TPC Benchmark™ H Full Disclosure Report

Cloud OceanBase

(with 64 Compute Instances)

using

OceanBase V3.2

and

OceanBase File System V1.0

Alibaba Group Cloud Linux Server release 2.19

First Edition

May 20, 2021

First Edition – May 20, 2021

OceanBase V3.2
OceanBase File System V1.0

The OceanBase Cloud products, services or features identified in this document may not yet be available or may not be available in all areas and may be subject to change without notice. Consult your local OceanBase business contact for information on the products or services available in your area. You can find additional information via OceanBase website at https://www.oceanbase.com/. Actual performance and environmental costs of OceanBase Cloud products will vary depending on individual customer configurations and conditions

Abstract

Overview

This report documents the methodology and results of the TPC Benchmark™ H test conducted on Alibaba Cloud ECS using OceanBase in conformance with the requirements of the TPC Benchmark™ H Standard Specification, Revision 3.0.0. The operating system used for the benchmark was Alibaba Group Cloud Linux Server release 2.19.

The TPC Benchmark™ H was developed by the Transaction Processing Performance Council (TPC). The TPC was founded to define transaction processing benchmarks and to disseminate objective, verifiable performance data to the industry.

TPC Benchmark™ H Full Disclosure Report and other information can be downloaded from the Transaction Processing Performance Council web site at www.tpc.org.

Executive Summary Section

The first section of this report contains the Executive Summary and Numerical Quantities Summary of the benchmark results.

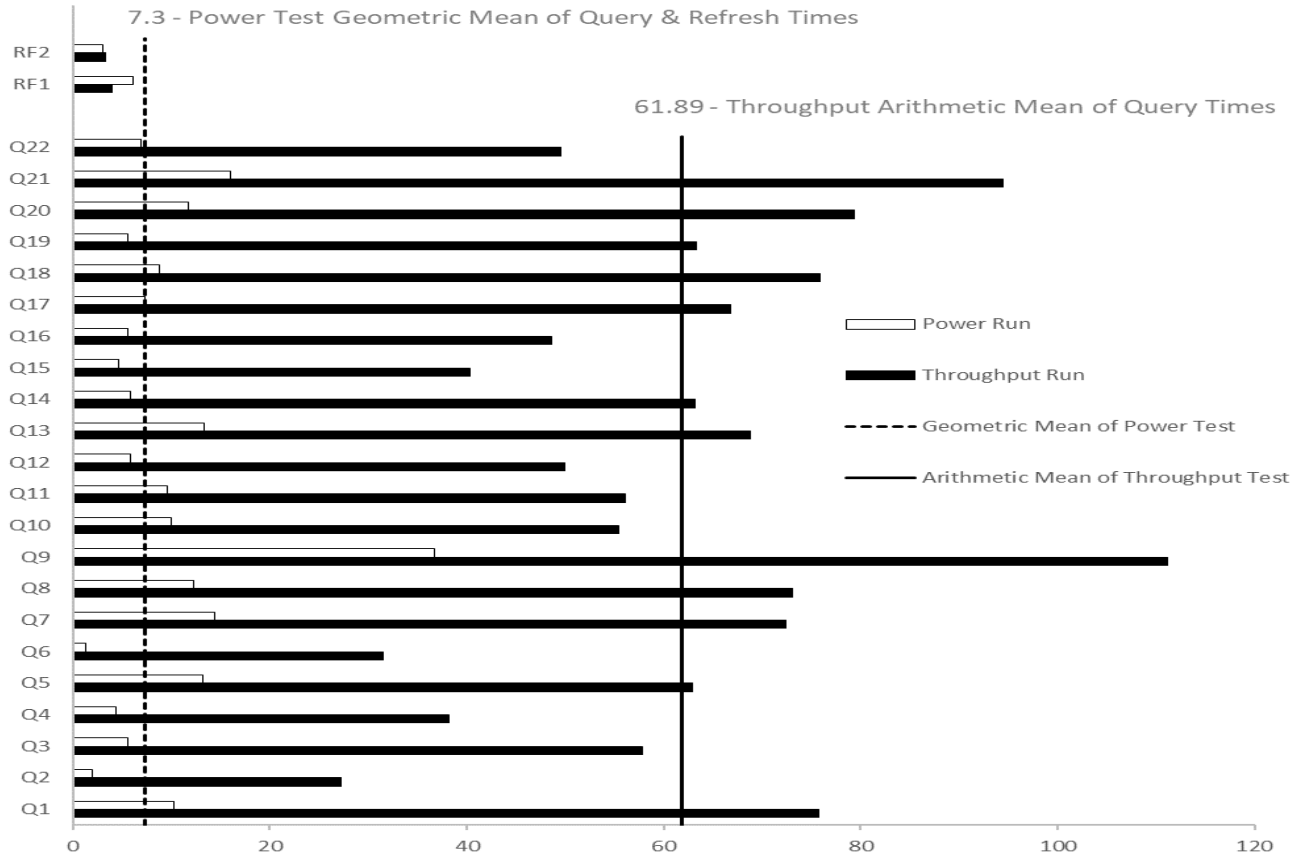
Auditor

The benchmark configuration, environment and methodology used to produce and validate the test results, and the pricing model used to calculate the cost per kQphH were audited by Doug Johnson of InfoSizing to verify compliance with the relevant TPC specifications.

The auditor’ s information is available in Section 9.1.

	Cloud OceanBase	TPC-H Rev. 3.0.0 TPC Pricing Rev. 2.7.0 Report Date: 20-May-21
Total System Cost	Composite Query per Hour Metric	Price/Performance
69,336,912 CNY	**15,265,305.7** **QphH@30000GB**	**4,542.13 CNY** **Price/kQphH@30000GB**

Database Size	Database Manager	Operating System	Other Software	Availability Date
30,000 GB	**OceanBase V3.2**	**Alibaba Cloud Linux Server 2.19**	**OceanBase File System V1.0**	**07/31/2021**



Database Load Time: 02:47:41	Load Includes Backup: N	Memory Ratio: 163.8%	Total Data Storage/Database Size: 1.35
Storage Redundancy Level: Level Two	**Base Table: Level Two**	**Auxiliary Data Structures: Level Two**	**Other: Level Two**

System Configuration

System Component	Per node Compute Node	System Total Compute Node	OFS Storage Service
Node:	1	64	N/A
vCPU:	80	5120	N/A
Memory:	768GB	49,152GB	N/A
Storage:	40 GB	2,560 GB	38,000 GB
Total Storage:	40,560GB		

Cloud OceanBase

TPC-H Rev. 3.0.0
TPC Pricing Rev. 2.7.0

Report Date:
20-May-21

Description	Part Number	Source	Unit Price (CNY)	Qty	Extended Price (CNY)	3 yr Maint. Price (CNY)
License Computer and Software Service						
Cloud OceanBase (64 nodes, 80 cores per node, 3 year prepay) Each node with 80 vCPUs 768 GB Memory	East China 2	1	66,908,928	1	66,908,928	Included
OceanBase File System (38,000 GB Storage Capacity, 3 year prepay)	East China 2	1	2,427,984	1	2,427,984	Included
				Subtotal	**69,336,912**	**0.00**

3-year Cost of Ownership: **69,336,912.00**
QphH@30000GB: **15,265,305.7**
¥/kQphH@30000GB: **4,542.13**

Audited by: Doug Johnson, 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 the stated prices are not available according to these terms, please inform the TPC at pricing@tpc.org. Thank you.

Cloud OceanBase

TPC-H Rev. 3.0.0
TPC Pricing Rev. 2.7.0

Report Date:
20-May-21

Measurement Results

Database Scale Factor	30,000
Total Data Storage/Database Size	1.35
Percentage Memory/Database Size	163.8%
Start of Database Load Time	2021-05-16 11:45:24
End of Database Load Time	2021-05-16 14:33:05
Database Load Time	02:47:41
Query Streams for Throughput Test (S)	10
TPC-H Power	14,788,059.7
TPC-H Throughput	15,757,953.6
TPC-H Composite	15,265,305.7
Total System Price Over 3 Years	69,336,912 (CNY)
TPC-H Price/Performance Metric (¥/kQphH@30000GB)	4,542.13 (CNY)

Measurement Interval

Measurement Interval in Throughput Test (Ts)	1,507.81

Duration of stream execution:

Power Run	Seed	Query Start Time Query End Time	Total Time (hh:mm:ss)	RF1 Start Time RF1 End Time	RF2 Start Time RF2 End Time
	516143305	2021-05-16 15:08:39 2021-05-16 15:12:12	00:03:33	2021-05-16 15:08:33 2021-05-16 15:08:39	2021-05-16 15:12:12 2021-05-16 15:12:15

Throughput Stream	Seed	Query Start Time Query End Time	Total Time (hh:mm:ss)	RF1 Start Time RF1 End Time	RF2 Start Time RF2 End Time
1	516143306	2021-05-16 15:12:14 2021-05-16 15:34:31	00:22:17	2021-05-16 15:36:09 2021-05-16 15:36:13	2021-05-16 15:36:13 2021-05-16 15:36:17
2	516143307	2021-05-16 15:12:14 2021-05-16 15:35:47	00:23:33	2021-05-16 15:36:17 2021-05-16 15:36:21	2021-05-16 15:36:21 2021-05-16 15:36:24
3	516143308	2021-05-16 15:12:14 2021-05-16 15:34:48	00:22:34	2021-05-16 15:36:24 2021-05-16 15:36:28	2021-05-16 15:36:28 2021-05-16 15:36:31
4	516143309	2021-05-16 15:12:14 2021-05-16 15:34:05	00:21:51	2021-05-16 15:36:31 2021-05-16 15:36:35	2021-05-16 15:36:35 2021-05-16 15:36:39
5	516143310	2021-05-16 15:12:14 2021-05-16 15:36:09	00:23:55	2021-05-16 15:36:39 2021-05-16 15:36:43	2021-05-16 15:36:43 2021-05-16 15:36:46
6	516143311	2021-05-16 15:12:14 2021-05-16 15:33:09	00:20:55	2021-05-16 15:36:46 2021-05-16 15:36:50	2021-05-16 15:36:50 2021-05-16 15:36:53
7	516143312	2021-05-16 15:12:14 2021-05-16 15:35:35	00:23:21	2021-05-16 15:36:53 2021-05-16 15:36:57	2021-05-16 15:36:57 2021-05-16 15:37:00
8	516143313	2021-05-16 15:12:14 2021-05-16 15:33:49	00:21:35	2021-05-16 15:37:00 2021-05-16 15:37:04	2021-05-16 15:37:04 2021-05-16 15:37:08
9	516143314	2021-05-16 15:12:14 2021-05-16 15:36:10	00:23:56	2021-05-16 15:37:08 2021-05-16 15:37:12	2021-05-16 15:37:12 2021-05-16 15:37:15
10	516143315	2021-05-16 15:12:14 2021-05-16 15:35:21	00:23:07	2021-05-16 15:37:15 2021-05-16 15:37:19	2021-05-16 15:37:19 2021-05-16 15:37:22

	Cloud OceanBase	TPC-H Rev. 3.0.0 TPC Pricing Rev. 2.7.0 Report Date: 20-May-21

TPC-H Timing Intervals (in seconds)

Duration of query execution:

Stream ID	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9	Q10	Q11	Q12
0	10.22	1.95	5.63	4.43	13.15	1.26	14.36	12.32	36.68	9.98	9.64	5.91
1	36.06	43.71	38.38	35.25	54.63	48.95	103.84	66.15	137.02	51.04	55.23	38.98
2	69.40	14.13	58.33	33.53	71.84	3.28	37.35	73.72	129.71	66.32	54.25	107.05
3	113.14	42.18	40.18	29.31	98.98	66.32	98.40	71.27	120.29	83.52	48.26	36.51
4	72.94	39.80	58.57	25.70	67.58	42.80	70.21	71.69	123.18	65.19	56.67	46.94
5	39.20	19.66	120.54	25.05	63.88	26.61	64.48	67.77	68.77	41.42	52.53	89.17
6	78.38	17.10	32.64	49.63	86.55	38.68	72.09	70.66	113.33	35.45	42.05	50.15
7	87.26	6.52	74.97	19.39	60.48	29.71	86.72	72.36	94.22	81.86	47.28	18.03
8	67.64	32.68	36.65	31.31	64.12	23.35	46.60	62.50	105.16	37.71	54.72	36.82
9	151.41	37.48	67.34	30.62	25.24	31.95	105.03	131.26	101.02	46.87	62.98	39.84
10	42.17	19.33	50.20	101.73	36.38	3.32	39.17	43.62	118.46	45.43	87.30	35.40
Minimum	10.22	1.95	5.63	4.43	13.15	1.26	14.36	12.32	36.68	9.98	9.64	5.91
Maximum	151.41	43.71	120.54	101.73	98.98	66.32	105.03	131.26	137.02	83.52	87.30	107.05
Average	69.80	24.96	53.04	35.09	58.44	28.75	67.11	67.57	104.35	51.34	51.90	45.89

Stream ID	Q13	Q14	Q15	Q16	Q17	Q18	Q19	Q20	Q21	Q22	RF1	RF2
0	13.36	5.79	4.69	5.58	7.25	8.72	5.63	11.69	16.05	6.89	6.16	3.04
1	67.82	33.75	40.37	38.77	80.02	69.62	68.08	117.74	50.50	60.40	3.98	3.23
2	38.04	79.02	36.68	73.24	107.08	44.75	43.07	133.91	85.18	51.97	4.18	3.29
3	65.67	35.17	83.82	30.78	30.17	27.11	46.70	71.43	67.13	46.62	3.91	3.25
4	68.17	35.77	25.83	43.14	64.99	69.28	71.94	46.83	76.93	65.84	3.93	3.30
5	134.84	76.47	78.04	28.68	43.93	119.15	73.28	91.45	52.86	57.03	4.05	3.28
6	41.18	39.09	14.92	53.55	55.93	36.56	130.48	41.13	94.46	60.62	4.01	2.98
7	69.09	25.33	26.50	33.06	39.99	181.21	76.02	43.05	188.51	39.21	3.94	3.02
8	46.26	116.93	32.17	121.28	92.81	80.43	21.38	55.34	90.23	38.67	4.01	3.66
9	57.49	12.91	36.45	10.12	90.14	79.97	54.78	103.54	126.88	31.51	4.10	3.28
10	100.18	177.03	28.64	52.98	62.37	50.32	47.51	89.06	112.02	43.34	3.96	3.55
Minimum	13.36	5.79	4.69	5.58	7.25	8.72	5.63	11.69	16.05	6.89	3.91	2.98
Maximum	134.84	177.03	83.82	121.28	107.08	181.21	130.48	133.91	188.51	65.84	6.16	3.66
Average	63.83	57.93	37.10	44.65	61.33	69.74	58.08	73.20	87.34	45.65	4.20	3.26

Table of Contents

TPC BENCHMARK H OVERVIEW

The TPC Benchmark^{TM} H (TPC-H) is a Decision Support benchmark. It is a suite of business-oriented adhoc queries and concurrent modifications. The queries and the data populating the database have been chosen to have broad industry-wide relevance while maintaining a sufficient degree of ease of implementation. This benchmark illustrates Decision Support systems that:

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

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

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

Further information is available at www.tpc.org.

0. General Items

0.1 Benchmark Sponsor

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

This benchmark is sponsored by Beijing OceanBase Technologies Inc.

0.2 Parameter Settings

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

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

In the event that some parameters and options are set multiple times, it must be easily discernible by an interested reader when the parameter or option was modified and what new value it received each time.

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

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

0.3 Configuration Diagram

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

- *Number and type of processors.*
- *Size of allocated memory, and any specific mapping/partitioning of memory unique to the test.*
- *Number and type of disk units (and controllers, if applicable).*

• Number of channels or bus connections to disk units, including their protocol type.

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

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

The system diagram of the measured system is depicted in Figure 1. and the configuration of the measured systems is shown in Table 1. The system is composed of 64 compute instances and an OFS (OceanBase File System) service with storage capacity of 38,000 GB. There is no difference between the priced and measured configurations.

Configuration	**Compute Instance**
#Instances	64
Per Instance Config	80 vCPUs
	768 GB RAM
	40 GB Local SSD Disk
Network	32 Gbits/s

Configuration	**OFS Storage**
Storage Capacity	38,000 GB

Table 1. Configuration of the measured system.

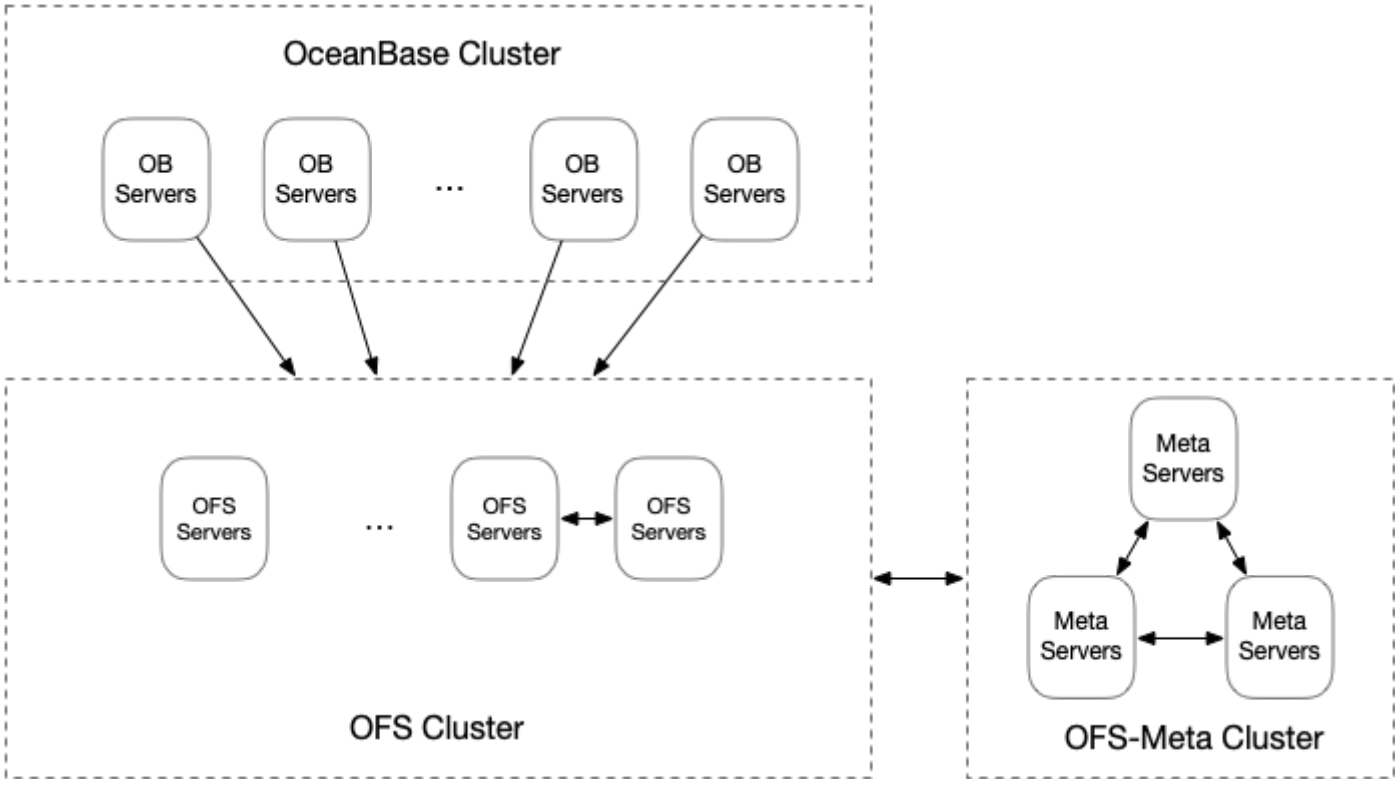


Fig. 1. System diagram of the measured configuration.

1. Clause 1: Logical Database Design

1.1 Database Definition Statements

Listings must be provided for all table definition statements and all other statements used to set-up the test and qualification databases. All listings must be reported in the supporting files archive.

The Supporting File Archive contains the table definitions and all other statements to set up the tables used in this benchmarking.

1.2 Physical Organization

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

The concept of physical organization includes, but is not limited to: record clustering (i.e., rows from different logical tables are co-located on the same physical data page), index clustering (i.e., rows and leaf nodes of an index to these rows are co-located on the same physical data page), and partial fill-factors (i.e., physical data pages are left partially empty even though additional rows are available to fill them).

Physical organization requires no user input. No column reordering nor other physical alteration to the database is used.

1.3 Horizontal Partitioning

Horizontal partitioning of tables and rows in the test and qualification databases (see Clause 1.5.4) must be disclosed. Scripts to perform horizontal partitioning must be reported in the supporting files archive.

ORDERS and LINEITEM tables are partitioned with key partitioning on O_ORDERKEY and L_ORDERKEY respectively, the partition number is 4096. PART and PARTSUPP tables are partitioned with key partitioning on P_PARTKEY and PS_PARTKEY respectively, the partition number is 4096. All the partitioning is done via table definition DDL as provided in the Supporting File Archive.

1.4 Replication

Any replication of physical objects must be disclosed and must conform to the requirements of Clause 1.5.7. Scripts to perform any replication must be reported in the supporting files archive.

No replication was used.

2. Clause 2: Queries and Refresh Functions Related Items

2.1 Query Language

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

SQL was the query language used to implement the queries.

2.2 Verifying Method of Random Number Generation

The version number, release number, modification number, and patch level of QGen must be disclosed. Any modifications to the QGen source code must be reported in the supporting files archive.

TPC-supplied DBGen version 3.0.0 and QGen version 3.0.0 were used.

2.3 Query Text and Output Data from Qualification Database

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

The actual query text and query output are included in the Supporting Files Archive. The standard queries were used throughout with the following modifications:

- LIMIT syntax used to restrict the number of output rows (Q2, Q3, Q10, Q18, Q21).
- "c_orders" was used as subquery alias instead of "c_orders (c_custkey, c_count)" in Q13
- removed "AS" in the alias naming part of the SQL (Q7, Q8, Q9, Q22)
- "substr" was used instead of "substring" in Q22

2.4 Query Substitution Parameters and Seeds Used

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

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

2.5 Isolation Level

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

The queries and transactions were run with isolation level 3.

2.6 Source Code of Refresh Functions

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

Supporting Files Archive contains the Source Code of the refresh functions.

3. Clause 3: Database System Properties

The results of the ACID tests must be disclosed along with a description of how the ACID requirements were met. All code (including queries, stored procedures etc.) used to test the ACID requirements and their entire output must be reported in the supporting files archive.

The results of the ACID tests are disclosed as required.

3.1 ACID Properties

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

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

3.2 Atomicity Requirements

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 Atomicity of the Completed Transactions

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

The following operations were performed to verify the atomicity of the completed transactions:

1. Randomly selected one order key, and then retrieved the total price from table ORDERS and the extended price from table LINEITEM.
2. One transaction was performed using the selected order key.

3. The transaction was committed.
4. Retrieved the total price from the table ORDERS and the extended price from the table LINEITEM for the order key.
5. It was verified that the rows had been changed.

3.2.2 Atomicity of Aborted Transactions

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

The following operations were performed to verify the atomicity of the aborted transactions:

1. Randomly selected one order key, and then retrieved the total price from table ORDERS and the extended price from table LINEITEM.
2. One transaction was performed using the selected order key and the transaction was stopped before commit.
3. The transaction was rolled back.
4. Retrieved the total price from the table ORDERS and the extended price from the table LINEITEM for the order key.
5. It was verified that the rows had not been changed.

3.3 Consistency Requirements

Consistency is the property of the application that requires any execution of transactions to take the database from one consistent state to another. A consistent state for the TPC-H database is defined to exist when:

O_TOTALPRICE = SUM(trunc(trunc(L_EXTENDEDPRICE(1 - L_DISCOUNT), 2) * (1 + L_TAX), 2)) For each ORDER and LINEITEM defined by (O_ORDERKEY = L_ORDERKEY).*

3.3.1 Consistency Test

Verify that ORDERS and LINEITEM tables are initially consistent, submit the prescribed number of ACID Transactions with randomly selected input parameters, and re-verify the consistency of the ORDERS and LINEITEM.

The following query was executed before and after the consistency tests to demonstrate that the database kept staying in a consistent state both initially and after submitting transactions:

```
select  *
from (
```

```
    select o_orderkey, (o_totalprice - sum( truncate((truncate((l_extendedprice * (1-l_discount)),
2)*(1+l_tax)),2))) part_res
   from orders, lineitem
   where o_orderkey=l_orderkey
   group by o_orderkey, o_totalprice
) temp where not part_res=0;
```

The following operations were performed to verify the consistency of ACID transactions:

1. The consistency of the ORDERS and LINEITEM tables was verified.
2. For each of the 11 execution streams, 100 transactions were prepared.
3. For all 11 execution streams, the prepared ACID transactions were executed.
4. The consistency of the ORDERS and LINEITEM tables was verified again.

3.4 Isolation Requirements

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

The steps of the isolation tests were adapted to the OceanBase isolation environment.

3.4.1 Isolation Test 1 – Read-Write Conflict with Commit

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

The following operations were performed to verify the isolation for a read-only and a read-write committed transaction:

1. Started a query and verified that the row was retrieved.
2. Started an update transaction, read and updated the same row. Stalled before commit.
3. Started the same query and verified that the row has not changed.
4. Committed the update transaction.
5. Started a query and verified that the new row was retrieved.

3.4.2 Isolation Test 2 – Read-Write Conflict with Rollback

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

The following operations were performed to verify the isolation for a read-only and a rolled back read-write transaction:

1. Started a query and verified that the row was retrieved.
2. Started an update transaction, read and updated the same row. Stalled before commit.
3. Started the same query and verified that the row has not changed.
4. Rolled back the update transaction.
5. Started a query and verified that the old row was retrieved.

3.4.3 Isolation Test 3 – Write-Write Conflict with Commit

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

The following operations were performed to verify the isolation of two update transactions:

1. Started an update transaction T1 with a random O_KEY and L_KEY, stopped T1 immediately prior to COMMIT.
2. Started an update transaction T2 with the same O_KEY, verify T2 was blocked.
3. Committed T1. T2 reported an error about "can't serialize "and rollbacked.
4. Retrieved the same row and verified extended_price was updated to the value as calculated in T1.

3.4.4 Isolation Test 4 – Write-Write Conflict with Rollback

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

The following operations were performed to verify isolation of two update transactions after the first one is rolled back:

1. Started an update transaction T1, and stopped T1 immediately prior to COMMIT.
2. Started an update transaction T2. T2 was blocked.
3. Rollbacked T1. T2 then completed and committed.
4. Retrieved the same row and verified extended_price was updated to the value as calculated in T2.

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

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

The following operations were performed to demonstrate the ability of read and write transactions involving different tables to make progress concurrently:

1. Started a query and verified that the row was retrieved.
2. Started an update transaction, read and updated the same row. Stalled before commit.

3. Started another transaction that performed the following operation: Select random values of PS_PARTKEY AND PS_SUPPKEY. Return all columns of the PARTSUPP table for which PS_PARTKEY and PS_SUPPKEY are equal to the selected values.
4. Verified that the read transaction completed.
5. Committed the update transaction.
6. Started the same query and verified that the new row was retrieved.

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

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

The following query was used to ensure sufficient execution to perform the test:

```
SELECT   l1.l_orderkey
FROM     lineitem l1,
         lineitem l2,
         lineitem l3,
        lineitem l4,
        lineitem l5
WHERE    l1.l_shipdate <= date '1998-12-01'
AND      l1.l_orderkey = l2.l_orderkey
AND      l1.l_linenumber = l2.l_linenumber
AND      l1.l_extendedprice = l3.l_extendedprice
AND      l3.l_quantity<30
AND      l4.l_quantity = l1.l_quantity
AND      l4.l_orderkey < 20000
AND      l5.l_receiptdate = l1.l_receiptdate
AND      l5.l_partkey < 140
GROUP BY l1.l_orderkey
ORDER BY l1.l_orderkey
LIMIT    2 ;
```

1. A transaction, T1, which executed the above query on the qualification database, was started using a random DELTA.
2. A transaction, T2, was started for a randomly selected O KEY, L KEY, and DELTA.
3. T2 completed and appropriate rows in the tables ORDERS, LINEITEM, and HISTORY had been changed.
4. T1 was still executing.
5. Transaction T1 completed executing the query.

3.5 Durability Requirements

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

The following steps were performed for the durability test:

1. The consistency of the ORDERS and LINEITEM tables was verified.
2. 400 transactions for each of the 257 executions streams were prepared.
3. After at least 100 ACID transactions were completed by each of the 257 execution streams.
4. A durability failure was induced (see details for each failure shown below).
5. The consistency of the ORDERS and LINEITEM tables was verified again.
6. The durability success files were compared with the HISTORY table.

All durability tests were performed on the cluster shown in Section 0.3.

3.5.1 Permanent Unrecoverable Failure of Any Durable Medium

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

Disk, node, and controller failure tests were performed together as shown in Section 3.5.5.

3.5.2 System Crash

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

The system crash, memory failure and node or controller failure tests were performed together as shown in Section 3.5.5.

3.5.3 Memory Failure

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

The system crash, memory failure and node or controller failure tests were performed together as shown in Section 3.5.5.

3.5.4 Loss of External Power

Loss of External Power: Guarantee the database and the effects of committed updates are preserved during the loss of all external power to the SUT for an indefinite time period.

Each OceanBase compute node, OFS node and OFS Meta node are deployed on Alibaba Cloud ECS and Alibaba Cloud ECS is configured with redundant power supplies and Uninterruptible Power Supply. The UPS capabilities are sufficient to keep the entire SUT running for a period of at least 30 minutes in the event of a total loss of all external power.

In addition, various roles of single node crash were performed as described in section 3.5.5.

3.5.5 Node or Controller Failure

Guarantee the database and committed updates are preserved across failure of the controller or the whole node.

The tested cluster consists of 64 computing instances and 64 OFS instances. Four tests were performed to test the durability:

1. Shutdown the host machine of one of the computing instances. After completion of the durability test, we will bring up this same instance and the instance will join the cluster automatically.
2. Shutdown the host machine of the root node (node where OceanBase Root Service resides). After completion of the durability test, the full cluster was restored automatically after the root node was started manually.
3. Shutdown the host machine of one of the OFS storage service instances. After completion of the durability test, we will bring up this same instance and the instance will join the OFS cluster automatically.
4. Shutdown the host machine of one of the OFS MetaDB instances. After completion of the durability test, we will bring up this same instance and the instance will join the OFS cluster automatically.

4. Clause 4: Scaling and Database Population

4.1 Initial Cardinality of Tables

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

Table 2 lists the TPC Benchmark™ H defined tables and the row count for each table as they existed upon completion of the test database build.

Table	#Rows
Lineitem	179,999,978,268
Order	45,000,000,000
Partsupp	24,000,000,000
Part	6,000,000,000
Customer	4,500,000,000
Supplier	300,000,000
Nation	25

Region	5

Table 2. Table Cardinalities.

4.2 Distribution of Tables and Logs Across Media

The distribution of tables and logs across all media must be explicitly described using a format similar to that shown in the following example for both the measured and priced configurations.

Server Node	Disk Type	Disk Drive	Content
Compute Node	Local SSD DISK	/dev/vda1	OS, RDBMS logs

Table 3. Distribution of tables and logs.

All the base tables were stored on the OFS storage, the sizes of tables are shown in Table 4.

Table Name	Lineitem	Orders	Partsupp	Part	Customer	Supplier	Nation	Region
Table Size	6,943GB	1,978GB	1,105GB	216GB	332GB	21GB	1,517 Bytes	705 Bytes

Table 4. Table Size on OFS storage.

4.3 Mapping of Database Partitions/Replication

The mapping of database partitions/replications must be explicitly described.
The intent is to provide sufficient detail about partitioning and replication to allow independent reconstruction of the test database.

LINEITEM, ORDER, PART, PARTSUPP, CUSTOMER, SUPPLIER tables are horizontally partitioned using key partitioning. The columns used for the key partitioning are controlled by DDL statements (see Supporting Files Archive, keyword: partition by). NATION and REGION tables are not partitioned in any way.

In this benchmarking, the above mentioned LINEITEM, ORDER, PART, PARTSUPP, CUSTOMER, SUPPLIER tables are partitioned into 4096 logical key partitions.

4.4 Implementation of Data Redundancy

Implementations may use data redundancy mechanism(s). The type of data redundancy mechanism(s) and any configuration parameters (e.g., RAID level used must be disclosed for each device). If data redundancy mechanism(s) are used in an implementation, the logical intent of their use must be disclosed.

RAID is not used. The Data Redundancy is internally managed by OFS, a distributed file system, which uses multiple replica and erasure code to ensure high availability for data access service.

4.5 DBGen Modifications

The version number, release number, modification number, and patch level of DBGen must be disclosed. Any modifications to the DBGen (see Clause 4.2.1) source code (see Appendix D) must be reported in the supporting files archive.

The supplied DBGen version 3.0.0 was used, no modifications were made.

4.6 Database Load Time

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

See Numerical Quantities Summary in the Executive Summary.

4.7 Data Storage Ratio

The data storage ratio must be disclosed. It is computed by dividing the total data storage of the priced configuration (expressed in GB) by the size chosen for the test database as defined in Clause 4.1.3.1. Let r be the ratio. The reported value for r must be rounded to the nearest 0.01. That is, reported value=round(r,2). For example, a system configured with 96 disks of 2.1 GB capacity for a 100GB test database has a data storage ratio of 2.02.

For the reporting of configured disk capacity, gigabyte (GB) is defined to be 2^{30} bytes.

Server Node	Total Storage Capacity (GB)
OFS	38,000
OceanBase Compute Node	40 x 64 = 2,560

Total disk capacity: 40,560 GB Scale factor: 30,000
The database storage ratio is 1.35.

4.8 Database Load Mechanism Details and Illustration

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

Figure 2 shows the database build procedure. The raw data flat files were created using DBGen and stored on Cloud OceanBase’s Compute Instance’s local disk. The configuration for loading data is disclosed in the Supporting Files Archive.

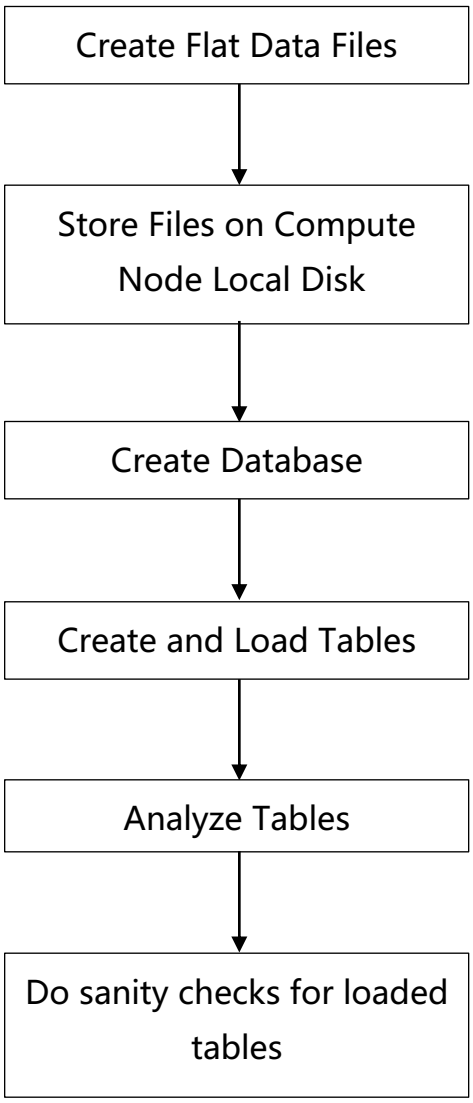


Fig. 2. Block Diagram of Database Loading Procedure.

4.9 Qualification Database Configuration

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

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

4.10 Memory to Database Size Percentage

The memory to database size percentage must be disclosed. It is computed by multiplying by 100 the total memory size priced on the SUT (see clause 6.2.1) and dividing this number by the size chosen for the test database as defined in Clause 4.1.3.1. Let r be this ratio. The reported ratio must be rounded to the nearest 0.1. That is, reported value=round(r,1). For example, a system configured with 256GB of memory for a 1000GB test database has a memory/database size percentage of 25.6.

Available Memory: 49,152 GB Scale Factor: 30,000
The memory to database size percentage is 163.8%.

5. Clause 5: Performance Metrics and Execution Rules Related Items

5.1 System Activity between Load and Performance Tests

Any system activity on the SUT that takes place between the conclusion of the load test and the beginning of the performance test must be fully reported in the supporting files archive including listings of scripts, command logs and system activity.

There is no activity on the SUT between the conclusion of the load test and the beginning of the performance test.

5.2 Steps in the Power Test

The details of the steps followed to implement the power test (e.g., system boot, database restart, etc.) must be reported in the supporting files archive.

The following steps were used to implement the power test:

- RF1 Refresh Function from the refresh stream.
- Query Execution from query stream 0.
- RF2 Refresh Function from the refresh stream.

5.3 Timing Interval for Each Query and Refresh Functions

The timing intervals (see Clause 5.3.7) for each query and for both refresh functions must be reported for the power test. The output for each query and for both refresh functions must be reported in the supporting files archive.

See the Numerical Quantities Summary in the Executive Summary at the beginning of this report.

5.4 Number of Streams for the Throughput Test

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

10 query streams were used for the throughput test.

5.5 Start and End Date/Time of Each Query Stream

The start time and finish time for each query stream for the throughput test must be disclosed. The output for each query stream for the throughput test must be reported in the supporting files archive.

See the Numerical Quantities Summary in the Executive Summary at the beginning of this report.

5.6 Total Elapsed Time of the Measurement Interval

The total elapsed time of the measurement interval (see Clause 5.3.6) must be disclosed for the throughput test.

See the Numerical Quantities Summary in the Executive Summary at the beginning of this report.

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

The start time and, finish time for each refresh function in the refresh stream for the throughput test must be disclosed. The output of each refresh function in the refresh stream for the throughput test must be reported in the supporting files archive.

See the Numerical Quantities Summary in the Executive Summary at the beginning of this report.

5.8 Timing Metric Precision

The start time and finish time for each query and refresh stream shall be reported to the hundredth of a second. If times are measured with the precision greater than one hundredth of a second, the reported times shall be truncated to the hundredth of a second.

See the Numerical Quantities Summary in the Executive Summary at the beginning of this report.

5.9 Performance Metrics

The computed performance metric, related numerical quantities and the price/performance metric must be disclosed.

See the Numerical Quantities Summary in the Executive Summary at the beginning of this report.

5.10 The Performance Metric and Numerical Quantities from Both Runs

The performance metric (QphH@Size) and the numerical quantities (TPC-H Power@Size and TPC-H Through-put@Size) from both of the runs must be disclosed.

Run ID	QphH@30,000G	QppH@30,000G	QthH@30,000G
Run 1	15,594,638.8	15,476,210.0	15,713,973.9
Run 2	15,265,305.7	14,788,059.7	15,757,953.6

5.11 System Activity between Performance Tests

Any activity on the SUT that takes place between the conclusion of Run1 and the beginning of Run2 must be fully disclosed including system activity, listings of scripts or command logs along with any system reboots or database restarts.

A major compaction is performed between Run 1 and Run 2

5.12 Documentation to satisfy Clause 5.2.7

All documentation necessary to satisfy Clause 5.2.7 must be made available upon request.

OceanBase document can be obtained from www.oceanbase.com

5.13 Query Output Validation

The output of the Query Output Validation Test must be reported in the supporting files archive.

The output of the validation test is available in the Supporting Files Archive.

6. Clause 6: SUT and Driver Implementation Related Items

6.1 Driver

A detailed textual description of how the driver performs its functions, how its various components interact and any product functionalities or environmental settings on which it relies and all related source code, scripts and configuration files must be reported in the supporting files archive. The information provided should be sufficient for an independent reconstruction of the driver.

All stream executions are performed by a script. QGEN is used to produce query text. For each power-test run:

1. A shell script is started, executes RF1 and then waits for the query stream to complete.
2. A shell script is started, executes the 22 queries in the required order for stream 0 and then signals to the shell script started in step 1.

3. The shell script started in step 1 is released and executes RF2.

For each throughput-test run:

1. The queries as generated by QGEN are submitted in the order defined by Clause 5.3.5.4 from the driver in several streams (the number of streams is listed in the Numerical Quantities) and runs concurrently.
2. Then throughput RF function pairs are scheduled to run sequentially in one update stream.

The source code of the used scripts are disclosed in the Supporting Files Archive.

6.2 Implementation Specific Layer (ISL)

If an implementation specific layer is used, then a detailed description of how it performs its functions, how its various components interact and any product functionalities or environmental setting on which it relies must be disclosed. All related source code, scripts and configuration files must be reported in the supporting files archive. The information provided should be sufficient for an independent reconstruction of the implementation specific layer.

The scripts used to implement the ISL are available in the Supporting Files Archive.

6.3 Profile-Directed Optimization

If profile-directed optimization as described in Clause 5.2.9 is used, such use must be disclosed. In particular, the procedure and any scripts used to perform the optimization must be reported in the supporting files archive.

Profile-directed optimization was not used.

7 Clause 7: Pricing

7.1 Priced Configuration

The pricing methodology used for pricing the Priced Configuration is the "Default 3-Year Pricing Methodology", as defined in the current revision of the TPC Pricing specification.

The system to be priced shall include the hardware, Licensed Compute Services and software components present in the System Under Test (SUT), a communication interface that can support user interface devices, additional operational components configured on the test system, and maintenance on all of the above:

- *System Under Test.*
- *User Interface Devices and Communications.*
- *Database Storage and Recovery Log.*

- *Additional Operational Components.*
- *Software.*

A 3-Year Pricing Methodology is used for this benchmark. A detailed list of hardware and software used in the priced system is included in the pricing sheet in the Executive Summary at the beginning of this report. The price quotations are included in Appendix A.

7.2 Availability Date and Orderability Date

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

All components of the priced configuration will be available for purchase on 07/31/2021.

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 priced components will be orderable on 07/31/2021.

7.3 Country-Specific Pricing

The Priced Locale and Priced Currency of the Priced Configuration must be disclosed.

The configuration is priced in CNY for the China market.

8. Clause 8: Full Disclosure

8.1 Supporting Files Index Table

An index for all files and/or directories included in the Supporting Files Archive as required by Clauses 8.3.2 through 8.3.8 must be provided in the report.

Clause	Description	Archive Files	Path Name
1	Parameter Settings	benchmark_scripts.zip	RUN/params.log

	DB Creation Scripts		KIT/sql/create_user_mysql.sql KIT/sql/create_schema_mysql.sql KIT/sql/analyze_database.sql
	System Verification		KIT/script/tools/hwinfo.sh
	Toolkit Common Scripts		KIT/script
2	Minor query modifications	benchmark_scripts.zip	KIT/tpch_archives/tpch_3_0_0.zip.patch
3	ACID Test Scripts	benchmark_scripts.zip	KIT/ACID/
	ACID Test Results		ACID/
4	Database Load Scripts	benchmark_scripts.zip	KIT/script/load_init.sh
	Qualification Test Results		VLD/
5	Query Output Results	run1results.zip	RUN//run1/
		run2results.zip	RUN//run2/
6	Source Codes and Scripts of Driver	benchmark_scripts.zip	KIT/script/query_streams
7	There are no files to be included for Clause 7.	N/A	N/A
8	Query Parameters & Seeds	benchmark_scripts.zip	RUN/run1/substitution_parameters.txt
	Executable Query Text		RUN/run1/stream*.sql
	RF function source code		KIT/script/tpc_h_full.sh KIT/script/refresh_functions/

9. Clause 9: Audit Related Items

9.1 Auditor's Report

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

This implementation of the TPC Benchmark™ H was audited by Doug Johnson of InfoSizing, a certified TPC-H auditor. Further information regarding the audit process may be obtained from:

Doug Johnson
InfoSizing (www.sizing.com)

63 Lourdes Dr
Leominster, MA 01453
(978) 343-6562

TPC Benchmark™ H Full Disclosure Report and other information can be downloaded from the Transaction Processing Performance Council web site at www.tpc.org.

InfoSizing

TPC Certified Auditor

Benchmark sponsor:	Mengmeng Chen Huanglongguoji Building I, Xihu District, Hangzhou 310000, Zhejiang, China

May 19, 2021

I verified the TPC Benchmark H (TPC-H™ v3.0.0) performance of the following configuration:

Platform:	Cloud OceanBase (w/ 64 Computing Instances)
Operating System:	Alibaba Group Cloud Linux Server release 2.19
Database Manager:	OceanBase V3.2
Other Software:	OceanBase File System V1.0

The results were:

Performance Metric	**15,265,305.7 QphH@30,000GB**
TPC-H Power	14,788,059.7
TPC-H Throughput	15,757,953.6
Database Load Time	00d 02h 47m 41s

Server	**Cloud OceanBase (w/ 64 Computing Instances):**		
CPUs	80x vCPU (per Computing Instance)		
Memory	768 GiB (per Computing Instance)		
Storage	***Qty***	***Size***	***Type***
	1	40 GiB	Local SSD (per Computing Instance)
	1	38,000 GiB	OFS Storage Service

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 database records were defined with the proper layout and size
- The database population was generated using DBGen
- The database was properly scaled to 30,000GB and populated accordingly
- The compliance of the database auxiliary data structures was verified
- The database load time was correctly measured and reported

- The required ACID properties were verified and met
- The query input variables were generated by QGen
- The query text was produced using minor modifications and no query variant
- The execution of the queries against the SF1 database produced compliant answers
- A compliant implementation specific layer was used to drive the tests
- The throughput tests involved 10 query streams
- The ratio between the longest and the shortest query was such that no query timings were adjusted
- The execution times for queries and refresh functions were correctly measured and reported
- The repeatability of the measured results was verified
- The system pricing was verified for major components and maintenance
- The major pages from the FDR were verified for accuracy

Additional Audit Notes:

None.

Respectfully Yours,

Doug Johnson, TPC Certified Auditor

Appendix A: Price Quotes