TPC Benchmark™ H
Full Disclosure Report

ThinkServer RD630
VectorWise 3.0.0
RedHat Enterprise Linux 6.4

First Edition

May 2013

First Edition – May, 2013

Lenovo Corp., the sponsor of this benchmark test, 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. The sponsor assumes no responsibility for any errors that may appear in this document. The pricing in this document is believed to accurately reflect the current prices as of the date of publication. However, the sponsor provides no warranty of the pricing information in this document.

Vectorwise is a trademark of Actian. Intel and Intel Xeon are trademarks or registered trademarks of Intel Corporation. TPC Benchmark, TPC-H, QphH, and $/QphH are trademarks of Transaction Processing Performance Council. All other trademarks and copyrights are property of their respective owners.

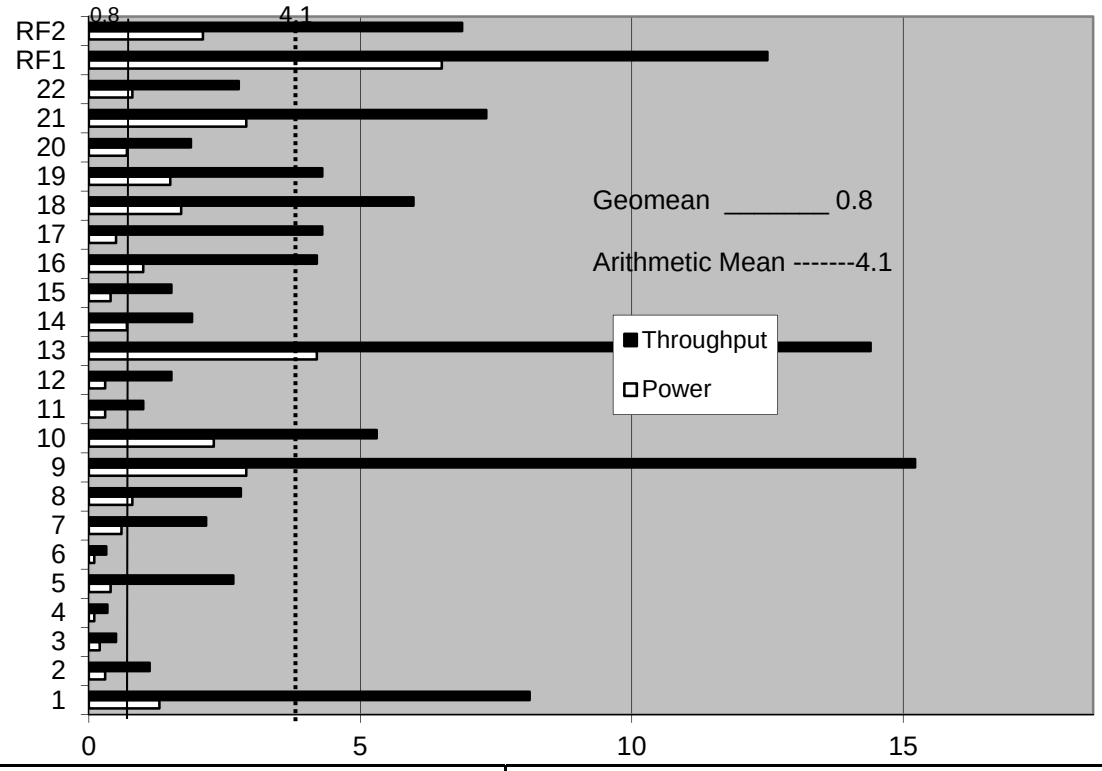
Benchmark results are highly dependent upon workload, specific application requirements, and system design and implementation. Relative system performance will vary as a result of these and other factors. Therefore, TPC Benchmark H should not be used as a substitute for a specific customer application 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. No warranty of system performance or price/performance is expressed or implied in this report.

lenovo FOR THOSE WHO DO.	**Lenovo ThinkServer RD630**	TPC-H Rev. 2.15.0 TPC-Pricing Rev. 1.7.0 Report Date: May 10, 2013

Total System Cost	Composite Query per Hour Metric	Price / Performance
$45,469 USD	**420,092.4** QphH@100GB	**$0.11 USD** $ / QphH@100GB

Database Size	Database Manager	Operating System	Other Software	Availability Date
100GB	**VectorWise 3.0.0**	**RedHat Enterprise Linux 6.4**		May 10, 2013



Database Load Time = 1:18:35	**Storage Redundancy Level**	
Load Includes Backup: N	Base Tables and Auxiliary Data Structures	5
Total Data Storage / Database Size = 24.00	DBMS Temporary Space	5
Percentage Memory / Database Size = 64.0%	OS and DBMS Software	0

System Configuration:	**Lenovo ThinkServer RD630**
Memory:	64 GB
Disk Controller	ThinkServer RAID 700 controller
Storage:	8 x 300GB 10K RPM HDD
Total Storage:	2400 GB
LAN Controller	Onboard dual port 1G NIC

lenovo FOR THOSE WHO DO.	Lenovo Think Server RD630	TPC-H Rev. 2.15.0 TPC Pricing Rev. 1.7.0 Report Date: May 10, 2013

Description	Part Number	Pricing Source	Unit Price	Qty	Extended Price	3 yr. Maint. Price
Server Hardware						
ThinkServer RD630 - 2595CTO	2595CTO	1	$13,550.00	1	$13,550.00	
2.5" Chassis with up to 8 Hard Drives		1	$0.00	1	$0.00	
E5-2690 2.90GHz,20M Cache,135W		1	$0.00	2	$0.00	
8GB 1RX4 RDIMM DDR3 1600		1	$0.00	8	$0.00	
ThinkServer 2.5" 300GB 10K SAS 6GPBS HS HDD		1	$0.00	8	$0.00	
ThinkServer RAID 700 Adapter with Battery		1	$0.00	1	$0.00	
Slim DVD RW SATA		1	$0.00	1	$0.00	
Basic DIT		1	$0.00	1	$0.00	
800W Redundant PSU		1	$0.00	1	$0.00	
800W Redundant PSU (Second module)		1	$0.00	1	$0.00	
On-site Warranty 3 Year Parts / 3 Year Labor		1	$0.00	1	$0.00	
On-site Warranty 3 Year 24x7x4Hr response upgrade	0C08358	1	$789.00	1		$789.00
				Subtotal	**$13,550.00**	**$789.00**
No external storage						
Software						
VectorWise 3.0 3-year license Lenovo server**	VW-3Y-GB-LENOVO	2	$250.00	100	$25,000.00	
VectorWise 1-year maintenance bug fixes**	VW-3Y-GB-LENOVO-MNT	2	$25.00	300		$7,500.00
VectorWise discount	15%	2			-$3,750.00	-$1,125.00
Red Hat Enterprise Linux 3 year Enterprise subscription	2246201	3	$3,504.99	1	$3,504.99	
				Subtotal	**$24,754.99**	**$6,375.00**
Audited by: Lorna Livingtree for Performance Metrics, Inc.				**Total**	**$38,304.99**	**$7,164.00**

Price Key: 1-Lenovo: Pricing may be verified by calling 1-855-253-6686			
2-Actian Pricing may be verified by calling 1-650-587-5500	**Cost of Ownership:**	**$45,469**	**USD**
contact: Mark Van de Wiel, mark.vandewiel@Actian.com			
3 - CDW: Red Hat pricing may be verified at CDW.com	**QphH@100GB:**	**420,092.4**	
All discounts are based on list prices and for similar quantities and configurations.			
Audited by: Lorna Livingtree for Performance Metrics, Inc.	**$/QphH@100GB:**	**$0.11**	**USD**

Prices used in TPC benchmarks reflect the 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 from their respective vendors. 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 items, please inform the TPC at pricing@tpc.org. Thank you.

	Lenovo ThinkServer RD630	TPC-H Rev.2.15.0 TPC-Pricing 1.7 Report Date May 10, 2013

Measurement Results

Database Scaling (SF/Size)	100
Total Data Storage/Database Size	8.00
Percentage Memory/Database Size	64.00%
Start of Database Load Time	4/17/13 7:55:27
End of Database Load Time	4/17/13 9:14:02
Database Load Time	1:18:35
Query Streams for Throughput Test (S)	5
TPC-H Power @100GB	458,663.5
TPC-H Throughput @100GB	384,764.9
TPC-H Composite @100GB	420,092.4
Total System Price Over 3 Years	$45,469
TPC-H Price/Performance Metrics ($/QphH@100GB)	$0.11

Measurement Interval in Throughput Test (Ts) 103

Power Run	Seed	Query Start Time Query End Time	Duration(sec)	RF1 Start Time RF1 End Time	RF 2 Start Time RF 2 End Time
	417091402	4/17/2013 09:29:46 4/17/2013 09:30:10	23.9	4/2/2013 00:33:15 4/2/2013 00:33:35	4/2/2013 00:34:43 4/2/2013 00:34:49
Throughput Streams	**Seed**	**Query Start Time Query End Time**	**Duration(sec)**	**RF1 Start Time RF1 End Time**	**RF 2 Start Time RF 2 End Time**
1	417091403	4/17/2013 09:30:35 4/17/2013 09:32:06	91.0	4/17/2013 09:31:24 4/17/2013 09:31:36	4/17/2013 09:31:36 4/17/2013 09:31:46
2	417091404	4/17/2013 09:30:35 4/17/2013 09:32:11	96.0	4/17/2013 09:31:46 4/17/2013 09:31:59	4/17/2013 09:31:59 4/17/2013 09:32:06
3	417091405	4/17/2013 09:30:35 4/17/2013 09:32:00	84.6	4/17/2013 09:32:06 4/17/2013 09:32:14	4/17/2013 09:32:14 4/17/2013 09:32:18
4	417091406	4/17/2013 09:30:35 4/17/2013 09:32:14	99.1	4/17/2013 09:30:35 4/17/2013 09:30:52	4/17/2013 09:30:52 4/17/2013 09:30:59
5	417091407	4/17/2013 09:30:35 4/17/2013 09:32:11	96.5	4/17/2013 09:30:59 4/17/2013 09:31:12	4/17/2013 09:31:12 4/17/2013 09:31:23

Stream ID	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9	Q10	Q11	Q12
0	3.8	0.7	0.6	0.2	1.4	0.2	1.4	1.9	11.7	5.2	0.9	0.8
1	20.6	5.9	1.6	0.9	5.4	1.5	9.3	6.7	64.5	15.1	6.3	3.0
2	20.6	2.2	1.1	0.7	3.9	1.1	18.6	12.0	93.8	15.7	2.7	2.7
3	15.7	6.2	0.9	1.3	8.4	1.3	13.9	18.8	182.6	15.8	4.2	2.6
4	17.3	2.9	1.0	1.2	5.8	1.4	5.3	12.3	95.7	13.4	6.4	5.6
5	30.3	2.4	1.8	1.2	7.7	1.5	8.7	12.9	84.2	15.7	2.8	6.8
6	20.0	2.9	2.0	0.9	8.5	0.7	6.7	8.6	49.3	22.9	2.1	3.9
Minimu	3.8	0.7	0.6	0.2	1.4	0.2	1.4	1.9	11.7	5.2	0.9	0.8
Maximu	30.3	6.2	2.0	1.3	8.5	1.5	18.6	18.8	182.6	22.9	6.4	6.8
Average	18.3	3.3	1.3	0.9	5.9	1.1	9.1	10.5	83.1	14.8	3.6	3.6

Stream ID	Q13	Q14	Q15	Q16	Q17	Q18	Q19	Q20	Q21	Q22	RF1	RF2
0	11.4	1.7	0.8	2.6	1.4	5.6	4.0	1.7	7.9	2.2	19.9	6.4
1	70.9	6.8	8.4	20.1	10.0	24.9	20.7	6.6	26.9	5.9	39.0	21.0
2	31.9	9.5	3.3	12.5	26.6	24.9	18.1	5.4	21.3	17.0	33.6	16.8
3	21.7	7.7	2.1	4.3	9.9	24.8	9.7	6.0	17.9	14.4	30.9	20.9
4	65.6	11.4	3.6	14.5	9.9	17.9	11.5	3.7	36.3	6.5	34.2	25.4
5	71.0	11.2	1.8	20.9	12.3	23.7	17.6	6.3	26.9	7.6	39.1	18.2
6	54.8	5.4	3.4	6.4	11.6	17.4	12.4	5.9	38.8	8.0	37.3	16.3
Minimu	11.4	1.7	0.8	2.6	1.4	5.6	4.0	1.7	7.9	2.2	19.9	6.4
Maximu	71.0	11.4	8.4	20.9	26.6	24.9	20.7	6.6	38.8	17.0	39.1	25.4
Average	46.8	7.7	3.3	11.6	11.7	19.9	13.4	5.1	25.1	8.8	33.4	17.9

Table of Contents

Abstract

This report documents the methodology and results of the TPC Benchmark© H test conducted on the  ThinkServer RD630 using RedHat Enterprise Linux 6.4 and Actian VectorWise 3.0.0

Company Name	System Name	Database Software	Operating System
Lenovo Corp.	ThinkServer RD630	VectorWise 3.0.0	RedHat Enterprise Linux 6.4
TPC Benchmark© H Metrics			
Total System Cost	TPC-H Throughput	Price/Performance	Availability Date
$45,469 USD	420,092.4 QphH@100GB	$0.11 USD $/QphH@100GB	May 10, 2013

Preface

The Transaction Processing Performance Council (TPC) is a non-profit corporation founded to define transaction processing and database benchmarks and to disseminate objective, verifiable TPC performance data to the industry. The TPC Benchmark© H (TPC-H) is a decision support benchmark.

TPC Benchmark© H Overview

The TPC Benchmark© H (TPC-H) consists of a suite of business oriented ad-hoc queries and concurrent data modifications. The queries and the data populating the database have been chosen to have broad industry-wide relevance while maintaining a sufficient degree of ease of implementation. This benchmark illustrates decision support systems that

- Examine large volumes of data;
- Execute queries with a high degree of complexity;
- Give answers to critical business questions.

TPC-H evaluates the performance of various decision support systems by the execution of sets of queries against a standard database under controlled conditions. The TPC-H queries:

- Give answers to real-world business questions;
- Simulate generated ad-hoc queries (e.g., via a point and click GUI interface);
- Are far more complex than most OLTP transactions;
- Include a rich breadth of operators and selectivity constraints;
- Generate intensive activity on the part of the database server component of the system under test;
- Are executed against a database complying to specific population and scaling requirements;
- Are implemented with constraints derived from staying closely synchronized with an on-line production database.

The TPC-H operations are modeled as follows:

- The database is continuously available 24 hours a day, 7 days a week, for ad-hoc queries from multiple end users and data modifications against all tables, except possibly during infrequent (e.g., once a month) maintenance sessions;
- The TPC-H database tracks, possibly with some delay, the state of the OLTP database through on-going refresh functions which batch together a number of modifications impacting some part of the decision support database;
- Due to the world-wide nature of the business data stored in the TPC-H database, the queries and the refresh functions may be executed against the database at any time, especially in relation to each other. In addition, this mix of queries and refresh functions is subject to specific ACIDity requirements, since queries and refresh functions may execute concurrently;

- To achieve the optimal compromise between performance and operational requirements, the database administrator can set, once and for all, the locking levels and the concurrent scheduling rules for queries and refresh functions.

The performance metric reported by TPC-H is called the TPC-H Composite Query-per-Hour Performance Metric (QphH@Size), and reflects multiple aspects of the capability of the system to process queries. These aspects include the selected database size against which the queries are executed, the query processing power when queries are submitted by a single stream and the query throughput when queries are submitted by multiple concurrent users. The TPC-H Price/Performance metric is expressed as $/QphH@Size. To be compliant with the TPC-H standard, all references to TPC-H results for a given configuration must include all required reporting components. The TPC believes that comparisons of TPC-H results measured against different database sizes are misleading and discourages such comparisons.

The TPC-H database must be implemented using a commercially available database management system (DBMS) and the queries executed via an interface using dynamic SQL. The specification provides for variants of SQL, as implementers are not required to have implemented a specific SQL standard in full.

TPC-H uses terminology and metrics that are similar to other benchmarks, originated by the TPC and others. Such similarity in terminology does not in any way imply that TPC-H results are comparable to other benchmarks. The only benchmark results comparable to TPC-H are other TPC-H results compliant with the same revision.

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

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

Further information is available at www.tpc.org

General Items

0.1 Test Sponsor

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

This benchmark was sponsored by lenovo Corp. and developed and engineered in partnership with Actian Corporation.

0.2 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 Tuning Options*
- *Optimizer/Query execution options*
- *Query processing tool/language configuration parameters*
- *Recovery/commit options*
- *Consistency/locking options*
- *Operating system and configuration parameters*
- *Configuration parameters and options for any other software component incorporated into the pricing structure*
- *Compiler optimization options*

This requirement can be satisfied by providing a full list of all parameters and options, as long as all those which have been modified from their default values have been clearly identified and these parameters and options are only set once.

The Supporting File Archive contains the Operating System and DBMS parameters used in this benchmark.

0.3 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 their protocol type.*
- *Number of LAN (e.g. Ethernet) Connections, including routers, workstations, terminals, etc., that were physically used in the test or are incorporated into the pricing structure.*

- *Type and the run-time execution location of software components (e.g., DBMS, query processing tools/languages, middle-ware components, software drivers, etc.).*

Both the measured and priced configurations are the same and consist of a lenovo ThinkServer RD 630 with:

- Two Intel Xeon E5-2690 Series processors with 8 cores @ 2.90 GHz
- 64GB (8 DIMM Slots each with 8GB DDR3 at 1600MHz)
- Eight 2.5" 300GB SAS drives at 10K RPM
- ThinkServer RAID 700 controller
- Onboard dual port 1G NIC
- Redundant power supplies

Clause 1: Logical Database Design

1.1 Database Definition Statements

Listings must be provided for all table definition statements and all other statements used to set up the test and qualification databases

The Supporting File Archive contains the table definitions and all other statements used to set up the test and qualification databases.

1.2 Physical Organization

The physical organization of tables and indices, within the test and qualification databases, must be disclosed. If the column ordering of any table is different from that specified in Clause 1.4, it must be noted.

No record clustering or index clustering was used. No column reordering was used.

1.3 Horizontal Partitioning

Horizontal partitioning of tables and rows in the test and qualification databases (see Clause 1.5.4) must be disclosed.

No horizontal partitioning was used.

1.4 Replication

Any replication of physical objects must be disclosed and must conform to the requirements of Clause 1.5.6.

No replication was used.

Clause 2: Queries and Refresh Functions

2.1 Query Language

The query language used to implement the queries must be identified.

SQL was the query language used to implement the queries.

2.2 Verifying Method of Random Number Generation

The method of verification for the random number generation must be described unless the supplied DBGEN and QGEN were used.

TPC-supplied DBGEN version 2.15.0 and QGEN version 2.15.0 were used.

2.3 Generating Values for Substitution Parameters

The method used to generate values for substitution parameters must be disclosed. If QGEN is not used for this purpose, then the source code of any non-commercial tool used must be disclosed. If QGEN is used, the version number, release number, modification number and patch level of QGEN must be disclosed.

TPC supplied QGEN version 2.15.0 was used to generate the substitution parameters.

2.4 Query Text and Output Data from Qualification Database

The executable query text used for query validation must be disclosed along with the corresponding output data generated during the execution of the query text against the qualification database. If minor modifications (see Clause 2.2.3) have been applied to any functional query definitions or approved variants in order to obtain executable query text, these modifications must be disclosed and justified. The justification for a particular minor query modification can apply collectively to all queries for which it has been used. The output data for the power and throughput tests must be made available electronically upon request.

Supporting Files Archive contains the actual query text and query output. Following are the modifications to the query.

- In Q1, Q4, Q5, Q6, Q10, Q12, Q14, Q15 and Q20, the “dateadd” function is used to perform date arithmetic.
- In Q7, Q8 and Q9, the “datepart” function is used to extract part of a date (e.g., datepart(yy,…)).
- In Q2, Q3, Q10, Q18 and Q21, the “top” function is used to restrict the number of output rows.
- The word GO is used as a command delimiter.

2.5 Query Substitution Parameters and Seeds Used

All the query substitution parameters used during the performance test must be disclosed in tabular format, along with the seeds used to generate these parameters.

Supporting Files Archive contains the query substitution parameters and seed used.

2.6 Isolation Level

The isolation level used to run the queries must be disclosed. If the isolation level does not map closely to one of the isolation levels defined in Clause 3.4, additional descriptive detail must be provided.

The queries and transactions were run with “Snapshot Isolation”.

2.7 Source Code of Refresh Functions

The details of how the refresh functions were implemented must be disclosed (including source code of any non-commercial program used).

Supporting Files Archive contains the Source Code of refresh functions.

Clause 3: Database System Properties

3.1 ACID Properties

The ACID (Atomicity, Consistency, Isolation, and Durability) properties of transaction processing systems must be supported by the system under test during the timed portion of this benchmark. Since TPC-H is not a transaction processing benchmark, the ACID properties must be evaluated outside the timed portion of the test.

All ACID tests were conducted according to specification. The Supporting Files Archive contains the source code of the ACID test scripts.

3.2 Atomicity Requirements

The results of the ACID tests must be disclosed along with a description of how the ACID requirements were met. This includes disclosing the code written to implement the ACID Transaction and Query.

3.2.1 Atomicity of the Completed Transactions

Perform the ACID Transaction for a randomly selected set of input data and verify that the appropriate rows have been changed in the ORDER, LINEITEM, and HISTORY tables.

The following steps were performed to verify the Atomicity of completed transactions:

1. The total price from the ORDER table and the extended price from the LINEITEM table were retrieved for a randomly selected order key.
2. The ACID Transaction was performed using the order key from step 1.
3. The ACID Transaction was committed.
4. The total price from the ORDER table and the extended price from the LINEITEM table were retrieved for the same order key. It was verified that the appropriate rows had been changed.

3.2.2 Atomicity of Aborted Transactions

Perform the ACID transaction for a randomly selected set of input data, submitting a ROLLBACK of the transaction for the COMMIT of the transaction. Verify that the appropriate rows have not been changed in the ORDER, LINEITEM, and HISTORY tables.

The following steps were performed to verify the Atomicity of the aborted ACID transaction:

1. The total price from the ORDER table and the extended price from the LINEITEM table were retrieved for a randomly selected order key.
2. The ACID Transaction was performed using the order key from step 1. The transaction was stopped prior to the commit.
3. The ACID Transaction was ROLLED BACK. .
4. The total price from the ORDER table and the extended price from the LINEITEM table were retrieved for the same order key used in steps 1 and 2. It was verified that the appropriate rows had not been changed.

3.3 Consistency Requirements

Consistency is the property of the application that requires any execution of transactions to take the database from one consistent state to another.

A consistent state for the TPC-H database is defined to exist when:

*O_TOTALPRICE = SUM(L_EXTENDEDPRICE – L_DISCOUNT) * (1 + L_TAX)*
For each ORDER and LINEITEM defined by (O_ORDERKEY = L_ORDERKEY)

3.3.1 Consistency Test

Verify that ORDER and LINEITEM tables are initially consistent as defined in Clause 3.3.2.1, based upon a random sample of at least 10 distinct values of O_ORDERKEY.

The following steps were performed to verify consistency:

1. The consistency of the ORDER and LINEITEM tables was verified based on a sample of O_ORDERKEYs.
2. One hundred ACID Transactions were submitted from each of nine execution streams.
3. The consistency of the ORDER and LINEITEM tables was re-verified.

3.4 Isolation Requirements

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.

3.4.1 Isolation Test 1 - Read-Write Conflict with Commit

Demonstrate isolation for the read-write conflict of a read-write transaction and a read-only transaction when the read-write transaction is committed.

The following steps were performed to satisfy the test of isolation for a read-only and a read-write committed transaction:

1. An ACID query was run with randomly selected values for O_KEY, L_KEY and DELTA to get the initial value for O_TOTALPRICE.
2. An ACID transaction was started using O_KEY, L_KEY and DELTA from step 1. The ACID transaction was suspended prior to commit.
3. An ACID query was started for the same O_KEY used in step 1. The ACID query ran to completion and did not see any uncommitted changes made by the ACID Transaction.
4. The ACID transaction was allowed to resumed and committed normally.
5. The ACID query was run again and completed. It returned the data as committed by the ACID transaction.

3.4.2 Isolation Test 2 - Read-Write Conflict with Rollback

Demonstrate isolation for the read-write conflict of a read-write transaction and a read-only transaction when the read-write transaction is rolled back.

The following steps were performed to satisfy the test of isolation for read-only and a rolled back read-write transaction:

1. An ACID query was run with randomly selected values for O_KEY, L_KEY and DELTA to get the initial value for O_TOTALPRICE.

2. An ACID transaction was started using O_KEY, L_KEY and DELTA from step 1.
3. The ACID transaction was suspended prior to Rollback.
4. An ACID query was started for the same O_KEY used in step 1. The ACID query did not see any uncommitted changes made by the ACID transaction.
5. The ACID transaction was rolled back.
6. The ACID query was run again and completed. It showed no changes in the data from step 1.

3.4.3 Isolation Test 3 - Write-Write Conflict with Commit

Demonstrate isolation for the write-write conflict of two update transactions when the first transaction is committed.

Two tests were run, the first with a T2 transaction that commits and the second with a T2 transaction that rolls back. Results from the first test were as follows:

1. An ACID query was run with randomly selected values for O_KEY, L_KEY and DELTA to get the initial value for O_TOTALPRICE.
2. An ACID transaction, T1, was started using the same O_KEY, L_KEY and DELTA as step 1.
3. T1 was suspended prior to commit.
4. Another ACID transaction, T2, was started using the same O_KEY and L_KEY as step 1, and a randomly selected DELTA.
5. T2 did not see T1's uncommitted data. T2 completed the commit before T1's sleep ended.
6. T1 was allowed to resume and received and error. This was expected due to the "Snapshot Isolation" in use by the DBMS. This is also known as "First Committer Wins".
7. T1 was retried and committed successfully.
8. The ACID query was run again to verify that the O_TOTALPRICE was the correct value.
9. The HISTORY table was examined and demonstrated the commits of both T1 and T2 in the correct order by timestamp.

Results from the second test were as follows:

1. An ACID query was run for a randomly selected O_KEY, L_KEY and DELTA to get the initial value for O_TOTALPRICE.
2. An ACID transaction, T1, was started with the values used in step 1.
3. T1 was suspended prior to commit.
4. A second ACID transaction, T2, was started with the same O_KEY and L_KEY as step 1 and a different value for DELTA.
5. T2 did not see any of T1's updates. T2 rolled back.
6. T1 resumed and completed normally.
7. The ACID query was run again to verify the database was updated with the values from T1 and not T2.
8. The HISTORY table was examined and demonstrated that only T1 committed without seeing any T2 data.

3.4.4 Isolation Test 4 - Write-Write Conflict with Rollback

Demonstrate isolation for the write-write conflict of two update transactions when the first transaction is rolled back.

Two tests were run, the first with a T2 transaction that commits and the second with a T2 transaction that rolls back.

The results from the first test were as follows:

1. An ACID query was run for a randomly selected O_KEY, L_KEY and DELTA to get the initial value for O_TOTALPRICE
2. An ACID transaction, T1, was started using the values from step 1.
3. T1 was suspended prior to rollback.
4. A second ACID transaction, T2, was started with the same O_KEY and L_KEY as step 1 and a different value for DELTA.
5. T2 completed normally and committed.
6. T1 was allowed to rollback.
7. The ACID query was run again to verify the database was updated with the values from T2 and not from T1.
8. The HISTORY table was examined and demonstrated that only T2 committed.

The results from the second test were as follows

1. An ACID query was run for a randomly selected O_KEY, L_KEY and DELTA to get the initial value for O_TOTALPRICE.
2. An ACID transaction, T1, was started with the same values as from step 1.
3. T1 suspended prior to rollback.
4. A second ACID transaction, T2, was started with the same O_KEY and L_KEY as step 1 and a different value for DELTA.
5. T2 rolled back its updates and completed normally.
6. T1 resumed, rolled back its updates and completed normally.
7. An ACID query was run to verify that O_TOTALPRICE was the original value before T1 or T2.
8. The HISTORY table was examined and demonstrated that neither transaction committed.

3.4.5 Isolation Test 5 – Concurrent Read and Write Transactions on Different Tables

Demonstrate the ability of read and write transactions affecting different database tables to make progress concurrently.

The following steps were performed to verify isolation of concurrent read and write transactions on different tables:

1. An ACID query was run for a randomly selected O_KEY, L_KEY and DELTA to get the initial value for O_TOTALPRICE.
2. An ACID transaction, T1, was started with the values from step 1.
3. T1 was suspended prior to commit.
4. A query was started using random values for PS_PARTKEY and PS_SUPPKEY, all columns of the PARTSUPP table for which PS_PARTKEY and PS_SUPPKEY are equal were returned. The query completed normally.
5. T1 was allowed to resume and committed normally.
6. It was verified that O_TOTALPRICE had been changed by T1.

3.4.6 Isolation Test 6 – Update Transactions during Continuous Read-Only Query Stream

Demonstrate the continuous submission of arbitrary (read-only) queries against one or more tables of the database does not indefinitely delay update transactions affecting those tables from making progress.

The following steps were performed to verify isolation of update transaction during continuous read-only query:

1. A stream was submitted that executed Q1 ten times in a row with a delta of 0 to ensure that each query ran as long as possible.
2. An ACID transaction, T1, was started for a randomly selected O_KEY, L_KEY and DELTA.
3. T1 completed and it was verified that O_TOTALPRICE was updated correctly.
4. The stream submitting Q1 finished.

3.5 Durability Requirements

The tested system must guarantee durability: the ability to preserve the effects of committed transactions and insure database consistency after recovery from any one of the failures listed in Clause 3.5.2.

3.5.1 Permanent Unrecoverable Failure of Any Durable Medium

Guarantee the database and committed updates are preserved across a permanent irrecoverable failure of any single durable medium containing TPC-H database tables or recovery log tables.

The database files including the logs were distributed across 7 disk drives in a RAID 5 configuration. The tests were conducted on the qualification database. These steps were performed to demonstrate that committed updates are preserved across a permanent irrecoverable failure of a disk drive:

1. The consistency of the ORDERS and LINEITEM tables was verified using 120 randomly chosen values for O_ORDERKEY.
2. Exactly 9 streams of ACID transactions were started.
3. When the driver script indicated that at least 100 transactions had completed in each stream, a randomly selected disk drive was removed from the SUT and the SUT continued to process work until all streams had completed 200 transactions.
4. An analysis of the transaction start and end times from each stream showed that there was at least 1 transaction in-flight at all times.
5. An analysis of the HISTORY table showed that all of the values used for O_ORDERKEY in step 1 were used by some transaction in step 2.
6. An analysis of the success file and the HISTORY table showed that all entries in the HISTORY table had a corresponding entry in the success file and that every entry in the success file had a corresponding entry in the HISTORY table.
7. The consistency of the database was re-verified.

3.5.2 System Crash

Guarantee the database and committed updates are preserved across an instantaneous interruption (system crash/system hang) in processing which requires the system to reboot to recover.

1. The system crash and memory failure tests were combined. First the consistency of the ORDER and LINEITEM tables was verified.
2. Then transactions were submitted from exactly 9 streams.
3. Once the driver script indicated that 100 transactions had been submitted from each stream, power to the SUT was removed by removing both power cords.
4. When power was restored to the SUT, the system rebooted and the database was restarted.
5. The HISTORY table and success files were compared to verify that every record in the HISTORY table had a corresponding record in the success file and that each record in the success file had a corresponding entry in the HISTORY table.

6. The consistency of the ORDERS and LINEITEM tables was then verified again.

3.5.3 Memory Failure

Guarantee the database and committed updates are preserved across failure of all or part of memory (loss of contents).

See section 3.5.2

Clause 4: Scaling and Database Population

4.1 Initial Cardinality of Tables

The cardinality (e.g., the number of rows) of each table of the test database, as it existed at the completion of the database load (see clause 4.2.5) must be disclosed.

Table 4.1 lists the TPC Benchmark H defined tables and the row count for each table as they existed upon completion of the build.

Table 4.1: Initial Number of Rows

Table Name	Row Count
Region	5
Nation	25
Supplier	1,000,000
Customer	15,000,000
Part	20,000,000
Partsupp	80,000,000
Orders	150,000,000
Lineitem	600,037,902

4.2 Distribution of Tables and Logs Across Media

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

The storage system consisted of:

- Eight 2.5" SAS drives at 10K RPM
- ThinkServer RAID 700 - LSI MegaRAID 9260-8i 512GB NVRAM 8-port RAID controller

The database tables and the temporary files were distributed on a RAID 5 volume created across seven 300 GB 10K RPM SAS disk drives.

4.3 Mapping of Database Partitions/Replications

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

Database partitioning/replication were not used.

4.4 Implementation of RAID

Implementations may use some form of RAID to ensure high availability. If used for data, auxiliary storage (e.g. indexes) or temporary space, the level of RAID used must be disclosed for each device.

The database tables and the temporary files were distributed on a RAID 5 volume created across seven 300 GB 10K RPM SAS disk drives.

4.5 DBGEN Modifications

The version number, release number, modification number, and patch level of DBGEN must be disclosed. Any modifications to the DBGEN (see Clause 4.2.1) source code must be disclosed. In the event that a program other than DBGEN was used to populate the database, it must be disclosed in its entirety.

DBGEN version 2.15.0 was used, no modifications were made.

4.6 Database Load time

The database load time for the test database (see clause 4.3) must be disclosed.

The database load time was 1 hour 18 minutes and 35 seconds.

4.7 Data Storage Ratio

The data storage ratio must be disclosed. It is computed by dividing the total data storage of the priced configuration (expressed in GB) by the size chosen for the test database as defined in 4.1.3.1. The ratio must be reported to the nearest 1/100th, rounded up.

The database storage ratio can be found in Table 4.7

Table 4.7: Data Storage Ratio

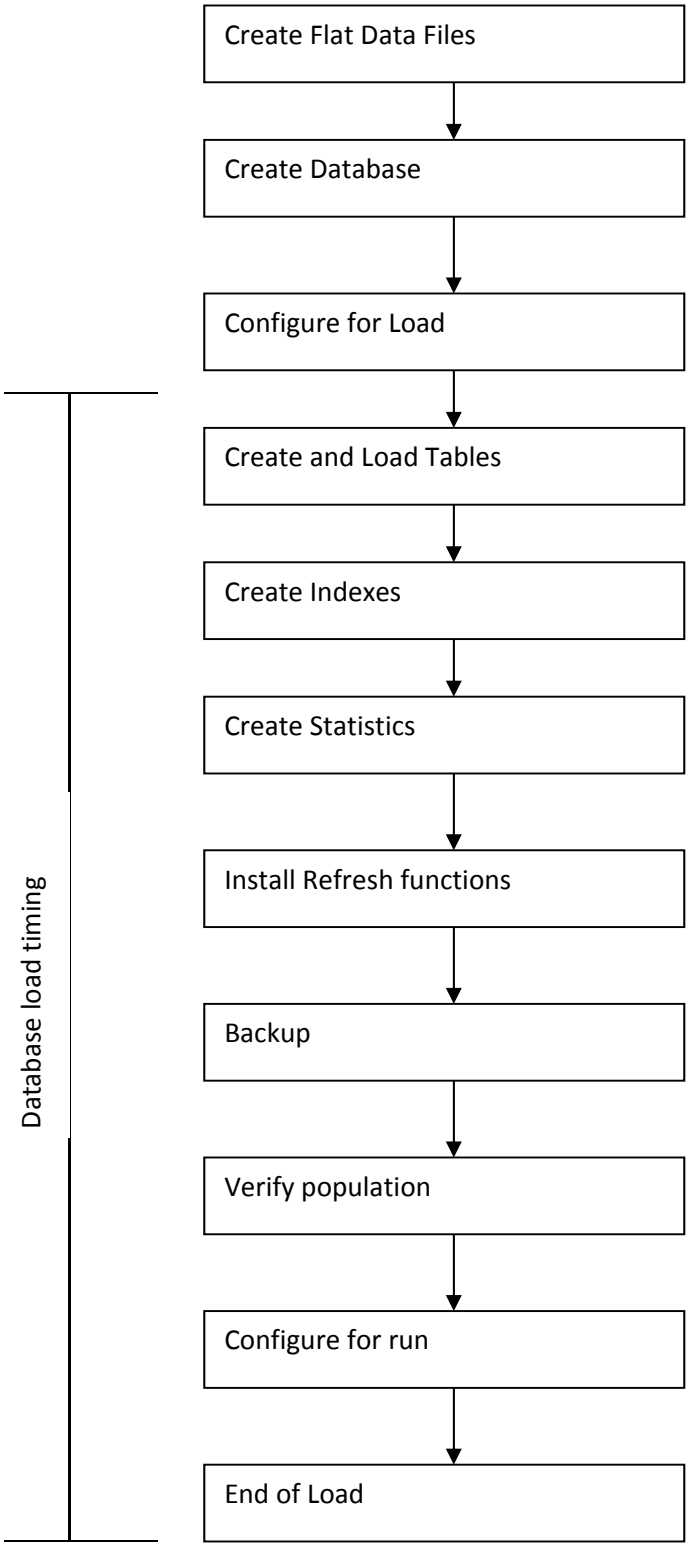
Storage Devices	Total Storage Capacity	Scale factor	Data Storage Ratio
8 x 300 GB disk drives	2,400 GB	100	24.00

4.8 Database Load Mechanism Details and Illustration

The details of the database load must be disclosed, including a block diagram illustrating the overall process. Disclosure of the load procedure includes all steps, scripts, input and configuration files required to completely reproduce the test and qualification databases.

Flat files were created using DBGEN. The tables were loaded as shown in Figure 4.8.

Figure 4.8: Block Diagram of Database Load Process



4.9 Qualification Database Configuration

Any differences between the configuration of the qualification database and the test database must be disclosed.

The qualification database used identical scripts to create and load the data with changes to adjust for the database scale factor.

4.10 Memory to Database Size Percentage

The memory to database size percentage must be disclosed.

The installed memory was 64GB. The database size was scale factor 100. The database size percentage is 64.0%.

Clause 5: Performance Metrics and Execution Rules

5.1 System Activity Between Load and Performance Tests

Any system activity on the SUT that takes place between the conclusion of the load test and the beginning of the performance test must be fully disclosed.

Auditor requested script was run to display the indices that had been created on the database and the row counts at the end of the load. All scripts and queries used are included in the Supporting Files Archive.

5.2 Steps in the Power Test

The details of the steps followed to implement the power test (e.g., system boot, database restart, etc.) must be disclosed.

The following steps were used to implement the power test:

1. RF1 Refresh Transaction
2. Stream0 Execution
3. RF2 Refresh Transaction.

5.3 Timing Intervals for Each Query and Refresh Function

The timing intervals (see Clause 5.3.6) for each query of the measured set and for both refresh functions must be reported for the power test.

The timing intervals for each query and both refresh functions are given in the Numerical Quantities Summary earlier in the Executive Summary.

5.4 Number of Streams for The Throughput Test

The number of execution streams used for the throughput test must be disclosed.

Five streams were used for the Throughput Test.

5.5 Start and End Date/Times for Each Query Stream

The start time and finish time for each query execution stream must be reported for the throughput test.

The Numerical Quantities Summary contains the start and stop times for the query execution streams run on the system reported.

5.6 Total Elapsed Time for the Measurement Interval

The total elapsed time of the measurement interval (see Clause 5.3.5) must be reported for the throughput test.

The Numerical Quantities Summary contains the timing intervals for the throughput test run on the system reported.

5.7 Refresh Function Start Date/Time and Finish Date/Time

Start and finish time for each update function in the update stream must be reported for the throughput test.

Start and finish time for each update function in the update stream are included in the Numerical Quantities Summary earlier in the Executive Summary.

5.8 Timing Intervals for Each Query and Each Refresh Function for Each Stream

The timing intervals (see Clause 5.3.6) for each query of each stream and for each update function must be reported for the throughput test.

The timing intervals for each query and each update function are included in the Numerical Quantities Summary earlier in the Executive Summary.

5.9 Performance Metrics

The computed performance metrics, related numerical quantities and the price performance metric must be reported.

The Numerical Quantities Summary contains the performance metrics, related numerical quantities, and the price/performance metric for the system reported.

5.10 The Performance Metric and Numerical Quantities from Both Runs

A description of the method used to determine the reproducibility of the measurement results must be reported. This must include the performance metrics (QppH and QthH) from the reproducibility runs.

Performance results from the first two executions of the TPC-H benchmark indicated the following difference for the metric points:

Table 5.10: Performance Metric

Run	QppH @ 100GB	QthH @ 100GB	QphH @ 100GB
Run 1	458,663.5	407,491.3	432,320.9
Run 2	458,663.5	384,764.9	420,092.4

5.11 System Activity Between Tests

Any activity on the SUT that takes place between the conclusion of Run1 and the beginning of Run2 must be disclosed.

There was no activity between Run1 and Run2.

5.12 Dataset Verification

Verify that the rows in the loaded database after the performance test are correct by comparing some small number of rows extracted at random from any two files of the corresponding Base, Insert and Delete reference data set files for each table and the corresponding rows of the database

Verified according to specification.

5.13 Referential Integrity

Verify referential integrity in the database after the initial load.

An auditor supplied script was used to verify referential integrity.

Clause 6: SUT and Driver Implementation

6.1 Driver

A detailed description of how the driver performs its functions must be supplied, including any related source code or scripts. This description should allow an independent reconstruction of the driver.

The supporting files archive contains the scripts that were used to implement the driver.
The power test is invoked through the script power_test.sh. It starts the stream 0 SQL script along with the refresh functions such that:

- The SQL for RF1 is submitted and executed by the database
- Then the queries as generated by QGEN are submitted in the order defined by Clause 5.3.5.4
- The SQL for RF2 is then submitted from the same connection used for RF1 and executed by database

The Throughput test is invoked through the script throughput_test.sh. This script then initiates all of the SQL streams and the refresh stream.

6.2 Implementation Specific Layer

If an implementation-specific layer is used, then a detailed description of how it performs its functions must be supplied, including any related source code or scripts. This description should allow an independent reconstruction of the implementation-specific layer.

There was no Implementation Specific Layer, only native scripts and SQL.

6.3 Profile-Directed Optimization

If profile-directed optimization as described in Clause 5.2.9 is used, such used must be disclosed.

Profile-directed optimization was not used.

Clause 7: Pricing

7.1 Hardware and Software 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 date. 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.

A detailed list of all hardware and software, including the 3-year support, is provided in the Executive Summary in the Abstract section of this report. The price quotations are included in Appendix A.

7.2 Three Year Price

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.

The pricing details for this disclosure are contained in the executive summary pages.

7.3 Availability Dates

The committed delivery date for general availability of products used in the price calculations must be reported. When the priced system includes products with different availability dates, the availability date reported on the executive summary must be the date by which all components are committed to being available. The full disclosure report must report availability dates individually for at least each of the categories for which a pricing subtotal must be provided.

All components of the SUT will be available on the date of publication.

Supporting Files Index

An index for all files included in the supporting files archive as required by Clause 8.3.2 through 8.3.8 must be provided in the report.

Table 8.0: Supporting File Index

Clause	Description	Archive File	Pathname
Clause 1	Device setup	benchmark_scripts.zip	sysinfo/df_h.out sysinfo/etc.fstab sysinfo/mount.out
	Installation and configuration	benchmark_scripts.zip	Sysinfo/proc-cpuinfo sysinfo/proc-meminfo sysinfo/vectorwise.conf
	OS Tunable Parameters	benchmark_scripts.zip	sysinfo/sysctl-A.out
	DB creation scripts	benchmark_scripts.zip	ddl/create_index.sql ddl/create_tables.sql ddl/grant.sql create_db.sh
Clause 2	QGen Modifications	benchmark_scripts.zip	None
Clause 3	ACID Test scripts	benchmark_scripts.zip	acid/Makefile acid/acidkeys.sc acid/acidquery.sh acid/acidtrans.sc acid/acidtrans_rb.sc acid/create_history.sql acid/get_iisysdepvar.sh acid/pskeys.sc acid/run_aci.sh acid/atom/atom1.sh acid/atom/atom2.sh acid/cons/Makefile acid/cons/consist.sh acid/cons/consist_condition.sc acid/dur/Makefile acid/dur/acidtrans_dur.sc acid/dur/dur.sh acid/dur/post-failure.sh acid/iso/iso1.sh acid/iso/iso2.sh acid/iso/iso3.sh acid/iso/iso3b.sh acid/iso/iso4.sh acid/iso/iso4b.sh acid/iso/iso5.sh acid/iso/iso6.sh acid/iso/iso6_q1.sql acid/iso/iso6_txn.sh

	ACID Test Results	benchmark_scripts.zip	output/atom/atom1_output output/atom/atom2_output output/cons/consist_output output/cons/step1_stream*_output output/cons/step2_stream*_output output/cons/step3_stream*_output output/cons/stream*_rnd output/dur/diskloss/POST_FAILURE_HISTORY_TABLE output/dur/diskloss/SUCCESS_FILE output/dur/diskloss/step1_stream*_output output/dur/diskloss/step2_stream*_output output/dur/diskloss/step6_stream*_output output/dur/diskloss/stream*_rnd output/dur/powerloss/POST_FAILURE_HISTORY_TABLE output/dur/powerloss/SUCCESS_FILE output/dur/powerloss/step1_stream*_output output/dur/powerloss/step2_stream*_output output/dur/powerloss/step6_stream*_output output/dur/powerloss/stream*_rnd output/iso/iso1_output output/iso/iso2_output output/iso/iso3_output output/iso/iso3b_output output/iso/iso4_output output/iso/iso4b_output output/iso/iso5_output
Clause 4	Qualification db load results	benchmark_scripts.zip	outuput/1G-load-console.log
	Qualification db validation results	benchmark_scripts.zip	output/validation.output
	DBGEN Modifications	benchmark_scripts.zip	None.
	Database Load Scripts	benchmark_scripts.zip	load_test.sh
	Test db Load results	benchmark_scripts.zip	output/DB-load
Clause 5	Run 1	run1results.zip	console.log performance_test_report.txt power-test/* throughput-test/*
	Run 2	run1results.zip	console.log power-test/* throughput-test/*
Clause 6	implementation scripts	benchmark_scripts.zip	performance_test.sh power_test.sh tput_test.sh
Clause 7	n/a	n/a	n/a
Clause 8	Query substitution parameters	run1report.zip run2report.zip	power-test/queries/stream0/q*_param throughput-test/queries/stream*/q*_param
	RF function source code	benchmark_scripts.zip	RF/header RF/rf1.sql RF/rf2.sql

Auditors' Information and Attestation Letter

The auditor's agency name, address, phone number, and Attestation letter with a brief audit summary report indicating compliance must be included in the full disclosure report. A statement should be included specifying who to contact in order to obtain further information regarding the audit process.

The auditor's letter is included in the following section.

This benchmark was audited by:

Lorna Livingtree
Performance Metrics, Inc.
PO Box 984
Klamath, CA 95548
Phone: 707-954-7757

PERFORMANCE METRICS INC.
TPC Certified Auditors

April 26, 2013

Mr. Michael Mettler
Enterprise Solutions Delivery ODTL
Lenovo Corp.
1009 Think Place
Morrisville, NC 27560

I have verified by remote the TPC Benchmark™ H for the following configuration:

Platform: Lenovo ThinkServer RD630
Database Manager: VectorWise 3.0.0
Operating System: Red Hat Enterprise Linux 6.4

CPU's	Memory	Total Disks	Qpph@ 100GB	QthH @100GB	QphH@100GB
2 Intel Xeon E5-2690 @ 2.90 GHz	64 GB	8 @ 300 GB	**458,663.5**	**384,764.9**	**420,092.4**

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 database tables were defined with the proper columns, layout and sizes.
- The tested database was correctly scaled and populated for 100GB using DBGEN. The version of DBGEN was 2.15.0.
- The data generated by DBGEN was successfully compared to reference data.
- The qualification database layout was identical to the tested database except for the size of the files.
- The query text was verified to use only compliant variants and minor modifications.
- The executable query text was generated by QGEN and submitted through a standard interactive interface. The version of QGEN was 2.15.0.
- The validation of the query text against the qualification database produced compliant results.

PO Box 984 Klamath, CA 95548
(707) 954-7757 fax: (707) 482-0575
Page 1
email: LornaL@PerfMetrics.com

- The refresh functions were properly implemented and executed the correct number of inserts and deletes.
- The load timing was properly measured and reported.
- The execution times were correctly measured and reported.
- The performance metrics were correctly computed and reported.
- The repeatability of the measurement was verified.
- The ACID properties were successfully demonstrated and verified.
- The system pricing was checked for major components and maintenance.
- The executive summary pages of the FDR were verified for accuracy.

Auditor's Notes:

Atomicity, isolation and consistency were successfully demonstrated on this exact same system configured for a benchmark with scale factor 300 with 128GB of memory. It is my opinion that this is sufficient evidence without requiring a duplicate set of these three tests.

Sincerely,

Lorna Livingtree
Auditor

PO Box 984 Klamath, CA 95548
(707) 954-7757 fax: (707) 482-0575

email: LornaL@PerfMetrics.com

Appendix A

ACTIAN Order Confirmation

Actian Corporation
500 Arguello Street Suite 200
Redwood City, CA 94063
United States

Phone: (650) 587-5500
Fax: (650) 649-2358

Order Number: 05-130413
Account Number: 123457
Agreement ID:
Payment Terms: Net 30
Partner:

Prepared by:
Mark Van de Wiel
(650) 587-5538
mark.vandewiel@actian.com

Message:

PREPARED FOR (BILL TO)	SHIP TO
Michael Mettler **Technical Solutions Center** **ThinkServer Business Unit** **Office: 919.294.0410** **Email: mmettler@lenovo.com**	**Lenovo – ThinkServer Business Unit** **Attn. Michael Mettler** **1009 Think Place** **Morrisville, NC 27560** **USA**

Product item	Memo	Contract Start	Contract End	Unit	Qty	Price	Discount %	Extended Price
VW-3Y-GB-LENOVO	3 year license, 1 GB source data, Lenovo server	01-May-2013	30-April -2016	GB	100	USD 250	15	USD 21,250
VW-3Y-GB-LENOVO-MNT	1 year professional support, bug fixes only, 1 GB source data, Lenovo server	01-May-2013	30-April-2016	GB	300	USD 25	15	USD 6,375

Grand Total:	USD 27,625