



TPC Benchmark™ C Full Disclosure Report

Alibaba Cloud Elastic Compute Service Cluster (with
1554 OceanBase Data Nodes)

Using

*OceanBase v2.2 Enterprise Edition with Partitioning,
Horizontal Scalability and Advanced Compression*

First Edition
May 15, 2020

First Printing –May 15, 2020

Copyright © 2020 Ant Financial (Hang Zhou) Network Technology Co., Ltd. and/or its affiliates. All rights reserved.

As the Sponsor of this benchmark test, Ant Financial (Hang Zhou) Network Technology Co., Ltd. believes that information in this document is accurate as of the publication date. The information is subject to change without notice. The sponsor assumes no responsibility for any errors that may appear in this document.

The pricing information in this document accurately reflects the current prices as of the publication date. However, the Sponsor provides no warranty of the pricing information in this document.

The performance information in this document is for guidance only. System performance is highly dependent on many factors including hardware, software, and user application characteristics; relative system performance may vary significantly as a result of these and other factors. The Sponsor does not warrant or represent that a user can or will achieve similar performance. No warranty on system performance or price/performance is expressed or implied in this document.

OceanBase is registered trademark of Alibaba and/or its affiliates.

Intel and Intel Xeon are trademarks or registered trademarks of Intel Corporation.

Other names may be trademarks of their respective owners.

This product and related documentation are protected by copyright and distributed under licenses restricting its use, copying, and distribution. No part of this product may be reproduced in any form by any means without prior written authorization of Ant Financial (Hang Zhou) Network Technology Co., Ltd. and its licensors, if any.

THIS PUBLICATION IS PROVIDED WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NON-INFRINGEMENT.

THIS PUBLICATION COULD INCLUDE TECHNICAL INACCURACIES OR TYPOGRAPHICAL ERRORS. Ant Financial (Hang Zhou) Network Technology Co., Ltd. MAY MAKE IMPROVEMENTS AND/OR CHANGES IN THE PRODUCT (S) AND/OR THE PROGRAM (S) DESCRIBED IN THIS DOCUMENT.

Abstract

This report documents the methodology and results of the TPC Benchmark™ C test conducted on the following environment as measured by Ant Financial (Hang Zhou) Network Technology Co., Ltd. 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 Doug Johnson of InfoSizing to verify compliance with the relevant TPC specifications.

System	Processors	Database Environment	Operating System
Alibaba Cloud Elastic Compute Service Cluster (with 1554 OceanBase Data Nodes)	84 vCPU, 2.5 GHz Intel Xeon Platinum 8163(Skylake) (Per ecs.i2d.21xlarge instance)	OceanBase v2.2 Enterprise Edition with Partitioning, Horizontal Scalability and Advanced Compression	Aliyun Linux 2

TPC Benchmark C Metrics

Total System Cost	TPC-C Throughput	Price / Performance	Availability Date
Three year cost includes: <ul style="list-style-type: none">• Hardware• Software• Maintenance	Maximum Qualified Throughput expressed as transactions per minute – C (tpmC)	Total System Cost / tpmC	Date for which all components, hardware and software are available for purchase
CNY 2,814,509,552.00	707,351,007	CNY 3.98	June 08, 2020

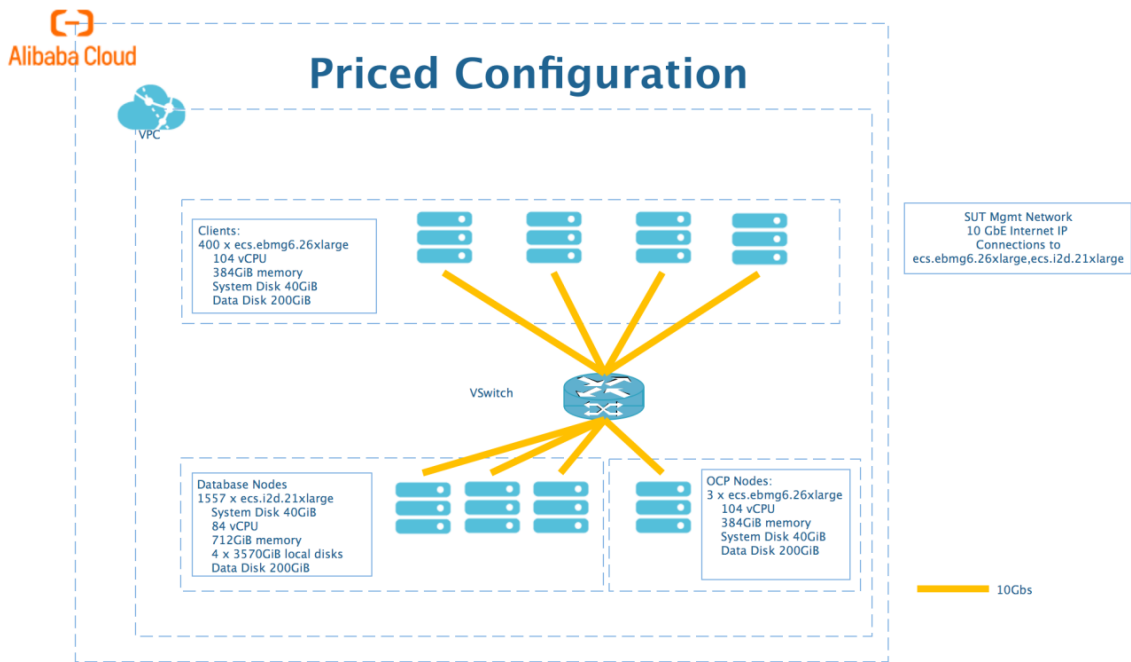


Alibaba Cloud Elastic Compute Service Cluster (with 1554 OceanBase Data Nodes)

**TPC-C 5.11.0
TPC-Pricing 2.5.0**

**Report Date
May 15, 2020**

Total System Cost	TPC-C Throughput	Price/Performance	Availability Date	
CNY 2,814,509,552.00	707,351,007 tpmC	CNY 3.98/tpmC	June 08, 2020	
Database Server Processors/Cores/Threads	Database Manager	Operating System	Other Software	Number of Users
Intel Xeon Platinum 8163(Skylake) 2.5GHz 130,788 vCPU	OceanBase v2.2 Enterprise Edition with Partitioning, Horizontal Scalability and Advanced Compression	Aliyun Linux 2	Nginx 1.15.8	559,440,000



System Component	1557 x Database Nodes (each with)		3 x OCP Nodes (each with)		400 x Clients (each with)		Total
Proc./Core/Thread (Virtual CPU)	2/42/84 (84 vCPU)	Intel Xeon Platinum 8163 (2.5GHz, 33MB)	2/52/104 (104 vCPU)	Intel Xeon Platinum 8269CY 2.5GHz (2.5GHz, 35.75B)	2/52/104 (104 vCPU)	Intel Xeon Platinum 8269CY 2.5GHz (2.5GHz, 35.75MB)	3920/86,350/172,700 (172,700 vCPU)
Memory		712GB		384GB		384GB	1,263,336 GB
OS Disk	1	40GB Ultra Cloud Disk	1	40GB Ultra Cloud Disk	1	40GB Ultra Cloud Disk	
Local Disk	4	3,570GB NVMe SSD					
Cloud Disk	1	200GB Ultra Cloud Disk	1	200GB Ultra Cloud Disk	1	200GB Ultra Cloud Disk	
Total Storage		22,607,640 GB		720GB		96,000GB	22,704,360GB



Alibaba Cloud Elastic Compute Service Cluster (with 1554 OceanBase Data Nodes)

TPC-C 5.11.0
TPC-Pricing 2.5.0

Report Date
May 15, 2020

Description	Part Number	Src	Unit Price (CNY)	Qty	Ext. Price (CNY)	3-Y Maintenance (CNY)
Licensed Compute Services						
<u>Virtual cloud server</u>						
ECS Instance ecs.i2d.21xlarge (3-year price) - NVMe SSD Local Disk (4 x 3570GB) - vCPU (84) - Memory (712 GB)	ecs.i2d.21xlarge (South China 2)	2	349,287.16	1,557	543,840,108.12	
ECS System Disk (Ultra Cloud Disk 40GB)	Included	2	0.00	1,557	0.00	
ECS Data Disk (Ultra Cloud Disk 200GB)	Included	2	0.00	1,557	0.00	
Aliyun Linux 2	Included	2	0.00	1,557	0.00	
Virtual cloud server						
ECS Instance ecs.ebmg6.26xlarge (3-year price) - vCPU (104) - Memory (384 GB)	ecs.ebmg6.26xlarge (South China 2)	2	250,956.00	400	100,382,400.00	
ECS System Disk (Ultra Cloud Disk 40GB)	Included	2	0.00	400	0.00	
ECS Data Disk (Ultra Cloud Disk 200GB)	Included	2	0.00	400	0.00	
Aliyun Linux 2	Included	2	0.00	400	0.00	
Virtual cloud server						
ECS Instance ecs.ebmg6.26xlarge (3-year price) - vCPU (104) - Memory (384 GB)	ecs.ebmg6.26xlarge (South China 2)	2	250,956.00	3	752,868.00	
ECS System Disk (Ultra Cloud Disk 40GB)	Included	2	0.00	3	0.00	
ECS Data Disk (Ultra Cloud Disk 200GB)	Included	2	0.00	3	0.00	
Aliyun Linux 2	Included	2	0.00	3	0.00	
Sub-Total					644,975,376.12	
Licensed Software Services						
OceanBase v2.2 Enterprise Edition, per vCPU for 3 years		1	14,850.00	130,788	1,942,201,800.00	
OceanBase v2.2 Partitioning, per vCPU for 3 years		1	2,475.00	130,788	323,700,300.00	
OceanBase v2.2 Horizontal Scalability, per vCPU for 3 years		1	2,475.00	130,788	323,700,300.00	
OceanBase v2.2 Advanced Compression, per vCPU for 3 years		1	2,475.00	130,788	323,700,300.00	3,178,148,400.00
Sub-Total					2,913,302,700.00	3,178,148,400.00
Other Components						
Lenovo ThinkPad X1 Yoga 2019 14" Laptop (includes 2 spares)	20SA000BCD	3	16,999.00	3	50,997.00	
Sub-Total					50,997.00	

Other Services				
Alibaba Cloud Enterprise Support Service for 3 years 7x24x4hrs - ECS support included - Aliyun Linux 2 support included	2	300,000.00	3	900,000.00
OpenResty Professional Edition Support Service for 3 years 7x24x4hrs - Nginx support included	4	15,300.00	400	6,120,000.00
			Sub-Total	0.00
				7,020,000.00
Discounts*				
OceanBase v2.2 Enterprise Edition Discount	1	-9,578.25	130,788	-1,252,720,161.00
OceanBase v2.2 Partitioning Discount	1	-1,596.38	130,788	-208,787,347.44
OceanBase v2.2 Horizontal Scalability Discount	1	-1,596.38	130,788	-208,787,347.44
OceanBase v2.2 Advanced Compression Discount	1	-1,596.38	130,788	-208,787,347.44
OceanBase v2.2 Support Service Discount	1	-15,673.50	130,788	-2,049,905,718.00
			Sub-Total	-1,879,082,203.32
				-2,049,905,718.00
			Total	1,679,246,869.80
				1,135,262,682.00
			3-Y Cost of Ownership (CNY):	2,814,509,552.00
			tpmC	707,351,007
			CNY/tpmC	3.98
Audited by Doug Johnson of InfoSizing.				
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. For complete details, see the pricing section of the TPC benchmark specifications. If you find that stated prices are not available according to these terms, please inform the TPC at pricing@tpc.org . Thank you.				



Alibaba Cloud Elastic Compute Service Cluster (with 1554 OceanBase Data Nodes)

TPC-C 5.11.0
TPC-Pricing 2.5.0

Report Date
May 15, 2020

Numerical Quantities Summary:

MQTh (Maximum Qualified Throughput)

707,351,007 tpmC

Response Time (sec.)	Min	Average	90th	Max
New-Order	0.103	0.127	0.160	5.555
Payment	0.104	0.123	0.154	5.527
Order-Status	0.102	0.117	0.141	5.354
Delivery (Interactive)	0.101	0.101	0.101	0.970
Delivery (Deferred)	0.008	1.550	6.340	13.376
Stock-Level	0.102	0.120	0.152	5.087
Menu	0.101	0.101	0.101	21.932

Emulated Display Delay: 0.1 sec.

Transaction Mix	Percent	Number
New-Order	44.960%	339,528,483,405
Payment	43.010%	324,805,893,484
Order-Status	4.010%	30,282,810,672
Delivery	4.010%	30,283,015,648
Stock-Level	4.010%	30,282,852,629

Keying Times (sec.)	Min	Average	Max
New-Order	18.002	18.068	19.744
Payment	3.002	3.065	4.739
Order-Status	2.002	2.063	3.735
Delivery	2.002	2.063	3.736
Stock-Level	2.002	2.063	3.743

Think Times (sec.)	Min	Average	Max
New-Order	0.002	12.065	121.456
Payment	0.002	12.065	121.558
Order-Status	0.002	10.065	101.152
Delivery	0.002	5.064	51.159
Stock-Level	0.002	5.064	51.156

Test Duration

Ramp-up Time	55 min.
Measurement Interval (MI)	480 min.
Checkpoints in MI per Node (Min/Avg/Max)	23/23.18/24
Maximum Checkpoint Interval (All Nodes)	22:27
Number of transactions (all types) completed in Measurement Interval	755,183,055,838

0	General Items.....	14
0.1	Application Code Disclosure.....	14
0.2	Sponsor.....	14
0.3	Parameter Settings.....	14
0.4	Configuration Diagrams.....	14
1	Clause 1: Logical Database Design Related Items.....	18
1.1	Table Definitions.....	18
1.2	Physical Organization of Database.....	18
1.3	Insert and/or Delete Operations.....	18
1.4	Partitioning.....	18
1.5	Replication of tables.....	19
2	Clause 2: Transaction and Terminal Profiles Related Items.....	20
2.1	Random Number Generation.....	20
2.2	Input/output Screen Layouts.....	20
2.3	Verification of Terminal Features.....	20
2.4	Presentation Manager or Intelligent Terminal.....	20
2.5	Percentage of Home and Remote Order-lines.....	20
2.6	Percentage of Rolled Back New-Orders.....	20
2.7	Number of Items per New-Order.....	20
2.8	Percentage of Home and Remote Payments.....	20
2.9	Percentage of Access by Non-Primary Key.....	20
2.10	Percentage of Skipped Delivery Transactions.....	21
2.11	Numeric Quantities for Transactions and Terminal.....	21
2.12	Queuing Mechanism.....	21
3	Clause 3 Transaction and System Properties Related Items.....	22
3.1	Transaction System Properties (ACID).....	22
3.2	Atomicity.....	22
3.2.1	Completed Transaction.....	22
3.2.2	Aborted Transaction.....	22
3.3	Consistency.....	22
3.4	Isolation Tests.....	23
3.4.1	Isolation Test 1.....	23
3.4.2	Isolation Test 2.....	23
3.4.3	Isolation Test 3.....	23
3.4.4	Isolation Test 4.....	24
3.4.5	Isolation Test 5.....	24
3.4.6	Isolation Test 6.....	24
3.4.7	Isolation Test 7.....	24
3.4.8	Isolation Test 8.....	25
3.4.9	Isolation Test 9.....	25
3.5	Durability.....	25
3.5.1	Instantaneous Interruption, Power Failure, Failure of Memory, Loss of Log, Network Failure, Loss of Durable Media Containing Database Tables.....	25
3.5.2	Power Failure, Full cluster failure.....	26
4	Clause 4: Scaling and Database Population Related Items.....	27
4.1	Initial Cardinality of Tables.....	27
4.2	Distribution of Tables and Logs.....	27
4.3	Data Model and Interface of Database.....	27
4.4	Mapping of Database Partitions/Replications.....	27
4.5	60-Day Space Computation.....	28
5	Clause 5: Performance Metrics and Response Time Related Items.....	30
5.1	Measured tpmC.....	30
5.2	Response Times.....	30
5.3	Keying and Think Times.....	30
5.4	Response Time Frequency Distribution Curves.....	30
5.5	Think Time Frequency Distribution.....	33
5.6	Response Times versus Throughput.....	34
5.7	Throughput versus Elapsed Time.....	34

5.8	Steady State Determination	34
5.9	Work Performed During Steady State	35
5.10	Measurement Interval	35
5.11	Transaction Mix Regulation	35
5.12	Transaction Mix	35
5.13	Percentage of New-Order Transactions	35
5.14	Number of Order-lines per New-Order	35
5.15	Percentage of Remote Order-lines per New-Order	35
5.16	Percentage of Remote Payments	36
5.17	Percentage of access customer by C_LAST for Payment and Order-Status	36
5.18	Percentage of Skipped Delivery Transactions	36
5.19	Checkpoints	36
6	Clause 6: SUT, Driver and Communications Related Items	37
6.1	RTE Description	37
6.2	Number of Terminal Connections Lost	37
6.3	Emulated Components	37
6.4	Configuration Diagrams	37
6.5	Network Configuration	38
6.6	Operator Intervention	38
7	Clause 7: Pricing Related Items	39
7.1	Hardware and Software Price	39
7.2	Total 3-Year Cost	39
7.3	Availability Date	39
7.4	Hardware and Software Support	39
7.5	Statement of measured tpmC and Price/Performance	39
7.6	Country Specific Pricing	40
7.7	Orderability Date	40
8	Clause 8: Auditor Attestation	41
8.1	Auditor Information	41
8.2	Attestation Letter	41
	Appendix A: Source Code Files List	44
	Appendix B: Database Design	45
	create-tenant.sql	45
	create-user.sql	45
	create-procedure.sql	45
	RTE parameters	52
	Appendix C: Configuration Options	53
	Database Node	53
	File system info	53
	/etc/security/limits.conf	53
	Observer.cfg	53
	Client Node	57
	File system info	57
	/etc/security/limits.conf	57
	Nginx.conf	57
	OCP Node	60
	File system info	60
	/etc/security/limits.conf	60
	Appendix D: Third Party Price Quotes	61
	ecs.i2d.21xlarge	61
	ecs.ebmg6.26xlarge	62
	Alibaba Cloud Support Price	63
	Lenovo ThinkPad X1 Yoga 2019 14" Laptop	64
	OpenResty Support Price	65

List of Figures

Figure 1 Measured Configuration	16
Figure 2 Priced Configuration	17
Figure 3 Frequency distribution of response times for New-Order	30
Figure 4 Frequency distribution of response times for Payment	31
Figure 5 Frequency distribution of response times for Delivery (interactive)	31
Figure 6 Frequency distribution of response times for Delivery (deferred)	32
Figure 7 Frequency distribution of response times for Order-Status	32
Figure 8 Frequency distribution of response times for Stock-Level	33
Figure 9 Frequency distribution of think times for New-Order	33
Figure 10 New-Order Response Time versus Throughput	34
Figure 11 New-Order throughput versus Time	34

List of Tables

<i>Table 1 ecs.i2d.21xlarge Configuration</i>	14
<i>Table 2 ecs.ebmg6.26xlarge Configuration</i>	15
<i>Table 3 ecs.ebmg6.26xlarge Configuration</i>	15
<i>Table 4 Table Customer Partition Information</i>	18
<i>Table 5 Numeric Quantities for Transactions and Terminal</i>	21
<i>Table 6 Table Cardinalities</i>	27
<i>Table 7 Database mapping-1</i>	28
<i>Table 8 Database mapping-2</i>	28
<i>Table 9 Space Computations (Note: All numbers about space are in GB)</i>	29
<i>Table 10 RTE parameters input</i>	37
<i>Table 11 Statement of tpmC and price/performance</i>	39

Preface

This report documents the compliance of Ant Financial (Hang Zhou) Network Technology Co., Ltd. TPC Benchmark™ C testing on Alibaba Cloud Elastic Compute Service Cluster (with 1554 OceanBase Data Nodes) running OceanBase v2.2 Enterprise Edition with Partitioning, Horizontal Scalability and Advanced Compression, executing the TPC Benchmark™ C Standard, Revision 5.11.0.

The TPC Benchmark™ C Full Disclosure Report is organized as follows:

The main body of the document lists each item in Clause 8 of the TPC Benchmark™ C Standard and explains how each specification is satisfied.

- Appendix A contains the list of source code in supporting files package, which contains Nginx module code, acid code and load database code.
- Appendix B contains the code used to create the database and 5 kinds of TPC-C transactions implemented in Stored Procedure Language.
- Appendix C contains the configuration information for the Aliyun Linux 2, Nginx 1.15.8 and OceanBase v2.2 Enterprise Edition with Partitioning, Horizontal Scalability and Advanced Compression.
- Appendix D contains the third-party price quotes for Alibaba Cloud Elastic Compute Service instances: ecs.i2d.21xlarge, ecs.ebmg6.26xlarge, Alibaba Cloud Support Price, Lenovo ThinkPad X1 Yoga 2019 14" Laptop and OpenResty Support Price.

Alibaba Cloud Elastic Compute Service Cluster TPC Benchmark™ C Full Disclosure

Introduction

The TPC Benchmark™ C Standard Specification requires test sponsors to publish, and make available to the public, a full disclosure report for the results to be considered compliant with the Standard.

This report is intended to satisfy the Standard's requirement for full disclosure. It documents the compliance of the benchmark tests required in the TPC Benchmark™ C results for the Alibaba Cloud Elastic Compute Service Cluster (with 1554 OceanBase Data Nodes) running OceanBase v2.2 Enterprise Edition with Partitioning, Horizontal Scalability and Advanced Compression.

In the *Standard Specification*, the main headings in Clause 8 are keyed to the other clauses. The headings in this report use the same sequence, so that they correspond to the titles or subjects referred to in Clause 8.

Each section in this report begins with the text of the corresponding item from Clause 8 of the *Standard Specification*, printed in italic type. The plain type text that follows explains how the tests comply with the TPC-C Benchmark.

0 General Items

0.1 Application Code Disclosure

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 and output functions.

Appendix A contains the file list of application source code that handles terminal input and output, communicates with database and implements deferred delivery. The files can be found in supporting files package.

0.2 Sponsor

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

This benchmark test was sponsored by Ant Financial (Hang Zhou) Network Technology Co., Ltd.

0.3 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 tuning options^[SEP]
- Recovery/commit options
- Consistency/locking options^[SEP]
- Operating system and application configuration parameters

Appendix C contains parameter settings for the ecs.i2d.21xlarge OceanBase Cluster nodes, ecs.ebm6.26xlarge OceanBase Cloud Platform (OCP) nodes and ecs.ebm6.26xlarge client nodes, all Aliyun Linux 2 tunable parameters, along with parameters for OceanBase v2.2 Enterprise Edition with Partitioning, Horizontal Scalability and Advanced Compression, Nginx 1.15.8.

0.4 Configuration Diagrams

Diagrams of both measured and priced configurations must be provided, accompanied by a description of the differences.

The instances (nodes) in the Alibaba Cloud Elastic Compute Service Cluster are configured based on a selected number of virtual CPUs (vCPU). A vCPU corresponds to one thread of the underlying physical processor.

Database Nodes Description

The SUT is an Alibaba Cloud Elastic Compute Service Cluster (with 1554 OceanBase Data Nodes) running OceanBase v2.2 Enterprise Edition with Partitioning, Horizontal Scalability and Advanced Compression. The configuration of ecs.i2d.21xlarge instances is described below:

	ecs.i2d.21xlarge
Processors	Intel Xeon Platinum 8163(Skylake) 2.5GHz
Processors/Cores/Threads	84 vCPU
Memory	712 GB
Cloud Disk (for OS)	40GB Ultra Cloud Disk
Cloud Disk (for HOME)	200GB Ultra Cloud Disk
Local Storage	4 * 3570GB NVMe SSD

Table 1 ecs.i2d.21xlarge Configuration

The 1557 database nodes consist of 1554 data nodes and 3 root nodes. OceanBase Cluster stores tables, indexes and logs on data nodes' local disks. Root nodes store the metadata of OceanBase Cluster and provide some important

capabilities such as balancing storage space utilization among data nodes etc. Among the three root nodes, one is leader and the other two are followers.

All nodes in OceanBase Cluster are connected by one VSwitch and communicate over a network within a Virtual Private Cloud (VPC) provided by Alibaba Cloud.

OCP (OceanBase Cloud Platform) Nodes Description

OceanBase Cloud Platform (OCP) is used to manage and monitor OceanBase Cluster. Database administrators can use OCP to deploy OceanBase Cluster instance, start or stop instance, monitor instance's status. There are three ecs.ebmg6.26xlarge nodes for OCP. Each of the nodes is configured as described below:

	ecs.ebmg6.26xlarge
Processors	Intel(R) Xeon(R) Platinum 8269CY CPU @ 2.50GHz
Processors/Cores/Threads	104 vCPU
Memory	384 GB
Cloud Disk (for OS)	40GB Ultra Cloud Disk
Cloud Disk (for HOME)	200GB Ultra Cloud Disk

Table 2 ecs.ebmg6.26xlarge Configuration

OCP Nodes are connected to OceanBase Cluster by one VSwitch and communicate over a network within a Virtual Private Cloud (VPC) provided by Alibaba Cloud.

Client Configuration Description

The TPC-C transactions are entered via a Remote Terminal Emulator that communicates with Nginx running on 400 ecs.ebmg6.26xlarge clients. Each client is configured with:

	ecs.ebmg6.26xlarge
Processors	Intel(R) Xeon(R) Platinum 8269CY CPU @ 2.50GHz
Processors/Cores/Threads	104 vCPU
Memory	384 GB
Cloud Disk (for OS)	40GB Ultra Cloud Disk
Cloud Disk (for HOME)	200GB Ultra Cloud Disk

Table 3 ecs.ebmg6.26xlarge Configuration

The client systems receive transactions via Nginx that communicates with OBProxy that connects to the OceanBase Cluster.

The nodes of OceanBase Cluster, the nodes of OceanBase Cloud Platform (OCP), the nodes of client and the nodes of RTE are in the same network. They use the VSwitch within the same VPC.

There are two other networks in the system that does not participate in transactions processing. The first is a separate network named Internet IP address, which is public network. Each Elastic Compute Service instance has an Internet IP. Users can logon and use ECS instances (all OceanBase Cluster nodes, all OCP nodes, all client nodes and all RTE nodes) by the public network. The other network provides administration for all ECS instances; users can access Alibaba Cloud Management Console by web browser and manage their ECS instances, such as start or stop the instance, monitor instance running status, etc.

Figure 1 shows the measured configuration and Figure 2 the priced configuration. The only difference between the two is that RTEs are showed in measured configuration.

Measured Configuration

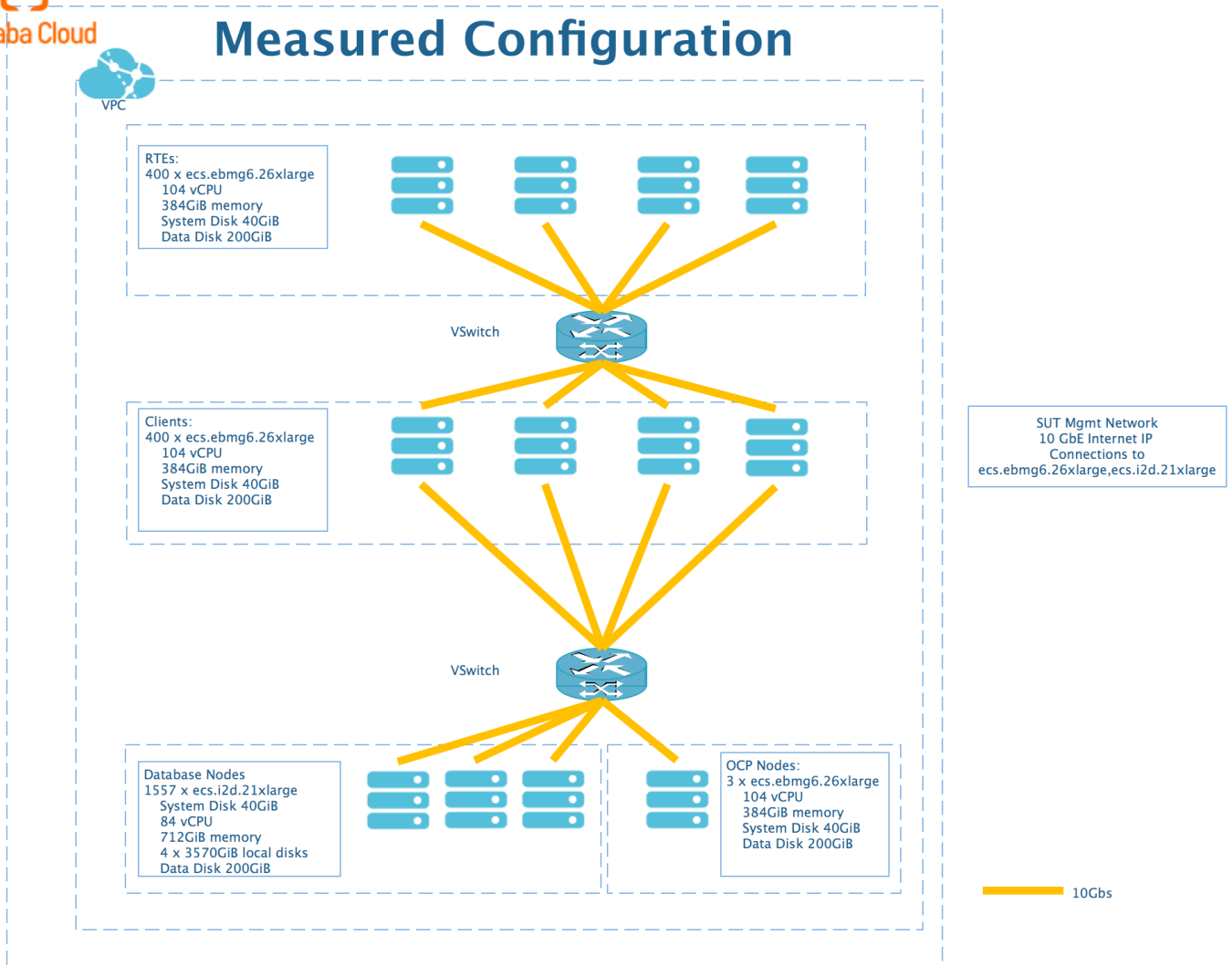


Figure 1 Measured Configuration

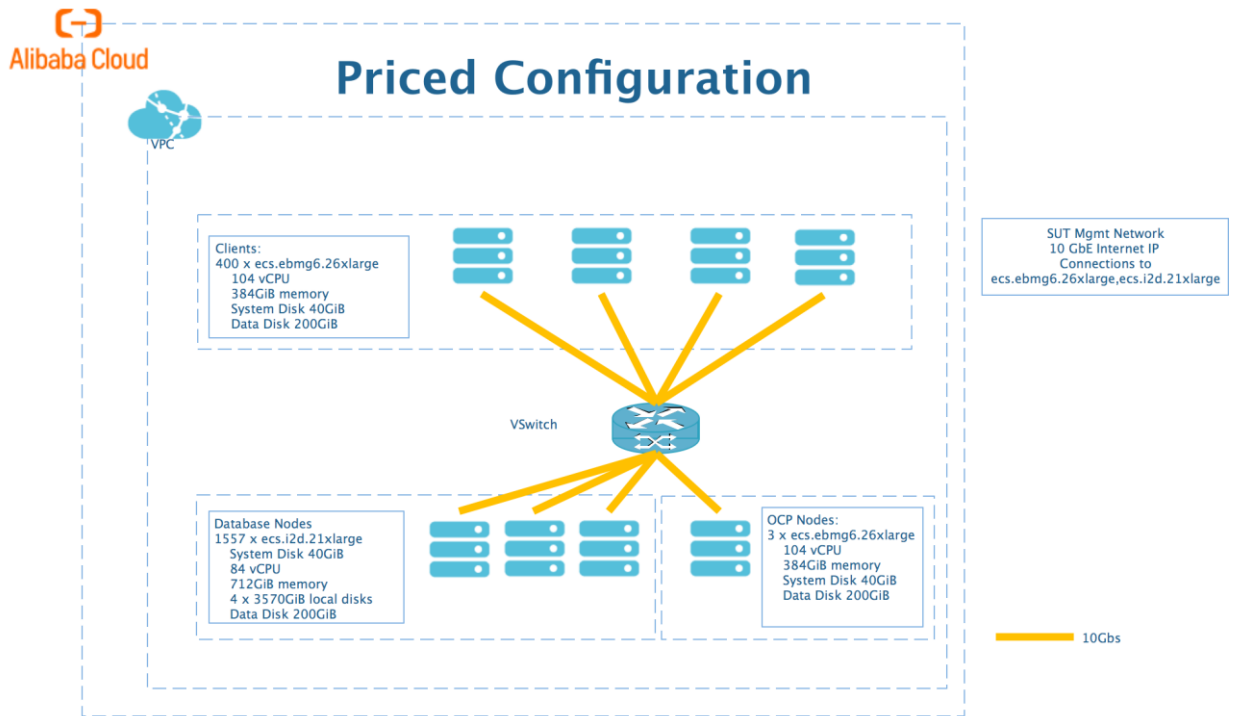


Figure 2 Priced Configuration

1 Clause 1: Logical Database Design Related Items

1.1 Table Definitions

Listing must be provided for all table definition statements and all other statements used to set up the database. Scripts used to set up the OceanBase database used for this testing are described in Appendix B.

1.2 Physical Organization of Database

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

Detailed SUT environment is described in section 0.4. Each database node ecs.i2d.21xlarge has 4 SSD disks (3,570GB NVMe each). These raw devices are formatted and managed by Aliyun Linux 2. OceanBase v2.2 Enterprise Edition with Partitioning, Horizontal Scalability and Advanced Compression creates data files and log files on these devices by file system APIs. All database objects including table, index, and log are stored on these local disks. The scripts used to create tables and indices are in Appendix B.

1.3 Insert and/or 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.

All insert and delete operations were verified and there is no restriction during the entire benchmark.

1.4 Partitioning

While there are a few restrictions placed upon horizontal or vertical partitioning of tables and rows in the TPC-C benchmark (see Clause 1.6), any such partitioning must be disclosed.

Tables except ITEM are partitioned, Customer is partitioned both horizontally and vertically, and other tables are horizontally partitioned only.

Customer	
C_VP1_HP1	C_W_ID
	C_D_ID
	C_ID
	C_DISCOUNT
	C_CREDIT
	C_LAST
	C_FIRST
	C_MIDDLE
	C_BALANCE
	C_YTD_PAYMENT
	C_PAYMENT_CNT
	C_CREDIT_LIM
	C_STREET_1
	C_STREET_2
	C_CITY
	C_STATE
	C_ZIP
C_PHONE	
C_SINCE	
C_DELIVERY_CNT	
-----vertical partition-----	
C_VP2_HP1	C_DATA

Table 4 Table Customer Partition Information

1.5 Replication of tables

Replication of tables, if used, must be disclosed (see Clause 1.4.6).

ITEM table is fully replicated on every OceanBase Cluster data node (ECS instance). There is a full replica of ITEM on every OceanBase Cluster data node (ECS instance).

All other tables are replicated into 3 replicas: one full replica, one data replica and one log replica that are distributed among database nodes. A full replica contains cardinality, in-memory increments (mutations) that could be checkpointed to disk and redo log of the corresponding table. A data replica contains cardinality, checkpoints of in-memory increments (mutations) and redo log (both from the full replica). A log replica contains redo log only.

2 Clause 2: Transaction and Terminal Profiles Related Items

2.1 Random Number Generation

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

The Random Number Generators used were `lrand48_r()` and `drand48_r()` LINUX calls.

2.2 Input/output Screen Layouts

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

All screen layouts followed the specification exactly. The source code used to generate the screens is available in supporting files package. Appendix A describes the list of source code files.

2.3 Verification of Terminal Features

The method used to verify that the emulated terminals provide all the features described in Clause 2.2.2.4 must be explained.

The auditors manually exercising each specification during the onsite audit portion of this benchmark verified the terminal attributes.

2.4 Presentation Manager or Intelligent Terminal

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

No presentation manager software or intelligent terminal features are used. The response data is passed to the terminals using the HTML format, which can be displayed with any standard Web browser. The application source code files for the HTML display generation is listed in Appendix A.

2.5 Percentage of Home and Remote Order-lines

The percentage of home and remote order-lines in the New-Order transactions must be disclosed.

The Numeric Quantities for Transactions and Terminal Table contains the percentage of home and remote order-lines for all of the New-Order transactions completed during the measurement interval.

2.6 Percentage of Rolled Back New-Orders

The percentage of New-Order transactions that were rolled back as a result of an unused item number must be disclosed.

The Numeric Quantities for Transactions and Terminal table contains the percentage of New-Order transactions that were rolled back due to an unused item number during the measurement interval.

2.7 Number of Items per New-Order

The number of items per orders entered by New-Order transactions must be disclosed.

The Numeric Quantities for Transactions and Terminal table contains the average number of items in each New-Order transaction during the measurement interval.

2.8 Percentage of Home and Remote Payments

The percentage of home and remote Payment transactions must be disclosed.

The Numeric Quantities for Transactions and Terminal table contains the percentage of home and remote Payment transactions during the measurement interval.

2.9 Percentage of Access by Non-Primary Key

The percentage of Payment and Order-Status transactions that used non-primary key (C_LAST) access to the database must be disclosed.

The Numeric Quantities for Transactions and Terminal table contains the percentage of Payment and Order-Status transactions that were accessed by non-primary key (C_LAST) access during the measurement interval.

2.10 Percentage of Skipped Delivery Transactions

The percentage of Delivery transactions that were skipped as a result of an insufficient number of rows in the NEW-ORDER table must be disclosed.

The Numeric Quantities for Transactions and Terminal table contains the percentage of Delivery transactions that were “skipped” due to insufficient number of rows in the NEW-ORDER table.

2.11 Numeric Quantities for Transactions and Terminal

The mix (i.e., percentages) of transaction types seen by the SUT must be disclosed.

The Numeric Quantities for Transactions and Terminal table contains the percentage of each transaction type executed by the SUT and other numeric quantities for transactions and terminal.

New-Order	
Percentage of Home order-lines	99.000%
Percentage of Remote order-lines	1.000%
Percentage of Rolled Back Transactions	1.000%
Avg. Number of Items per Transactions	10.000
Payment	
Percentage of Home Transactions	85.010%
Percentage of Remote Transactions	14.990%
Access by C_LAST (Non-primary key)	
Percentage of Payment Transactions	60.000%
Percentage of Order-Status Transactions	60.000%
Delivery	
Skipped transactions	0
Transaction Mix	
New-Order	44.960%
Payment	43.010%
Order-Status	4.010%
Delivery	4.010%
Stock-Level	4.010%

Table 5 Numeric Quantities for Transactions and Terminal

2.12 Queuing Mechanism

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

Delivery transactions were handled by Nginx 1.15.8 like other transactions. The difference was that after parsing the parameters in the request, the Nginx worker created a task and put the task to delivery tasks pool. Besides worker threads inside Nginx, there were an amount of threads that serve the delivery tasks pool. Nginx worker threads were able to complete the on-line part of the Delivery transaction and immediately return successful queuing responses to the drivers. The threads servicing the tasks pool were responsible for completing the deferred part of the transaction asynchronously. They accessed OceanBase Cluster by ODBC, sent Stored Procedure Calls and got execution results.

3 Clause 3 Transaction and System Properties Related Items

3.1 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.

This section defines each of ACID properties, describes a series of tests done to demonstrate that OceanBase database compliances with the standard described in the specification.

3.2 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.

3.2.1 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, DISTRICT, and WAREHOUSE tables have been changed appropriately.

A row was randomly selected from customer table and the balance of the selected customer was noted. The balance of the warehouse and the district that the selected customer belonged to were noted too. With the warehouse, district and customer identifier selected above and a known amount, a payment transaction was started. The payment transaction was committed and the rows in warehouse, district and customer were verified to contain updated balances correctly.

The test was repeated using rows selected randomly from the warehouse, district and customer tables. The warehouse id of the customer selected is different from the warehouse id of the rows selected from warehouse and district. The balances are noted. A payment transaction was started using the selected warehouse, district and customer identifiers and a known amount. The payment was committed and the rows in warehouse, district and customer were verified to contain updated balances correctly.

3.2.2 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, DISTRICT, and WAREHOUSE tables have NOT been changed.

A row was randomly selected from customer table and the balance of the selected customer was noted. The balance of the warehouse and the district that the selected customer belonged to were noted too. A payment transaction was started with the same warehouse, district and customer identifier and a known amount. The payment transaction was rolled back and the rows were verified to contain the original balances.

The test was repeated using rows selected randomly from the warehouse, district and customer tables. The warehouse id of the customer selected was different from the warehouse id of the rows selected from warehouse and district. The balances were noted. A payment transaction was started using the selected warehouse, district and customer identifiers and a known amount. The payment transaction was rolled back and the rows were verified to contain the original balances.

3.3 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 that the database is initially in a consistent state.

There are four consistency conditions needed to be demonstrated explicitly according to the benchmark specification:

- 1) The sum of the district year-to-date balance (d_ytd) for all districts in a warehouse equals to the balance of the warehouse year-to-date balance (w_ytd) for the district's warehouse;
- 2) For each district, the next order id (d_next_o_id) minus one equal to the most recent order id (max (o_id)) from the ORDER table (ORDR) for the associated district and warehouse. The next order id (d_next_o_id) minus one is equal to the most recent new-order id (max (no_o_id)) from the NEW-ORDER table (NORD);

- 3) For each district, the maximum order id (max (no_o_id)) minus the minimum order id (min (no_o_id)) in the NEW-ORDER table (NORD) 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 in the ORDER table (ORDR) equals to the number of rows in the ORDER-LINE table (ORDL) for the same district;

These consistency conditions were tested using a C program to issue queries to the 1557-nodes OceanBase Cluster. For Consistency Conditions 2 and 4, the program samples the first, last, and two other random warehouses.

The four consistency tests were performed twice, one was after data loading and the other was after the reported performance run. The results of the queries verified that the database was consistent for all four tests.

3.4 Isolation Tests

The specification defines nine tests used to demonstrate that required level of transaction isolation is met in SUT database.

These tests are performed on the 1557-nodes OceanBase Cluster configured for 55,944,000 warehouses. These tests were executed twice. In the first case, New-Order and Payment transactions were “local”. For New-Order transaction, all items have the same supply warehouse id as the customer. For Payment transaction, customer’s warehouse id was the same as payment warehouse id. In the second case, New-Order and Payment transactions were “remote”, items ids, customer’s warehouse id, payment id were generated randomly and different from each other.

In both cases, all tests have been verified to demonstrate the desired transaction isolation level.

3.4.1 Isolation Test 1

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

The test proceeds as follows:

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

3.4.2 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 test proceeds as follows:

- 1) For a randomly selected customer, An Order-Status transaction T0 was executed and committed. The order returned was noted.
- 2) For the same customer used in T0, A New-Order transaction T1 was started. T1 stopped immediately prior to ROLLBACK.
- 3) For the same customer used in T1, An Order-Status transaction T2 was started. 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) For the same customer used in T1, An Order-Status transaction T3 was started. T3 returned the same order that T0 had returned.

3.4.3 Isolation Test 3

This test demonstrates isolation for write-write conflicts of two New-Order transactions when both transactions are committed.

The test proceeds as follows:

- 1) The D_NEXT_O_ID of a randomly selected district was retrieved.

- 2) For a randomly selected customer within the district selected in step 1, A New-Order transaction T1 was started. T1 stopped immediately prior to COMMIT.
- 3) For the same customer used in T1, Another New-Order transaction T2 was started and 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.

3.4.4 Isolation Test 4

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

The test proceeds as follows:

- 1) The D_NEXT_O_ID of a randomly selected district was retrieved.
- 2) For a randomly selected customer within the district selected in step 1, A New-Order transaction T1 was started. T1 stopped immediately prior to ROLLBACK.
- 3) For the same customer used in T1, Another New-Order transaction T2 was started. T2 waited.
- 4) T1 was allowed to ROLLBACK, 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.

3.4.5 Isolation Test 5

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

The test proceeds as follows:

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

3.4.6 Isolation Test 6

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

The test proceeds as follows:

- 1) For a randomly selected warehouse and district, A query was executed to find out the customer who is to be updated by the next delivery transaction
- 2) The C_BALANCE of the customer selected in step 1 was retrieved.
- 3) For the same warehouse used in step 1, A Delivery transaction T1 was started. T1 was stopped immediately prior to ROLLBACK.
- 4) For the same customer found in step 1, A Payment transaction T2 was started. T2 waited.
- 5) T1 was allowed to ROLLBACK. T2 completed and was committed. The C_BALANCE of the customer selected in step 1 was retrieved again. The C_BALANCE reflected the results of only T2.

3.4.7 Isolation Test 7

This test demonstrates repeatable reads for the New-Order transaction while an interactive transaction updates the prices of some items.

The test proceeds as follows:

- 1) The I_PRICE of two randomly selected items X and Y were retrieved.
- 2) A New-Order transaction T1 with several items including items X and Y was started. T1 was stopped after querying the price of item X the first time and immediately before querying the prices of item Y and of item X the second time.
- 3) A transaction T2 was started to increase the price of items X and Y by 10%.
- 4) T2 did not stall and was committed.
- 5) T1 was resumed, and the prices of Y and X (the second time) were retrieved within T1. The prices of items X and Y were the same as those retrieved in step 1.
- 6) T1 was committed.
- 7) The prices of items X and Y were retrieved again. The values were the same as the values set by T2. Execution followed *Case D* of *Clause 3.4.2.7*.

3.4.8 Isolation Test 8

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

The test proceeds as follows:

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

3.4.9 Isolation Test 9

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

The test proceeds as follows:

- 1) For a randomly selected customer, An Order-Status transaction T1 was started.
- 2) T1 stopped immediately after reading the order table for the selected customer to find the most recent order for that customer.
- 3) For the same customer, A New-Order transaction T2 was started. T2 completed and was committed without being blocked by T1.
- 4) For the same customer, T1 resumed and the ORDER table was read again to determine the most recent order. The order found was the same as the one found in step 2.
- 5) T1 completed and was committed.

3.5 Durability

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 3.5.3.

3.5.1 Instantaneous Interruption, Power Failure, Failure of Memory, Loss of Log, Network Failure, Loss of Durable Media Containing Database Tables

This test was executed by following steps:

- 1) The total number of orders was determined by the sum of D_NEXT_O_ID from all rows in the district table giving the beginning count.
- 2) The RTE was started with full user load.

- 3) The test was allowed to ramp up and ran into steady state within 30 minutes.
- 4) The test finished a steady run of which measurement interval is 20 minutes. A checkpoint is finished during the measurement interval.
- 5) Logon Aliyun management platform to destroy an ECS instance, which was a randomly selected Data Node of OceanBase Cluster.
- 6) Because of multi-replica, the OceanBase Cluster recovered automatically and restored to normal state in several minutes.
- 7) The RTE reported some errors and restored to steady state in several minutes. The test continued in steady state for more than 5 minutes.
- 8) Logon Aliyun management platform to destroy an ECS instance, which was the leader of three root nodes for OceanBase Cluster.
- 9) Because of multi-replica, the OceanBase Cluster recovered automatically and restored to normal state in several minutes.
- 10) The RTE reported some errors and continued running for several minutes then finished. The RTE recorded all successful New-Order transactions to a success file during testing period.
- 11) Records from the success file were compared against the OceanBase Cluster to verify that no committed New-Order transactions were lost and no rollbacked New-Order transactions in database. The total number of all committed New-Order transactions was counted.
- 12) Repeat step 1 to determine the total number of orders.
- 13) The difference between the counts in Steps 1 and 12 was compared with the RTE count in success file to verify that the difference was less than or equal to the number of terminals simulated.
- 14) Consistency Test 3 was verified.

3.5.2 Power Failure, Full cluster failure

This test is justified by documentation. Alibaba Cloud Elastic Compute Service features (the combination of the UPS and the diesel generators) comply with the 30 minutes UPS requirement.

4 Clause 4: Scaling and Database Population Related Items

4.1 Initial Cardinality of Tables

The cardinality (e.g. number of rows) of each table, as it existed at the start of the benchmark run (see Clause 4.2), must be disclosed. If the database was over-scaled and inactive rows of the WAREHOUSE table were deleted (see Clause 4.2.2) the cardinality of the WAREHOUSE table as initially configured and the number of rows deleted must be disclosed.

There are 55,944,000 warehouses in OceanBase database during the test. The following table shows the initial cardinality of the tables after table population and the cardinality prior to the measurement run.

Table	Initial Row Count	Row Count Prior to Measured Run
Warehouse (WARE)	55,944,000	55,944,000
District (DIST)	559,440,000	559,440,000
Customer (CUST)	1,678,320,000,000	1,678,320,000,000
History (HIST)	1,678,320,000,000	1,678,320,000,000
Order (ORDR)	1,678,320,000,000	1,678,320,000,000
New order (NORD)	503,496,000,000	503,496,000,000
Order line (ORDL)	16,783,241,432,501	16,783,241,432,501
Stock (STOK)	5,594,400,000,000	5,594,400,000,000
Item (ITEM)	100,000	100,000

Table 6 Table Cardinalities

4.2 Distribution of Tables and Logs

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

Details for a description of the distribution of tables and logs are described in section 0.4 Configuration. Each data node of OceanBase Cluster has data files and log files on local disks.

Tables except ITEM are partitioned, Customer is partitioned both horizontally and vertically, and other tables are horizontally partitioned only. Each partition has three replicas, one full replica which contains cardinality, in-memory increment which is periodically (~21 minutes) checkpointed to disk and redo log, and one data replica which contains cardinality, checkpoint files and redo log (both from the full replica), and one log replica which contains redo log only. These replicas are distributed across all data nodes of OceanBase Cluster. ITEM table is fully replicated on every data node of OceanBase Cluster.

4.3 Data Model and Interface of Database

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 (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.

OceanBase v2.2 Enterprise Edition with Partitioning, Horizontal Scalability and Advanced Compression is a distributed relational database management system. TPC-C transactions were implemented in SQL stored procedures and invoked via Open Database Connectivity (ODBC) interface. The application code files are listed in Appendix A.

4.4 Mapping of Database Partitions/Replications

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

Tables except ITEM are partitioned, Customer is partitioned both horizontally and vertically, and other tables are horizontally partitioned only. The method for horizontal partition is RANGE with 46620 partitions. Each partition has

three replicas. ITEM table is fully replicated on every data node of OceanBase Cluster. Except ITEM, all other tables form a table group tpcc_group and the load balancer take partitions with the same partition ID of all tables in a table group as one entity, referred as a partition group. All replicas are distributed across all data nodes of OceanBase Cluster (ECS instances) automatically.

Customer		Stock		Warehouse		
C_VP1_HP1	C_W_ID	S_HP1	S_I_ID	W_HP1	W_ID	
	C_D_ID		S_W_ID		W_YTD	
	C_ID		S_ORDER_CNT		W_TAX	
	C_DISCOUNT		S_YTD		W_NAME	
	C_CREDIT		S_REMOTE_CNT		W_STREET_1	
	C_LAST		S_QUANTITY		W_STREET_2	
	C_FIRST		S_DATA		W_CITY	
	C_MIDDLE		S_DIST_01		W_STATE	
	C_BALANCE		S_DIST_02		W_ZIP	
	C_YTD_PAYMENT		S_DIST_03		District	
	C_PAYMENT_CNT		S_DIST_04	D_HP1	D_W_ID	
	C_CREDIT_LIM		S_DIST_05		D_ID	
	C_STREET_1		S_DIST_06		D_NEXT_O_ID	
	C_STREET_2		S_DIST_07		D_TAX	
	C_CITY		S_DIST_08		D_YTD	
	C_STATE		S_DIST_09		D_NAME	
	C_ZIP		S_DIST_10		D_STREET_1	
	C_PHONE				D_STREET_2	
	C_SINCE		New-Order		D_CITY	
	C_DELIVERY_CNT		NO_HP1		NO_W_ID	D_STATE
-----vertical partition-----			NO_D_ID	D_ZIP		
C_VP2_HP1	C_DATA		NO_O_ID			

Table 7 Database mapping-1

Order		Order-Line		History	
O_HP1	O_W_ID	OL_HP1	OL_W_ID	H_HP1	H_C_ID
	O_D_ID		OL_D_ID		H_C_D_ID
	O_ID		OL_O_ID		H_C_W_ID
	O_C_ID		OL_NUMBER		H_D_ID
	O_CARRIER_ID		OL_DELIVERY_D		H_W_ID
	O_OL_CNT		OL_AMOUNT		H_DATE
	O_ALL_LOCAL		OL_I_ID		H_AMOUNT
	O_ENTRY_D		OL_SUPPLY_W_ID		H_DATA
	OL_QUANTITY				
	OL_DIST_INFO				

Table 8 Database mapping-2

As an example, Partition Group 1 contains following partitions: {C_VP1_HP1, C_VP2_HP1, S_HP1, NO_HP1, W_HP1, D_HP1, O_HP1, OL_HP1, H_HP1}

Three replicas of the partition group are distributed on three data nodes automatically.

Each data node of OceanBase Cluster has 14,480GB storage capacity, which consists of 4 SSD disks and 1 ultra cloud disk.

4.5 60-Day Space Computation

Details of the 60-day space computations along with proof that the database is configured to sustain 8 hours of growth for the dynamic tables (Order, Order-Line, and History) must be disclosed (see Clause 4.2.3).

Terminals	559,440,000	tpmC	707,351,007	tpmC/W	12.64	
Table	Population (row)	Data	Index	5 % Space	8-H Space	Total Space
Warehouse (WARE)	55,944,000	183		10		193
District (DIST)	559,440,000	365		19		384
Customer (CUST)	1,678,320,000,000	1,596,461	55,386	82,593		1,734,440
Item (ITEM)	100,000	13		1		14
Stock (STOK)	5,594,400,000,000	3,111,423		155,572		3,266,995
New-Order (NORD)	503,496,000,000	365		19		384
Order (ORDR)	1,678,320,000,000	12,748	13,112		2,579	28,439
Order-Line (ORDL)	16,783,241,432,501	979,253			198,105	1,177,358
History (HIST)	1,678,320,000,000	74,532			15,078	89,610
Totals		5,775,343	68,498	238,214	215,762	6,297,817
OceanBase Space Usage	Data Nodes	Allocated		Free Space		Growth
After DB Load	1,554	6,188,028		344,169		-
After 8-hour Perf. Run	1,554	6,188,028		8,771		335,397
Free space	344,187					
Dynamic Space	1,066,533					
Static Space	4,777,308					
Daily Growth	215,762					
Daily Spread	20,543					
60-Day Space	18,955,656					
8-Hour Checkpoint	830,913					
	Data Nodes	Storage/Node	Total	Needed		Margin
Server Storage	1,554	13,372	20,780,088	19,786,570		993,518

Table 9 Space Computations (Note: All numbers about space are in GB)

5 Clause 5: Performance Metrics and Response Time Related Items

5.1 Measured tpmC

Measured tpmC must be reported.

The measured tpmC was 707,351,007.

5.2 Response Times

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

Numerical Quantities Summary Section of the Executive Summary of this report gives the detail of the Response Times of all kinds of transactions and menu.

5.3 Keying and Think Times

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

Numerical Quantities Summary Section of the Executive Summary of this report gives the detail of the Keying and Think Times of all kinds of transaction types.

5.4 Response Time Frequency Distribution Curves

Response Time frequency distribution curves (see Clause 5.6.1) must be reported for each transaction type.

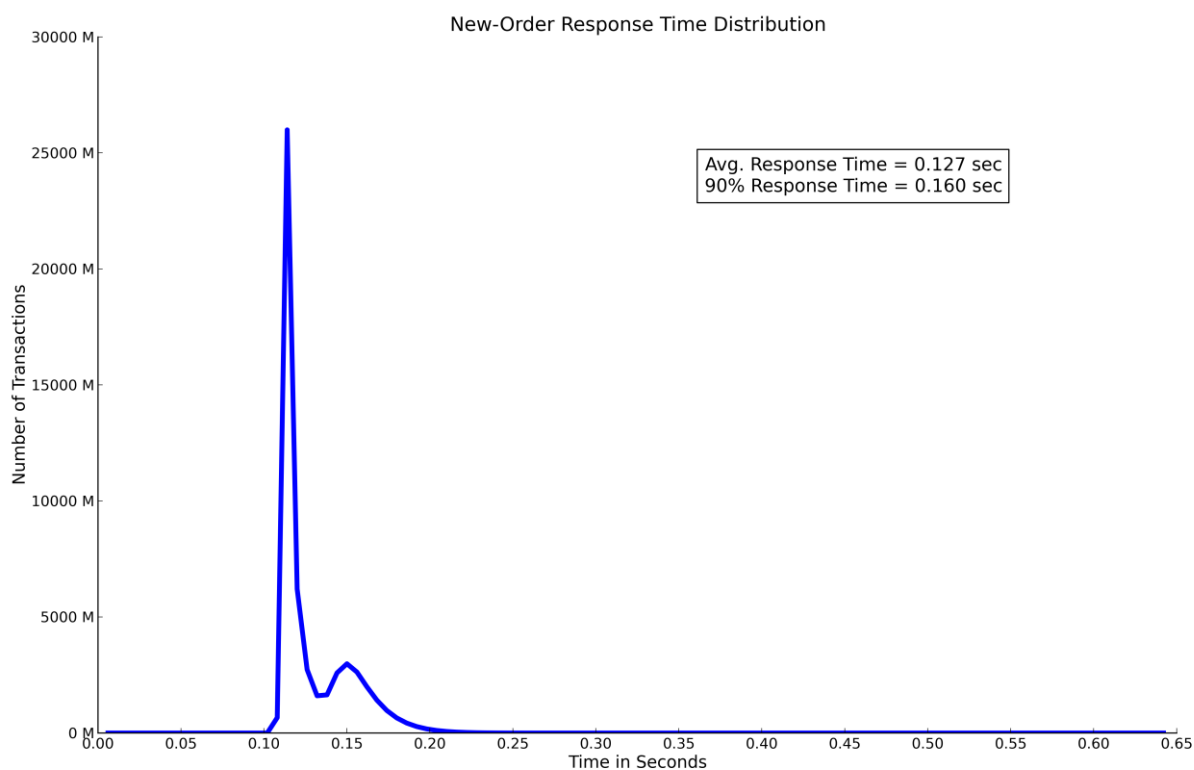


Figure 3 Frequency distribution of response times for New-Order

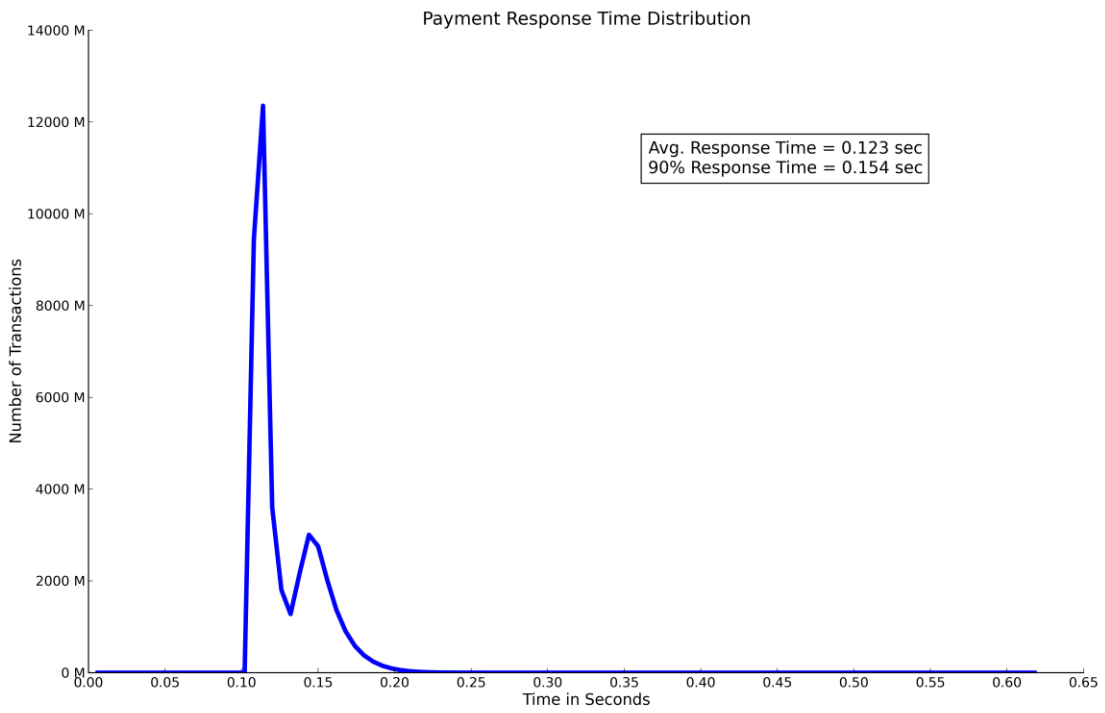


Figure 4 Frequency distribution of response times for Payment

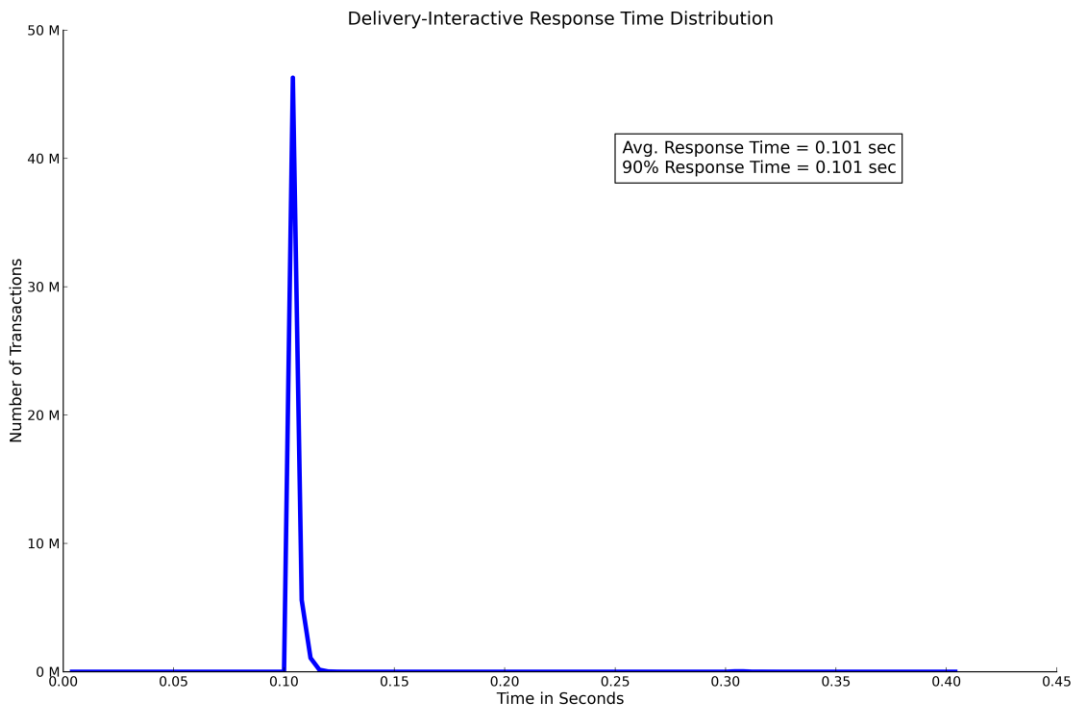


Figure 5 Frequency distribution of response times for Delivery (interactive)

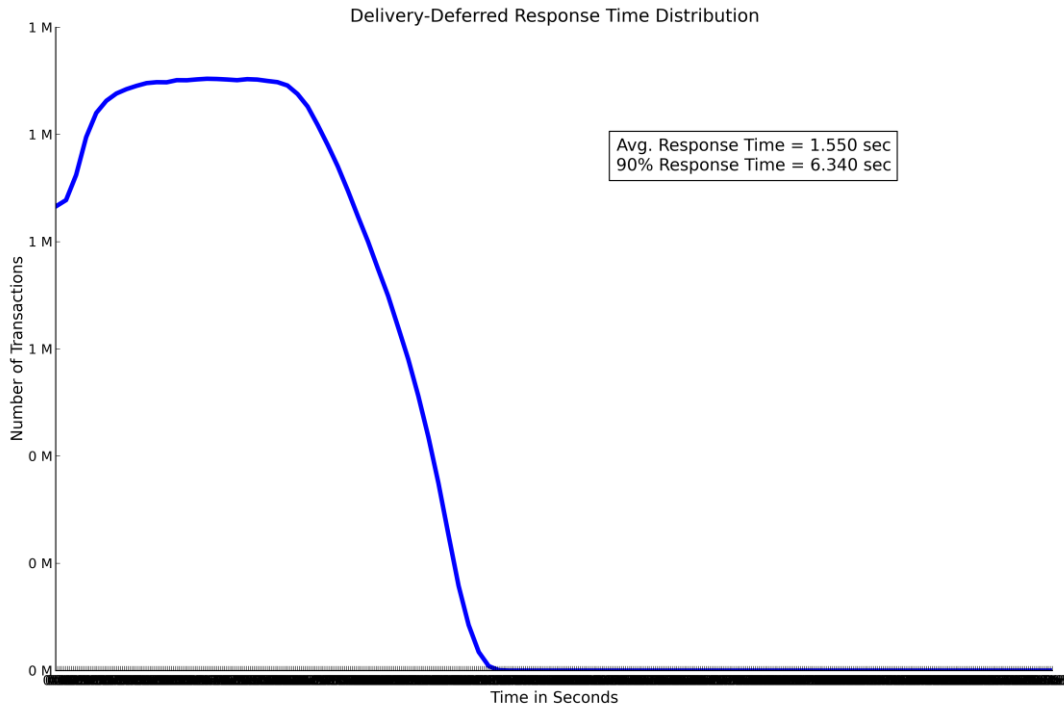


Figure 6 Frequency distribution of response times for Delivery (deferred)

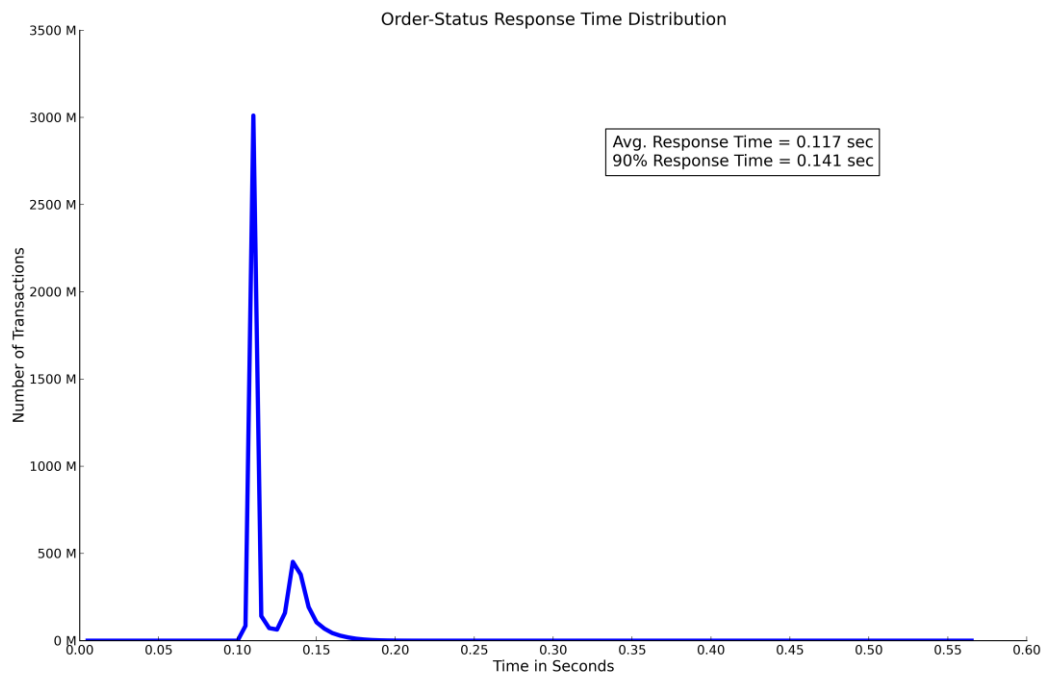


Figure 7 Frequency distribution of response times for Order-Status

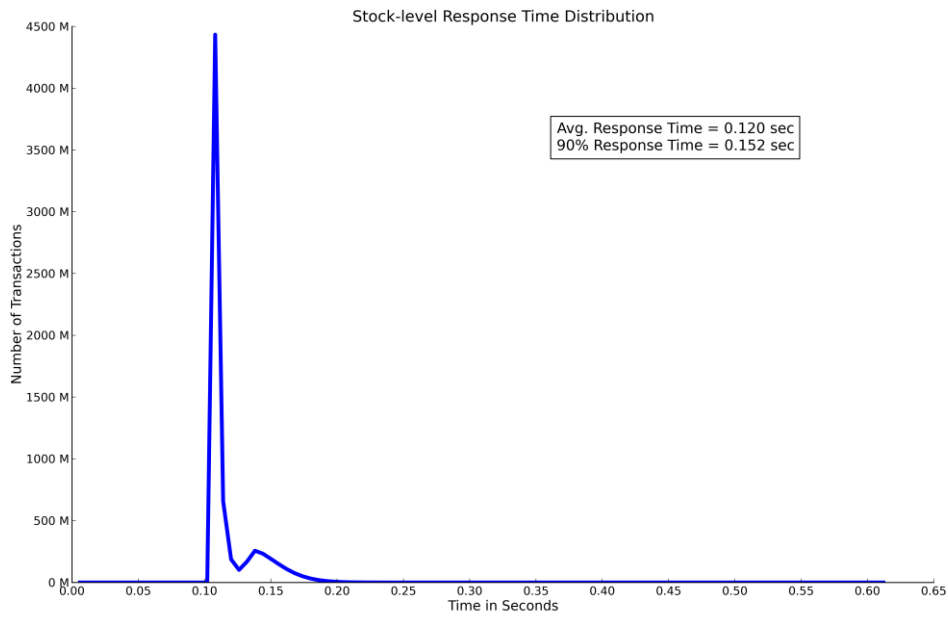


Figure 8 Frequency distribution of response times for Stock-Level

5.5 Think Time Frequency Distribution

Think Time frequency distribution curves (see Clause 5.6.3) must be reported for the New-Order transaction.

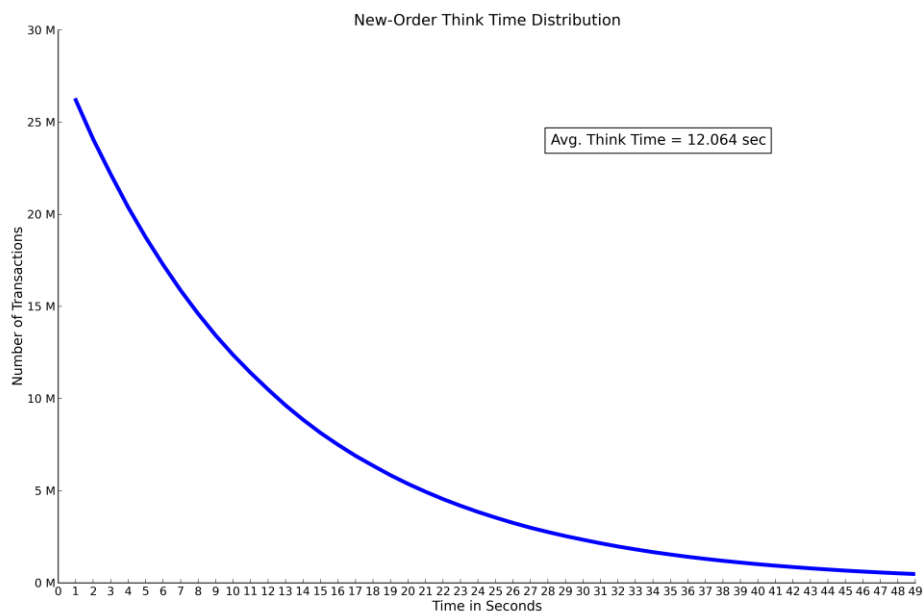


Figure 9 Frequency distribution of think times for New-Order

5.6 Response Times versus Throughput

The performance curve for response times versus throughput (see Clause 5.6.2) must be reported for the New-Order transaction.

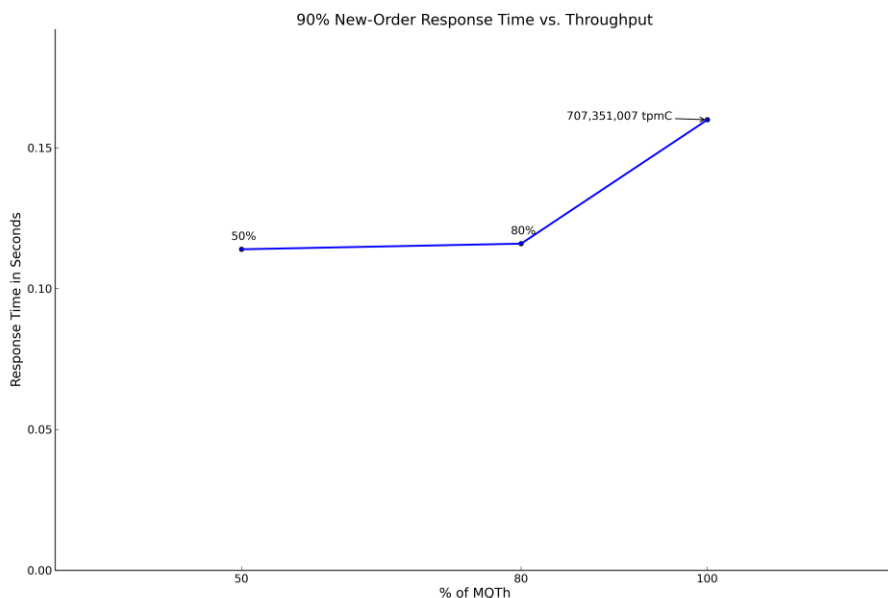


Figure 10 New-Order Response Time versus Throughput

5.7 Throughput versus Elapsed Time

A graph of throughput versus elapsed time (see Clause 5.6.4) must be reported for the New-Order transaction.

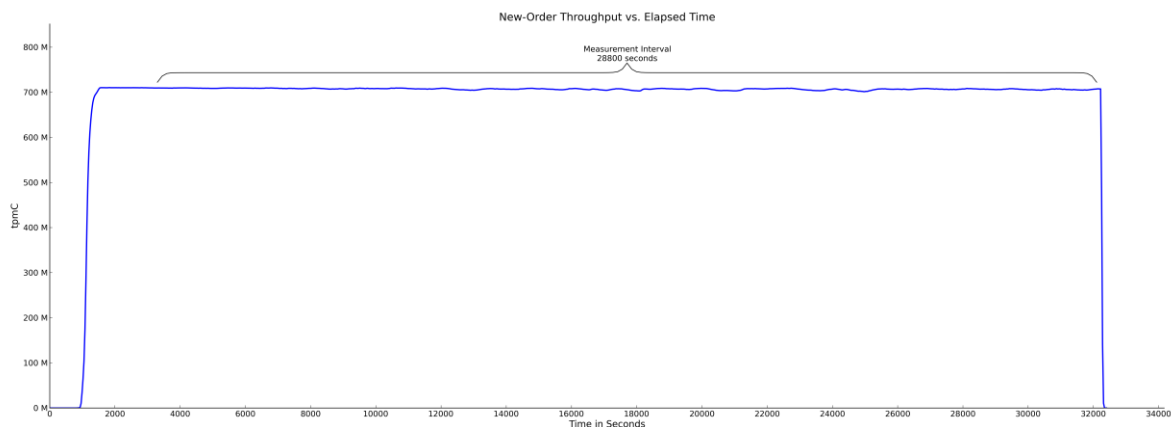


Figure 11 New-Order throughput versus Time

5.8 Steady State Determination

The method used to determine that the SUT had reached a steady state prior to commencing the measurement interval (see Clause 5.5) must be described.

The New-Order throughput versus Time graph was reported at 30-second intervals for the duration of the benchmark. During the ramp-up period, terminals logon on and then began to submit requests and the graph reflected the throughput increment. After ramp-up period, the graph was in steady state and the throughput kept steady for whole measurement interval.

5.9 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.

During the steady state of the test, RTEs generated TPC-C transactions according to the requirements of the specification. Emulated users submitted input to web server through the HTTP connection built during ramp-up period and waited for the response synchronously. The response time was recorded from the start of the transaction to the last byte received. First the emulated user selected a transaction type and got the input page. RTE recorded menu time after the last byte received from web server. The emulated user waited for a period of time to simulate inputting data (Keying Time). After that, the emulated user submitted TPC-C transactions input to web server and got back the response. RTE recorded transaction response time after the last byte received. The emulated user waited for a period of time to simulate thinking (Think Time). The cycle continued for the next transaction until the Test Run completes.

OceanBase Cluster maintained ACID properties of all transactions submitted to OceanBase according to the specification. Committed transactions are “logged” and the logs are written to disk on a majority of the corresponding replicas. These logs ensure the system never loses any committed transactions even in the case of the failure of any single ECS instance. To ensure modified data are not left in memory for more than 30 minutes, OceanBase Cluster implements checkpoint to flush modified records to checkpoint files on disk periodically (~21 minutes during the test).

5.10 Measurement Interval

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

The reported measured interval was 28,800 seconds (8 hours) long.

5.11 Transaction Mix Regulation

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.

Weighted random distribution algorithm was used by the RTE to regulate the transaction mix percentage. Weights were assigned before the measurement started and were not adjusted during the test.

5.12 Transaction Mix

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

Numeric Quantities for Transactions and Terminal Table on page 21 shows the detail.

5.13 Percentage of New-Order Transactions

The percentage of New-Order transactions rolled back as a result of invalid item number must be disclosed.

Numeric Quantities for Transactions and Terminal Table on page 21 shows the detail.

5.14 Number of Order-lines per New-Order

The average number of order-lines entered per New-Order transaction must be disclosed.

Numeric Quantities for Transactions and Terminal table on page 21 shows the detail.

5.15 Percentage of Remote Order-lines per New-Order

The percentage of remote order-lines entered per New-Order transaction must be disclosed.

Numeric Quantities for Transactions and Terminal table on page 21 shows the detail.

5.16 Percentage of Remote Payments

The percentage of remote payment transactions must be disclosed.

Numeric Quantities for Transactions and Terminal table on page 21 shows the detail.

5.17 Percentage of access customer by C_LAST for Payment and Order-Status

The percentage of customer selections by customer last name in the Payment and Order-Status transactions must be disclosed.

Numeric Quantities for Transactions and Terminal table on page 21 shows the detail.

5.18 Percentage of Skipped Delivery Transactions

The percentage of Delivery transactions skipped due to there being fewer than necessary orders in the New-Order table must be disclosed.

Numeric Quantities for Transactions and Terminal table on page 21 shows the detail.

5.19 Checkpoints

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.

OceanBase checkpoints ensure that modified data is written to durable media.

OceanBase controls checkpoints using one or combination of the following mechanisms:

- 1) The amount of memory used to store transaction mutations (increments) reaches the pre-configured threshold, parameter `freeze_trigger_percentage` can be used.
- 2) An explicit command

In TPC-C test, by setting value for system parameter `freeze_trigger_percentage` to 37, interval between two checkpoints was about 21 minutes. The start and end time and other verbose information of each checkpoint could be determined by querying system view `gv$minor_merge_info`.

In OceanBase, each database data node triggers checkpoint according to `freeze_trigger_percentage` alone. Since transaction speed is not identical on different database data nodes, checkpoint interval on different database data nodes is slightly different. The longest checkpoint interval is 22 minutes 27 seconds and the longest checkpoint duration is 13 minutes 20 seconds.

6 Clause 6: SUT, Driver and Communications Related Items

6.1 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 used was developed by Ant Financial (Hang Zhou) Network Technology Co., Ltd. It consists of a master program, which creates threads to do the work. Generator threads are used to generate transactions according to the TPC-C standard specification. Dispatch threads are used to dispatch tasks, which meet the requirement of KEYING time and THINK time etc. Processor threads are used to execute tasks, communicate with the clients, send requests and get results. Dumper threads are used to collect run-time statistics and record them in a metadata database and result files. The master program collects the statistics after a complete run and records them in the metadata database too.

Inputs to the RTE include the database scale, the ramp-up time, measurement interval and ramp-down time. The main information related to the RTE is as follows:

Input Type	Value
Ramp-up Duration	3,300 seconds
Ramp-down Duration	150 seconds
Measurement Interval	28,800 seconds
Database Scale	55,944,000 warehouses
Total terminals	559,440,000
Terminals/Driver	55,944
Number of RTEs nodes/instances	10,000

Table 10 RTE parameters input

6.2 Number of Terminal Connections Lost

The number of terminal connections lost during the Measurement Interval must be disclosed (see Clause 6.6.2).

The connections were established when the RTE was started. RTE built one http connection to Nginx Web Server for each terminal and the number of errors related to connection that occurred during the benchmark run was counted and logged. The number of connection errors was recorded to metadata database when the run completed. There were no errors logged and the number of errors recorded was zero during the Measurement Interval.

6.3 Emulated Components

It must be demonstrated that the functionality and performance of the components being emulated in the Driver System are equivalent to that of the priced system. The results of the test described in Clause 6.6.3.4 must be disclosed.

Workstations connected to clients via HTTP in the same way as the emulated system. The driver system emulated the workstations by establishing an http connection to the SUT for each terminal.

6.4 Configuration Diagrams

A complete functional diagram of both the benchmark configuration and the configuration of the proposed (target) system must be disclosed. A detailed list of all software and hardware functionality being performed on the Driver System, and its interface to the SUT must be disclosed (see Clause 6.6.3.6).

The Measured Configuration on page 16 shows the details of configuration used during the measurement. The Priced Configuration on page 17 shows the details of the priced components. Sections 6.1 and 6.3 above describe the emulated components of the Driver System.

6.5 Network Configuration

The network configurations of both the tested services and the proposed (target) services, which are being represented, and a thorough explanation of exactly which parts of the proposed configuration are being replaced with the Driver System must be disclosed (see Clause 6.6.4).

There are three separate networks to administer, maintain, setup, and execute the TPC-C transactions in this test. The first network is within a Virtual Private Cloud (VPC) that provides an isolated cloud network to operate resources in a secure environment, connected by 1 VSwitch. It is the backbone network for communicating between the ecs.i2d.21xlarge OceanBase Cluster data nodes, ecs.ebmg6.26xlarge OceanBase Cloud Platform (OCP) nodes and connecting the ecs.ebmg6.26xlarge client nodes. The network between the ecs.ebmg6.26xlarge clients and the RTE is also through the VSwitch within the same VPC.

The other separate network is Internet IP address, the public network. Each ECS instance will have an Internet IP after purchase. Users can logon ECS instances by the public network to configure and use the ECS instances, including all nodes of OceanBase Cluster, OceanBase Cloud Platform, Clients and RTE.

The third network provides administration and monitoring for all of the above ECS instances and network over Alibaba Cloud Console, users can access the console by a web browser.

6.6 Operator Intervention

If the configuration requires operator intervention, the mechanism and the frequency of this intervention must be disclosed.

The Alibaba Cloud Elastic Compute Service and OceanBase v2.2 Enterprise Edition with Partitioning, Horizontal Scalability and Advanced Compression configuration do not require any operator intervention to sustain the reported throughput for the 8-hour business day.

7 Clause 7: Pricing Related Items

7.1 Hardware and Software Price

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

A detailed list of hardware and software used in the priced system is listed as part of the Executive Summary included with this report. Third-party price is available in Appendix D.

7.2 Total 3-Year Cost

The total 3-year price of the entire **Priced Configuration** must be reported, including: hardware, software, and maintenance charges. The justification of any **Discounts** applied must be disclosed in the price sheet. Sufficient detail of what items are being discounted and by how much they are being discounted must be provided so that the **Discount** amount used in the computation of the total system cost can be independently reproduced.

The total 3-year price of the entire Priced Configuration for all components used in this measurement is included in the Executive Summary at the beginning of this document. The discounts are based upon list prices in the People's Republic of China and for similar quantities and configurations.

7.3 Availability Date

The Committed delivery date for general availability (availability date) of products used in the price calculations must be reported. The Availability Date must be reported on the first page of the Executive Summary and with a precision of one day. 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 **Generally Available**. Each **Component** used in the **Priced Configuration** is considered to be Available on the Availability Date unless an earlier date is specified.

All products will be available on June 08, 2020.

7.4 Hardware and Software Support

Ant Financial (Hang Zhou) Network Technology Co., Ltd. provides Support service for OceanBase v2.2 Enterprise Edition with Partitioning, Horizontal Scalability and Advanced Compression with 7x24 hours service, work order and exclusive business support.

Alibaba Cloud provides Enterprise Support Service consists of services in support of all ECS instances with 7x24 hours phone (95187) service, work order and exclusive business support. And Alibaba Cloud Enterprise Support Service support for Aliyun Linux 2.

OpenResty Inc. provides Nginx support service with 7x24 hours phone service.

The response time of all support services listed above is within 4 hours.

7.5 Statement of measured tpmC and Price/Performance

A statement of the measured tpmC, as well as the respective calculations for 3-year pricing, price/performance (price/tpmC), and the availability date must be included.

System	tpmC	3-year System Cost	CNY/tpmC	Availability Date
Alibaba Cloud Elastic Compute Service Cluster (with 1554 OceanBase Data Nodes)	707,351,007	CNY 2,814,509,552.00	3.98	June 08, 2020

Table 11 Statement of tpmC and price/performance

7.6 Country Specific Pricing

Additional Clause 7 related items may be included in the Full Disclosure Report for each country specific priced configuration. Country specific pricing is subject to Clause 7.1.7

The components for this configuration are priced using currency from the People's Republic of China. All prices listed in the report are based upon Chinese list prices.

7.7 Orderability Date

For each of the components that are not orderable on the report date of the FDR, the following information must be included in the FDR:

- *Name and part number of the item that is not orderable*
- *The date when the component can be ordered (on or before the Availability Date)*
- *The method to be used to order the component (at or below the quoted price) when that date arrives*
- *The method for verifying the price*

All components of the SUT will be available on June 8, 2020.

8 Clause 8: Auditor Attestation

8.1 Auditor Information

The auditor's name, address, phone number, and a copy of the auditor's attestation letter indicating compliance must be included in the Full Disclosure Report

This benchmark was audited by:

PerfLabs, Inc. DBA InfoSizing

Doug Johnson

63 Lourdes Dr.

Leominster, MA 01453

USA

www.sizing.com

8.2 Attestation Letter

The Auditor's Attestation Letter is included in the following pages.

Zou Yinchao
Senior Engineer
Room 706, Building 5, Xixi New Block,
Xihu District- Hangzhou- Zhejiang

May 13, 2020

I verified the TPC Benchmark™ C v5.11.0 performance of the following configuration:

Platform: Alibaba Cloud Elastic Compute Service Cluster
(with 1,554 OceanBase Data Nodes)
Operating System: Aliyun Linux 2
Database Manager: OceanBase v2.2 Enterprise Edition with Partitioning, Horizontal Scalability and Advanced Compression

The results were:

Performance Metric **707,351,007 tpmC**
Number of Users 559,440,000

Server	Alibaba Cloud Elastic Compute Service Cluster		
Nodes	1,554 Data, 3 Root, 3 OCP		
CPU	84x vCPU Data, Root nodes (based on Intel® Xeon® Platinum 8163, 2.5GHz, 33MB) 104 vCPU OCP nodes (based on Intel® Xeon® Platinum 8269CY, 2.5GHz, 35.75MB)		
Memory	712 GB (Data & Root nodes), 384 GB (OCP nodes)		
Storage	Qty	Size	Type
	1	40 GB	Ultra Cloud Disk (all nodes)
	1	200 GB	Ultra Cloud Disk (all nodes)
	4	3,570 GB	NVMe SSD Local Disk (Data, Root nodes)

In my opinion, these performance results were produced in compliance with the TPC requirements for the benchmark.

The following verification items were given special attention:

- The transactions were correctly implemented
- The database records were the proper size
- The database was properly scaled and populated
- The ACID properties were met

- Input data was generated according to the specified percentages
- The transaction cycle times included the required keying and think times
- The reported response times were correctly measured
- At least 90% of all delivery transactions met the 80 Second completion time limit
- All 90% response times were under the specified maximums
- The measurement interval was representative of steady state conditions
- The reported measurement interval was over 120 minutes
- Checkpoint intervals were under 30 minutes
- The 60-day storage requirement was correctly computed
- The system pricing was verified for major components and maintenance

Additional Audit Notes:

None.

Respectfully Yours,

A handwritten signature in black ink that reads "Doug Johnson". The signature is written in a cursive style with a long horizontal flourish extending to the right.

Doug Johnson, Certified TPC Auditor

Appendix A: Source Code Files List

The source code and scripts used to implement the benchmark test is provided as supporting files. Here is a list for source code and scripts files:

/src-200410

/load

```
-- breakpoint.c
-- load.c
-- support.c
-- breakpoint.h
-- support.h
```

/nginx

```
-- ngx_http_oceanbase_handler.c
-- ngx_http_oceanbase_keepalive.c
-- ngx_http_oceanbase_module.c
-- ngx_http_oceanbase_output.c
-- ngx_http_oceanbase_processor.c
-- ngx_http_oceanbase_upstream.c
-- ngx_http_oceanbase_util.c
-- ngx_tpcc_delivery_pool.c
-- ngx_tpcc_delivery_thread.c
-- ngx_tpcc_txn.c
-- ngx_tpcc_utils.c
-- ngx_http_oceanbase_handler.h
-- ngx_http_oceanbase_keepalive.h
-- ngx_http_oceanbase_module.h
-- ngx_http_oceanbase_output.h
-- ngx_http_oceanbase_processor.h
-- ngx_http_oceanbase_upstream.h
-- ngx_http_oceanbase_util.h
-- ngx_tpcc_common.h
-- ngx_tpcc_delivery_pool.h
-- ngx_tpcc_delivery_thread.h
-- ngx_tpcc_txn.h
```

/tools

```
-- merge_result.c
-- sort_result.c
-- common.h
```

/aci

/acid-src-200410

```
-- ob_tpcc_acid_a.c
-- ob_tpcc_acid_c.c
-- ob_tpcc_acid_del.c
-- ob_tpcc_acid_i1.c
```

```
-- ob_tpcc_acid_i2.c
-- ob_tpcc_acid_i3.c
-- ob_tpcc_acid_i4.c
-- ob_tpcc_acid_i5.c
-- ob_tpcc_acid_i6.c
-- ob_tpcc_acid_i7.c
-- ob_tpcc_acid_i8.c
-- ob_tpcc_acid_i9.c
-- ob_tpcc_acid_no.c
-- ob_tpcc_acid_os.c
-- ob_tpcc_acid_pay.c
-- ob_tpcc_acid_utils.c
-- ob_tpcc_acid.h
```

Appendix B: Database Design

create-tenant.sql

```
create resource unit unit_for_ob6 max_cpu 70, max_memory '580G', max_disk_size '3982G',
max_session_num 65535, MIN_CPU 70, MIN_MEMORY='580G';
create resource pool pool_for_ob6 unit = 'unit_for_ob6', unit_num = 518, zone_list =
('zone1','zone2','zone3');
create resource unit sys_unit_for_ob6 max_cpu 15, max_memory '50G', max_disk_size '3982G',
max_session_num 65535, MIN_CPU 15, MIN_MEMORY='50G';
create resource pool sys_pool_for_ob6 unit = 'sys_unit_for_ob6', unit_num = 1, zone_list =
('zone1','zone2','zone3'), is_tenant_sys_pool=true;
create tenant tpcc replica_num = 3, locality='F,F{1,memstore_percent:0},L@[zone1, zone2, zone3]',
resource_pool_list=('pool_for_ob6', 'sys_pool_for_ob6') set ob_tcp_invited_nodes='% ' ,
ob_compatibility_mode='oracle';
```

create-user.sql

```
create user TPCC identified by "123456"; grant all on *.* to TPCC WITH GRANT OPTION; set global
autocommit=OFF;
```

create-procedure.sql

```
set autocommit=1;
set ob_query_timeout=4000000000;
set ob_trx_timeout=4000000000;
```

```
CREATE OR REPLACE TYPE intarray IS TABLE OF INTEGER;
CREATE OR REPLACE TYPE numarray IS TABLE OF NUMBER;
CREATE OR REPLACE TYPE distarray IS TABLE OF VARCHAR(24);
CREATE OR REPLACE TYPE chararray IS TABLE OF VARCHAR(1);
CREATE OR REPLACE TYPE datarray IS TABLE OF DATE;
```

```
CREATE OR REPLACE VIEW stock_item AS
SELECT /*+ leading(s) use_merge(i) */
    i_price,
    i_name,
    i_data,
    s_i_id,
    s_w_id,
    s_order_cnt,
    s_ytd,
    s_remote_cnt,
    s_quantity,
    s_data,
    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 s.s_i_id = i.i_id;
```

delimiter //

```
CREATE OR REPLACE PROCEDURE neworder (
    par_w_id INTEGER,
    par_d_id INTEGER,
    par_c_id INTEGER,
    par_o_all_local INTEGER,
    par_o_ol_cnt IN OUT BINARY_INTEGER,
    par_w_tax OUT NUMBER,
    par_d_tax OUT NUMBER,
    par_o_id OUT INTEGER,
    par_c_discount OUT NUMBER,
    par_c_credit OUT varchar2,
    par_c_last OUT varchar2,
    par_retry IN OUT BINARY_INTEGER,
    par_cr_date DATE,
    par_ol_i_id intarray,
    par_ol_supply_w_id intarray,
    par_i_price OUT numarray,
    par_i_name OUT distarray,
    par_s_quantity OUT intarray,
    par_brand_generic OUT chararray,
    par_ol_amount OUT numarray,
    par_s_remote intarray,
    par_ol_quantity intarray
)
IS
    TYPE dist_array IS TABLE OF VARCHAR(24);
    TYPE int_array IS TABLE OF BINARY_INTEGER;
    s_dist dist_array := dist_array();
    idx1arr intarray := intarray();
    nulldate DATE;
    dummy_local BINARY_INTEGER;
    max_index BINARY_INTEGER;
    not_serializable EXCEPTION;
    PRAGMA EXCEPTION_INIT(not_serializable,-6235);
BEGIN
    LOOP
        BEGIN
```

```

UPDATE dist SET d_next_o_id = d_next_o_id + 1
WHERE d_w_id = par_w_id AND d_id = par_d_id
RETURNING d_next_o_id-1, d_tax
INTO par_o_id, par_d_tax;

SELECT c_discount, c_credit, c_last
INTO par_c_discount, par_c_credit, par_c_last
FROM cust
WHERE c_w_id = par_w_id AND c_d_id = par_d_id AND c_id = par_c_id;

SELECT w_tax
INTO par_w_tax
FROM ware
WHERE w_id = par_w_id;

INSERT INTO nord
VALUES (par_w_id, par_d_id, par_o_id);

INSERT INTO ord
VALUES (par_w_id, par_d_id, par_o_id, par_c_id, null,
par_o_ol_cnt, par_o_all_local, par_cr_date);

CASE par_d_id
WHEN 1 THEN
BEGIN
FORALL idx IN 1 .. par_o_ol_cnt
UPDATE stock_item
SET s_order_cnt = s_order_cnt + 1,
s_ytd = s_ytd + par_ol_quantity(idx),
s_remote_cnt = s_remote_cnt + par_s_remote(idx),
s_quantity = (CASE WHEN s_quantity < par_ol_quantity (idx) + 10
THEN s_quantity + 91
ELSE s_quantity
END) - par_ol_quantity(idx)
WHERE s_i_id = par_ol_i_id(idx)
AND s_w_id = par_ol_supply_w_id(idx)
RETURNING i_price, i_name, s_quantity, s_dist_01,
i_price*par_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 par_i_price, par_i_name, par_s_quantity, s_dist,
par_ol_amount, par_brand_generic;
END;
WHEN 2 THEN
BEGIN
FORALL idx IN 1 .. par_o_ol_cnt
UPDATE stock_item

```

```

SET s_order_cnt = s_order_cnt + 1,
s_ytd = s_ytd + par_ol_quantity(idx),
s_remote_cnt = s_remote_cnt + par_s_remote(idx),
s_quantity = (CASE WHEN s_quantity < par_ol_quantity (idx) + 10
THEN s_quantity + 91
ELSE s_quantity
END) - par_ol_quantity(idx)
WHERE s_i_id = par_ol_i_id(idx)
AND s_w_id = par_ol_supply_w_id(idx)
RETURNING i_price, i_name, s_quantity, s_dist_02,
i_price*par_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 par_i_price, par_i_name, par_s_quantity, s_dist,
par_ol_amount, par_brand_generic;
END;
WHEN 3 THEN
BEGIN
FORALL idx IN 1 .. par_o_ol_cnt
UPDATE stock_item
SET s_order_cnt = s_order_cnt + 1,
s_ytd = s_ytd + par_ol_quantity(idx),
s_remote_cnt = s_remote_cnt + par_s_remote(idx),
s_quantity = (CASE WHEN s_quantity < par_ol_quantity (idx) + 10
THEN s_quantity + 91
ELSE s_quantity
END) - par_ol_quantity(idx)
WHERE s_i_id = par_ol_i_id(idx)
AND s_w_id = par_ol_supply_w_id(idx)
RETURNING i_price, i_name, s_quantity, s_dist_03,
i_price*par_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 par_i_price, par_i_name, par_s_quantity, s_dist,
par_ol_amount, par_brand_generic;
END;
WHEN 4 THEN
BEGIN
FORALL idx IN 1 .. par_o_ol_cnt
UPDATE stock_item
SET s_order_cnt = s_order_cnt + 1,
s_ytd = s_ytd + par_ol_quantity(idx),
s_remote_cnt = s_remote_cnt + par_s_remote(idx),
s_quantity = (CASE WHEN s_quantity < par_ol_quantity (idx) + 10

```

```

                THEN s_quantity +91
                ELSE s_quantity
            END) - par_ol_quantity(idx)
WHERE s_i_id = par_ol_i_id(idx)
    AND s_w_id = par_ol_supply_w_id(idx)
RETURNING i_price, i_name, s_quantity, s_dist_04,
    i_price*par_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 par_i_price, par_i_name, par_s_quantity, s_dist,
    par_ol_amount, par_brand_generic;
END;
WHEN 5 THEN
BEGIN
    FORALL idx IN 1 .. par_o_ol_cnt
        UPDATE stock_item
            SET s_order_cnt = s_order_cnt + 1,
                s_ytd = s_ytd + par_ol_quantity(idx),
                s_remote_cnt = s_remote_cnt + par_s_remote(idx),
                s_quantity = (CASE WHEN s_quantity < par_ol_quantity (idx) + 10
                    THEN s_quantity +91
                    ELSE s_quantity
                END) - par_ol_quantity(idx)
        WHERE s_i_id = par_ol_i_id(idx)
            AND s_w_id = par_ol_supply_w_id(idx)
        RETURNING i_price, i_name, s_quantity, s_dist_05,
            i_price*par_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 par_i_price, par_i_name, par_s_quantity, s_dist,
            par_ol_amount, par_brand_generic;
    END;
WHEN 6 THEN
BEGIN
    FORALL idx IN 1 .. par_o_ol_cnt
        UPDATE stock_item
            SET s_order_cnt = s_order_cnt + 1,
                s_ytd = s_ytd + par_ol_quantity(idx),
                s_remote_cnt = s_remote_cnt + par_s_remote(idx),
                s_quantity = (CASE WHEN s_quantity < par_ol_quantity (idx) + 10
                    THEN s_quantity +91
                    ELSE s_quantity
                END) - par_ol_quantity(idx)
        WHERE s_i_id = par_ol_i_id(idx)

```

```

            AND s_w_id = par_ol_supply_w_id(idx)
            RETURNING i_price, i_name, s_quantity, s_dist_06,
                i_price*par_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 par_i_price, par_i_name, par_s_quantity, s_dist,
            par_ol_amount, par_brand_generic;
    END;
WHEN 7 THEN
BEGIN
    FORALL idx IN 1 .. par_o_ol_cnt
        UPDATE stock_item
            SET s_order_cnt = s_order_cnt + 1,
                s_ytd = s_ytd + par_ol_quantity(idx),
                s_remote_cnt = s_remote_cnt + par_s_remote(idx),
                s_quantity = (CASE WHEN s_quantity < par_ol_quantity (idx) + 10
                    THEN s_quantity +91
                    ELSE s_quantity
                END) - par_ol_quantity(idx)
        WHERE s_i_id = par_ol_i_id(idx)
            AND s_w_id = par_ol_supply_w_id(idx)
        RETURNING i_price, i_name, s_quantity, s_dist_07,
            i_price*par_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 par_i_price, par_i_name, par_s_quantity, s_dist,
            par_ol_amount, par_brand_generic;
    END;
WHEN 8 THEN
BEGIN
    FORALL idx IN 1 .. par_o_ol_cnt
        UPDATE stock_item
            SET s_order_cnt = s_order_cnt + 1,
                s_ytd = s_ytd + par_ol_quantity(idx),
                s_remote_cnt = s_remote_cnt + par_s_remote(idx),
                s_quantity = (CASE WHEN s_quantity < par_ol_quantity (idx) + 10
                    THEN s_quantity +91
                    ELSE s_quantity
                END) - par_ol_quantity(idx)
        WHERE s_i_id = par_ol_i_id(idx)
            AND s_w_id = par_ol_supply_w_id(idx)
        RETURNING i_price, i_name, s_quantity, s_dist_08,
            i_price*par_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 par_i_price, par_i_name, par_s_quantity, s_dist,
        par_ol_amount, par_brand_generic;
END;
WHEN 9 THEN
BEGIN
    FORALL idx IN 1 .. par_o_ol_cnt
        UPDATE stock_item
            SET s_order_cnt = s_order_cnt + 1,
                s_ytd = s_ytd + par_ol_quantity(idx),
                s_remote_cnt = s_remote_cnt + par_s_remote(idx),
                s_quantity = (CASE WHEN s_quantity < par_ol_quantity (idx) + 10
                    THEN s_quantity + 91
                    ELSE s_quantity
                    END) - par_ol_quantity(idx)
        WHERE s_i_id = par_ol_i_id(idx)
            AND s_w_id = par_ol_supply_w_id(idx)
        RETURNING i_price, i_name, s_quantity, s_dist_09,
            i_price*par_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 par_i_price, par_i_name, par_s_quantity, s_dist,
        par_ol_amount, par_brand_generic;
END;
WHEN 10 THEN
BEGIN
    FORALL idx IN 1 .. par_o_ol_cnt
        UPDATE stock_item
            SET s_order_cnt = s_order_cnt + 1,
                s_ytd = s_ytd + par_ol_quantity(idx),
                s_remote_cnt = s_remote_cnt + par_s_remote(idx),
                s_quantity = (CASE WHEN s_quantity < par_ol_quantity (idx) + 10
                    THEN s_quantity + 91
                    ELSE s_quantity
                    END) - par_ol_quantity(idx)
        WHERE s_i_id = par_ol_i_id(idx)
            AND s_w_id = par_ol_supply_w_id(idx)
        RETURNING i_price, i_name, s_quantity, s_dist_10,
            i_price*par_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 par_i_price, par_i_name, par_s_quantity, s_dist,
            par_ol_amount, par_brand_generic;
    END;
    ELSE
        NULL; /* should never be here */
    END CASE;

    /* cache the no of rows processed */
    dummy_local := sql%rowcount;

    /* set rows with invalid item to NULL, par_i_price, par_i_name, par_ol_amount have been setted
before */
    IF (dummy_local != par_o_ol_cnt) THEN
        max_index := par_o_ol_cnt - dummy_local;
        par_i_price.EXTEND(max_index);
        par_i_name.EXTEND(max_index);
        par_ol_amount.EXTEND(max_index);
        par_s_quantity.EXTEND(max_index);
        s_dist.EXTEND(max_index);
        par_brand_generic.EXTEND(max_index);

        max_index := sql%rowcount;

        WHILE (max_index != par_o_ol_cnt) LOOP
            max_index := max_index + 1;
            par_i_price(max_index) := 0;
            par_i_name(max_index) := 'NO ITEM';
            par_ol_amount(max_index) := 0;
            par_s_quantity(max_index) := 0;
            s_dist(max_index) := NULL;
            par_brand_generic(max_index) := '';
        END LOOP;
    END IF;

    IF (idx1arr.count = 0) THEN
        idx1arr.EXTEND(par_o_ol_cnt);
        FOR var_x IN 1..par_o_ol_cnt LOOP
            idx1arr(var_x) := var_x;
        END LOOP;
    END IF;

    /* insert all items including invalid */
    FORALL idx IN 1..par_o_ol_cnt
        INSERT INTO ordl
            VALUES (par_w_id, par_d_id, par_o_id, idx1arr(idx),
                nulldate, par_ol_amount(idx), par_ol_i_id(idx), par_ol_supply_w_id(idx),
                par_ol_quantity(idx), s_dist(idx));

    /* If there are no errors, then COMMIT, else ROLLBACK */
    IF (dummy_local != par_o_ol_cnt) THEN

```



```

        par_o_ol_cnt := dummy_local;
        ROLLBACK;
    ELSE
        COMMIT;
    END IF;

    /* No exceptions, exit*/
    EXIT;

    EXCEPTION
    WHEN not_serializable THEN
        BEGIN
            ROLLBACK;
            par_retry := par_retry + 1;
        END;
    END;
END LOOP;

END neworder ;
//

CREATE OR REPLACE PROCEDURE orderstatus (
    ware_id INTEGER,
    dist_id INTEGER,
    cust_id IN OUT INTEGER,
    bylastname BINARY_INTEGER,
    cust_last IN OUT VARCHAR2,
    cust_first OUT VARCHAR2,
    cust_middle OUT VARCHAR2,
    cust_balance OUT NUMBER,
    ord_id IN OUT INTEGER,
    ord_entry_d OUT VARCHAR2,
    ord_carrier_id OUT INTEGER,
    ord_ol_cnt OUT INTEGER,
    oline_supply_w_id OUT intarray,
    oline_i_id OUT intarray,
    oline_quantity OUT intarray,
    oline_amount OUT numarray,
    oline_delivery_d OUT datarray
)
IS
    TYPE number_array IS TABLE OF number(38);
    rowArr number_array := number_array();

    read_nothing EXCEPTION;
    PRAGMA EXCEPTION_INIT(read_nothing,-4026);
    BEGIN
        IF bylastname != 0 THEN
            SELECT c_id BULK COLLECT INTO rowArr
            FROM cust

```

```

        WHERE c_last = cust_last AND c_d_id = dist_id AND c_w_id = ware_id
        ORDER BY c_first;
        IF 0 = rowArr.COUNT THEN
            RAISE read_nothing;
        END IF;
        cust_id := rowArr((rowArr.COUNT + 1) / 2);
    END IF;

    SELECT c_last, c_first, c_middle, c_balance
    INTO cust_last, cust_first, cust_middle, cust_balance
    FROM cust
        WHERE c_w_id = ware_id AND c_d_id = dist_id AND c_id = cust_id;

    /* Select the last ORDER for this customer. */
    SELECT o_id, o_entry_d_c, o_carrier_id_n, o_ol_cnt
    INTO ord_id, ord_entry_d, ord_carrier_id, ord_ol_cnt
    FROM (
        SELECT o_id, to_char(o_entry_d, 'DD-MM-YYYY.HH24:MI:SS') as o_entry_d_c,
        nvl(o_carrier_id,0) as o_carrier_id_n, o_ol_cnt
        FROM ordr
            WHERE o_w_id = ware_id AND o_d_id = dist_id AND o_c_id = cust_id
            ORDER BY o_w_id, o_d_id, o_c_id, o_id DESC
        ) tmp
        WHERE rownum <= 1;

    SELECT nvl(ol_delivery_d, DATE'1911-09-15') del_date, ol_amount, ol_i_id, ol_supply_w_id,
    ol_quantity
        BULK COLLECT INTO oline_delivery_d, oline_amount, oline_i_id, oline_supply_w_id,
    oline_quantity
    FROM ordl
        WHERE ol_w_id = ware_id AND ol_d_id = dist_id AND ol_o_id = ord_id;
    COMMIT;
END orderstatus;
//

CREATE OR REPLACE PROCEDURE delivery (
    ware_id IN INTEGER,
    dist_id OUT intarray,
    order_id OUT intarray,
    ordcnt OUT INTEGER,
    sums OUT numarray,
    del_date IN DATE,
    carrier_id IN INTEGER,
    order_c_id OUT intarray,
    retry IN OUT BINARY_INTEGER
)
IS
    TYPE int_array IS TABLE OF BINARY_INTEGER;
    var_dist int_array := int_array();

```

```

not_serializable EXCEPTION;
PRAGMA EXCEPTION_INIT(not_serializable,-6235);

read_nothing EXCEPTION;
PRAGMA EXCEPTION_INIT(read_nothing,-4026);
BEGIN

var_dist.EXTEND(10);
FOR var_x IN 1..10 LOOP
  var_dist(var_x) := var_x;
END LOOP;

LOOP
BEGIN
  ordcnt := 0;
  IF dist_id.count != 0 THEN
    dist_id.delete;
    order_id.delete;
  END IF;

  FORALL IDX IN 1..10
    DELETE FROM nord
      WHERE no_w_id = ware_id AND no_d_id = var_dist(IDX) AND no_o_id =
        (SELECT no_o_id FROM nord
          WHERE no_d_id = var_dist(IDX) AND no_w_id = ware_id AND rownum <= 1)
      RETURNING no_d_id, no_o_id BULK COLLECT INTO dist_id, order_id;

  ordcnt := SQL%ROWCOUNT;
  IF 0 = ordcnt THEN
    RAISE read_nothing;
  END IF;

  FORALL o in 1.. ordcnt
    UPDATE odr SET o_carrier_id = carrier_id WHERE o_w_id = ware_id
      AND o_d_id = dist_id(o)
      AND o_id = order_id(o)
      RETURNING o_c_id BULK COLLECT INTO order_c_id;

  FORALL o in 1.. ordcnt
    UPDATE ordl SET ol_delivery_d = del_date WHERE ol_w_id = ware_id
      AND ol_d_id = dist_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 = ware_id AND c_d_id = dist_id(c) AND c_id = order_c_id(c);

  COMMIT;

```

```

/* No exceptions, exit*/
EXIT;

EXCEPTION
  WHEN not_serializable THEN
    BEGIN
      ROLLBACK;
      retry := retry + 1;
    END;
  END;
END LOOP;

END delivery;
//

CREATE OR REPLACE PROCEDURE payment (
  ware_id INTEGER,
  dist_id INTEGER,
  cust_w_id INTEGER,
  cust_d_id INTEGER,
  cust_id IN OUT INTEGER,
  bylastname BINARY_INTEGER,
  hist_amount NUMBER,
  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 DATE,
  cust_credit OUT VARCHAR2,
  cust_credit_lim OUT NUMBER,
  cust_discount OUT NUMBER,
  cust_balance OUT NUMBER,
  cust_data OUT VARCHAR2,
  cr_date IN DATE,

```

```

        retry IN OUT BINARY_INTEGER
    )
IS

    dist_name VARCHAR2(10);
    ware_name VARCHAR2(10);

    TYPE number_array IS TABLE OF number(38);
    rowArr number_array := number_array();

    not_serializable EXCEPTION;
    PRAGMA EXCEPTION_INIT(not_serializable,-6235);

    read_nothing EXCEPTION;
    PRAGMA EXCEPTION_INIT(read_nothing,-4026);
BEGIN

    LOOP
    BEGIN
        UPDATE dist
        SET d_ytd = d_ytd + hist_amount
        WHERE d_w_id = ware_id AND d_id = dist_id
        RETURNING d_name, d_street_1, d_street_2, d_city, d_state, d_zip
        INTO dist_name, dist_street_1, dist_street_2, dist_city, dist_state, dist_zip;

    IF 0 = SQL%ROWCOUNT THEN
        RAISE read_nothing;
    END IF;

    UPDATE ware
    SET w_ytd = w_ytd + hist_amount
    WHERE w_id = ware_id
    RETURNING w_name, w_street_1, w_street_2, w_city, w_state, w_zip
    INTO ware_name, ware_street_1, ware_street_2, ware_city, ware_state, ware_zip;

    IF bylastname != 0 THEN
        SELECT c_id BULK COLLECT INTO rowArr
        FROM cust
        WHERE c_last = cust_last AND c_d_id = cust_d_id AND c_w_id = cust_w_id
        ORDER BY c_first;
        IF 0 = SQL%ROWCOUNT THEN
            RAISE read_nothing;
        END IF;
        cust_id := rowArr((rowArr.COUNT + 1) / 2);
    END IF;

    UPDATE cust
    SET c_balance = c_balance - hist_amount,
        c_ytd_payment = c_ytd_payment + hist_amount,
        c_payment_cnt = c_payment_cnt + 1

```

```

        WHERE c_w_id = cust_w_id AND c_d_id = cust_d_id AND c_id = cust_id
        RETURNING c_discount, c_credit, c_last, c_first, c_middle, c_balance,
            c_credit_lim, c_street_1, c_street_2, c_city, c_state, c_zip, c_phone, c_since
        INTO cust_discount, cust_credit, cust_last, cust_first, cust_middle, cust_balance,
            cust_credit_lim, cust_street_1, cust_street_2, cust_city,
            cust_state, cust_zip, cust_phone, cust_since;

    IF 0 = SQL%ROWCOUNT THEN
        RAISE read_nothing;
    END IF;

    /* Customer with bad credit, need to do the C_DATA work */
    IF cust_credit = 'BC' THEN
        UPDATE cust
        SET c_data = 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') || ' | ' )
            || c_data, 1, 500)
        WHERE c_w_id = cust_w_id AND c_d_id = cust_d_id AND c_id = cust_id
        RETURNING substr(c_data, 1, 200)
        INTO cust_data;
    ELSE
        cust_data := '';
    END IF;

    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, cr_date,
    hist_amount, ware_name || ' ' || dist_name);

    COMMIT;

    /* No exceptions, exit*/
    EXIT;

    EXCEPTION
    WHEN not_serializable THEN
    BEGIN
        ROLLBACK;
        retry := retry + 1;
    END;
    END;
    END LOOP;
END payment;

```

```

//
CREATE OR REPLACE PROCEDURE stocklevel (
  ware_id INTEGER,
  dist_id INTEGER,
  threshold INTEGER,
  low_stock OUT INTEGER
)
IS
BEGIN
  SELECT count (DISTINCT s_i_id)
  INTO low_stock
  FROM ordl, stok, dist
  WHERE ol_w_id = ware_id AND d_id = dist_id AND d_w_id = ware_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);

  COMMIT;
END stocklevel;
//
delimiter ;

```

RTE parameters

```

ramp_up = 3,300
mi = 28,800
ramp_down = 150
warehouse_num = 55,944,000
report_interval = 30

```

Appendix C: Configuration Options

Database Node

File system info

Filesystem	1K-blocks	Used	Available	Use%	Mounted on
devtmpfs	367266436	0	367266436	0%	/dev
tmpfs	367277408	0	367277408	0%	/dev/shm
tmpfs	367277408	864	367276544	1%	/run
tmpfs	367277408	0	367277408	0%	/sys/fs/cgroup
/dev/vdal	41147472	2690592	36553388	7%	/
/dev/mapper/ob_vg-ob_log	824646552	2210200	780476928	1%	/data/log1
/dev/mapper/ob_vg-ob_data	14022286784	5224054376	8091476620	40%	/data/1
/dev/vdf1	191196044	32580572	148833508	18%	/home
tmpfs	73455484	0	73455484	0%	/run/user/1000

PV	VG	Fmt	Attr	PSize	PFree							
/dev/vdb	ob_vg	lvm2	a--	3570.00g	0g							
/dev/vdc	ob_vg	lvm2	a--	3570.00g	0g							
/dev/vdd	ob_vg	lvm2	a--	3570.00g	0g							
/dev/vde	ob_vg	lvm2	a--	3570.00g	0g							
VG	#PV	#LV	#SN	Attr	VSize	VFree						
ob_vg	4	2	0	wz--n-	14279.98g	0g						
LV	VG	Attr	LSize	Pool	Origin	Data%	Meta%	Move	Log	Cpy%	Sync	Convert
ob_data	ob_vg	-wi-ao----	13479.98g									
ob_log	ob_vg	-wi-ao----	800.00g									

/etc/security/limits.conf

```
* soft nofile 1048576
* hard nofile 1048576
* soft nproc 655360
* hard nproc 655360
* soft core unlimited
* hard stack 20480
* soft stack 20480
root soft nofile 1048576
root hard nofile 1048576
root soft nproc 655360
root hard nproc 655360
root soft core unlimited
root hard stack 20480
root soft stack 20480
```

Observer.cfg

```
__easy_memory_limit=10G
```

```
_enable_svr_blacklist=False
_temporary_file_io_area_size=100G
migrate_concurrency=10
__schema_split_mode=True
__max_partition_cnt_per_server=500000
micro_block_merge_verify_level=0
major_compact_trigger=24
__ob_enable_major_minor_merge=False
__follower_replica_merge_level=2
__minor_compaction_interval=10m
minor_compact_trigger=32
minor_freeze_times=24
merge_stat_sampling_ratio=1
__mini_merge_concurrency=0
minor_merge_concurrency=30
merge_thread_count=70
sys_bkgd_net_percentage=100
data_disk_usage_limit_percentage=100
sys_bkgd_io_high_percentage=100
sys_bkgd_io_low_percentage=100
all_server_list=172.16.10.0:11940,172.16.10.1:11942,172.16.10.10:11944,172.16.10.100:11946,172.16.10.101:11948,172.16.10.102:11950,172.16.10.103:11952,172.16.10.104:11954,172.16.10.105:11956,172.16.10.106:11958,172.16.10.107:11960,172.16.10.108:11962,172.16.10.11:11964,172.16.10.110:11966,172.16.10.111:11968,172.16.10.112:11970,172.16.10.113:11972,172.16.10.114:11974,172.16.10.115:11976,172.16.10.116:11978,172.16.10.117:11980,172.16.10.118:11982,172.16.10.119:11984,172.16.10.12:11986,172.16.10.120:11988,172.16.10.121:11990,172.16.10.122:11992,172.16.10.123:11994,172.16.10.124:11996,172.16.10.125:11998,172.16.10.126:12000,172.16.10.127:12002,172.16.10.128:12004,172.16.10.129:12006,172.16.10.13:12008,172.16.14.246:12010,172.16.10.131:12012,172.16.10.132:12014,172.16.10.133:12016,172.16.10.134:12018,172.16.10.135:12020,172.16.10.136:12022,172.16.10.137:12024,172.16.10.138:12026,172.16.10.139:12028,172.16.10.14:12030,172.16.10.140:12032,172.16.10.141:12034,172.16.10.143:12036,172.16.10.144:12038,172.16.10.145:12040,172.16.10.146:12042,172.16.10.147:12044,172.16.10.148:12046,172.16.10.149:12048,172.16.10.15:12050,172.16.10.150:12052,172.16.10.151:12054,172.16.10.152:12056,172.16.10.153:12058,172.16.10.154:12060,172.16.10.155:12062,172.16.10.156:12064,172.16.10.157:12066,172.16.10.158:12068,172.16.10.159:12070,172.16.10.16:12072,172.16.10.160:12074,172.16.10.161:12076,172.16.10.162:12078,172.16.10.163:12080,172.16.10.164:12082,172.16.10.165:12084,172.16.10.166:12086,172.16.10.167:12088,172.16.10.168:12090,172.16.10.169:12092,172.16.10.17:12094,172.16.10.170:12096,172.16.10.171:12098,172.16.10.172:12100,172.16.10.173:12102,172.16.10.174:12104,172.16.10.175:12106,172.16.10.176:12108,172.16.10.177:12110,172.16.10.178:12112,172.16.10.179:12114,172.16.10.18:12116,172.16.10.180:12118,172.16.10.181:12120,172.16.10.182:12122,172.16.10.183:12124,172.16.15.117:12126,172.16.10.185:12128,172.16.10.186:12130,172.16.10.187:12132,172.16.10.188:12134,172.16.10.189:12136,172.16.10.19:12138,172.16.10.190:12140,172.16.10.191:12142,172.16.10.192:12144,172.16.10.193:12146,172.16.10.194:12148,172.16.10.195:12150,172.16.10.196:12152,172.16.10.197:12154,172.16.10.198:12156,172.16.10.199:12158,172.16.10.2:12160,172.16.10.20:12162,172.16.10.200:12164,172.16.10.201:12166,172.16.10.202:12168,172.16.10.203:12170,172.16.10.204:12172,172.16.10.205:12174,172.16.10.206:12176,172.16.10.207:12178,172.16.10.208:12180,172.16.10.209:12182,172.16.10.21:12184,172.16.10.210:12186,172.16.10.211:12188,172.16.15.119:12190,172.16.10.213:12192,172.16.10.214:12194,172.16.10.215:12196,172.16.10.216:12198,172.16.10.217:12200,172.16.10.218:12202,172.16.10.219:12204,172.16.10.22:12206,172.16.10.220:12208,172.16.10.221:12210,172.16.10.222:12212,172.16.10.223:12214,172.16.10.224:12216,172.16.10.225:12218,172.16.10.226:12220,172.16.10.227:12222,172.16.10.228:12224,172.16.10.229:12226,172.16.
```


16.13.89:13222,172.16.13.9:13224,172.16.13.90:13226,172.16.13.91:13228,172.16.13.92:13230,172.16.13.93:13232,172.16.13.94:13234,172.16.13.95:13236,172.16.13.97:13238,172.16.13.98:13240,172.16.13.99:13242,172.16.4.100:13244,172.16.4.101:13246,172.16.4.102:13248,172.16.4.103:13250,172.16.4.104:13252,172.16.4.105:13254,172.16.4.106:13256,172.16.4.107:13258,172.16.4.108:13260,172.16.4.109:13262,172.16.4.110:13264,172.16.4.111:13266,172.16.4.112:13268,172.16.4.113:13270,172.16.4.114:13272,172.16.4.115:13274,172.16.4.116:13276,172.16.4.117:13278,172.16.4.118:13280,172.16.4.119:13282,172.16.4.120:13284,172.16.4.121:13286,172.16.4.122:13288,172.16.4.123:13290,172.16.4.124:13292,172.16.4.125:13294,172.16.4.126:13296,172.16.4.127:13298,172.16.4.128:13300,172.16.4.129:13302,172.16.4.130:13304,172.16.4.131:13306,172.16.4.132:13308,172.16.4.133:13310,172.16.4.134:13312,172.16.4.135:13314,172.16.4.136:13316,172.16.4.137:13318,172.16.4.138:13320,172.16.4.139:13322,172.16.4.140:13324,172.16.4.141:13326,172.16.4.142:13328,172.16.4.143:13330,172.16.4.144:13332,172.16.4.145:13334,172.16.4.146:13336,172.16.4.147:13338,172.16.4.148:13340,172.16.4.149:13342,172.16.4.150:13344,172.16.4.151:13346,172.16.4.152:13348,172.16.4.153:13350,172.16.4.154:13352,172.16.4.155:13354,172.16.4.156:13356,172.16.4.157:13358,172.16.4.158:13360,172.16.4.159:13362,172.16.4.160:13364,172.16.4.161:13366,172.16.4.162:13368,172.16.4.164:13370,172.16.4.165:13372,172.16.4.166:13374,172.16.4.167:13376,172.16.4.168:13378,172.16.4.169:13380,172.16.4.170:13382,172.16.4.171:13384,172.16.4.72:13386,172.16.4.73:13388,172.16.4.74:13390,172.16.4.75:13392,172.16.4.76:13394,172.16.4.77:13396,172.16.4.78:13398,172.16.4.79:13400,172.16.4.80:13402,172.16.4.81:13404,172.16.4.82:13406,172.16.4.83:13408,172.16.4.84:13410,172.16.4.85:13412,172.16.4.86:13414,172.16.4.87:13416,172.16.4.88:13418,172.16.4.89:13420,172.16.4.90:13422,172.16.4.91:13424,172.16.4.92:13426,172.16.4.93:13428,172.16.4.94:13430,172.16.4.95:13432,172.16.4.96:13434,172.16.4.97:13436,172.16.4.98:13438,172.16.4.99:13440,172.16.5.100:13442,172.16.5.101:13444,172.16.5.102:13446,172.16.5.103:13448,172.16.5.104:13450,172.16.5.105:13452,172.16.5.106:13454,172.16.5.107:13456,172.16.5.108:13458,172.16.5.109:13460,172.16.5.110:13462,172.16.5.111:13464,172.16.5.112:13466,172.16.5.113:13468,172.16.5.114:13470,172.16.5.115:13472,172.16.5.116:13474,172.16.5.117:13476,172.16.5.118:13478,172.16.5.119:13480,172.16.5.20:13482,172.16.5.21:13484,172.16.5.216:13486,172.16.5.217:13488,172.16.5.218:13490,172.16.5.219:13492,172.16.5.22:13494,172.16.5.220:13496,172.16.5.221:13498,172.16.5.222:13500,172.16.5.223:13502,172.16.5.224:13504,172.16.5.225:13506,172.16.5.226:13508,172.16.5.227:13510,172.16.5.228:13512,172.16.5.229:13514,172.16.5.23:13516,172.16.5.230:13518,172.16.5.231:13520,172.16.5.232:13522,172.16.5.233:13524,172.16.5.234:13526,172.16.5.235:13528,172.16.5.236:13530,172.16.5.237:13532,172.16.5.238:13534,172.16.5.24:13536,172.16.5.240:13538,172.16.5.241:13540,172.16.5.242:13542,172.16.5.243:13544,172.16.5.244:13546,172.16.5.245:13548,172.16.5.246:13550,172.16.5.248:13552,172.16.5.249:13554,172.16.5.25:13556,172.16.5.250:13558,172.16.5.251:13560,172.16.5.252:13562,172.16.5.253:13564,172.16.5.254:13566,172.16.5.255:13568,172.16.5.26:13570,172.16.5.27:13572,172.16.5.28:13574,172.16.5.29:13576,172.16.5.30:13578,172.16.5.31:13580,172.16.5.32:13582,172.16.5.33:13584,172.16.5.34:13586,172.16.5.35:13588,172.16.5.36:13590,172.16.5.37:13592,172.16.5.38:13594,172.16.5.39:13596,172.16.5.40:13598,172.16.5.41:13600,172.16.5.42:13602,172.16.5.43:13604,172.16.5.44:13606,172.16.5.45:13608,172.16.5.46:13610,172.16.5.47:13612,172.16.5.49:13614,172.16.5.50:13616,172.16.5.51:13618,172.16.5.52:13620,172.16.5.53:13622,172.16.5.54:13624,172.16.5.55:13626,172.16.5.56:13628,172.16.5.57:13630,172.16.5.58:13632,172.16.5.59:13634,172.16.5.60:13636,172.16.5.61:13638,172.16.5.62:13640,172.16.5.63:13642,172.16.5.64:13644,172.16.5.65:13646,172.16.5.66:13648,172.16.5.67:13650,172.16.5.68:13652,172.16.5.69:13654,172.16.5.70:13656,172.16.5.72:13658,172.16.5.73:13660,172.16.5.74:13662,172.16.5.75:13664,172.16.5.76:13666,172.16.5.77:13668,172.16.5.78:13670,172.16.5.79:13672,172.16.5.80:13674,172.16.5.81:13676,172.16.5.82:13678,172.16.5.83:13680,172.16.5.84:13682,172.16.5.85:13684,172.16.5.86:13686,172.16.5.87:13688,172.16.5.88:13690,172.16.5.89:13692,172.16.5.90:13694,172.16.5.91:13696,172.16.5.92:13698,172.16.5.93:13700,172.16.5.94:13702,172.16.5.95:13704,172.16.5.96:13706,172.16.5.97:13708,172.16.5.98:13710,172.16.5.99:13712,172.16.6.0:13714,172.16.6.1:13716,172.16.6.10:13718,172.16.6.11:13720,172.16.6.12:13722,172.16.6.13:13724,172.16.6.14:13726,172.16.6.15:13728,172.16.6.16:13730,172.16.6.160:13732,172.16.6.161:13734,172.16.6.162:13736,172.16.6.163:13738,172.16.6.164:13740,172.16.6.165:13742,172.16.6.1

66:13744,172.16.6.167:13746,172.16.6.168:13748,172.16.6.169:13750,172.16.6.17:13752,172.16.6.170:13754,172.16.6.171:13756,172.16.6.172:13758,172.16.6.173:13760,172.16.6.174:13762,172.16.6.175:13764,172.16.6.176:13766,172.16.6.177:13768,172.16.6.178:13770,172.16.6.179:13772,172.16.6.18:13774,172.16.6.180:13776,172.16.6.181:13778,172.16.6.182:13780,172.16.6.183:13782,172.16.6.184:13784,172.16.6.185:13786,172.16.6.186:13788,172.16.6.187:13790,172.16.6.188:13792,172.16.6.189:13794,172.16.6.19:13796,172.16.6.190:13798,172.16.6.191:13800,172.16.6.193:13802,172.16.6.194:13804,172.16.6.195:13806,172.16.6.196:13808,172.16.6.197:13810,172.16.6.198:13812,172.16.6.199:13814,172.16.6.2:13816,172.16.6.200:13818,172.16.6.200:13820,172.16.6.201:13822,172.16.6.202:13824,172.16.6.203:13826,172.16.6.204:13828,172.16.6.205:13830,172.16.6.206:13832,172.16.6.207:13834,172.16.6.208:13836,172.16.6.209:13838,172.16.6.21:13840,172.16.6.210:13842,172.16.6.211:13844,172.16.6.212:13846,172.16.6.213:13848,172.16.6.214:13850,172.16.6.215:13852,172.16.6.216:13854,172.16.6.217:13856,172.16.6.218:13858,172.16.6.219:13860,172.16.6.22:13862,172.16.6.220:13864,172.16.6.221:13866,172.16.6.222:13868,172.16.6.223:13870,172.16.6.224:13872,172.16.6.225:13874,172.16.6.226:13876,172.16.6.227:13878,172.16.6.228:13880,172.16.6.229:13882,172.16.6.23:13884,172.16.6.230:13886,172.16.6.231:13888,172.16.6.232:13890,172.16.6.233:13892,172.16.6.234:13894,172.16.6.235:13896,172.16.6.236:13898,172.16.6.237:13900,172.16.6.238:13902,172.16.6.239:13904,172.16.6.24:13906,172.16.6.240:13908,172.16.6.241:13910,172.16.6.242:13912,172.16.6.243:13914,172.16.6.244:13916,172.16.6.245:13918,172.16.6.246:13920,172.16.6.247:13922,172.16.6.248:13924,172.16.6.249:13926,172.16.6.25:13928,172.16.6.250:13930,172.16.6.251:13932,172.16.6.252:13934,172.16.6.253:13936,172.16.6.254:13938,172.16.6.255:13940,172.16.6.26:13942,172.16.6.27:13944,172.16.6.28:13946,172.16.6.29:13948,172.16.6.3:13950,172.16.6.30:13952,172.16.6.31:13954,172.16.6.32:13956,172.16.6.33:13958,172.16.6.34:13960,172.16.6.35:13962,172.16.6.36:13964,172.16.6.37:13966,172.16.6.38:13968,172.16.6.39:13970,172.16.6.4:13972,172.16.6.40:13974,172.16.6.41:13976,172.16.6.42:13978,172.16.6.43:13980,172.16.6.44:13982,172.16.6.45:13984,172.16.6.46:13986,172.16.6.47:13988,172.16.6.48:13990,172.16.6.49:13992,172.16.6.5:13994,172.16.6.50:13996,172.16.6.51:13998,172.16.6.52:14000,172.16.6.53:14002,172.16.6.54:14004,172.16.6.55:14006,172.16.6.56:14008,172.16.6.57:14010,172.16.6.58:14012,172.16.6.59:14014,172.16.6.6:14016,172.16.6.7:14018,172.16.6.8:14020,172.16.6.9:14022,172.16.7.0:14024,172.16.7.1:14026,172.16.7.2:14028,172.16.7.3:14030,172.16.9.112:14032,172.16.9.113:14034,172.16.9.114:14036,172.16.9.115:14038,172.16.9.116:14040,172.16.9.117:14042,172.16.9.118:14044,172.16.9.119:14046,172.16.9.120:14048,172.16.9.121:14050,172.16.9.122:14052,172.16.9.123:14054,172.16.9.124:14056,172.16.9.125:14058,172.16.9.126:14060,172.16.9.127:14062,172.16.9.128:14064,172.16.9.129:14066,172.16.9.130:14068,172.16.9.131:14070,172.16.9.132:14072,172.16.9.133:14074,172.16.9.134:14076,172.16.9.135:14078,172.16.9.136:14080,172.16.9.137:14082,172.16.9.138:14084,172.16.9.139:14086,172.16.9.140:14088,172.16.9.141:14090,172.16.9.142:14092,172.16.9.143:14094,172.16.9.144:14096,172.16.9.145:14098,172.16.9.146:14100,172.16.9.147:14102,172.16.9.148:14104,172.16.9.149:14106,172.16.9.150:14108,172.16.9.151:14110,172.16.9.152:14112,172.16.9.153:14114,172.16.9.154:14116,172.16.9.155:14118,172.16.9.156:14120,172.16.9.157:14122,172.16.9.158:14124,172.16.9.159:14126,172.16.9.160:14128,172.16.9.161:14130,172.16.9.162:14132,172.16.9.163:14134,172.16.9.164:14136,172.16.9.165:14138,172.16.9.166:14140,172.16.9.167:14142,172.16.9.168:14144,172.16.9.169:14146,172.16.9.170:14148,172.16.9.171:14150,172.16.9.172:14152,172.16.9.173:14154,172.16.9.174:14156,172.16.9.175:14158,172.16.9.176:14160,172.16.9.177:14162,172.16.9.178:14164,172.16.9.179:14166,172.16.9.180:14168,172.16.9.181:14170,172.16.9.182:14172,172.16.9.183:14174,172.16.9.184:14176,172.16.9.185:14178,172.16.9.186:14180,172.16.9.187:14182,172.16.9.188:14184,172.16.9.189:14186,172.16.9.190:14188,172.16.9.191:14190,172.16.9.192:14192,172.16.9.193:14194,172.16.9.194:14196,172.16.9.195:14198,172.16.9.196:14200,172.16.9.197:14202,172.16.9.198:14204,172.16.9.199:14206,172.16.9.200:14208,172.16.9.201:14210,172.16.9.202:14212,172.16.9.203:14214,172.16.9.204:14216,172.16.9.205:14218,172.16.9.206:14220,172.16.9.207:14222,172.16.9.208:14224,172.16.9.209:14226,172.16.9.210:14228,172.16.9.211:14230,172.16.9.212:14232,172.16.9.213:14234,172.16.9.214:14236,172.16.9.215:14238,172.16.9.216:14240,172.16.9.217:14242,172.16.9.218:14244,172.16.9.219:14246,172.16.9.220:14248,172.16.9.221:14250,172.16.9.222:14252,172.16.9.223:14254,172.16.9.224:14256,172.16.9.225:14258,172.

16.9.226:14260,172.16.9.227:14262,172.16.9.228:14264,172.16.9.229:14266,172.16.9.230:14268,172.16.9.231:14270,172.16.9.232:14272,172.16.9.233:14274,172.16.9.234:14276,172.16.9.235:14278,172.16.9.236:14280,172.16.9.237:14282,172.16.9.238:14284,172.16.9.239:14286,172.16.9.240:14288,172.16.9.241:14290,172.16.9.242:14292,172.16.9.243:14294,172.16.9.244:14296,172.16.9.245:14298,172.16.9.246:14300,172.16.9.247:14302,172.16.9.248:14304,172.16.9.249:14306,172.16.9.250:14308,172.16.9.251:14310,172.16.9.252:14312,172.16.9.253:14314,172.16.9.254:14316,172.16.9.255:14318,172.16.14.11:5:14320,172.16.14.110:14322,172.16.14.128:14324,172.16.14.152:14326,172.16.14.101:14328,172.16.14.151:14330,172.16.14.147:14332,172.16.14.82:14334,172.16.14.146:14336,172.16.14.141:14338,172.16.14.72:14340,172.16.14.74:14342,172.16.14.149:14344,172.16.14.75:14346,172.16.14.148:14348,172.16.14.109:14350,172.16.14.112:14352,172.16.14.125:14354,172.16.14.127:14356,172.16.14.139:14358,172.16.14.130:14360,172.16.14.78:14362,172.16.14.133:14364,172.16.14.92:14366,172.16.14.137:14368,172.16.14.116:14370,172.16.14.120:14372,172.16.14.113:14374,172.16.14.94:14376,172.16.14.123:14378,172.16.14.108:14380,172.16.14.122:14382,172.16.14.73:14384,172.16.14.153:14386,172.16.14.67:14388,172.16.14.64:14390,172.16.14.121:14400,172.16.14.71:14394,172.16.14.100:14396,172.16.14.8:14398,172.16.14.65:14400,172.16.14.105:14402,172.16.14.119:14404,172.16.14.79:14406,172.16.14.81:14408,172.16.14.136:14410,172.16.14.103:14412,172.16.14.76:14414,172.16.14.86:14416,172.16.14.140:14418,172.16.14.131:14420,172.16.14.106:14422,172.16.14.84:14424,172.16.14.69:14426,172.16.14.129:14428,172.16.14.150:14430,172.16.14.83:14432,172.16.14.90:14434,172.16.14.138:14436,172.16.14.96:14438,172.16.14.102:14440,172.16.14.135:14442,172.16.14.95:14444,172.16.14.107:14446,172.16.14.88:14448,172.16.14.121:14450,172.16.14.89:14452,172.16.14.118:14454,172.16.14.97:14456,172.16.14.142:14458,172.16.14.85:14460,172.16.14.132:14462,172.16.14.80:14464,172.16.14.134:14466,172.16.14.70:14468,172.16.14.143:14470,172.16.14.154:14472,172.16.14.68:14474,172.16.14.126:14476,172.16.14.145:14478,172.16.14.104:14480,172.16.14.66:14482,172.16.14.91:14484,172.16.14.77:14486,172.16.14.117:14488,172.16.14.111:14490,172.16.14.114:14492,172.16.14.98:14494,172.16.14.93:14496,172.16.14.99:14498,172.16.13.254:14500,172.16.14.61:14502,172.16.13.255:14504,172.16.14.0:14506,172.16.14.53:14508,172.16.13.253:14510,172.16.13.241:14512,172.16.13.249:14514,172.16.14.39:14516,172.16.14.2:14518,172.16.14.60:14520,172.16.13.246:14522,172.16.14.35:14524,172.16.14.31:14526,172.16.14.55:14528,172.16.14.40:14530,172.16.14.54:14532,172.16.14.51:14534,172.16.14.29:14536,172.16.14.56:14538,172.16.13.221:14540,172.16.14.50:14542,172.16.13.235:14544,172.16.14.26:14546,172.16.13.227:14548,172.16.14.36:14550,172.16.14.4:14552,172.16.13.222:14554,172.16.13.226:14556,172.16.13.242:14558,172.16.14.57:14560,172.16.14.42:14562,172.16.14.52:14564,172.16.14.44:14566,172.16.14.9:14568,172.16.13.238:14570,172.16.13.250:14572,172.16.14.33:14574,172.16.14.3:14576,172.16.14.41:14578,172.16.13.251:14580,172.16.14.38:14582,172.16.13.245:14584,172.16.14.48:14586,172.16.14.11:14588,172.16.13.248:14590,172.16.14.25:14592,172.16.14.47:14594,172.16.13.230:14596,172.16.14.27:14598,172.16.14.22:14600,172.16.14.49:14602,172.16.14.16:14604,172.16.13.220:14606,172.16.13.233:14608,172.16.13.234:14610,172.16.14.18:14612,172.16.14.7:14614,172.16.14.5:14616,172.16.13.231:14618,172.16.13.240:14620,172.16.14.28:14622,172.16.13.252:14624,172.16.14.6:14626,172.16.13.228:14628,172.16.13.239:14630,172.16.14.37:14632,172.16.14.13:14634,172.16.13.225:14636,172.16.13.224:14638,172.16.14.30:14640,172.16.13.229:14642,172.16.13.244:14644,172.16.14.24:14646,172.16.14.59:14648,172.16.14.45:14650,172.16.14.14:14652,172.16.13.237:14654,172.16.14.15:14656,172.16.13.247:14658,172.16.14.62:14660,172.16.14.10:14662,172.16.13.243:14664,172.16.13.232:14666,172.16.14.46:14668,172.16.14.8:14670,172.16.14.32:14672,172.16.14.58:14674,172.16.14.2:14676,172.16.14.34:14678,172.16.14.1:14680,172.16.14.20:14682,172.16.14.19:14684,172.16.14.43:14686,172.16.13.236:14688,172.16.14.63:14690,172.16.14.17:14692,172.16.14.23:14694,172.16.15.70:14696,172.16.15.62:14698,172.16.15.85:14700,172.16.15.40:14702,172.16.15.82:14704,172.16.15.38:14706,172.16.15.34:14708,172.16.15.79:14710,172.16.15.78:14712,172.16.15.33:14714,172.16.15.75:14716,172.16.15.67:14718,172.16.15.69:14720,172.16.15.28:14722,172.16.15.65:14724,172.16.15.76:14726,172.16.15.88:14728,172.16.15.61:14730,172.16.15.32:14732,172.16.15.45:14734,172.16.15.87:14736,172.16.15.46:14738,172.16.15.94:14740,172.16.15.25:14742,172.16.15.92:14744,172.16.15.77:14746,172.16.15.26:14748,172.16.15.95:14750,172.16.15.72:14752,172.16.15.66:14754,172.16.15.80:14756,17

2.16.15.59:14758,172.16.15.51:14760,172.16.15.36:14762,172.16.15.27:14764,172.16.15.58:14766,172.16.15.29:14768,172.16.15.47:14770,172.16.15.96:14772,172.16.15.73:14774,172.16.15.86:14776,172.16.15.89:14778,172.16.15.48:14780,172.16.15.84:14782,172.16.15.83:14784,172.16.15.30:14786,172.16.15.99:14788,172.16.15.35:14790,172.16.15.90:14792,172.16.15.53:14794,172.16.15.43:14796,172.16.15.93:14798,172.16.15.37:14800,172.16.15.31:14802,172.16.14.233:14804,172.16.15.60:14806,172.16.15.42:14808,172.16.15.98:14810,172.16.15.68:14812,172.16.15.100:14814,172.16.15.41:14816,172.16.15.56:14818,172.16.15.50:14820,172.16.15.91:14822,172.16.15.81:14824,172.16.15.39:14826,172.16.15.71:14828,172.16.15.49:14830,172.16.15.44:14832,172.16.15.24:14834,172.16.15.63:14836,172.16.15.57:14838,172.16.15.97:14840,172.16.15.74:14842,172.16.15.54:14844,172.16.15.64:14846,172.16.15.52:14848,172.16.15.11:14850,172.16.15.9:14852,172.16.14.227:14854,172.16.14.255:14856,172.16.14.229:14858,172.16.15.3:14860,172.16.14.206:14862,172.16.14.196:14864,172.16.14.210:14866,172.16.14.243:14868,172.16.14.213:14870,172.16.15.10:14872,172.16.14.235:14874,172.16.14.234:14876,172.16.14.190:14878,172.16.15.16:14880,172.16.14.240:14882,172.16.14.182:14884,172.16.15.18:14886,172.16.15.23:14888,172.16.15.19:14890,172.16.14.236:14892,172.16.14.217:14894,172.16.15.12:14896,172.16.14.207:14898,172.16.15.15:14900,172.16.14.220:14902,172.16.14.248:14904,172.16.14.209:14906,172.16.14.198:14908,172.16.14.226:14910,172.16.15.20:14912,172.16.15.13:14914,172.16.14.221:14916,172.16.15.1:14918,172.16.14.200:14920,172.16.14.204:14922,172.16.15.21:14924,172.16.14.208:14926,172.16.14.194:14928,172.16.14.202:14930,172.16.14.232:14932,172.16.14.251:14934,172.16.14.212:14936,172.16.14.228:14938,172.16.14.216:14940,172.16.14.250:14942,172.16.15.22:14944,172.16.14.249:14946,172.16.14.224:14948,172.16.14.238:14950,172.16.14.214:14952,172.16.14.215:14954,172.16.15.5:14956,172.16.14.223:14958,172.16.15.14:14960,172.16.15.17:14962,172.16.14.254:14964,172.16.14.188:14966,172.16.14.218:14968,172.16.14.203:14970,172.16.14.230:14972,172.16.14.201:14974,172.16.15.0:14976,172.16.14.237:14978,172.16.14.205:14980,172.16.14.239:14982,172.16.15.7:14984,172.16.14.219:14986,172.16.14.184:14988,172.16.14.185:14990,172.16.14.247:14992,172.16.14.242:14994,172.16.15.2:14996,172.16.15.6:14998,172.16.14.183:15000,172.16.14.245:15002,172.16.14.197:15004,172.16.15.8:15006,172.16.14.187:15008,172.16.14.180:15010,172.16.14.193:15012,172.16.14.253:15014,172.16.14.244:15016,172.16.14.225:15018,172.16.14.189:15020,172.16.14.191:15022,172.16.14.195:15024,172.16.14.199:15026,172.16.15.4:15028,172.16.14.181:15030,172.16.14.186:15032,172.16.14.192:15034,172.16.14.241:15036,172.16.14.252:15038,172.16.15.133:15040,172.16.15.125:15042,172.16.15.129:15044,172.16.15.131:15046,172.16.15.127:15048,172.16.15.128:15050,172.16.15.132:15052

location_cache_refresh_min_interval=1s
location_refresh_thread_count=2
virtual_table_location_cache_expire_time=100h
location_cache_expire_time=100h
_ob_enable_fast_load=True
_ob_gts_reload_interval=5s
_ob_freeze_info_reload_interval=10s
_ob_gts_batch_count_between_replicas=3
_ob_get_gts_ahead_interval=8ms
_ob_clog_disk_reuse_percentage=76
_ob_btree_retire_limit=1024
_ob_enable_log_hot_cache=True
_ob_enable_auto_freeze_based_on_clog_disk=False
_ob_enable_log_batch_rpc=False
_ob_clog_timeout_to_force_switch_leader=10m
clog_max_unconfirmed_log_count=20000
gts_refresh_interval=5ms
election_blacklist_interval=0
clog_transport_compress_all=True
row_compaction_update_limit=6


```

max_kept_major_version_number=1
merger_check_interval=10s
enable_merge_by_turn=False
zone_merge_concurrency=2
major_freeze_duty_time=Disable
__min_full_resource_pool_memory=1073741824
server_data_copy_out_concurrency=3
server_data_copy_in_concurrency=10
balancer_idle_time=5d
enable_rebalance=False
enable_rereplication=False
resource_soft_limit=100
min_observer_version=2.2.21
partition_table_check_interval=10000d
replica_safe_remove_time=5m
partition_table_scan_batch_count=2000
server_permanent_offline_time=48h
merger_warm_up_duration_time=0s
__partition_distribution_degree=1
__create_table_partition_distribution_strategy=2
freeze_trigger_percentage=60
memstore_limit_percentage=80
__enable_freeze_trigger_dispersed=True
cpu_quota_concurrency=6
large_query_threshold=1000s
workers_per_cpu_quota=8
__hash_area_size=10G
memory_chunk_cache_size=580G
memory_limit_percentage=85
enable_syslog_recycle=True
enable_syslog_wf=False
max_syslog_file_count=100
syslog_level=INFO
obconfig_url=http://172.16.9.110:8080/oceanbase_obconfig/ob1.tpc.172.16.9.111.rslst
cluster_id=11
cluster=ob1.tpc
rootservice_list=172.16.10.0:11940:11941;172.16.10.1:11942:11943;172.16.10.10:11944:11945
weak_read_version_refresh_interval=0
schema_history_expire_time=1d
enable_separate_sys_clog=False
enable_perf_event=False
enable_record_trace_id=False
enable_sql_audit=False
__ob_enable_one_phase_commit=False
enable_record_trace_log=False
trace_log_slow_query_watermark=10s
__minor_deferred_gc_level=0
minor_deferred_gc_time=10m
minor_warm_up_duration_time=0
system_memory=50G

```

```

memory_limit=660GB
high_priority_net_thread_count=4
net_thread_count=6
zone=zone1
devname=eth0
mysql_port=11947
rpc_port=11946
__ob_checkpoint_space_limit=1000GB
datafile_disk_percentage=50
datafile_size=3982GB
data_dir=/home/tpcc/ob1.obs3/store

```

Client Node

File system info

Filesystem	1K-blocks	Used	Available	Use%	Mounted on
devtmpfs	197284956	0	197284956	0%	/dev
tmpfs	197295928	0	197295928	0%	/dev/shm
tmpfs	197295928	2304	197293624	1%	/run
tmpfs	197295928	0	197295928	0%	/sys/fs/cgroup
/dev/vda1	41147472	2929468	36314512	8%	/
/dev/vdb1	191196044	10651096	170762984	6%	/home
tmpfs	39459188	0	39459188	0%	/run/user/1000

/etc/security/limits.conf

```

* soft nofile 1048576
* hard nofile 1048576
* soft nproc 655360
* hard nproc 655360
* soft core unlimited
* hard stack 20480
* soft stack 20480
root soft nofile 1048576
root hard nofile 1048576
root soft nproc 655360
root hard nproc 655360
root soft core unlimited
root hard stack 20480
root soft stack 20480

```

Nginx.conf

```

#daemon off;
#user nobody;

```

```

env ODBC_LOG_ENABLE=false;
env ODBC_LOG_LEVEL=3;
env ODBC_SYSINI=/home/tpcc/ob1.c0/tpcc/odbc;
env ODBCINI=/home/tpcc/ob1.c0/tpcc/odbc/odbc.ini;
env LD_LIBRARY_PATH=/home/tpcc/ob1.c0/tpcc/ob1.c0/tpcc/ob1.c0/tpcc/odbc;
worker_processes 1;
worker_rlimit_nofile 1048575;

error_log logs/error.log error;
#error_log logs/error.log emerg;
#error_log logs/error.log notice;
#error_log logs/error.log info;

#pid logs/nginx.pid;

events {
    worker_connections 60000;
}

thread_pool delivery_thread_pool threads=100 max_queue=50000;

http {
    include mime.types;
    default_type application/octet-stream;
    # lua_package_path '/nginx1142-async/lualib/?.lua;;';

    #log_format main '$remote_addr - $remote_user [$time_local] "$request" '
    # '$status $body_bytes_sent "$http_referer" '
    # '"$http_user_agent" "$http_x_forwarded_for"';

    #access_log logs/access.log main;

    access_log off;
    sendfile on;
    #tcp_nopush on;

    #keepalive_timeout 0;
    keepalive_timeout 3600;
    keepalive_requests 10000000;

    #gzip on;
    upstream backend {
        oceanbase_server dsn=tpccdb password=123456 user=TPCC@tpcc;
        oceanbase_keepalive max=200 overflow=reject;
    }

    server {
        listen 63019 reuseport backlog=102400;
#        listen 63020 reuseport backlog=102400;

```

```

#        listen 63021 reuseport backlog=102400;
#        listen 63022 reuseport backlog=102400;
#        listen 63023 reuseport backlog=102400;
#        listen 63024 reuseport backlog=102400;
#        listen 63025 reuseport backlog=102400;
#        listen 63026 reuseport backlog=102400;
#        listen 63027 reuseport backlog=102400;
#        listen 63028 reuseport backlog=102400;
#        listen 63029 reuseport backlog=102400;
#        listen 63030 reuseport backlog=102400;
#        listen 63031 reuseport backlog=102400;
#        listen 63032 reuseport backlog=102400;
#        listen 63033 reuseport backlog=102400;
#        listen 63034 reuseport backlog=102400;
#        listen 63035 reuseport backlog=102400;
#        listen 63036 reuseport backlog=102400;
#        listen 63037 reuseport backlog=102400;
#        listen 63038 reuseport backlog=102400;
#        listen 63039 reuseport backlog=102400;
#        listen 63040 reuseport backlog=102400;
#        listen 63041 reuseport backlog=102400;
#        listen 63042 reuseport backlog=102400;
#        listen 63043 reuseport backlog=102400;
#        listen 63044 reuseport backlog=102400;
#        listen 63045 reuseport backlog=102400;
#        listen 63046 reuseport backlog=102400;
#        listen 63047 reuseport backlog=102400;
#        listen 63048 reuseport backlog=102400;
#        listen 63049 reuseport backlog=102400;
#        listen 63050 reuseport backlog=102400;
#        listen 63051 reuseport backlog=102400;
#        listen 63052 reuseport backlog=102400;
#        listen 63053 reuseport backlog=102400;
#        listen 63054 reuseport backlog=102400;
#        listen 63055 reuseport backlog=102400;
#        listen 63056 reuseport backlog=102400;
#        listen 63057 reuseport backlog=102400;
#        listen 63058 reuseport backlog=102400;
#        listen 63059 reuseport backlog=102400;
#        listen 63060 reuseport backlog=102400;
#        listen 63061 reuseport backlog=102400;
#        listen 63062 reuseport backlog=102400;
#        listen 63063 reuseport backlog=102400;
#        listen 63064 reuseport backlog=102400;
#        listen 63065 reuseport backlog=102400;
#        listen 63066 reuseport backlog=102400;
#        listen 63067 reuseport backlog=102400;
#        listen 63068 reuseport backlog=102400;
#        listen 63069 reuseport backlog=102400;
#        listen 63070 reuseport backlog=102400;

```

```

# listen 63071 reuseport backlog=102400;
# listen 63072 reuseport backlog=102400;
# listen 63073 reuseport backlog=102400;
# listen 63074 reuseport backlog=102400;
# listen 63075 reuseport backlog=102400;
# listen 63076 reuseport backlog=102400;
# listen 63077 reuseport backlog=102400;
# listen 63078 reuseport backlog=102400;
# listen 63079 reuseport backlog=102400;
# listen 63080 reuseport backlog=102400;
# listen 63081 reuseport backlog=102400;
# listen 63082 reuseport backlog=102400;
server_name localhost;

#charset koi8-r;

#access_log logs/host.access.log main;

location / {
    root html;
    index index.html index.htm;
}

#error_page 404 /404.html;

# redirect server error pages to the static page /50x.html
#
error_page 500 502 503 504 /50x.html;
location = /50x.html {
    root html;
}

location /tpcc {
    tpcc_terminal_num 4000000;
    tpcc_delivery_queue group_id=19 node_id=1 size=50000;
    tpcc_max_txn_cnt 0;
    tpcc_delivery_mdb dsn=tpccmeta user=tpcc password=tpcc;
    tpcc_delivery_runtime start=1587472244 rup=3300 rdown=150 mi=28800 ri=30 wts=70;

    oceanbase_pass backend;
    oceanbase_connect_timeout 600s; # default 60s
    oceanbase_send_query_timeout 600s; # default 60s
}

# proxy the PHP scripts to Apache listening on 127.0.0.1:80
#
#location ~ /\.php$ {
#    proxy_pass http://127.0.0.1;
#}

```

```

# pass the PHP scripts to FastCGI server listening on 127.0.0.1:9000
#
#location ~ /\.php$ {
#    root html;
#    fastcgi_pass 127.0.0.1:9000;
#    fastcgi_index index.php;
#    fastcgi_param SCRIPT_FILENAME /scripts$fastcgi_script_name;
#    include fastcgi_params;
#}

# deny access to .htaccess files, if Apache's document root
# concurs with nginx's one
#
#location ~ /\.ht {
#    deny all;
#}
}

# another virtual host using mix of IP-, name-, and port-based configuration
#
#server {
#    listen 8000;
#    listen somename:8080;
#    server_name somename alias another.alias;

#    location / {
#        root html;
#        index index.html index.htm;
#    }
#}

# HTTPS server
#
#server {
#    listen 443 ssl;
#    server_name localhost;

#    ssl_certificate cert.pem;
#    ssl_certificate_key cert.key;

#    ssl_session_cache shared:SSL:1m;
#    ssl_session_timeout 5m;

#    ssl_ciphers HIGH:!aNULL:!MD5;
#    ssl_prefer_server_ciphers on;

#    location / {
#        root html;

```

```
# index index.html index.htm;  
# }  
#}  
  
}
```

OCP Node

File system info

Filesystem	1K-blocks	Used	Available	Use%	Mounted on
devtmpfs	197285124	0	197285124	0%	/dev
tmpfs	197296100	0	197296100	0%	/dev/shm
tmpfs	197296100	2336	197293764	1%	/run
tmpfs	197296100	0	197296100	0%	/sys/fs/cgroup
/dev/vda1	41147472	2654652	36589328	7%	/
/dev/vdb1	191196044	39797296	141616784	22%	/home
tmpfs	39459220	0	39459220	0%	/run/user/1000

/etc/security/limits.conf

```
* soft nfile 1048576  
* hard nfile 1048576  
* soft nproc 655360  
* hard nproc 655360  
* soft core unlimited  
* hard stack 20480  
* soft stack 20480  
root soft nfile 1048576  
root hard nfile 1048576  
root soft nproc 655360  
root hard nproc 655360  
root soft core unlimited  
root hard stack 20480  
root soft stack 20480
```

Appendix D: Third Party Price Quotes

ecs.i2d.21xlarge

The screenshot displays the Alibaba Cloud ECS console configuration page for an instance of type `ecs.i2d.21xlarge`. The configuration is divided into several sections:

- 基础配置 (Basic Configuration):**
 - 付费模式: 包年包月 (Pay-as-you-go)
 - 购买数量: 1台 (Purchase quantity: 1)
 - 本地存储: NVMe SSD (4 * 3570 GiB) (Local storage: NVMe SSD (4 * 3570 GiB))
 - 地域及可用区: 华南2 (河源) / 随机分配 (Region and availability zone: South China 2 (Heyuan) / Randomly assigned)
 - 镜像: Aliyun Linux 2.1903 LTS 64位(安全加固) (Image: Aliyun Linux 2.1903 LTS 64-bit (Security reinforced))
 - 数据盘: 1块 (Data disk: 1)
 - 实例: 本地SSD型实例 / ecs.i2d.21xlarge(84vCPU 712GiB) (Instance: Local SSD type instance / ecs.i2d.21xlarge (84 vCPU 712 GiB))
 - 系统盘: 高效云盘 40GiB, 随实例释放 (System disk: High-efficiency cloud disk 40 GiB, released with instance)
- 网络和安全组 (Network and Security Group):**
 - 网络: 专有网络 (Network: VPC)
 - 公网带宽: 按固定带宽 1Mbps (Public network bandwidth: Fixed bandwidth 1 Mbps)
 - VPC: 默认专有网络 (VPC: Default VPC)
 - 安全组: 1), 默认安全组 (自定义端口) (Security group: 1), Default security group (Custom ports))
 - 交换机: 默认交换机 (Switch: Default switch)
- 系统配置 (System Configuration):**
 - 登录凭证: 创建后设置, 如需远程登录ECS, 可返回第三步系统配置里配置登录凭证 (Logon credential: Set after creation, required for remote login of ECS, return to step 3 to configure login credential)
 - 实例名称: launch-advisor-20200508 (Instance name: launch-advisor-20200508)

Additional options include "保存为自动模板" (Save as auto template) and "生成Open API最佳实践脚本" (Generate Open API best practice script). The purchase duration is set to 3 years, and the public network bandwidth is 1 Mbps. The total configuration cost is ¥349,287.16, with a product discount of ¥282,628.76.

ecs.ebm6.26xlarge

阿里云 云服务器 ECS

一键购买 自定义购买

基础配置 网络和安全组 系统配置 (选填) 分组设置 (选填) 确认订单

所选配置

基础配置	付费模式: 包年包月 购买数量: 1 台 数据盘: 1 块 ...	地域及可用区: 华南 2 (河源) / 随机分配 镜像: Aliyun Linux 2.1903 LTS 64位(安全加固)	实例: 通用型弹性裸金属服务器 ebmg6 / ecs.ebm6.26xlarge(104vCPU 384GB) 系统盘: 高效云盘 40GB, 随实例释放
网络和安全组	网络: 专有网络 公网带宽: 按固定带宽 1Mbps	VPC: 默认专有网络 安全组: 1) 默认安全组 (自定义端口)	交换机: 默认交换机
系统配置	登录凭证: 创建后设置, 若需远程登录ECS 可返回第三步系统配置里配置登录凭证	实例名称: launch-advisor-20200508	

保存为自动模板 生成Open API最佳实践脚本

购买时长: 1周 1个月 2个月 3个月 半年 1年 2年 3年 4年 5年 更多

自动续费: 启用自动续费

服务协议: 《云服务器 ECS 服务条款》
[退款规则及操作说明](#) | [购买须知](#)
 订单对应的变更信息, 请在 [管理控制台](#)-[费用中心](#)-[发票管理](#) 中设置。
 云产品默认采用 TCP 25 端口和基于此端口的邮箱服务, 特殊需求需报单审核后使用。 [查看详情](#)

购物车 | 购物车 | 添加更多产品

全选

云服务器ECS(包月) 节省 ¥202,176.00 ¥250,956.00

规格: 1Mbps, 按固定带宽
I/O 优化实例: I/O 优化实例
实例: 104核 384GB
系统盘: 高效云盘 40GB
可用区: 随机分配
地域: cn-heyuan
网络类型: 专有网络
操作系统: Aliyun Linux 2.1903 LTS 64...
数据盘: 随实例释放, 200GB, 高效...
数量: 1
购买时长: 3年

总配置费用: ¥250,956.00
产品优惠 ¥202,176.00

购买时长: 3年
公网带宽: 1Mbps 按固定带宽

配置费用: ¥250956.00
实例包3年享5折 省 ¥ 202176.00

上一步: 分组设置 加入购物车 确认下单 去购物车结算

Alibaba Cloud Support Price

支持计划

<< 回到

支持计划

商业级

企业级

顶级

订购企业级服务，将享受如下服务或权益
7X24小时云产品支持
提供IM企业群、工单联系渠道
可免费使用应用高可用服务AHAS（专业版）架构感知
可免费使用智能顾问Advisor

购买时长

1个月

2个月

3个月

4个月

5个月

6个月

7个月

8个月

9个月

10个月

11个月

1年

包年订购顶级服务，将额外享受如下服务或权益
云上护航服务（标准版）
应用水位评估服务
容灾服务
年度服务报告

总配置费用

¥ 300,000.00

立即购买

加入购物车

Lenovo ThinkPad X1 Yoga 2019 14" Laptop

首页 > 商品详情 > ThinkPad X1 Yoga2019英特尔酷睿i7 笔记本电脑20SA000BCD 水雾灰

联系客服



第十代英特尔® 酷睿™ i7

ThinkPad X1 Yoga2019英特尔酷睿i7 笔记本电脑20SA000BCD 水雾灰

全新十代英特尔酷睿i7-10710U/Windows 10家庭中文版/16GB/1TB SSD/手写笔/Intel® UHD Graphics /14英寸WQHD IPS LED背光触控显示屏/水雾灰

商城价	¥16999	累计评价 472
活动	赠品 赠完为止	
	VIP服务 七天无理由退换货 注册有礼 专属定制	展开活动
增值业务	支持 以旧换新 高价回收，急速到账换新机	
操作系统	Windows 10家庭中文版	
选择配置	八代i7/16GB/512GB SSD	八代i7/16GB/2TB SSD/14英寸UHD HD...
	八代i7/16GB/2TB SSD	十代i7/16GB/512GB SSD
	十代i7/16GB/1TB SSD	十代i5/8GB/512GB SSD
私人定制	选择A面喷图	更多专属定制 定制说明

OpenResty Support Price



报价单

客户名称	蚂蚁金服(杭州)网络技术有限公司	供货单位	鸥锐软件开发(天津)有限公司
地址	杭州市西湖区西溪新座5幢706室	地址	天津市武清区京津电子商务产业园宏旺道2号13号楼279室
电话	0571-26888888	电话	010-61654367
传真	0571-86432799	传真	010-61654367
联系人	邹银超	联系人	朱德江
手机	13456869396	手机	18620212952
邮箱	yinchao.zych@antfin.com	邮箱	dejiang@openresty.com
报价日期	2020年4月22日	项目	软件开发/软件咨询

感谢您的垂询。我们很高兴给您提供如下报价，价格明细表如下：

条目	产品名	单价 (CNY)	数量	价格 (CNY)
技术支持	OpenResty Professional Edition Support Service for 3 years 7x24x4hrs OpenResty 1.15.8.1 -Nginx 1.15.8 -lua-resty-odbc 1.0 -lua-resty-transaction-queue 1.0 -nginx-admin-plus 2.0.3	15,300.00	400	6,120,000.00
合计				CNY6,120,000.00

备注：

1. 以上报价 180 天内有效，超过 180 天请重新询价。
2. 此报价为含税报价 (17%增值税)
3. 付款方式，要求 100% 预付。
4. 此报价仅限客户需求数量报价，如其他数量则需另外提供报价。

如果对上述条款有任何问题，请随时向我们咨询。

鸥锐软件开发(天津)有限公司

