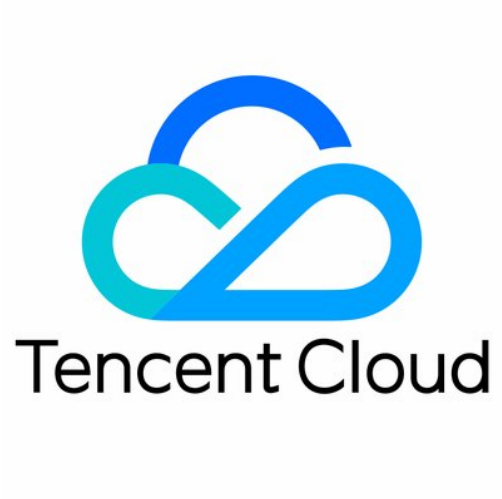
TencentDB for MySQL

White Paper

Product Documentation



Contents

White Paper
Performance White Paper
Performance Test Report
Performance Overview

Last updated：2023-09-13 15:48:59

TencentDB for MySQL is a high-performance distributed data storage service developed by Tencent Cloud based on the open-source MySQL database. It enables you to set up, manipulate, and scale relational databases in the cloud more easily. TencentDB for MySQL improves performance and stability by leveraging many core features of TXSQL, Tencent Cloud's proprietary kernel, such as enterprise-grade transparent data encryption, audit, and thread pool. Continuously tested and optimized by a professional team, TencentDB for MySQL can flexibly and efficiently process transactions. It also features advanced and complete security protection and data encryption capabilities, which comprehensively ensure the secure and smooth operations of your business.

This docuement describes how to use the performance testing tool SysBench to perform read-write performance tests on TencentDB for MySQL 5.6, 5.7, and 8.0 in full cache and disk I/O scenarios.

For more information about the test environment, test tools, test methods and test metrics, see following documents:

Test Environment

Test Tools

Test Methods

Test Metrics

For the performance test results of TencentDB for MySQL, see the following documentations:

MySQL 8.0 Test Results

MySQL 5.7 Test Results

MySQL 5.6 Test Results

Test Methods

Test Environment

Last updated：2024-07-23 17:40:42

This document describes the environment used for the TencentDB for MySQL performance test.

Region/AZ: Beijing - Beijing Zone 7

Client: S5.8XLARGE64 (32-core 64 GB Standard S5)

Client OS: CentOS 8.2 64-bit

Network: Both the CVM and TencentDB for MySQL instances are in the same VPC subnet.

The information of the tested TencentDB for MySQL instance is as follows:

Storage type: Local SSD disk

Instance specification: General

Parameter template: High-performance template

Test Tools

Last updated：2024-07-23 17:41:00

This document describes how to install SysBench, a performance testing tool for TencentDB for MySQL, in a CVM instance.

SysBench Overview

SysBench is a modular, cross-platform, and multi-threaded benchmark tool for evaluating OS parameters that are important for a system running a database under intensive load. The idea of this benchmark suite is to quickly get an impression about system performance without setting up complex database benchmarks or even without installing a database at all.

SysBench Testing Model

In a standard mixed OLTP read/write scenario of SysBench, a transaction contains 18 read/write SQL statements.
In a standard OLTP read-only scenario of SysBench, a transaction contains 14 SQL read statements (ten primary key-based `POINT SELECT` queries and four range queries).
In a standard OLTP write-only scenario of SysBench, a transaction contains four write SQL statements (two `UPDATE` statements, one `DELETE` statement, and one `INSERT` statement).

SysBench Parameter Description

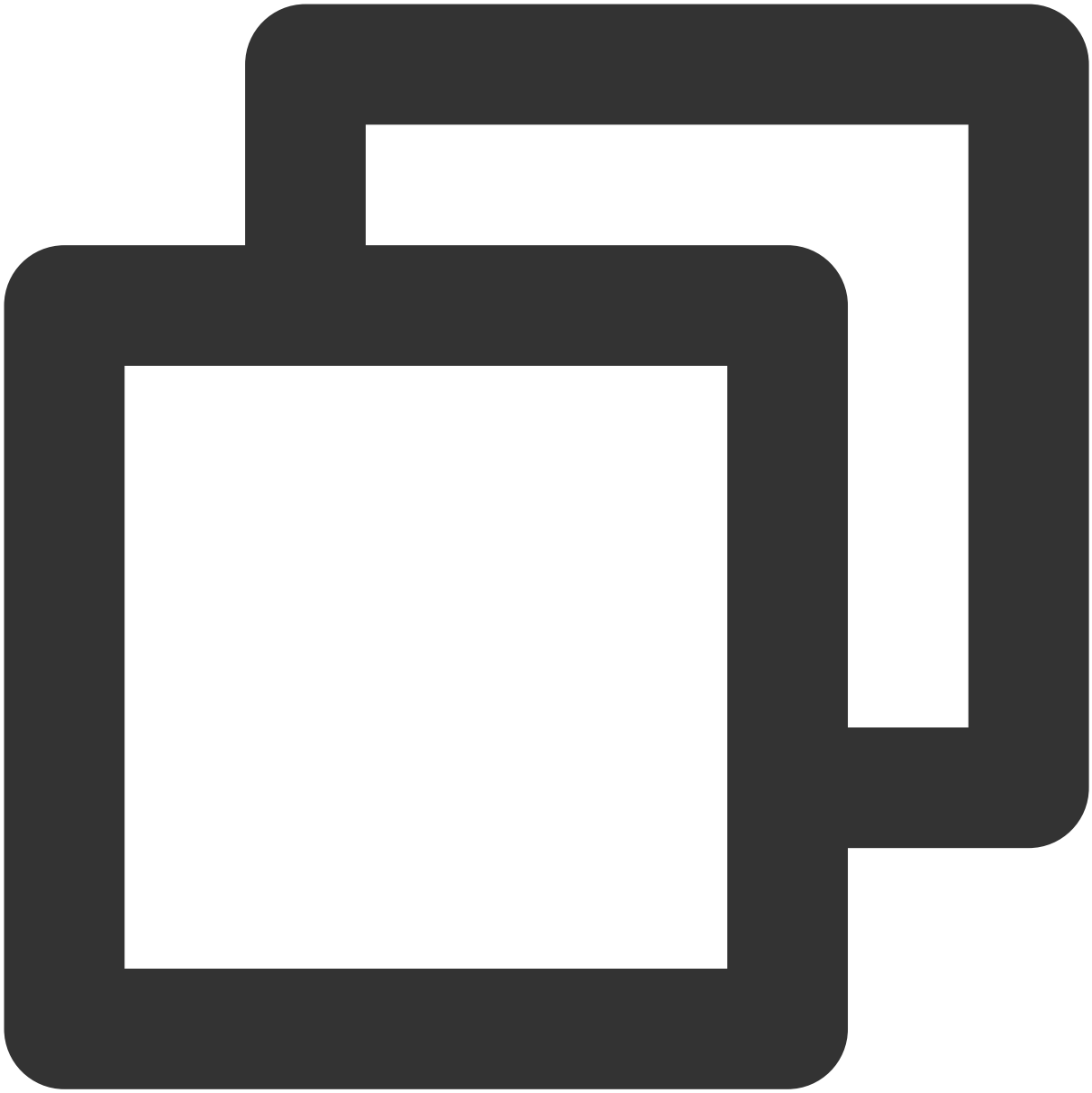
Parameter	Description
db-driver	Database engine
mysql-host	MySQL server host
mysql-port	MySQL server port
mysql-user	MySQL user
mysql-password	MySQL password
mysql-db	MySQL database name

table_size	Test table size
tables	Number of test tables
events	Number of test requests
time	Test time
threads	Number of test threads
percentile	The percentile range to be counted, which is 95% by default, i.e., the execution times of requests in 95% of the cases
report-interval	Interval for outputting a test progress report in seconds. 0 indicates to output only the final result but not the test progress report
skip-trx	Whether to skip transactions 1: Yes 0: No

How to Install

This stress test uses SysBench 1.0.20. For more information, see here.

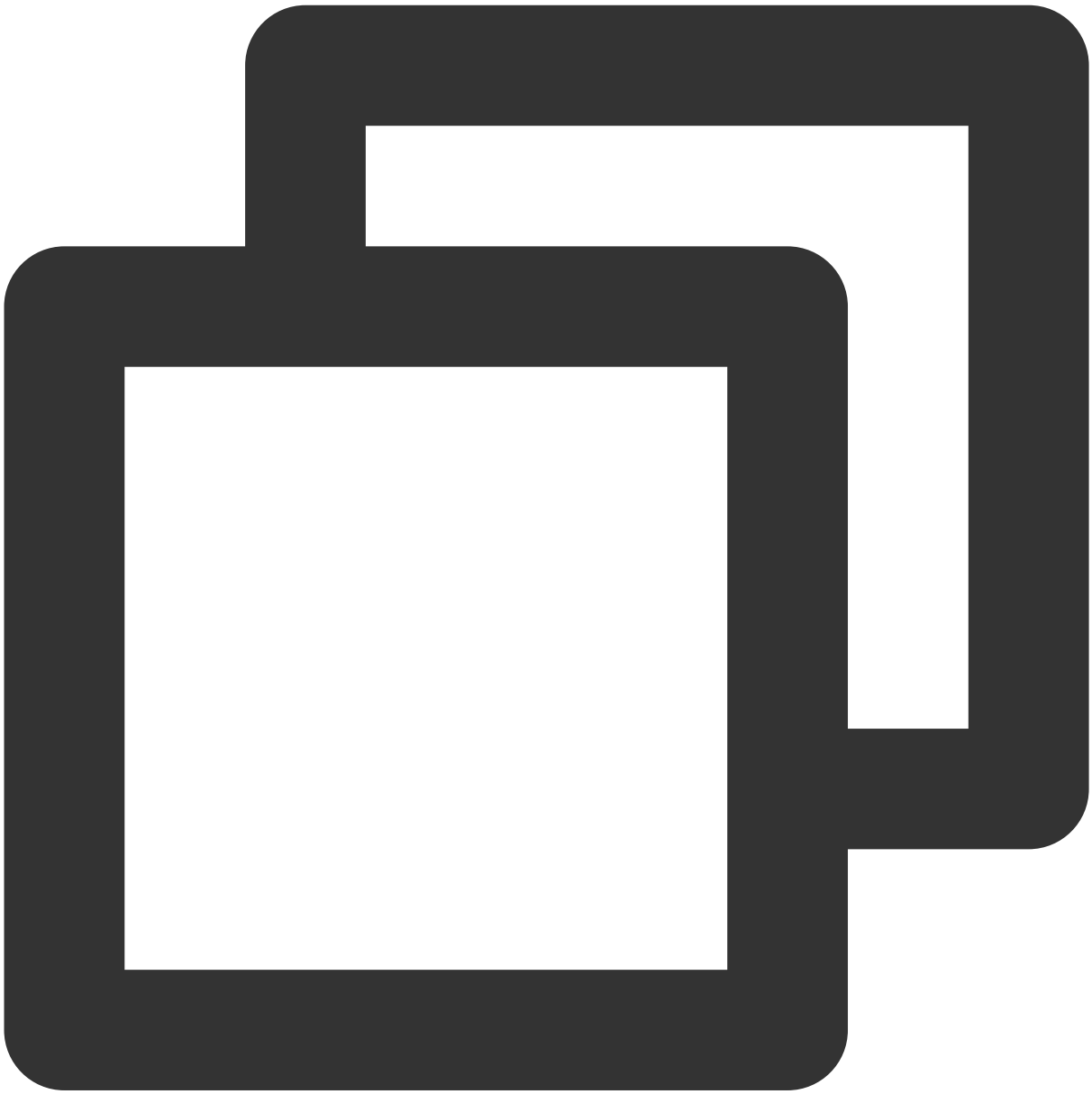
1. Run the following command to install SysBench in a CVM instance:



```
yum install gcc gcc-c++ autoconf automake make libtool bzr mysql-devel git mysql
git clone https://github.com/akopytov/sysbench.git
##Download SysBench from GitHub
cd sysbench
##Open the SysBench directory
git checkout 1.0.20
##Switch to SysBench 1.0.20
./autogen.sh
##Run `autogen.sh`
./configure --prefix=/usr --mandir=/usr/share/man
make
```

```
##Compile
make install
```

2. Run the following command to configure the client, so that the kernel can use all CPU cores to process data packets and reduce context switches within each CPU core.



```
sudo sh -c 'for x in /sys/class/net/eth0/queues/rx-*; do echo ffffffff>$x/rps_cpus;
sudo sh -c "echo 32768 > /proc/sys/net/core/rps_sock_flow_entries"
sudo sh -c "echo 4096 > /sys/class/net/eth0/queues/rx-0/rps_flow_cnt"
sudo sh -c "echo 4096 > /sys/class/net/eth0/queues/rx-1/rps_flow_cnt"
```

Note:

`ffffffff` indicates that 32 CPU cores are used (one `f` represents four CPU cores).

Test Methods

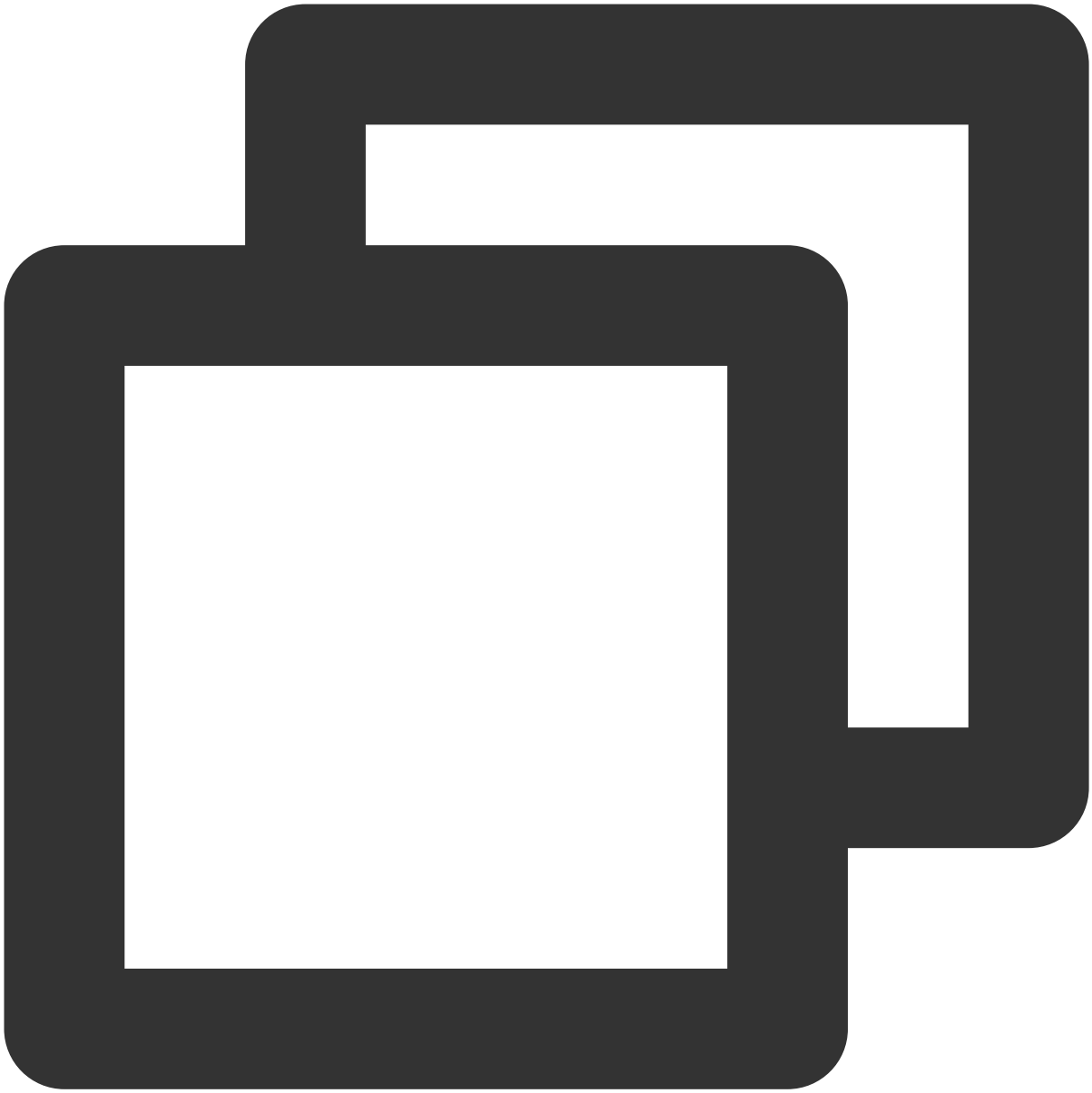
Last updated：2024-07-23 17:41:22

This document describes the method of TencentDB for MySQL performance test.

Directions

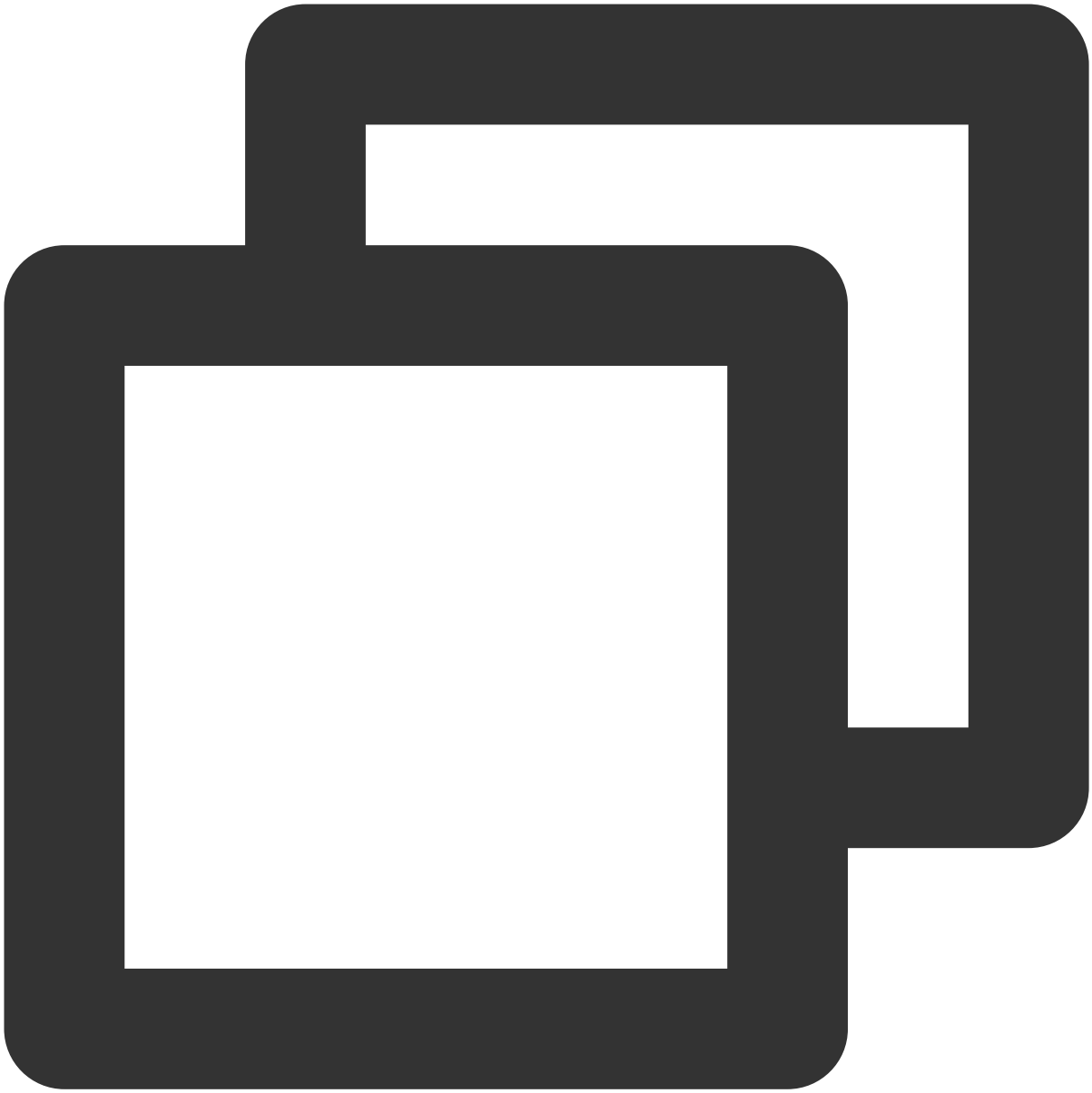
Use SysBench to test the mixed read/write performance of the TencentDB for MySQL instance.

1. Prepare the data.



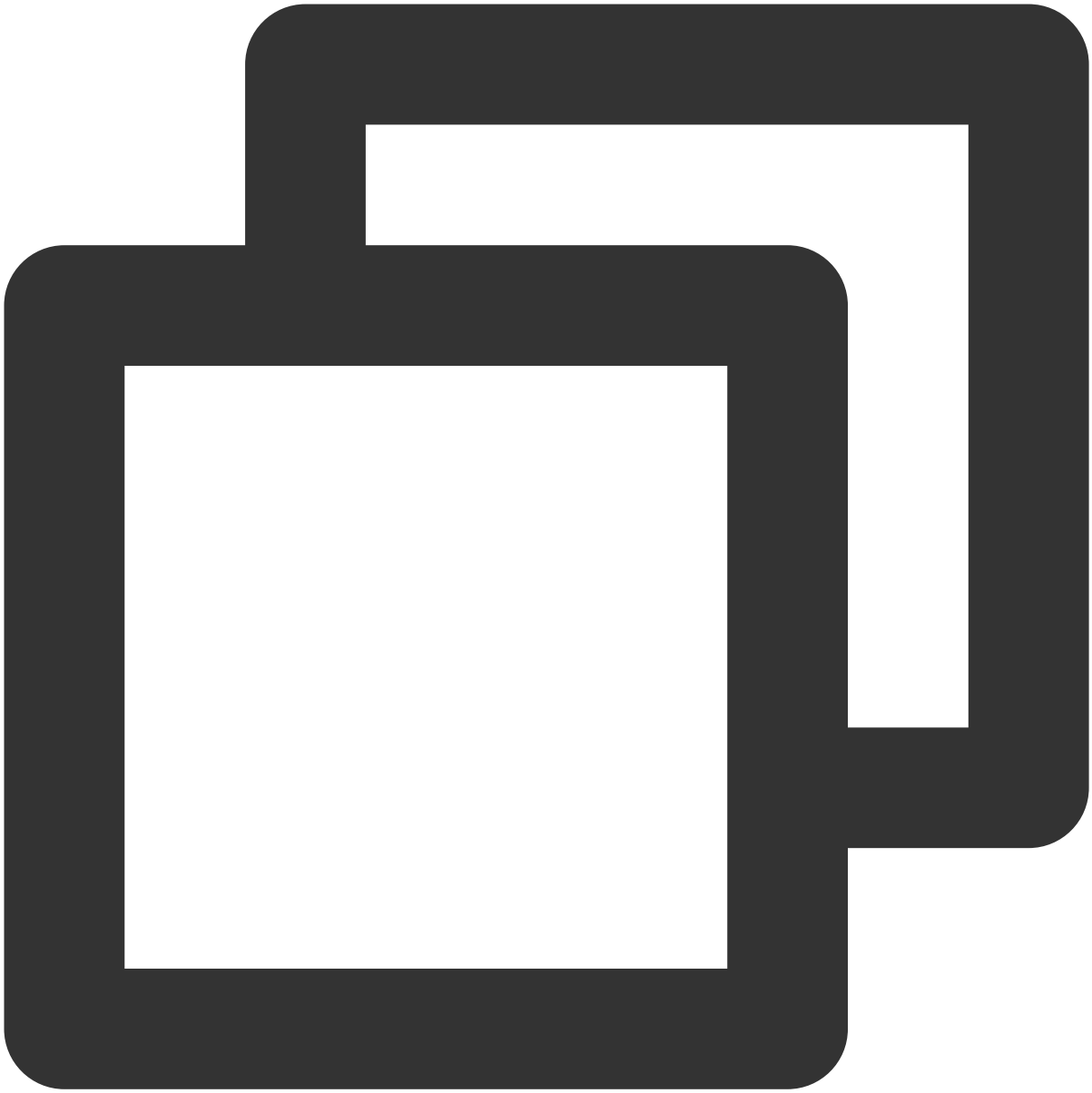
```
sysbench --db-driver=mysql --mysql-host=XXX --mysql-port=XXX --mysql-user=XXX --mys
```

2. Run the workload.



```
sysbench --db-driver=mysql  --mysql-host=XXX --mysql-port=XXX --mysql-user=XXX --my
```

3. Clear the data.



```
sysbench --db-driver=mysql  --mysql-host=XXX --mysql-port=XXX --mysql-user=XXX --my
```

Test Metrics

Last updated：2024-07-23 17:41:33

This document describes the metrics of TencentDB for MySQL performance test.

Metric	Definition
TPS (Transactions Per Second)	The number of successfully committed transactions executed by the database per second.
QPS (Queries Per Second)	The number of SQL statements executed by the database per second, including `INSERT`, `SELECT`, `UPDATE`, `DELETE`, and `COMMIT`.
Average Latency (avg_lat)	The average latency of all events in the database.
Concurrency	The number of concurrent requests initiated by the client during performance testing

Test Results

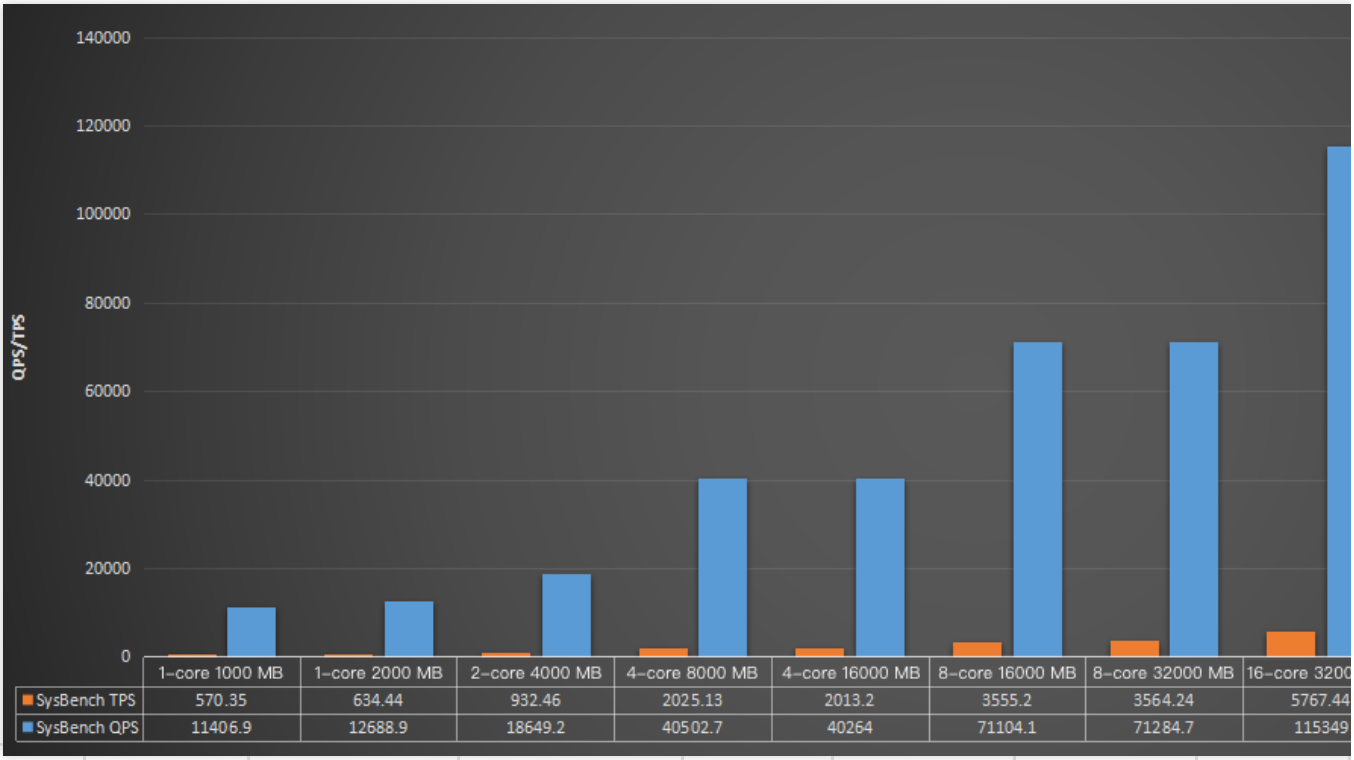
MySQL 8.0 Test Results

Last updated：2024-07-23 17:41:50

This document describes the performance test results of a general TencentDB for MySQL 8.0 instance.

Scenario 1: Full Cache

In the full cache scenario, as all data can be put into the cache, the disk doesn't need to be read and written to update the cache during queries.

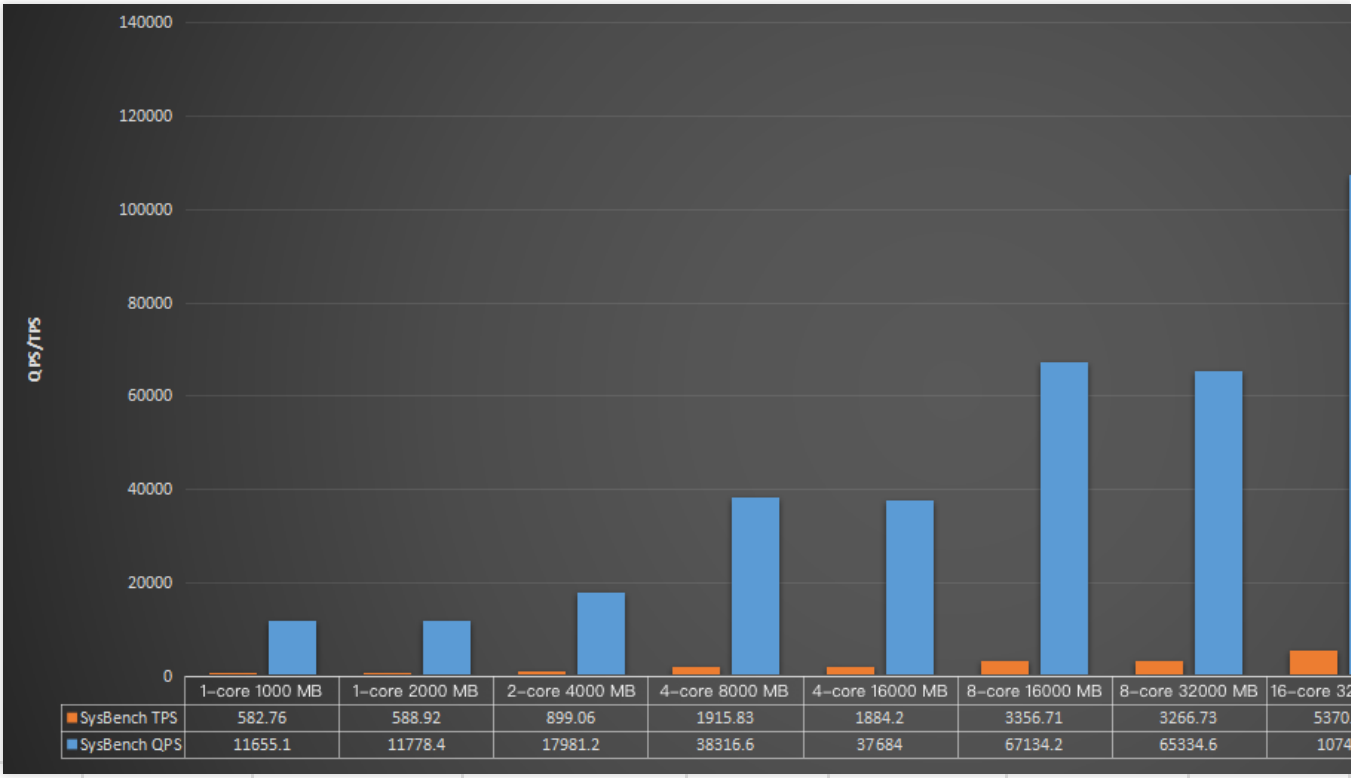


CPU (Core)	Memory (MB)	Concurrency	Data Volume/Table	Total Tables	SysBench TPS	SysBench QPS	avg_lat
1	1000	8	25000	150	570.35	11406.9	14.03
1	2000	8	25000	150	634.44	12688.9	12.61
2	4000	16	25000	150	932.46	18649.2	17.16
4	8000	32	25000	150	2025.13	40502.7	15.8
4	16000	32	25000	150	2013.2	40264	15.89

8	16000	64	25000	150	3555.2	71104.1	18
8	32000	64	25000	150	3564.24	71284.7	17.95
16	32000	128	25000	150	5767.44	115349	22.19
16	64000	128	25000	150	6558.19	131164	19.51
16	96000	128	25000	150	6517.01	130340	19.63
16	128000	128	25000	150	6539.98	130800	19.57

Scenario 2: Disk I/O

In the disk I/O scenario, as only part of the data can be put into the cache, the disk needs to be read and written to update the cache during queries.



CPU (Core)	Memory (MB)	Concurrency	Data Volume/Table	Total Tables	SysBench TPS	SysBench QPS	avg_lat
1	1000	8	800000	6	582.76	11655.1	13.73
1	2000	8	800000	12	588.92	11778.4	13.58
2	4000	16	800000	24	899.06	17981.2	17.8

4	8000	32	800000	48	1915.83	38316.6	16.7
4	16000	32	6000000	13	1884.2	37684	16.98
8	16000	64	6000000	13	3356.71	67134.2	19.06
8	32000	64	6000000	25	3266.73	65334.6	19.59
16	32000	128	6000000	25	5370.18	107404	23.83
16	64000	128	6000000	49	5910.85	118217	21.65
16	96000	128	6000000	74	5813.94	116279	22.01
16	128000	128	6000000	98	5700.06	114001	22.45

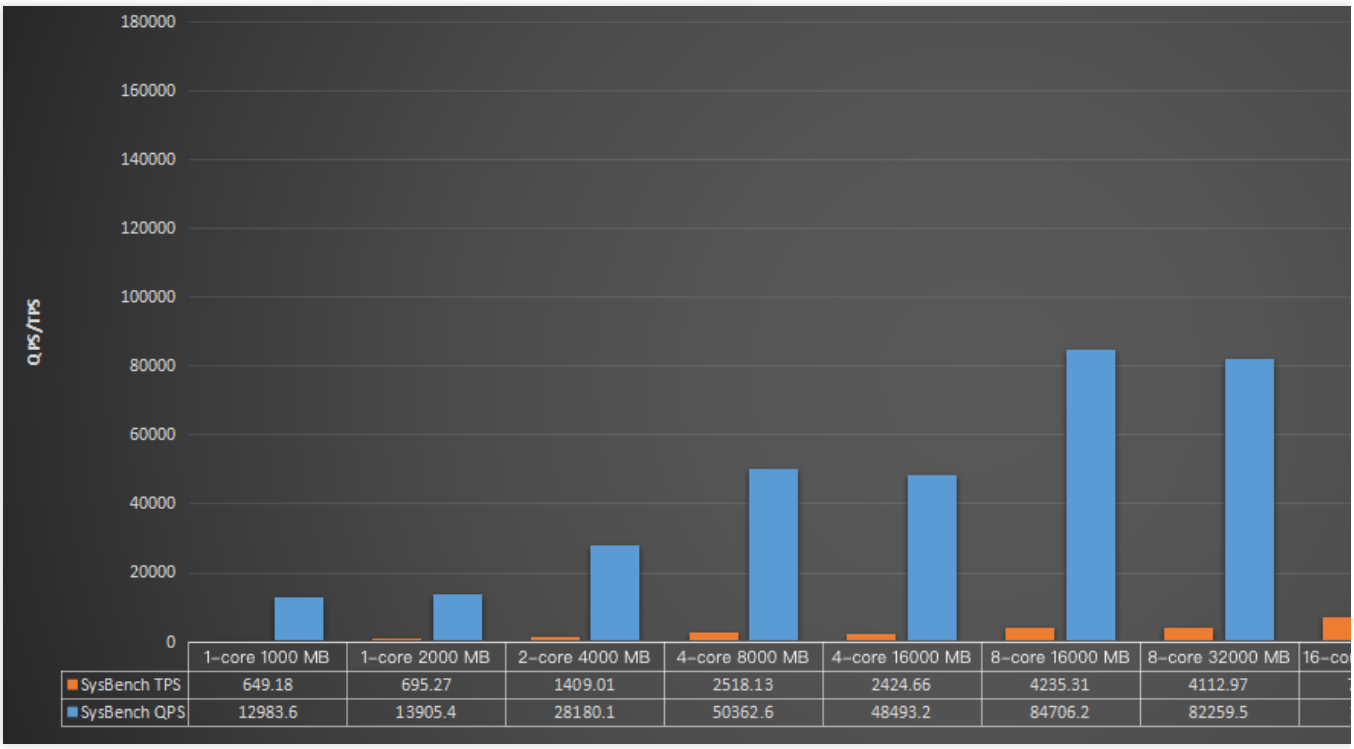
MySQL 5.7 Test Results

Last updated：2024-07-23 17:42:06

This document describes the performance test results of a general TencentDB for MySQL 5.7 instance.

Scenario 1: Full Cache

In the full cache scenario, as all data can be put into the cache, the disk doesn't need to be read and written to update the cache during queries.

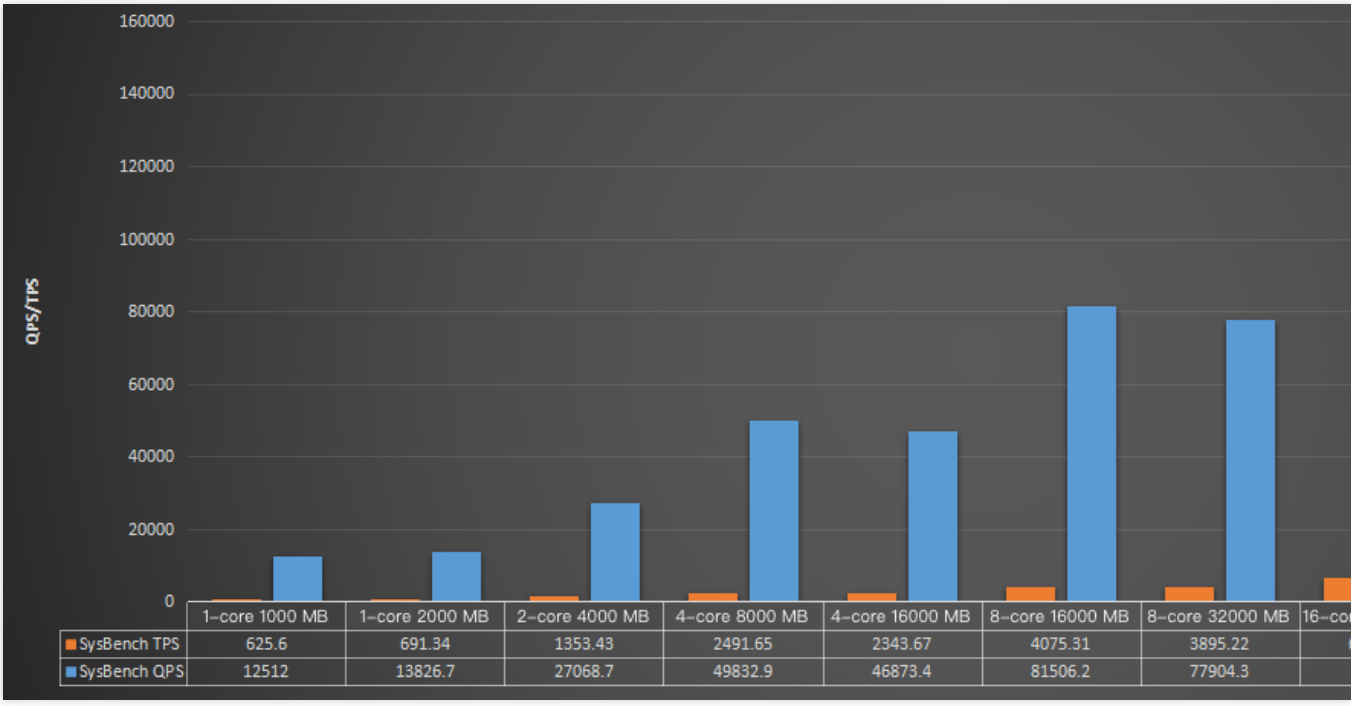


CPU (Core)	Memory (MB)	Concurrency	Data Volume/Table	Total Tables	SysBench TPS	SysBench QPS	avg_lat
1	1000	8	25000	150	649.18	12983.6	12.32
1	2000	8	25000	150	695.27	13905.4	11.51
2	4000	16	25000	150	1409.01	28180.1	11.35
4	8000	32	25000	150	2518.13	50362.6	12.71
4	16000	32	25000	150	2424.66	48493.2	13.2

8	16000	64	25000	150	4235.31	84706.2	15.11
8	32000	64	25000	150	4112.97	82259.5	15.56
16	32000	128	25000	150	7154.77	143095	17.88
16	64000	128	25000	150	7888.35	157767	16.22
16	96000	128	25000	150	7902.25	158045	16.19
16	128000	128	25000	150	7910.24	158205	16.17

Scenario 2: Disk I/O

In the disk I/O scenario, as only part of the data can be put into the cache, the disk needs to be read and written to update the cache during queries.



CPU (Core)	Memory (MB)	Concurrency	Data Volume/Table	Total Tables	SysBench TPS	SysBench QPS	avg_lat
1	1000	8	800000	6	625.6	12512	12.79
1	2000	8	800000	12	691.34	13826.7	11.57
2	4000	16	800000	24	1353.43	27068.7	11.82

4	8000	32	800000	48	2491.65	49832.9	12.84
4	16000	32	6000000	13	2343.67	46873.4	13.65
8	16000	64	6000000	13	4075.31	81506.2	15.7
8	32000	64	6000000	25	3895.22	77904.3	16.43
16	32000	128	6000000	25	6612.82	132256	19.35
16	64000	128	6000000	49	6995.9	139918	18.29
16	96000	128	6000000	74	6985.35	139707	18.32
16	128000	128	6000000	98	6980.42	139608	18.33

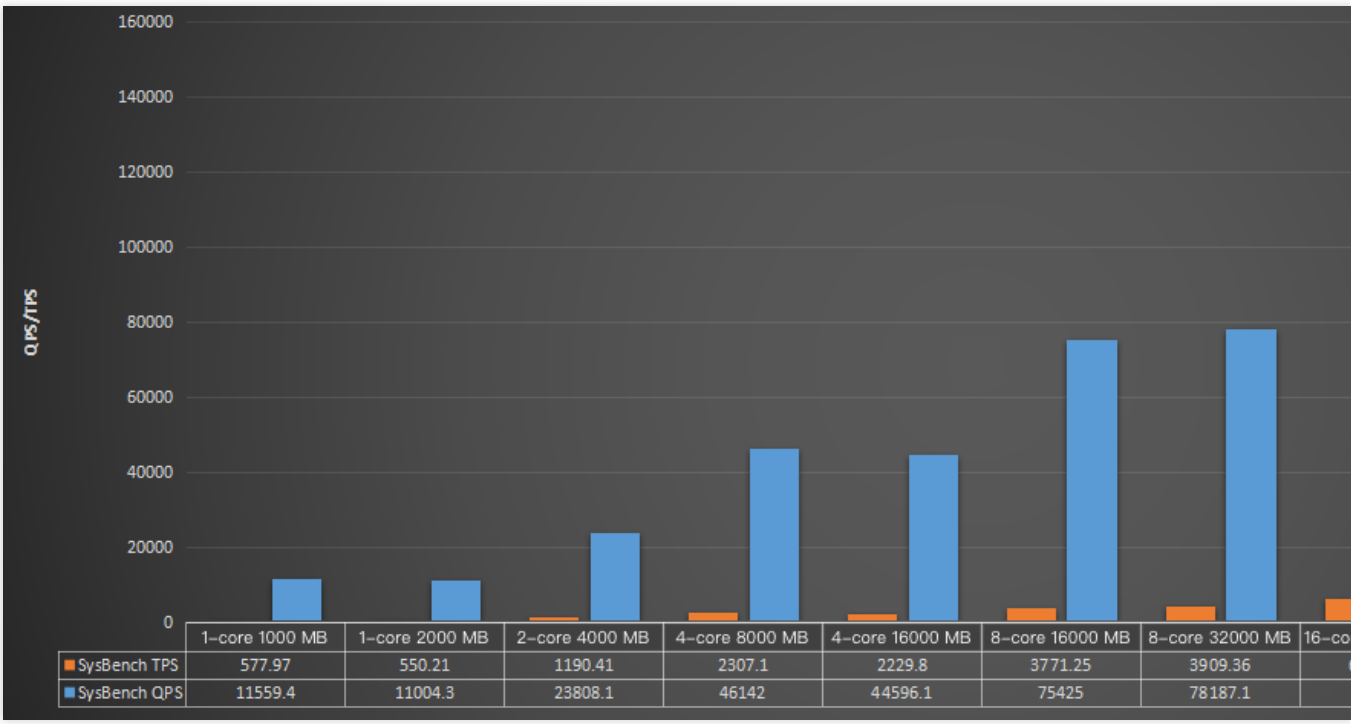
MySQL 5.6 Test Results

Last updated：2024-07-23 17:42:20

This document describes the performance test results of a general TencentDB for MySQL 5.6 instance.

Scenario 1: Full Cache

In the full cache scenario, as all data can be put into the cache, the disk doesn't need to be read and written to update the cache during queries.

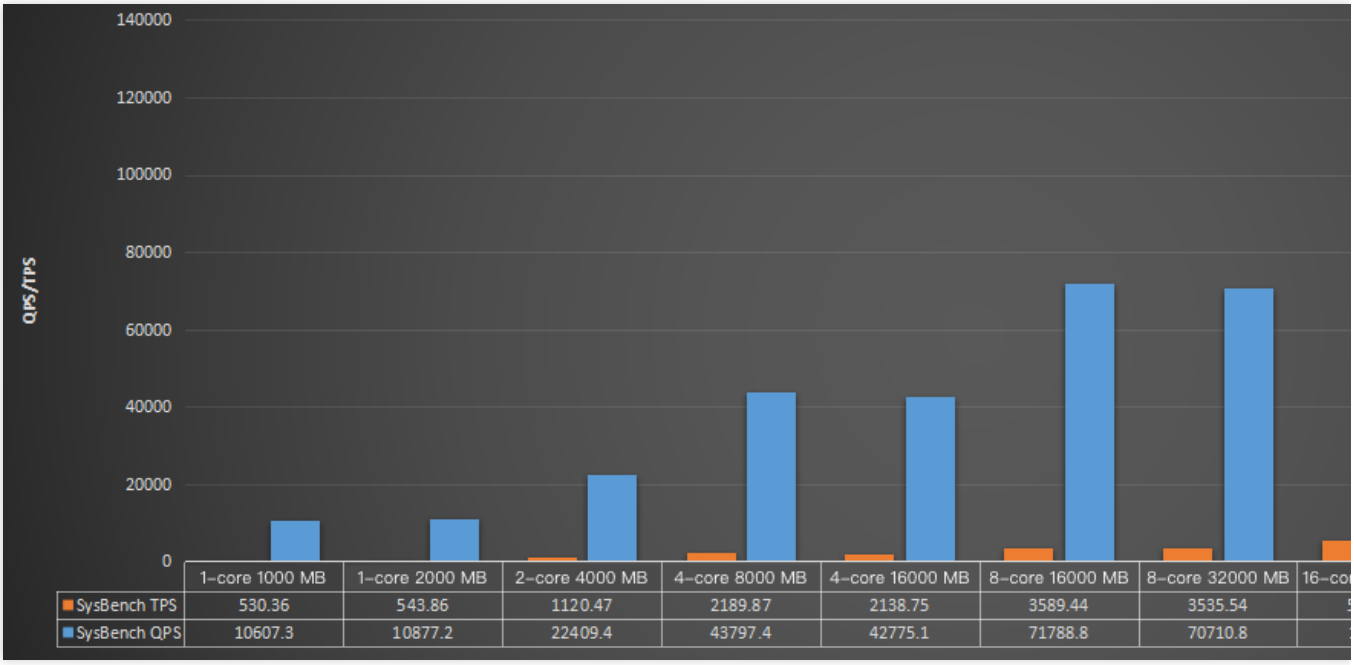


CPU (Core)	Memory (MB)	Concurrency	Data Volume/Table	Total Tables	SysBench TPS	SysBench QPS	avg_lat
1	1000	8	25000	150	577.97	11559.4	13.84
1	2000	8	25000	150	550.21	11004.3	14.54
2	4000	16	25000	150	1190.41	23808.1	13.44
4	8000	32	25000	150	2307.1	46142	13.87
4	16000	32	25000	150	2229.8	44596.1	14.35
8	16000	64	25000	150	3771.25	75425	16.97

8	32000	64	25000	150	3909.36	78187.1	16.37
16	32000	128	25000	150	6102.35	122047	20.97
16	64000	128	25000	150	6788.83	135777	18.85
16	96000	128	25000	150	6771.81	135436	18.9
16	128000	128	25000	150	7039.65	140793	18.18

Scenario 2: Disk I/O

In the disk I/O scenario, as only part of the data can be put into the cache, the disk needs to be read and written to update the cache during queries.



CPU (Core)	Memory (MB)	Concurrency	Data Volume/Table	Total Tables	SysBench TPS	SysBench QPS	avg_lat
1	1000	8	800000	6	530.36	10607.3	15.08
1	2000	8	800000	12	543.86	10877.2	14.71
2	4000	16	800000	24	1120.47	22409.4	14.28
4	8000	32	800000	48	2189.87	43797.4	14.61
4	16000	32	6000000	13	2138.75	42775.1	14.96

8	16000	64	6000000	13	3589.44	71788.8	17.83
8	32000	64	6000000	25	3535.54	70710.8	18.1
16	32000	128	6000000	25	5550.31	111006	23.06
16	64000	128	6000000	49	6414.39	128288	19.95
16	96000	128	6000000	74	5874.64	117493	21.78
16	128000	128	6000000	98	5611.06	112221	22.81

Network Architecture Performance Comparison

Last updated：2024-07-23 17:42:37

TencentDB for MySQL has upgraded the network architecture of database instances for a higher performance and lower latency. This document describes the performance tests on the old and new versions.

Note:

Starting from November 9, 2022, the new network architecture has been applied to newly purchased instances to deliver lower latency and higher performance.

On January 21, 2023, all existing database instances have been switched to the new network architecture.

Single-node instances of cloud disk edition are already in the optimal network architecture, so their details page does not indicate whether the network architecture is new.

The new network architecture cannot be used in the classic network. To use it, switch to VPC as instructed in Network Switch and wait for the network architecture to upgrade.

For more information, see Network Architecture Upgrade.

Test environment

Region/AZ: Beijing - Beijing Zone 6.

Client specification: S5.2XLARGE16 (8 CPU cores and 16 GB memory).

Client OS: TencentOS Server 3.2.

Network: Both the CVM and TencentDB for MySQL instances are in the same VPC subnet.

Storage type: Local SSD disk

Instance specification: General (4 CPU cores and 16 GB memory).

Parameter template: High-performance template

Replication method: Async replication

Test tool

SysBench is a modular, cross-platform, and multi-threaded benchmark tool for evaluating OS parameters that are important for a system running a database under intensive load. The idea of this benchmark suite is to quickly get an impression about system performance without setting up complex database benchmarks or even without installing a database at all. This stress test uses SysBench 1.0.20.

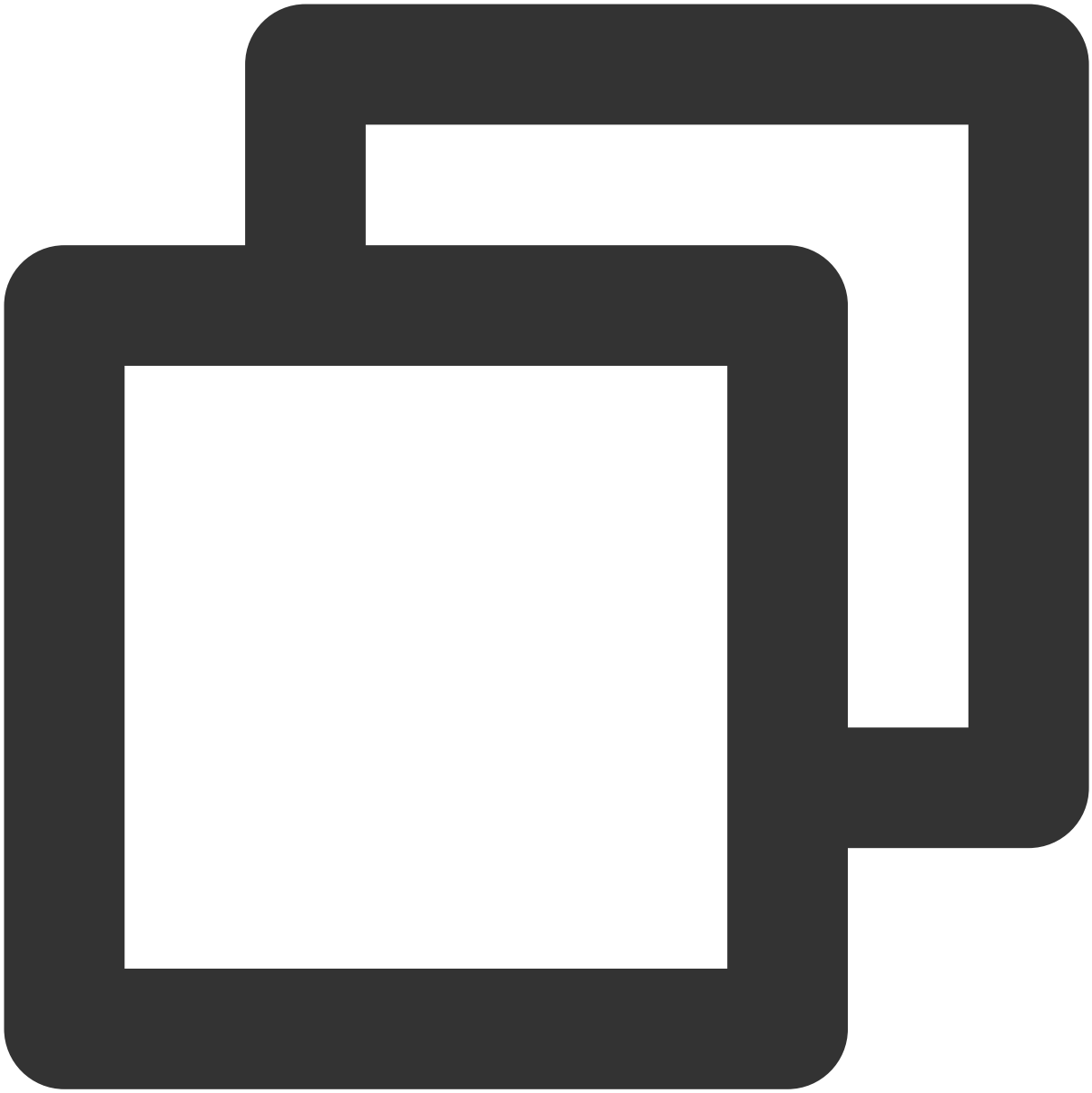
Test scenario

This stress test involves write-only, read-only, and read-write scenarios, each with 2–3,000 threads tested. The QPS is used as the performance metric.

Test method

Step 1. Prepare the data

Run the following command:



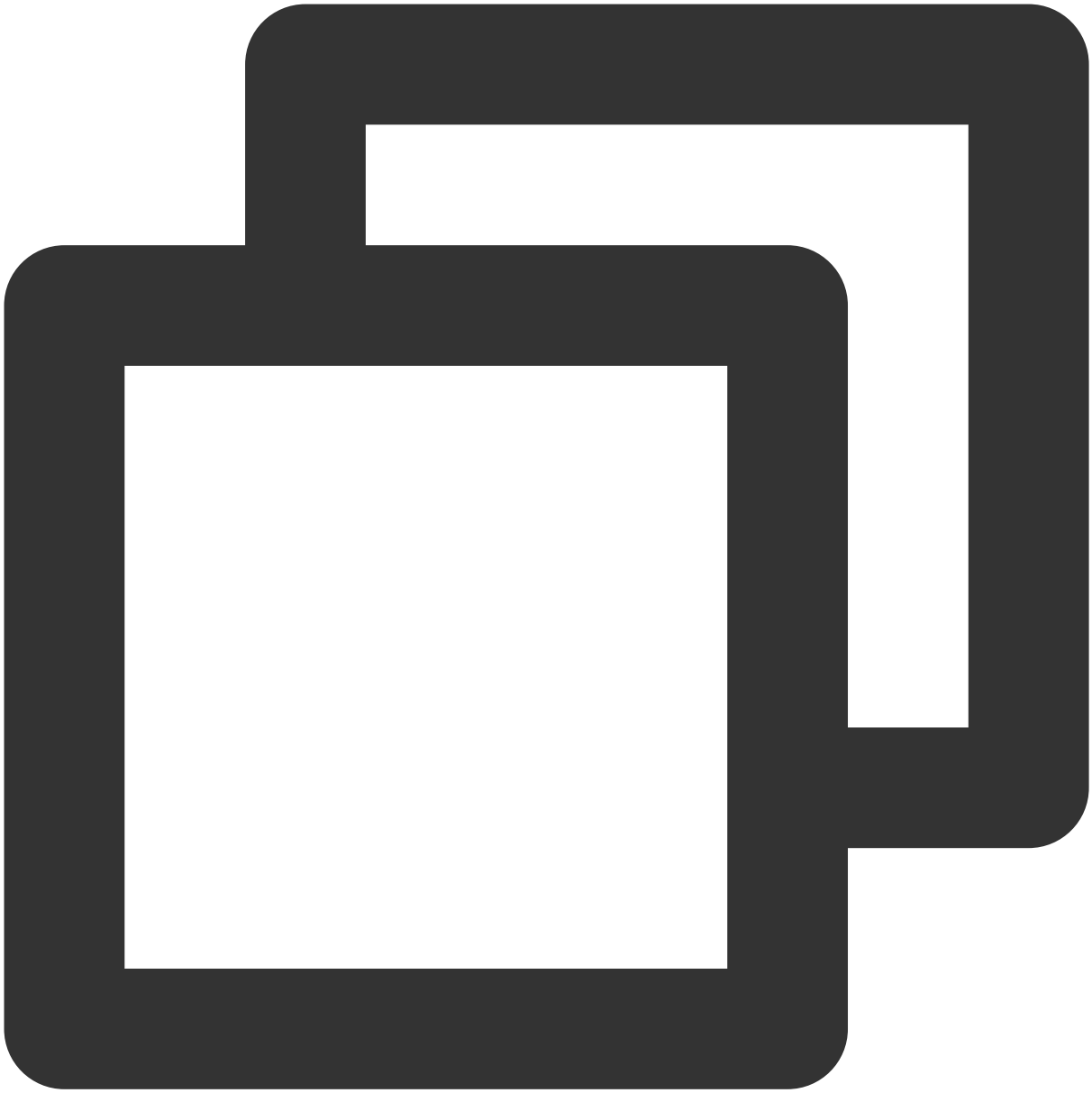
```
sysbench --db-driver=mysql --mysql-host=XXX --mysql-port=XXX
--mysql-user=XXX --mysql-password=XXX --mysql-db=sbtest --table_size=10000000
--tables=10 --events=0 --time=300 --threads={2~3000} oltp_read_write prepare
```

Step 2. Run the workload

Run the workload in write-only, read-only, and read-write scenarios. Make sure that the configuration is correct.

OLTP write-only scenario

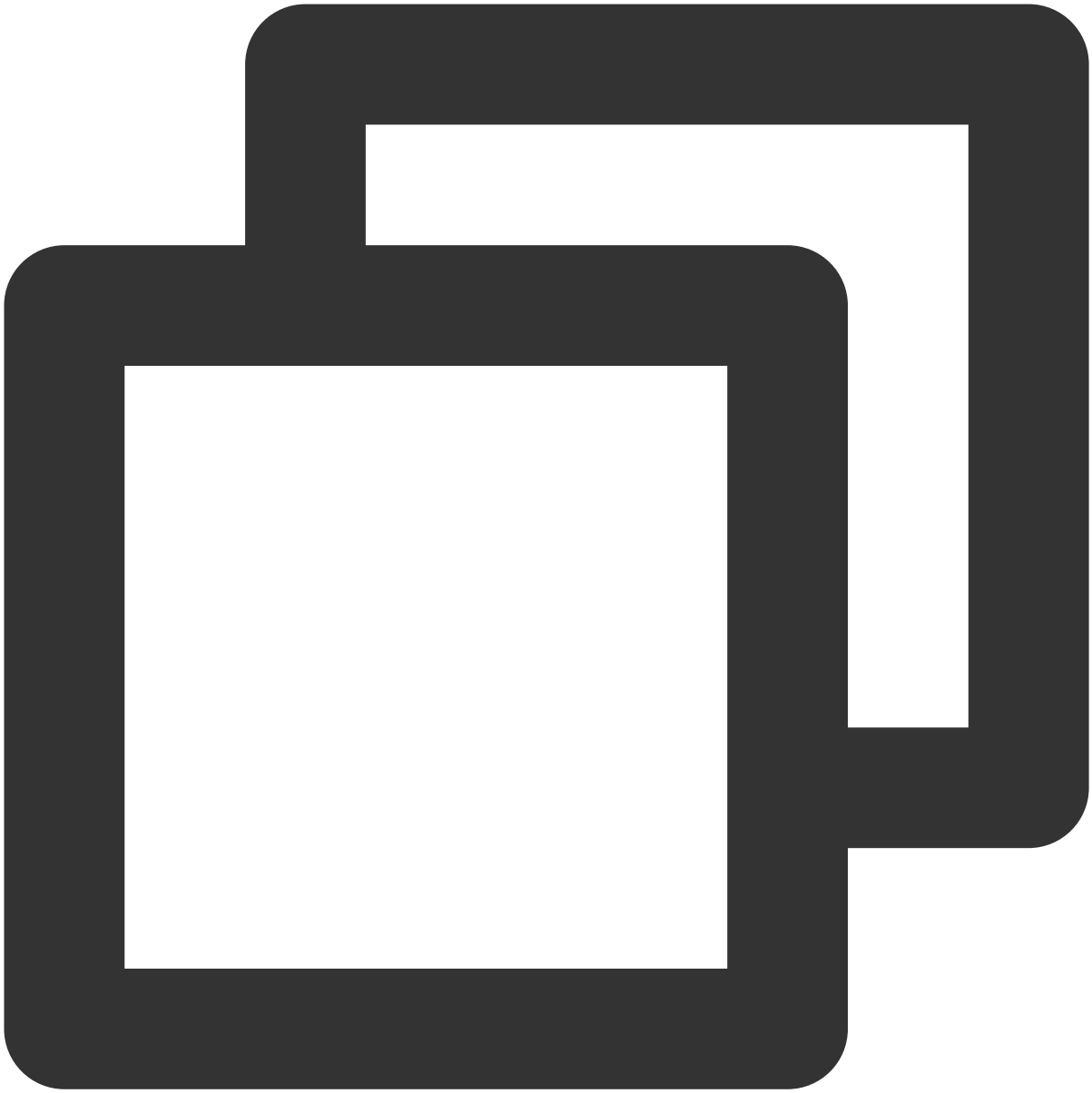
Run the following command:



```
sysbench --db-driver=mysql --mysql-host=XXX --mysql-port=XXX
--mysql-user=XXX --mysql-password=XXX --mysql-db=sbtest --table_size=10000000
--tables=10 --events=0 --time=300  --threads={2~3000} --percentile=95 --report-inte
run
```

OLTP read-only scenario

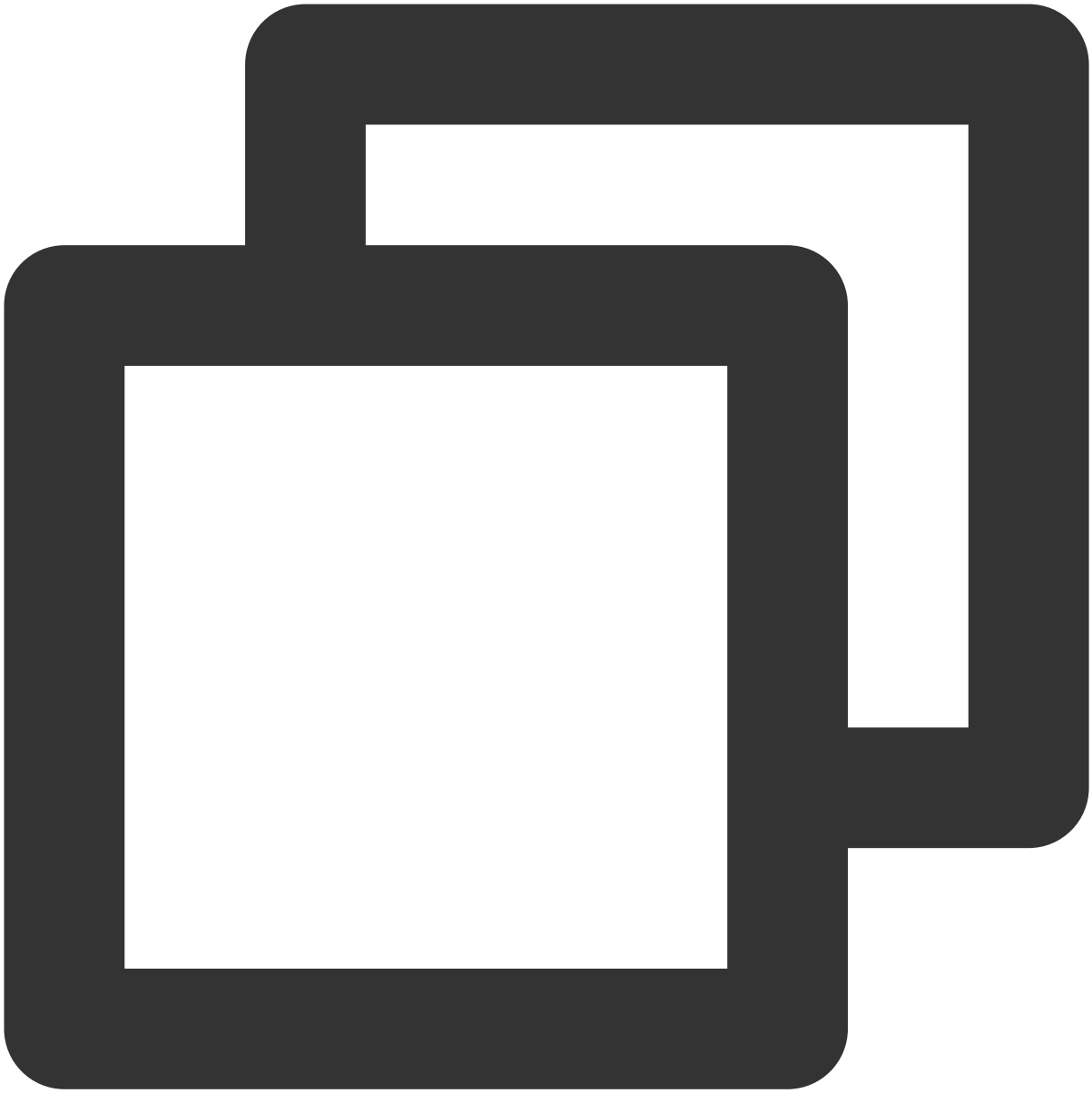
Run the following command:



```
sysbench --db-driver=mysql --mysql-host=XXX --mysql-port=XXX
--mysql-user=XXX --mysql-password=XXX --mysql-db=sbtest --table_size=10000000
--tables=10 --events=0 --time=300 --threads={2~3000} --percentile=95 --skip-trx=1 -
oltp_read_only
run
```

OLTP read-write scenario

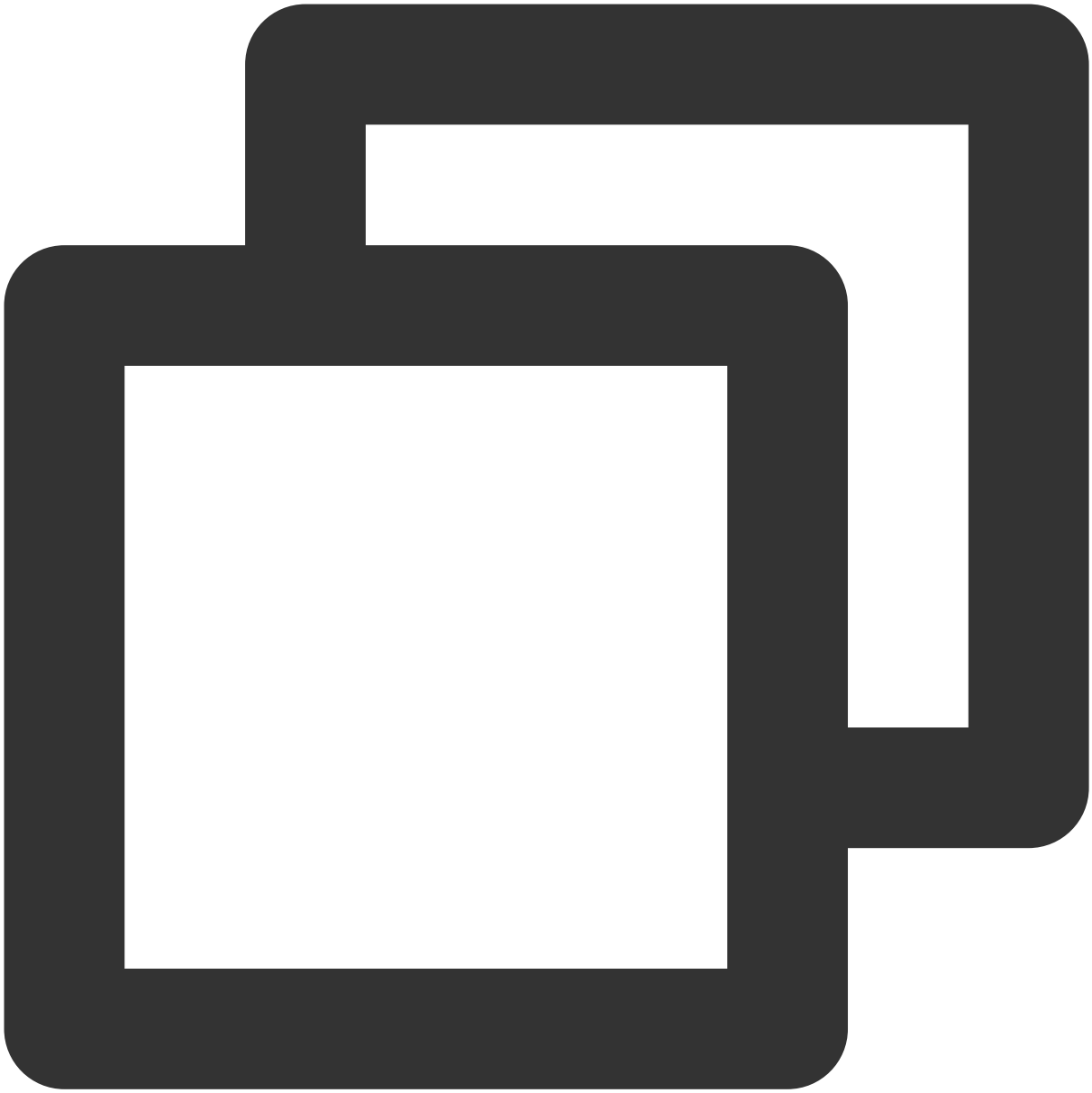
Run the following command:



```
sysbench --db-driver=mysql --mysql-host=XXX --mysql-port=XXX
--mysql-user=XXX --mysql-password=XXX --mysql-db=sbtest --table_size=10000000
--tables=10 --events=0 --time=300  --threads={2~3000} --percentile=95 --report-inte
run
```

Step 3. Clear the data

Run the following command to clear the data after the test is executed:



```
sysbench --db-driver=mysql --mysql-host=XXX --mysql-port=XXX --mysql-user=XXX --mys
--mysql-db=sbtest --table_size=25000 --tables=250 --events=0 --time=600   --threads
```

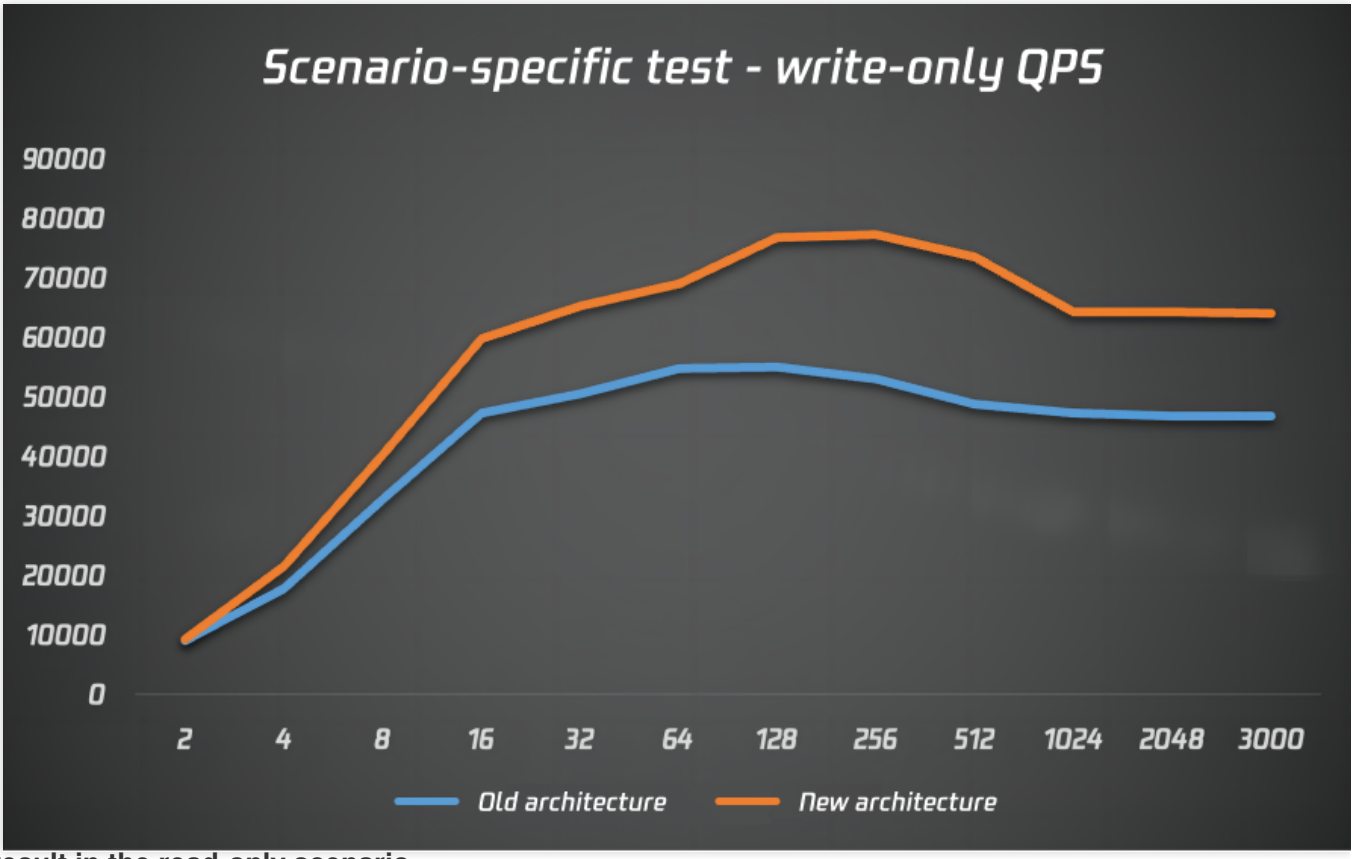
Test metric

The test metric is queries per second (QPS).

Test result

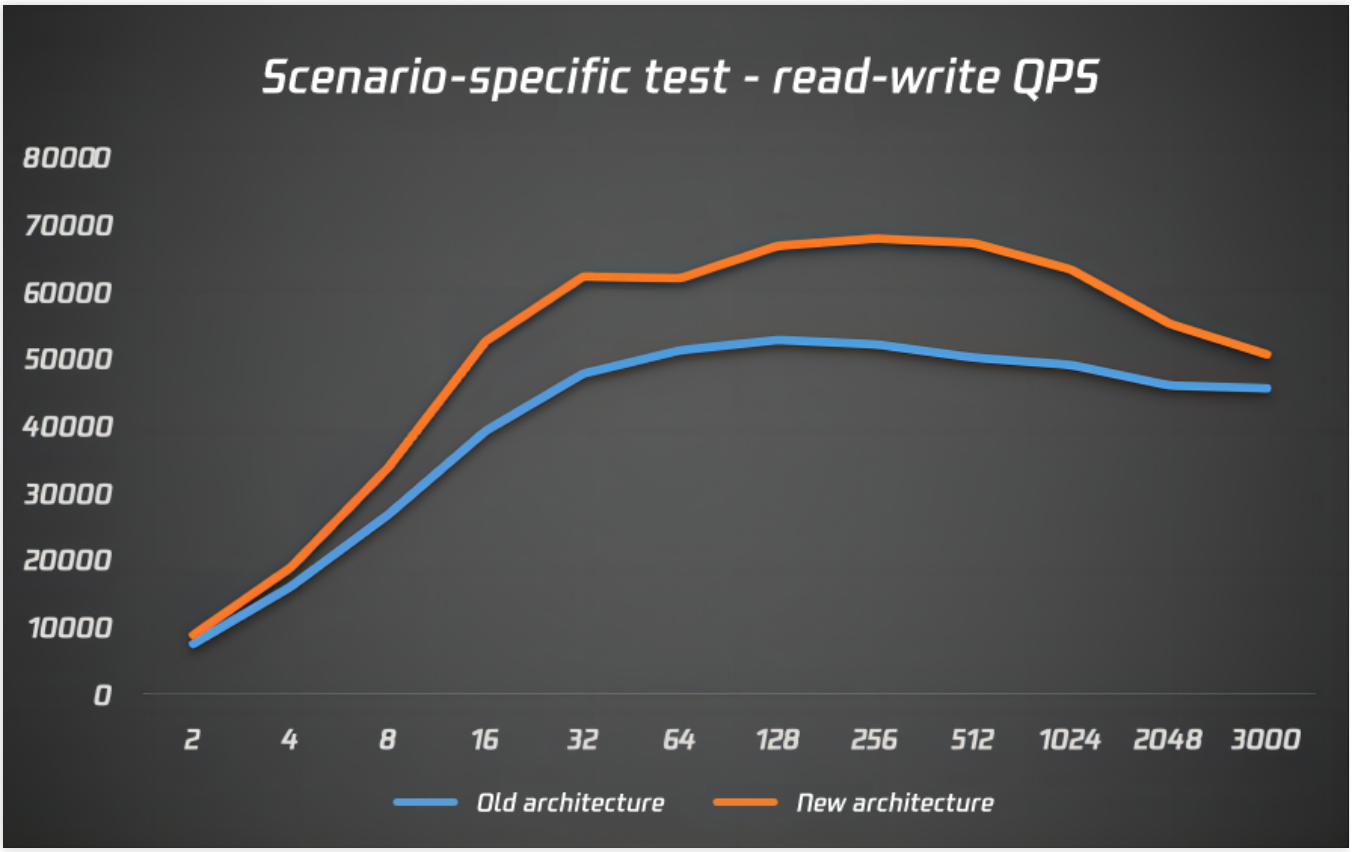
Test result in the write-only scenario

In the write-only scenario, the new architecture of TencentDB for MySQL always outperforms the old architecture as the thread quantity increases, with the QPS peaking at 256 threads and being 20% higher than the old architecture's at 512 threads.



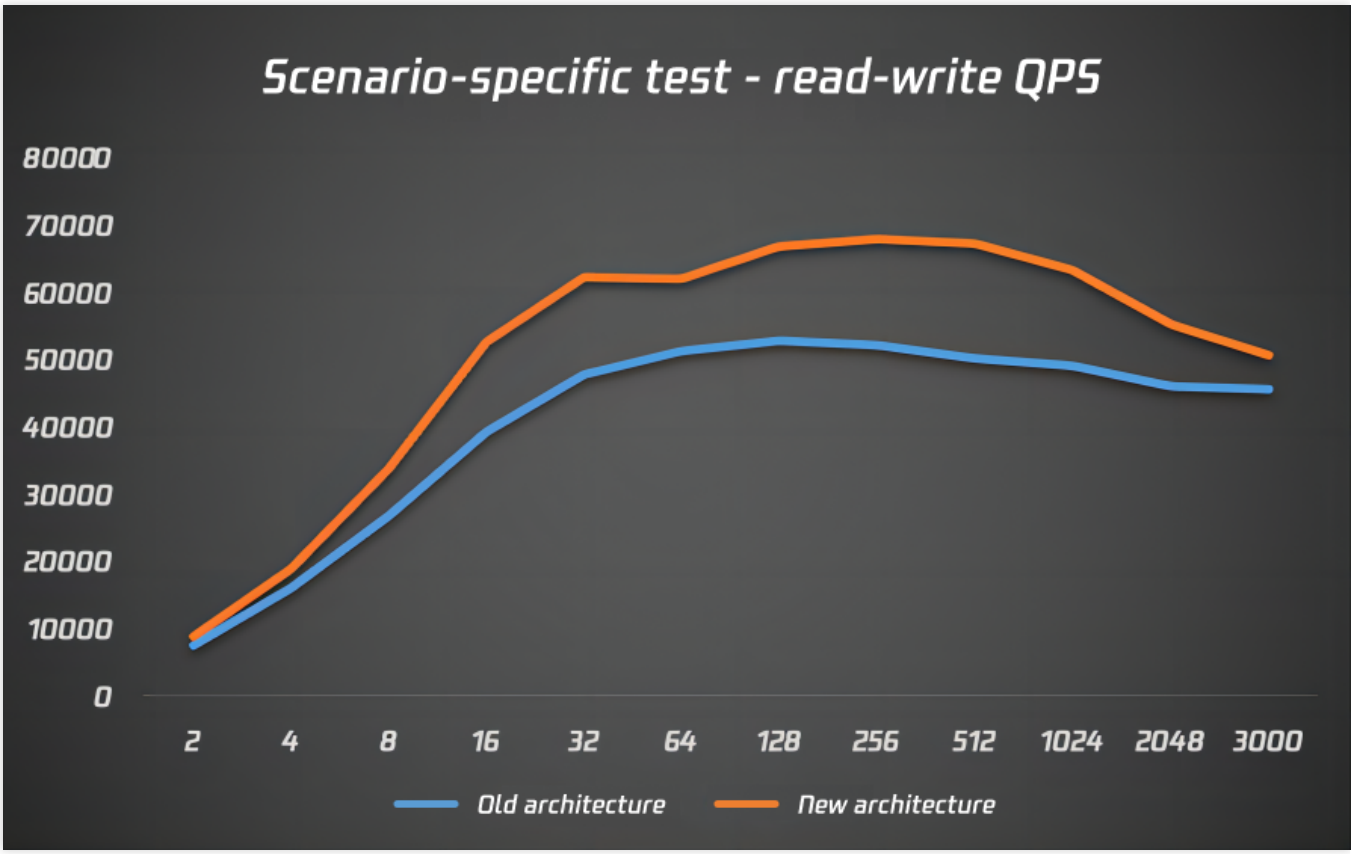
Test result in the read-only scenario

In the read-only scenario, the new architecture of TencentDB for MySQL sees a great linear QPS rise as the thread quantity increases from a small number, with the QPS 22% being higher than the old architecture's at 16 threads and going up steadily at 64 threads or more.



Test result in the read-write scenario

In the read-write scenario, the new architecture of TencentDB for MySQL sees a great QPS rise as the thread quantity increases from a small number, with the QPS hitting the peak (18% higher than the old architecture's) at 512 threads and falling steadily after that.



Conclusion

Note:

The above performance test results are for reference only.

The above scenario tests show that the new network architecture of TencentDB for MySQL greatly outperforms the old architecture. In the range of 2–3,000 threads, the QPS is improved by 20% on average.

Security White Paper

Overview

Last updated：2024-07-23 17:42:52

TencentDB for MySQL makes it simple to deploy and use MySQL database in the cloud. It allows you to deploy a scalable MySQL database instance in minutes at lower costs and adjust your hardware capacity without any shutdown. TencentDB for MySQL streamlines database operations with features like backup, rollback, monitoring, fast scaling, and data transfer, allowing you to focus more on business development.
TencentDB for MySQL provides diversified security reinforcement features to ensure the reliability and security of your data. In order to make your instances more secure, we recommend you use the following security features based on your business needs:

Security Feature	Security Capability
Data storage security	Automatic backup Archive backup retention Transparent data encryption Data security
Security audit	Compliance audit Security governance
Access control	Database account management Access management Password complexity
Data communication security	VPC Security group SSL encryption
Data disaster recovery	Intra-region disaster recovery Cross-region disaster recovery

Data Storage Security

Last updated：2024-07-23 17:43:06

TencentDB for MySQL ensures the confidentiality and integrity of data during storage.

Automatic backup

TencentDB for MySQL supports both automatic and manual backup to ensure data restorability that guarantees data integrity and reliability. It provides data backup and log backup features by default, where the frequency of automatic backup should be set to more than twice a week. If you have other backup needs, you can initiate manual backup through the console or APIs at any time.
For more information on how to use this feature, see Backing up Databases.

Archive backup retention

TencentDB for MySQL allows you to enable archive backup retention. You can flexibly configure the retention period of backup files as needed, which is 7 days by default and can be up to 1,830 days. Backup files that exceed the retention period will be automatically deleted. For data security considerations, we recommend you back up your data at least twice a week.
For more information on how to use this feature, see Backing up Databases.

Transparent data encryption

TencentDB for MySQL supports the transparent data encryption (TDE) feature developed by the Tencent Cloud database team. Transparent encryption means that the data encryption and decryption are imperceptible to you. When creating an encrypted table, you do not need to specify an encryption key, and the data will be encrypted during write to the disk and decrypted during read from the disk.
TDE uses the internationally popular AES algorithm and 256-bit encryption keys, which are managed in Tencent Cloud Key Management Service (KMS). You need to be authorized to access KMS and can rotate keys in the KMS console to further improve the system security.
For more information on how to use this feature, see Enabling Transparent Data Encryption.

Access Control

Last updated：2024-07-23 17:43:18

TencentDB for MySQL provides access control capabilities. By defining and verifying user permissions, regulating user access to database resources, and managing database resource permissions, you can ensure that only authorized users can access database objects within the scope of their permissions or at their security levels.

Database account management

You can create database accounts through the TencentDB for MySQL console or API. You can also grant management permissions at different levels to such accounts. We recommend you authorize accounts based on the principle of least privilege to ensure the data security.
For more information, see Creating Account.

Access management

Cloud Access Management (CAM) helps you securely manage and control access permissions to your Tencent Cloud resources. With CAM, you can create, manage, and terminate users (groups), and control the Tencent Cloud resources that can be used by the specified user through identity and policy management, which implements permission separation.
For more information, see CAM Overview.

Password complexity

Passwords are the most important means for protecting database security. As more data security regulations are introduced, there are higher requirements for the database password strength. TencentDB for MySQL supports the custom password complexity feature to protect your database security and meet your needs for compliance with applicable regulations.
You can configure the custom password complexity feature in the console to set the password strength for all console and database operations involving a password. This helps protect your passwords and prevent security risks such as password leakage. The feature offers the following configuration items:
Min Number of Uppercase or Lowercase Letters
Min Number of Digitals
Min Number of Symbols
Min Number of Password Characters

For more information, see Setting Password Complexity.

Data Communication Security

Last updated：2024-07-23 17:43:32

TencentDB for MySQL provides data communication security capabilities to ensure the confidentiality and integrity of data during communication.

VPC

TencentDB for MySQL supports using Virtual Private Cloud (VPC) to achieve a higher degree of network isolation and control. A VPC is a logically isolated network space in Tencent Cloud. In a VPC, you can customize IP ranges, IP addresses, and routing policies to implement network isolation at the resource level.
TencentDB for MySQL instances deployed in a VPC can only be accessed by CVM instances in the same VPC by default. If the CVM and MySQL instances are in different VPCs, they can communicate after you apply for public network access. For the sake of network security, we recommend you not access your databases over the public network. If you have to do so, configure appropriate security groups to implement access control for clients.
For more information on how to use this feature, see Creating VPCs for TencentDB for MySQL.

Security group

TencentDB for MySQL supports security group as an important means for network security isolation, which can be used to set network access controls for one or more TencentDB instances. Instances with the same network security isolation demands in one region can be put into the same security group, which is a logical group.
For more information on how to use this feature, see TencentDB Security Group.

SSL encryption

TencentDB for MySQL supports Secure Sockets Layer (SSL) authentication, which is a process that authenticates the connection from the user client to the TencentDB server. After SSL encryption is enabled, you can get a CA certificate and upload it to the server. Then, when the client accesses the database, the SSL protocol will be activated to establish an SSL secure channel between the client and the server. This implements encrypted data transfer, prevents data from being intercepted, tampered with, and eavesdropped during transfer, and ultimately ensures the data security.
Note that SSL encryption does not protect the data itself; instead, it secures the traffic between the database and the server. Encrypting the network connection at the transport layer can improve the security and integrity of the communication data, but will increase the response time of the network connection.

For more information on how to use this feature, see Setting SSL Encryption.

Data Disaster Recovery

Last updated：2024-07-23 17:43:44

TencentDB for MySQL provides cross-AZ and cross-region data disaster recovery solutions to help you deliver continued services at low costs while improving data reliability. It is also applicable to scenarios where compliance is required.

Intra-region disaster recovery

You can create two-node and three-node TencentDB for MySQL instances to support multi-AZ deployment. Physical servers of a multi-AZ instance are deployed in different AZs in the same region. When an AZ fails, the business traffic will be switched to another AZ swiftly, which is transparent to the business and requires no changes at the application layer to achieve intra-region disaster recovery.

Note:

As the nodes of a multi-AZ instance are in different AZs, there may be an additional network sync delay of 2–3 ms.

For more information on how to use this feature, see High Availability (Multi-AZ).

Cross-region disaster recovery

The intra-region disaster recovery capability of TencentDB for MySQL is limited to different AZs in the same region. To further improve the availability, TencentDB for MySQL also supports cross-region data disaster recovery.

You can asynchronously replicate data in a TencentDB for MySQL instance in region A to another instance (disaster recovery instance) in region B through DTS. The disaster recovery instance has an independent connection address, account, and permissions. If a major failure occurs in region A and cannot be fixed in a short time, you can perform failover whenever needed. Specifically, you can quickly forward application requests to the disaster recovery instance simply by modifying the database connection configuration in the application, thereby delivering a finance-grade database availability.

For more information on how to use this feature, see Disaster Recovery Instance.