;
set echo on ;
drop index istok ;
create unique index istok on stok ( s_i_id
, s_w_id )
pctfree 1 intrans 3
storage ( buffer_pool default )
parallel 64
compute statistics
tablespace istok_0 ;
set echo off
spool off
exit sql.sqlcode;
```

Createindex_aware.sql

```
/* created automatically by /BUILD/scripts/buildcreateindex.sh Wed Dec 6 08:29:07 PST
2006 */
set timing on
set sqlblanklines on
spool createindex_aware.log ;
set echo on ;
drop index aware ;
create unique index aware on ware ( w_id )
pctfree 1 intrans 3
storage ( buffer_pool default )
parallel 1
compute statistics
tablespace aware_0 ;
set echo off
spool off
exit sql.sqlcode;
```

createmisc.sh

```
#!/bin/sh

$tpcc_sqlplus $tpcc_sqlplus_args << !
$tpcc_internal_connect

spool createmisc.log
set echo on;
alter user tpcc temporary tablespace system;
grant execute on dbms_lock to public;
grant execute on dbms_pipe to public;
grant select on v_\\$parameter to public;

REM
REM begin plsql_mon.sql
REM

connect tpcc/tpcc;
set echo on;
CREATE OR REPLACE PACKAGE plsql_mon_pack
IS
PROCEDURE print
(
info VARCHAR2
);
END;
/
show errors;

CREATE OR REPLACE PACKAGE BODY plsql_mon_pack
IS
PROCEDURE print
(
info VARCHAR2
)
IS
s NUMBER;
BEGIN
```



```

dbms_pipe.pack_message (info);
s := dbms_pipe.send_message ('plsqli_mon');
IF (s <> 0) THEN
    raise_application_error (-20000, 'Error: ' || to_char(s) ||
        ' sending on pipe');
END IF;
END;
/
show errors;

set echo off;

REM
REM end plsqli_mon.sql
REM

REM
REM begin cre_tab.sql
REM

connect tpcc/tpcc;
set echo on;

drop table temp_o1;
drop table temp_no;
drop table temp_o2;
drop table temp_o1;
drop table tpcc_audit_tab;

create table temp_o1 (
    o_w_id integer,
    o_d_id integer,
    o_o_id integer);

create table temp_no (
    no_w_id integer,
    no_d_id integer,
    no_o_id integer);

create table temp_o2 (
    o_w_id integer,
    o_d_id integer,
    o_count integer);

create table temp_o1 (
    ol_w_id integer,
    ol_d_id integer,
    ol_count integer);

create table tpcc_audit_tab (starttime date);

delete from tpcc_audit_tab;

set echo off;

REM
REM end cre_tab.sql
REM

REM
REM begin views.sql
REM

connect tpcc/tpcc;
set echo on;

create or replace view wh_cust
(w_id, w_tax, c_id, c_d_id, c_w_id, c_discount, c_last, c_credit)
as select w.w_id, w.w_tax,
    c.c_id, c.c_d_id, c.e_w_id, c.c_discount, c.c_last, c.c_credit
    from cust c, ware w
    where w.w_id = c.e_w_id;

create or replace view wh_dist
(w_id, d_id, d_tax, d_next_o_id, w_tax)
as select w.w_id, d.d_id, d.d_tax, d.d_next_o_id, w.w_tax
    from dist d, ware w
    where w.w_id = d.d_w_id;

create or replace view stock_item
(i_id, s_w_id, i_price, i_name, i_data, s_data, s_quantity,
s_order_cnt, s_ytd, s_remote_cnt,
s_dist_01, s_dist_02, s_dist_03, s_dist_04, s_dist_05,
s_dist_06, s_dist_07, s_dist_08, s_dist_09, s_dist_10)
as
select i.i_id, s_w_id, i.i_price, i.i_name, i.i_data, s_data, s_quantity,
s_order_cnt, s_ytd, s_remote_cnt,
s_dist_01, s_dist_02, s_dist_03, s_dist_04, s_dist_05,
s_dist_06, s_dist_07, s_dist_08, s_dist_09, s_dist_10
    from stok s, item i
    where i.i_id = s.s_i_id;

set echo off;

```

```

REM
REM end views.sql
REM

REM
REM begin dml.sql
REM
connect tpcc/tpcc;
set echo on;

alter table ware disable table lock;
alter table dist disable table lock;
alter table cust disable table lock;
alter table hist disable table lock;
alter table item disable table lock;
alter table stok disable table lock;
alter table ordr disable table lock;
alter table nord disable table lock;
alter table ordl disable table lock;

set echo off;

REM
REM end dml.sql
REM

REM
REM begin extent.sql
REM

SSYS_CONNECTION_STRING

@Stpcc_sql_dir/extent

@Stpcc_sql_dir/freeext

exit sql.sqlcode;

!

```

createspacestats.sh

```

#!/bin/sh
cd $Stpcc_genscripts_dir
$Stpcc_sqlplus $Stpcc_dba_user_pass @$Stpcc_genscripts_dir/createspacestats > junk 2>&1

if test $? -ne 0
then
    exit 1;
else
    exit 0;
fi

```

creatstats.sh

```

#!/bin/sh

cstat=c_stat
if test $tpcc_np -gt 1 ; then
    cstat=c_stat_rac
fi

$Stpcc_sqlplus $Stpcc_sqlplus_args << !
$Stpcc_internal_connect

REM
REM create tablespace for statspack user sp begin
REM

spool creatstats.log

set echo on
drop tablespace sp_0 including contents;
create tablespace sp_0 datafile '$(tpcc_disks_location)sp_0' size $tpcc_statspack_size
reuse autoextend on extent management local uniform size 1M nologging ;
spool off

REM
REM create tablespace for statspack user sp end
REM

REM
REM begin now call spcreate to create statspack sp package
REM

$Stpcc_internal_connect

define default_tablespace='sp_0'

```

```

define temporary_tablespace='temp_0'

@SORACLE_HOME/rdbms/admin/spcreate
perfstat

REM note that the last thing (after spcreate) is the perfstat password.
REM since we're not worried about security, perfstat will do.

REM
REM tpcc stat table for NT, it is not working so I comment it out
REM shui.lau@oracle.com it is better to use perfmon
REM

@Stpcc_sql_dir/cs_tpcc
@Stpcc_sql_dir/cs_cpu
@Stpcc_sql_dir/cs_os
@Stpcc_sql_dir/cs_proc
@Stpcc_sql_dir/cs_thread

REM
REM tpcc result table for unix and NT
REM

@Stpcc_sql_dir/${cstat}
@Stpcc_sql_dir/pst_c

!

```

createstoredprocs.sh

```

#!/bin/sh
cd $tpcc_genscripts_dir
$tpcc_sqlplus $tpcc_user_pass @$({tpcc_genscripts_dir}/createstoredprocs > junk 2>&1

if test $? -ne 0
then
  exit 1;
else
  exit 0;
fi

```

createtable_cust.sql

```

/* created automatically by /BUILD/scripts/buildcreatetable.sh Wed Dec 6 08:27:46 PST
2006 */
set timing on
set sqlblanklines on
spool createtable_cust.log
set echo on
drop cluster custcluster including tables ;

create cluster custcluster (
  c_id number
  , c_d_id number
  , c_w_id number
)
single table
hashkeys 10483200000
hash is ( (abs(c_id - 1) * 13650 + mod((c_w_id - 1), 1365) * 10 + abs(c_d_id - 1) + trunc
((c_w_id - 1) / 1365) * 1365 * 30000) )
size 360
pctfree 0 intrans 3
storage ( buffer_pool recycle ) parallel ( degree 128 )
tablespace cust_0;

create table cust (
  c_id number
  , c_d_id number
  , c_w_id number
  , c_discount number
  , c_credit char(2)
  , c_last varchar2(16)
  , c_first varchar2(16)
  , c_credit_lim number
  , c_balance number
  , c_ytd_payment number
  , c_payment_cnt number
  , c_delivery_cnt number
  , c_street_1 varchar2(20)
  , c_street_2 varchar2(20)
  , c_city varchar2(20)
  , c_state char(2)
  , c_zip char(9)
  , c_phone char(16)
  , c_since date
  , c_middle char(2)
  , c_data char(500)
)
cluster custcluster (
  c_id

```

```

, c_d_id
, c_w_id
);
set echo off
spool off
exit sql.sqlcode;

```

Createtable_dist.sql

```

/* created automatically by /BUILD/scripts/buildcreatetable.sh Wed Dec 6 08:27:53 PST
2006 */
set timing on
set sqlblanklines on
spool createtable_dist.log
set echo on
drop cluster distcluster including tables ;

create cluster distcluster (
  d_id number
  , d_w_id number
)
single table
hashkeys 3494400
hash is ( ((d_w_id * 10) + d_id )
size 1448
intrans 4
storage ( buffer_pool default )
tablespace dist_0;

create table dist (
  d_id number
  , d_w_id number
  , d_ytd number
  , d_next_o_id number
  , d_tax number
  , d_name varchar2(10)
  , d_street_1 varchar2(20)
  , d_street_2 varchar2(20)
  , d_city varchar2(20)
  , d_state char(2)
  , d_zip char(9)
)
cluster distcluster (
  d_id
  , d_w_id
);
set echo off
spool off
exit sql.sqlcode;

```

createtable_hist.sql

```

/* created automatically by /BUILD/scripts/buildcreatetable.sh Wed Dec 6 08:27:57 PST
2006 */
set timing on
set sqlblanklines on
spool createtable_hist.log
set echo on
drop table hist ;

create table hist (
  h_c_id number
  , h_c_d_id number
  , h_c_w_id number
  , h_d_id number
  , h_w_id number
  , h_date date
  , h_amount number
  , h_data varchar2(24)
)
partition by range(h_w_id) (
  partition hist_0 values less than (5460) pctfree 5 intrans 4 storage (buffer_pool recycle)
  tablespace hist_0,
  partition hist_1 values less than (10920) pctfree 5 intrans 4 storage (buffer_pool recycle)
  tablespace hist_0,
  partition hist_2 values less than (16380) pctfree 5 intrans 4 storage (buffer_pool recycle)
  tablespace hist_0,
  partition hist_3 values less than (21840) pctfree 5 intrans 4 storage (buffer_pool recycle)
  tablespace hist_0,
  partition hist_4 values less than (27300) pctfree 5 intrans 4 storage (buffer_pool recycle)
  tablespace hist_0,
  partition hist_5 values less than (32760) pctfree 5 intrans 4 storage (buffer_pool recycle)
  tablespace hist_0,
  partition hist_6 values less than (38220) pctfree 5 intrans 4 storage (buffer_pool recycle)
  tablespace hist_0,
  partition hist_7 values less than (43680) pctfree 5 intrans 4 storage (buffer_pool recycle)
  tablespace hist_0,
  partition hist_8 values less than (49140) pctfree 5 intrans 4 storage (buffer_pool recycle)
  tablespace hist_0,

```

```

partition hist_9 values less than (54600) pctfree 5 intrans 4 storage (buffer_pool recycle)
tablespace hist_0,
partition hist_10 values less than (60060) pctfree 5 intrans 4 storage (buffer_pool recycle)
tablespace hist_0,
partition hist_11 values less than (65520) pctfree 5 intrans 4 storage (buffer_pool recycle)
tablespace hist_0,
partition hist_12 values less than (70980) pctfree 5 intrans 4 storage (buffer_pool recycle)
tablespace hist_0,
partition hist_13 values less than (76440) pctfree 5 intrans 4 storage (buffer_pool recycle)
tablespace hist_0,
partition hist_14 values less than (81900) pctfree 5 intrans 4 storage (buffer_pool recycle)
tablespace hist_0,
partition hist_15 values less than (87360) pctfree 5 intrans 4 storage (buffer_pool recycle)
tablespace hist_0,
partition hist_16 values less than (92820) pctfree 5 intrans 4 storage (buffer_pool recycle)
tablespace hist_0,
partition hist_17 values less than (98280) pctfree 5 intrans 4 storage (buffer_pool recycle)
tablespace hist_0,
partition hist_18 values less than (103740) pctfree 5 intrans 4 storage (buffer_pool recycle)
tablespace hist_0,
partition hist_19 values less than (109200) pctfree 5 intrans 4 storage (buffer_pool recycle)
tablespace hist_0,
partition hist_20 values less than (114660) pctfree 5 intrans 4 storage (buffer_pool recycle)
tablespace hist_0,
partition hist_21 values less than (120120) pctfree 5 intrans 4 storage (buffer_pool recycle)
tablespace hist_0,
partition hist_22 values less than (125580) pctfree 5 intrans 4 storage (buffer_pool recycle)
tablespace hist_0,
partition hist_23 values less than (131040) pctfree 5 intrans 4 storage (buffer_pool recycle)
tablespace hist_0,
partition hist_24 values less than (136500) pctfree 5 intrans 4 storage (buffer_pool recycle)
tablespace hist_0,
partition hist_25 values less than (141960) pctfree 5 intrans 4 storage (buffer_pool recycle)
tablespace hist_0,
partition hist_26 values less than (147420) pctfree 5 intrans 4 storage (buffer_pool recycle)
tablespace hist_0,
partition hist_27 values less than (152880) pctfree 5 intrans 4 storage (buffer_pool recycle)
tablespace hist_0,
partition hist_28 values less than (158340) pctfree 5 intrans 4 storage (buffer_pool recycle)
tablespace hist_0,
partition hist_29 values less than (163800) pctfree 5 intrans 4 storage (buffer_pool recycle)
tablespace hist_0,
partition hist_30 values less than (169260) pctfree 5 intrans 4 storage (buffer_pool recycle)
tablespace hist_0,
partition hist_31 values less than (174720) pctfree 5 intrans 4 storage (buffer_pool recycle)
tablespace hist_0,
partition hist_32 values less than (180180) pctfree 5 intrans 4 storage (buffer_pool recycle)
tablespace hist_0,
partition hist_33 values less than (185640) pctfree 5 intrans 4 storage (buffer_pool recycle)
tablespace hist_0,
partition hist_34 values less than (191100) pctfree 5 intrans 4 storage (buffer_pool recycle)
tablespace hist_0,
partition hist_35 values less than (196560) pctfree 5 intrans 4 storage (buffer_pool recycle)
tablespace hist_0,
partition hist_36 values less than (202020) pctfree 5 intrans 4 storage (buffer_pool recycle)
tablespace hist_0,
partition hist_37 values less than (207480) pctfree 5 intrans 4 storage (buffer_pool recycle)
tablespace hist_0,
partition hist_38 values less than (212940) pctfree 5 intrans 4 storage (buffer_pool recycle)
tablespace hist_0,
partition hist_39 values less than (218400) pctfree 5 intrans 4 storage (buffer_pool recycle)
tablespace hist_0,
partition hist_40 values less than (223860) pctfree 5 intrans 4 storage (buffer_pool recycle)
tablespace hist_0,
partition hist_41 values less than (229320) pctfree 5 intrans 4 storage (buffer_pool recycle)
tablespace hist_0,
partition hist_42 values less than (234780) pctfree 5 intrans 4 storage (buffer_pool recycle)
tablespace hist_0,
partition hist_43 values less than (240240) pctfree 5 intrans 4 storage (buffer_pool recycle)
tablespace hist_0,
partition hist_44 values less than (245700) pctfree 5 intrans 4 storage (buffer_pool recycle)
tablespace hist_0,
partition hist_45 values less than (251160) pctfree 5 intrans 4 storage (buffer_pool recycle)
tablespace hist_0,
partition hist_46 values less than (256620) pctfree 5 intrans 4 storage (buffer_pool recycle)
tablespace hist_0,
partition hist_47 values less than (262080) pctfree 5 intrans 4 storage (buffer_pool recycle)
tablespace hist_0,
partition hist_48 values less than (267540) pctfree 5 intrans 4 storage (buffer_pool recycle)
tablespace hist_0,
partition hist_49 values less than (273000) pctfree 5 intrans 4 storage (buffer_pool recycle)
tablespace hist_0,
partition hist_50 values less than (278460) pctfree 5 intrans 4 storage (buffer_pool recycle)
tablespace hist_0,
partition hist_51 values less than (283920) pctfree 5 intrans 4 storage (buffer_pool recycle)
tablespace hist_0,
partition hist_52 values less than (289380) pctfree 5 intrans 4 storage (buffer_pool recycle)
tablespace hist_0,
partition hist_53 values less than (294840) pctfree 5 intrans 4 storage (buffer_pool recycle)
tablespace hist_0,
partition hist_54 values less than (300300) pctfree 5 intrans 4 storage (buffer_pool recycle)
tablespace hist_0,
partition hist_55 values less than (305760) pctfree 5 intrans 4 storage (buffer_pool recycle)
tablespace hist_0,

```

```

partition hist_56 values less than (311220) pctfree 5 intrans 4 storage (buffer_pool recycle)
tablespace hist_0,
partition hist_57 values less than (316680) pctfree 5 intrans 4 storage (buffer_pool recycle)
tablespace hist_0,
partition hist_58 values less than (322140) pctfree 5 intrans 4 storage (buffer_pool recycle)
tablespace hist_0,
partition hist_59 values less than (327600) pctfree 5 intrans 4 storage (buffer_pool recycle)
tablespace hist_0,
partition hist_60 values less than (333060) pctfree 5 intrans 4 storage (buffer_pool recycle)
tablespace hist_0,
partition hist_61 values less than (338520) pctfree 5 intrans 4 storage (buffer_pool recycle)
tablespace hist_0,
partition hist_62 values less than (343980) pctfree 5 intrans 4 storage (buffer_pool recycle)
tablespace hist_0,
partition hist_63 values less than (MAXVALUE) pctfree 5 intrans 4 storage (buffer_pool
recycle) tablespace hist_0);
set echo off
spool off
exit sql.sqlcode;

```

createtable_item.sql

```

/* created automatically by /BUILD/scripts/buildcreatetable.sh Wed Dec 6 08:28:05 PST
2006 */
set timing on
set sqlblanklines on
spool createtable_item.log
set echo on
drop cluster itemcluster including tables;

```

```

create cluster itemcluster (
i_id number(6,0)
)
single table
hashkeys 100000
hash is ( i_id )
size 120
pctfree 0 intrans 3
storage ( buffer_pool keep )
tablespace item_0;

```

```

create table item (
i_id number(6,0)
,i_name varchar2(24)
,i_price number
,i_data varchar2(50)
,i_im_id number
)
cluster itemcluster (
i_id
);
set echo off
spool off
exit sql.sqlcode;

```

createtable_nord.sql

```

/* created automatically by /BUILD/scripts/buildcreatetable.sh Wed Dec 6 08:28:14 PST
2006 */
set timing on
set sqlblanklines on
spool createtable_nord.log
set echo on
drop cluster nordcluster_queue including tables;

```

```

create cluster nordcluster_queue (
no_w_id number
,no_d_id number
,no_o_id number SORT
)

hashkeys 3494400
hash is ( (no_w_id - 1) * 10 + no_d_id - 1 )
size 190
tablespace nord_0;

create table nord (
no_w_id number
,no_d_id number
,no_o_id number sort
,constraint nord_uk primary key ( no_w_id
,no_d_id
,no_o_id )
)
cluster nordcluster_queue (
no_w_id
,no_d_id
,no_o_id
);
set echo off

```

```

spool off
exit sql.sqlcode;

```

createtable_ordl.sql

```

/* created automatically by /BUILD/scripts/buildcreatetable.sh Wed Dec 6 08:28:11 PST
2006 */
set timing on
set sqlblanklines on
spool createtable_ordl.log
set echo on
create table ordl (
  ol_w_id number
, ol_d_id number
, ol_o_id number sort
, ol_number number sort
, ol_i_id number
, ol_delivery_d date
, ol_amount number
, ol_supply_w_id number
, ol_quantity number
, ol_dist_info char(24)
, constraint ordl_uk primary key (ol_w_id, ol_d_id, ol_o_id, ol_number )) CLUSTER
ordrcluster_queue(ol_w_id, ol_d_id, ol_o_id, ol_number);
set echo off
spool off
exit sql.sqlcode;

```

createtable_ordr.sql

```

/* created automatically by /BUILD/scripts/buildcreatetable.sh Wed Dec 6 08:28:08 PST
2006 */
set timing on
set sqlblanklines on
spool createtable_ordr.log
set echo on
drop cluster ordrcluster_queue including tables ;

create cluster ordrcluster_queue (
  o_w_id number
, o_d_id number
, o_id number SORT
, o_number number SORT
)

  hashkeys 3494400
  hash is ( (o_w_id - 1) * 10 + o_d_id - 1 )
  size 1490
  tablespace ordr_0;

create table ordr (
  o_w_id number
, o_d_id number
, o_c_id number
, o_carrier_id number
, o_ol_cnt number
, o_all_local number
, o_entry_d date
, constraint ordr_uk primary key ( o_w_id
, o_d_id
, o_id )
) cluster ordrcluster_queue (
  o_w_id
, o_d_id
, o_id
);
set echo off
spool off
exit sql.sqlcode;

```

createtable_stok.sql

```

/* created automatically by /BUILD/scripts/buildcreatetable.sh Wed Dec 6 08:37:13 PST
2006 */

alter session set "_column_compression_factor"=175;

set timing on
set sqlblanklines on
spool createtable_stok.log
set echo on
drop cluster stokcluster including tables ;

create cluster stokcluster (
  s_i_id number
, s_w_id number

```

```

)
single table
hashkeys 34944000000
hash is ( (abs(s_i_id - 1) * 560 + mod((s_w_id - 1), 560) + trunc ((s_w_id - 1) / 560) *
560* 100000))
size 270
pctfree 0 intrans 2 maxtrans 2
storage ( buffer_pool keep ) parallel ( degree 128 )
tablespace stok_0;

```

```

create table stok (
  s_i_id number
, s_w_id number
, s_quantity number
, s_ytd number
, s_order_cnt number
, s_remote_cnt number
, s_data varchar2(50)
, s_dist_01 char(24)
, s_dist_02 char(24)
, s_dist_03 char(24)
, s_dist_04 char(24)
, s_dist_05 char(24)
, s_dist_06 char(24)
, s_dist_07 char(24)
, s_dist_08 char(24)
, s_dist_09 char(24)
, s_dist_10 char(24)
)
cluster stokcluster (
  s_i_id
, s_w_id
);

```

```

set echo off
spool off
exit sql.sqlcode;

```

Createtable_ware.sql

```

/* created automatically by /BUILD/scripts/buildcreatetable.sh Wed Dec 6 08:27:42 PST
2006 */
set timing on
set sqlblanklines on
spool createtable_ware.log
set echo on
drop cluster warecluster including tables ;

```

```

create cluster warecluster (
  w_id number
)
single table
hashkeys 349440
hash is ( (w_id - 1) )
size 1448
intrans 2
storage ( buffer_pool default )
tablespace ware_0;

```

```

create table ware (
  w_id number
, w_ytd number
, w_tax number
, w_name varchar2(10)
, w_street_1 varchar2(20)
, w_street_2 varchar2(20)
, w_city varchar2(20)
, w_state char(2)
, w_zip char(9)
)
cluster warecluster (
  w_id
);
set echo off
spool off
exit sql.sqlcode;

```

Createts.sh

```

#created automatically by /BUILD/scripts/buildcreatets.sh Wed Dec 6 08:36:43 PST 2006

# Tablespace ware, ts size 740M (757760K)
# each file 740M (757760K)
# extents 749524K (749524K)
# 1 files

Stpcc_createts ware 1 1 740M 749524K unix 0 0 128 2K t
if expr $? != 0 > /dev/null; then
echo Creating tablespace for ware failed. Exiting.
exit 0
fi

```

```
# Tablespace cust, ts size 10974600M (11237990400K)
# each file 16380M (16773120K)
# extents 2394948K (2394948K)
# 670 files
```

```
Stpc_createts cust 662 1 16100M 4020M unix 0 1 128 auto t
if expr $? != 0 > /dev/null; then
echo Creating tablespace for cust failed. Exiting.
exit 0
fi
```

```
# Tablespace dist, ts size 7320M (7495680K)
# each file 7320M (7495680K)
# extents 1872276K (1872276K)
# 1 files
```

```
Stpc_createts dist 1 1 7320M 1872276K unix 0 663 128 2K t
if expr $? != 0 > /dev/null; then
echo Creating tablespace for dist failed. Exiting.
exit 0
fi
```

```
# Tablespace hist, ts size 881820M (902983680K)
# each file 16330M (16721920K)
# extents 103164K (103164K)
# 54 files
```

```
Stpc_createts hist 54 1 16383M 321M unix 0 664 128 auto t
if expr $? != 0 > /dev/null; then
echo Creating tablespace for hist failed. Exiting.
exit 0
fi
```

```
# Tablespace stok, ts size 13300560M (13619773440K)
# each file 16380M (16773120K)
# extents 2794356K (2794356K)
# 812 files
```

```
Stpc_createts stok 661 1 15800M 3157M unix 0 718 128 auto t
if expr $? != 0 > /dev/null; then
echo Creating tablespace for stok failed. Exiting.
exit 0
fi
```

```
# Tablespace item, ts size 20M (20480K)
# each file 20M (20480K)
# extents 16892K (16892K)
# 1 files
```

```
Stpc_createts item 1 1 20M 16892K unix 0 1379 128 2K t
if expr $? != 0 > /dev/null; then
echo Creating tablespace for item failed. Exiting.
exit 0
fi
```

```
# Tablespace ordr, ts size 11914200M (12200140800K)
# each file 66190M (6778560K)
# extents 103312K (103312K)
# 180 files
```

```
Stpc_createts ordr 180 1 65535M 3276M unix 0 1380 128 16K t
if expr $? != 0 > /dev/null; then
echo Creating tablespace for ordr failed. Exiting.
exit 0
fi
```

```
# Tablespace nord, ts size 116480M (119275520K)
# each file 14560M (14909440K)
# extents 1490388K (1490388K)
# 8 files
```

```
Stpc_createts nord 16 1 8191M 210M unix 0 1560 128 2K t
if expr $? != 0 > /dev/null; then
echo Creating tablespace for nord failed. Exiting.
exit 0
fi
```

```
# Tablespace iware, ts size 430M (440320K)
# each file 430M (440320K)
# extents 437824K (437824K)
# 1 files
```

```
Stpc_createts iware 1 1 430M 437824K unix 0 1576 128 auto t
if expr $? != 0 > /dev/null; then
echo Creating tablespace for iware failed. Exiting.
exit 0
fi
```

```
# Tablespace icust1, ts size 243040M (248872960K)
# each file 60760M (62218240K)
# extents 60752K (60752K)
# 4 files
```

```
Stpc_createts icust1 4 1 65535M 180M unix 0 1577 128 16K t
```

```
if expr $? != 0 > /dev/null; then
echo Creating tablespace for icust1 failed. Exiting.
exit 0
fi
```

```
# Tablespace icust2, ts size 529440M (542146560K)
# each file 66180M (67768320K)
# extents 66176K (66176K)
# 8 files
```

```
Stpc_createts icust2 8 1 65535M 180M unix 0 1581 128 16K t
if expr $? != 0 > /dev/null; then
echo Creating tablespace for icust2 failed. Exiting.
exit 0
fi
```

```
# Tablespace idist, ts size 1710M (1751040K)
# each file 1710M (1751040K)
# extents 1748224K (1748224K)
# 1 files
```

```
Stpc_createts idist 1 1 1710M 1748224K unix 0 1589 128 auto t
if expr $? != 0 > /dev/null; then
echo Creating tablespace for idist failed. Exiting.
exit 0
fi
```

```
# Tablespace istok, ts size 719180M (736440320K)
# each file 65380M (66949120K)
# extents 65376K (65376K)
# 11 files
```

```
Stpc_createts istok 11 1 65535M 180M unix 0 1590 128 16K t
if expr $? != 0 > /dev/null; then
echo Creating tablespace for istok failed. Exiting.
exit 0
fi
```

```
# Tablespace iitem, ts size 20M (20480K)
# each file 20M (20480K)
# extents 11264K (11264K)
# 1 files
```

```
Stpc_createts iitem 1 1 20M 11264K unix 0 1601 128 auto t
if expr $? != 0 > /dev/null; then
echo Creating tablespace for iitem failed. Exiting.
exit 0
fi
```

```
# Tablespace iordr2, ts size 536040M (548904960K)
# each file 59560M (60989440K)
# extents 59552K (59552K)
# 9 files
```

```
Stpc_createts iordr2 9 1 65535M 260M unix 0 1602 128 16K t
if expr $? != 0 > /dev/null; then
echo Creating tablespace for iordr2 failed. Exiting.
exit 0
fi
```

```
# Tablespace temp, ts size 1778760M (1821450240K)
# each file 16470M (16865280K)
# extents 205624K (205624K)
# 108 files
```

```
Stpc_createts temp 108 1 16383M 260M unix 1 1611 128 auto t
if expr $? != 0 > /dev/null; then
echo Creating tablespace for temp failed. Exiting.
exit 0
fi
```

createuser.sh

```
#!/bin/sh
```

```
echo Creating user tpcc...
Stpc_sqlplus Stpc_dba_user_pass @$tpcc_sql_dir/createuser > junk 2>&1
if test $? -ne 0
then
exit 1;
else
exit 0;
fi
```

Createuser.sql

```
spool createusertpcc.log;
```

```
set echo on;
```

```

create user tpcc identified by tpcc;

grant dba to tpcc;

set echo off;
spool off;

exit ;

```

ddview.sh

```

#!/bin/sh

Stpcc_sqlplus $tpcc_sqlplus_args << !
Stpcc_internal_connect

spool ddview.log

REM
REM In an ade/nde view we might need to run standard.sql and dbmsstdx manually
REM catalog and catproc suppose to take care of it
REM

@${ORACLE_HOME}/plsql/admin/standard
@${ORACLE_HOME}/rdbms/admin/dbmsstdx

@${ORACLE_HOME}/rdbms/admin/catalog
@${ORACLE_HOME}/rdbms/admin/catproc

REM
REM In an ade/nde view we might need to run pupbld manually
REM catalog and catproc suppose to take care of it
REM

connect system/manager
REM @${ORACLE_HOME}/sqlplus/admin/pupbld

REM
REM Oracle
REM

REM if test $NUMBER_ORACLE_NODE -gt 1
REM then

REM @${ORACLE_HOME}/rdbms/admin/catparr

REM fi

spool off
!

#sh $tpcc_scripts/queue.sh

```

dml.sql

```

REM=====
=====+
REM      Copyright (c) 1996 Oracle Corp. Redwood Shores, CA   |
REM      OPEN SYSTEMS PERFORMANCE GROUP                     |
REM      All Rights Reserved                                   |
REM=====
=====+
REM FILENAME
REM      dml.sql
REM DESCRIPTION
REM      Disable table locks for TPC-C tables.
REM USAGE
REM      sqlplus tpcc/tpcc dml.sql
REM=====
=====

connect tpcc/tpcc;
set echo on;

alter table ware disable table lock;
alter table dist disable table lock;
alter table cust disable table lock;
alter table hist disable table lock;
alter table item disable table lock;
alter table stok disable table lock;
alter table ordr disable table lock;
alter table nord disable table lock;
alter table ordl disable table lock;

set echo off;

exit;

```

driver.sh

```

#!/bin/sh

./stepenv.sh

if expr $# \< 1 > /dev/null; then
echo "$0 <starting stepname> <optional: only>"
echo OR use:
echo "$0 buildcreate - to build the database creation scripts"
echo "$0 create - to create the database (after buildcreate)"
echo "$0 steps - to list individual steps"
exit 1
fi

if expr x$1 = xsteps > /dev/null; then
echo stepnames are from creation scripts: $tpcc_create_steps
echo
echo or running steps: $tpcc_steps
echo "use the 'only' option to only do that step (otherwise all steps after will also be
executed.)"
echo " (e.g. $0 listfiles only)"
echo "use the 'through' option to do a sequence of steps (inclusively.)"
echo " (e.g. $0 shutdowndb through startupdb-p_build)"
exit 1
fi

startstep=$1
controlcmd=$2
endstep=$3

# Aliases for special steps
if test $startstep = buildcreate; then
startstep='echo $tpcc_create_steps | cut -d' ' -f1'
fi

if test $startstep = create; then
startstep='echo $tpcc_steps | cut -d' ' -f1'
fi

if test "x$controlcmd" = x; then
# endstep=
# Since endstep is null it won't match any other steps, so we keep going.
# HRT: letting it run to the last step is asking for trouble. So, change
# is so that a num controlcmd is equivalent to "only"
controlcmd=only
elif test "x$controlcmd" = xonly; then
controlcmd=only
# this is allowed
elif test "x$controlcmd" = xthrough; then
actualstep=f
for step in $tpcc_create_steps $tpcc_steps ; do
if test "x$step" = "x$endstep"; then
actualstep=t
fi
done
if test $actualstep = f; then
echo "Invalid step $endstep. Use $0 steps to show steps."
exit 1
fi
else
echo "Invalid syntax. Use $0 by itself for help."
exit 1
fi

echo Starting from step: $startstep

dostep=f
for step in $tpcc_create_steps $tpcc_steps ; do
if expr $step = $startstep > /dev/null; then
dostep=t
fi

if expr $dostep = t > /dev/null; then
echo $step
cd $tpcc_bench
$tpcc_scripts/'echo $step | cut -d- -f1'.sh `echo $step | sed -e's/\/-/' | cut -d- -f2- | sed -
e's/\/-/'g`
lasterror=$?
cd $tpcc_bench
if test -n "`find $tpcc_bench/scripts -name '*.log'`; then
mv -f *.log `find $tpcc_bench/scripts -name '*.log' $tpcc_bench/log/`
else
mv -f *.log $tpcc_bench/log/
fi

if expr $lasterror != 0 > /dev/null; then
if expr $lasterror != 99 > /dev/null; then
echo Step $step failed. Stopping driver.
exit 1
else
echo Step $step has completed and requested stop. Stopping driver.
exit 0
fi

```

```

fi
if test "x$controlcmd" = xonly; then
  exit 0
fi
if test "x$endstep" = "x$step"; then
  echo The driver reached the last desired step. Stopping driver.
  exit 0
fi
fi
done

if expr $dostep = f > /dev/null; then
  echo No such step: $1
fi

```

extent.sql

```

REM      Copyright (c) 1994 Oracle Corp. Belmont, CA      |
REM      OPEN SYSTEMS PERFORMANCE GROUP                  |
REM      All Rights Reserved                               |
REM=====
=====+
REM FILENAME
REM extent.sql
REM DESCRIPTION
REM List all extents in all the TPCC tablespaces.
REM
REM Usage: sqlplus 'sys/change_on_install as sysdba' @extent
REM=====
=====*/
set space 2
set pagesize 2000
set echo off
set termout off
set verify off
set feedback off
spool extent.rpt
select substr(e.tablespace_name,1,8) tspace,
       substr(segment_name,1,11) segment, substr(segment_type,1,15) type,
       substr(extent_id,1,5) eid, substr(file_id,1,5) fid, blocks,
       blocks * t.block_size / 1048576 size_MB
from dba_extents e, dba_tablespaces t
where owner = 'TPCC' AND ( segment_type = 'INDEX' OR
segment_type = 'INDEX PARTITION' OR segment_type = 'CLUSTER'
OR segment_type = 'TABLE' OR segment_type = 'TABLE PARTITION')
AND e.tablespace_name <> 'SYSTEM'
AND e.tablespace_name = t.tablespace_name
order by e.tablespace_name, segment_name, extent_id, file_id;

select substr(e.tablespace_name,1,8) tspace,
       substr(segment_name,1,11) segment,
       sum(blocks) tot_blk, sum(blocks) * t.block_size / 1048576 size_MB
from dba_extents e, dba_tablespaces t
where owner = 'TPCC' AND ( segment_type = 'INDEX' OR
segment_type = 'INDEX PARTITION' OR segment_type = 'CLUSTER'
OR segment_type = 'TABLE' OR segment_type = 'TABLE PARTITION')
AND e.tablespace_name <> 'SYSTEM'
AND e.tablespace_name = t.tablespace_name
group by e.tablespace_name, segment_name, t.block_size
order by e.tablespace_name, segment_name;
spool off;

```

freext.sql

```

REM=====
=====+
REM      Copyright (c) 1994 Oracle Corp. Belmont, CA      |
REM      OPEN SYSTEMS PERFORMANCE GROUP                  |
REM      All Rights Reserved                               |
REM=====
=====+
REM FILENAME
REM freext.sql
REM DESCRIPTION
REM List all free extents in all the TPCC tablespaces
REM
REM Usage: sqlplus 'sys/change_on_install as sysdba' @freext
REM=====
=====*/
set space 2
set pagesize 2000
set echo off
set termout off
set verify off
set feedback off

```

```

spool freextent.rpt
select substr(e.tablespace_name,1,8) tspace, file_id, block_id, blocks,
       blocks * t.block_size / 1048576 size_MB
from dba_free_space e, dba_tablespaces t
where e.tablespace_name = t.tablespace_name
order by e.tablespace_name, file_id, block_id;

select substr(e.tablespace_name,1,8) tspace, sum(blocks) tot_blk,
       sum(blocks) * t.block_size / 1048576 size_MB
from dba_free_space e, dba_tablespaces t
where e.tablespace_name = t.tablespace_name
group by e.tablespace_name, t.block_size
order by e.tablespace_name;

```

initpay.sql

```

CREATE OR REPLACE PACKAGE initpay
AS
  TYPE rowidarray IS TABLE OF ROWID INDEX BY BINARY_INTEGER;
  row_id          rowidarray;
  cust_rowid      ROWID;
  dist_name       VARCHAR2(11);
  ware_name       VARCHAR2(11);
  c_num           BINARY_INTEGER;
  PROCEDURE pay_init;
END initpay;
/

CREATE OR REPLACE PACKAGE BODY initpay AS
  PROCEDURE pay_init IS
  BEGIN
    NULL;
  END pay_init;
END initpay;
/

exit;

```

loadcust.sh

```

#created automatically by /BUILD/scripts/evenload.sh Fri Dec 8 17:07:20 PST 2006
rm -f loadcust*.log
cd $tpcc_bench
allprocs=
$tpcc_load -M 349440 -C -1 -1 -m 23 >> loadcust0.log 2>&1 &
allprocs="$allprocs $(!)"
$tpcc_load -M 349440 -C -1 24 -m 46 >> loadcust1.log 2>&1 &
allprocs="$allprocs $(!)"
$tpcc_load -M 349440 -C -1 47 -m 69 >> loadcust2.log 2>&1 &
allprocs="$allprocs $(!)"
$tpcc_load -M 349440 -C -1 70 -m 92 >> loadcust3.log 2>&1 &
allprocs="$allprocs $(!)"
$tpcc_load -M 349440 -C -1 93 -m 115 >> loadcust4.log 2>&1 &
allprocs="$allprocs $(!)"
$tpcc_load -M 349440 -C -1 116 -m 138 >> loadcust5.log 2>&1 &
allprocs="$allprocs $(!)"
$tpcc_load -M 349440 -C -1 139 -m 161 >> loadcust6.log 2>&1 &
allprocs="$allprocs $(!)"
$tpcc_load -M 349440 -C -1 162 -m 184 >> loadcust7.log 2>&1 &
allprocs="$allprocs $(!)"
$tpcc_load -M 349440 -C -1 185 -m 207 >> loadcust8.log 2>&1 &
allprocs="$allprocs $(!)"
$tpcc_load -M 349440 -C -1 208 -m 230 >> loadcust9.log 2>&1 &
allprocs="$allprocs $(!)"
$tpcc_load -M 349440 -C -1 231 -m 253 >> loadcust10.log 2>&1 &
allprocs="$allprocs $(!)"
$tpcc_load -M 349440 -C -1 254 -m 276 >> loadcust11.log 2>&1 &
allprocs="$allprocs $(!)"
$tpcc_load -M 349440 -C -1 277 -m 299 >> loadcust12.log 2>&1 &
allprocs="$allprocs $(!)"
$tpcc_load -M 349440 -C -1 300 -m 322 >> loadcust13.log 2>&1 &
allprocs="$allprocs $(!)"
$tpcc_load -M 349440 -C -1 323 -m 345 >> loadcust14.log 2>&1 &
allprocs="$allprocs $(!)"
$tpcc_load -M 349440 -C -1 346 -m 368 >> loadcust15.log 2>&1 &
allprocs="$allprocs $(!)"
$tpcc_load -M 349440 -C -1 369 -m 391 >> loadcust16.log 2>&1 &
allprocs="$allprocs $(!)"
$tpcc_load -M 349440 -C -1 392 -m 414 >> loadcust17.log 2>&1 &
allprocs="$allprocs $(!)"
$tpcc_load -M 349440 -C -1 415 -m 437 >> loadcust18.log 2>&1 &
allprocs="$allprocs $(!)"
$tpcc_load -M 349440 -C -1 438 -m 460 >> loadcust19.log 2>&1 &
allprocs="$allprocs $(!)"
$tpcc_load -M 349440 -C -1 461 -m 483 >> loadcust20.log 2>&1 &
allprocs="$allprocs $(!)"
$tpcc_load -M 349440 -C -1 484 -m 506 >> loadcust21.log 2>&1 &
allprocs="$allprocs $(!)"
$tpcc_load -M 349440 -C -1 507 -m 529 >> loadcust22.log 2>&1 &

```



```

allprocs="$allprocs ${!}"
Stpc_load -M 349440 -C -1 2761 -m 2784 >> loadcust118.log 2>&1 &
allprocs="$allprocs ${!}"
Stpc_load -M 349440 -C -1 2785 -m 2808 >> loadcust119.log 2>&1 &
allprocs="$allprocs ${!}"
Stpc_load -M 349440 -C -1 2809 -m 2832 >> loadcust120.log 2>&1 &
allprocs="$allprocs ${!}"
Stpc_load -M 349440 -C -1 2833 -m 2856 >> loadcust121.log 2>&1 &
allprocs="$allprocs ${!}"
Stpc_load -M 349440 -C -1 2857 -m 2880 >> loadcust122.log 2>&1 &
allprocs="$allprocs ${!}"
Stpc_load -M 349440 -C -1 2881 -m 2904 >> loadcust123.log 2>&1 &
allprocs="$allprocs ${!}"
Stpc_load -M 349440 -C -1 2905 -m 2928 >> loadcust124.log 2>&1 &
allprocs="$allprocs ${!}"
Stpc_load -M 349440 -C -1 2929 -m 2952 >> loadcust125.log 2>&1 &
allprocs="$allprocs ${!}"
Stpc_load -M 349440 -C -1 2953 -m 2976 >> loadcust126.log 2>&1 &
allprocs="$allprocs ${!}"
Stpc_load -M 349440 -C -1 2977 -m 3000 >> loadcust127.log 2>&1 &
allprocs="$allprocs ${!}"
error=0
for curproc in $allprocs; do
    wait $curproc
    error=$((error + $?)
done
exit `expr $error != 0`

```

loaddist.sh

```

cd $stpc_bench
Stpc_load -M $stpc_scale -d > loaddist.log 2>&1

```

loadhist.sh

```

#created automatically by /BUILD/scripts/eventload.sh Wed Dec 6 08:28:15 PST 2006
rm -f loadhist*.log
cd $stpc_bench
allprocs=
Stpc_load -M 349440 -h -b 1 -e 1365 >> loadhist0.log 2>&1 &
allprocs="$allprocs ${!}"
Stpc_load -M 349440 -h -b 1366 -e 2730 >> loadhist1.log 2>&1 &
allprocs="$allprocs ${!}"
Stpc_load -M 349440 -h -b 2731 -e 4095 >> loadhist2.log 2>&1 &
allprocs="$allprocs ${!}"
Stpc_load -M 349440 -h -b 4096 -e 5460 >> loadhist3.log 2>&1 &
allprocs="$allprocs ${!}"
Stpc_load -M 349440 -h -b 5461 -e 6825 >> loadhist4.log 2>&1 &
allprocs="$allprocs ${!}"
Stpc_load -M 349440 -h -b 6826 -e 8190 >> loadhist5.log 2>&1 &
allprocs="$allprocs ${!}"
Stpc_load -M 349440 -h -b 8191 -e 9555 >> loadhist6.log 2>&1 &
allprocs="$allprocs ${!}"
Stpc_load -M 349440 -h -b 9556 -e 10920 >> loadhist7.log 2>&1 &
allprocs="$allprocs ${!}"
Stpc_load -M 349440 -h -b 10921 -e 12285 >> loadhist8.log 2>&1 &
allprocs="$allprocs ${!}"
Stpc_load -M 349440 -h -b 12286 -e 13650 >> loadhist9.log 2>&1 &
allprocs="$allprocs ${!}"
Stpc_load -M 349440 -h -b 13651 -e 15015 >> loadhist10.log 2>&1 &
allprocs="$allprocs ${!}"
Stpc_load -M 349440 -h -b 15016 -e 16380 >> loadhist11.log 2>&1 &
allprocs="$allprocs ${!}"
Stpc_load -M 349440 -h -b 16381 -e 17745 >> loadhist12.log 2>&1 &
allprocs="$allprocs ${!}"
Stpc_load -M 349440 -h -b 17746 -e 19110 >> loadhist13.log 2>&1 &
allprocs="$allprocs ${!}"
Stpc_load -M 349440 -h -b 19111 -e 20475 >> loadhist14.log 2>&1 &
allprocs="$allprocs ${!}"
Stpc_load -M 349440 -h -b 20476 -e 21840 >> loadhist15.log 2>&1 &
allprocs="$allprocs ${!}"
Stpc_load -M 349440 -h -b 21841 -e 23205 >> loadhist16.log 2>&1 &
allprocs="$allprocs ${!}"
Stpc_load -M 349440 -h -b 23206 -e 24570 >> loadhist17.log 2>&1 &
allprocs="$allprocs ${!}"
Stpc_load -M 349440 -h -b 24571 -e 25935 >> loadhist18.log 2>&1 &
allprocs="$allprocs ${!}"
Stpc_load -M 349440 -h -b 25936 -e 27300 >> loadhist19.log 2>&1 &
allprocs="$allprocs ${!}"
Stpc_load -M 349440 -h -b 27301 -e 28665 >> loadhist20.log 2>&1 &
allprocs="$allprocs ${!}"
Stpc_load -M 349440 -h -b 28666 -e 30030 >> loadhist21.log 2>&1 &
allprocs="$allprocs ${!}"
Stpc_load -M 349440 -h -b 30031 -e 31395 >> loadhist22.log 2>&1 &
allprocs="$allprocs ${!}"
Stpc_load -M 349440 -h -b 31396 -e 32760 >> loadhist23.log 2>&1 &
allprocs="$allprocs ${!}"
Stpc_load -M 349440 -h -b 32761 -e 34125 >> loadhist24.log 2>&1 &
allprocs="$allprocs ${!}"

```

```

Stpc_load -M 349440 -h -b 34126 -e 35490 >> loadhist25.log 2>&1 &
allprocs="$allprocs ${!}"
Stpc_load -M 349440 -h -b 35491 -e 36855 >> loadhist26.log 2>&1 &
allprocs="$allprocs ${!}"
Stpc_load -M 349440 -h -b 36856 -e 38220 >> loadhist27.log 2>&1 &
allprocs="$allprocs ${!}"
Stpc_load -M 349440 -h -b 38221 -e 39585 >> loadhist28.log 2>&1 &
allprocs="$allprocs ${!}"
Stpc_load -M 349440 -h -b 39586 -e 40950 >> loadhist29.log 2>&1 &
allprocs="$allprocs ${!}"
Stpc_load -M 349440 -h -b 40951 -e 42315 >> loadhist30.log 2>&1 &
allprocs="$allprocs ${!}"
Stpc_load -M 349440 -h -b 42316 -e 43680 >> loadhist31.log 2>&1 &
allprocs="$allprocs ${!}"
Stpc_load -M 349440 -h -b 43681 -e 45045 >> loadhist32.log 2>&1 &
allprocs="$allprocs ${!}"
Stpc_load -M 349440 -h -b 45046 -e 46410 >> loadhist33.log 2>&1 &
allprocs="$allprocs ${!}"
Stpc_load -M 349440 -h -b 46411 -e 47775 >> loadhist34.log 2>&1 &
allprocs="$allprocs ${!}"
Stpc_load -M 349440 -h -b 47776 -e 49140 >> loadhist35.log 2>&1 &
allprocs="$allprocs ${!}"
Stpc_load -M 349440 -h -b 49141 -e 50505 >> loadhist36.log 2>&1 &
allprocs="$allprocs ${!}"
Stpc_load -M 349440 -h -b 50506 -e 51870 >> loadhist37.log 2>&1 &
allprocs="$allprocs ${!}"
Stpc_load -M 349440 -h -b 51871 -e 53235 >> loadhist38.log 2>&1 &
allprocs="$allprocs ${!}"
Stpc_load -M 349440 -h -b 53236 -e 54600 >> loadhist39.log 2>&1 &
allprocs="$allprocs ${!}"
Stpc_load -M 349440 -h -b 54601 -e 55965 >> loadhist40.log 2>&1 &
allprocs="$allprocs ${!}"
Stpc_load -M 349440 -h -b 55966 -e 57330 >> loadhist41.log 2>&1 &
allprocs="$allprocs ${!}"
Stpc_load -M 349440 -h -b 57331 -e 58695 >> loadhist42.log 2>&1 &
allprocs="$allprocs ${!}"
Stpc_load -M 349440 -h -b 58696 -e 60060 >> loadhist43.log 2>&1 &
allprocs="$allprocs ${!}"
Stpc_load -M 349440 -h -b 60061 -e 61425 >> loadhist44.log 2>&1 &
allprocs="$allprocs ${!}"
Stpc_load -M 349440 -h -b 61426 -e 62790 >> loadhist45.log 2>&1 &
allprocs="$allprocs ${!}"
Stpc_load -M 349440 -h -b 62791 -e 64155 >> loadhist46.log 2>&1 &
allprocs="$allprocs ${!}"
Stpc_load -M 349440 -h -b 64156 -e 65520 >> loadhist47.log 2>&1 &
allprocs="$allprocs ${!}"
Stpc_load -M 349440 -h -b 65521 -e 66885 >> loadhist48.log 2>&1 &
allprocs="$allprocs ${!}"
Stpc_load -M 349440 -h -b 66886 -e 68250 >> loadhist49.log 2>&1 &
allprocs="$allprocs ${!}"
Stpc_load -M 349440 -h -b 68251 -e 69615 >> loadhist50.log 2>&1 &
allprocs="$allprocs ${!}"
Stpc_load -M 349440 -h -b 69616 -e 70980 >> loadhist51.log 2>&1 &
allprocs="$allprocs ${!}"
Stpc_load -M 349440 -h -b 70981 -e 72345 >> loadhist52.log 2>&1 &
allprocs="$allprocs ${!}"
Stpc_load -M 349440 -h -b 72346 -e 73710 >> loadhist53.log 2>&1 &
allprocs="$allprocs ${!}"
Stpc_load -M 349440 -h -b 73711 -e 75075 >> loadhist54.log 2>&1 &
allprocs="$allprocs ${!}"
Stpc_load -M 349440 -h -b 75076 -e 76440 >> loadhist55.log 2>&1 &
allprocs="$allprocs ${!}"
Stpc_load -M 349440 -h -b 76441 -e 77805 >> loadhist56.log 2>&1 &
allprocs="$allprocs ${!}"
Stpc_load -M 349440 -h -b 77806 -e 79170 >> loadhist57.log 2>&1 &
allprocs="$allprocs ${!}"
Stpc_load -M 349440 -h -b 79171 -e 80535 >> loadhist58.log 2>&1 &
allprocs="$allprocs ${!}"
Stpc_load -M 349440 -h -b 80536 -e 81900 >> loadhist59.log 2>&1 &
allprocs="$allprocs ${!}"
Stpc_load -M 349440 -h -b 81901 -e 83265 >> loadhist60.log 2>&1 &
allprocs="$allprocs ${!}"
Stpc_load -M 349440 -h -b 83266 -e 84630 >> loadhist61.log 2>&1 &
allprocs="$allprocs ${!}"
Stpc_load -M 349440 -h -b 84631 -e 85995 >> loadhist62.log 2>&1 &
allprocs="$allprocs ${!}"
Stpc_load -M 349440 -h -b 85996 -e 87360 >> loadhist63.log 2>&1 &
allprocs="$allprocs ${!}"
Stpc_load -M 349440 -h -b 87361 -e 88725 >> loadhist64.log 2>&1 &
allprocs="$allprocs ${!}"
Stpc_load -M 349440 -h -b 88726 -e 90090 >> loadhist65.log 2>&1 &
allprocs="$allprocs ${!}"
Stpc_load -M 349440 -h -b 90091 -e 91455 >> loadhist66.log 2>&1 &
allprocs="$allprocs ${!}"
Stpc_load -M 349440 -h -b 91456 -e 92820 >> loadhist67.log 2>&1 &
allprocs="$allprocs ${!}"
Stpc_load -M 349440 -h -b 92821 -e 94185 >> loadhist68.log 2>&1 &
allprocs="$allprocs ${!}"
Stpc_load -M 349440 -h -b 94186 -e 95550 >> loadhist69.log 2>&1 &
allprocs="$allprocs ${!}"
Stpc_load -M 349440 -h -b 95551 -e 96915 >> loadhist70.log 2>&1 &
allprocs="$allprocs ${!}"
Stpc_load -M 349440 -h -b 96916 -e 98280 >> loadhist71.log 2>&1 &
allprocs="$allprocs ${!}"
Stpc_load -M 349440 -h -b 98281 -e 99645 >> loadhist72.log 2>&1 &

```



```

allprocs="$Sallprocs ${!}"
Stpcc_load -M 349440 -n -b 248431 -e 251160 >> loadnord91.log 2>&1 &
allprocs="$Sallprocs ${!}"
Stpcc_load -M 349440 -n -b 251161 -e 253890 >> loadnord92.log 2>&1 &
allprocs="$Sallprocs ${!}"
Stpcc_load -M 349440 -n -b 253891 -e 256620 >> loadnord93.log 2>&1 &
allprocs="$Sallprocs ${!}"
Stpcc_load -M 349440 -n -b 256621 -e 259350 >> loadnord94.log 2>&1 &
allprocs="$Sallprocs ${!}"
Stpcc_load -M 349440 -n -b 259351 -e 262080 >> loadnord95.log 2>&1 &
allprocs="$Sallprocs ${!}"
Stpcc_load -M 349440 -n -b 262081 -e 264810 >> loadnord96.log 2>&1 &
allprocs="$Sallprocs ${!}"
Stpcc_load -M 349440 -n -b 264811 -e 267540 >> loadnord97.log 2>&1 &
allprocs="$Sallprocs ${!}"
Stpcc_load -M 349440 -n -b 267541 -e 270270 >> loadnord98.log 2>&1 &
allprocs="$Sallprocs ${!}"
Stpcc_load -M 349440 -n -b 270271 -e 273000 >> loadnord99.log 2>&1 &
allprocs="$Sallprocs ${!}"
Stpcc_load -M 349440 -n -b 273001 -e 275730 >> loadnord100.log 2>&1 &
allprocs="$Sallprocs ${!}"
Stpcc_load -M 349440 -n -b 275731 -e 278460 >> loadnord101.log 2>&1 &
allprocs="$Sallprocs ${!}"
Stpcc_load -M 349440 -n -b 278461 -e 281190 >> loadnord102.log 2>&1 &
allprocs="$Sallprocs ${!}"
Stpcc_load -M 349440 -n -b 281191 -e 283920 >> loadnord103.log 2>&1 &
allprocs="$Sallprocs ${!}"
Stpcc_load -M 349440 -n -b 283921 -e 286650 >> loadnord104.log 2>&1 &
allprocs="$Sallprocs ${!}"
Stpcc_load -M 349440 -n -b 286651 -e 289380 >> loadnord105.log 2>&1 &
allprocs="$Sallprocs ${!}"
Stpcc_load -M 349440 -n -b 289381 -e 292110 >> loadnord106.log 2>&1 &
allprocs="$Sallprocs ${!}"
Stpcc_load -M 349440 -n -b 292111 -e 294840 >> loadnord107.log 2>&1 &
allprocs="$Sallprocs ${!}"
Stpcc_load -M 349440 -n -b 294841 -e 297570 >> loadnord108.log 2>&1 &
allprocs="$Sallprocs ${!}"
Stpcc_load -M 349440 -n -b 297571 -e 300300 >> loadnord109.log 2>&1 &
allprocs="$Sallprocs ${!}"
Stpcc_load -M 349440 -n -b 300301 -e 303030 >> loadnord110.log 2>&1 &
allprocs="$Sallprocs ${!}"
Stpcc_load -M 349440 -n -b 303031 -e 305760 >> loadnord111.log 2>&1 &
allprocs="$Sallprocs ${!}"
Stpcc_load -M 349440 -n -b 305761 -e 308490 >> loadnord112.log 2>&1 &
allprocs="$Sallprocs ${!}"
Stpcc_load -M 349440 -n -b 308491 -e 311220 >> loadnord113.log 2>&1 &
allprocs="$Sallprocs ${!}"
Stpcc_load -M 349440 -n -b 311221 -e 313950 >> loadnord114.log 2>&1 &
allprocs="$Sallprocs ${!}"
Stpcc_load -M 349440 -n -b 313951 -e 316680 >> loadnord115.log 2>&1 &
allprocs="$Sallprocs ${!}"
Stpcc_load -M 349440 -n -b 316681 -e 319410 >> loadnord116.log 2>&1 &
allprocs="$Sallprocs ${!}"
Stpcc_load -M 349440 -n -b 319411 -e 322140 >> loadnord117.log 2>&1 &
allprocs="$Sallprocs ${!}"
Stpcc_load -M 349440 -n -b 322141 -e 324870 >> loadnord118.log 2>&1 &
allprocs="$Sallprocs ${!}"
Stpcc_load -M 349440 -n -b 324871 -e 327600 >> loadnord119.log 2>&1 &
allprocs="$Sallprocs ${!}"
Stpcc_load -M 349440 -n -b 327601 -e 330330 >> loadnord120.log 2>&1 &
allprocs="$Sallprocs ${!}"
Stpcc_load -M 349440 -n -b 330331 -e 333060 >> loadnord121.log 2>&1 &
allprocs="$Sallprocs ${!}"
Stpcc_load -M 349440 -n -b 333061 -e 335790 >> loadnord122.log 2>&1 &
allprocs="$Sallprocs ${!}"
Stpcc_load -M 349440 -n -b 335791 -e 338520 >> loadnord123.log 2>&1 &
allprocs="$Sallprocs ${!}"
Stpcc_load -M 349440 -n -b 338521 -e 341250 >> loadnord124.log 2>&1 &
allprocs="$Sallprocs ${!}"
Stpcc_load -M 349440 -n -b 341251 -e 343980 >> loadnord125.log 2>&1 &
allprocs="$Sallprocs ${!}"
Stpcc_load -M 349440 -n -b 343981 -e 346710 >> loadnord126.log 2>&1 &
allprocs="$Sallprocs ${!}"
Stpcc_load -M 349440 -n -b 346711 -e 349440 >> loadnord127.log 2>&1 &
allprocs="$Sallprocs ${!}"
error=0
for curproc in $Sallprocs; do
  wait $curproc
  error=`expr $? + $error`
done
exit `expr $error != 0`

```

loadordrordl.sh

```

#created automatically by /BUILD/scripts/evenload.sh Fri Dec 8 17:06:57 PST 2006
rm -f loadordrordl*.log
cd $Stpcc_bench
allprocs=
Stpcc_load -M 349440 -o ${tpcc_disks_location}dummy0.dat -b 1 -e 2730 >>
loadordrordl0.log 2>&1 &
allprocs="$Sallprocs ${!}"
Stpcc_load -M 349440 -o ${tpcc_disks_location}dummy1.dat -b 2731 -e 5460 >>
loadordrordl1.log 2>&1 &

```

```

allprocs="$Sallprocs ${!}"
Stpcc_load -M 349440 -o ${tpcc_disks_location}dummy2.dat -b 5461 -e 8190 >>
loadordrordl2.log 2>&1 &
allprocs="$Sallprocs ${!}"
Stpcc_load -M 349440 -o ${tpcc_disks_location}dummy3.dat -b 8191 -e 10920 >>
loadordrordl3.log 2>&1 &
allprocs="$Sallprocs ${!}"
Stpcc_load -M 349440 -o ${tpcc_disks_location}dummy4.dat -b 10921 -e 13650 >>
loadordrordl4.log 2>&1 &
allprocs="$Sallprocs ${!}"
Stpcc_load -M 349440 -o ${tpcc_disks_location}dummy5.dat -b 13651 -e 16380 >>
loadordrordl5.log 2>&1 &
allprocs="$Sallprocs ${!}"
Stpcc_load -M 349440 -o ${tpcc_disks_location}dummy6.dat -b 16381 -e 19110 >>
loadordrordl6.log 2>&1 &
allprocs="$Sallprocs ${!}"
Stpcc_load -M 349440 -o ${tpcc_disks_location}dummy7.dat -b 19111 -e 21840 >>
loadordrordl7.log 2>&1 &
allprocs="$Sallprocs ${!}"
Stpcc_load -M 349440 -o ${tpcc_disks_location}dummy8.dat -b 21841 -e 24570 >>
loadordrordl8.log 2>&1 &
allprocs="$Sallprocs ${!}"
Stpcc_load -M 349440 -o ${tpcc_disks_location}dummy9.dat -b 24571 -e 27300 >>
loadordrordl9.log 2>&1 &
allprocs="$Sallprocs ${!}"
Stpcc_load -M 349440 -o ${tpcc_disks_location}dummy10.dat -b 27301 -e 30030 >>
loadordrordl10.log 2>&1 &
allprocs="$Sallprocs ${!}"
Stpcc_load -M 349440 -o ${tpcc_disks_location}dummy11.dat -b 30031 -e 32760 >>
loadordrordl11.log 2>&1 &
allprocs="$Sallprocs ${!}"
Stpcc_load -M 349440 -o ${tpcc_disks_location}dummy12.dat -b 32761 -e 35490 >>
loadordrordl12.log 2>&1 &
allprocs="$Sallprocs ${!}"
Stpcc_load -M 349440 -o ${tpcc_disks_location}dummy13.dat -b 35491 -e 38220 >>
loadordrordl13.log 2>&1 &
allprocs="$Sallprocs ${!}"
Stpcc_load -M 349440 -o ${tpcc_disks_location}dummy14.dat -b 38221 -e 40950 >>
loadordrordl14.log 2>&1 &
allprocs="$Sallprocs ${!}"
Stpcc_load -M 349440 -o ${tpcc_disks_location}dummy15.dat -b 40951 -e 43680 >>
loadordrordl15.log 2>&1 &
allprocs="$Sallprocs ${!}"
Stpcc_load -M 349440 -o ${tpcc_disks_location}dummy16.dat -b 43681 -e 46410 >>
loadordrordl16.log 2>&1 &
allprocs="$Sallprocs ${!}"
Stpcc_load -M 349440 -o ${tpcc_disks_location}dummy17.dat -b 46411 -e 49140 >>
loadordrordl17.log 2>&1 &
allprocs="$Sallprocs ${!}"
Stpcc_load -M 349440 -o ${tpcc_disks_location}dummy18.dat -b 49141 -e 51870 >>
loadordrordl18.log 2>&1 &
allprocs="$Sallprocs ${!}"
Stpcc_load -M 349440 -o ${tpcc_disks_location}dummy19.dat -b 51871 -e 54600 >>
loadordrordl19.log 2>&1 &
allprocs="$Sallprocs ${!}"
Stpcc_load -M 349440 -o ${tpcc_disks_location}dummy20.dat -b 54601 -e 57330 >>
loadordrordl20.log 2>&1 &
allprocs="$Sallprocs ${!}"
Stpcc_load -M 349440 -o ${tpcc_disks_location}dummy21.dat -b 57331 -e 60060 >>
loadordrordl21.log 2>&1 &
allprocs="$Sallprocs ${!}"
Stpcc_load -M 349440 -o ${tpcc_disks_location}dummy22.dat -b 60061 -e 62790 >>
loadordrordl22.log 2>&1 &
allprocs="$Sallprocs ${!}"
Stpcc_load -M 349440 -o ${tpcc_disks_location}dummy23.dat -b 62791 -e 65520 >>
loadordrordl23.log 2>&1 &
allprocs="$Sallprocs ${!}"
Stpcc_load -M 349440 -o ${tpcc_disks_location}dummy24.dat -b 65521 -e 68250 >>
loadordrordl24.log 2>&1 &
allprocs="$Sallprocs ${!}"
Stpcc_load -M 349440 -o ${tpcc_disks_location}dummy25.dat -b 68251 -e 70980 >>
loadordrordl25.log 2>&1 &
allprocs="$Sallprocs ${!}"
Stpcc_load -M 349440 -o ${tpcc_disks_location}dummy26.dat -b 70981 -e 73710 >>
loadordrordl26.log 2>&1 &
allprocs="$Sallprocs ${!}"
Stpcc_load -M 349440 -o ${tpcc_disks_location}dummy27.dat -b 73711 -e 76440 >>
loadordrordl27.log 2>&1 &
allprocs="$Sallprocs ${!}"
Stpcc_load -M 349440 -o ${tpcc_disks_location}dummy28.dat -b 76441 -e 79170 >>
loadordrordl28.log 2>&1 &
allprocs="$Sallprocs ${!}"
Stpcc_load -M 349440 -o ${tpcc_disks_location}dummy29.dat -b 79171 -e 81900 >>
loadordrordl29.log 2>&1 &
allprocs="$Sallprocs ${!}"
Stpcc_load -M 349440 -o ${tpcc_disks_location}dummy30.dat -b 81901 -e 84630 >>
loadordrordl30.log 2>&1 &
allprocs="$Sallprocs ${!}"
Stpcc_load -M 349440 -o ${tpcc_disks_location}dummy31.dat -b 84631 -e 87360 >>
loadordrordl31.log 2>&1 &
allprocs="$Sallprocs ${!}"
Stpcc_load -M 349440 -o ${tpcc_disks_location}dummy32.dat -b 87361 -e 90090 >>
loadordrordl32.log 2>&1 &
allprocs="$Sallprocs ${!}"

```



```

Stpcc_load -M 349440 -o ${tpcc_disks_location}dummy96.dat -b 262081 -e 264810 >>
loadordrordl96.log 2>&1 &
allprocs="Sallprocs ${!}"
Stpcc_load -M 349440 -o ${tpcc_disks_location}dummy97.dat -b 264811 -e 267540 >>
loadordrordl97.log 2>&1 &
allprocs="Sallprocs ${!}"
Stpcc_load -M 349440 -o ${tpcc_disks_location}dummy98.dat -b 267541 -e 270270 >>
loadordrordl98.log 2>&1 &
allprocs="Sallprocs ${!}"
Stpcc_load -M 349440 -o ${tpcc_disks_location}dummy99.dat -b 270271 -e 273000 >>
loadordrordl99.log 2>&1 &
allprocs="Sallprocs ${!}"
Stpcc_load -M 349440 -o ${tpcc_disks_location}dummy100.dat -b 273001 -e 275730 >>
loadordrordl100.log 2>&1 &
allprocs="Sallprocs ${!}"
Stpcc_load -M 349440 -o ${tpcc_disks_location}dummy101.dat -b 275731 -e 278460 >>
loadordrordl101.log 2>&1 &
allprocs="Sallprocs ${!}"
Stpcc_load -M 349440 -o ${tpcc_disks_location}dummy102.dat -b 278461 -e 281190 >>
loadordrordl102.log 2>&1 &
allprocs="Sallprocs ${!}"
Stpcc_load -M 349440 -o ${tpcc_disks_location}dummy103.dat -b 281191 -e 283920 >>
loadordrordl103.log 2>&1 &
allprocs="Sallprocs ${!}"
Stpcc_load -M 349440 -o ${tpcc_disks_location}dummy104.dat -b 283921 -e 286650 >>
loadordrordl104.log 2>&1 &
allprocs="Sallprocs ${!}"
Stpcc_load -M 349440 -o ${tpcc_disks_location}dummy105.dat -b 286651 -e 289380 >>
loadordrordl105.log 2>&1 &
allprocs="Sallprocs ${!}"
Stpcc_load -M 349440 -o ${tpcc_disks_location}dummy106.dat -b 289381 -e 292110 >>
loadordrordl106.log 2>&1 &
allprocs="Sallprocs ${!}"
Stpcc_load -M 349440 -o ${tpcc_disks_location}dummy107.dat -b 292111 -e 294840 >>
loadordrordl107.log 2>&1 &
allprocs="Sallprocs ${!}"
Stpcc_load -M 349440 -o ${tpcc_disks_location}dummy108.dat -b 294841 -e 297570 >>
loadordrordl108.log 2>&1 &
allprocs="Sallprocs ${!}"
Stpcc_load -M 349440 -o ${tpcc_disks_location}dummy109.dat -b 297571 -e 300300 >>
loadordrordl109.log 2>&1 &
allprocs="Sallprocs ${!}"
Stpcc_load -M 349440 -o ${tpcc_disks_location}dummy110.dat -b 300301 -e 303030 >>
loadordrordl110.log 2>&1 &
allprocs="Sallprocs ${!}"
Stpcc_load -M 349440 -o ${tpcc_disks_location}dummy111.dat -b 303031 -e 305760 >>
loadordrordl111.log 2>&1 &
allprocs="Sallprocs ${!}"
Stpcc_load -M 349440 -o ${tpcc_disks_location}dummy112.dat -b 305761 -e 308490 >>
loadordrordl112.log 2>&1 &
allprocs="Sallprocs ${!}"
Stpcc_load -M 349440 -o ${tpcc_disks_location}dummy113.dat -b 308491 -e 311220 >>
loadordrordl113.log 2>&1 &
allprocs="Sallprocs ${!}"
Stpcc_load -M 349440 -o ${tpcc_disks_location}dummy114.dat -b 311221 -e 313950 >>
loadordrordl114.log 2>&1 &
allprocs="Sallprocs ${!}"
Stpcc_load -M 349440 -o ${tpcc_disks_location}dummy115.dat -b 313951 -e 316680 >>
loadordrordl115.log 2>&1 &
allprocs="Sallprocs ${!}"
Stpcc_load -M 349440 -o ${tpcc_disks_location}dummy116.dat -b 316681 -e 319410 >>
loadordrordl116.log 2>&1 &
allprocs="Sallprocs ${!}"
Stpcc_load -M 349440 -o ${tpcc_disks_location}dummy117.dat -b 319411 -e 322140 >>
loadordrordl117.log 2>&1 &
allprocs="Sallprocs ${!}"
Stpcc_load -M 349440 -o ${tpcc_disks_location}dummy118.dat -b 322141 -e 324870 >>
loadordrordl118.log 2>&1 &
allprocs="Sallprocs ${!}"
Stpcc_load -M 349440 -o ${tpcc_disks_location}dummy119.dat -b 324871 -e 327600 >>
loadordrordl119.log 2>&1 &
allprocs="Sallprocs ${!}"
Stpcc_load -M 349440 -o ${tpcc_disks_location}dummy120.dat -b 327601 -e 330330 >>
loadordrordl120.log 2>&1 &
allprocs="Sallprocs ${!}"
Stpcc_load -M 349440 -o ${tpcc_disks_location}dummy121.dat -b 330331 -e 333060 >>
loadordrordl121.log 2>&1 &
allprocs="Sallprocs ${!}"
Stpcc_load -M 349440 -o ${tpcc_disks_location}dummy122.dat -b 333061 -e 335790 >>
loadordrordl122.log 2>&1 &
allprocs="Sallprocs ${!}"
Stpcc_load -M 349440 -o ${tpcc_disks_location}dummy123.dat -b 335791 -e 338520 >>
loadordrordl123.log 2>&1 &
allprocs="Sallprocs ${!}"
Stpcc_load -M 349440 -o ${tpcc_disks_location}dummy124.dat -b 338521 -e 341250 >>
loadordrordl124.log 2>&1 &
allprocs="Sallprocs ${!}"
Stpcc_load -M 349440 -o ${tpcc_disks_location}dummy125.dat -b 341251 -e 343980 >>
loadordrordl125.log 2>&1 &
allprocs="Sallprocs ${!}"
Stpcc_load -M 349440 -o ${tpcc_disks_location}dummy126.dat -b 343981 -e 346710 >>
loadordrordl126.log 2>&1 &
allprocs="Sallprocs ${!}"
Stpcc_load -M 349440 -o ${tpcc_disks_location}dummy127.dat -b 346711 -e 349440 >>
loadordrordl127.log 2>&1 &

```

```

allprocs="Sallprocs ${!}"
error=0
for curproc in Sallprocs; do
  wait Scurproc
  error=`expr $? + $error`
done
exit `expr $error != 0`

```

loadstok.sh

```

#created automatically by /BUILD/scripts/evenload.sh Fri Dec 8 17:07:30 PST 2006
rm -f loadstok*.log
cd Stpcc_bench
allprocs=
Stpcc_load -M 349440 -S -j 1 -k 781 >> loadstok0.log 2>&1 &
allprocs="Sallprocs ${!}"
Stpcc_load -M 349440 -S -j 782 -k 1562 >> loadstok1.log 2>&1 &
allprocs="Sallprocs ${!}"
Stpcc_load -M 349440 -S -j 1563 -k 2343 >> loadstok2.log 2>&1 &
allprocs="Sallprocs ${!}"
Stpcc_load -M 349440 -S -j 2344 -k 3124 >> loadstok3.log 2>&1 &
allprocs="Sallprocs ${!}"
Stpcc_load -M 349440 -S -j 3125 -k 3905 >> loadstok4.log 2>&1 &
allprocs="Sallprocs ${!}"
Stpcc_load -M 349440 -S -j 3906 -k 4686 >> loadstok5.log 2>&1 &
allprocs="Sallprocs ${!}"
Stpcc_load -M 349440 -S -j 4687 -k 5467 >> loadstok6.log 2>&1 &
allprocs="Sallprocs ${!}"
Stpcc_load -M 349440 -S -j 5468 -k 6248 >> loadstok7.log 2>&1 &
allprocs="Sallprocs ${!}"
Stpcc_load -M 349440 -S -j 6249 -k 7029 >> loadstok8.log 2>&1 &
allprocs="Sallprocs ${!}"
Stpcc_load -M 349440 -S -j 7030 -k 7810 >> loadstok9.log 2>&1 &
allprocs="Sallprocs ${!}"
Stpcc_load -M 349440 -S -j 7811 -k 8591 >> loadstok10.log 2>&1 &
allprocs="Sallprocs ${!}"
Stpcc_load -M 349440 -S -j 8592 -k 9372 >> loadstok11.log 2>&1 &
allprocs="Sallprocs ${!}"
Stpcc_load -M 349440 -S -j 9373 -k 10153 >> loadstok12.log 2>&1 &
allprocs="Sallprocs ${!}"
Stpcc_load -M 349440 -S -j 10154 -k 10934 >> loadstok13.log 2>&1 &
allprocs="Sallprocs ${!}"
Stpcc_load -M 349440 -S -j 10935 -k 11715 >> loadstok14.log 2>&1 &
allprocs="Sallprocs ${!}"
Stpcc_load -M 349440 -S -j 11716 -k 12496 >> loadstok15.log 2>&1 &
allprocs="Sallprocs ${!}"
Stpcc_load -M 349440 -S -j 12497 -k 13277 >> loadstok16.log 2>&1 &
allprocs="Sallprocs ${!}"
Stpcc_load -M 349440 -S -j 13278 -k 14058 >> loadstok17.log 2>&1 &
allprocs="Sallprocs ${!}"
Stpcc_load -M 349440 -S -j 14059 -k 14839 >> loadstok18.log 2>&1 &
allprocs="Sallprocs ${!}"
Stpcc_load -M 349440 -S -j 14840 -k 15620 >> loadstok19.log 2>&1 &
allprocs="Sallprocs ${!}"
Stpcc_load -M 349440 -S -j 15621 -k 16401 >> loadstok20.log 2>&1 &
allprocs="Sallprocs ${!}"
Stpcc_load -M 349440 -S -j 16402 -k 17182 >> loadstok21.log 2>&1 &
allprocs="Sallprocs ${!}"
Stpcc_load -M 349440 -S -j 17183 -k 17963 >> loadstok22.log 2>&1 &
allprocs="Sallprocs ${!}"
Stpcc_load -M 349440 -S -j 17964 -k 18744 >> loadstok23.log 2>&1 &
allprocs="Sallprocs ${!}"
Stpcc_load -M 349440 -S -j 18745 -k 19525 >> loadstok24.log 2>&1 &
allprocs="Sallprocs ${!}"
Stpcc_load -M 349440 -S -j 19526 -k 20306 >> loadstok25.log 2>&1 &
allprocs="Sallprocs ${!}"
Stpcc_load -M 349440 -S -j 20307 -k 21087 >> loadstok26.log 2>&1 &
allprocs="Sallprocs ${!}"
Stpcc_load -M 349440 -S -j 21088 -k 21868 >> loadstok27.log 2>&1 &
allprocs="Sallprocs ${!}"
Stpcc_load -M 349440 -S -j 21869 -k 22649 >> loadstok28.log 2>&1 &
allprocs="Sallprocs ${!}"
Stpcc_load -M 349440 -S -j 22650 -k 23430 >> loadstok29.log 2>&1 &
allprocs="Sallprocs ${!}"
Stpcc_load -M 349440 -S -j 23431 -k 24211 >> loadstok30.log 2>&1 &
allprocs="Sallprocs ${!}"
Stpcc_load -M 349440 -S -j 24212 -k 24992 >> loadstok31.log 2>&1 &
allprocs="Sallprocs ${!}"
Stpcc_load -M 349440 -S -j 24993 -k 25773 >> loadstok32.log 2>&1 &
allprocs="Sallprocs ${!}"
Stpcc_load -M 349440 -S -j 25774 -k 26554 >> loadstok33.log 2>&1 &
allprocs="Sallprocs ${!}"
Stpcc_load -M 349440 -S -j 26555 -k 27335 >> loadstok34.log 2>&1 &
allprocs="Sallprocs ${!}"
Stpcc_load -M 349440 -S -j 27336 -k 28116 >> loadstok35.log 2>&1 &
allprocs="Sallprocs ${!}"
Stpcc_load -M 349440 -S -j 28117 -k 28897 >> loadstok36.log 2>&1 &
allprocs="Sallprocs ${!}"
Stpcc_load -M 349440 -S -j 28898 -k 29678 >> loadstok37.log 2>&1 &
allprocs="Sallprocs ${!}"
Stpcc_load -M 349440 -S -j 29679 -k 30459 >> loadstok38.log 2>&1 &
allprocs="Sallprocs ${!}"
Stpcc_load -M 349440 -S -j 30460 -k 31240 >> loadstok39.log 2>&1 &

```


localoptions.sh

```
#LOCAL OPTION FILE- You must fill these in
# before the driver will work.
```

```
#oracle sid to use for the run
ORACLE_SID=tpcc
```

```
#folder location of the database files (or links to raw partitions)
tpcc_disks_location=/BUILD/dbs/tpcc_disks
```

```
#FOR NT
#tpcc_disks_location=\\\\\\
```

```
#FOR RAC
```

```
#node id
#tpcc_rac_id=1
```

```
# How many createts_node*.sh will be run in this node, started from tpcc_rac_id
# eq. if tpcc_rac_id is 3 and tpcc_rac_createts_count is 2
# createts_node3.sh and createts_node4.sh will be executed
```

```
#tpcc_rac_createts_count=1
```

```
#locations of various files used in the generation scripts.
#(you can usually leave these alone.)
tpcc_sql_dir=${tpcc_bench}/scripts/sql
tpcc_log_dir=${tpcc_bench}/log
tpcc_genscripts_dir=${tpcc_bench}/scripts/generated
```

```
#Once you have filled all the options, comment
#out or delete this line.
#tpcc_no_options=t
```

new.sql

```
rem
rem
=====
rem      Copyright (c) 1996 Oracle Corp, Redwood Shores, CA   |
rem      OPEN SYSTEMS PERFORMANCE GROUP                      |
rem      All Rights Reserved                                   |
rem
=====
rem FILENAME
rem new.sql
rem DESCRIPTION
rem SQL script to create a stored package for new order
rem transactions.
rem
=====
=
rem
```

```
CREATE OR REPLACE PACKAGE neworder
IS
```

```
  PROCEDURE enterorder
  (
    ware_id          INTEGER,
    dist_id          INTEGER,
    cust_id          INTEGER,
    ord_o1_cnt       INTEGER,
    ord_all_local    INTEGER,
    cust_discount    OUT NUMBER,
    cust_last        OUT VARCHAR2,
    cust_credit      OUT VARCHAR2,
    dist_tax         OUT NUMBER,
    ware_tax         OUT NUMBER,
    ord_id           IN OUT INTEGER,
    ord_entry_d      IN OUT VARCHAR2,
    retry           IN OUT INTEGER,
    cur_date         IN          DATE
  );
END;
/
show errors;
```

```
CREATE OR REPLACE PACKAGE BODY neworder
IS
```

```
PROCEDURE enterorder
```

```
(
  ware_id          INTEGER,
  dist_id          INTEGER,
  cust_id          INTEGER,
  ord_o1_cnt       INTEGER,
  ord_all_local    INTEGER,
  cust_discount    OUT NUMBER,
  cust_last        OUT VARCHAR2,
  cust_credit      OUT VARCHAR2,
  dist_tax         OUT NUMBER,
  ware_tax         OUT NUMBER,
  ord_id           IN OUT INTEGER,
  ord_entry_d      IN OUT VARCHAR2,
  retry           IN OUT INTEGER,
  cur_date         IN          DATE
)
IS
  timestamp        DATE;
  dist_rowid       rowid;
  node_num         varchar2(512);
  not_serializable EXCEPTION;
  PRAGMA EXCEPTION_INIT(not_serializable,-8177);
  deadlock         EXCEPTION;
  PRAGMA EXCEPTION_INIT(deadlock,-60);
  snapshot_too_old EXCEPTION;
  PRAGMA EXCEPTION_INIT(snapshot_too_old,-1555);
BEGIN
  SELECT substr(value,1,5)
    INTO node_num
   FROM v$parameter
   WHERE name = 'instance_number';

  plsqli_mon_pack.print ('New Order started at ' ||
    to_char(sysdate, 'HH24:MI:SS') || ' on node ' ||
    node_num);
LOOP BEGIN
  SELECT dist_rowid, d_tax, d_next_o_id, w_tax
    INTO dist_rowid, dist_tax, ord_id, ware_tax
   FROM dist, ware
   WHERE d_id = dist_id AND d_w_id = ware_id
        AND w_id = ware_id;
  UPDATE dist SET d_next_o_id = ord_id + 1
   WHERE rowid = dist_rowid;
  SELECT c_discount, c_last, c_credit
    INTO cust_discount, cust_last, cust_credit
   FROM cust
   WHERE c_id = cust_id AND c_d_id = dist_id AND c_w_id = ware_id;
  timestamp := cur_date;
  ord_entry_d := TO_CHAR(timestamp, 'DD-MM-YYYY.HH24:MI:SS');
  INSERT INTO nord(no_o_id, no_d_id, no_w_id) VALUES
    (ord_id, dist_id, ware_id);
  INSERT INTO ord(o_id, o_d_id, o_w_id, o_c_id, o_entry_d, o_carrier_id,
    o_o1_cnt, o_all_local)
  VALUES (ord_id, dist_id, ware_id, cust_id,
    timestamp, 11, ord_o1_cnt, ord_all_local);
  EXIT;

EXCEPTION
  WHEN not_serializable OR deadlock OR snapshot_too_old THEN
    ROLLBACK;
    retry := retry + 1;
  END;
END LOOP;
END;
END;
/
show errors;

quit;
```

p_build.ora

```
compatible = 10.1.0.0.0
db_name = tpcc
control_files = (/BUILD/dbs/tpcc_disks/control_001,/BUILD/dbs/tpcc_disks/control_002)
parallel_max_servers = 1000
recovery_parallelism = 40
db_files = 1958
db_cache_size = 170666M
db_8k_cache_size = 64000M
db_16k_cache_size = 170666M
dml_locks = 500
statistics_level = basic
log_buffer = 16777216
processes = 4200
sessions = 4200
transactions = 4200
shared_pool_size = 32000M
cursor_space_for_time = TRUE
db_block_size = 4096
undo_management = auto
```

```

undo_retention = 2
plsql_optimize_level=2
# cpu_count = 32
HPUX_SCHED_NOAGE=180

UNDO_TABLESPACE = undo_1
db_2k_cache_size = 20M

```

p_create.ora

```

compatible = 10.1.0.0
db_name = tpcc
control_files = (/BUILD/dbs/tpcc_disks/control_001, /BUILD/dbs/tpcc_disks/control_002)
db_block_size = 4096
db_cache_size = 170666M
db_8k_cache_size = 64000M
log_buffer = 1048576
db_16k_cache_size = 170666M
undo_management = manual
statistics_level = basic
shared_pool_size = 32000M
plsql_optimize_level=2
processes = 100
db_2k_cache_size = 20M

```

pay.sql

```

rem
rem
=====
rem      Copyright (c) 1996 Oracle Corp. Redwood Shores, CA   |
rem      OPEN SYSTEMS PERFORMANCE GROUP                     |
rem      All Rights Reserved                                  |
rem
=====
rem FILENAME
rem      pay.sql
rem DESCRIPTION
rem      SQL script to create a stored procedure for payment
rem      transactions.
rem
=====
rem
CREATE OR REPLACE PACKAGE payment
IS
PROCEDURE dopayment_z
(
  ware_id      INTEGER,
  dist_id     INTEGER,
  cust_w_id   INTEGER,
  cust_d_id   INTEGER,
  cust_id     IN OUT INTEGER,
  bylastname  INTEGER,
  hist_amount INTEGER,
  cust_last   IN OUT VARCHAR2,
  ware_street_1 OUT VARCHAR2,
  ware_street_2 OUT VARCHAR2,
  ware_city   OUT VARCHAR2,
  ware_state  OUT VARCHAR2,
  ware_zip    OUT VARCHAR2,
  dist_street_1 OUT VARCHAR2,
  dist_street_2 OUT VARCHAR2,
  dist_city   OUT VARCHAR2,
  dist_state  OUT VARCHAR2,
  dist_zip    OUT VARCHAR2,
  cust_first  OUT VARCHAR2,
  cust_middle OUT VARCHAR2,
  cust_street_1 OUT VARCHAR2,
  cust_street_2 OUT VARCHAR2,
  cust_city   OUT VARCHAR2,
  cust_state  OUT VARCHAR2,
  cust_zip    OUT VARCHAR2,
  cust_phone  OUT VARCHAR2,
  cust_since  OUT VARCHAR2,
  cust_credit IN OUT VARCHAR2,
  cust_credit_lim OUT NUMBER,
  cust_discount OUT NUMBER,
  cust_balance IN OUT NUMBER,
  cust_data   OUT VARCHAR2,
  hist_date  OUT VARCHAR2,
  retry      IN OUT INTEGER,
  cur_date   IN DATE
);
PROCEDURE dopayment_nz

```

```

(
  ware_id      INTEGER,
  dist_id     INTEGER,
  cust_w_id   INTEGER,
  cust_d_id   INTEGER,
  cust_id     IN OUT INTEGER,
  bylastname  INTEGER,
  hist_amount INTEGER,
  cust_last   IN OUT VARCHAR2,
  ware_street_1 OUT VARCHAR2,
  ware_street_2 OUT VARCHAR2,
  ware_city   OUT VARCHAR2,
  ware_state  OUT VARCHAR2,
  ware_zip    OUT VARCHAR2,
  dist_street_1 OUT VARCHAR2,
  dist_street_2 OUT VARCHAR2,
  dist_city   OUT VARCHAR2,
  dist_state  OUT VARCHAR2,
  dist_zip    OUT VARCHAR2,
  cust_first  OUT VARCHAR2,
  cust_middle OUT VARCHAR2,
  cust_street_1 OUT VARCHAR2,
  cust_street_2 OUT VARCHAR2,
  cust_city   OUT VARCHAR2,
  cust_state  OUT VARCHAR2,
  cust_zip    OUT VARCHAR2,
  cust_phone  OUT VARCHAR2,
  cust_since  OUT VARCHAR2,
  cust_credit IN OUT VARCHAR2,
  cust_credit_lim OUT NUMBER,
  cust_discount OUT NUMBER,
  cust_balance IN OUT NUMBER,
  cust_data   OUT VARCHAR2,
  hist_date  OUT VARCHAR2,
  retry      IN OUT INTEGER,
  cur_date   IN DATE
);
END;
/
show errors;

CREATE OR REPLACE PACKAGE BODY payment
IS
PROCEDURE dopayment_z
(
  ware_id      INTEGER,
  dist_id     INTEGER,
  cust_w_id   INTEGER,
  cust_d_id   INTEGER,
  cust_id     IN OUT INTEGER,
  bylastname  INTEGER,
  hist_amount INTEGER,
  cust_last   IN OUT VARCHAR2,
  ware_street_1 OUT VARCHAR2,
  ware_street_2 OUT VARCHAR2,
  ware_city   OUT VARCHAR2,
  ware_state  OUT VARCHAR2,
  ware_zip    OUT VARCHAR2,
  dist_street_1 OUT VARCHAR2,
  dist_street_2 OUT VARCHAR2,
  dist_city   OUT VARCHAR2,
  dist_state  OUT VARCHAR2,
  dist_zip    OUT VARCHAR2,
  cust_first  OUT VARCHAR2,
  cust_middle OUT VARCHAR2,
  cust_street_1 OUT VARCHAR2,
  cust_street_2 OUT VARCHAR2,
  cust_city   OUT VARCHAR2,
  cust_state  OUT VARCHAR2,
  cust_zip    OUT VARCHAR2,
  cust_phone  OUT VARCHAR2,
  cust_since  OUT VARCHAR2,
  cust_credit IN OUT VARCHAR2,
  cust_credit_lim OUT NUMBER,
  cust_discount OUT NUMBER,
  cust_balance IN OUT NUMBER,
  cust_data   OUT VARCHAR2,
  hist_date  OUT VARCHAR2,
  retry      IN OUT INTEGER,
  cur_date   IN DATE
)
IS
TYPE rowidarray IS TABLE OF ROWID INDEX BY BINARY_INTEGER;
cust_rowid      ROWID;
ware_rowid      ROWID;
dist_ytd        NUMBER(12);
dist_name       VARCHAR2(11);
ware_ytd        NUMBER(12);
ware_name       VARCHAR2(11);
history_date    DATE;
c_num           BINARY_INTEGER;
row_id          rowidarray;
cust_payments   PLS_INTEGER;
cust_ytd        NUMBER(12);
cust_data_temp  VARCHAR2(500);

```

```

node_num          VARCHAR2(512);
not_serializable  EXCEPTION;
PRAGMA EXCEPTION_INIT(not_serializable,-8177);
deadlock          EXCEPTION;
PRAGMA EXCEPTION_INIT(deadlock,-60);
snapshot_too_old  EXCEPTION;
PRAGMA EXCEPTION_INIT(snapshot_too_old,-1555);
CURSOR c_cur IS
    SELECT rowid
    FROM cust
    WHERE c_d_id = cust_d_id AND c_w_id = cust_w_id AND c_last = cust_last
    ORDER BY c_w_id, c_d_id, c_last, c_first;
BEGIN
SELECT substr(value,1,5)
INTO node_num
FROM v$parameter
WHERE name = 'instance_number';

plsql_mon_pack.print ('Payment started at ' ||
    to_char(sysdate, 'HH24:MI:SS') || ' on node ' ||
    node_num);

LOOP BEGIN
    SELECT rowid, c_first, c_middle, c_last, c_street_1, c_street_2,
        c_city, c_state, c_zip, c_phone,
        to_char(c_since, 'DD-MM-YYYY'), c_credit, c_credit_lim,
        c_discount, c_balance - hist_amount, c_payment_cnt,
        c_ytd_payment + hist_amount, decode(c_credit, 'BC', c_data,
        ')
        INTO cust_rowid, cust_first, cust_middle, cust_last,
            cust_street_1, cust_street_2, cust_city, cust_state,
            cust_zip, cust_phone, cust_since, cust_credit,
            cust_credit_lim, cust_discount, cust_balance, cust_payments,
            cust_ytd, cust_data_temp
        FROM cust
        WHERE c_id = cust_id AND c_d_id = cust_d_id AND
            c_w_id = cust_w_id;
    cust_payments := cust_payments + 1;
    IF cust_credit = 'BC' THEN
        cust_data_temp := substr((to_char(cust_id) || ' ' ||
            to_char(cust_d_id) || ' ' ||
            to_char(cust_w_id) || ' ' ||
            to_char(dist_id) || ' ' ||
            to_char(ware_id) || ' ' ||
            to_char(hist_amount, '9999.99') || ' ')
            || cust_data_temp, 1, 500);

    UPDATE cust
    SET c_balance = cust_balance,
        c_ytd_payment = cust_ytd,
        c_payment_cnt = cust_payments,
        c_data = cust_data_temp
    WHERE rowid = cust_rowid;

    cust_data := substr(cust_data_temp,1, 200);

ELSE

    UPDATE cust
    SET c_balance = cust_balance,
        c_ytd_payment = cust_ytd,
        c_payment_cnt = cust_payments
    WHERE rowid = cust_rowid;

    cust_data := cust_data_temp;

END IF;

SELECT dist.rowid, d_name, d_street_1, d_street_2, d_city,
    d_state, d_zip, d_ytd + hist_amount,
    ware.rowid, w_name, w_street_1, w_street_2, w_city,
    w_state, w_zip, w_ytd + hist_amount
INTO cust_rowid, dist_name, dist_street_1, dist_street_2, dist_city,
    dist_state, dist_zip, dist_ytd,
    ware_rowid, ware_name, ware_street_1, ware_street_2, ware_city,
    ware_state, ware_zip, ware_ytd
FROM dist, ware
WHERE d_id = dist_id
AND d_w_id = ware_id
AND w_id = ware_id;

UPDATE dist
SET d_ytd = dist_ytd
WHERE rowid = cust_rowid;

UPDATE ware
SET w_ytd = ware_ytd
WHERE rowid = ware_rowid;

history_date := cur_date;

INSERT INTO hist(h_c_id,h_c_d_id, h_c_w_id,h_d_id,h_w_id,h_date,
    h_amount, h_data) VALUES
(cust_id, cust_d_id, cust_w_id, dist_id, ware_id, history_date,
    hist_amount, ware_name || ' ' || dist_name);
COMMIT;
hist_date := to_char(history_date, 'DD-MM-YYYY.HH24:MI:SS');

EXIT;

EXCEPTION
    WHEN not_serializable OR deadlock OR snapshot_too_old THEN
        ROLLBACK;
        retry := retry + 1;
    END;

END LOOP;
END;

PROCEDURE dopayment_nz
(
    ware_id          INTEGER,
    dist_id          INTEGER,
    cust_w_id        INTEGER,
    cust_d_id        INTEGER,
    cust_id          IN OUT INTEGER,
    bylastname       INTEGER,
    hist_amount      INTEGER,
    cust_last        IN OUT VARCHAR2,
    ware_street_1    OUT VARCHAR2,
    ware_street_2    OUT VARCHAR2,
    ware_city        OUT VARCHAR2,
    ware_state       OUT VARCHAR2,
    ware_zip         OUT VARCHAR2,
    dist_street_1    OUT VARCHAR2,
    dist_street_2    OUT VARCHAR2,
    dist_city        OUT VARCHAR2,
    dist_state       OUT VARCHAR2,
    dist_zip         OUT VARCHAR2,
    cust_first       OUT VARCHAR2,
    cust_middle      OUT VARCHAR2,
    cust_street_1    OUT VARCHAR2,
    cust_street_2    OUT VARCHAR2,
    cust_city        OUT VARCHAR2,
    cust_state       OUT VARCHAR2,
    cust_zip         OUT VARCHAR2,
    cust_phone       OUT VARCHAR2,
    cust_since       OUT VARCHAR2,
    cust_credit      IN OUT VARCHAR2,
    cust_credit_lim  OUT NUMBER,
    cust_discount    OUT NUMBER,
    cust_balance     IN OUT NUMBER,
    cust_data        OUT VARCHAR2,
    hist_date        OUT VARCHAR2,
    retry            IN OUT INTEGER,
    cur_date         IN DATE
)
IS
    TYPE rowidarray IS TABLE OF ROWID INDEX BY BINARY_INTEGER;
    cust_rowid      ROWID;
    ware_rowid      ROWID;
    dist_ytd        NUMBER(12);
    dist_name       VARCHAR2(11);
    ware_ytd        NUMBER(12);
    ware_name       VARCHAR2(11);
    history_date    DATE;
    c_num           BINARY_INTEGER;
    row_id          rowidarray;
    cust_payments   PLS_INTEGER;
    cust_ytd        NUMBER(12);
    cust_data_temp  VARCHAR2(500);
    node_num        VARCHAR2(512);
    not_serializable EXCEPTION;
    PRAGMA EXCEPTION_INIT(not_serializable,-8177);
    deadlock        EXCEPTION;
    PRAGMA EXCEPTION_INIT(deadlock,-60);
    snapshot_too_old EXCEPTION;
    PRAGMA EXCEPTION_INIT(snapshot_too_old,-1555);
    CURSOR c_cur IS
        SELECT rowid
        FROM cust
        WHERE c_d_id = cust_d_id AND c_w_id = cust_w_id AND c_last = cust_last
        ORDER BY c_w_id, c_d_id, c_last, c_first;
    BEGIN
    SELECT substr(value,1,5)
    INTO node_num
    FROM v$parameter
    WHERE name = 'instance_number';

    plsql_mon_pack.print ('Payment started at ' ||
        to_char(sysdate, 'HH24:MI:SS') || ' on node ' ||
        node_num);

    LOOP BEGIN

        c_num := 0;
        FOR c_id_rec IN c_cur LOOP
            c_num := c_num + 1;
            row_id(c_num) := c_id_rec.rowid;
        END LOOP;
        cust_rowid := row_id((c_num + 1) / 2); -- use row_id.count ?

        SELECT c_id, c_first, c_middle, c_last, c_street_1, c_street_2,

```

```

c_city, c_state, c_zip, c_phone,
to_char(c_since, 'DD-MM-YYYY'), c_credit, c_credit_lim,
c_discount, c_balance - hist_amount, c_payment_cnt,
c_ytd_payment + hist_amount, decode(c_credit, 'BC', c_data,
')
INTO cust_id, cust_first, cust_middle, cust_last,
cust_street_1, cust_street_2, cust_city, cust_state,
cust_zip, cust_phone, cust_since, cust_credit,
cust_credit_lim, cust_discount, cust_balance, cust_payments,
cust_ytd, cust_data_temp
FROM cust
WHERE rowid = cust_rowid;
cust_payments := cust_payments + 1;
IF cust_credit = 'BC' THEN
cust_data_temp := substr((to_char(cust_id) || ' ' ||
to_char(cust_d_id) || ' ' ||
to_char(cust_w_id) || ' ' ||
to_char(dist_id) || ' ' ||
to_char(ware_id) || ' ' ||
to_char(hist_amount/100, '9999.99') || ' ')
|| cust_data_temp, 1, 500);
UPDATE cust
SET c_balance = cust_balance,
c_ytd_payment = cust_ytd,
c_payment_cnt = cust_payments,
c_data = cust_data_temp
WHERE rowid = cust_rowid;
cust_data := substr(cust_data_temp, 1, 200);
ELSE
UPDATE cust
SET c_balance = cust_balance,
c_ytd_payment = cust_ytd,
c_payment_cnt = cust_payments
WHERE rowid = cust_rowid;
cust_data := cust_data_temp;
END IF;
SELECT dist.rowid, d_name, d_street_1, d_street_2, d_city,
d_state, d_zip, d_ytd + hist_amount,
ware.rowid, w_name, w_street_1, w_street_2, w_city,
w_state, w_zip, w_ytd + hist_amount
INTO cust_rowid, dist_name, dist_street_1, dist_street_2, dist_city,
dist_state, dist_zip, dist_ytd,
ware_rowid, ware_name, ware_street_1, ware_street_2, ware_city,
ware_state, ware_zip, ware_ytd
FROM dist, ware
WHERE d_id = dist_id
AND d_w_id = ware_id
AND w_id = ware_id;
UPDATE dist
SET d_ytd = dist_ytd
WHERE rowid = cust_rowid;
UPDATE ware
SET w_ytd = ware_ytd
WHERE rowid = ware_rowid;
history_date := cur_date;
INSERT INTO hist VALUES
(cust_id, cust_d_id, cust_w_id, dist_id, ware_id, history_date,
hist_amount, ware_name || ' ' || dist_name);
COMMIT;
hist_date := to_char(history_date, 'DD-MM-YYYY.HH24:MI:SS');
EXIT;
EXCEPTION
WHEN not_serializable OR deadlock OR snapshot_too_old THEN
ROLLBACK;
retry := retry + 1;
END;
END LOOP;
END;
END;
/
show errors;
quit;

```

plsql_mon.sql

```

rem
rem
=====
=+

```

```

rem Copyright (c) 1995 Oracle Corp, Redwood Shores, CA |
rem OPEN SYSTEMS PERFORMANCE GROUP |
rem All Rights Reserved |
rem
=====
=+
rem FILENAME
rem plsql_mon.sql
rem DESCRIPTION
rem SQL script to create a stored package for PL/SQL stored
rem procedures to dump messages.
rem
=====
=
rem
rem Usage: sqlplus tpcc/tpcc @plsql_mon
rem
connect tpcc/tpcc;
set echo on;
CREATE OR REPLACE PACKAGE plsql_mon_pack
IS
PROCEDURE print
(
info VARCHAR2
);
END;
/
show errors;
CREATE OR REPLACE PACKAGE BODY plsql_mon_pack
IS
PROCEDURE print
(
info VARCHAR2
)
IS
s NUMBER;
BEGIN
dbms_pipe.pack_message(info);
s := dbms_pipe.send_message('plsql_mon');
IF (s <> 0) THEN
raise_application_error(-20000, 'Error!' || to_char(s) ||
'sending on pipe');
END IF;
END;
END;
/
show errors;
set echo off;

```

Pst_c.sql

```

rem
rem
rem
=====
=+
rem Copyright (c) 1992 Oracle Corp, Belmont, CA |
rem OPEN SYSTEMS PERFORMANCE GROUP |
rem All Rights Reserved |
rem
=====
=+
rem FILENAME
rem pst_c.sql
rem DESCRIPTION
rem Create Table for OS Specific Process Stats
rem
=====
= */
rem
rem Tables for Unix-specific process statistics
rem
rem Usage: sqlplus internal/internal @pst_c
rem
connect tpcc/tpcc;
set echo on;
DROP TABLE proc_resource;
DROP TABLE os_stat;
rem
rem Resource usage for a process.
rem
CREATE TABLE proc_resource
(
config VARCHAR2(10),
run NUMBER,
proc NUMBER,
child NUMBER,

```

```

user_cpu_ms NUMBER,
system_cpu_ms NUMBER,
maxrss NUMBER,
pagein NUMBER,
reclaim NUMBER,
zerofill NUMBER,
pffincr NUMBER,
pffdecr NUMBER,
swap NUMBER,
syscall NUMBER,
volcsw NUMBER,
involcsw NUMBER,
signal NUMBER,
lread NUMBER,
lwrite NUMBER,
bread NUMBER,
bwrite NUMBER,
phread NUMBER,
phwrite NUMBER
);

rem
rem OS statistics.
rem These results are from the measurement interval only.
rem

```

```

CREATE TABLE os_stat
(
  config VARCHAR2(10),
  run NUMBER,
  hid NUMBER,
  syscall NUMBER,
  intr NUMBER,
  cswitch NUMBER,
  pagefault NUMBER,
  usr NUMBER,
  sys NUMBER,
  idl NUMBER,
  wio NUMBER
);

set echo off;

```

Shutdowndb.sh

```

#!/bin/sh

echo "Shutting down database..."

$tpcc_sqlplus $tpcc_sqlplus_args << !
$tpcc_internal_connect

spool shutdowndb.log;

set echo on;

alter system switch logfile;
alter system switch logfile;

shutdown immediate;

set echo off;
spool off;

exit
!

```

Space_get.sql

```

REM=====
=====+
REM Copyright (c) 1995 Oracle Corp, Redwood Shores, CA |
REM OPEN SYSTEMS PERFORMANCE GROUP |
REM All Rights Reserved |
REM=====
=====+
REM FILENAME
REM space_get.sql
REM DESCRIPTION
REM Get sizes of tables, indexes and tablespaces.
REM Usage: sqlplus 'sys/change_on_install as sysdba' @space_get <[tpm] <# of
warehouses>]
REM=====
=====*/

set echo on;
delete from tpcc_data;
delete from tpcc_space;
delete from tpcc_totSPACE;

```

```

insert into tpcc_data
select substr(segment_name,1,18), substr(segment_type,1,15),
sum(blocks), t.block_size,
round(sum(blocks) * 0.05), 0,
sum(blocks) + round(sum(blocks) * 0.05)
from dba_extents e, dba_tablespaces t
where owner = 'TPCC' AND ( segment_type = 'INDEX' OR
segment_type = 'INDEX PARTITION' OR segment_type = 'CLUSTER'
OR segment_type = 'TABLE' OR segment_type = 'TABLE PARTITION')
AND e.tablespace_name <> 'SYSTEM' AND e.tablespace_name <> 'SP_0'
AND e.tablespace_name = t.tablespace_name
group by segment_name, segment_type, t.block_size;

insert into tpcc_data
select 'SYSTEM', 'SYS', sum(blocks), t.block_size, 0, 0, sum(blocks)
from dba_data_files f, dba_tablespaces t
where f.tablespace_name = 'SYSTEM' and t.tablespace_name = f.tablespace_name
group by t.block_size;

insert into tpcc_data
select 'SYSAUX', 'SYS', sum(blocks), t.block_size, 0, 0, sum(blocks)
from dba_data_files f, dba_tablespaces t
where f.tablespace_name = 'SYSAUX' and t.tablespace_name = f.tablespace_name
group by t.block_size;

insert into tpcc_data
select 'ROLL_SEG', 'SYS', sum(blocks), t.block_size, 0, 0, sum(blocks)
from dba_data_files f, dba_tablespaces t
where f.tablespace_name like '%UNDO_TS%' and f.tablespace_name =
t.tablespace_name
group by f.tablespace_name, t.block_size;

insert into tpcc_data
select 'DB_STAT', 'SYS', sum(blocks), t.block_size, 0, 0, sum(blocks)
from dba_data_files f, dba_tablespaces t
where f.tablespace_name like '%SP_0%' and f.tablespace_name = t.tablespace_name
group by f.tablespace_name, t.block_size;

update tpcc_data
set five_pct = 0,
daily_grow = round(blocks * &&1 / 62.5 / &&2),
total = blocks + round(blocks * &&1 / 62.5 / &&2)
where segment = 'HIST' OR segment = 'ORDERCLUSTER_QUEUE' OR
segment = 'IORDL';

```

```

insert into tpcc_space
select substr(ex$.name,1,18), sum(sp$.sz_blocks), sp$.block_size, 0, 0, 0, 0
from
(select f.tablespace_name, sum(blocks) sz_blocks, t.block_size block_size
from dba_data_files f, dba_tablespaces t
where f.tablespace_name <> 'SYSTEM' and f.tablespace_name = t.tablespace_name
group by f.tablespace_name, t.block_size
) sp$,
(select distinct tablespace_name, segment_name name
from dba_extents
where owner = 'TPCC'
and (segment_type = 'CLUSTER' or segment_type = 'TABLE'
or segment_type = 'TABLE PARTITION' or segment_type = 'INDEX'
or segment_type = 'INDEX PARTITION')
and tablespace_name <> 'SYSTEM'
) ex$
where sp$.tablespace_name = ex$.tablespace_name
group by ex$.name, sp$.block_size;

```

```

insert into tpcc_space
select substr(f.tablespace_name,1,18), sum(blocks), t.block_size, 0, 0, 0, 0
from dba_data_files f, dba_tablespaces t
where (f.tablespace_name = 'SYSTEM' or f.tablespace_name = 'SYSAUX')
and f.tablespace_name = t.tablespace_name
group by f.tablespace_name, t.block_size;

```

```

insert into tpcc_space
select 'ROLL_SEG', sum(blocks), t.block_size, 0, 0, 0, 0
from dba_data_files f, dba_tablespaces t
where f.tablespace_name = 'UNDO_TS' and f.tablespace_name = t.tablespace_name
group by f.tablespace_name, t.block_size;

```

```

insert into tpcc_space
select 'DB_STAT', sum(blocks), t.block_size, 0, 0, 0, 0
from dba_data_files f, dba_tablespaces t
where f.tablespace_name = 'SP_0' and f.tablespace_name = t.tablespace_name
group by f.tablespace_name, t.block_size;

```

```

update tpcc_space
set required =
(
select sum(total)
from tpcc_data
where tpcc_data.segment = tpcc_space.segment
)
where segment in
(
select segment from tpcc_data
);

```

```

update tpcc_space
set static =
(
  select sum(total)
  from tpcc_data
  where tpcc_data.segment = tpcc_space.segment
)
where segment in
(
  select segment from tpcc_data
);

update tpcc_space
set static = 0,
dynamic =
(
  select sum(blocks)
  from tpcc_data
  where tpcc_data.segment = tpcc_space.segment
)
where segment in ('HIST', 'ORDRCLUSTER_QUEUE', 'IORDL');

update tpcc_space
set oversize = blocks - required;

insert into tpcc_totSPACE
select &&1, &&2, sum(static * block_size)/1024, sum(dynamic * block_size)/1024,
sum(oversize * block_size)/1024, 0, 0, 0
from tpcc_space;

update tpcc_totSPACE
set daily_grow =
(
  select sum(daily_grow * block_size)/1024
  from tpcc_data
);
update tpcc_totSPACE
set space60 = static + 60 * daily_grow;
set echo off;

```

Space_init.sql

```

REM=====
=====+
REM FILENAME
REM space_init.sql
REM DESCRIPTION
REM Create tables for space calculations.
REM Usage: sqlplus 'sys/change_on_install as sysdba' @space_init.sql
REM=====
=====*/
set echo on;
drop table tpcc_data;
drop table tpcc_space;
drop table tpcc_totSPACE;
create table tpcc_data (
  segment varchar2(18),
  type varchar2(15),
  blocks number,
  block_size number,
  five_pct number,
  daily_grow number,
  total number
);
create table tpcc_space (
  segment varchar2(18),
  blocks number,
  block_size number,
  required number,
  static number,
  dynamic number,
  oversize number
);
create table tpcc_totSPACE (
  tpm number,
  nware number,
  static number,
  dynamic number,
  oversize number,
  daily_grow number,
  daily_spre number,
  space60 number
);
create unique index itpcc_data on tpcc_data (segment);
create unique index itpcc_space on tpcc_space (segment);
set echo off;

```

Space_rpt.sql

```

REM=====
=====+
REM Copyright (c) 1995 Oracle Corp, Redwood Shores, CA |
REM OPEN SYSTEMS PERFORMANCE GROUP |
REM All Rights Reserved |
REM=====
=====+
REM FILENAME
REM space_rpt.sql
REM DESCRIPTION
REM Generate space report and save it in space.rpt
REM Usage: sqlplus 'sys/change_on_install as sysdba' @space_rpt.sql
REM=====
=====*/
set space 2
set pagesize 2000
set echo off
set termout off
set verify off
set feedback off
set pagesize 60 linesize 120
spool space.rpt
select tpm, nware from tpcc_totSPACE;
select * from tpcc_data order by segment;
select * from tpcc_space order by segment;
select static, dynamic, oversize, daily_grow, daily_spre, space60
from tpcc_totSPACE;
spool off;

```

startupdb.sh

```

#!/bin/sh

echo "Starting up database using $1..."

init_file=${1}.ora

if test $tpcc_np -gt 1 ; then
  init_file=build_init_${tpcc_rac_id}.ora
fi

$tpcc_sqlplus $tpcc_sqlplus_args << !
$tpcc_internal_connect

spool startdb.log

set echo on

startup pfile=$init_file open

spool off
set echo off
exit sql.sqlcode
!
```

stepenv.sh

```

# forces any env variables we set to be exported
set -a
tpcc_kit=t
tpcc_bench=$PWD
tpcc_scripts=$tpcc_bench/scripts
tpcc_require=$tpcc_scripts/require_vars.sh
tpcc_lcm=$tpcc_scripts/lcm.sh
tpcc_tokilobytes=$tpcc_scripts/tokilobytes.sh
tpcc_fromkilobytes=$tpcc_scripts/fromkilobytes.sh
tpcc_estsize=$tpcc_scripts/estsize.sh
tpcc_notneg=$tpcc_scripts/notneg.sh
tpcc_isneg=$tpcc_scripts/isneg.sh

# need a better way to check for bc, may
# resort to checking each directory in path
# if this doesn't work
# 11/7/02 - alex.ni this is causing too many problems
# because systems have bc in some odd place. typically
# mangled cygwin installs w/ mksnt/cygwin mixes
# if test -x /usr/bin/bc -o -x /bin/bc; then
tpcc_bcexpr=$tpcc_scripts/bcexpr.sh
#else
#tpcc_bcexpr=expr
#fi

# the ksh version is a bit faster, so we want
# to use it if we have ksh. Otherwise we have
# a compatible version.

```

```

# if test -x /bin/ksh; then
# tpcc_createts=$tpcc_scripts/createts.sh
# else
tpcc_createts=$tpcc_scripts/createts.sh
# fi

tpcc_tabledata=$tpcc_scripts/tabledata.sh
tpcc_load=$tpcc_bench/benchrun/bin/tpccload.exe
tpcc_createtablespace=$tpcc_scripts/createtablespace.sh

###
tpcc_sqlplus=cat
tpcc_sqlplus_args="/nolog"
tpcc_internal_connect="connect / as sysdba"
tpcc_user_pass="tpcc/tpcc"
tpcc_dba_user_pass="system/manager"
oracle_dba=system
oracle_dba_password=manager
tpcc_sqlplus=sqlplus

# import options generated by gui
. ${tpcc_bench}/options.sh

# 8gb oracle file size limit (in k)
tpcc_fsize_limit_k=8388608
# 2gb - 1k oracle extent limit (in k)
tpcc_extent_limit_k=2097151

# Runlen calculations should be in hours, but
# this was the old calculation, which assumed
# minutes, and also 8 times:
# tpcc_runlen=$tpcc_bcexpr 8 \* 60 \* $tpcc_runlen`
# we just want to keep the value as it is.

tpcc_system_size=400M
tpcc_kilo_bytes=1024
#tpcc_logfile_size=$tpcc_bcexpr 20 + \( $tpcc_scale \)`

if test $tpcc_np -gt 1 ; then
# 4.69k per commit * 2.1 commit per TPMC ~ 9.85K
# 9.85k * 30 minutes * 12.5 TPMC per Warehouse = 3693
tpcc_logfile_size=$tpcc_bcexpr \( $tpcc_scale \* 3693 \) / $tpcc_kilo_bytes`
else
# 2.4k per commit * 2.1 commit per TPMC ~ 5k
# 5k * 30 minutes * 12.5 TPMC per Warehouse = 1875
tpcc_logfile_size=$tpcc_bcexpr \( $tpcc_scale \* 1875 \) / $tpcc_kilo_bytes`
fi

if test $tpcc_logfile_size -lt 1024; then
tpcc_logfile_size=1024
fi
tpcc_logfile_size="${tpcc_logfile_size}M"

tpcc_undo_size=$tpcc_bcexpr 2 \* $tpcc_scale`
if test $tpcc_undo_size -gt 8096; then
tpcc_undo_size=8096
fi
if test $tpcc_undo_size -lt 512; then
tpcc_undo_size=512
fi
tpcc_undo_size="${tpcc_undo_size}M"

tpcc_undo_bs=8K

tpcc_statspack_size=$tpcc_bcexpr 1 \* $tpcc_scale`
if test $tpcc_statspack_size -gt 2048; then
tpcc_statspack_size=2048
fi
if test $tpcc_statspack_size -lt 300; then
tpcc_statspack_size=300
fi
tpcc_statspack_size="${tpcc_statspack_size}M"

tpcc_sysaux_size=120M

# fixed table params

# table list (note temp is always at the end since it may use numbers from other tables, and
# it's not included in these lists)
tpcc_table_list="ware cust dist hist stok item ordr ordl nordr"
tpcc_index_list="iware icust1 icust2 idist istok ititem iordr1 iordr2 iordl inordr"
# for these I use average row length, calculated from multi-blocksize stats.
# we figure out how many new rows we will gain in a run (in createtablespace.sh)
# and add that much to the base tablespace size.
tpcc_hist_growth=51
tpcc_ordr_growth=35
tpcc_nordr_growth=regular
tpcc_ordl_growth=660
tpcc_ordl_growth=900

# i started indices at 1/10th... need an exact figure
tpcc_iordr1_growth=20
tpcc_iordr2_growth=20
tpcc_iordl_growth=66
tpcc_inordr_growth=2

tpcc_item_growth=0
tpcc_iitem_growth=0
tpcc_temp_growth=0

tpcc_cust_growth=regular
tpcc_icust1_growth=regular
tpcc_icust2_growth=regular

tpcc_stok_growth=regular
tpcc_istok_growth=regular

tpcc_ware_growth=regular
tpcc_iware_growth=regular

tpcc_dist_growth=regular
tpcc_idist_growth=regular

# minimum size of temp tablespace
tpcc_tempts_min=10240

# for Linux, set appropriate tablespace heuristics
# to set high io tables to have 64 files, and minimize
# others.
if expr $tpcc_os = linux > /dev/null; then
# for table in $tpcc_table_list $tpcc_index_list temp; do
# eval "tpcc_${table}_tsfileinc=1"
# done
if test $tpcc_numfiles = 0 ; then
tpcc_numfiles=256
fi
tpcc_os=unix

# tpcc_stok_tsfileinc=64
# tpcc_cust_tsfileinc=64
# tpcc_iordl2_tsfileinc=16
# tpcc_icust2_tsfileinc=16
# tpcc_iordl_tsfileinc=16
else
# in case someone changes out of linux, and the shell is stuck
# for table in $tpcc_table_list $tpcc_index_list temp; do
eval "tpcc_${table}_tsfileinc="
done
fi
tpcc_stok_tsfileinc=
tpcc_cust_tsfileinc=
tpcc_iordl2_tsfileinc=
tpcc_icust2_tsfileinc=
tpcc_iordl_tsfileinc=
# fi

# import local options
. ${tpcc_bench}/localoptions.sh

if expr `echo x$tpcc_no_options` = xt > /dev/null; then
echo Please modify ${tpcc_bench}/localoptions.sh to configure the generator.
exit 1
fi

tpcc_fixordrordl=${tpcc_genscripts_dir}/loadfixordrordl.sh
tpcc_updateordrordl=${tpcc_scripts}/updateordrordl.sh

# tp- get table param. (that is, $tpcc_tablename_tableparam)
tp(){
eval echo \`${tpcc}_${1}_$2\`
}

# automatically generated variables
if expr `echo $tpcc_version | cut -b1` = t > /dev/null; then
tpcc_auto_undo=t
else
tpcc_auto_undo=f
fi
if expr `echo $tpcc_version | cut -b2` = t > /dev/null; then
tpcc_autospace_avail=t
else
tpcc_autospace_avail=f
fi
if expr `echo $tpcc_version | cut -b3` = t > /dev/null; then
tpcc_queue_avail=t
tpcc_use_sysaux=t
else
tpcc_queue_avail=f
tpcc_use_sysaux=f
fi

# for NT, ORACLE does not like $variables in sql scripts, so we must
# hardcode these things for it.
if test x$tpcc_os = xnt; then
tpcc_hardcode=t
else
tpcc_hardcode=f
fi

```

```

# if this is unset we need to make sure it's something anyway
if test x$tpcc_defbs = x; then
  tpcc_defbs=2
fi

# used for loading program
if test x$tpcc_hash_overflow = xt; then
  tpcc_hash_overflow=t
else
  unset tpcc_hash_overflow
fi
if test x$tpcc_overflow = xt; then
  tpcc_hash_overflow=t
else
  unset tpcc_hash_overflow
fi

tpcc_create_steps="buildtpccflags buildcreatets buildcreatedb \
buildcreatetable-ware buildcreatetable-cust buildcreatetable-dist buildcreatetable-hist \
buildcreatetable-stok buildcreatetable-item buildcreatetable-ordr buildcreatetable-ordl \
buildcreatetable-nord \
buildloadware buildloadhist buildloaditem buildloadhist buildloadnord buildloadordrdrl \
buildloadcust buildloadstok \
buildcreateindex-iware buildcreateindex-icust1 buildcreateindex-icust2 buildcreateindex- \
idist buildcreateindex-istok buildcreateindex-item buildcreateindex-iordr1 \
buildcreateindex-iordr2 buildcreateindex-iordl buildcreateindex-inord \
buildstoreprosql buildspacestats listfiles
"

# remove runscript-loadfixordrdrl - shuang, 030626

tpcc_steps="runsqllocal-createdb shutdowndb startupdb-p_build createuser ddview \
runscript-createts assigntemp \
  runsql-createtable-ware runsql-createtable-cust runsql-createtable-dist runsql- \
createtable-hist runsql-createtable-stok runsql-createtable-item runsql-createtable-ordr \
runsql-createtable-ordl runsql-createtable-nord \
runscript-loadware runscript-loadhist runscript-loaditem runscript-loadhist runscript- \
loadnord runscript-loadordrdrl runscript-loadcust runscript-loadstok \
analyze runsql-createindex-iware runsql-createindex-icust1 runsql-createindex-icust2 \
runsql-createindex-idist runsql-createindex-istok runsql-createindex-item runsql- \
createindex-iordr1 runsql-createindex-iordr2 runsql-createindex-iordl runsql- \
createindex-inord \
createts createstoredprocs createspacestats createmisc"

tpcc_total_files=524

# no longer automatically exports env variables
set +a

# check for problems with configuration
badconf=
for table in $tpcc_table_list; do
  if expr `tp $table imp` = queue > /dev/null; then
    if expr $tpcc_queue_avail = f > /dev/null; then
      echo Table $table may not be a queue, since queues are
      echo are unavailable in the selected Oracle version.
      badconf=t
    fi
  fi
  if expr $tpcc_autospace_avail = f & `tp $table autospace` = t > /dev/null; then
    echo Table $table may not use bitmapped space management
    echo since it is not available in the selected Oracle version.
    badconf=t
  fi
done

if test -n "$badconf"; then
  exit 1
fi

# make sure we have everything
if $tpcc_require ORACLE_SID \
  tpcc_tokilobytes tpcc_createts tpcc_lcm \
  tpcc_sqlplus tpcc_internal_connect \
  tpcc_np tpcc_cpu tpcc_os tpcc_runlen tpcc_ldrive tpcc_scale tpcc_disks_location \
  tpcc_auto_undo tpcc_tempt_min \
  tpcc_system_size tpcc_logfile_size \
  tpcc_undo_size tpcc_undo_bs \
  oracle_dba oracle_dba_password tpcc_dba_user_pass
then exit 1; fi

if test x$tpcc_hardcode != xt; then
  tpcc_disks_location=${tpcc_disks_location}/
# tpcc_sql_dir=$tpcc_sql_dir
# tpcc_statspack_size=$tpcc_statspack_size
# tpcc_genscripts_dir=$tpcc_genscripts_dir
fi

```

```

set echo on;

create or replace view wh_cust
(w_id, w_tax, c_id, c_d_id, c_w_id, c_discount, c_last, c_credit)
as select w.w_id, w.w_tax,
      c.c_id, c.c_d_id, c.c_w_id, c.c_discount, c.c_last, c.c_credit
from cust c, ware w
where w.w_id = c.c_w_id;

create or replace view wh_dist
(w_id, d_id, d_tax, d_next_o_id, w_tax )
as select w.w_id, d.d_id, d.d_tax, d.d_next_o_id, w.w_tax
from dist d, ware w
where w.w_id = d.d_w_id;

create or replace view stock_item
(i_id, s_w_id, i_price, i_name, i_data, s_data, s_quantity,
s_order_cnt, s_ytd, s_remote_cnt,
s_dist_01, s_dist_02, s_dist_03, s_dist_04, s_dist_05,
s_dist_06, s_dist_07, s_dist_08, s_dist_09, s_dist_10)
as
select /*+ leading(s) use_nl(i) */
i.i_id, s.w_id, i.i_price, i.i_name, i.i_data, s_data, s_quantity,
s_order_cnt, s_ytd, s_remote_cnt,
s_dist_01, s_dist_02, s_dist_03, s_dist_04, s_dist_05,
s_dist_06, s_dist_07, s_dist_08, s_dist_09, s_dist_10
from stok s, item i
where i.i_id = s.s_i_id;

set echo off;

```

views.sql

```
connect tpcc/tpcc;
```


Appendix C Tunable Parameters

The HP-UX operating system tunable parameters employed to generate the kernel for the HP Integrity Superdome - Dual-Core Itanium2/1.6GHz/24MB iL3 - 64p/128c and the 96 HP 9000 Model rx1620 clients are listed below. Included as well are the Oracle Database 10g Enterprise Edition Release2 and TUXEDO 8.1 parameters.

C.1 HP-UX Configuration - Clients

Config/Client2/ostune.ver

```
*
* Created on Fri Feb 2 16:49:03 2007
*
version 1
configuration nextboot "" [45c3cc9e]
*
* Module entries
*
module gvid_him_cons auto 0.1.0
module drmfgr auto 0.1.0
module drmfgr auto 0.1.0
module gvid_him_fgl auto 0.1.0
module gvid_info loaded 0.1.0
module root best [413F2E8E]
module sba best [413F2ED5]
module lba best [413F2E92]
module side_multi best [413F2ED6]
module side best [413F2ED6]
module tgt best [413F2ED6]
module sdisk best [413F2ED6]
module sctl best [413F2ED6]
module asio0 best [413F2E8F]
module gvid_core best [413F2E92]
module azusa_psm best [413F2E8E]
module pty0 best [413F2E95]
module pty1 best [413F2E95]
module LCentf best [413F2E8F]
module acpi_node best [413F2E8E]
module sac best [413F2E8F]
module wxb_lip best [413F2E8F]
module ia64_psm best [413F2E8E]
module lion_psm best [413F2E8E]
module pdh best [413F2E8E]
module c8xx best [413F2ED6]
module diag2 best [413F2ED6]
module dmem best [413F2E93]
module dev_config best [413F2ED5]
module cdfs best 0.1.0
module rng loaded 0.1.0
module inet best [413F2EA3]
module uipc best [413F2E94]
module tun best [413B744]
module telm best [412E8D79]
module tels best [412E8D79]
module netdiag1 best [413F2EDB]
module btlan best [412E8A46]
module intl100 best [412E8A84]
module dlpi best [412E9113]
module token_arp best [412E9113]
module nms best [413F2EAE]
module hpstreams best [412E9162]
module clone best [412E9162]
module strlog best [412E9162]
module sad best [412E9162]
module echo best [412E9162]
module sc best [412E9162]
module timod best [412E9162]
module tirdwr best [412E9162]
module pipedev best [412E9162]
module pipemod best [412E9162]
module ffs best [412E9162]
module ldterm best [413F2EB5]
module ptem best [413F2EB5]
module pts best [413F2EB5]
module ptm best [413F2EB5]
module pckt best [413F2EB5]
module nfs_core best [412E8CC1]
```

```
module nfs_server best [412E8CC4]
module nfs_client best [412E8CC1]
module nfsm best [412E8CC4]
module rpcmod best [412E8C55]
module autofsc best [412E92E1]
module cachefsc best [412E936B]
module cifs best [410AC7FA]
module td best [412342AF]
module fcd best [4134EEB7]
module fcd_fcp best [4134EEB7]
module fcd_vbus best [4134EEB7]
module fddi4 best [41237311]
module gelan best [4122D39E]
module iether best [4122D3DB]
module igelan best [413369E8]
module vxfs best [413F2EBF]
module vxportal best [413F2EBF]
module lvm best [413F2E9E]
module lv best [413F2E9E]
module ipmi best [413F2E8F]
module ipmi_psm best [413F2E8F]
module mip6mod best [412E9271]
module asyncdsk best [413F2ED9]
module ciss best [4122361B]
module hed best [413F2ED7]
module hub best [413F2ED7]
module hid best [413F2ED7]
module UsbMiniBus best [4120D929]
module UsbOhci best [4120D928]
module vxvm best [41258D12]
module vxdump best [4121E998]
module vol best [41258D12]
module vols best [41258D12]
module mpt best [41223F81]
module pfil auto 0.1.0
*
* Swap entries
*
*
* Dump entries
*
dump lvol
*
* Driver binding entries
*
*
* Tunables entries
*
tunable swapmem_on 1
tunable bufpages 8192
tunable vx_ninode 20000
tunable secure_sid_scripts 0
tunable vxfs_bc_bufhwm 6144
tunable vxfs_ifree_timelag 0x0000000
tunable default_disk_ir 1
tunable fs_async 1
tunable maxfiles 2048
tunable maxfiles_lim 2048
tunable maxdsiz 0x80000000
tunable nkthread 62000
tunable msgmni 3875
tunable nfile 62000
tunable semvmx 62000
tunable shmmax 0x40000000
tunable maxdsiz_64bit 4294967296
tunable aio_proc_thread_pct 70
tunable cmc_plat_poll 15
tunable nstrpty 200
tunable create_fastlinks 1
tunable hfs_max_ra_blocks 20
tunable hfs_max_revra_blocks 20
tunable hfs_ra_per_disk 256
tunable hfs_revra_per_disk 256
tunable max_thread_proc 1030
tunable maxssiz 0x10000000
tunable maxtsiz 1073741824
tunable maxuprc 1024
tunable maxvgs 80
tunable msgmap 32000
tunable msgmax 32768
tunable msgmnb 67108864
tunable msgseg 32767
tunable msgssz 2048
tunable msgtql 65536
tunable ninode 1024
tunable npty 128
tunable nswapdev 25
tunable semmni 32
tunable semmns 8192
tunable semmnu 4092
tunable semume 4
tunable shmuni 16
tunable shmseg 16
tunable STRMSGSZ 65535
tunable vps_ceiling 4
```

tunable dbc_min_pct 1
tunable dbc_max_pct 5

C.2 HP-UX Configuration – Server

Config/Server/ostune.ver

```

*
* Created on Mon Feb 19 11:43:16 2007
*
version 1
configuration nextboot "jazz @ 20070112.14:00:29PST; jmkvw -proj kern1 -RW -c Task:
ruemmler_i80_io_master10 ruemmler_i80_io_master10 i80(I80.BL2006_1230)
cup2_ruemmler_i80_io_master10(auto); FLAVOR=perf " [45d9ea72]
*
* Module entries
*
module iospy loaded 0.1.[45D0EA20]
module asio0 best 1.0.[45975AB7]
module cell best 1.0.[4580B501]
module lha best 1.0.[459757F5]
module pci_slot best 1.0.[4580B277]
module root best 1.0.[4580B191]
module sba best 1.0.[45975667]
module inet best 1.0.[459758C4]
module uipc best 1.0.[4597557E]
module tun best 1.0.[45975572]
module telm best 1.0.[459755C0]
module tels best 1.0.[4580B084]
module netdiag1 best 1.0.[45975792]
module dlpi best 1.0.[45975A30]
module token_arp best 1.0.[45975593]
module btlan best 1.0.[45975A93]
module intl100 best 1.0.[459758DA]
module nms best 1.0.[4597577F]
module hpstreams best 1.0.[45975956]
module clone best 1.0.[4580B4FB]
module strlog best 1.0.[459755FC]
module sad best 1.0.[4597565F]
module echo best 1.0.[459759EC]
module sc best 1.0.[4597564E]
module timod best 1.0.[459755A5]
module tirdwr best 1.0.[459755A2]
module pipedev best 1.0.[4580B25A]
module pipemod best 1.0.[45975732]
module ffs best 1.0.[4597599C]
module ptym best 1.0.[4597568B]
module pty5 best 1.0.[4580B1A5]
module ipmi best 1.0.[45975875]
module acpi_node best 1.0.[45975AD7]
module sac best 1.0.[4597565E]
module wxb_hp best 1.0.[45974CA6]
module ia64_psm best 1.0.[459758F6]
module pdh best 1.0.[4597574E]
module rmp3f01 best 1.0.[45975670]
module procsm best 1.0.[4597569E]
module c8xx best 1.0.[45975AB2]
module side best 1.0.[45975637]
module side_multi best 1.0.[4597561E]
module eschgr best 1.0.[459DD551]
module esctl best 1.0.[459DD583]
module esdisk best 1.0.[459DD567]
module eslpt best 1.0.[4580B472]
module estape best 1.0.[459DD55C]
module estp best 1.0.[4580B46F]
module esvroot best 1.0.[4580B46E]
module schgr best 1.0.[4580B16E]
module sctl best 1.0.[4580B16C]
module sdisk best 1.0.[4580B169]
module stape best 1.0.[4580B129]
module tgt best 1.0.[4580B032]
module asyncdsk best 1.0.[45975AA9]
module diag2 best 1.0.[45975A36]
module dnmem best 1.0.[459759F6]
module dev_config best 1.0.[45A804D2]
module cdfs auto 0.1.[45975A6C]
module rng loaded 0.1.[45975671]
module lterm best 1.0.[459757E7]
module ptem best 1.0.[45975688]
module pts best 1.0.[4580B1AA]
module ptm best 1.0.[45975690]
module pckt best 1.0.[4597573E]
module nfswrp best 1.0.[459758A2]
module nfsrv best 1.0.[459758C8]
module nfs_client_pv4 best 1.0.[459758AE]
module nfs_client_pv3 best 1.0.[459758B0]

```

```

module nfs_client_pv2 best 1.0.[459758B6]
module nfs_client best 1.0.[4597593A]
module rpc best 1.0.[459758A4]
module rpcsec best 1.0.[45975889]
module rpcsec_gss best 1.0.[45975789]
module klmmmod best 1.0.[459758FF]
module kgssapi best 1.0.[4597590C]
module krb5 best 1.0.[459758F7]
module rpcmod best 1.0.[45975888]
module autofsd best 1.0.[4597591C]
module cachefs best 1.0.[45975942]
module fcp best 1.0.[459759A0]
module fcpararray best 1.0.[4597599C]
module fcpdev best 1.0.[4597599B]
module td best 1.0.[45974AFE]
module fcd best 1.0.[459750A2]
module gelan best 1.0.[45974BC3]
module iether best 1.0.[45974EF8]
module igelan best 1.0.[45974DF7]
module ixgbe best 1.0.[4597695B]
module vxfs static 41.0.[459765E3]
module vxportal static 41.0.[45976463]
module lvm best 1.0.[4597602C]
module oncsupp best 1.0.[45975761]
module prm best 1.0.[459756A7]
module ciss best 1.0.[45975F55]
module sasd best 1.0.[45976AFA]
module vxdump best 1.0.[4597531C]
module vol best 1.0.[4597540E]
module vols best 1.0.[459752F4]
module dmpaabbest 0.1.[45975321]
module dmpaaa best 0.1.[4597531F]
module dmpapbest 0.1.[4597531B]
module dmpapp best 0.1.[4597530F]
module dmpapf best 0.1.[45975315]
module dmpjbd best 0.1.[459752F5]
module dmphdalua best 0.1.[45975305]
module dmpupalua best 0.1.[459752FF]
module mpt best 1.0.[45974FCE]
*
* Dump entries
*
dump lvvol
*
* Tunables entries
*
tunable vx_ninode 146208
tunable lockable_mem_pct 99%
tunable vps_chattr_ceiling 4194304
tunable filecache_max 640MB
tunable filecache_min 640MB
tunable process_id_max 99999
tunable maxfiles_lim 8192
tunable nkthread 20000
tunable process_id_min 0
tunable pagezero_daemon_enabled 0
tunable nstrpty200
tunable vps_ceiling 64
tunable semmni 4096
tunable semmns 8192
tunable semmnu 4092
tunable semume 512
tunable shmmni 512
tunable shmseg 512
tunable nswapdev 10
tunable npty 200
tunable user:msgmap 5122
tunable user:msgmax 32768
tunable msgmnb 65536
tunable user:msgseg 20480
tunable user:msgsz 128
tunable msgtql5120
tunable maxuprc 6000
tunable user:maxvgs 40
tunable maxssiz 100610048
tunable maxtsiz 1073741824
tunable hfs_max_ra_blocks 20
tunable hfs_max_revra_blocks 20
tunable hfs_ra_per_disk 256
tunable hfs_revra_per_disk 256
tunable create_fastlinks 1
tunable maxtsiz_64bit 4294967296
tunable maxssiz_64bit 1073741824
tunable maxdsiz_64bit 274877906944
tunable maxdsiz 3221225472
tunable shmmxax 0x20000000000
tunable nfile 10000000
tunable secure_sid_scripts 0
tunable semmsl 128
tunable vxfs_ifree_timelag 3600000
tunable user:unlockable_mem 1
tunable swchunk 65536
tunable user:swapmem_on 0
tunable STRMSGSZ 65535
tunable o_sync_is_o_dsync 1

```

```

tunable perfmon_allow_user_per_cpu      1
tunable user:scsi_maxphys      8388608
tunable as_isolation_level      1
tunable default_disk_ir      1
tunable max_mem_window      10
tunable remote_nfs_swap      1
tunable intr_strobe_ics_pct      95
tunable expanded_node_host_names      1
tunable nproc      6700
tunable maxfiles      8192
tunable lcpu_attr      1
tunable max_thread_proc      2048

```

Config/Server/dbtune.ver

```

compatible = 10.1.0.0.0
db_name = tpcc
control_files = (/BUILD/dbs/tpcc_disks/control_001,/BUILD/dbs/tpcc_disks/control_002)
parallel_max_servers=600
recovery_parallelism = 512
db_files = 1900
db_cache_size = 1G
db_2k_cache_size = 60G
db_8k_cache_size = 10G
db_16k_cache_size = 476G
db_keep_cache_size = 1220G
db_recycle_cache_size = 16G
dml_locks = 500
statistics_level = basic
log_buffer = 67108864
processes = 5650
sessions = 6250
transactions = 5650
shared_pool_size = 45G
cursor_space_for_time = TRUE
db_block_size = 4096
undo_management = auto
undo_retention = 1
plsql_optimize_level=2
UNDO_TABLESPACE = undo_1
HPUX_SCHED_NOAGE=180
fast_start_mttr_target = 0
java_pool_size = 0
job_queue_processes = 0
lock_sga = TRUE
log_checkpoint_interval = 0
log_checkpoint_timeout = 0
log_checkpoints_to_alert = true
pga_aggregate_target = 0
query_rewrite_enabled = false
replication_dependency_tracking = false
timed_statistics = false
trace_enabled=false
db_block_checking=false
db_block_checksum = false
aq_tm_processes = 0
disk_asynch_io = true
max_dump_file_size = 20

```

C.3 Tuxedo UBBconfig

Config/Client2/tmcfgr.ver

```

# This is a UBBconfig for a client1-server configuration.
#
# This UBBconfig requires settings for:
# SERVER_NAME CLIENT_NAME MASTER_NAME SERVER_ADDR
# CLIENT_ADDR NODE_NAMES
# TLISTEN_PORT TBRIDGE_PORT
# In addition, it requires setting the things all UBBconfig.gens need:
# IPCKEY some decent IPCKEY, should be different for
# each config
# ROOTDIR
# TUXCONFIG
# APPDIR
# ULOGDIR
#
#-----
*RESOURCES
#-----
IPCKEY 40001
PERM 0666

```

```

MASTER plebe113
MAXACCESSERS 2550 # num_users + 50
MAXGTT 1024
MAXSERVERS 56 # num_servers + 5
MAXSERVICES 265 # num_servers * #-of-services-each-server(5) + 10( for
BBL)
MODEL SHM
LDBAL N
OPTIONS NO_AA,NO_XA

```

```

# During benchmark, don't want to scan too often. In particular, while
# the client1s are stabilizing in virtual memory, we don't want to sanity
# scan; and if we do sanity scan, we want large timeouts, since the BRIDGE
# the BBL, the DBBL, and the client1s aren't getting much CPU time during that
# period. Current settings:
# * scan servers every 5 minutes (maximum allowed by TUXEDO);
# * wait 1 minute for sanity responses (maximum allowed by TUXEDO);
# * scan all the BBLs from DBBL every 30 minutes (want one scan in the
# audited results);
# * timeout a blocking call after 5 minutes (the maximum).

```

```

SCANUNIT 60
SANITYSCAN 5
DBBLWAIT 1
BBLQUERY 30
BLOCKTIME 5
#-----
*MACHINES
#-----
DEFAULT:
TUXCONFIG="/project/tpcc/tuxconfs/TUXconfig.plebe113"
ROOTDIR="/project/tuxedo"
APPDIR="/project/tpcc/bin"
ULOGPFX="/tmp/TUXEDO_LOG"

```

```

# for debugging, put both into the same log on the same machine
# ULOGPFX="/home/tuxedo/confs/tpcc/ULOG"
# but for a big run, need some space, and want them local to the
# machine rather than across the net.

```

```

# Leave TUXCONFIG alone on the MASTER machine; over-ride for each
# other machine?

```

```

plebe113 LMID=plebe113
TUXCONFIG="/project/tpcc/tuxconfs/TUXconfig.plebe113"
#-----

```

```

*GROUPS
#-----
group1 LMID=plebe113
GRPNO=1

```

```

#-----
#-----

```

```

#-----
*SERVERS
#-----
#
# "-" is application-specific arguments to be passed to server
# "n" is designed to specify server-id

```

```

service SRVGRP=group1
CLOPT="-s NEWO_SVC -s PMT_SVC -s ORDS_SVC -s STKL_SVC -s
DVRV_SVC"
MIN=51 MAX=51 RQADDR=tpcc_1 REPLYQ=Y SRVID=1
#-----

```

```

*SERVICES
#-----
*ROUTING
#-----

```

Appendix D RTE Configuration

This appendix lists RTE input parameters and code fragments used to generate each transaction input file, to demonstrate the RTE was configured to generate transaction input data as specified in *Clause 2* of the specification.

D.1 Field Value Generation

Source/src/driver/generate.c

```
*****
@(#) Version: A.10.10 $Date: 2005/04/11 09:40:08 $

(c) Copyright 1996, Hewlett-Packard Company, all rights reserved.
*****

#include <stdio.h>
#include <stdlib.h>
#include <values.h>
#include <unistd.h>
#include <time.h>
#include <sys/types.h>
#include <sys/ipc.h>
#include <fcntl.h>
#include <signal.h>
#include <math.h>
#include <string.h>

#include "random.h"
#include "shm_lookup.h"

#include <time.h>

int CLAST_CONST_C = 208;
int CID_CONST_C = 37;
int IID_CONST_C = 75;

int trans_type = 0; /* type of transaction 0 == all */

void
neworder_gen(neworder_trans *t, ID warehouse, ID district)
{
    int i;

    t->W_ID = warehouse;

    t->D_ID = RandomNumber(1, no_dist_pw);
    t->C_ID = NURandomNumber(1023, 1, no_cust_pd, CID_CONST_C);

    t->O_OL_CNT = RandomNumber(5, 15);

    for (i=0; i<t->O_OL_CNT; i++)
    {
        t->item[i].OL_I_ID = NURandomNumber(8191, 1, no_item, IID_CONST_C);
        t->item[i].OL_SUPPLY_W_ID = RandomWarehouse(warehouse, db_size, 1);
        t->item[i].OL_QUANTITY = RandomNumber(1, 10);
    }

    /* Zero out the non-used items as the oracle driver does. */
    for (; i< 15; i++)
    {
        t->item[i].OL_I_ID = 0;
        t->item[i].OL_SUPPLY_W_ID = 0;
        t->item[i].OL_QUANTITY = 0;
    }

    /* 1% of transactions roll back. Give the last order line a bad item */
    if (RandomNumber(1, 100) == 1)
        t->item[t->O_OL_CNT - 1].OL_I_ID = -1;
}

void
payment_gen(payment_trans *t, ID warehouse, ID district)
{
    /* home warehouse is fixed */
```

```
t->W_ID = warehouse;

/* Random district */
t->D_ID = RandomNumber(1, no_dist_pw);

/* Customer is from remote warehouse and district 15% of the time */
t->C_W_ID = RandomWarehouse(warehouse, db_size, 15);
if (t->C_W_ID == t->W_ID)
    t->C_D_ID = t->D_ID;
else
    t->C_D_ID = RandomNumber(1, no_dist_pw);

/* by name 60% of the time */
t->byname = RandomNumber(1, 100) <= 60;
if (t->byname)
    LastName(NURandomNumber(255, 0, no_cust_pd/3 - 1, CLAST_CONST_C),
             t->C_LAST);
else
    t->C_ID = NURandomNumber(1023, 1, no_cust_pd, CID_CONST_C);

/* amount is random from [1.00..5,000.00] */
t->H_AMOUNT = RandomNumber(100, 500000);
}

void
ordstat_gen(ordstat_trans *t, ID warehouse, ID district)
{
    /* home warehouse is fixed */
    t->W_ID = warehouse;

    /* district is randomly selected from warehouse */
    t->D_ID = RandomNumber(1, no_dist_pw);

    /* by name 60% of the time */
    t->byname = RandomNumber(1, 100) <= 60;
    if (t->byname)
        LastName(NURandomNumber(255, 0, no_cust_pd/3 - 1, CLAST_CONST_C),
                 t->C_LAST);
    else
        t->C_ID = NURandomNumber(1023, 1, no_cust_pd, CID_CONST_C);
}

void
delivery_gen(delivery_trans *t, ID warehouse, ID district)
{
    t->del.W_ID = warehouse;
    t->del.O_CARRIER_ID = RandomNumber(1,10);
}

void
stocklev_gen(stocklev_trans *t, ID warehouse, ID district)
{
    t->W_ID = warehouse;
    t->D_ID = district;
    t->threshold = RandomNumber(10, 20);
}

int get_trans_type()
/******
 * get_trans_type selects a transaction according to the weighted average
 * For TPC-C rev 3.0 and less and TPC-C rev 3.2 this is:
 * new-order : ???
 * payment : 43.0%
 * order stat: 4.0%
 * delivery : 4.0%
 * stock : 4.0%
 *****/
{
    static const double weight[] = { 0.0, 0.0, .4301, .0401, .0402, .0401};
    int type;
    double r;

    /* choose a random number between 0.0 and 1.0 */
    if (trans_type == 0) {
        r = RandomValue();

        /*
         * select one of STOCKLEV, DELIVERY, ORDSTAT and PAYMENT
         * based on weight
         */
        for (type = STOCKLEV; type > NEWORDER; type--) {
            r -= weight[type];
            if (r < 0) break;
        }
    } else if (trans_type > 0) {
        /* user wants only a certain type (say all stocklevel) so do that
         instead */
        type = trans_type;
    }

    /* return the value of the selected card, or NEWORDER if none selected */
    return type;
}
```

Appendix E Disk Storage

The calculations for the 8 hours recovery log were based on how often the oracle redo log files were filling up and needed to be switched. The database took a checkpoint, and switched from the "current" log file to the other, "active" log file, every 29 minutes, 27 seconds during the steady state portion of the run. Each log file is 650000MB.

So, to run for 8 hours, we need:

$$((8 * 60) / 29.45) * 650000 / 1024 = 10,345.93 \text{ GB (must be mirrored)}$$

The priced log disk arrays have a total of $8 * 1899.28 = 15194.24$ GB (mirrored) available for the 8-hour recovery log.

TPC-C 60-Day Space Requirements

TPM		4,092,799.42					
Warehouses		349,440					
SEGMENT	TYPE	TSPACE	BLOCKS	FIVE_PCT	DAILY_GROWTH	BLOCK_SIZE	TOTAL in MB
CUSTCLUSTER	CLUSTER	CUST_0	2,453,422,080	122,671,104	0	4,096	10,062,864
DISTCLUSTER	CLUSTER	MISC_0	3,744,552	187,228	0	2,048	7,679
HIST	TABLE	HIST_0	157,613,568	0	29,536,660	4,096	731,056
ICUST1	INDEX	ICUST1_0	15,148,800	757,440	0	16,384	248,535
ICUST2	INDEX	ICUST2_0	32,290,560	1,614,528	0	16,384	529,767
IDIST	INDEX	MISC_0	437,056	21,853	0	4,096	1,793
IITEM	INDEX	MISC_0	2,816	141	0	4,096	12
IORDR2	INDEX	IORDR2_0	22,863,360	1,143,168	0	16,384	375,102
ISTOK	INDEX	ISTOK_0	43,902,720	2,195,136	0	16,384	720,279
ITEMCLUSTER	CLUSTER	MISC_0	8,446	422	0	2,048	17
IWARE	INDEX	MISC_0	109,456	5,473	0	4,096	449
NORDCLUSTER	TABLE	NORD_0	38,922,240	1,946,112	0	2,048	79,821
ORDRCLUSTER	TABLE	ORDR_0	540,094,464	0	101,213,282	16,384	10,020,434
STOKCLUSTER	CLUSTER	STOK_0	2,498,929,664	124,946,483	0	4,096	10,249,516
SYSTEM	SYS	SYS	102,400	0	0	4,096	400
WARECLUSTER	CLUSTER	MISC_0	374,762	18,738	0	2,048	769
SYSAUX	SYS	SYSAUX	30,720	0	0	4,096	120
Benchmark stats	SYS	SYS	524,288	0	0	4,096	2,048
SYS_IQ0000008885\$	INDEX	ORDR_0	2,515,968	125,798	0	16,384	41,278
SYS_IQ0000008889\$	INDEX	NORD_0	860,160	43,008	0	2,048	1,764
ROLL_SEG	SYS	UNDO_TS	524,288	0	0	8,192	4,096
Total							33,077,797

Dynamic space	9,054,654	Initial MB for (History+Orders)
Static space	22,326,308	Initial Blocks + 5% - Dynamic
Daily Growth	1,696,835	Total Dynamic [(calc. as (Initial Blocks)*tpmC/(WHS*62.5)]
Daily Spread	-	Oracle may be configured so that daily spread is 0
60-day space (MB)	124,136,415	Static + 60*(Daily Growth+Daily Spread)
60-day (GB)	121,227	Excludes OS, Paging and RDBMS Logs
8-hour log (GB)	10,346	RDBMS Logs
Server swap (GB)	297	OS: Paging
Server OS (GB)	19	OS: UNIX File System
Total Space Needed	131,889	GB

Priced-Sytem Configuration	Size in GB	Quantity	Total (GB)
Log arrays: MSA1500 with 56 72.8GB disk drives in RAID 1 mode	1,899.3	8	15,194.4
Data arrays: MSA1500 with 56 36.4GB disk drives in RAID 0 mode	1,899.2	75	142,440.0
Data arrays: MSA1500 with 56 73GB disk drives in RAID 0 mode	3,798.6	43	163,339.8
Total Storage in Priced System (GB)			320,974.2

Appendix F Price Quotes

The following pages contain the price quotes for the hardware included in this FDR.

TO: Curt Thiem
 HP
 Cupertino, CA 95014
 2/26/2007



HP Unix Sales Development
 19111 Pruneridge Avenue
 Cupertino, CA 95014
 (408) 447-2320

Description	Part Number	Brand	Price Key	US List Price	Qty	Price	3Year Main.Price
Server Hardware							
Superdome left chassis	A9834A, Opt 429		1	235,950	1	235,950	
Superdome right chassis	A9835A, Opt 429		1	249,950	1	249,950	
Superdome sx2000 Cell Board	A9837A		1	19,250	16	308,000	
3 Year Svc & Support Price (Hardware and Software)							\$1,376,539
Dual-Core Intel Itanium2 9050/1.6GHz/24MB L3	A9840A		1	31,500	64	2,016,000	
512GB Memory Bundle (128x4GB dimms)	A9857A		1	1,599,078	4	6,396,312	
12-Slot PCI-X I/O Chassis	A9836A		1	16,950	16	271,200	
I/O Chassis Enclosure for 12-Slot PCI-X Chassis	A9852A		1	25,750	4	103,000	
Graphite I/O expansion power subsystem	A5861D		1	34,860	2	69,720	
HP Surestore Disk System 2120	A7382A		1	995	1	995	
36GB LP 15K HDD	A7527A		1	966	1	966	
HP Universal Rack10642 G2 Pallet Rack	AF001A		1	1,249	1	1,249	
Modular Power Dist Unit for std racks	A5137AZ		1	145	1	145	
200-240 volts North America	A5137AZ, Opt AW4		1	94	1	94	
PCI-X 133MHz 10GbE SR Fiber Adapter	AB287A		1	5,795	4	23,180	
PCI-X Dual Ultra320 SCSI Adapter	A7173A		1	795	1	795	
PCI-X 4Gb Fibre Channel Adapter (dual port)	AB379B		1	3,495	40	139,800	
HP DVD + RW Array Field Module	Q1592B		1	649	1	649	
HP rx2620 Server Solution (inc memory, disk, monitor,	AB332A		1	5,315	1	\$5,315	
				Subtotal		9,823,320	1,376,539
Server Software							
HPUX 11i v3 Foundation Operating Environment	B9429AC		1	2,370	128	303,360	
HPUX 11i v3 HP9000 Integrity FOE Media	BA489AA, Opt AJR		1	565	1	565	
				Subtotal		303,925	0
Storage							
Rack System/E R3000 XR UPS	192186-001		1	1,365	1	1,365	
5 meter Fibre Optic Cable	221691-B22		1	82	346	28,372	
15 meter Fibre Optic Cable	221691-B23		1	103	30	3,090	
HP StorageWorks MSA 1500 w/MSA30 Enc (126+ 13 s	AD510A		1	8,995	139	1,250,305	
HP StorageWorks Enclosures MSA30 (378+38 spares	302969-B21		1	2,829	416	1,176,864	
HP StorageWorks MSA 1000 Controller (126+13 spares	218231-B22		1	4,290	139	596,310	
MSA1500 Fibre Channel I/O Module 126+13 spares)	AA987A		1	375	139	52,125	
MSA1500 Dual Channel SCSI I/O Module (378 + 38 sp	AA988A		1	525	416	218,400	
3 Yr Support Price for MSA1000, MSA30, disks, switches							INCLUDED
73GB 15K Ultra320 Hard Drive (2856 + 286 spares)	286778-B22		1	439	3142	1,379,338	
36GB 15K Ultra320 Hard Drive (4200 + 420 spares)	286776-B22		1	319	4620	1,473,780	
HP StorageWorks 4/64 Full SAN Switch	AG457A		1	62,999	2	125,998	
HP StrgWrks4gbSW SnglPK SFP Transcvr	A7446B		1	199	124	24,676	
HP StrgWrks2gbSW SnglPK SFP Transcvr	221470-B21		1	199	296	58,904	
HP StorageWorks SAN Switch 4/8	A8000A		1	4,999	60	299,940	
HP Universal Rack10642 G2 Pallet Rack	AF001A		1	1,249	44	54,956	
				Subtotal		6,744,423	0
Client Hardware							
HP Integrity rx1620 w 1.3GHz Intel Itanium 2 Processor	AB430A		1	3,445	95	327,275	
1.3GHz Intel Itanium 2 Processor	AB481A		1	1,650	95	156,750	
3 Year Support Price (Hardware & Software)							444,192
36GB 15K RPM Ultra320 SCSI Internal Disk	AB420A		1	389	95	36,955	
2GB Memory Module (2 x 1GB DIMMS)	AB223A		1	1,275	380	484,500	
HP Server Thin Client (monitor, keyboard/mouse, cable	AB300B		1	1,250	1	1,250	
HP Universal Rack 10642 G2 Shock Rack	AF001A		1	1,249	4	4,944	
Modular Power Dist.	A5137AZ		1	145	16	2,296	
200-240 Volts Power Option	A5137AZ, Opt AW4		1	94	16	1,488	
						565	
HP-UX Fndn OE DVD Media	B9106AA, Opt. AJR		1	565	1		
						199	
HP-UX Fndn OE DVD Media, Factory Integrated	B9106AA, Opt. OD1		1	199	1	199	
HP-UX Fndn OE Integrity PPL max2CPU w/sys	B9430AC		1	455	190	86,450	
				Subtotal		1,102,672	444,192
Client Software							
HP C++/ANSI C Developer's Bundle	B9007AA, Opt. 2AH		1	966	1	966	170
				Subtotal		966	170
User Connectivity							
HP ProCurve Switch 2724 (24-port)	J4897A		1	1,599	4	6,396	
HP ProCurve Switch 3400cl-24G	J4905A		1	3,795	2	7,590	
ProCurve 10-GbE X2-SC SR Optic transceivers	J8436A		1	2,999	4	11,996	
HP ProCurve 10GbE Meida Flex Module	J8435A		1	2,699	2	5,398	
				Subtotal		31,380	0
						(\$8,292,394)	(\$926,397)
HP's Large configuration Discount and Support Prepayment*							
				Total		9,714,292	894,504

This quote is valid for 90 days

Sent: Friday, February 09, 2007 3:31 PM
 To: Boushey, Lucille
 Subject: Revised TPC-C quote

Dear Lucille,

Please find the attached Oracle quote:

Product	Price	Quantity	Extended Price
Oracle Database 10g Enterprise Edition Release2, Processor License for 3 years	20,000	64**	\$1,280,000
Partitioning	5,000	64**	\$320,000
Oracle Database Server Support Package for 3 years	2,000	3	\$6,000
Oracle Mandatory E-Business Discount (license and support)*			<\$401,500>
Oracle Total			\$1,204,500

(*) Oracle pricing contact: Ravi Rajamani, ravi.rajamani@oracle.com, (650) 506 5776
 (**) 64 = 128 x 0.50. Explanation: for the purposes of counting the number of processors which require licensing for Intel multicore chips, “n” cores shall be determined by multiplying the total number of cores by a core processor licensing factor of .50.

This quote is valid for 60 days.

February 5, 2007

**Lucille Boushey
Hewlett Packard Company
(408) 447-7364; Phone
(408) 447-5958; Fax**

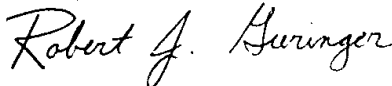
Per your request I am enclosing the pricing information regarding TUXEDO 8.0 that you requested. This pricing applies to Tuxedo 6.4, 6.5, 7.1,8.0,8.1,9.0, and 9.1. Please note that Tuxedo 9.0 is our most recent version of Tuxedo. Core functionality services (CFS)-R pricing is appropriate for your activities. As per the table below HP/Compaq systems are classified as either a Tier 1, 2, 3, 4 or 5 systems depending on the performance and CPU capacity of the system. The HP RX 1620 machines are Tier 1 machines – price is \$1,200 per server (License), eligible for a 10% discount = \$1,080 per server + \$252 per server (7x24) for support – support is non discountable. This quote is valid for 90 days from the date of this letter.

Tuxedo Core Functionality Services (CFS-R) Program Product Pricing and Description

TUX-CFS-R provides a basic level of middleware support for distributed computing, and is best used by organizations with substantial resources and knowledge for advanced distributed computing implementations.

TUX-CFS-R prices are server only and are based on the overall performance characteristics of the server and uses the same five tier computer classification as TUXEDO 6.4, 6.5,7.1,8.0, 8.1,9.0, and 9.1. Prices range from \$1,200 for Tier 1 to \$100,000 for Tier 5. Under this pricing option EVERY system running TUX-CFS-R at the user site must have a TUXEDO license installed and pay the appropriate per server license fees.

Very Truly Yours,



**Rob Gieringer,
Worldwide Pricing Manager**

BEA Tux/CFS-R Unlimited User License Fees Per Server

Unlimited User License fees per server	Number of Users	Dollar Amount	Maintenance (5 x 9) per year	Maintenance (7 x 24) per year
Tier 1 -- PC Servers with 1 or 2 CPUs, entry level RISC Uni-processor workstations and servers	Unlimited	\$1,200.00	\$216	\$252
Tier 2 - PC Servers with 3 or 4 CPUs, Midrange RISC Uni-processor servers and workstations with up to 2 CPUs	Unlimited	\$4,800.00	\$864	\$1,008
Tier 3 - Midrange Multiprocessors, up to 8 CPUs per system capacity	Unlimited	\$12,000.00	\$2,160	\$2,520
Tier 4 - Large (more than 8, less than 32 CPUs)	Unlimited	\$40,000.00	\$7,200	\$8,400
Tier 5 - Massively Parallel Systems, > 32 processors	Unlimited	\$100,000.00	\$18,000	\$21,000

	Tier 1	Tier 1	Tier 2	Tier 3	Tier 4	Tier 5
Operating System						
HP/UX 9.X;10.X	Uni-processor Workstation B Class - 132/180/2000 C Class (3000/3600 / 3700) 2P Client Machines Compaq DL360	9000/E25 9000/E35 9000/E45 9000/E55 9000/G30 9000/G40 9000/A180 9000/A180C 9000/A400 RX 2600 RX 2620 RX 1620	9000/G50 9000/G60 Multi-Processor Workstations J Class (J282/J2240/J5600/J6000/J6700) 9000/R380,390 9000/D200,210 220/30/50/60/80 D310/20/30 D350/60/70/80 9000 /A500 9000 – L1000 9000 – R Class	9000/H20, 30 9000/H40, 50 9000/I30, 40 9000/K1XX 9000 – L2000/L3000 9000/I50,60 9000/H60 9000/G70 9000/H70 9000/I70 9000/K2XX 9000/K3XX 9000/K4XX 9000/K5XX N4xxx Series	9000/T500, T520, T600 1-16 CPUs S-Class RP8400 Superdome < 32 CPU's RP8420 SuperDome (PA-8800 bases) w/ < 32 dual-core CPU's	9000/V series all models X-Class Superdome >= 32 CPU's SuperDome (PA-8800 bases) w/ >= 32 dual-core CPU's

