TaurusDB

Performance White Paper

Issue 01

Date 2025-02-07

HUAWEI

HUAWEI CLOUD COMPUTING TECHNOLOGIES CO., LTD.

Trademarks and Permissions


and other Huawei trademarks are the property of Huawei Technologies Co., Ltd.
All other trademarks and trade names mentioned in this document are the property of their respective holders.

Notice

The purchased products, services and features are stipulated by the contract made between Huawei Cloud and the customer. All or part of the products, services and features described in this document may not be within the purchase scope or the usage scope. Unless otherwise specified in the contract, all statements, information, and recommendations in this document are provided "AS IS" without warranties, guarantees or representations of any kind, either express or implied.

The information in this document is subject to change without notice. Every effort has been made in the preparation of this document to ensure accuracy of the contents, but all statements, information, and recommendations in this document do not constitute a warranty of any kind, express or implied.

Contents

1 Test Method

TaurusDB is an enterprise-grade cloud-native database fully compatible with MySQL. It decouples compute from storage and supports up to 128 TB of storage per instance. With TaurusDB, there is no need to worry about data loss. It provides the high availability and superior performance of commercial databases at the price of open-source databases.

Test Environment

The TaurusDB test environment is as follows:

- Region: AP-Singapore
- AZ: multiple AZs
- TaurusDB instance: an instance with a primary node and a read replica
- Elastic Cloud Server (ECS): general computing-plus | c7.8xlarge.4 | 32 vCPUs | 128 GB, CentOS 7.6 (64-bit). The ECS and instance nodes are in the same AZ. Bind an EIP to the ECS because additional compilation tools need to be installed on stress testing tools.

Test Tool

Table 1-1 Test tool

Tool	Description	Version
Sysbench	Sysbench is a multi-threaded benchmark tool based on LuaJIT. It is most frequently used for database benchmarks. With sysbench, you can quickly get an impression of database performance. For details, visit **https://github.com/akopytov/sysbench**	**Sysbench 1.0.18**

Perform the following commands to install sysbench:

Log in to an ECS and download the sysbench software package.

wget https://codeload.github.com/akopytov/sysbench/zip/refs/tags/1.0.18

yum install -y autoconf libtool mysql mysql-devel vim unzip

Decompress the software package.

unzip 1.0.18

Install the software package.

cd sysbench-1.0.18

./autogen.sh

./configure

make

make install

Test Procedure

NOTICE

The following tests are performed on an ECS. Replace the number of concurrent threads, connection IP address, connection port, username, and user password based on the site requirements.

Performance test data (including SQL) is automatically generated by the sysbench tool.

The ECS and the instance are in the same AZ.

To ensure that sysbench runs properly in high-concurrency scenarios (concurrent requests: 512-1000), increase the value of **max_prepared_stmt_count**. The recommended value is **1048576**. Too many Prepare statements consume a lot of memory space, resulting in out-of-memory (OOM). For an instance with 4 vCPUs and 16 GB memory, set this parameter to **400000**.

Testing write-only performance

Step 1 Import data.

1. Create the test database **sbtest**.

   mysql -u<*user*>**-P** <*port*> **-h** <*host*> **-p -e "create database sbtest"**

2. Import the test background data to the **sbtest** database.

   sysbench --db-driver=mysql --mysql-host=<*host*> **--mysql-port=**<*port*> **--mysql-user=**<*user*> **--mysql-password=**<*password*> **--mysql-db=sbtest --table_size=25000 --tables=250 --threads=**<*thread_num*> **oltp_common prepare**

Step 2 Test the write-only performance. The process takes about 10 minutes.

sysbench --db-driver=mysql --mysql-host=<*host*> **--mysql-port=**<*port*> **--mysql-user=**<*user*> **--mysql-password=**<*password*> **--mysql-db=sbtest --table_size=25000 --tables=250 --time=600 --threads=**<*thread_num*> **--percentile=95 --report-interval=1 oltp_write_only run**

Step 3 Delete data.

sysbench --db-driver=mysql --mysql-host=<*host*> --mysql-port=<*port*> --mysql-user=<*user*> --mysql-password=<*password*> --mysql-db=sbtest --table_size=25000 --tables=250 --threads=<*thread_num*> oltp_common cleanup

----End

Testing read-only performance

Step 1 Import data.

1. Create the test database **sbtest**.

   mysql -u<*user*> -P<*port*> -h<*host*> -p -e "create database sbtest"

2. Import the test background data to the **sbtest** database.

   sysbench --db-driver=mysql --mysql-host=<*host*> --mysql-port=<*port*> --mysql-user=<*user*> --mysql-password=<*password*> --mysql-db=sbtest --table_size=25000 --tables=250 --threads=<*thread_num*> oltp_common prepare

Step 2 Test the read-only performance. The process takes about 10 minutes.

sysbench --db-driver=mysql --mysql-host=<*host*> --mysql-port=<*port*> --mysql-user=<*user*> --mysql-password=<*password*> --mysql-db=sbtest --table_size=25000 --tables=250 --time=600 --range_selects=0 --skip-trx=1 --threads=<*thread_num*> --percentile=95 --report-interval=1 oltp_read_only run

Step 3 Delete data.

sysbench --db-driver=mysql --mysql-host=<*host*> --mysql-port=<*port*> --mysql-user=<*user*> --mysql-password=<*password*> --mysql-db=sbtest --table_size=25000 --tables=250 --threads=<*thread_num*> oltp_common cleanup

----End

Testing read/write performance

Step 1 Import data.

1. Create the test database **sbtest**.

   mysql -u<*user*> -P<*port*> -h <*host*> **-p -e "create database sbtest"**

2. Import the test background data to the **sbtest** database.

   sysbench --db-driver=mysql --mysql-host=<*host*> --mysql-port=<*port*> --mysql-user=<*user*> --mysql-password=<*password*> --mysql-db=sbtest --table_size=250000 --tables=25 --threads=<*thread_num*> oltp_common prepare

Step 2 Test the read/write performance. The process takes about 10 minutes.

sysbench --db-driver=mysql --mysql-host=<*host*> --mysql-port=<*port*> --mysql-user=<*user*> --mysql-password=<*password*> --mysql-db=sbtest --table_size=250000 --tables=25 --time=600 --threads=<*thread_num*> --percentile=95 --report-interval=1 oltp_read_write run

Step 3 Delete data.

sysbench --db-driver=mysql --mysql-host=<*host*> --mysql-port=<*port*> --mysql-user=<*user*> --mysql-password=<*password*> --mysql-db=sbtest --table_size=250000 --tables=25 --threads=<*thread_num*> oltp_common cleanup

----End

Test Metrics

- Transactions per second (TPS) indicates the number of transactions executed per second.
- Queries per second (QPS) indicates the number of SQL statements, including INSERT, SELECT, UPDATE, and DELETE statements, executed per second.

2 Performance Data

2.1 Read/Write Mode

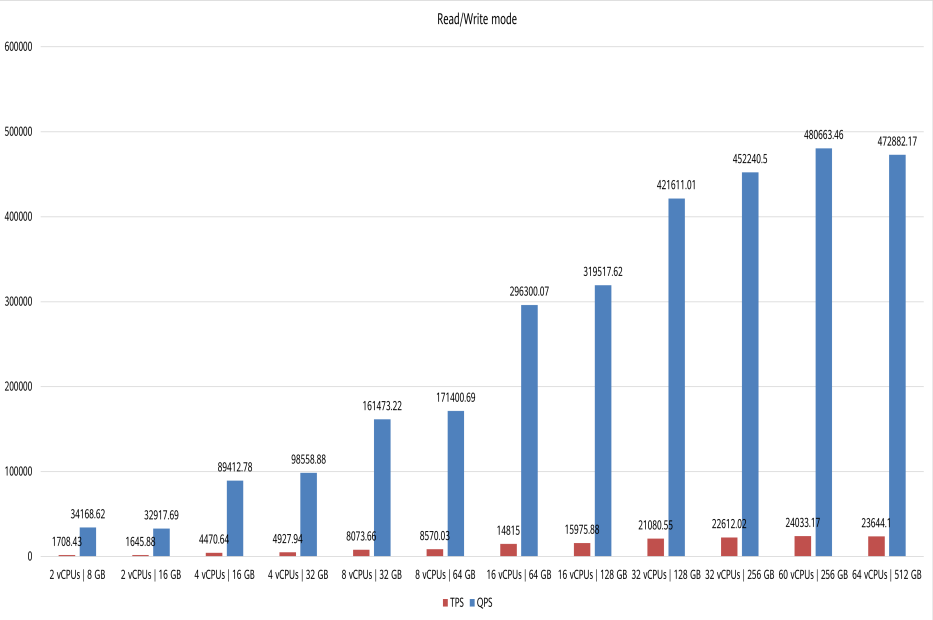
Dedicated DB Instance Test Data

Table 2-1 Test data in read/write mode (x86 architecture, multi-AZ deployment)

Mode	Tables	Table Rows	Threads	Instance Specifications	TPS	QPS
Read/ Write	25	250,000	64	2 vCPUs | 8 GB	1,708.43	34,168.62
			64	2 vCPUs | 16 GB	1,645.88	32,917.69
			128	4 vCPUs | 16 GB	4,470.64	89,412.78
			128	4 vCPUs | 32 GB	4,927.94	98,558.88
			64	8 vCPUs | 32 GB	8,073.66	161,473.22
			64	8 vCPUs | 64 GB	8,570.03	171,400.69
			256	16 vCPUs | 64 GB	14,815	296,300.07
			256	16 vCPUs | 128 GB	15,975.88	319,517.62
			512	32 vCPUs | 128 GB	21,080.55	421,611.01
			512	32 vCPUs | 256 GB	22,612.02	452,240.5
			512	60 vCPUs | 256 GB	24,033.17	480,663.46
			512	64 vCPUs | 512 GB	23,644.1	472,882.17

Dedicated DB Instance Test Results

Figure 2-1 Test results



2.2 Read-only Mode

Dedicated DB Instance Test Data

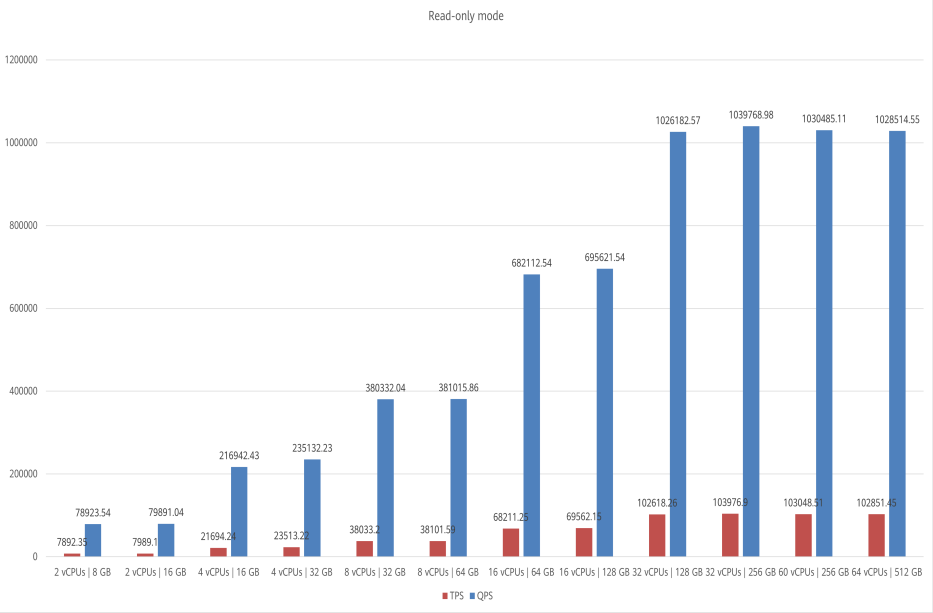
Table 2-2 Test data in read-only mode (x86 architecture, multi-AZ deployment)

Mode	Tables	Table Rows	Threads	Instance Specifications	TPS	QPS
Read-only	250	25,000	64	2 vCPUs | 8 GB	7,892.35	78,923.54
			64	2 vCPUs | 16 GB	7,989.1	79,891.04
			64	4 vCPUs | 16 GB	21,694.24	216,942.43
			64	4 vCPUs | 32 GB	23,513.22	235,132.23
			512	8 vCPUs | 32 GB	38,033.2	380,332.04
			512	8 vCPUs | 64 GB	38,101.59	381,015.86
			1,000	16 vCPUs | 64 GB	68,211.25	682,112.54

			1,000	16 vCPUs | 128 GB	69,562.15	695,621.54
			1,000	32 vCPUs | 128 GB	102,618.26	1,026,182.57
			1,000	32 vCPUs | 256 GB	103,976.9	1,039,768.98
			1,000	60 vCPUs | 256 GB	103,048.51	1,030,485.11
			1,000	64 vCPUs | 512 GB	102,851.45	1,028,514.55

Dedicated DB Instance Test Results

Figure 2-2 Test results



2.3 Write-only Mode

Dedicated DB Instance Test Data

Table 2-3 Test data in write-only mode (x86 architecture, multi-AZ deployment)

Mode	Tables	Table Rows	Threads	Instance Specifications	TPS	QPS
Write-only	250	25,000	128	2 vCPUs | 8 GB	5,473.22	32,839.34

			128	2 vCPUs | 16 GB	5,685.26	34,111.57
			256	4 vCPUs | 16 GB	16,312.03	97,872.16
			256	4 vCPUs | 32 GB	17,025.3	102,151.81
			512	8 vCPUs | 32 GB	30,702.7	184,216.21
			512	8 vCPUs | 64 GB	32,941.36	197,648.16
			512	16 vCPUs | 64 GB	55,022.87	330,137.28
			512	16 vCPUs | 128 GB	61,818.75	370,912.52
			512	32 vCPUs | 128 GB	73,708.67	442,252.03
			512	32 vCPUs | 256 GB	75,602.57	453,615.46
			512	60 vCPUs | 256 GB	77,489.58	464,937.48
			512	64 vCPUs | 512 GB	77,939.14	467,634.84

Dedicated DB Instance Test Results

Figure 2-3 Test results