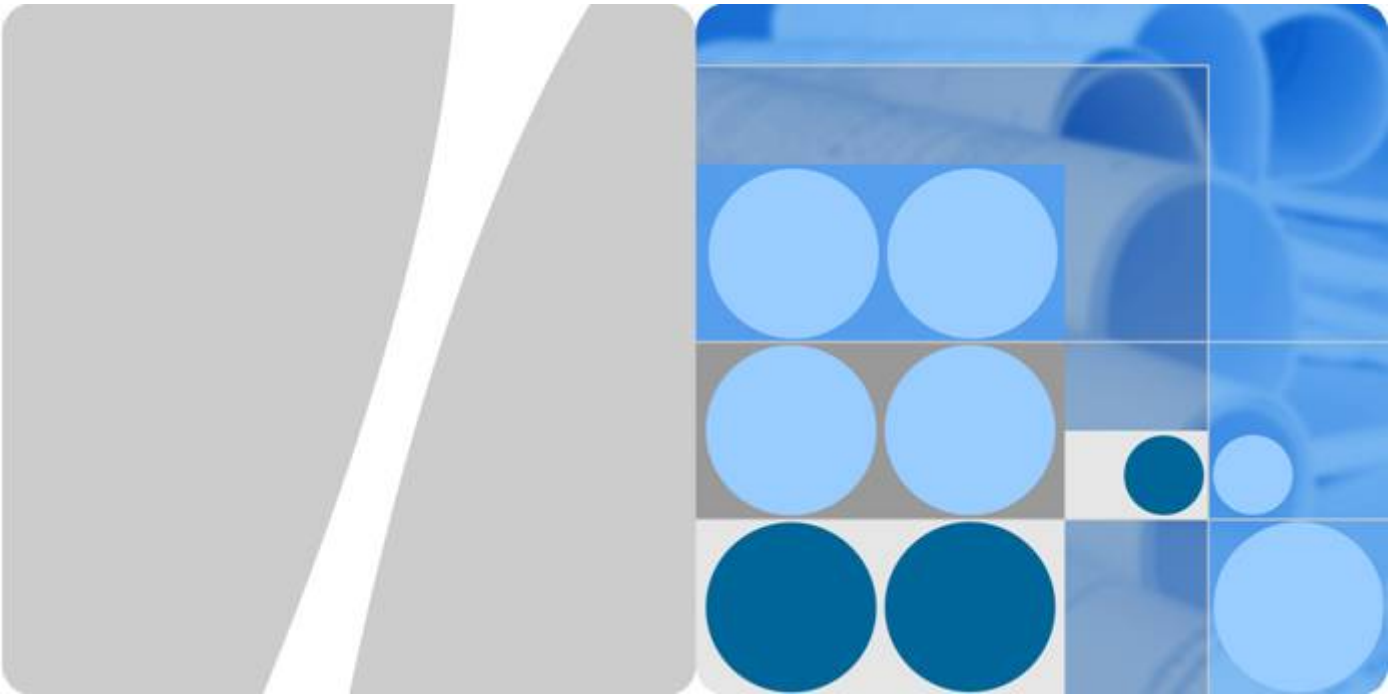


Relational Database Service

Performance White Paper

Issue 01

Date 2020-12-29

HUAWEI TECHNOLOGIES CO., LTD.

HUAWEI

Copyright © Huawei Technologies Co., Ltd. 2020. All rights reserved.

No part of this document may be reproduced or transmitted in any form or by any means without prior written consent of Huawei Technologies Co., Ltd.

Trademarks and Permissions


and other Huawei trademarks are trademarks 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 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.

Huawei Technologies Co., Ltd.

Address: Huawei Industrial Base
Bantian, Longgang
Shenzhen 518129
People's Republic of China

Website: https://www.huawei.com

Email: support@huawei.com

Contents

1 RDS for MySQL....1
1.1 Test Method....1
1.2 MySQL 5.6 Test Data....4
1.3 MySQL 5.7 Test Data....7
1.4 MySQL 8.0 Test Data....10
2 RDS for PostgreSQL....12
2.1 Test Method....12
2.2 PostgreSQL 9.5 Test Data....19
2.3 PostgreSQL 9.6 Test Data....20
2.4 PostgreSQL 10 Test Data....21
2.5 PostgreSQL 11 Test Data....23
2.6 PostgreSQL 11 Enhanced Edition Test Data....24
2.7 PostgreSQL 12 Test Data....25
3 PostgreSQL....28
3.1 Test Method....28
3.2 PostgreSQL 9.6 Test Data....31
4 RDS for SQL Server....34
4.1 Test Method....34
4.2 Test Data....43

 Copyright © Huawei Technologies Co., Ltd.

1 RDS for MySQL

1.1 Test Method

MySQL is one of the world's most popular open-source relational databases. It works with the Linux, Apache, and PHP (LAMP) stack to provide efficient web solutions. It solves problems such as poor database performance, long data replication delay, and long fault recovery time in high concurrency scenarios.

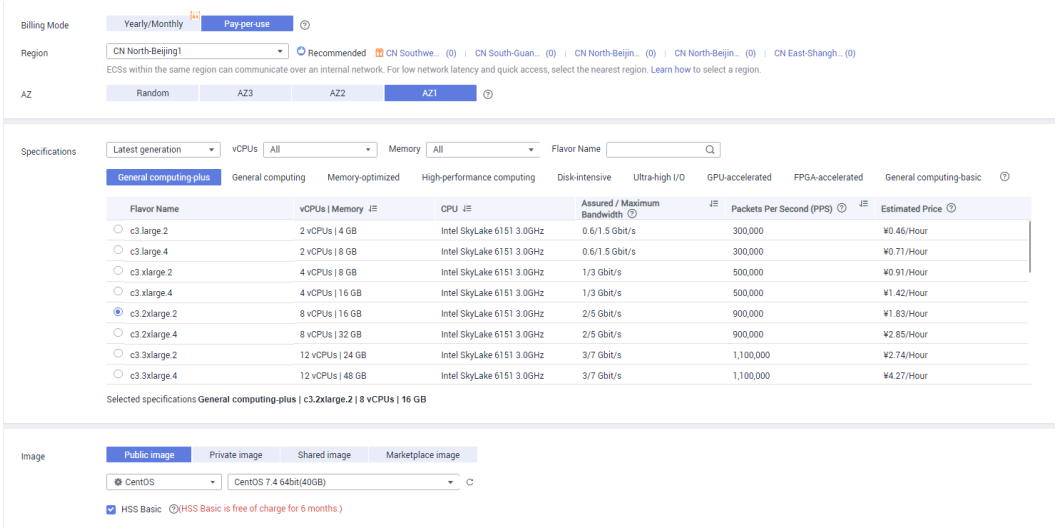
RDS for MySQL is immediate use, and provides backup and restoration, data migration, security protection, high availability, and elastic scalability. You can obtain a production database with high performance and scalability in a few minutes after simple configurations while your data integrity and service continuity are guaranteed.

Test Environment

- Region: CN North-Beijing1
- AZ: AZ1
- Elastic Cloud Server (ECS): general computing | c3.2xlarge.2 | 8 vCPUs | 16 GB, CentOS7.4 64 bit image Bind an elastic IP address (EIP) to the ECS because additional compilation tools need to be installed on stress testing tools.

 Copyright © Huawei Technologies Co., Ltd.

Figure 1-1 ECS configuration



NOTE

MySQL 8.0 test environment is as follows:

- Region: CN North-Beijing4
- AZ: AZ2
- ECS: general computing-plus | c6.4xlarge.2 | 16 vCPUs | 32 GB, CentOS 7.6 (64 bit) Bind an EIP to the ECS because additional compilation tools need to be installed on stress testing tools.

Test Tool

Sysbench is a multi-threaded benchmark tool based on LuaJIT, allowing you to quickly get an impression of system performance by using a built-in database test model. For details, visit **https://github.com/akopytov/sysbench**.

Sysbench 1.0.12 is used as an example. Run the following commands to install sysbench:

 # wget -c https://github.com/akopytov/sysbench/archive/1.0.12.zip

 # yum install autoconf libtool mysql mysql-devel vim unzip

 # unzip 1.0.12.zip

 # cd sysbench-1.0.12

 # ./autogen.sh

 # ./configure

 # make

 # make install

Test Procedure

Replace the database name, connection IP address, and user password based on the site requirements.

 Copyright © Huawei Technologies Co., Ltd.

Step 1 Import data.

1. Run the following command to log in to a database and create the test database **loadtest**:

   mysql -u root -P 3306 -h *<host>* **-p -e "create database loadtest"**

2. Run the following command to import the test background data to the **loadtest** database:

   sysbench --test=/usr/local/share/sysbench/tests/include/oltp_legacy/oltp.lua --db-driver=mysql --mysql-db=loadtest --mysql-user=root --mysql-password=*<password>* **--mysql-port=3306 --mysql-host=***<host>* **--oltp-tables-count=64 --oltp-table-size=10000000 --num-threads=20 prepare**

Step 2 Run the following command to perform a stress testing:

sysbench --test=/usr/local/share/sysbench/tests/include/oltp_legacy/oltp.lua --db-driver=mysql --mysql-db=loadtest --mysql-user=root --mysql-password=*<password>* **--mysql-port=3306 --mysql-host=***<host>* **--oltp-tables-count=64 --oltp-table-size=10000000 --max-time=3600 --max-requests=0 --num-threads=200 --report-interval=3 --forced-shutdown=1 run**

Step 3 Run the following command to delete the test data:

sysbench --test=/usr/local/share/sysbench/tests/include/oltp_legacy/oltp.lua --db-driver=mysql --mysql-db=loadtest --mysql-user=root --mysql-password=*<password>* **--mysql-port=3306 --mysql-host=***<host>* **--oltp-tables-count=64 --oltp-table-size=10000000 --max-time=3600 --max-requests=0 --num-threads=200 cleanup**

----End

Testing Model

1. Table structure:

   CREATE TABLE `sbtest` (
   `id` INTEGER UNSIGNED NOT NULL AUTO_INCREMENT,
   `k` INTEGER UNSIGNED DEFAULT '0' NOT NULL,
   `c` CHAR(120) DEFAULT '' NOT NULL,
   `pad` CHAR(60) DEFAULT '' NOT NULL,
   PRIMARY KEY (`id`)
) ENGINE=InnoDB

2. Read/write ratio:

   The default transaction submitted by sysbench contains 18 SQL statements. The details are as follows:

   - Ten primary key select statements:

     SELECT c FROM ${rand_table_name} where id=${rand_id};

   - Four range select statements:

     SELECT c FROM ${rand_table_name} WHERE id BETWEEN ${rand_id_start} AND ${rand_id_end};

     SELECT SUM(K) FROM ${rand_table_name} WHERE id BETWEEN ${rand_id_start} AND ${rand_id_end};

 Copyright © Huawei Technologies Co., Ltd.

SELECT c FROM ${rand_table_name} WHERE id BETWEEN ${rand_id_start} AND ${rand_id_end} ORDER BY c;

SELECT DISTINCT c FROM ${rand_table_name} WHERE id BETWEEN ${rand_id_start} AND ${rand_id_end} ORDER BY c;

- Two update statements:

  UPDATE ${rand_table_name} SET k=k+1 WHERE id=${rand_id}

  UPDATE ${rand_table_name} SET c=${rand_str} WHERE id=${rand_id}

- One delete statement:

  DELETE FROM ${rand_table_name} WHERE id=${rand_id}

- One insert statement:

  INSERT INTO ${rand_table_name} (id, k, c, pad) VALUES (${rand_id},${rand_k},${rand_str_c},${rand_str_pad})

Test Metric

- Transaction per second (TPS) refers to the number of transactions executed per second by a database. Each transaction contains 18 SQL statements.
- Query per second (QPS) refers to the number of SQL statements, including INSERT, SELECT, UPDATE, and DELETE statements, executed per second.

1.2 MySQL 5.6 Test Data

About IOPS

Input/output operations per second (IOPS) is closely related to storage space (in GB) of the DB instance.

IOPS = min{3,500 + (Instance storage space – 40) x 50, 33,000}

For example, if the storage space of DB instance **rds-test01** is 400 GB, IOPS is calculated as follows: IOPS = min{3,500 + (400 - 40) x 50, 33,000} = min{21,500, 33,000} = 21,500

Test Data of General-Enhanced DB Instances

NOTICE

The **Connections (in Stress Testing)** columns indicate the results of the RDS performance stress testing. For services running on the live network, set the **max_connections** parameter.

Copyright © Huawei Technologies Co., Ltd.

Table 1-1 vCPU:Memory=1:2

vCPUs	Memory (GB)	Connections (in Stress Testing)	TPS	QPS	IOPS
1	2	800	197.19	3,943.85	See **About IOPS**.
2	4	1,500	470.4	9,400.32	
4	8	2,500	768.23	15,364.64	
8	16	5,000	1,728.84	34,576.84	
16	32	10,000	2,947.42	58,948.35	
32	64	18,000	3,239.9	64,798.05	
60	128	30,000	4,735.63	94,712.67	

Table 1-2 vCPU:Memory=1:4

vCPUs	Memory (GB)	Connections (in Stress Testing)	TPS	QPS	IOPS
1	4	1,500	466.73	9,334.62	See **About IOPS**.
2	8	2,500	519.24	10,384.74	
4	16	5,000	1,191.98	23,839.68	
8	32	10,000	2,698.01	53,960.27	
16	64	18,000	3,148.77	62,975.44	
32	128	30,000	4,385.29	87,705.81	
60	256	60,000	4,810.94	95,117.75	

Table 1-3 vCPU:Memory=1:8

vCPUs	Memory (GB)	Connections (in Stress Testing)	TPS	QPS	IOPS
1	8	2,500	487.4	9,748.07	See **About IOPS**.
2	16	5,000	617.59	12,351.83	
4	32	10,000	1,374.01	27,480.14	
8	64	18,000	2,824.63	56,492.64	
16	128	30,000	4,215.09	84,301.79	

Copyright © Huawei Technologies Co., Ltd.

vCPUs	Memory (GB)	Connections (in Stress Testing)	TPS	QPS	IOPS
60	512	100,000	4,838.57	96,771.34	

Test Data of General-Enhanced DB Instances

Figure 1-2 vCPU:Memory=1:2

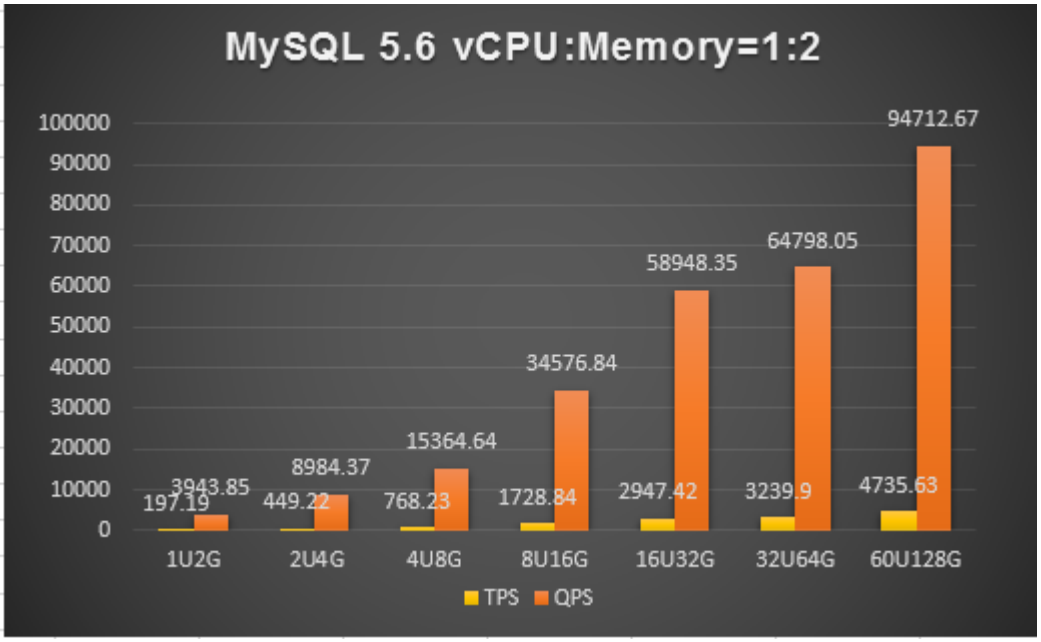
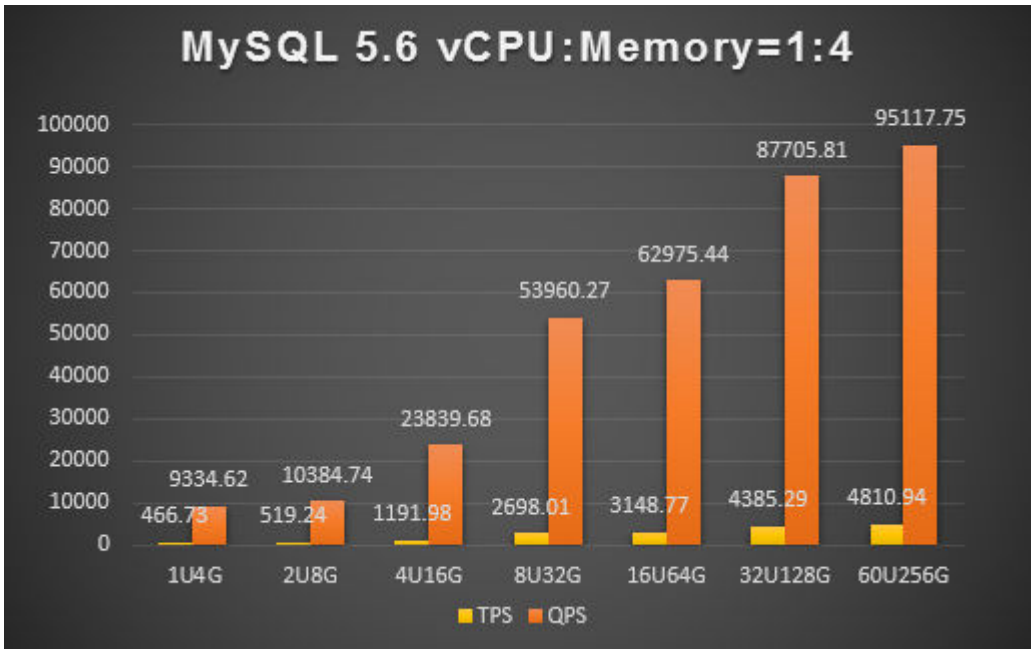
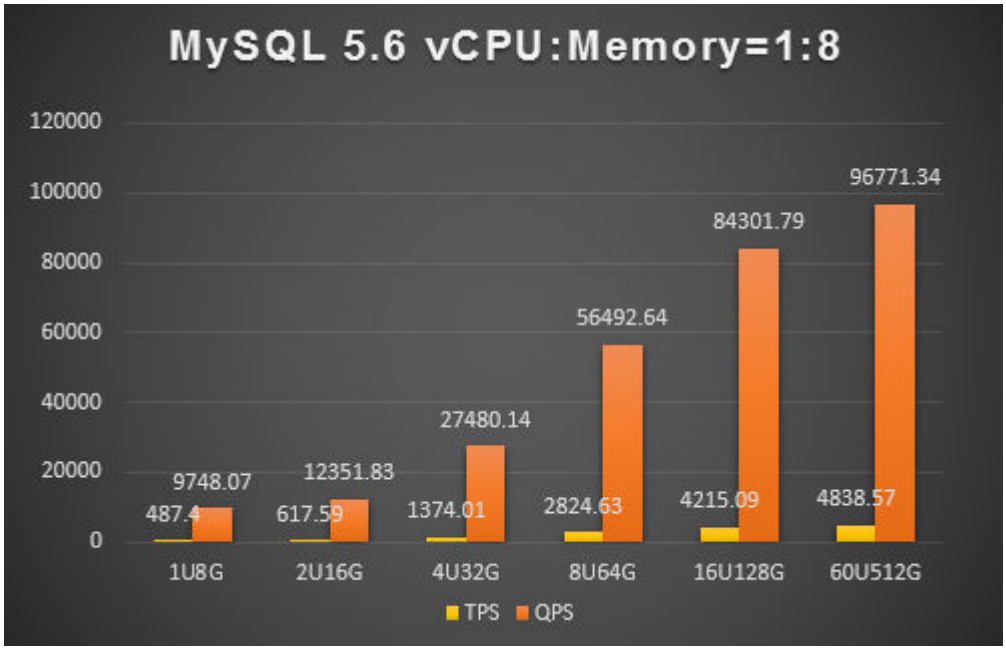


Figure 1-3 vCPU:Memory=1:4



 Copyright © Huawei Technologies Co., Ltd.

Figure 1-4 vCPU:Memory=1:8



1.3 MySQL 5.7 Test Data

About IOPS

Input/output operations per second (IOPS) is closely related to storage space (in GB) of the DB instance.

IOPS = min{3,500 + (Instance storage space – 40) x 50, 33,000}

For example, if the storage space of DB instance **rds-test01** is 400 GB, IOPS is calculated as follows: IOPS = min{3,500 + (400 - 40) x 50, 33,000} = min{21,500, 33,000} = 21,500

Test List of General-Enhanced Instances

NOTICE

The **Connections (in Stress Testing)** columns indicate the results of the RDS performance stress testing. For services running on the live network, set the **max_connections** parameter.

 Copyright © Huawei Technologies Co., Ltd.

Table 1-4 vCPU:Memory=1:2

vCPUs	Memory (GB)	Connections (Workload Test Value)	TPS	QPS	IOPS
1	2	800	236.59	4731.85	See **About IOPS**.
2	4	1500	470.35	9400.04	
4	8	2500	969.14	19382.91	
8	16	5000	1768.01	35360.39	
16	32	10000	2630.5	52000.99	
32	64	18000	3011.89	60237.87	
60	128	30000	4471.39	88428.09	

Table 1-5 vCPU:Memory=1:4

vCPUs	Memory (GB)	Connections (Workload Test Value)	TPS	QPS	IOPS
1	4	1500	468.11	9362.23	See **About IOPS**.
2	8	2500	620.7	12393.97	
4	16	5000	1230.39	24607.81	
8	32	10000	2514.48	50289.62	
16	64	18000	3016.84	60336.77	
32	128	30000	4367.67	87353.69	
60	256	60000	4536.41	90728.55	

Table 1-6 vCPU:Memory=1:8

vCPUs	Memory (GB)	Connections (Workload Test Value)	TPS	QPS	IOPS
1	8	2500	612.75	12255.01	See **About IOPS**.
2	16	5000	675.66	13513.18	
4	32	10000	1488.25	29765.06	
8	64	18000	2810.79	56215.81	
16	128	30000	4095.49	81909.79	

Copyright © Huawei Technologies Co., Ltd.

vCPUs	Memory (GB)	Connections (Workload Test Value)	TPS	QPS	IOPS
60	512	100000	4626.18	96823.95	

Test Result of the General-Enhanced Instances

Figure 1-5 vCPU:Memory=1:2

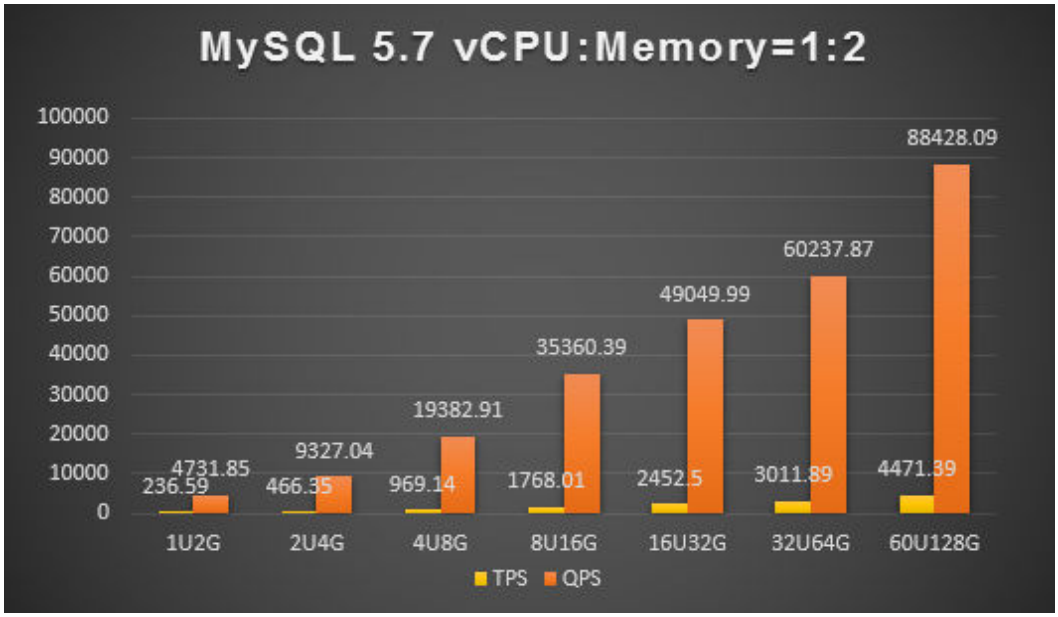
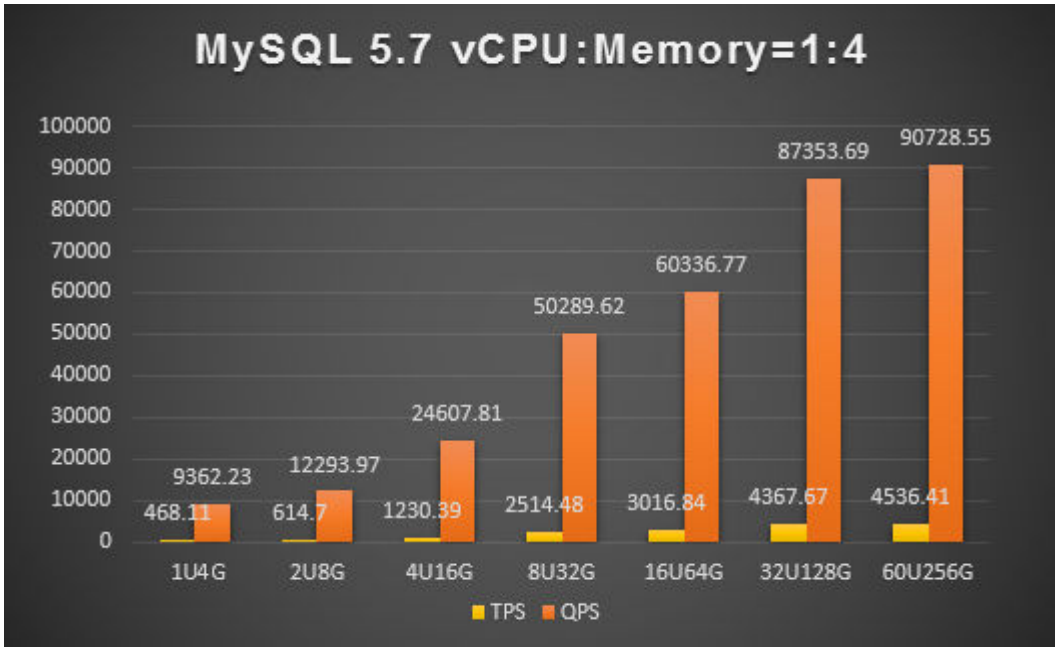
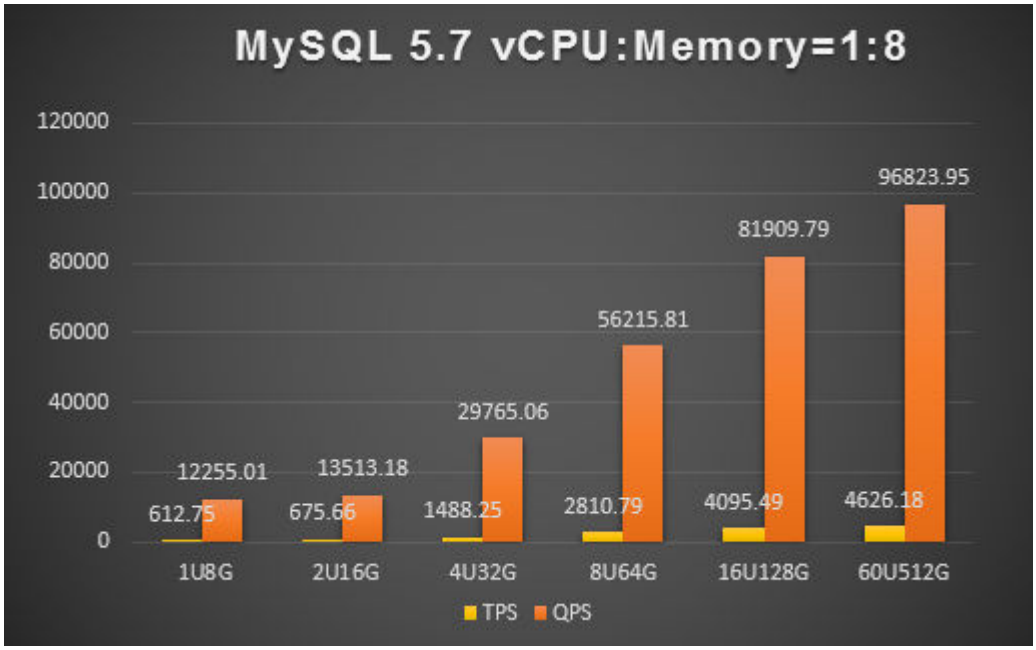


Figure 1-6 vCPU:Memory=1:4



 Copyright © Huawei Technologies Co., Ltd.

Figure 1-7 vCPU:Memory=1:8



1.4 MySQL 8.0 Test Data

NOTICE

--table_size=25000 --tables=250

--table_size=800000 --tables=150

About IOPS

Input/output operations per second (IOPS) is closely related to storage space (in GB) of the DB instance.

IOPS = min{3,500 + (Instance storage space – 40) x 50, 33,000}

For example, if the storage space of DB instance **rds-test01** is 400 GB, IOPS is calculated as follows: IOPS = min{3,500 + (400 - 40) x 50, 33,000} = min{21,500, 33,000} = 21,500

NOTICE

The **Connections (in Stress Testing)** columns indicate the results of the RDS performance stress testing. For services running on the live network, set the **max_connections** parameter.

Copyright © Huawei Technologies Co., Ltd.

Test Data of General-Enhanced DB Instances

Table 1-7 vCPU:Memory=1:2

vCPUs	Memory (GB)	Connections (in Stress Testing)	TPS	QPS	IOPS
2	4	1,500	966.39	19,328.36	See **About IOPS**.
4	8	2,500	1,494.59	29,892.24	
8	16	5,000	3,588.11	71,763.02	
16	32	10,000	6,485.36	129,708.2	
32	64	18,000	6,199.46	123,991.48	
64	128	30,000	5,509.54	110,192.49	

Table 1-8 CPU:Memory=1:4

vCPUs	Memory (GB)	Connections (in Stress Testing)	TPS	QPS	IOPS
2	8	2,500	868.63	17,399.15	See **About IOPS**.
4	16	5,000	1,995.58	39,911.92	
8	32	10,000	3,722.75	74,456.24	
16	64	18,000	6,657.96	133,162.9	
32	128	30,000	6,169.94	123,401.24	
64	256	60,000	5,456.48	109,131.22	

Table 1-9 CPU:Memory=1:8

vCPUs	Memory (GB)	Connections (in Stress Testing)	TPS	QPS	IOPS
2	16	5,000	884.77	17,695.67	See **About IOPS**.
4	32	10,000	1,870.16	37,403.31	
8	64	18,000	3,592.93	71,861.52	
16	128	30,000	6,361.57	127,235.54	
64	512	100,000	5,494.23	109,886.72	

Copyright © Huawei Technologies Co., Ltd.

2 RDS for PostgreSQL

2.1 Test Method

PostgreSQL is an open-source object-relational database management system with an emphasis on extensibility and standards compliance. It is known as the most advanced open-source database. It applies to business-oriented online transaction processing (OLTP) scenarios and supports NoSQL (JSON, XML, or hstore) and geographic information system (GIS) data types. It has earned a reputation for reliability and data integrity, and applies to Internet websites, location-based applications, and complex data object processing.

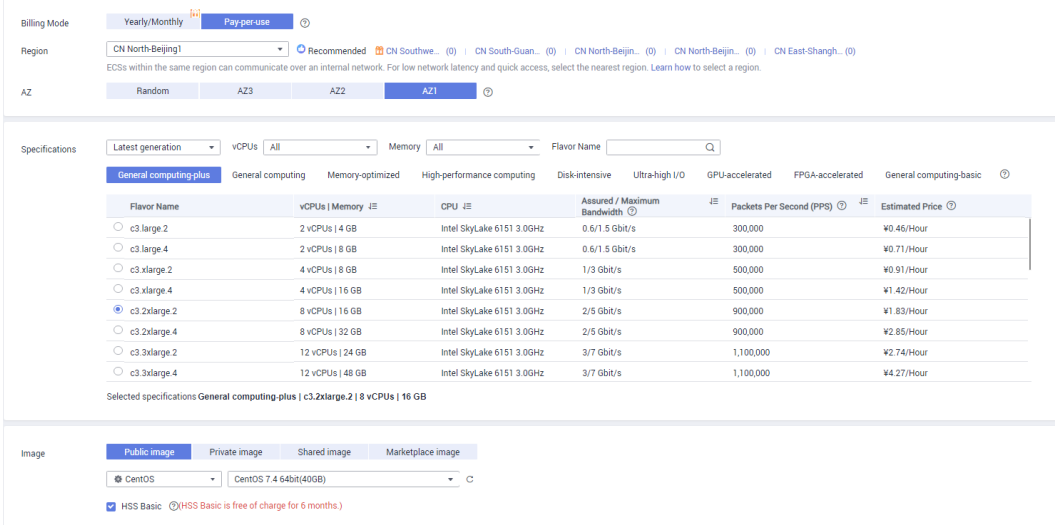
- RDS for PostgreSQL supports the postgis plugin, which provides excellent spatial performance at an international standard level.
- RDS for PostgreSQL applies to various scenarios and is cost-effective. You can flexibly scale resources based on your service requirements and pay for only what you use.

Test Environment

- Region: CN North-Beijing1
- AZ: AZ1
- ECS: generral computing | c3.2xlarge.2 | 8 vCPUs | 16 GB, CentOS7.4 64 bit image Bind an EIP to the ECS because additional compilation tools need to be installed on stress testing tools.

 Copyright © Huawei Technologies Co., Ltd.

Figure 2-1 ECS configuration



Test Tool

PostgreSQL provides a lightweight stress testing tool: pgbench. pgbench is a simple program for running benchmark tests on PostgreSQL. It runs the same sequence of SQL commands over and over, possibly in multiple concurrent database sessions.

Step 1 Use a remote connection tool, such as PuTTY, to connect to the ECS through an EIP. For details, see **Login Overview**.

Step 2 Download, compile, and install the PostgreSQL tool.

yum install -y https://dl.fedoraproject.org/pub/epel/epel-release-latest-7.noarch.rpm

yum install -y https://download.postgresql.org/pub/repos/yum/11/redhat/rhel-7-x86_64/pgdg-redhat-repo-latest.noarch.rpm

yum install -y postgresql11*

Step 3 Configure environment variables for the **postgres** user.

su - postgres

vi .bash_profile

export PS1="$USER@`/bin/hostname -s`-> "

export LANG=en_US.utf8

export PGHOME=/usr/pgsql-11

export LD_LIBRARY_PATH=$PGHOME/lib:/lib64:/usr/lib64:/usr/local/lib64:/lib:/usr/lib:/usr/local/lib:$LD_LIBRARY_PATH

export DATE=`date +"%Y%m%d%H%M"`

export PATH=$PGHOME/bin:$PATH:.

export MANPATH=$PGHOME/share/man:$MANPATH

 Copyright © Huawei Technologies Co., Ltd.

export PGHOST=XXXXXX

export PGPORT= XXXXXX

export PGDATABASE=postgres

export PGUSER= XXXXXX

export PGPASSWORD= XXXXXX

Set the values of **PGHOST**, **PGPORT**, **PGUSER**, and **PGPASSWORD** to the IP address, port number, login username, and password of RDS, respectively.

----End

Test Procedure

Replace the database name, connection IP address, and user password based on the site requirements.

Step 1 Modify RDS instance parameters and then reboot the PostgreSQL DB instance for the modification to take effect.

Table 2-1 Parameter configuration

Synchronize Model	Asynchronous
wal_compression	on
bgwriter_delay	500
bgwriter_flush_after	256
bgwriter_lru_maxpages	1000
bgwriter_lru_multiplier	10
checkpoint_timeout	1800
random_page_cost	1.1
max_wal_size	1/2 of the memory size of the PostgreSQL DB instance (change it based on the site requirements)
min_wal_size	1/8 of the memory size of the PostgreSQL DB instance (change it based on the site requirements)
shared_buffers	3/5 of the memory size of the PostgreSQL DB instance (change it based on the site requirements)

Step 2 Prepare for the test.

1. Create a test database.
   createdb ${dataname}

 Copyright © Huawei Technologies Co., Ltd.

2. Initialize the test data based on the size of the destination database.
   - To initialize 5 billion of data: **pgbench -i -s 50000**
   - To initialize 10 billion of data: **pgbench -i -s 10000**
   - To initialize 0.5 billion of data: **pgbench -i -s 5000**
   - To initialize 0.1 million of data: **pgbench -i -s 1000**
3. Create read-only and read/write test scripts.

   Create the read-only script ro.sql:

```
\set aid random_gaussian(1, :range, 10.0)
SELECT abalance FROM pgbench_accounts WHERE aid = :aid;
```

   Create the read/write script rw.sql:

```
\set aid random_gaussian(1, :range, 10.0)
\set bid random(1, 1 * :scale)
\set tid random(1, 10 * :scale)
\set delta random(-5000, 5000)
BEGIN;
UPDATE pgbench_accounts SET abalance = abalance + :delta WHERE aid = :aid;
SELECT abalance FROM pgbench_accounts WHERE aid = :aid;
UPDATE pgbench_tellers SET tbalance = tbalance + :delta WHERE tid = :tid;
UPDATE pgbench_branches SET bbalance = bbalance + :delta WHERE bid = :bid;
INSERT INTO pgbench_history (tid, bid, aid, delta, mtime) VALUES (:tid, :bid, :aid, :delta,
CURRENT_TIMESTAMP);
END;
```

Step 3 Perform the test.

Table 2-2 Test methods

Instance Class	Command
2 vCPUs | 8 GB	Read-only: **pgbench -M prepared -v -r -P 1 -f ./ro.sql -c 8 -j 8 -T 120 -D scale=1000 -D range=50000000 pgbench** **pgbench -M prepared -v -r -P 1 -f ./ro.sql -c 8 -j 8 -T 120 -D scale=1000 -D range=100000000 pgbench** Read/Write: **pgbench -M prepared -v -r -P 1 -f ./rw.sql -c 8 -j 8 -T 120 -D scale=1000 -D range=50000000 pgbench** **pgbench -M prepared -v -r -P 1 -f ./rw.sql -c 8 -j 8 -T 120 -D scale=1000 -D range=100000000 pgbench**

 Copyright © Huawei Technologies Co., Ltd.

Instance Class	Command
2 vCPUs | 16 GB	Read-only: **pgbench -M prepared -v -r -P 1 -f ./ro.sql -c 8 -j 8 -T 120 -D scale=1000 -D range=50000000 pgbench** **pgbench -M prepared -v -r -P 1 -f ./ro.sql -c 8 -j 8 -T 120 -D scale=1000 -D range=100000000 pgbench** Read/Write: **pgbench -M prepared -v -r -P 1 -f ./rw.sql -c 8 -j 8 -T 120 -D scale=1000 -D range=50000000 pgbench** **pgbench -M prepared -v -r -P 1 -f ./rw.sql -c 8 -j 8 -T 120 -D scale=1000 -D range=100000000 pgbench**
4 vCPUs | 16 GB	Read-only: **pgbench -M prepared -v -r -P 1 -f ./ro.sql -c 16 -j 16 -T 120 -D scale=5000 -D range=100000000 pgbench** **pgbench -M prepared -v -r -P 1 -f ./ro.sql -c 16 -j 16 -T 120 -D scale=5000 -D range=500000000 pgbench** Read/Write: **pgbench -M prepared -v -r -P 1 -f ./rw.sql -c 16 -j 16 -T 120 -D scale=5000 -D range=100000000 pgbench** **pgbench -M prepared -v -r -P 1 -f ./rw.sql -c 16 -j 16 -T 120 -D scale=5000 -D range=500000000 pgbench**
4 vCPUs | 32 GB	Read-only: **pgbench -M prepared -v -r -P 1 -f ./ro.sql -c 16 -j 16 -T 120 -D scale=5000 -D range=100000000 pgbench** **pgbench -M prepared -v -r -P 1 -f ./ro.sql -c 16 -j 16 -T 120 -D scale=5000 -D range=500000000 pgbench** Read/Write: **pgbench -M prepared -v -r -P 1 -f ./rw.sql -c 16 -j 16 -T 120 -D scale=5000 -D range=100000000 pgbench** **pgbench -M prepared -v -r -P 1 -f ./rw.sql -c 16 -j 16 -T 120 -D scale=5000 -D range=500000000 pgbench**

Copyright © Huawei Technologies Co., Ltd.

Instance Class	Command
8 vCPUs | 32 GB	Read-only: **pgbench -M prepared -v -r -P 1 -f ./ro.sql -c 32 -j 32 -T 120 -D scale=10000 -D range=100000000 pgbench** **pgbench -M prepared -v -r -P 1 -f ./ro.sql -c 32 -j 32 -T 120 -D scale=10000 -D range=500000000 pgbench** **pgbench -M prepared -v -r -P 1 -f ./ro.sql -c 32 -j 32 -T 120 -D scale=10000 -D range=1000000000 pgbench** Read/Write: **pgbench -M prepared -v -r -P 1 -f ./rw.sql -c 32 -j 32 -T 120 -D scale=10000 -D range=100000000 pgbench** **pgbench -M prepared -v -r -P 1 -f ./rw.sql -c 32 -j 32 -T 120 -D scale=10000 -D range=500000000 pgbench** **pgbench -M prepared -v -r -P 1 -f ./rw.sql -c 32 -j 32 -T 120 -D scale=10000 -D range=1000000000 pgbench**
8 vCPUs | 64 GB	Read-only: **pgbench -M prepared -v -r -P 1 -f ./ro.sql -c 32 -j 32 -T 120 -D scale=10000 -D range=100000000 pgbench** **pgbench -M prepared -v -r -P 1 -f ./ro.sql -c 32 -j 32 -T 120 -D scale=10000 -D range=500000000 pgbench** **pgbench -M prepared -v -r -P 1 -f ./ro.sql -c 32 -j 32 -T 120 -D scale=10000 -D range=1000000000 pgbench** Read/Write: **pgbench -M prepared -v -r -P 1 -f ./rw.sql -c 32 -j 32 -T 120 -D scale=10000 -D range=100000000 pgbench** **pgbench -M prepared -v -r -P 1 -f ./rw.sql -c 32 -j 32 -T 120 -D scale=10000 -D range=500000000 pgbench** **pgbench -M prepared -v -r -P 1 -f ./rw.sql -c 32 -j 32 -T 120 -D scale=10000 -D range=1000000000 pgbench**

Copyright © Huawei Technologies Co., Ltd.

Instance Class	Command
16 vCPUs | 64 GB	Read-only: **pgbench -M prepared -v -r -P 1 -f ./ro.sql -c 64 -j 64 -T 120 -D scale=10000 -D range=100000000 pgbench** **pgbench -M prepared -v -r -P 1 -f ./ro.sql -c 64 -j 64 -T 120 -D scale=10000 -D range=500000000 pgbench** **pgbench -M prepared -v -r -P 1 -f ./ro.sql -c 64 -j 64 -T 120 -D scale=10000 -D range=1000000000 pgbench** Read/Write: **pgbench -M prepared -v -r -P 1 -f ./rw.sql -c 64 -j 64 -T 120 -D scale=10000 -D range=100000000 pgbench** **pgbench -M prepared -v -r -P 1 -f ./rw.sql -c 64 -j 64 -T 120 -D scale=10000 -D range=500000000 pgbench** **pgbench -M prepared -v -r -P 1 -f ./rw.sql -c 64 -j 64 -T 120 -D scale=10000 -D range=1000000000 pgbench**
16 vCPUs | 128 GB	Read-only: **pgbench -M prepared -v -r -P 1 -f ./ro.sql -c 64 -j 64 -T 120 -D scale=10000 -D range=100000000 pgbench** **pgbench -M prepared -v -r -P 1 -f ./ro.sql -c 64 -j 64 -T 120 -D scale=10000 -D range=500000000 pgbench** **pgbench -M prepared -v -r -P 1 -f ./ro.sql -c 64 -j 64 -T 120 -D scale=10000 -D range=1000000000 pgbench** Read/Write: **pgbench -M prepared -v -r -P 1 -f ./rw.sql -c 64 -j 64 -T 120 -D scale=10000 -D range=100000000 pgbench** **pgbench -M prepared -v -r -P 1 -f ./rw.sql -c 64 -j 64 -T 120 -D scale=10000 -D range=500000000 pgbench** **pgbench -M prepared -v -r -P 1 -f ./rw.sql -c 64 -j 64 -T 120 -D scale=10000 -D range=1000000000 pgbench**

NOTE

- Scale multiplied by 100,000: indicates the test data volume.
- range: indicates the amount of active data.
- -c: indicates the number of test connections. The number of test connections is not equal to the maximum number of connections supported by the instance class. The maximum number of connections is not fixed.

----End

Test Metric

- TPS refers to the number of transactions executed per second by a database. Each transaction contains 18 SQL statements.

 Copyright © Huawei Technologies Co., Ltd.

- QPS refers to the number of SQL statements, including INSERT, SELECT, UPDATE, and DELETE statements, executed per second.
- Hot (active) data volume refers to the range of the number of query and update SQL records in the test.

2.2 PostgreSQL 9.5 Test Data

Table 2-3 describes the test results.

Table 2-3 Test results

Instance Class	Test Time (s)	Data Volume (Million)	Hot Data Volume (Million)	Scale	TPS (Read-Only)	QPS (Read-Only)	Average Latency (Read-Only)	TPS (Read/Write)	QPS (Read/Write)	Average Latency (Read/Write)
2 vCPUs | 16 GB	120	100	50	1,000	38,182.15	38,182.15	0.419	4,231.173	21,156	3.778
	120	100	100	1,000	35,673.1	35,673.1	0.896	4,330.887	21,654	7.375
4 vCPUs | 32 GB	120	500	100	5,000	69,462.87	69,462.87	0.46	8,218.108	41,091	7.774
	120	500	500	5,000	60,999.6	60,999.6	1.046	7,563.455	37,817	8.442
8 vCPUs | 64 GB	120	1,000	100	10,000	91,376.6	91,376.6	0.35	10,769.54	53,848	5.933
	120	1,000	500	10,000	87,050.39	87,050.39	0.688	9,827.995	49,140	6.499
	120	1,000	1,000	10,000	26,447.25	26,447.25	0.605	9,494.244	47,471	6.721
16 vCPUs	120	1,000	100	10,000	94,726.29	94,726.29	0.584	11,212.86	56,064	5.704

Copyright © Huawei Technologies Co., Ltd.

Instance Class	Test Time (s)	Data Volume (Million)	Hot Data Volume (Million)	Scale	TPS (Read-Only)	QPS (Read-Only)	Average Latency (Read-Only)	TPS (Read/Write)	QPS (Read/Write)	Average Latency (Read/Write)
| 128 GB	120	1,000	500	10,000	93,091	93,091	0.599	11,311.36	56,557	5.654
	120	1,000	1,000	10,000	92,007.3	92,007.3	0.609	11,036.39	55,182	5.796

2.3 PostgreSQL 9.6 Test Data

Test Result

Table 2-4 describes the test results.

Table 2-4 Test results

Instance Class	Test Time (s)	Data Volume (Million)	Hot Data Volume (Million)	Scale	TPS (Read-Only)	QPS (Read-Only)	Average Latency (Read-Only)	TPS (Read/Write)	QPS (Read/Write)	Average Latency (Read/Write)
2 vCPUs | 16 GB	120	100	50	1,000	36,542.44	36,542.44	0.875	3,880.385	19,402	16.448
	120	100	100	1,000	35,476.06	35,476.06	0.451	4,029.257	20,146	15.839
4 vCPUs	120	500	100	5,000	72,689.28	72,689.28	0.44	7,696.777	38,484	4.154

Copyright © Huawei Technologies Co., Ltd.

Instance Class	Test Time (s)	Data Volume (Million)	Hot Data Volume (Million)	Scale	TPS (Read-Only)	QPS (Read-Only)	Average Latency (Read-Only)	TPS (Read/Write)	QPS (Read/Write)	Average Latency (Read/Write)
| 32 GB	120	500	500	5,000	63,544.7	63,544.7	0.503	7,203.794	36,019	8.871
8 vCPUs | 64 GB	120	1,000	100	10,000	98,303.09	98,303.09	0.324	11,100.02	55,500	5.761
	120	1,000	500	10,000	92,038.66	92,038.66	0.347	10,053.22	50,266	6.355
	120	1,000	1,000	10,000	25,499.84	25,499.84	0.627	10,251.77	51,259	6.24
16 vCPUs | 128 GB	120	1,000	100	10,000	98,165.3	98,165.3	0.326	11,767.64	58,838	5.436
	120	1,000	500	10,000	96,328.43	96,328.43	0.332	11,142.02	55,710	5.738
	120	1,000	1,000	10,000	94,182.02	94,182.02	0.339	11,042.51	55,213	5.79

2.4 PostgreSQL 10 Test Data

Test Result

Table 2-5 describes the test results.

Copyright © Huawei Technologies Co., Ltd.

Table 2-5 Test results

Instance Class	Test Time (s)	Data Volume (Million)	Hot Data Volume (Million)	Scale	TPS (Read-Only)	QPS (Read-Only)	Average Latency (Read-Only)	TPS (Read/Write)	QPS (Read/Write)	Average Latency (Read/Write)
2 vCPUs | 16 GB	120	100	50	1,000	34,997.24	34,997.24	0.913	4,116.554	20,583	15.507
	120	100	100	1,000	33,386.52	33,386.52	0.479	4,117.187	20,586	15.489
4 vCPUs | 32 GB	120	500	100	5,000	67,719.94	67,719.94	0.472	7,070.501	35,353	4.518
	120	500	500	5,000	57,937.06	57,937.06	0.552	6,039.305	30,197	5.294
8 vCPUs | 64 GB	120	1,000	100	10,000	94,576.89	94,576.89	0.338	11,082.72	55,414	5.766
	120	1,000	500	10,000	91,279.68	91,279.68	0.35	10,102.88	50,514	6.323
	120	1,000	1,000	10,000	26,045.6	26,045.6	0.614	10,148.18	50,741	6.296
16 vCPUs | 128 GB	120	1,000	100	10,000	104,788.8	104,788.8	0.305	12,115.58	60,578	5.279
	120	1,000	500	10,000	103,259.5	103,259.5	0.309	11,941.04	59,705	5.358
	120	1,000	1,000	10,000	101,724.5	101,724.5	0.314	12,038.66	60,193	5.31

Copyright © Huawei Technologies Co., Ltd.

2.5 PostgreSQL 11 Test Data

Test Result

Table 2-6 Test results

Instance Class	Test Time (s)	Data Volume (Million)	Hot Data Volume (Million)	Scale	TPS (Read-Only)	QPS (Read-Only)	Average Latency (Read-Only)	TPS (Read/Write)	QPS (Read/Write)	Average Latency (Read/Write)
2 vCPUs | 16 GB	120	100	50	1,000	27,492.9	27,492.9	1.163	3,584.291	17,921	8.914
	120	100	100	1,000	28,846.86	28,846.86	0.554	3,488.25	17,441	9.16
4 vCPUs | 32 GB	120	500	100	5,000	56,261.5	56,261.5	0.568	6,354.399	31,772	20.074
	120	500	500	5,000	47,058.23	47,058.23	0.679	5,825.554	29,128	21.897
8 vCPUs | 64 GB	120	1,000	100	10,000	95,182.58	95,182.58	0.336	10,721.74	53,609	11.895
	120	1,000	500	10,000	89,324.21	89,324.21	0.358	10,079.84	50,399	12.647
	120	1,000	1,000	10,000	23,774.92	23,774.92	0.673	9,017.991	45,090	7.078
16 vCPUs | 128 GB	120	1,000	100	10,000	104,788.8	104,788.8	0.305	12,115.58	60,578	5.279
	120	1,000	500	10,000	103,259.5	103,259.5	0.309	11,941.04	59,705	5.358

Copyright © Huawei Technologies Co., Ltd.

Instance Class	Test Time (s)	Data Volume (Million)	Hot Data Volume (Million)	Scale	TPS (Read-Only)	QPS (Read-Only)	Average Latency (Read-Only)	TPS (Read/Write)	QPS (Read/Write)	Average Latency (Read/Write)
	120	1,000	1,000	10,000	101,724.5	101,724.5	0.314	12,038.66	60,193	5.31

2.6 PostgreSQL 11 Enhanced Edition Test Data

Test Result

Table 2-7 describes the test results.

Table 2-7 Test results

Instance Class	Test Time (s)	Data Volume (Million)	Hot Data Volume (Million)	Scale	TPS (Read-Only)	QPS (Read-Only)	Average Latency (Read-Only)	TPS (Read/Write)	QPS (Read/Write)	Average Latency (Read/Write)
2 vCPUs | 16 GB	120	100	50	1,000	29,662.82	29,662.82	0.539	3,212.977	16,065	4.975
	120	100	100	1,000	28,749.75	28,749.75	0.556	3,376.902	16,885	4.734
4 vCPUs | 32 GB	120	500	100	5,000	58,017.65	58,017.65	0.551	6,877.078	34,385	18.544
	120	500	500	5,000	45,958.22	45,958.22	2.768	6,038.081	30,190	21.113

Copyright © Huawei Technologies Co., Ltd.

Instance Class	Test Time (s)	Data Volume (Million)	Hot Data Volume (Million)	Scale	TPS (Read-Only)	QPS (Read-Only)	Average Latency (Read-Only)	TPS (Read/Write)	QPS (Read/Write)	Average Latency (Read/Write)
8 vCPUs | 64 GB	120	500	100	5,000	92,893.5	92,893.5	0.344	10,584.47	52,922	12.051
	120	500	500	5,000	90,279.03	90,279.03	0.354	10,650.45	53,252	11.976
	120	1,000	100	10,000	94,563.73	94,563.73	0.338	9,910.109	49,551	12.876
	120	1,000	500	10,000	86,176.2	86,176.2	0.371	9,334.534	46,673	13.665
	120	1,000	1,000	10,000	22,945.38	22,945.38	0.697	9,139.488	45,697	13.96
16 vCPUs | 128 GB	120	1,000	100	10,000	91,045.98	91,045.98	0.635	11,684.77	58,424	5.474
	120	1,000	500	10,000	90,318.33	90,318.33	0.643	11,591.92	57,960	5.516
	120	1,000	1,000	10,000	89,622.71	89,622.71	0.649	11,519.55	57,598	5.553

2.7 PostgreSQL 12 Test Data

Test Result

Table 2-8 describes the test results.

 Copyright © Huawei Technologies Co., Ltd.

Table 2-8 Test results

Instance Class	Test Time (s)	Data Volume (Million)	Hot Data Volume (Million)	Scale	TPS (Read-Only)	QPS (Read-Only)	Average Latency (Read-Only)	TPS (Read/Write)	QPS (Read/Write)	Average Latency (Read/Write)
2 vCPUs | 16 GB	120	100	50	1,000	36,927.61	36,927.61	0.866	3,840.838	19,204	4.163
	120	100	100	1,000	35,618.78	35,618.78	0.449	4,080.838	20,404	3.918
4 vCPUs | 32 GB	120	500	100	5,000	71,903.41	71,903.41	0.445	7,464.693	37,323	17.08
	120	500	500	5,000	60,513.53	60,513.53	1.056	7,336.201	36,681	17.35
8 vCPUs | 64 GB	120	500	100	5,000	99,406.14	99,406.14	0.321	11,642.26	58,211	5.491
	120	500	500	5,000	97,165.48	97,165.48	0.328	11,212.74	56,064	5.699
	120	1,000	100	10,000	96,790.84	96,790.84	0.329	10,707.55	53,538	5.97
	120	1,000	500	10,000	92,442.36	92,442.36	0.345	9,630.608	48,153	3.32
	120	1,000	1,000	10,000	23,799.6	23,799.6	0.672	9,167.153	45,836	6.962
16 vCPUs | 128 GB	120	1,000	100	10,000	103,650.7	103,650.7	0.308	14,034.1	70,170	4.557

Copyright © Huawei Technologies Co., Ltd.

Instance Class	Test Time (s)	Data Volume (Million)	Hot Data Volume (Million)	Scale	TPS (Read-Only)	QPS (Read-Only)	Average Latency (Read-Only)	TPS (Read/Write)	QPS (Read/Write)	Average Latency (Read/Write)
	120	1,000	500	10,000	102,389.5	102,389.5	0.312	12,914.5	64,572	4.952
	120	1,000	1,000	10,000	100,970.1	100,970.1	0.317	13,379.79	66,899	4.775

Copyright © Huawei Technologies Co., Ltd.

3 PostgreSQL

3.1 Test Method

PostgreSQL is an open-source object-relational database management system with an emphasis on extensibility and standards compliance. It is known as the most advanced open-source database. It applies to business-oriented online transaction processing (OLTP) scenarios and supports NoSQL (JSON, XML, or hstore) and geographic information system (GIS) data types. It has earned a reputation for reliability and data integrity, and applies to Internet websites, location-based applications, and complex data object processing.

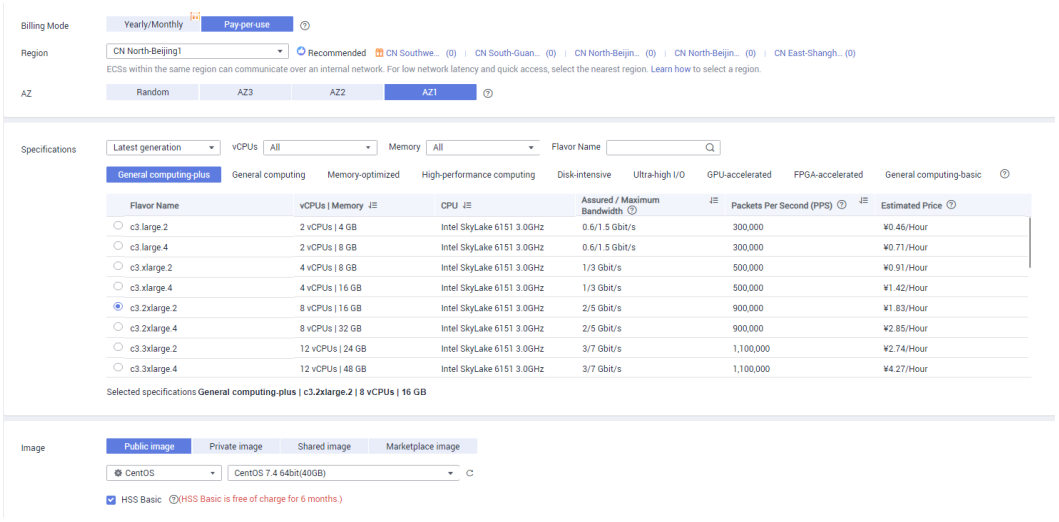
- RDS for PostgreSQL supports the postgis plugin, which provides excellent spatial performance at an international standard level. RDS for PostgreSQL offers functions similar to those of Oracle database systems, but is less expensive.
- RDS for PostgreSQL applies to various scenarios and is cost-effective. You can flexibly scale resources based on your service requirements and pay for only what you use.

Test Environment

- Region: CN North-Beijing1
- AZ: AZ1
- ECS: general computing | c3.2xlarge.2 | 8 vCPUs | 16 GB, CentOS7.4 64 bit image Bind an EIP to the ECS because additional compilation tools need to be installed on stress testing tools.

 Copyright © Huawei Technologies Co., Ltd.

Figure 3-1 ECS configuration



Test Tool

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 1.0.12 is used as an example. Run the following commands to install it:

#wget -c https://github.com/akopytov/sysbench/archive/1.0.12.zip

#yum install make automake libtool pkgconfig libaio-devel postgresql-devel

#unzip 1.0.12.zip

#cd sysbench-1.0.12

#./autogen.sh

#./configure --with-pgsql --without-mysql

#make

#make install

Test Procedure

Replace the database name, connection IP address, and user password based on the site requirements.

Step 1 Import data.

1. Run the following commands to log in to a database and create the test database **loadtest**:

   psql -h*<host>* **-p5432 "dbname=postgres user=root password=<password>" <<TEST**

   create database loadtest;

   TEST

 Copyright © Huawei Technologies Co., Ltd.

2. Run the following command to import the test background data to the **loadtest** database:

   sysbench --test=/usr/local/share/sysbench/tests/include/oltp_legacy/oltp.lua --db-driver=pgsql --pgsql-db=loadtest --pgsql-user=root --pgsql-password=*<password>* **--pgsql-port=5432 --pgsql-host=***<host>* **--oltp-tables-count=64 --oltp-table-size=10000000 --num-threads=20 prepare**

Step 2 Run the following command to perform a stress testing:

sysbench --test=/usr/local/share/sysbench/tests/include/oltp_legacy/oltp.lua --db-driver=pgsql --pgsql-db=loadtest --pgsql-user=root --pgsql-password=*<password>* **--pgsql-port=5432 --pgsql-host=***<host>* **--oltp-tables-count=64 --oltp-table-size=10000000 --max-time=3600 --max-requests=0 --num-threads=64 --report-interval=3 --forced-shutdown=1 run**

Step 3 Run the following command to delete the test data:

sysbench --test=/usr/local/share/sysbench/tests/include/oltp_legacy/oltp.lua --db-driver=pgsql --pgsql-db=loadtest --pgsql-user=root --pgsql-password=*<password>* **--pgsql-port=5432 --pgsql-host=***<host>* **--oltp-tables-count=64 --oltp-table-size=10000000 --max-time=3600 --max-requests=0 --num-threads=200 cleanup**

----End

Testing Model

1. Table structure:

   CREATE TABLE `sbtest` (
   `id` INTEGER IDENTITY(1,1) NOT NULL,
   `k` INTEGER DEFAULT '0' NOT NULL,
   `c` CHAR(120) DEFAULT '' NOT NULL,
   `pad` CHAR(60) DEFAULT '' NOT NULL,
   PRIMARY KEY (`id`)
)

2. Read/write ratio:

   The default transaction submitted by sysbench contains 18 SQL statements. The details are as follows:

   - Ten primary key SELECT statements:

     SELECT c FROM ${rand_table_name} where id=${rand_id};

   - Four range SELECT statements:

     SELECT c FROM ${rand_table_name} WHERE id BETWEEN ${rand_id_start} AND ${rand_id_end};

     SELECT SUM(K) FROM ${rand_table_name} WHERE id BETWEEN ${rand_id_start} AND ${rand_id_end};

     SELECT c FROM ${rand_table_name} WHERE id BETWEEN ${rand_id_start} AND ${rand_id_end} ORDER BY c;

     SELECT DISTINCT c FROM ${rand_table_name} WHERE id BETWEEN ${rand_id_start} AND ${rand_id_end} ORDER BY c;

 Copyright © Huawei Technologies Co., Ltd.

  - Two UPDATE statements:

    UPDATE ${rand_table_name} SET k=k+1 WHERE id=${rand_id}

    UPDATE ${rand_table_name} SET c=${rand_str} WHERE id=${rand_id}
  - One DELETE statement:

    DELETE FROM ${rand_table_name} WHERE id=${rand_id}
  - One INSERT statement:

    INSERT INTO ${rand_table_name} (id, k, c, pad) VALUES (${rand_id},${rand_k},${rand_str_c},${rand_str_pad})

Test Metric

- TPS refers to the number of transactions executed per second by a database. Each transaction contains 18 SQL statements.
- QPS refers to the number of SQL statements, including INSERT, SELECT, UPDATE, and DELETE statements, executed per second.

3.2 PostgreSQL 9.6 Test Data

NOTICE

The **Connections (in Stress Testing)** columns indicate the results of the RDS performance stress testing. For services running on the live network, set the **max_connections** parameter.

Test Data of General-Enhanced DB Instances

Table 3-1 vCPU:Memory=1:2

vCPUs	Memory (GB)	Connections (in Stress Testing)	TPS	QPS	IOPS
1	2	300	150	3,086	1,200
2	4	500	263	5,257	2,000
4	8	1,200	578	11,554	6,000
60	128	12,000	1,398	27,965	20,000

Copyright © Huawei Technologies Co., Ltd.

Table 3-2 CPU:Memory=1:4

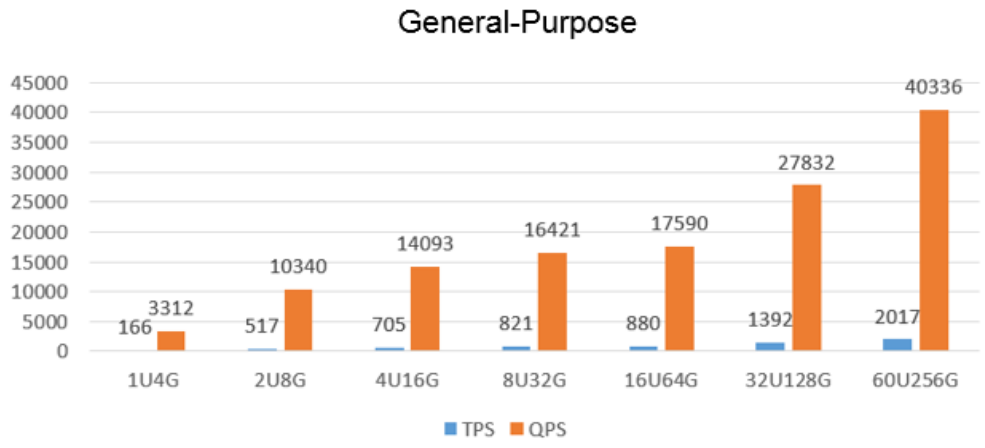
vCPUs	Memory (GB)	Connections (in Stress Testing)	TPS	QPS	IOPS
1	4	500	166	3,312	1,400
2	8	1,200	517	10,340	4,000
4	16	2,500	705	14,093	8,000
8	32	5,000	821	16,421	12,000
16	64	10,000	880	17,590	16,000
32	128	12,000	1,392	27,832	20,000
60	256	20,000	2,017	40,336	20,000

Table 3-3 CPU:Memory=1:8

vCPUs	Memory (GB)	Connections (in Stress Testing)	TPS	QPS	IOPS
2	16	2,500	554	11,099	5,000
4	32	5,000	781	15,614	7,000
8	64	5,000	931	18,613	11,000

Test Data of General-Enhanced DB Instances

Figure 3-2 General-purpose



 Copyright © Huawei Technologies Co., Ltd.

Figure 3-3 Compute

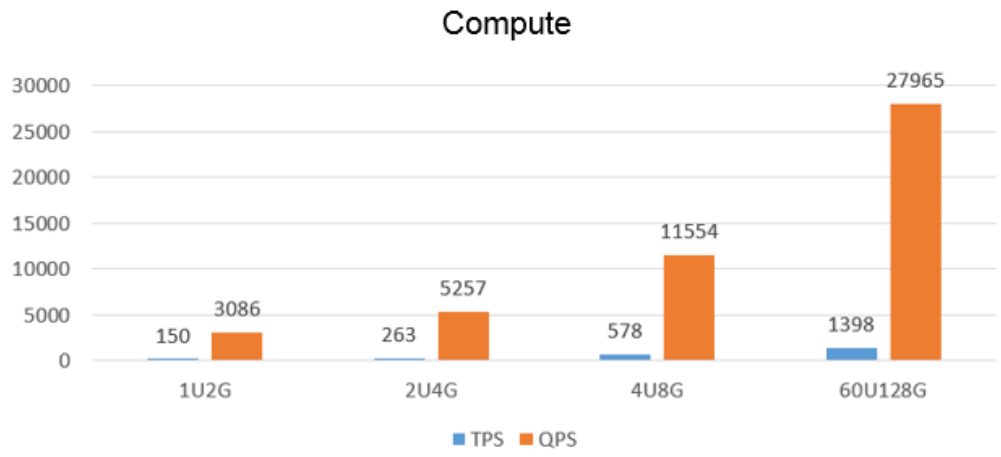
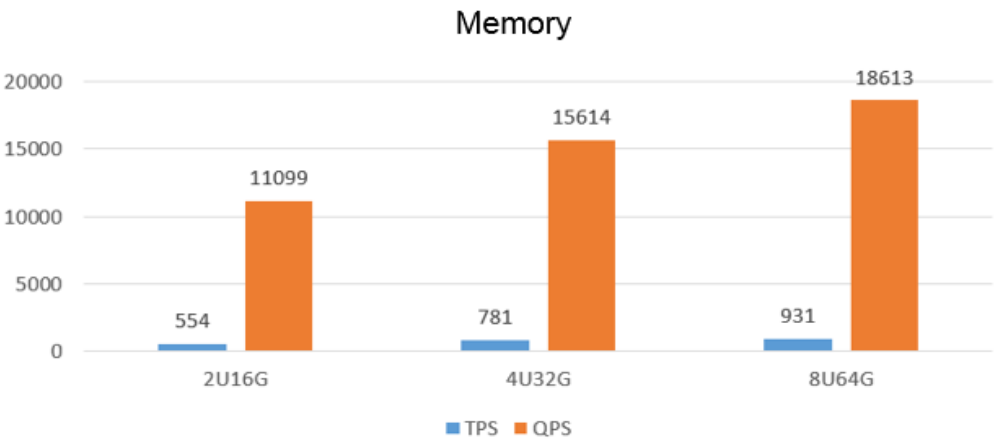


Figure 3-4 Memory



 Copyright © Huawei Technologies Co., Ltd.

4 RDS for SQL Server

4.1 Test Method

HUAWEI CLOUD SQL Server is an online Microsoft SQL Server-compatible relational database service. RDS for SQL Server can provide more secure, reliable, stable services than the open-source database of the Community Edition. Based on the primary/standby architecture, it provides all necessary functions you need, including backup, restoration, monitoring, and migration. It supports two billing mode: yearly/monthly and pay-per use.

Test Environment

- Region: CN North-Beijing1
- AZ: AZ1
- ECS specifications: high-performance computing | hc2.2xlarge.2 | 8 vCPUs | 16 GB, SSD disk, 200 GB of storage, Windows Server 2012 R2 Standard 64bit image and VPC network

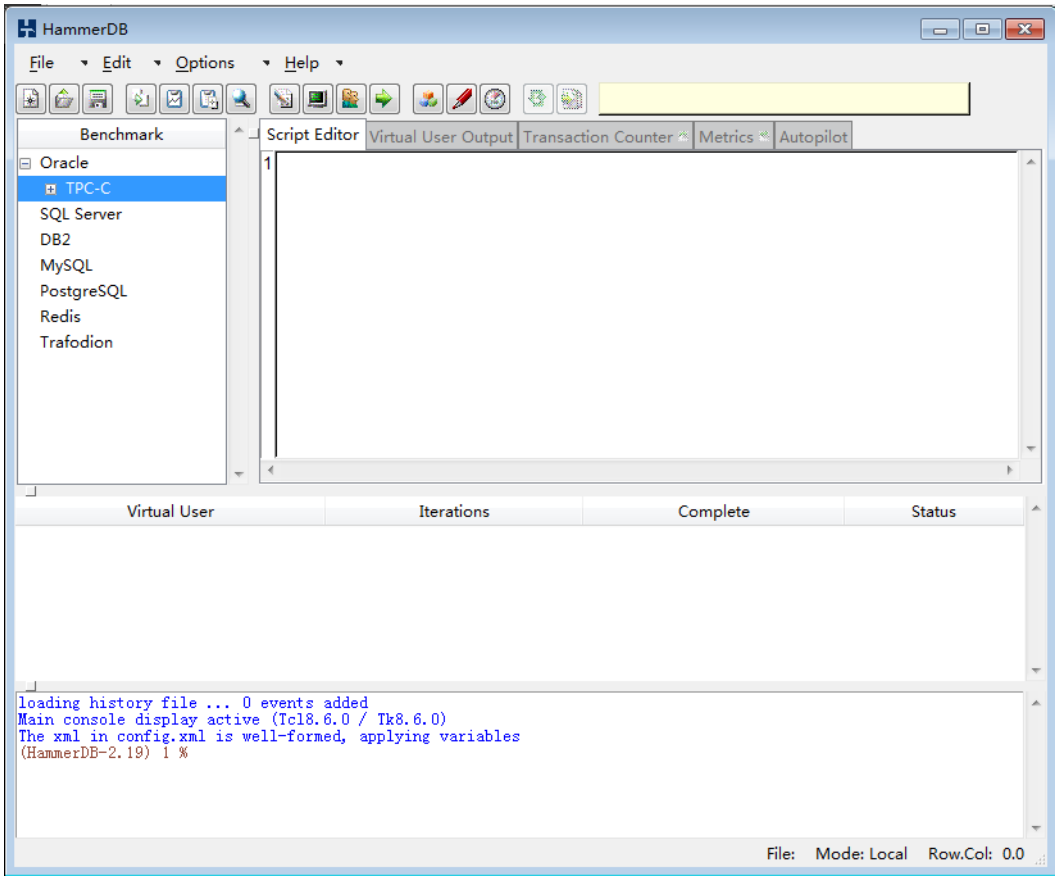
Test Tool

HammerDB is a graphical open-source database stress testing and benchmarking tool for Linux and Windows to test databases running on any operating system. HammerDB is automated, multi-threaded and extensible with dynamic scripting support. You can use HammerDB to create a test schema, load data, and simulate workloads of multiple virtual users on databases in online transaction processing (OLTP) and online analytical processing (OLAP) scenarios.

HammerDB 2.19 is used as an example. **Download the latest version.**

HammerDB started

 Copyright © Huawei Technologies Co., Ltd.



Test Benchmarks

The Transaction Processing Performance Council (TPC) is a non-profit corporation founded to define transaction processing and database benchmarks and to disseminate objective, verifiable performance data to the industry. TPC provides multiple test benchmarks, such as TPC-A, TPC-C, and TPC-H. For details, see the official document. TPC-C is different from and more complex than TPC-A because of its multiple transaction types, complex databases, and overall execution structure.

This test uses the TPC-C test benchmark.

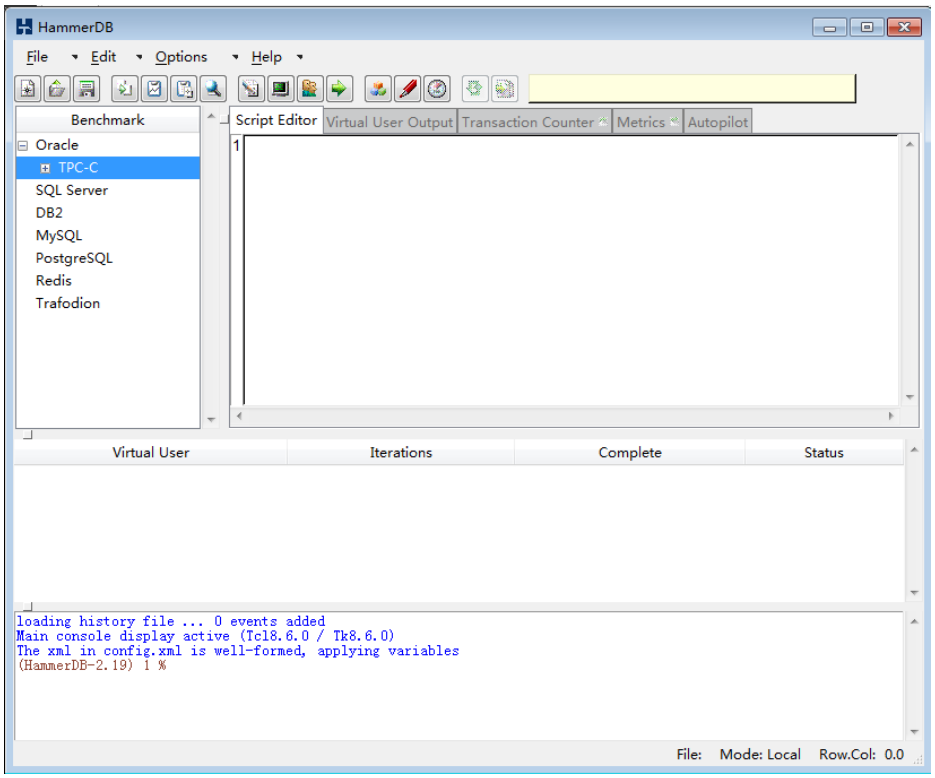
The test model is developed by HUAWEI CLOUD based on HammerDB without any optimization and modification on the model structure.

Test Procedure

Step 1 Start HammerDB.

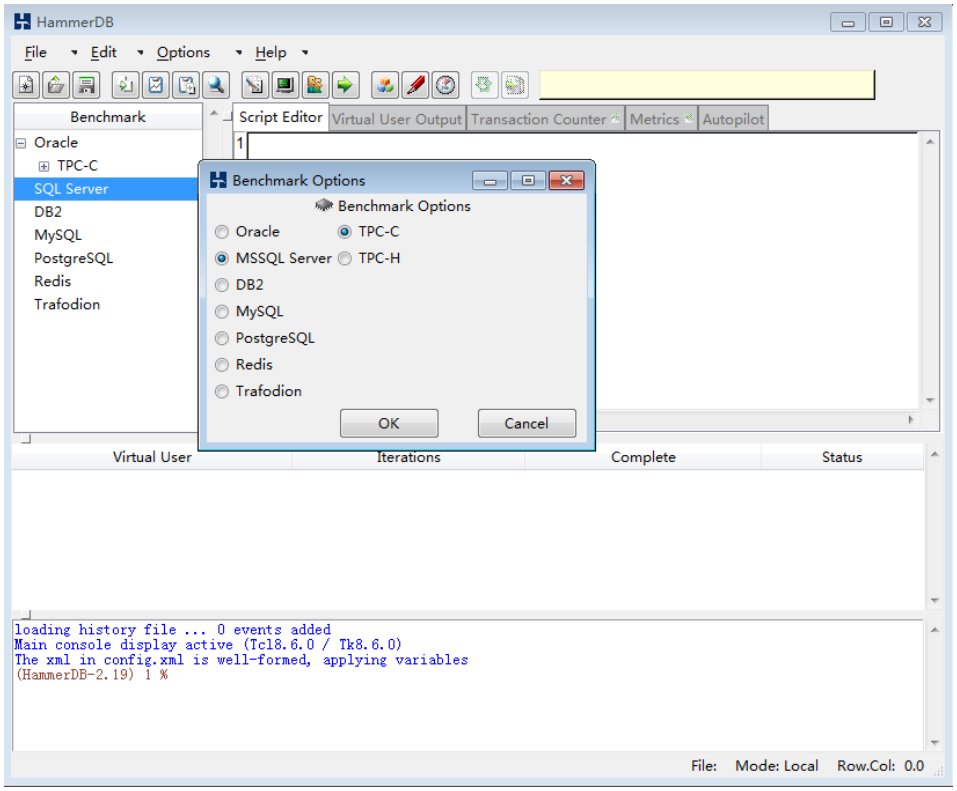
 Copyright © Huawei Technologies Co., Ltd.

Figure 4-1 HammerDB started



Step 2 In the **Benchmark** area, double-click **SQL Server**. In the displayed dialog box, select **MSSQL Server** and **TPC-C**, and click **OK**.

Figure 4-2 Benchmark Options



 Copyright © Huawei Technologies Co., Ltd.

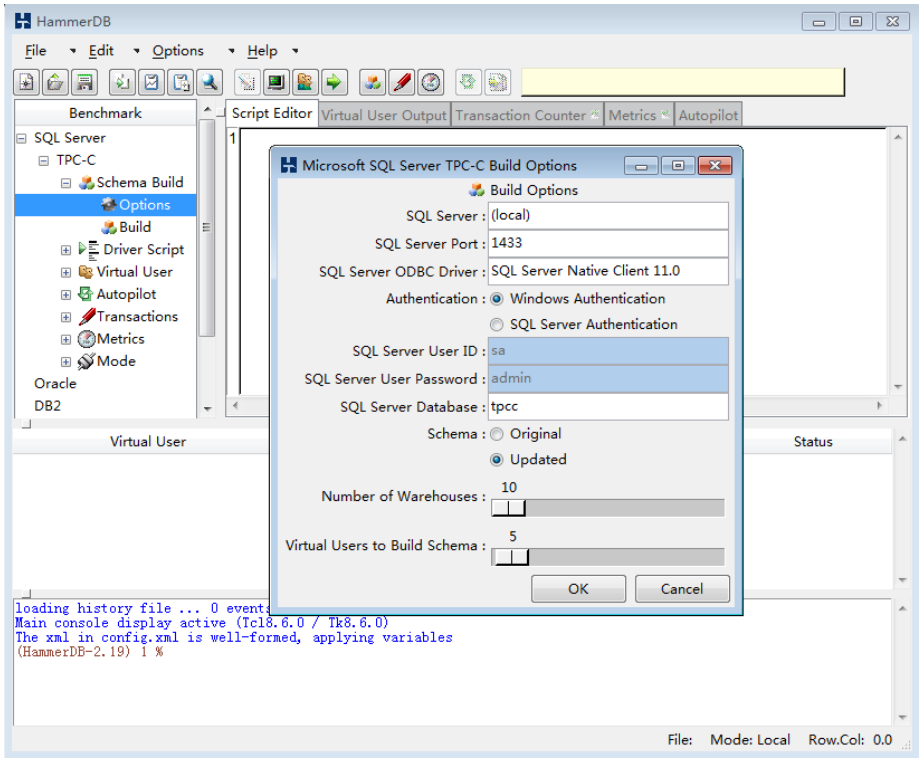
Step 3 Configure the connection information and create the object database **tpcc**.

Choose **SQL Server** > **TPC-C** > **Schema Build** and double-click **Options**.

NOTICE

In the displayed dialog box, select **Updated** for **Schema**.

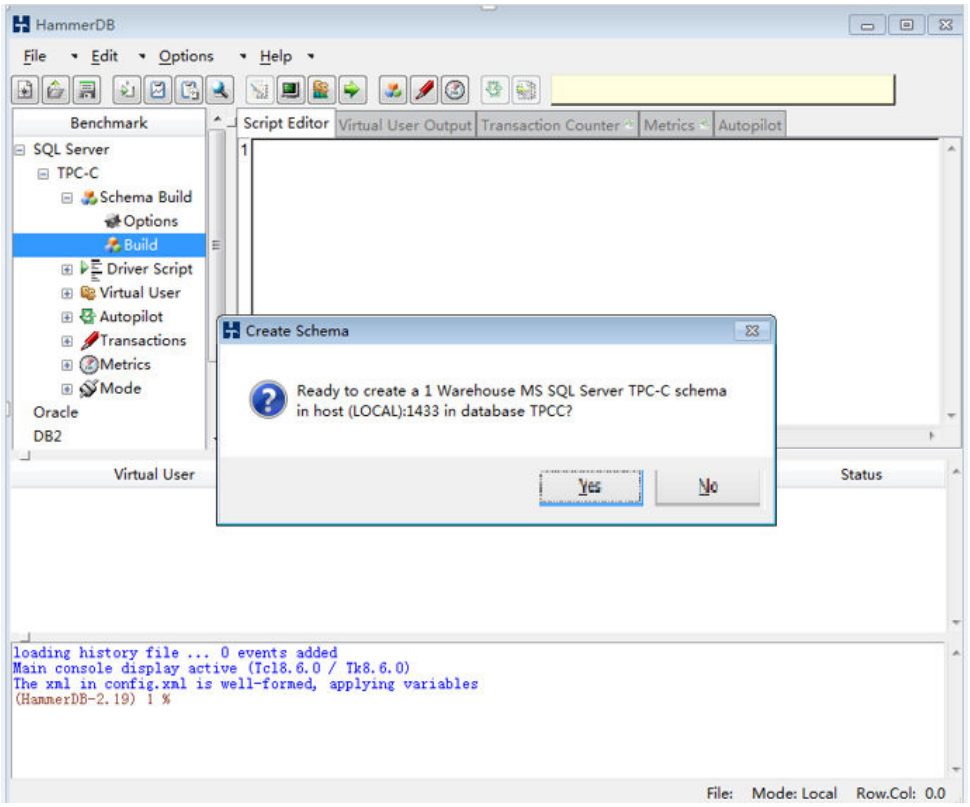
Figure 4-3 Microsoft SQL Server TPC-C Build Options



Step 4 Choose **SQL Server** > **TPC-C** > **Schema Build** > **Build**. In the displayed dialog box, click **Yes** to create a schema.

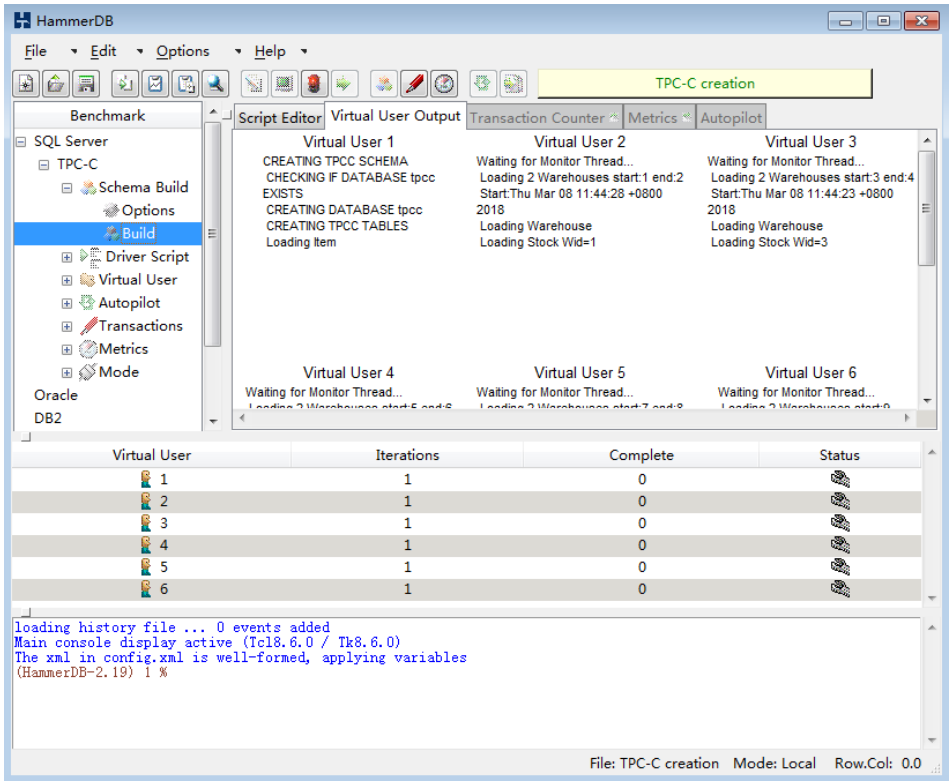
Copyright © Huawei Technologies Co., Ltd.

Figure 4-4 Build



Wait until the initialization is complete.

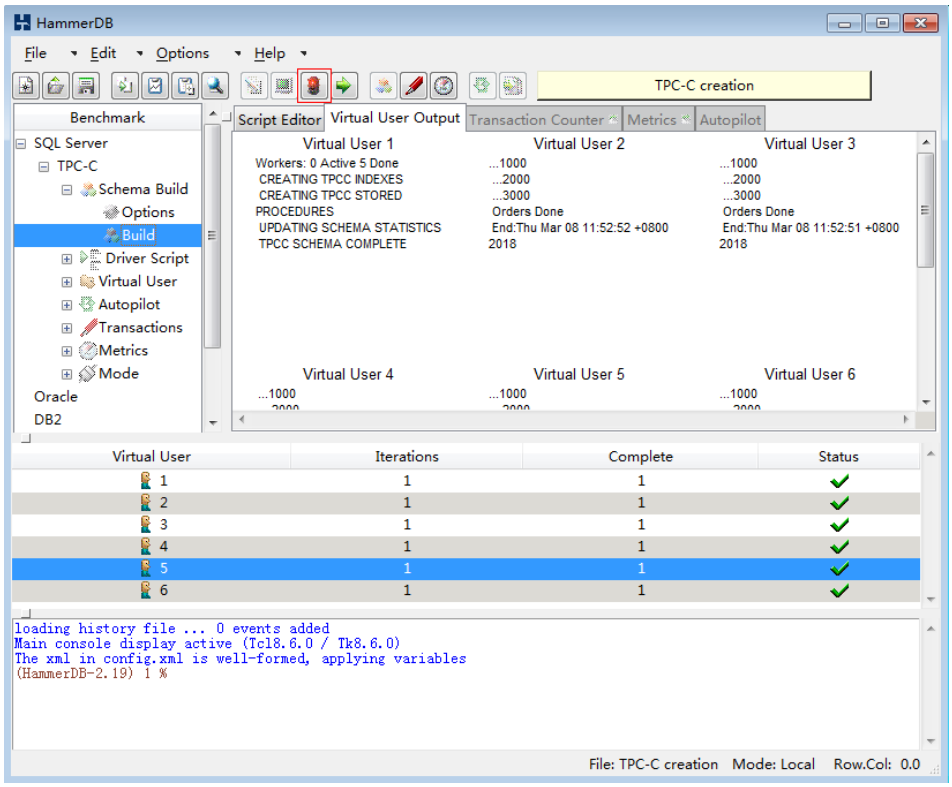
Figure 4-5 Initialization completed



 Copyright © Huawei Technologies Co., Ltd.

Step 5 Click  to stop the execution.

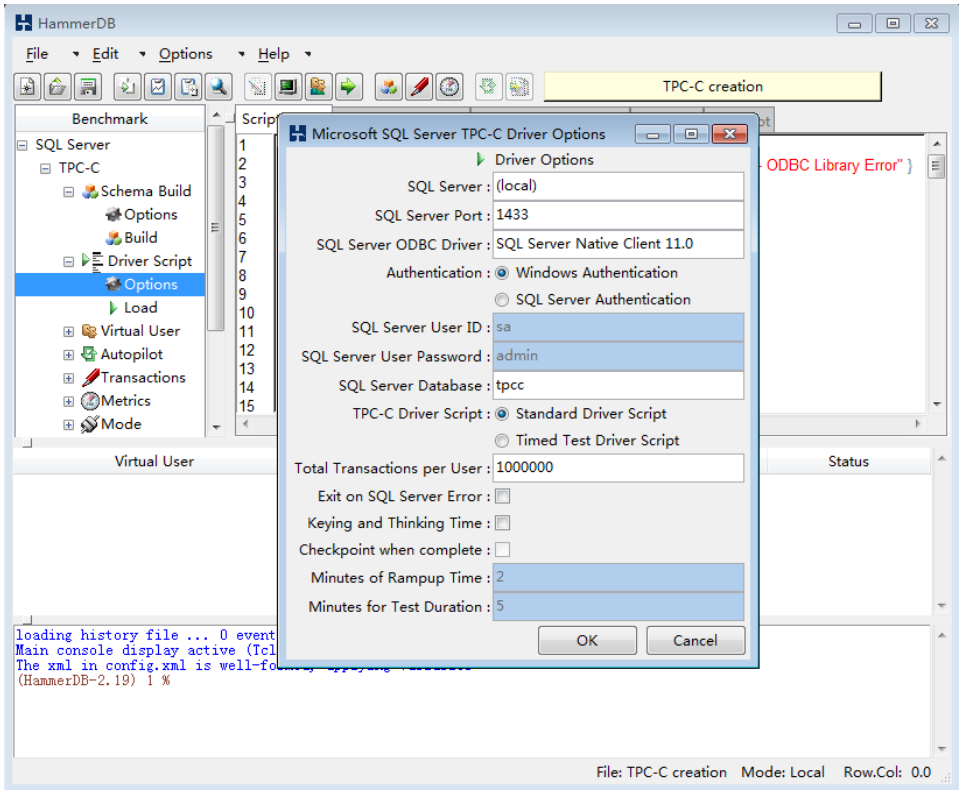
Figure 4-6 Execution stopped



Step 6 Select **SQL Server** > **TPC-C** > **Driver Script** and double-click **Options** to ensure that the connection information is correct.

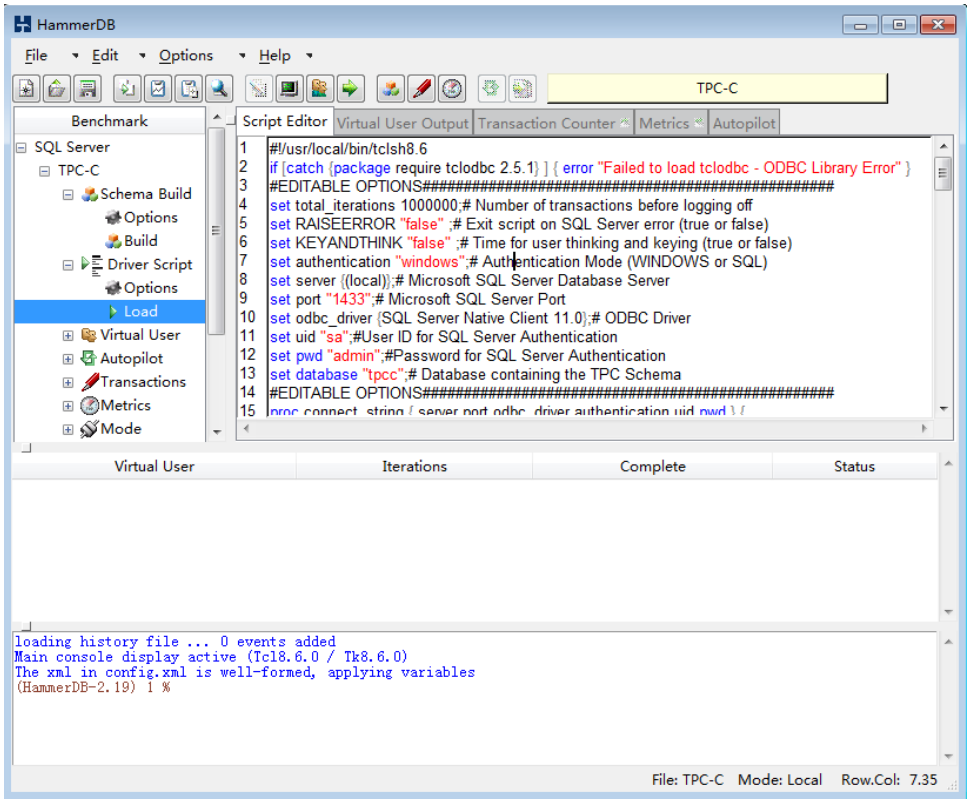
 Copyright © Huawei Technologies Co., Ltd.

Figure 4-7 Checking the connection information



Step 7 Choose **SQL Server** > **TPC-C** > **Driver Script** and double-click **Load**.

Figure 4-8 Load

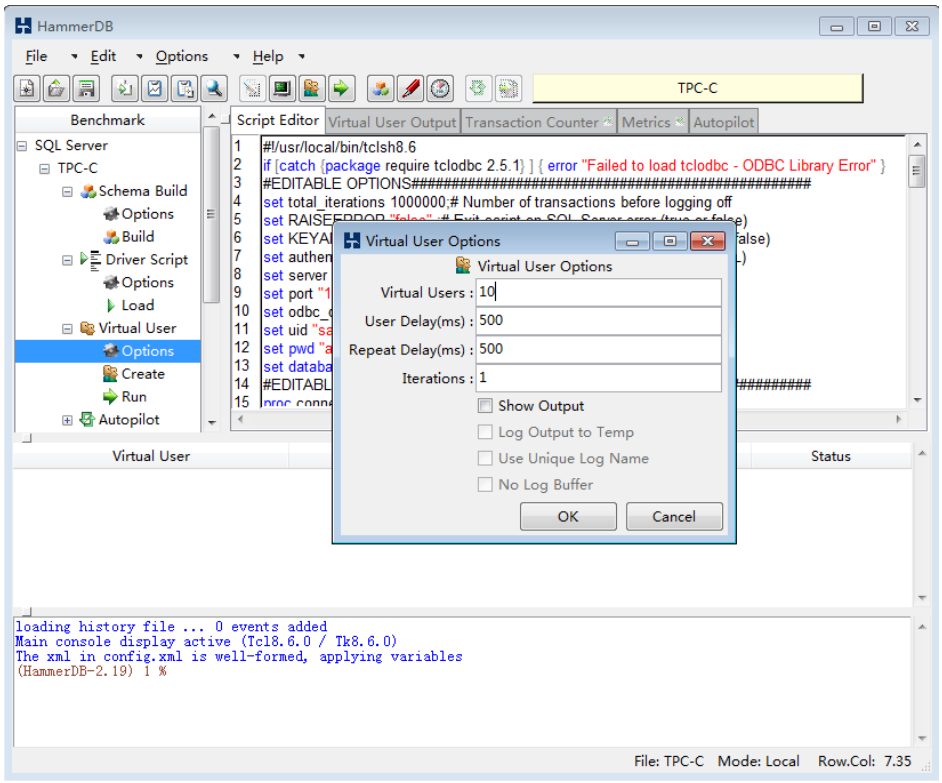


 Copyright © Huawei Technologies Co., Ltd.

Step 8 Choose **SQL Server** > **TPC-C** > **Virtual User** and double-click **Options**. In the displayed dialog box, you can adjust the number of virtual users repeatedly to generate test results until the Transactions Per Minute (TPM) peak values become consistent. TPM is an important benchmark for measuring database performance.

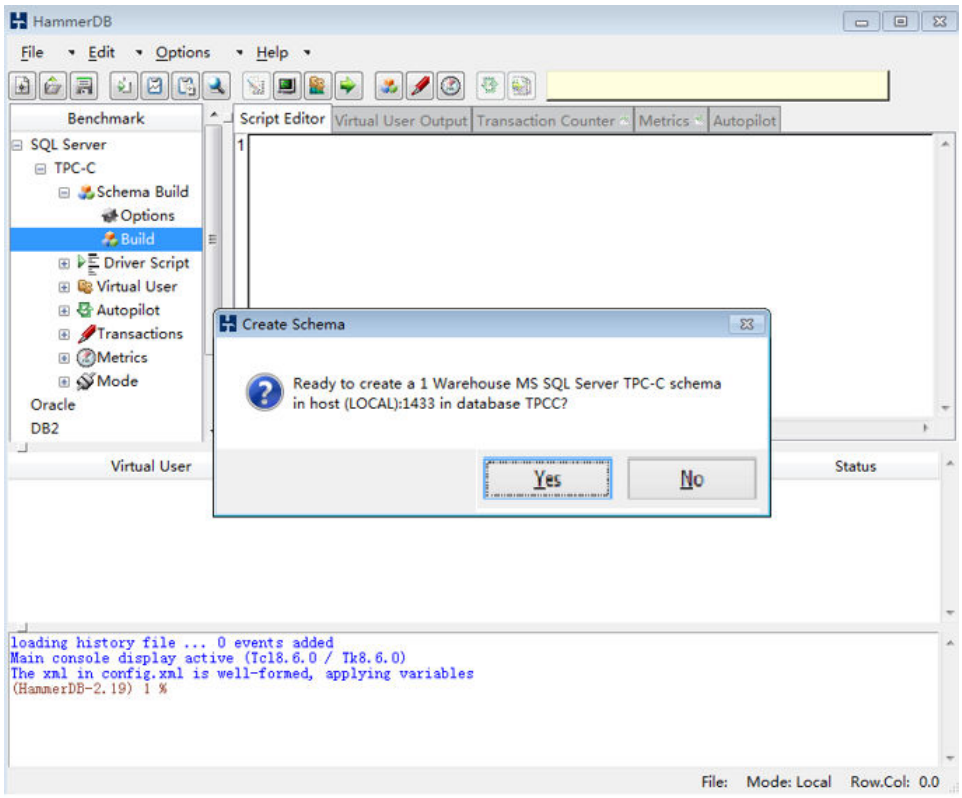
NOTICE

You are advised not to select **Show Output** because the client may not respond.



Step 9 Click **Yes**.

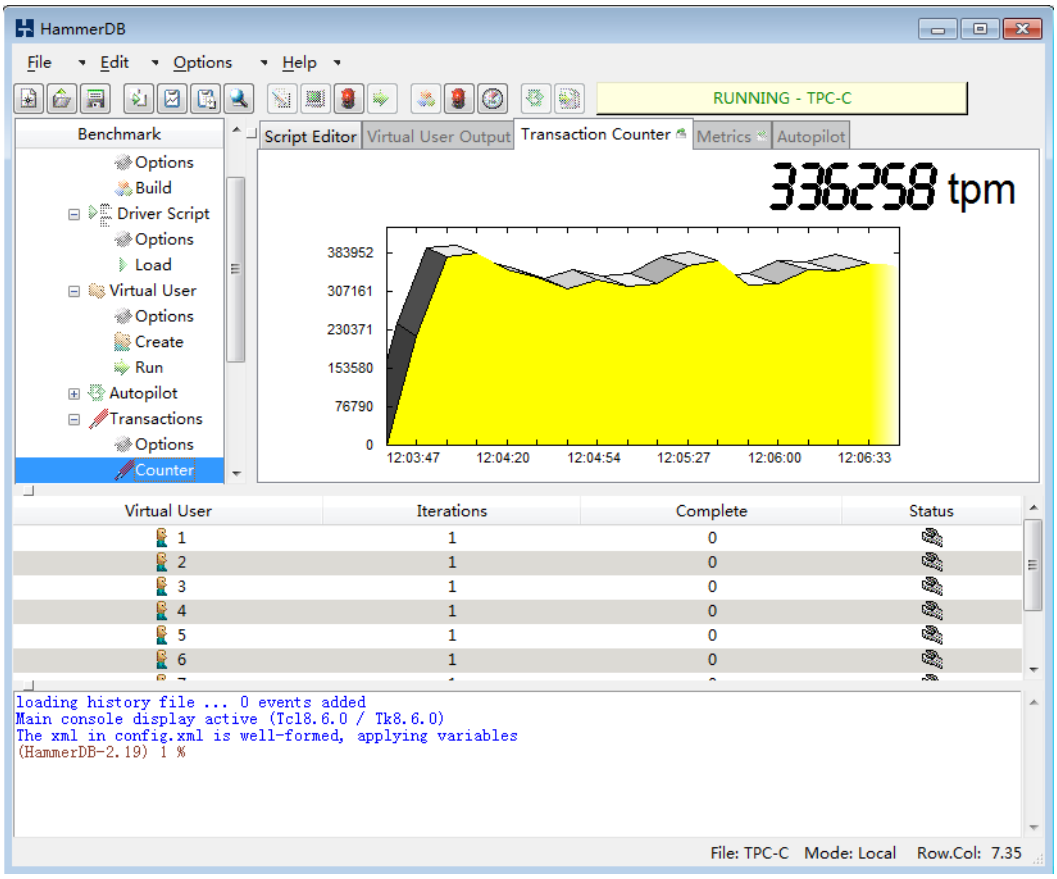
 Copyright © Huawei Technologies Co., Ltd.



Step 10 Choose **Virtual User** and double-click **Run** until the TPM peak value is stable.

 Copyright © Huawei Technologies Co., Ltd.

Figure 4-9 Stable TPM peak value



----End

4.2 Test Data

Concepts

Transactions per minute (TPM): Number of simulated orders processed by the system in the TPCC standard model per minute.

TPS: Number of simulated orders processed by the system in the TPCC standard model per second.

Input/output operations per second (IOPS): Number of disk read/write operations per second. IOPS in this document refers to the IOPS displayed when the performance reaches the peak in the stress testing, instead of the maximum IOPS capability.

NOTICE

TPM reflects more comprehensive performance than TPS. IOPS indicates the disk read/write capability in the current stress testing, which is for reference only.

Copyright © Huawei Technologies Co., Ltd.

Test Result

Table 4-1 Test results

DB Instance Type	Edition	vCPUs	Memory (GB)	TPM	TPS	IOPS
Primary/ Standby	2008 R2 Enterprise	2	8	300,000	5,500	4,000
	2008 R2 Enterprise	4	16	530,000	9,700	7,000
	2008 R2 Enterprise	8	32	930,000	17,050	15,000
	2008 R2 Enterprise	16	64	1,250,000	23,000	20,000
	2008 R2 Enterprise	2	16	290,000	5,300	4,000
	2008 R2 Enterprise	4	32	540,000	9,900	7,000
	2008 R2 Enterprise	8	64	960,000	17,600	15,000
	2008 R2 Enterprise	16	128	1350000	24750	20000
Single	2014 Enterprise and 2014 Standard	4	16	550,000	10,083	7,000
	2014 Enterprise and 2014 Standard	8	32	1,100,000	20,166	16,000
	2014 Enterprise and 2014 Standard	16	64	1,500,000	27,500	22,000
Primary/ Standby	2014 Enterprise and 2014 Standard	4	32	500,000	9,000	7,000
	2014 Enterprise and 2014 Standard	8	64	1,000,000	18,333	16,000

Copyright © Huawei Technologies Co., Ltd.

DB Instance Type	Edition	vCPUs	Memory (GB)	TPM	TPS	IOPS
	2014 Enterprise and 2014 Standard	16	128	1,400,000	24,000	21,000
Single	2014 Web	4	16	550,000	10,000	6,000
	2014 Web	8	32	1,100,000	20,000	12,000
	2014 Web	16	64	1,500,000	27,000	18,000

Copyright © Huawei Technologies Co., Ltd.