TPC Benchmark ^{TM} C
Full Disclosure Report

Alibaba Cloud Elastic Compute Service Cluster (with 204 OceanBase Data Nodes)
Using
OceanBase v2.2 Enterprise Edition with Partitioning, Horizontal Scalability and Advanced Compression

First Edition
October 2, 2019

First Edition –October 2, 2019

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.

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 Francois Raab and 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 204 OceanBase Data Nodes)	64 vCPU, 2.5 GHz Intel Xeon Platinum 8163(Skylake) (Per ecs.i2.16xlarge instance)	OceanBase v2.2 Enterprise Edition with Partitioning, Horizontal Scalability and Advanced Compression	Aliyun Linux 2

TPC Benchmark C Metrics

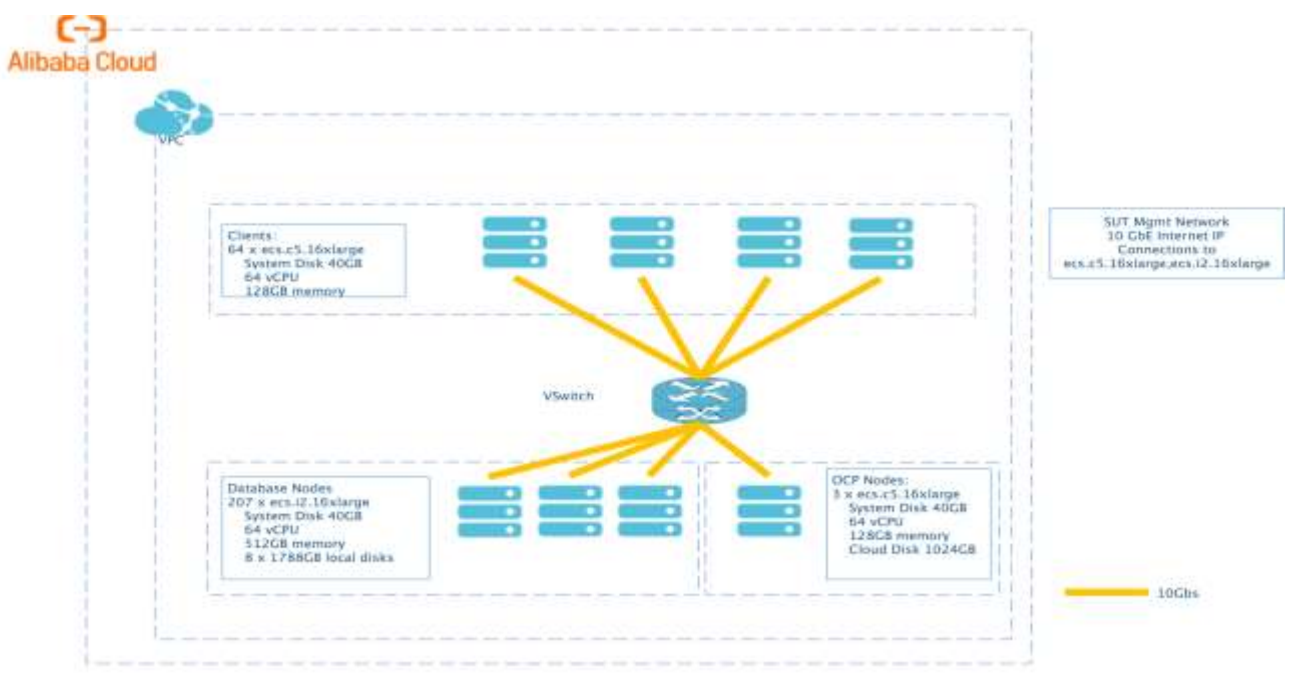
Total System Cost	TPC-C Throughput	Price / Performance	Availability Date
Three year cost includes: • Hardware • Software • Maintenance	Maximum Qualified Throughput expressed as transactions per minute – C (tpmC)	Total System Cost / tpmC	Date for which all components, hardware and software are available for purchase
CNY380, 452,842	60,880,800	CNY6.25	October 2, 2019

Alibaba Cloud Elastic Compute Service Cluster (with 204 OceanBase Data Nodes)

TPC-C 5.11.0
TPC-Pricing 2.4.0

Report Date
October 2, 2019

Total System Cost	TPC-C Throughput	Price/Performance	Availability Date	
CNY380, 452,842	**60,880,800 tpmC**	**CNY6.25/tpmC**	**October 2, 2019**	
Database Server Processors/Cores/Threads	Database Manager	Operating System	Other Software	Number of Users
Intel Xeon Platinum 8163(Skylake) 2.5GHz 13,248 vCPU	OceanBase v2.2 Enterprise Edition with Partitioning, Horizontal Scalability and Advanced Compression	Aliyun Linux 2	Nginx 1.15.8	47,942,400



System Component	207 x Database Nodes (each with)		3 x OCP Nodes (each with)		64 x Clients (each with)		Total
Proc./Core/Thread (Virtual CPU)	2/32/64 (64 vCPU)	Intel Xeon Platinum 8163 (2.5GHz, 33MB)	2/32/64 (64 vCPU)	Intel Xeon Platinum 8163 (2.5GHz, 33MB)	2/32/64 (64 vCPU)	Intel Xeon Platinum 8163 (2.5GHz, 33MB)	548/8,768/17,536 (17,536 vCPU)
Memory		512GB		128GB		128GB	114,560GB
OS Disk	1	40GB Ultra Cloud Disk	1	40GB Ultra Cloud Disk	1	40GB Ultra Cloud Disk	
Local Disk	8	1,788GB NVMe SSD					
Cloud Disk			1	1,024GB Cloud SSD			
Total Storage		2,969, 208GB		3,192GB		2,560GB	2,974,960GB

ANT FINANCIAL

Alibaba Cloud Elastic Compute Service Cluster (with 204 OceanBase Data Nodes)

TPC-C 5.11.0
TPC-Pricing 2.4.0

Report Date
October 2, 2019

Description	Part Number	Src	Unit Price (CNY)	Qty	Ext. Price (CNY)	3-Y Maintenance (CNY)
Licensed Compute Services						
Virtual cloud server						
ECS Instance ecs.i2.16xlarge (3-year price) - NVMe SSD Local Disk (8 x 1788 GB) - vCPU (64) - Memory (512GB)	ecs.i2.16xlarge (East China 2)	2	287,343.00	207	59,480,001.00	
ECS System Disk (Ultra Cloud Disk 40GB)	Included	2	0.00	207	0.00	
Aliyun Linux 2	Included	2	0.00	207	0.00	
Virtual cloud server						
ECS Instance ecs.c5.16xlarge (3-year price) - vCPU (64) - Memory (128GB)	ecs.c5.16xlarge (East China 2)	2	114,370.20	64	7,319,692.80	
ECS System Disk (Ultra Cloud Disk 40GB)	Included	2	0.00	64	0.00	
Aliyun Linux 2	Included	2	0.00	64	0.00	
Virtual cloud server						
ECS Instance ecs.c5.16xlarge (3-year price) - vCPU (64) - Memory (128GB)	ecs.c5.16xlarge (East China 2)	2	132,802.20	3	398,406.60	
ECS System Disk (Ultra Cloud Disk 40GB)	Included	2	0.00	3	0.00	
ECS Cloud Disk (Standard SSD 1024GB)	Included	2	0.00	3	0.00	
Aliyun Linux 2	Included	2	0.00	3	0.00	
				Sub-Total	67,198,100.40	
Licensed Software Services						
OceanBase v2.2 Enterprise Edition, per vCPU for 3 years		1	16,500.00	13,440	221,760,000.00	
OceanBase v2.2 Partitioning, per vCPU for 3 years		1	3437.50	13,440	46,200,000.00	
OceanBase v2.2 Horizontal Scalability, per vCPU for 3 years		1	3437.50	13,440	46,200,000.00	
OceanBase v2.2 Advanced Compression, per vCPU for 3 years		1	3437.50	13,440	46,200,000.00	
OceanBase v2.2 Support Service (Partitioning, Horizontal Scalability and Advanced Compression included), per vCPU for 3 years		1	29,250.00	13,440		393,120,000.00
				Sub-Total	360,360,000.00	393,120,000.00
Other Components						
Lenovo ThinkPad X1 Carbon 2019 14" Laptop (includes 2 spares)	20QDA009CD	3	16,999.00	3	50,997.00	
				Sub-Total	50,997.00	

Other Services					
Alibaba Cloud Enterprise Support Service for 3 years 7*24*4hours - ECS support included - Aliyun Linux 2 support included	2	300,000.00	3		900,000.00
OpenResty Professional Edition Support Service for 3 years 7*24*4hrs - Nginx support included	4	6,600.00	192		1,267,200.00
			Sub-Total	0.00	2,167,200.00
Discounts*					
OceanBase v2.2 Enterprise Edition Discount	1	-9,688.80	13,440	-130,217,472.00	
OceanBase v2.2 Partition Discount	1	-2,018.50	13,440	-27,128,640.00	
OceanBase v2.2 Horizontal Discount	1	-2,018.50	13,440	-27,128,640.00	
OceanBase v2.2 Advanced Compression Discount	1	-2,018.50	13,440	-27,128,640.00	
OceanBase v2.2 Support Service (Partitioning, Horizontal Scalability and Advanced Compression included) Discount	1	-17,175.60	13,440		-230,840,064.00
			Sub-Total	-211,603,392.00	-230,840,064.00
			Total	216,005,705.40	164,447,136.00

1 Ant Financial
2 Alibaba Cloud
3 thinkpad.lenovo.com.cn
4 OpenResty Inc.

3-Y Cost of Ownership (CNY):	380,452,842
tpmC	60,880,800
CNY/tpmC	6.25

Audited by Francois Raab and 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.

ANT FINANCIAL	Alibaba Cloud Elastic Compute Service Cluster (with 204 OceanBase Data Nodes)	TPC-C 5.11.0 TPC-Pricing 2.4.0 Report Date October 2, 2019

Numerical Quantities Summary:

MQTh (Maximum Qualified Throughput) **60,880,800 tpmC**

Response Time (sec.)	**Min**	**Average**	**90th**	**Max**
New-Order	0.103	0.114	0.123	5.754
Payment	0.103	0.112	0.121	5.761
Order-Status	0.102	0.106	0.11	5.793
Delivery (Interactive)	0.101	0.101	0.103	3.926
Delivery (Deferred)	0.006	0.019	0.024	2.483
Stock-Level	0.102	0.109	0.115	5.312
Menu	0.101	0.101	0.103	20.547

Emulated Display Delay: 0.1 sec.

Transaction Mix	**Percent**	**Number**
New-Order	44.960%	29,222,784,280
Payment	43.010%	27,955,489,832
Order-Status	4.010%	2,606,386,247
Delivery	4.010%	2,606,365,527
Stock-Level	4.010%	2,606,426,679

Keying Times (sec.)	**Min**	**Average**	**Max**
New-Order	18.002	18.025	22.435
Payment	3.002	3.025	7.465
Order-Status	2.002	2.024	6.447
Delivery	2.002	2.024	6.441
Stock-Level	2.002	2.024	6.451

Think Times (sec.)	**Min**	**Average**	**Max**
New-Order	0.002	12.024	123.886
Payment	0.002	12.024	124.165
Order-Status	0.002	10.024	101.219
Delivery	0.002	5.024	51.263
Stock-Level	0.002	5.024	51.617

Test Duration	
Ramp-up Time	35 min.
Measurement Interval (MI)	480 min.
Checkpoints in MI per Node (Min/Avg/Max)	26/26.6/28
Maximum Checkpoint Interval (All Nodes)	19:17
Number of transactions (all types) completed in Measurement Interval	64,997,452,565

List of Figures

List of Tables

Preface

This report documents the compliance of Ant Financial (Hang Zhou) Network Technology Co., Ltd. TPC Benchmark ^{TM} C testing on Alibaba Cloud Elastic Compute Service Cluster (with 204 OceanBase Data Nodes) running OceanBase v2.2 Enterprise Edition with Partitioning, Horizontal Scalability and Advanced Compression, executing the TPC Benchmark ^{TM} C Standard, Revision 5.11.0.

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

The main body of the document lists each item in Clause 8 of the TPC Benchmark ^{TM} 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.i2.16xlarge, ecs.c5.16xlarge, Alibaba Cloud Support Price, Lenovo ThinkPad X1 Carbon 2019 14" Laptop and OpenResty Support Price.

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

Introduction

The TPC Benchmark ^{TM} 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 ^{TM} C results for the Alibaba Cloud Elastic Compute Service Cluster (with 204 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*
- *Recovery/commit options*
- *Consistency/locking options*
- *Operating system and application configuration parameters*

Appendix C contains parameter settings for the ecs.i2.16xlarge OceanBase Cluster nodes, ecs.c5.16xlarge OceanBase Cloud Platform (OCP) nodes and ecs.c5.16xlarge 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 204 OceanBase Data Nodes) running OceanBase v2.2 Enterprise Edition with Partitioning, Horizontal Scalability and Advanced Compression. The configuration of ecs.i2.16xlarge instances is described below:

	ecs.i2.16xlarge
Processors	Intel Xeon Platinum 8163(Skylake) 2.5GHz
Processors/Cores/Threads	2/32/64 (64 vCPU)
Memory	512 GB
Cloud Disk (for OS)	40GB Ultra Cloud Disk
Local Disk	8 * 1788GB NVMe SSD

Table 1 ecs.i2.16xlarge Configuration

The 207 database nodes consist of 204 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.c5.16xlarge nodes for OCP. Each of the nodes is configured as described below:

	ecs.c5.16xlarge
Processors	Intel Xeon Platinum 8163(Skylake) 2.5GHz
Processors/Cores/Threads	2/32/64 (64 vCPU)
Memory	128 GB
Cloud Disk (for OS)	40GB Ultra Cloud Disk
Cloud Disk (for Data)	1,024GB Standard SSD Cloud Disk

Table 2 ecs.c5.16xlarge instance

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 64 ecs.c5.16xlarge clients. Each client is configured with:

	ecs.c5.16xlarge
Processors	Intel Xeon Platinum 8163(Skylake) 2.5GHz
Processors/Cores/Threads	2/32/64 (64 vCPU)
Memory	128 GB
Cloud Disk (for OS)	40GB Ultra Cloud Disk

Table 3 ecs.c5.16xlarge 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.

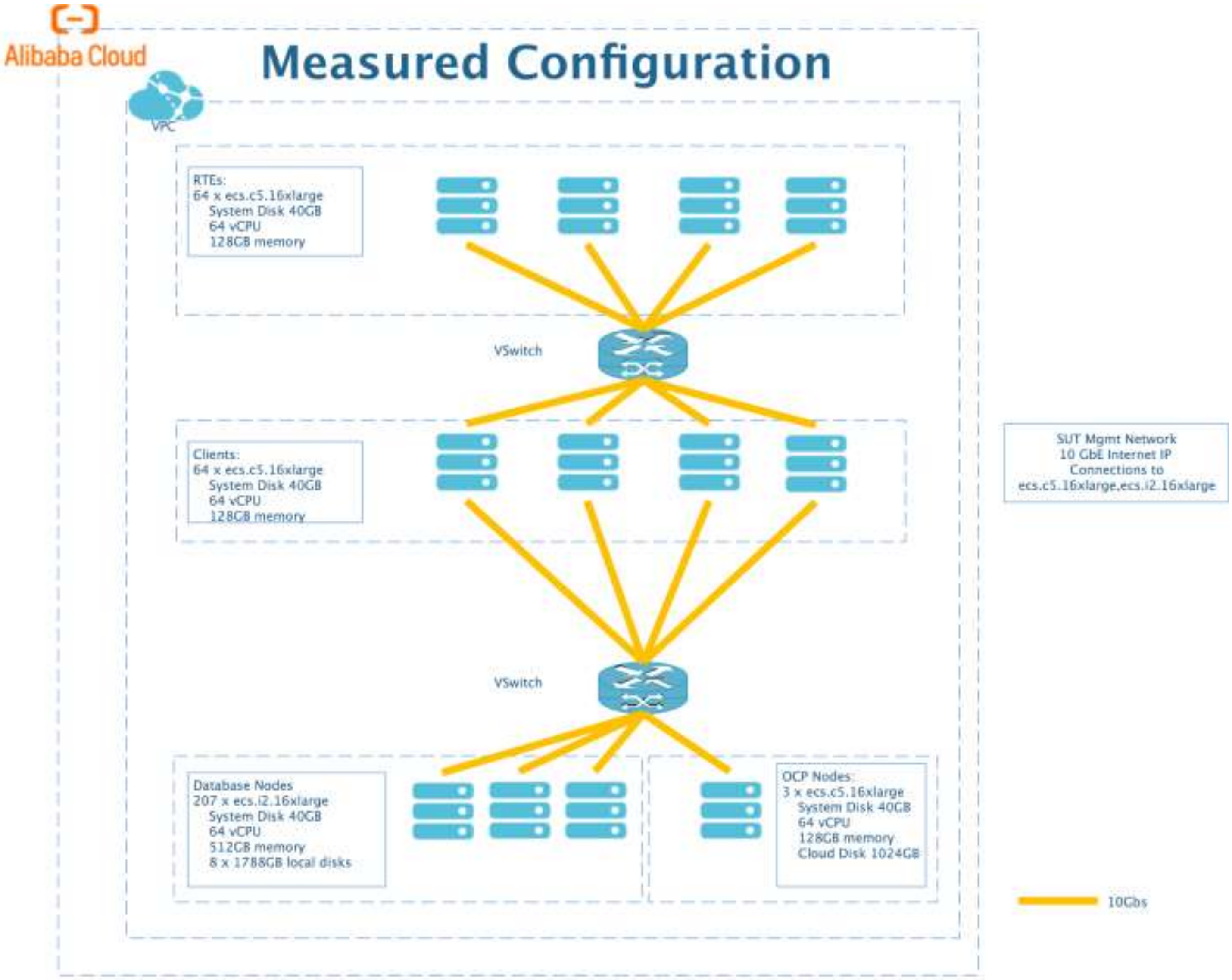


Figure 1 Measured Configuration

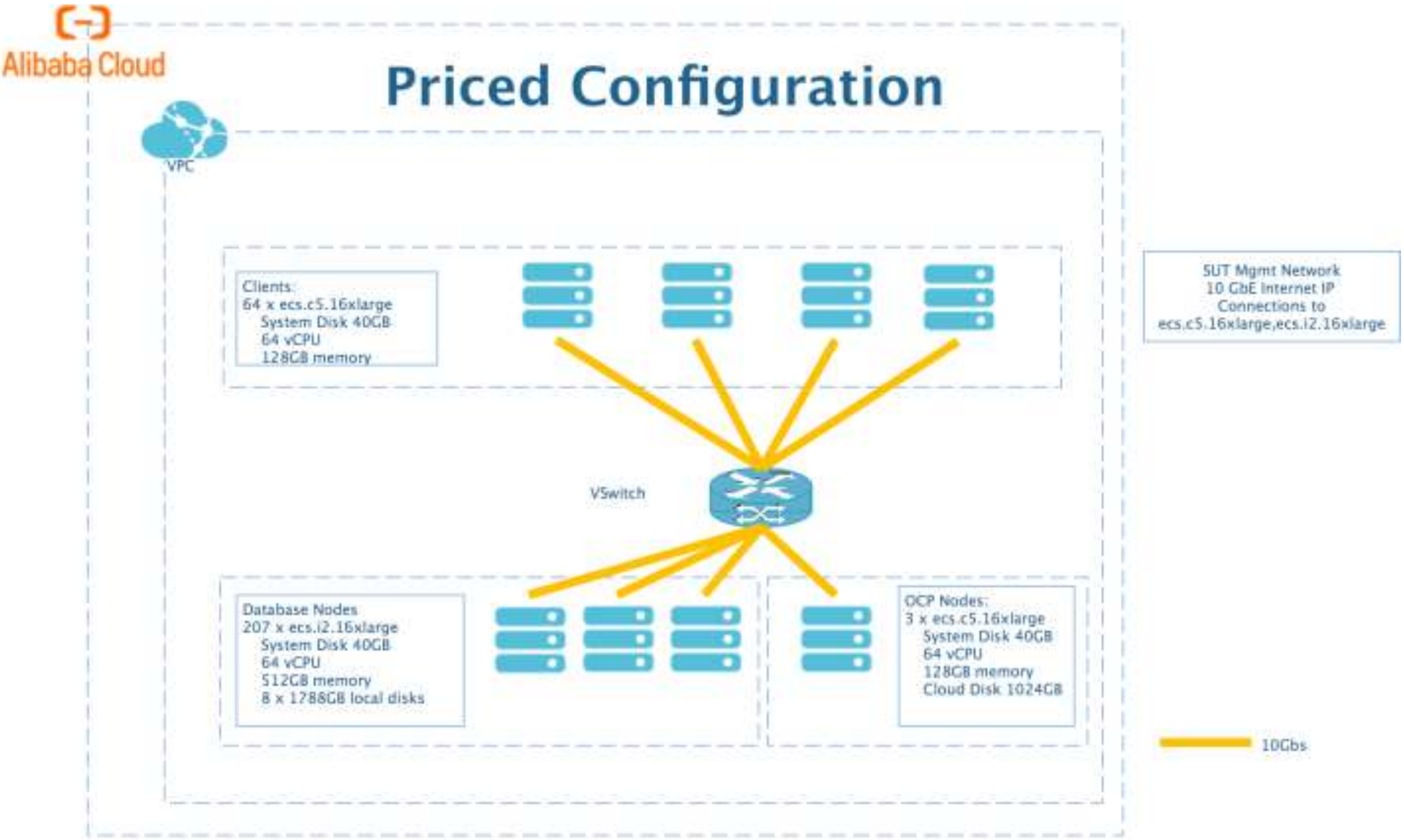


Figure 2 Priced Configuration

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.i2.16xlarge has 8 SSD disks (1,788GB 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 and Stock are 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

Stock	
S_VP1_HP1	S_I_ID
	S_W_ID
	S_ORDER_CNT
	S_YTD
	S_REMOTE_CNT
	S_QUANTITY
------Vertical partition------	
S_VP2_HP1	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

Table 5 Table Stock 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	99.010%
Percentage of Remote order-lines	0.990%
Percentage of Rolled Back Transactions	0.990%
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	59.990%
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 6 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 equals 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 207-node 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 207-node OceanBase Cluster configured for 4,794,240 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 was 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 was 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 10 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 4,794,240 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)	4,794,240	4,794,240
District (DIST)	47,942,400	47,942,400
Customer (CUST)	143,827,200,000	143,827,200,000
History (HIST)	143,827,200,000	143,827,200,000
Order (ORDR)	143,827,200,000	143,827,200,000
New order (NORD)	43,148,160,000	43,148,160,000
Order line (ORDL)	1,438,276,717,027	1,438,276,717,027
Stock (STOK)	479,424,000,000	479,424,000,000
Item (ITEM)	100,000	100,000

Table 7 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 and Stock are 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 (~19 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/1, 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 and Stock are partitioned both horizontally and vertically, and other tables are horizontally partitioned only. The method for horizontal partition is HASH with 8192 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 takes 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_VP1_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	------vertical partition------			W_CITY
	C_MIDDLE	S_VP2_HP1	S_DATA		W_STATE
	C_BALANCE		S_DIST_01		W_ZIP
	C_YTD_PAYMENT		S_DIST_02	District	
	C_PAYMENT_CNT		S_DIST_03	D_HP1	D_W_ID
	C_CREDIT_LIM		S_DIST_04		D_ID
	C_STREET_1		S_DIST_05		D_NEXT_O_ID
	C_STREET_2		S_DIST_06		D_TAX
	C_CITY		S_DIST_07		D_YTD
	C_STATE		S_DIST_08		D_NAME
	C_ZIP		S_DIST_09		D_STREET_1
	C_PHONE		S_DIST_10		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 8 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 9 Database mapping-2

As an example, Partition Group 1 contains following partitions: {C_VP1_HP1, C_VP2_HP1, S_VP1_HP1, S_VP2_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 14,304GB storage capacity, which consists of 8 SSD disks.

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).

Warehouses	4,794,240	**tpmC**	60,880,800	**tpmC/W**	12.70

Table	**Population (row)**	**Data**	**Index**	**5 % Space**	**8-H Space**	**Total Space**
Warehouse (WARE)	4,794,240	32	0	2		34
District (DIST)	47,942,400	32	0	2		34
Customer (CUST)	143,827,200,000	139,000	5,527	7,227		151,754
Item (ITEM)	100,000	2	0	1		3
Stock (STOK)	479,424,000,000	270,404	0	13,521		283,925
New-Order (NORD)	43,148,160,000	319	0	16		335
Order (ORDR)	143,827,200,000	2,540	2,581		516	5,637
Order-Line (ORDL)	1,438,276,717,027	93,371	0		18,971	112,342
History (HIST)	143,827,200,000	7,152	0		1,453	8,605
Totals		512,852	8,108	20,769	20,940	562,669

OceanBase Space Usage	**Data Nodes**	**Allocated**	**Free Space**	**Growth**
After DB Load	204	566,712	45,751	-
After 8-hour Perf. Run	204	566,712	11,677	34,074

Free space	45,752
Dynamic Space	103,063
Static Space	417,897
Daily Growth	20,940
Daily Spread	14,342
60-Day Space	2,534,807
8-Hour Checkpoint	71,792

	Data Nodes	**Storage/Node**	**Total**	**Needed**	**Margin**
Server Storage	204	12,975	2,646,900	2,606,600	40,300

Table 10 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 60,880,800.

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.

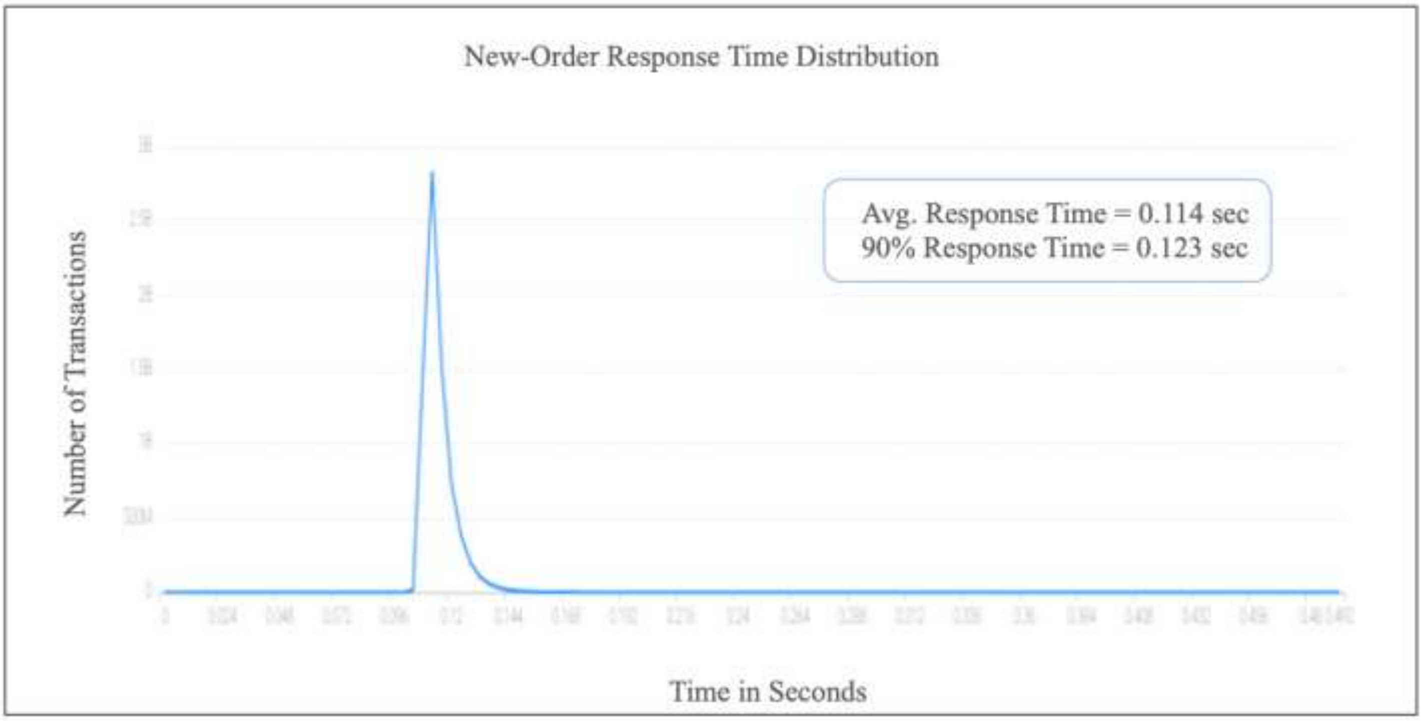


Figure 3 Frequency distribution of response times for New-Order

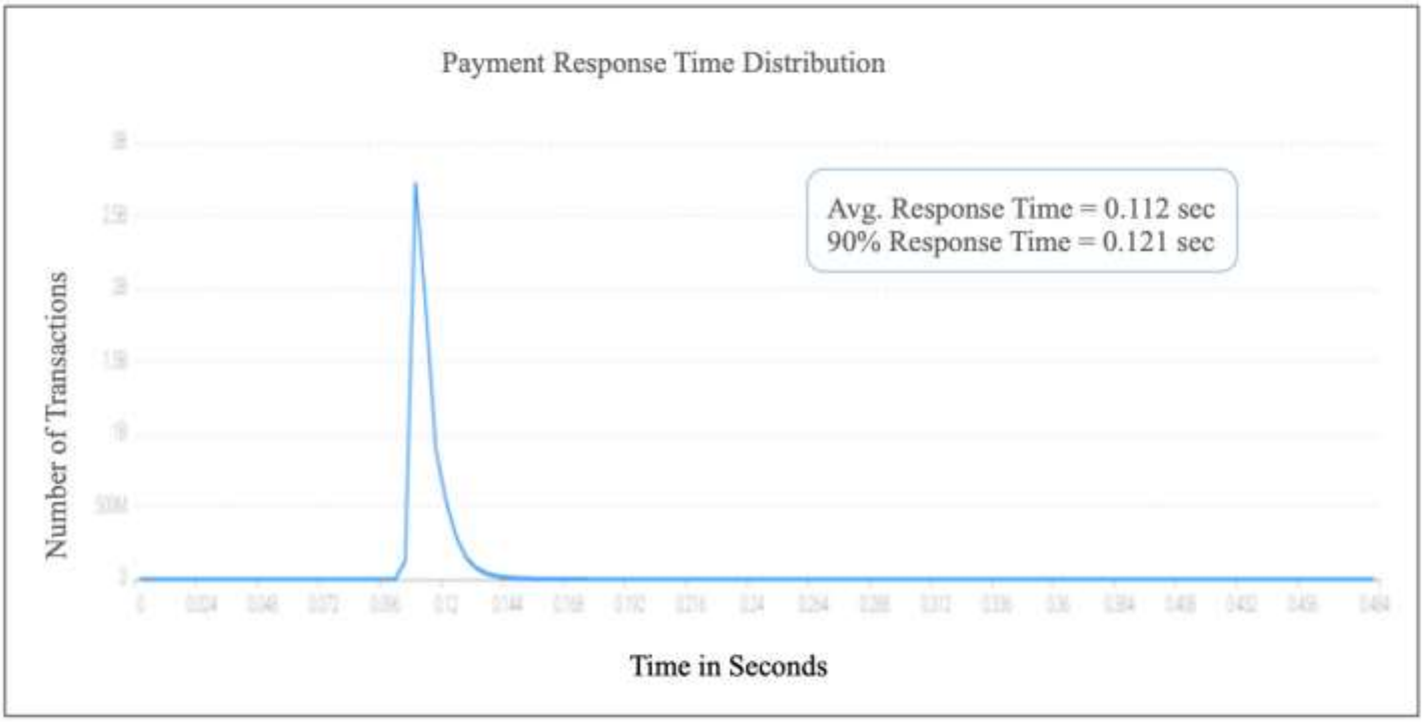


Figure 4 Frequency distribution of response times for Payment

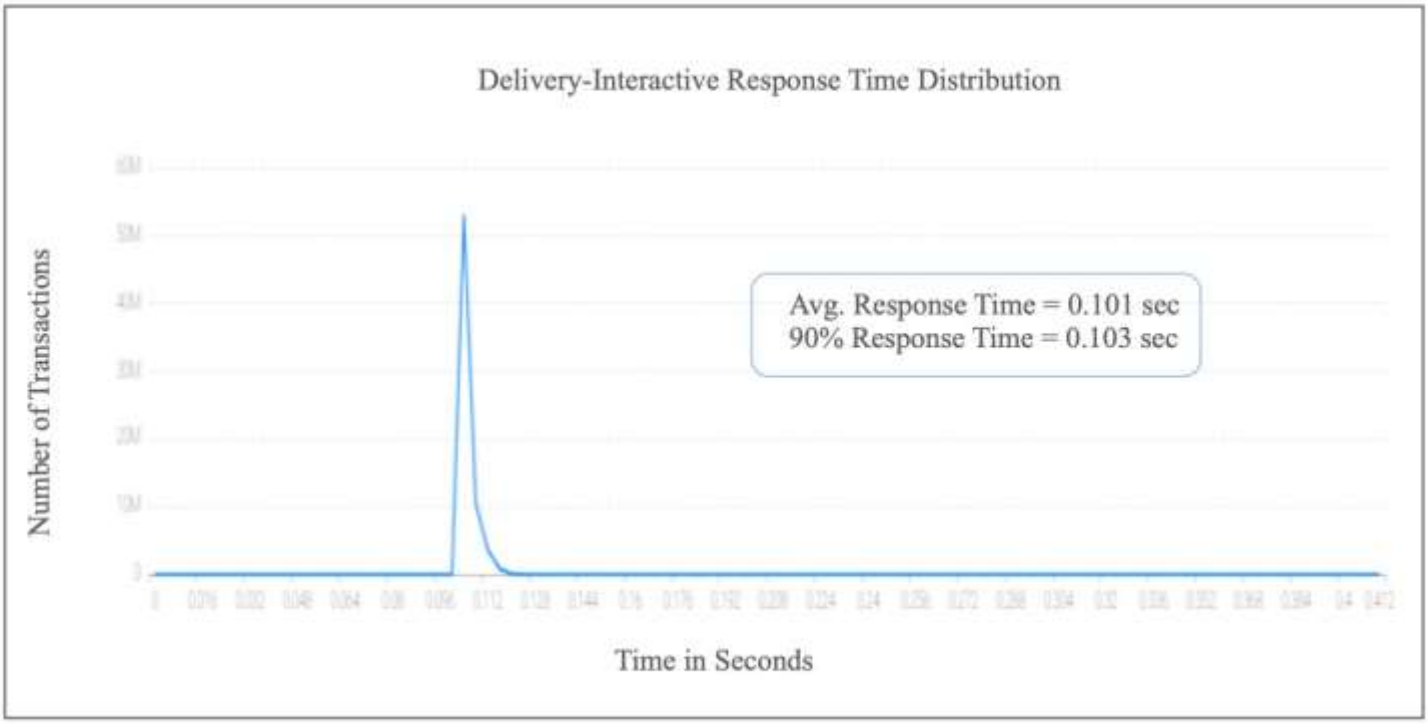


Figure 5 Frequency distribution of response times for Delivery (interactive)

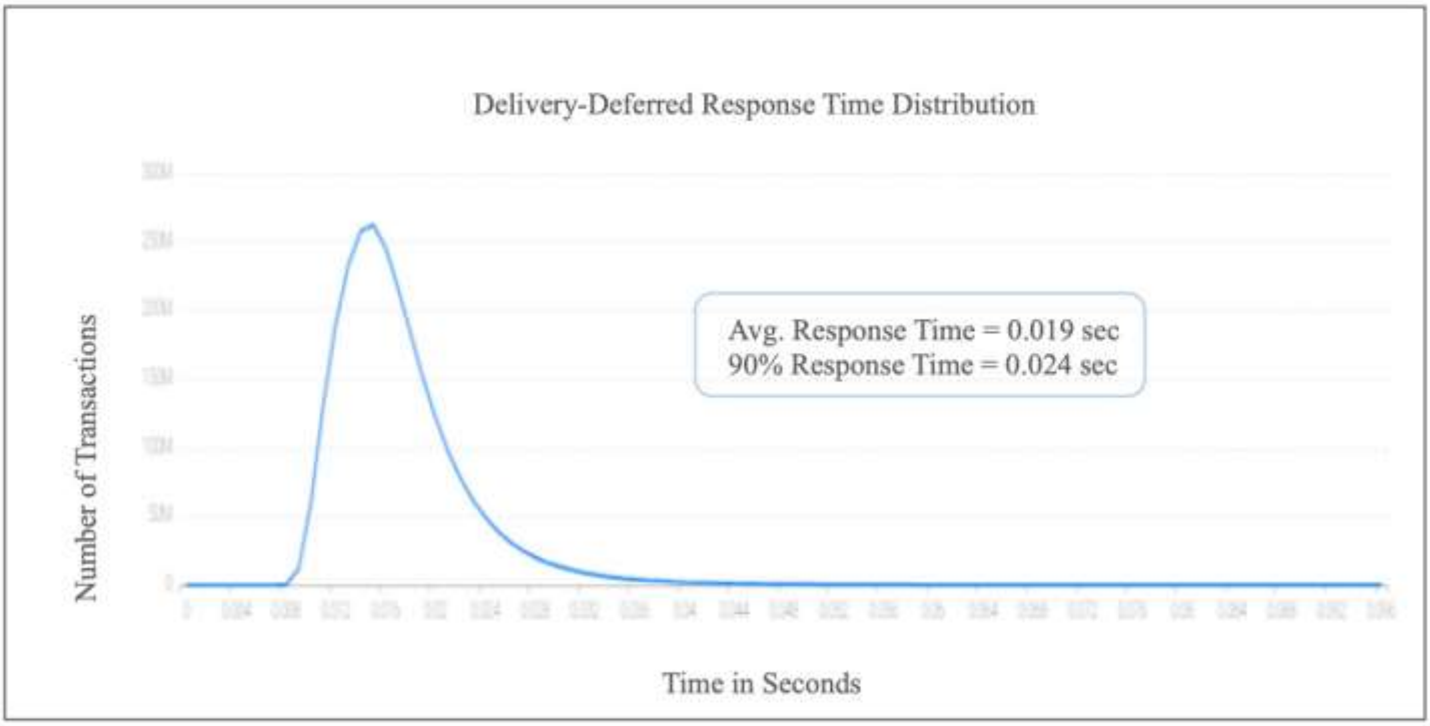


Figure 6 Frequency distribution of response times for Delivery (deferred)

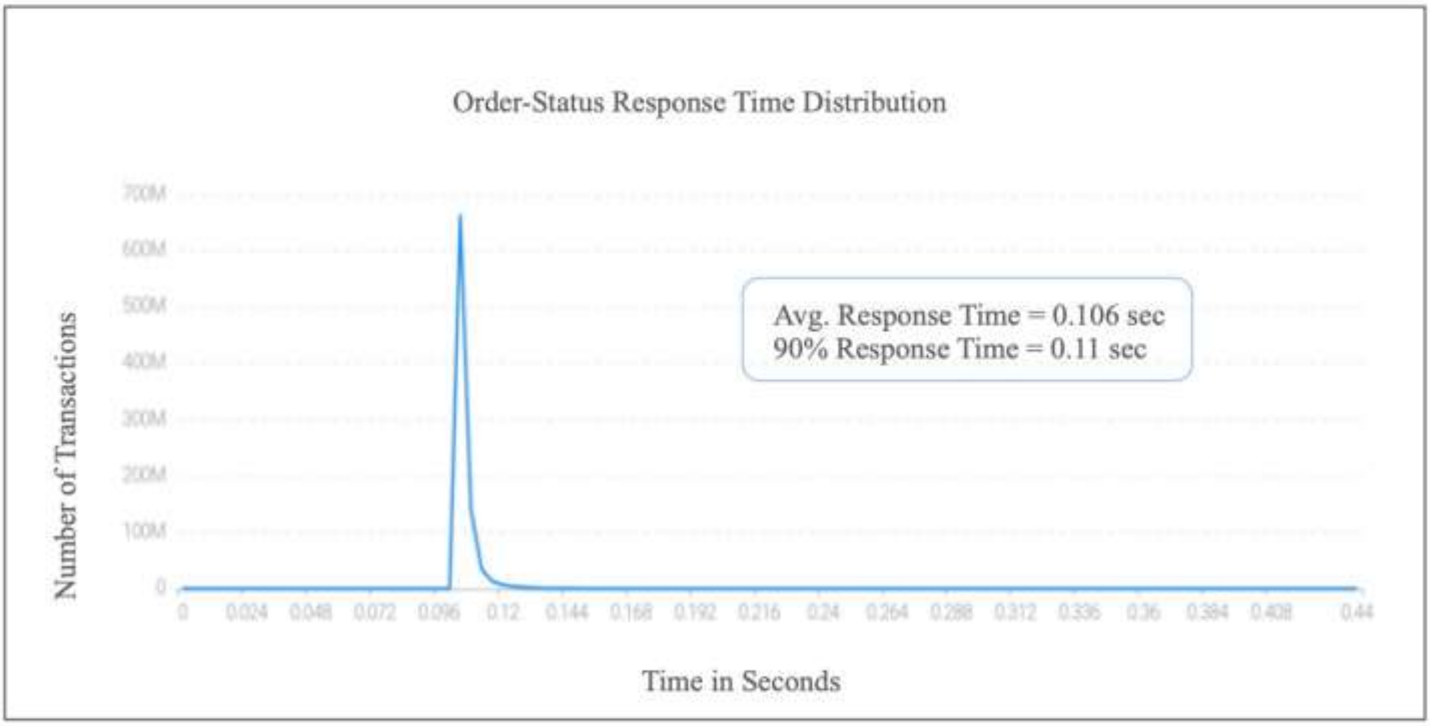


Figure 7 Frequency distribution of response times for Order-Status

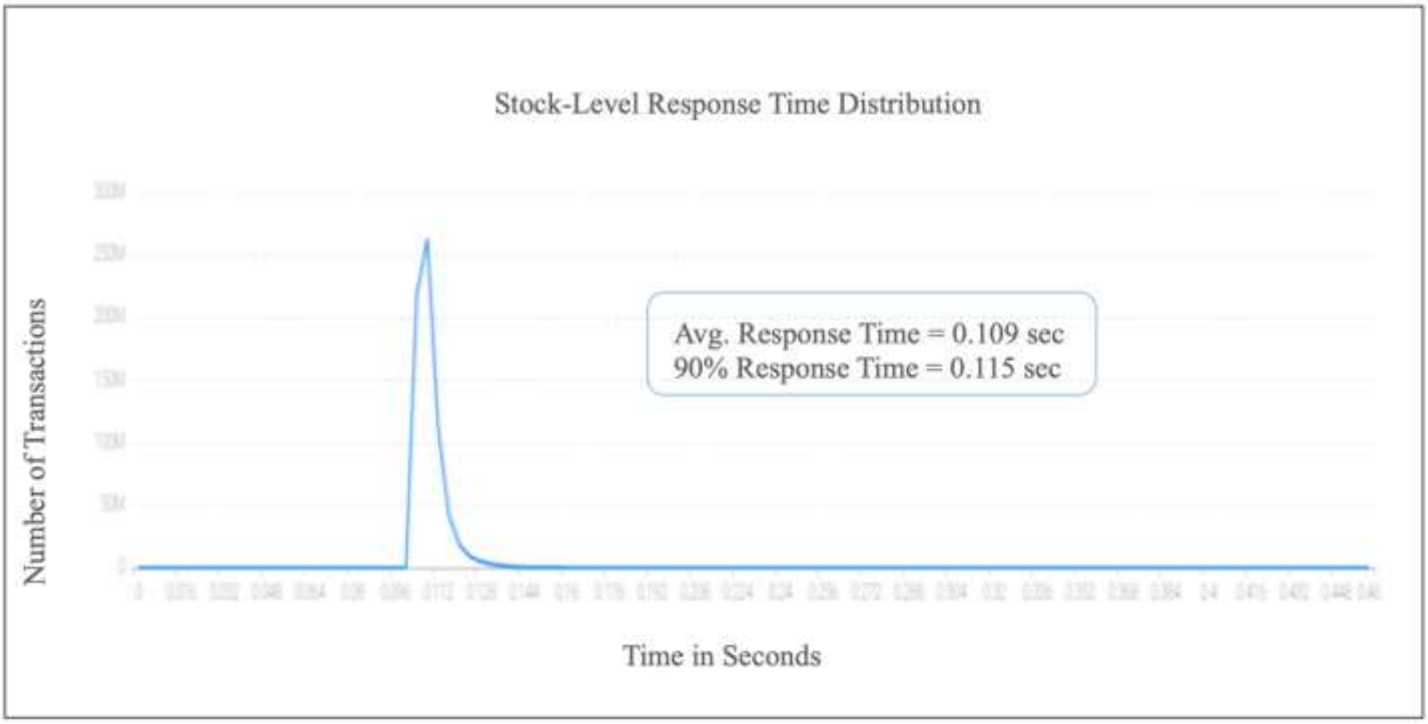


Figure 8 Frequency distribution of response times for Stock-Level

5.5 Think Time Frequency Distribution

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

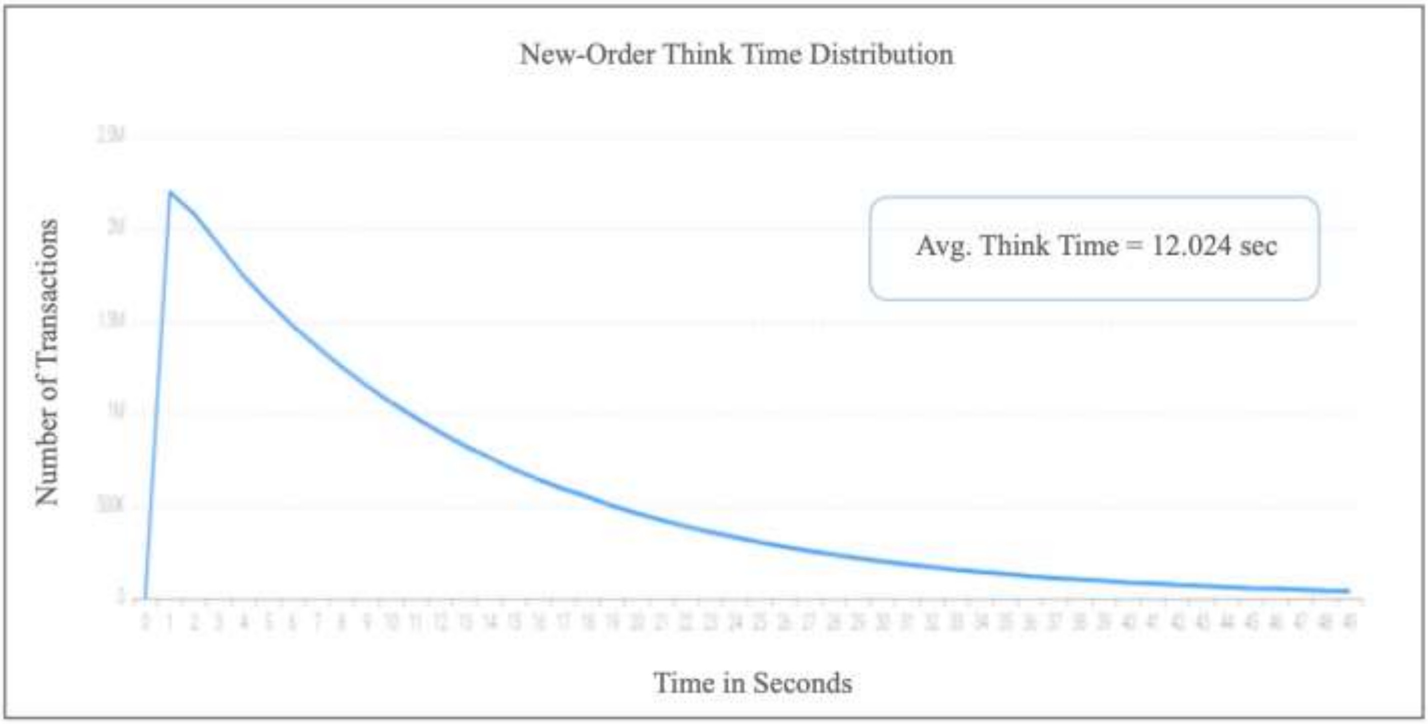


Figure 9 Frequency distribution of think times for New-Order

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.

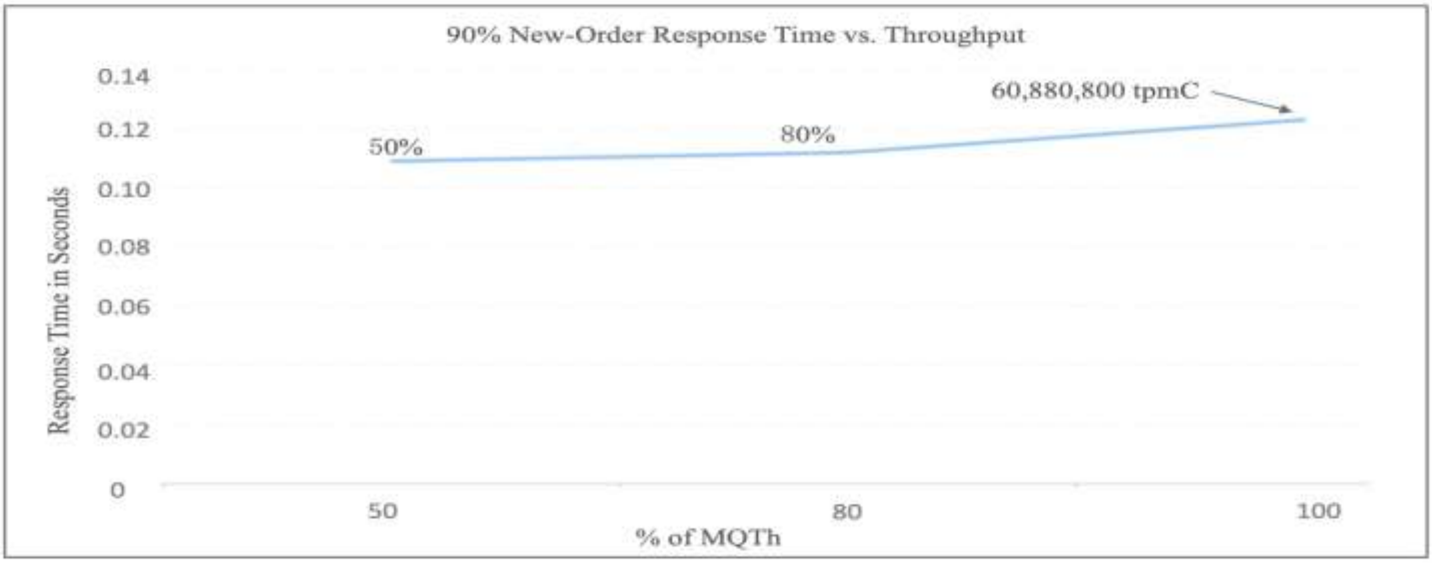


Figure 10 New-Order Response Time versus Throughput

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.

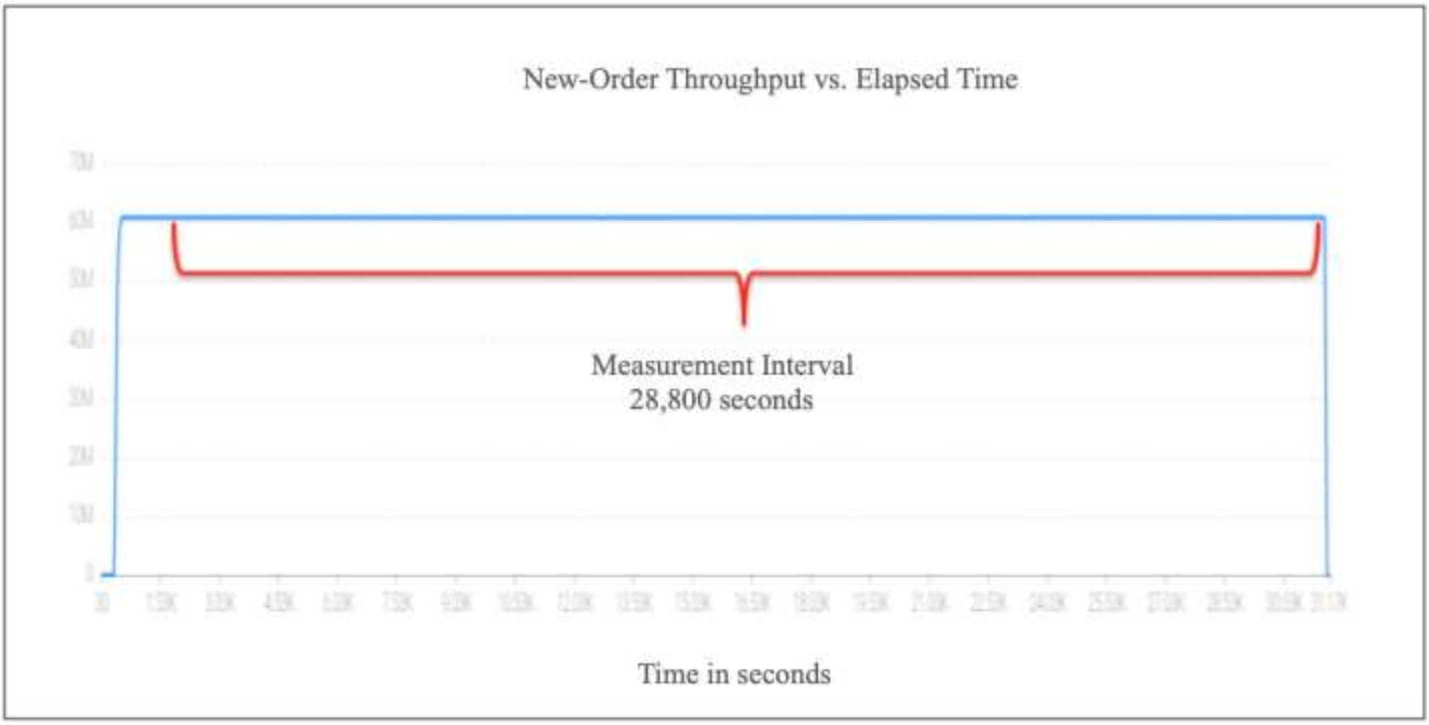


Figure 11 New-Order throughput versus Time

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 (~19 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 30, interval between two checkpoints was about 19 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 19 minutes 17 seconds and the longest checkpoint duration is 7 minutes 8 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	2,100 seconds
Ramp-down Duration	150 seconds
Measurement Interval	28,800 seconds
Database Scale	4,794,240 warehouses
Total terminals	47,942,400
Terminals/Driver	49,940
Number of RTEs nodes/instances	64/960

Table 11 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.i2.16xlarge OceanBase Cluster data nodes, ecs.c5.16xlarge OceanBase Cloud Platform (OCP) nodes and connecting the ecs.c5.16xlarge client nodes. The network between the ecs.c5.16xlarge 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 October 2, 2019.

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 204 OceanBase Data Nodes)	60,880,800	CNY380, 452,842	6.25	October 2, 2019

Table 12 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 this system under test are orderable as of the date of publishing.

8 Clause 8: Auditor Attestation

8.1 Auditor Information

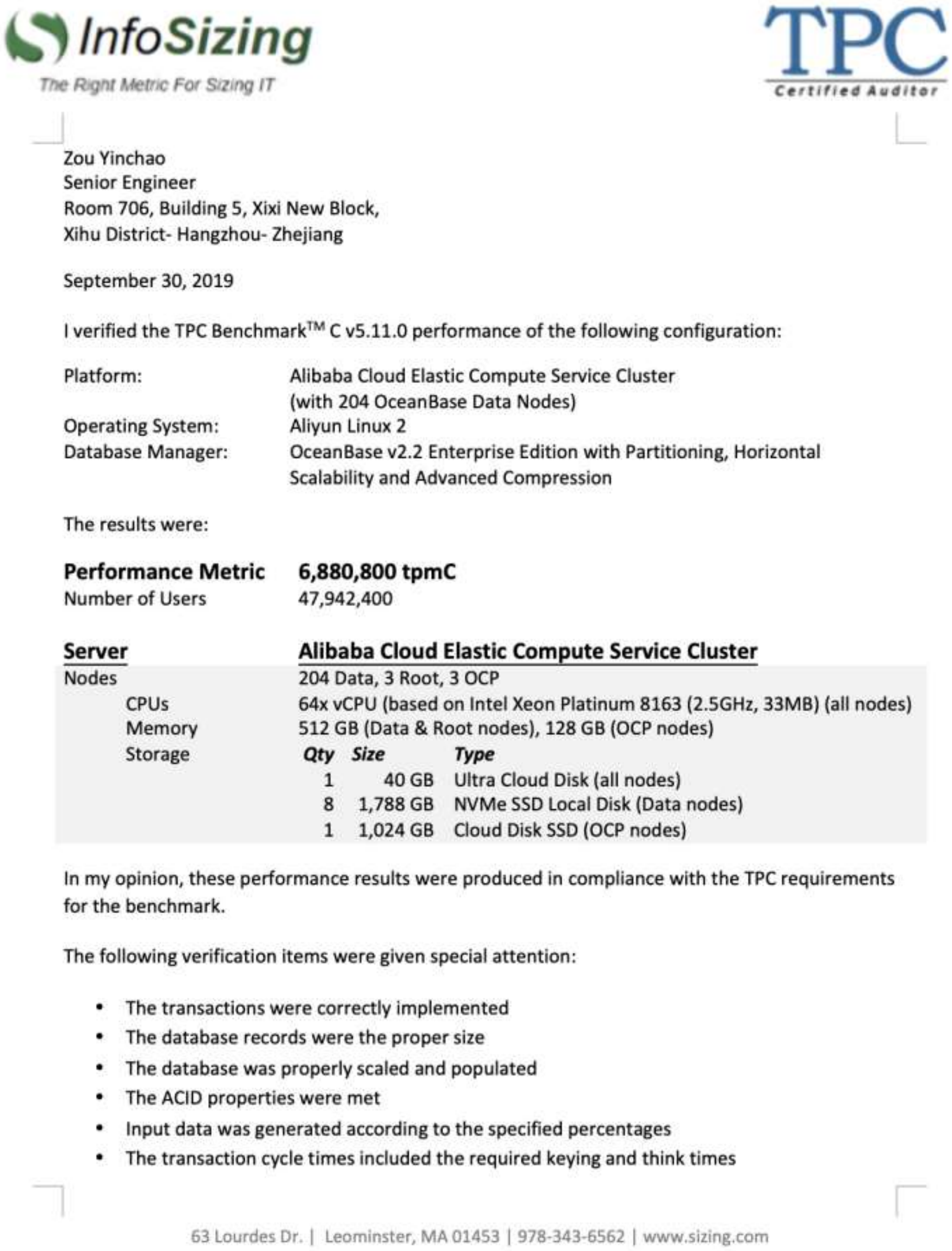
The auditor's name, address, phone number, and a copy of the auditor's attestation letter indicating compliance must be included in the Full Disclosure Report

This benchmark was audited by:

InfoSizing
Francois Raab and 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.



InfoSizing
The Right Metric For Sizing IT

TPC
Certified Auditor

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

September 30, 2019

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

Platform:	Alibaba Cloud Elastic Compute Service Cluster (with 204 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:

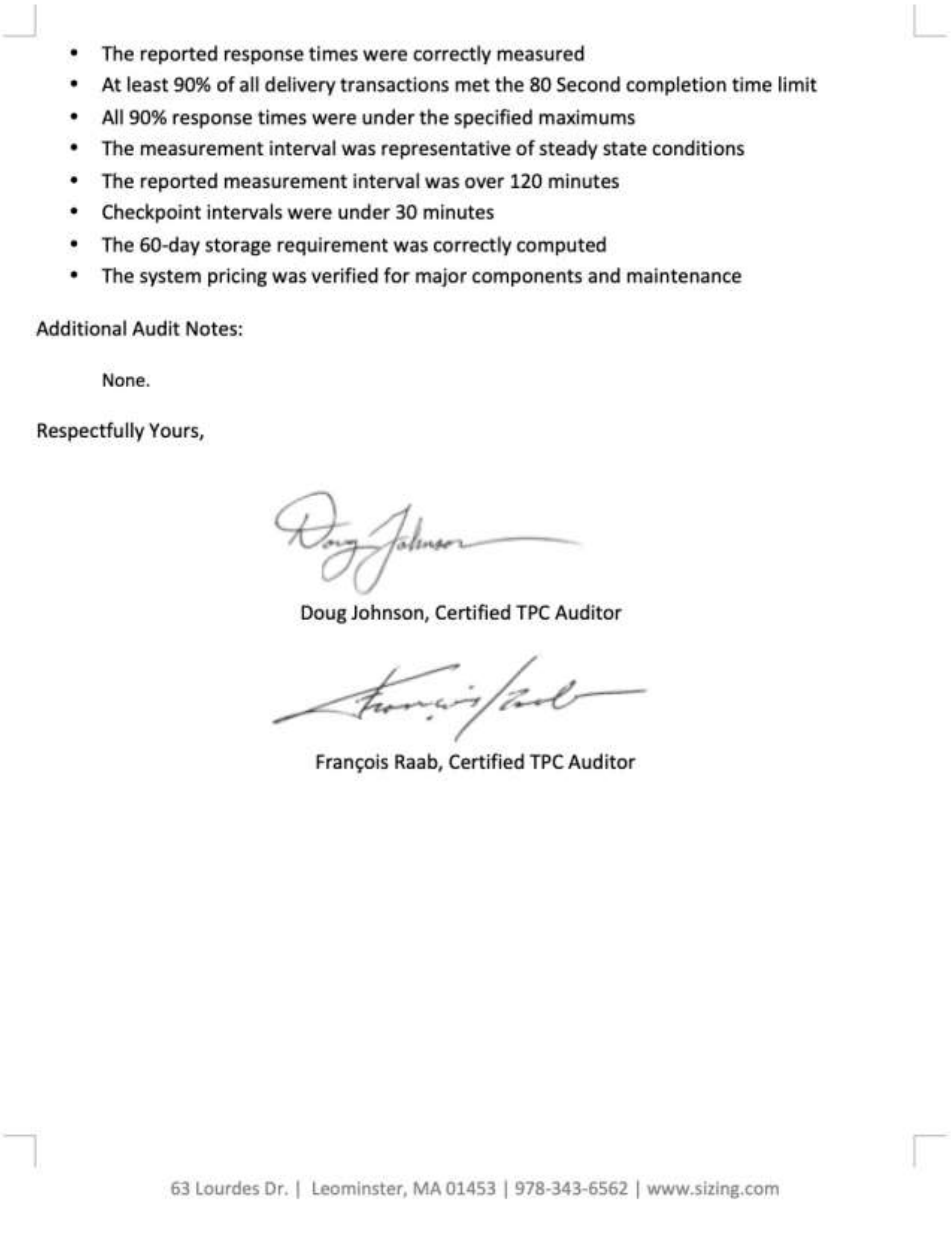
Performance Metric	**6,880,800 tpmC**
Number of Users	47,942,400

Server	**Alibaba Cloud Elastic Compute Service Cluster**		
Nodes	204 Data, 3 Root, 3 OCP		
CPUs	64x vCPU (based on Intel Xeon Platinum 8163 (2.5GHz, 33MB) (all nodes)		
Memory	512 GB (Data & Root nodes), 128 GB (OCP nodes)		
Storage	***Qty***	***Size***	***Type***
	1	40 GB	Ultra Cloud Disk (all nodes)
	8	1,788 GB	NVMe SSD Local Disk (Data nodes)
	1	1,024 GB	Cloud Disk SSD (OCP 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,

Doug Johnson

Doug Johnson, Certified TPC Auditor

François Raab

François Raab, 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-190920
/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

create-procedure.sql

/aci
/acid-src-190920
- - 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 50, max_memory 429496729600, max_iops 128,
max_disk_size '2778G', max_session_num 64, MIN_CPU 50, MIN_MEMORY=429496729600,
MIN_IOPS=128;
create resource pool pool_for_ob6 unit = 'unit_for_ob6', unit_num = 68, zone_list =
('zone1','zone2','zone3');
create resource unit sys_unit_for_ob6 max_cpu 15, max_memory 53687091200, max_iops 128,
max_disk_size '2778G', max_session_num 64, MIN_CPU 15, MIN_MEMORY=53687091200,
MIN_IOPS=128;
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-table-3zone.sql

```
set autocommit=1;
set ob_query_timeout=4000000000;
set ob_trx_timeout=4000000000;
create table load_proc(l_part int, w_id int, primary key(l_part, w_id));
create tablegroup tpcc_group binding true partition by hash partitions 8192;
create table ware(w_id int
   ,   w_ytd decimal(12,2)
   ,   w_tax decimal(4,4)
   ,   w_name varchar(10)
   ,   w_street_1 varchar(20)
   ,   w_street_2 varchar(20)
   ,   w_city varchar(20)
   ,   w_state char(2)
   ,   w_zip char(9)
   , primary key(w_id))tablegroup tpcc_group NOCOMPRESS pctfree=0 BLOCK_SIZE=8192 partition
by hash(w_id) partitions 8192;

create table dist (d_w_id int
      , d_id int
      , d_next_o_id int
      , d_tax decimal(4,4)
      , d_ytd decimal(12,2)
      , d_name varchar(10)
      , d_street_1 varchar(20)
      , d_street_2 varchar(20)
      , d_city varchar(20)
      , d_state char(2)
      , d_zip char(9)
       , primary key(d_w_id, d_id))tablegroup tpcc_group NOCOMPRESS pctfree=0 BLOCK_SIZE=8192
partition by hash(d_w_id) partitions 8192;

create table cust (c_w_id int
      ,   c_d_id int
      ,   c_id int
      ,   c_discount decimal(4, 4)
      ,   c_credit char(2)
      ,   c_last varchar(16)
      ,   c_first varchar(16)
      ,   c_middle char(2)
       ,   c_balance decimal(12, 2)
      ,   c_ytd_payment decimal(12, 2)
      ,   c_payment_cnt int
      ,   c_credit_lim decimal(12, 2)
      ,   c_street_1 varchar(20)
      ,   c_street_2 varchar(20)
      ,   c_city varchar(20)
      ,   c_state char(2)
      ,   c_zip char(9)
      ,   c_phone char(16)
      ,   c_since date
      ,   c_delivery_cnt int
      ,   c_data varchar(500)
      , index icust(c_last, c_d_id, c_w_id, c_first, c_id) local BLOCK_SIZE=8192
      ,   primary key (c_w_id, c_d_id, c_id) )tablegroup tpcc_group COMPRESS FOR QUERY pctfree=0
BLOCK_SIZE=8192 USE_BLOOM_FILTER = TRUE partition by hash(c_w_id) partitions 8192 partition
by column((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));

create table hist (h_c_id int
      , h_c_d_id int
      , h_c_w_id int
      , h_d_id int
      , h_w_id int
      , h_date date
      , h_amount decimal(6, 2)
      , h_data varchar(24))tablegroup tpcc_group COMPRESS FOR QUERY pctfree=0
BLOCK_SIZE=8192 partition by hash(h_w_id) partitions 8192;

create table nord (no_w_id int
      ,   no_d_id int
```

```
    ,  no_o_id int
    ,  primary key ( no_w_id, no_d_id, no_o_id ))tablegroup tpcc_group TABLE_MODE =
'QUEUING' COMPRESS FOR QUERY pctfree=0 BLOCK_SIZE=8192 progressive_merge_num=1
USE_BLOOM_FILTER = TRUE partition by hash(no_w_id) partitions 8192;

create table ordr (o_w_id int
    , o_d_id int,
    , o_id int
    , o_c_id int
    , o_carrier_id int
    , o_ol_cnt int
    , o_all_local int
    , o_entry_d date
    , index iordr(o_w_id, o_d_id, o_c_id, o_id) storing (o_entry_d,o_carrier_id,o_ol_cnt ) local
BLOCK_SIZE=8192
     , primary key ( o_w_id, o_d_id, o_id ))tablegroup tpcc_group COMPRESS FOR QUERY pctfree=0
BLOCK_SIZE=8192 progressive_merge_num=1 USE_BLOOM_FILTER = TRUE partition by
hash(o_w_id) partitions 8192;

create table ordl (ol_w_id int
    , ol_d_id int
    , ol_o_id int
    , ol_number int
    , ol_delivery_d date
    , ol_amount decimal(6, 2)
    , ol_i_id int
    , ol_supply_w_id int
    , ol_quantity int
    , ol_dist_info char(24)
    , primary key (ol_w_id, ol_d_id, ol_o_id, ol_number ))tablegroup tpcc_group COMPRESS FOR
QUERY pctfree=0 BLOCK_SIZE=8192 partition by hash(ol_w_id) partitions 8192;

create table item (i_id int
    ,  i_name varchar(24)
    ,  i_price decimal(5,2)
    ,  i_data varchar(50)
    ,  i_im_id int
  ,  primary key(i_id)) COMPRESS FOR QUERY pctfree=0 BLOCK_SIZE=16384
duplicate_scope='cluster' locality='F,R{all_server}@zone1, F,R{all_server}@zone2,
F,R{all_server}@zone3' primary_zone='zone1';

create table stok (s_i_id int
    ,  s_w_id int
    ,  s_order_cnt int
    ,  s_ytd decimal(8)
    ,  s_remote_cnt int
    ,  s_quantity int
    ,  s_data varchar(50)
    ,  s_dist_01 char(24)
    ,  s_dist_02 char(24)
    ,  s_dist_03 char(24)
    ,  s_dist_04 char(24)
    ,  s_dist_05 char(24)
    ,  s_dist_06 char(24)
    ,  s_dist_07 char(24)
    ,  s_dist_08 char(24)
    ,  s_dist_09 char(24)
    ,  s_dist_10 char(24)
    ,  primary key (s_i_id, s_w_id))tablegroup tpcc_group COMPRESS FOR QUERY pctfree=0
BLOCK_SIZE=8192 USE_BLOOM_FILTER = TRUE partition by hash(s_w_id) partitions 8192 partition
by column((s_i_id, s_w_id, s_order_cnt, s_ytd, s_remote_cnt, s_quantity));

create table load_hist(
        total_w_cnt int,
        min_w_id int,
        max_w_id int,
        thread_num int,
        dsn varchar(64),
        load_begin varchar(30),
        load_end varchar(30),
        seed_val int,
        c_val int,
        mem_ratio int,
        status int);

SET global NLS_DATE_FORMAT='YYYY-MM-DD HH24:MI:SS';
SET global NLS_TIMESTAMP_FORMAT='YYYY-MM-DD HH24:MI:SS.FF';
```

RTE parameters

```
ramp_up = 2,100
mi = 28,800
ramp_down = 150
warehouse_num = 4,794,240
report_interval = 30
```

Appendix C: Configuration Options

Database Node

File system info

```
Filesystem                      1K-blocks        Used    Available Use% Mounted on
devtmpfs                        264048996           0    264048996   0% /dev
tmpfs                           264059968           0    264059968   0% /dev/shm
tmpfs                           264059968         816    264059152   1% /run
tmpfs                           264059968           0    264059968   0% /sys/fs/cgroup
/dev/vda1                        41147472     7856500     31387480  21% /
/dev/mapper/ob_vg-ob_log       1055840692   790468360    211668860  79% /data/log1
/dev/mapper/ob_vg-ob_data     13606147200  3306204044   9614159708  26% /data/1
/dev/mapper/ob_vg-ob_home       205375464     4710856    190162464   3% /home
tmpfs                            52811996           0     52811996   0% /run/user/1002
```

/etc/security/limits.conf

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

Observer.cfg

```
__easy_memory_limit=30G
migrate_concurrency=10
__schema_split_mode=True
micro_block_merge_verify_level=0
major_compact_trigger=500
_follower_replica_merge_level=0
_minor_compaction_interval=10m
minor_compact_trigger=5
minor_freeze_times=500
merge_stat_sampling_ratio=1
minor_merge_concurrency=30
merge_thread_count=30
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=172.19.252.161:61111,172.19.252.162:61113,172.19.252.163:61115,172.19.252.100:6111
7,172.19.252.101:61119,172.19.252.103:61121,172.19.252.104:61123,172.19.252.105:61125,172.19.252
.106:61127,172.19.252.107:61129,172.19.252.109:61131,172.19.252.110:61133,172.19.252.111:61135,1
72.19.252.112:61137,172.19.252.113:61139,172.19.252.114:61141,172.19.252.115:61143,172.19.252.11
7:61145,172.19.252.118:61147,172.19.252.119:61149,172.19.252.120:61151,172.19.252.121:61153,172.
19.252.122:61155,172.19.252.124:61157,172.19.252.125:61159,172.19.252.126:61161,172.19.252.127:6
1163,172.19.252.128:61165,172.19.252.129:61167,172.19.252.130:61169,172.19.252.131:61171,172.19.
252.132:61173,172.19.252.133:61175,172.19.252.134:61177,172.19.252.135:61179,172.19.252.136:611
81,172.19.252.137:61183,172.19.252.138:61185,172.19.252.139:61187,172.19.252.140:61189,172.19.25
2.141:61191,172.19.252.142:61193,172.19.252.143:61195,172.19.252.144:61197,172.19.252.146:61199,
172.19.252.147:61201,172.19.252.148:61203,172.19.252.149:61205,172.19.252.150:61207,172.19.252.
151:61209,172.19.252.102:61211,172.19.252.153:61213,172.19.252.154:61215,172.19.252.69:61217,17
2.19.252.156:61219,172.19.252.157:61221,172.19.252.158:61223,172.19.252.159:61225,172.19.252.16
0:61227,172.19.252.165:61229,172.19.252.166:61231,172.19.252.167:61233,172.19.252.168:61235,172
.19.252.169:61237,172.19.252.170:61239,172.19.252.171:61241,172.19.252.172:61243,172.19.252.174:
61245,172.19.252.175:61247,172.19.252.176:61249,172.19.252.177:61251,172.19.252.152:61253,172.1
9.252.116:61255,172.19.252.180:61257,172.19.252.181:61259,172.19.252.182:61261,172.19.252.183:6
1263,172.19.252.184:61265,172.19.252.185:61267,172.19.252.186:61269,172.19.252.187:61271,172.19
.252.188:61273,172.19.252.189:61275,172.19.252.190:61277,172.19.252.191:61279,172.19.252.192:61
281,172.19.252.193:61283,172.19.252.194:61285,172.19.252.195:61287,172.19.252.196:61289,172.19.
252.197:61291,172.19.252.198:61293,172.19.252.199:61295,172.19.252.200:61297,172.19.252.201:612
99,172.19.252.202:61301,172.19.252.203:61303,172.19.252.204:61305,172.19.252.205:61307,172.19.2
52.206:61309,172.19.252.207:61311,172.19.252.208:61313,172.19.252.209:61315,172.19.252.210:6131
7,172.19.252.211:61319,172.19.252.212:61321,172.19.252.213:61323,172.19.252.214:61325,172.19.25
2.215:61327,172.19.252.216:61329,172.19.252.217:61331,172.19.252.218:61333,172.19.252.219:61335
,172.19.252.220:61337,172.19.252.221:61339,172.19.252.222:61341,172.19.252.223:61343,172.19.252.
224:61345,172.19.252.225:61347,172.19.252.226:61349,172.19.252.227:61351,172.19.252.48:61353,17
2.19.252.49:61355,172.19.252.50:61357,172.19.252.51:61359,172.19.252.52:61361,172.19.252.53:6136
3,172.19.252.54:61365,172.19.252.55:61367,172.19.252.56:61369,172.19.252.57:61371,172.19.252.58:
61373,172.19.252.179:61375,172.19.252.60:61377,172.19.252.61:61379,172.19.252.62:61381,172.19.2
52.63:61383,172.19.252.64:61385,172.19.252.65:61387,172.19.252.66:61389,172.19.252.67:61391,172.
19.252.68:61393,172.19.252.70:61395,172.19.252.71:61397,172.19.252.72:61399,172.19.252.73:61401,
172.19.252.74:61403,172.19.252.75:61405,172.19.252.76:61407,172.19.252.77:61409,172.19.252.78:61
411,172.19.252.79:61413,172.19.252.80:61415,172.19.252.81:61417,172.19.252.82:61419,172.19.252.8
3:61421,172.19.252.84:61423,172.19.252.85:61425,172.19.252.86:61427,172.19.252.87:61429,172.19.2
52.88:61431,172.19.252.89:61433,172.19.252.90:61435,172.19.252.91:61437,172.19.252.92:61439,172.
19.252.93:61441,172.19.252.94:61443,172.19.252.95:61445,172.19.252.96:61447,172.19.252.97:61449,
172.19.252.98:61451,172.19.252.99:61453,172.19.254.0:61455,172.19.254.1:61457,172.19.254.10:6145
9,172.19.254.11:61461,172.19.254.12:61463,172.19.254.13:61465,172.19.254.14:61467,172.19.254.15:
61469,172.19.254.16:61471,172.19.254.17:61473,172.19.254.18:61475,172.19.254.19:61477,172.19.25
4.2:61479,172.19.254.20:61481,172.19.254.21:61483,172.19.254.22:61485,172.19.254.23:61487,172.19
.254.24:61489,172.19.254.25:61491,172.19.254.26:61493,172.19.254.27:61495,172.19.254.28:61497,17
2.19.254.3:61499,172.19.254.4:61501,172.19.254.5:61503,172.19.254.6:61505,172.19.254.7:61507,172.
19.254.8:61509,172.19.252.173:61511,172.19.252.145:61513,172.19.254.29:61515,172.19.254.9:61517,
172.19.252.155:61519,172.19.252.123:61521,172.19.252.164:61523
location_cache_refresh_min_interval=1s
location_refresh_thread_count=2
virtual_table_location_cache_expire_time=100h
location_cache_expire_time=100h
_ob_get_gts_ahead_interval=1ms
_ob_btree_retire_limit=1024
_ob_enable_log_hot_cache=True
```

```
_ob_enable_log_batch_rpc=True
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=15:30
__min_full_resource_pool_memory=1073741824
server_data_copy_out_concurrency=10
server_data_copy_in_concurrency=10
enable_rebalance=False
resource_soft_limit=100
min_observer_version=2.2.0
replica_safe_remove_time=5m
server_permanent_offline_time=48h
merger_warm_up_duration_time=0s
write_throttling_trigger_percentage=80
freeze_trigger_percentage=30
memstore_limit_percentage=80
cpu_quota_concurrency=4
workers_per_cpu_quota=10
memory_chunk_cache_size=400G
memory_limit_percentage=85
enable_syslog_recycle=True
enable_syslog_wf=False
max_syslog_file_count=100
syslog_level=WARN
obconfig_url=http://172.19.253.4:8080/oceanbase_obconfig/ob1.tpcc.172.19.253.68.rslist
cluster_id=11
cluster=ob1.tpcc
rootservice_list=172.19.252.161:61111:61112;172.19.252.162:61113:61114;172.19.252.163:61115:61116
weak_read_version_refresh_interval=0
schema_history_expire_time=1d
enable_separate_sys_clog=False
enable_perf_event=True
enable_record_trace_id=False
enable_sql_audit=False
enable_record_trace_log=False
trace_log_slow_query_watermark=10s
system_memory=30G
memory_limit=480GB
__inet_thread_count=
net_thread_count=6
zone=zone1
devname=eth0
mysql_port=61118
rpc_port=61117
_ob_checkpoint_space_limit=375GB
datafile_disk_percentage=50
datafile_size=2778GB
data_dir=/home/tpcc/ob1.obs3/store
```

Client Node

File system info

```
Filesystem      1K-blocks     Used Available Use% Mounted on
devtmpfs         65868324        0  65868324   0% /dev
tmpfs            65879300        0  65879300   0% /dev/shm
tmpfs            65879300     1308  65877992   1% /run
tmpfs            65879300        0  65879300   0% /sys/fs/cgroup
/dev/vda1        41147472  6252896  32991084  16% /
tmpfs            13175860        0  13175860   0% /run/user/0
tmpfs            13175860        0  13175860   0% /run/user/1002
```

/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
```

Nginx.conf

```
#daemon off;
#user  nobody;
env ODBC_LOG_ENABLE=false;
env ODBC_LOG_LEVEL=3;
env ODBCSYSINI=/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/:/home/tpcc/ob1.c0/tpcc/odbc;
worker_processes  1;
worker_rlimit_nofile 100000;

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=10 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=160 overflow=reject;
    }

    server {
        listen       12106 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 2000000;
             tpcc_delivery_queue group_id=106 node_id=1 size=50000;
             tpcc_max_txn_cnt 0;
             tpcc_delivery_mdb dsn=tpccmeta user=tpcc password=tpcc;
             tpcc_delivery_runtime start=1567046366 rup=7200 rdown=15300 mi=7200 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

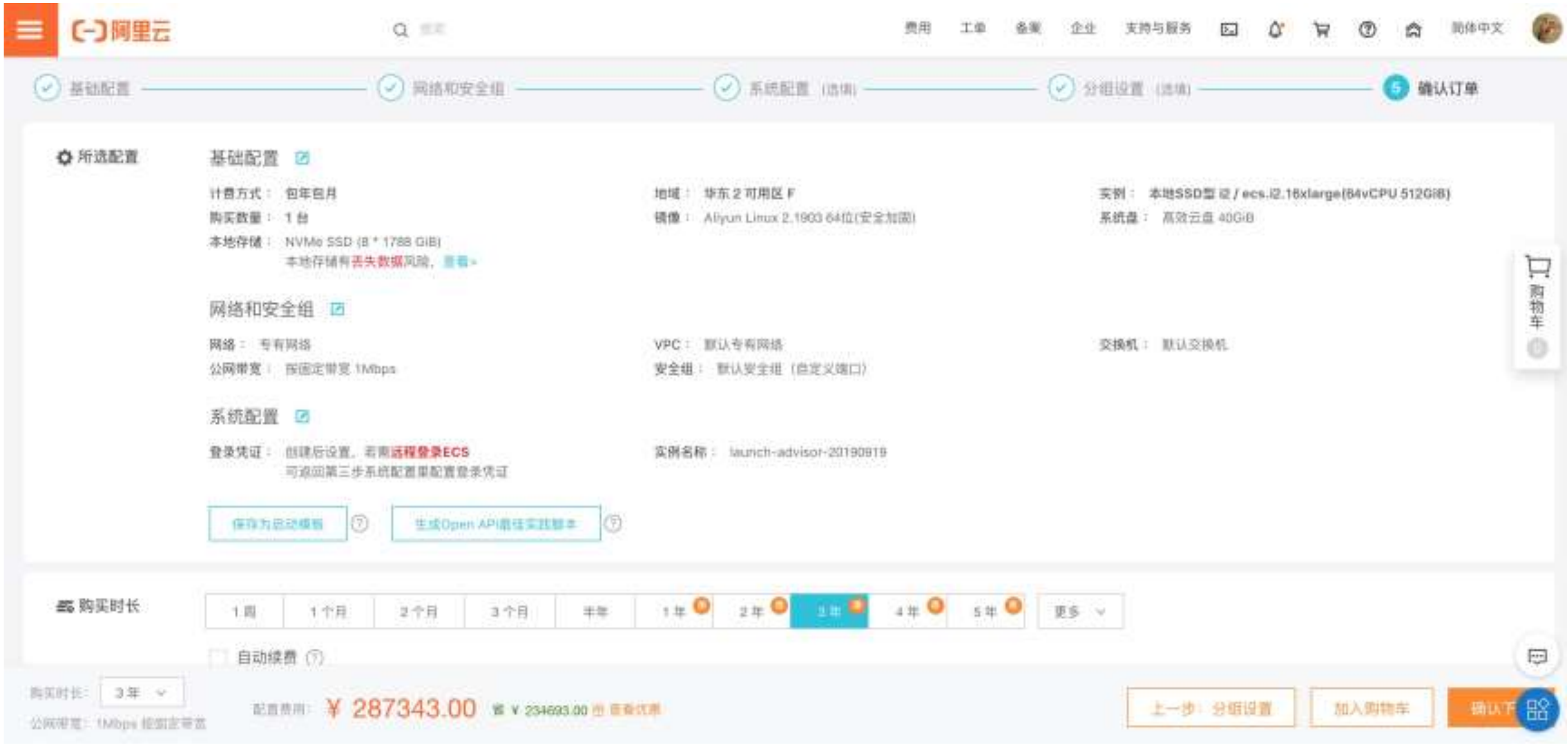
```
Filesystem      1K-blocks      Used Available Use% Mounted on
devtmpfs         65868324         0  65868324   0% /dev
tmpfs            65879300         0  65879300   0% /dev/shm
tmpfs            65879300       808  65878492   1% /run
tmpfs            65879300         0  65879300   0% /sys/fs/cgroup
/dev/vda1        41147472   3714528  35529452  10% /
tmpfs            13175860         0  13175860   0% /run/user/0
/dev/vdb1       210327156  58555168 141018248  30% /home
/dev/vdb2       421707440 352513064  47703128  89% /data/1
/dev/vdb3       421705392 207340760 192873484  52% /data/log1
tmpfs            13175860         0  13175860   0% /run/user/1002
```

/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
```

Appendix D: Third Party Price Quotes

ecs.i2.16xlarge



所选配置

基础配置

计费方式： 包年包月

购买数量： 1 台

本地存储： NVMe SSD (8 * 1788 GiB)
本地存储有丢失数据风险，查看>

地域： 华东 2 可用区 F

镜像： Aliyun Linux 2.1903 64位(安全加固)

实例： 本地SSD型 i2 / ecs.i2.16xlarge(64vCPU 512GiB)

系统盘： 高效云盘 40GiB

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： 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-20190919

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

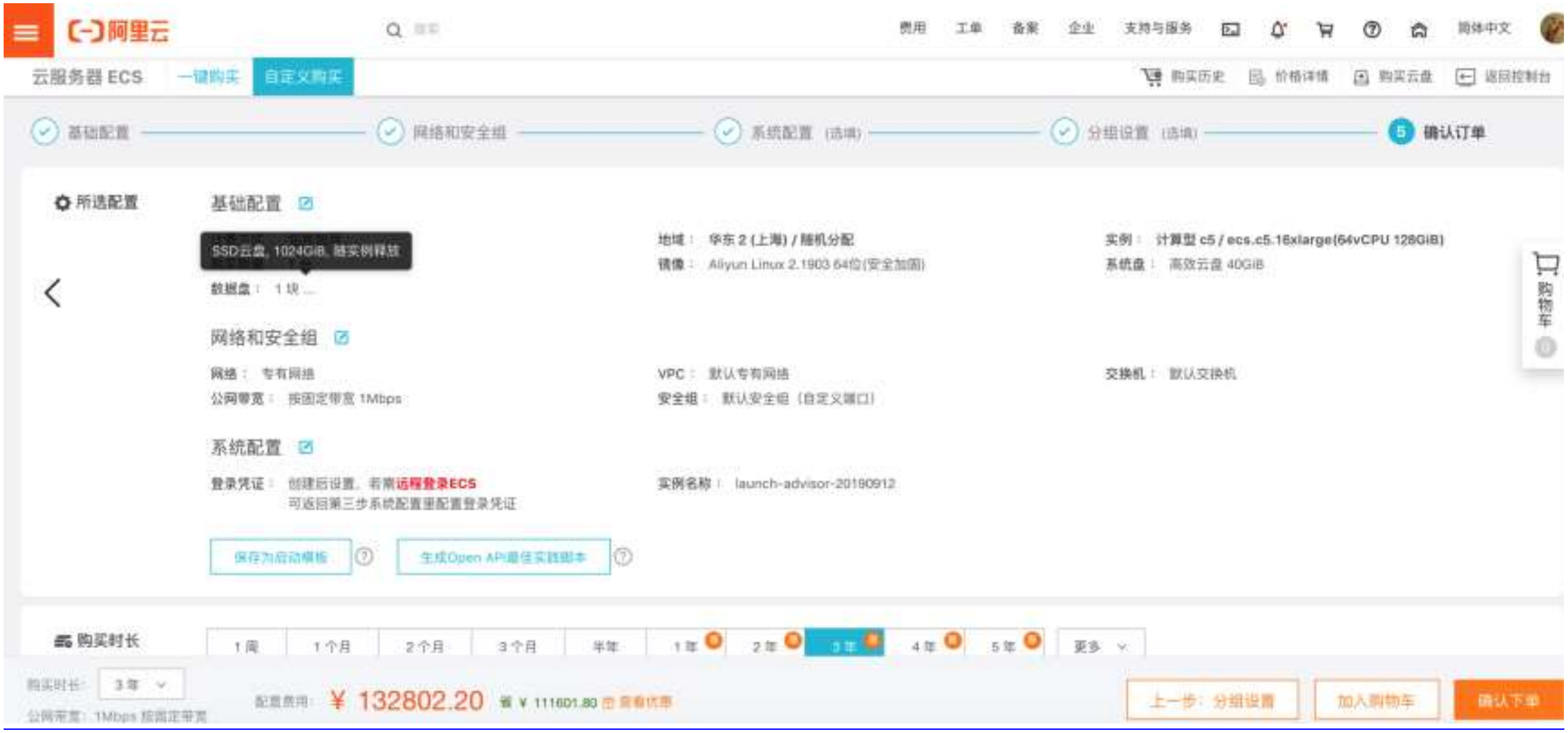
□ 自动续费

购买时长： 3 年

公网带宽： 1Mbps 按固定带宽

配置费用： ¥ 287343.00 省 ¥ 234693.00 查看优惠

ecs.c5.16xlarge (With 1024GB Standard SSD Disk)



所选配置

基础配置

SSD云盘, 1024GiB, 随实例释放

数据盘： 1 块 ...

地域： 华东 2 (上海) / 随机分配

镜像： Aliyun Linux 2.1903 64位(安全加固)

实例： 计算型 c5 / ecs.c5.16xlarge(64vCPU 128GiB)

系统盘： 高效云盘 40GiB

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： 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重配置登录凭证

实例名称： launch-advisor-20190912

购买时长

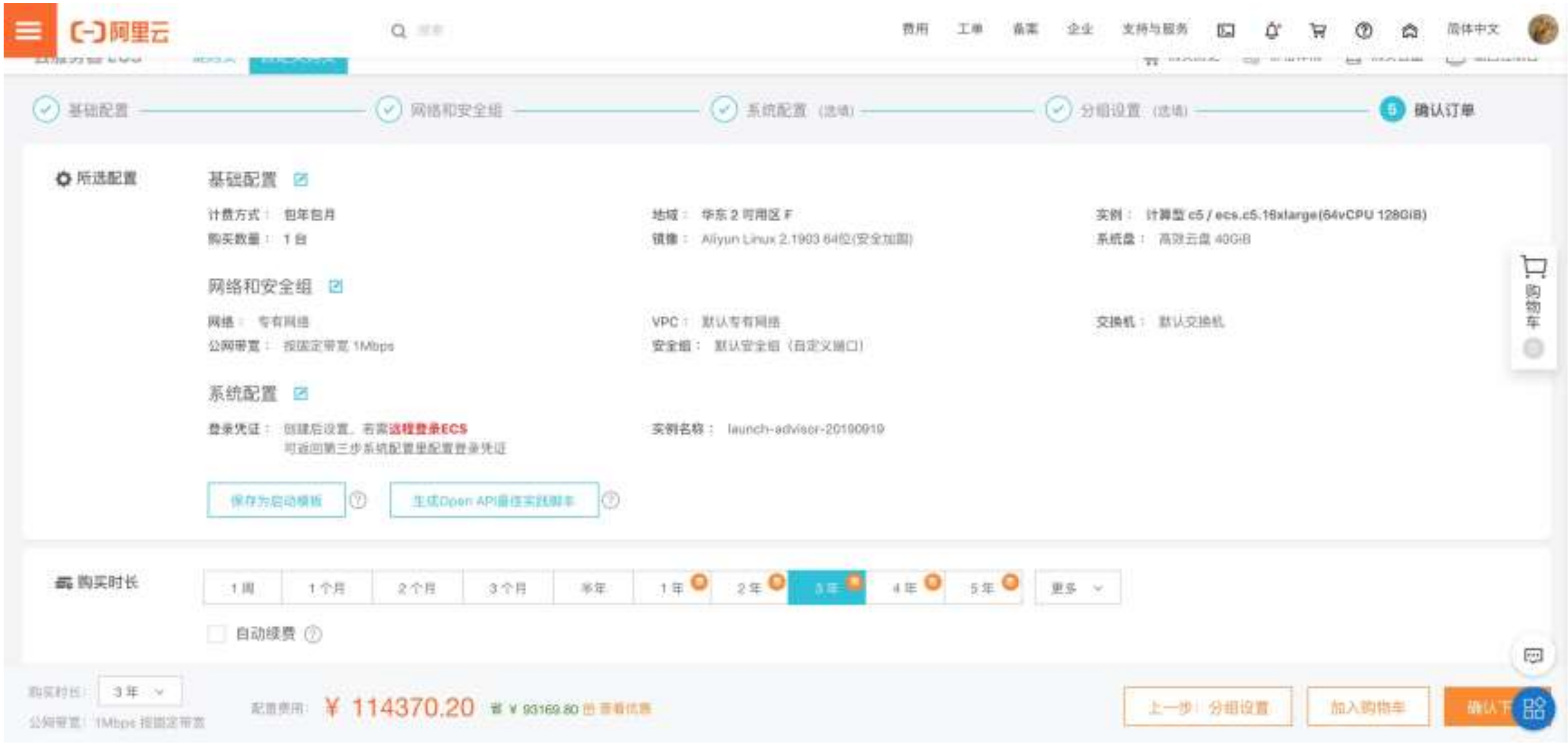
1 周 | 1 个月 | 2 个月 | 3 个月 | 半年 | 1 年 | 2 年 | 3 年 | 4 年 | 5 年 | 更多

购买时长： 3 年

公网带宽： 1Mbps 按固定带宽

配置费用： ¥ 132802.20 省 ¥ 111601.80 查看优惠

ecs.c5.16xlarge



所选配置

基础配置

计费方式： 包年包月
购买数量： 1 台
地域： 华东 2 可用区 F
镜像： Aliyun Linux 2.1903 64位(安全加固)
实例： 计算型 c5 / ecs.c5.16xlarge(64vCPU 128GiB)
系统盘： 高效云盘 40GiB

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： 默认安全组（自定义端口）
交换机： 默认交换机

系统配置

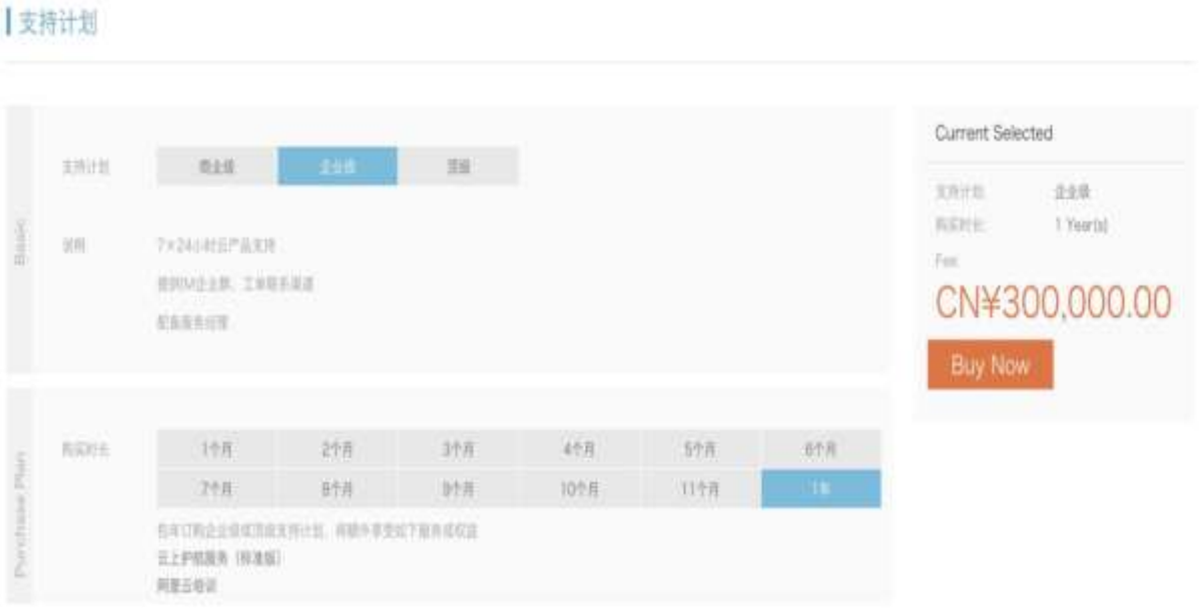
登录凭证： 创建后设置，若需远程登录ECS
可返回第三步系统配置里配置登录凭证
实例名称： launch-advisor-20190919

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

购买时长： 3 年
公网带宽： 1Mbps 按固定带宽
配置费用： ¥ 114370.20 省 ¥ 93169.80 查看优惠

Alibaba Cloud Support Price



支持计划

Basic

支持计划　商业级　[✓ 企业级]　顶级

说明　7×24小时云产品支持

提供IM企业群、工单联系渠道

配备服务经理

Purchase Plan

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]

包年订购企业级或顶级支持计划，将额外享受如下服务或权益

云上护航服务（标准版）

阿里云培训

Current Selected

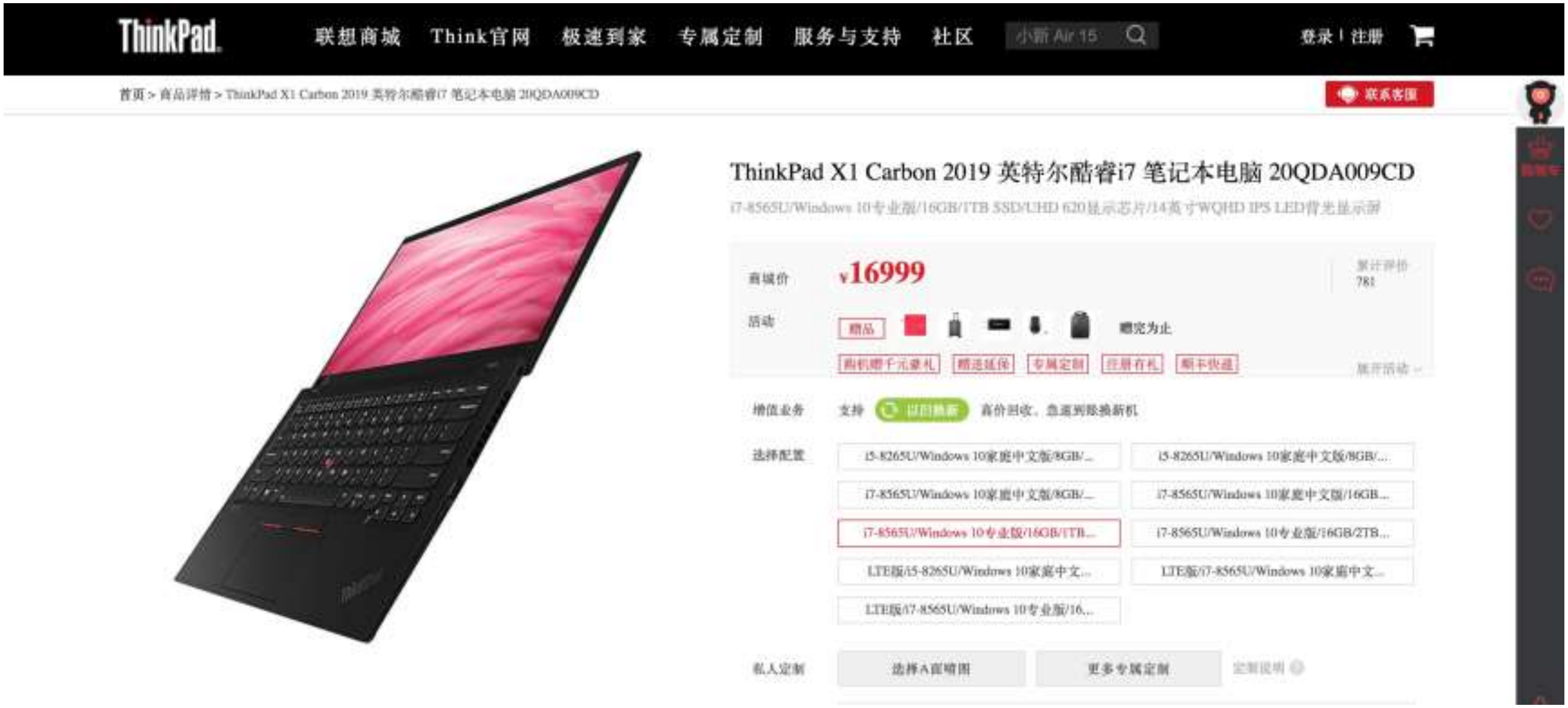
支持计划:　企业级

购买时长:　1 Year(s)

Fee

CN¥300,000.00

Lenovo ThinkPad X1 Carbon 2019 14" Laptop



首页 > 商品详情 > ThinkPad X1 Carbon 2019 英特尔酷睿i7 笔记本电脑 20QDA009CD

ThinkPad X1 Carbon 2019 英特尔酷睿i7 笔记本电脑 20QDA009CD

i7-8565U/Windows 10专业版/16GB/1TB SSD/UHD 620显示芯片/14英寸WQHD IPS LED背光显示屏

商城价 ¥16999

累计评价 781

活动 赠品 赠完为止

购机赠千元豪礼 赠送延保 专属定制 注册有礼 顺丰快递

展开活动

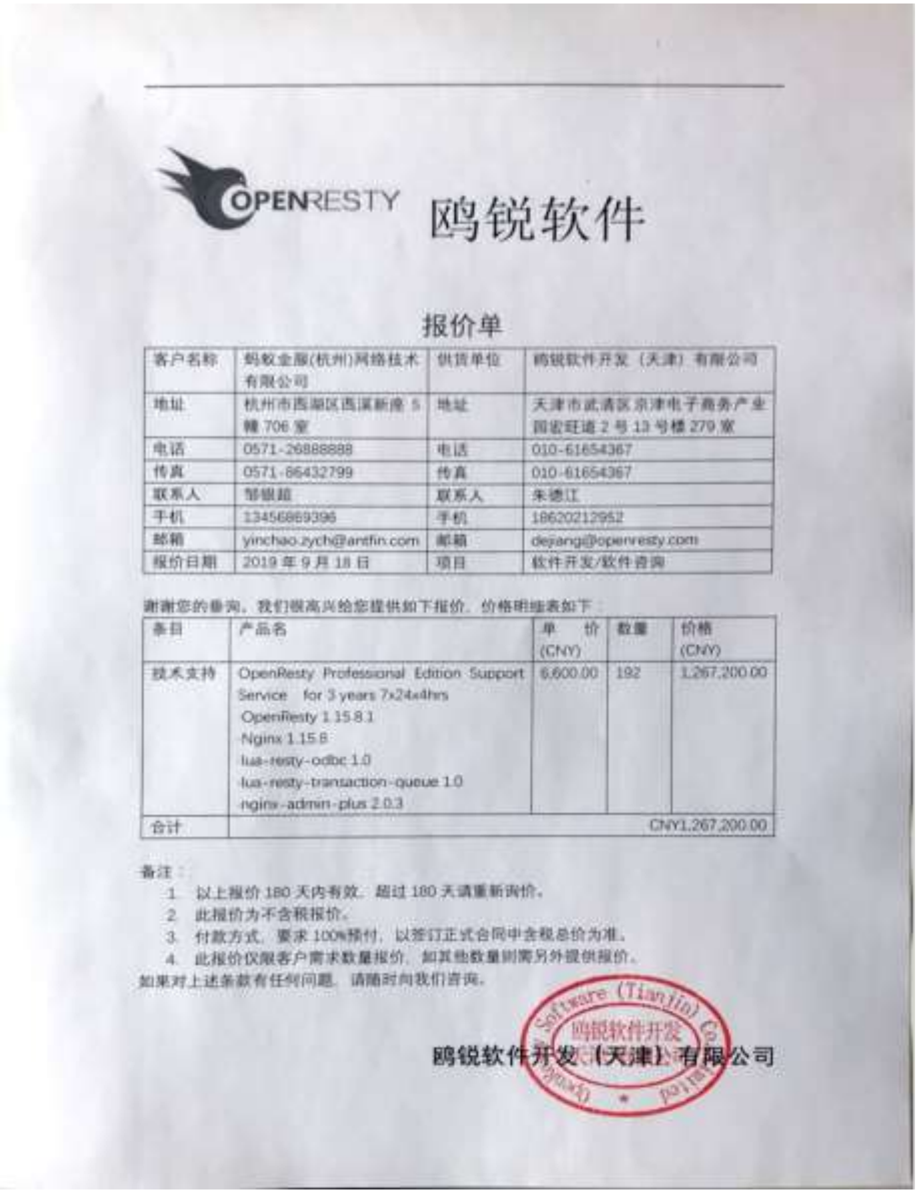
增值业务 支持 以旧换新 高价回收，急速到账换新机

选择配置

i5-8265U/Windows 10家庭中文版/8GB/...	i5-8265U/Windows 10家庭中文版/8GB/...
i7-8565U/Windows 10家庭中文版/8GB/...	i7-8565U/Windows 10家庭中文版/16GB...
i7-8565U/Windows 10专业版/16GB/1TB...	i7-8565U/Windows 10专业版/16GB/2TB...
LTE版/i5-8265U/Windows 10家庭中文...	LTE版/i7-8565U/Windows 10家庭中文...
LTE版/i7-8565U/Windows 10专业版/16...	

私人定制 选择A面喷图 更多专属定制 定制说明

OpenResty Support Price



OPENRESTY 鸥锐软件

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	2019 年 9 月 18 日	项目	软件开发/软件咨询

谢谢您的垂询。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	6,600.00	192	1,267,200.00
合计				CNY1,267,200.00

备注：

1. 以上报价 180 天内有效，超过 180 天请重新询价。
2. 此报价为不含税报价。
3. 付款方式：要求 100%预付，以签订正式合同中含税总价为准。
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。

Openresty Software (Tianjin) Co., Limited 鸥锐软件开发

鸥锐软件开发（天津）有限公司