Relational Database Service

Performance White Paper

Issue 01
Date 2019-09-29
# 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

2 RDS for PostgreSQL.......................................................................................................................11
  2.1 Test Method.................................................................................................................................11
  2.2 PostgreSQL 9.6 Test Data...........................................................................................................14

3 RDS for SQL Server.........................................................................................................................17
  3.1 Test Method.................................................................................................................................17
  3.2 Test Results..................................................................................................................................26
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.

Huawei’s RDS MySQL databases supports immediate use, backup and recovery, 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 configuration while your data integrity and service continuity is guaranteed.

Test Environment

- Region: CN North-Beijing1
- AZ: AZ1
- Elastic Cloud Server (ECS): c3.2xlarge.2 flavor with 8 vCPUs, 16 GB of memory, and CentOS7.4 64 bit image. Bind an elastic IP address (EIP) to the ECS to access the Internet.
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.

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 the load 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 --num-threads=20 --max-time=3600 --max-requests=0 --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 --num-threads=20 --max-time=3600 --max-requests=0 --forced-shutdown=1 cleanup
```

-----End

Testing Models

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 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;
     ```

   - 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}
     ```
Test Metrics

- 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 (unit: GB) of the DB instance.

\[
\text{IOPS} = \min\{3500 + (\text{Instance storage space} - 40) \times 50, 33000\}
\]

For example, if the storage space of DB instance rds-test01 is 400 GB, IOPS is 21500 (or \(\min\{3500 + (400 - 40) \times 50, 33000\}=\min\{21500, 33000\}\)).

Test List of General-Enhanced Instances

<table>
<thead>
<tr>
<th>vCPUs</th>
<th>Memory (GB)</th>
<th>Connections (Workload Test Value)</th>
<th>TPS</th>
<th>QPS</th>
<th>IOPS</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2</td>
<td>800</td>
<td>197.19</td>
<td>3943.85</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>4</td>
<td>1500</td>
<td>470.4</td>
<td>9400.32</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>8</td>
<td>2500</td>
<td>768.23</td>
<td>15364.64</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>16</td>
<td>5000</td>
<td>1728.84</td>
<td>34576.84</td>
<td></td>
</tr>
<tr>
<td>16</td>
<td>32</td>
<td>10000</td>
<td>2947.42</td>
<td>58948.35</td>
<td></td>
</tr>
<tr>
<td>32</td>
<td>64</td>
<td>18000</td>
<td>3239.9</td>
<td>64798.05</td>
<td></td>
</tr>
</tbody>
</table>

The values in the Connections (Workload Test Value) column in the following tables are the result of the RDS performance stress test. For services running on the live network, set the parameter max_connections parameter.
### Table 1-2 vCPU:Memory=1:4

<table>
<thead>
<tr>
<th>vCPUs</th>
<th>Memory (GB)</th>
<th>Connections (Workload Test Value)</th>
<th>TPS</th>
<th>QPS</th>
<th>IOPS</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>4</td>
<td>1500</td>
<td>466.73</td>
<td>9334.62</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>8</td>
<td>2500</td>
<td>519.24</td>
<td>10384.74</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>16</td>
<td>5000</td>
<td>1191.98</td>
<td>23839.68</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>32</td>
<td>10000</td>
<td>2698.01</td>
<td>53960.27</td>
<td></td>
</tr>
<tr>
<td>16</td>
<td>64</td>
<td>18000</td>
<td>3148.77</td>
<td>62975.44</td>
<td></td>
</tr>
<tr>
<td>32</td>
<td>128</td>
<td>30000</td>
<td>4385.29</td>
<td>87705.81</td>
<td></td>
</tr>
<tr>
<td>60</td>
<td>256</td>
<td>60000</td>
<td>4810.94</td>
<td>95117.75</td>
<td></td>
</tr>
</tbody>
</table>

See [About IOPS](#).

### Table 1-3 vCPU:Memory=1:8

<table>
<thead>
<tr>
<th>vCPUs</th>
<th>Memory (GB)</th>
<th>Connections (Workload Test Value)</th>
<th>TPS</th>
<th>QPS</th>
<th>IOPS</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>8</td>
<td>2500</td>
<td>487.4</td>
<td>9748.07</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>16</td>
<td>5000</td>
<td>617.59</td>
<td>12351.83</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>32</td>
<td>10000</td>
<td>1374.01</td>
<td>27480.14</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>64</td>
<td>18000</td>
<td>2824.63</td>
<td>56492.64</td>
<td></td>
</tr>
<tr>
<td>16</td>
<td>128</td>
<td>30000</td>
<td>4215.09</td>
<td>84301.79</td>
<td></td>
</tr>
<tr>
<td>60</td>
<td>512</td>
<td>100000</td>
<td>4838.57</td>
<td>96771.34</td>
<td></td>
</tr>
</tbody>
</table>

See [About IOPS](#).
Test Result of the General-Enhanced Instances

**Figure 1-2** vCPU:Memory=1:2

![MySQL 5.6 vCPU:Memory=1:2](image)

**Figure 1-3** vCPU:Memory=1:4

![MySQL 5.6 vCPU:Memory=1:4](image)
1.3 MySQL 5.7 Test Data

About IOPS

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

\[
IOPS = \min\{3500 + (\text{Instance storage space} - 40) \times 50, 33000\}
\]

For example, if the storage space of DB instance `rds-test01` is 400 GB, IOPS is 21500 (or \(\min\{3500 + (400 - 40) \times 50, 33000\} = \min\{21500, 33000\}\)).

Test List of General-Enhanced Instances

<table>
<thead>
<tr>
<th>vCPUs</th>
<th>Memory (GB)</th>
<th>Connections (Workload Test Value)</th>
<th>TPS</th>
<th>QPS</th>
<th>IOPS</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2</td>
<td>800</td>
<td>236.59</td>
<td>4731.85</td>
<td>21500</td>
</tr>
<tr>
<td>2</td>
<td>4</td>
<td>1500</td>
<td>470.35</td>
<td>9400.04</td>
<td>95771.34</td>
</tr>
</tbody>
</table>

See About IOPS.
<table>
<thead>
<tr>
<th>vCPUs</th>
<th>Memory (GB)</th>
<th>Connections (Workload Test Value)</th>
<th>TPS</th>
<th>QPS</th>
<th>IOPS</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>8</td>
<td>2500</td>
<td>969.14</td>
<td>19382.91</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>16</td>
<td>5000</td>
<td>1768.01</td>
<td>35360.39</td>
<td></td>
</tr>
<tr>
<td>16</td>
<td>32</td>
<td>10000</td>
<td>2630.5</td>
<td>52000.99</td>
<td></td>
</tr>
<tr>
<td>32</td>
<td>64</td>
<td>18000</td>
<td>3011.89</td>
<td>60237.87</td>
<td></td>
</tr>
<tr>
<td>60</td>
<td>128</td>
<td>30000</td>
<td>4471.39</td>
<td>88428.09</td>
<td></td>
</tr>
</tbody>
</table>

**Table 1-5** vCPU:Memory=1:4

<table>
<thead>
<tr>
<th>vCPUs</th>
<th>Memory (GB)</th>
<th>Connections (Workload Test Value)</th>
<th>TPS</th>
<th>QPS</th>
<th>IOPS</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>4</td>
<td>1500</td>
<td>468.11</td>
<td>9362.23</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>8</td>
<td>2500</td>
<td>620.7</td>
<td>12393.97</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>16</td>
<td>5000</td>
<td>1230.39</td>
<td>24607.81</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>32</td>
<td>10000</td>
<td>2514.48</td>
<td>50289.62</td>
<td></td>
</tr>
<tr>
<td>16</td>
<td>64</td>
<td>18000</td>
<td>3016.84</td>
<td>60336.77</td>
<td></td>
</tr>
<tr>
<td>32</td>
<td>128</td>
<td>30000</td>
<td>4367.67</td>
<td>87353.69</td>
<td></td>
</tr>
<tr>
<td>60</td>
<td>256</td>
<td>60000</td>
<td>4536.41</td>
<td>90728.55</td>
<td></td>
</tr>
</tbody>
</table>

**Table 1-6** vCPU:Memory=1:8

<table>
<thead>
<tr>
<th>vCPUs</th>
<th>Memory (GB)</th>
<th>Connections (Workload Test Value)</th>
<th>TPS</th>
<th>QPS</th>
<th>IOPS</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>8</td>
<td>2500</td>
<td>612.75</td>
<td>12255.01</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>16</td>
<td>5000</td>
<td>675.66</td>
<td>13513.18</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>32</td>
<td>10000</td>
<td>1488.25</td>
<td>29765.06</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>64</td>
<td>18000</td>
<td>2810.79</td>
<td>56215.81</td>
<td></td>
</tr>
<tr>
<td>16</td>
<td>128</td>
<td>30000</td>
<td>4095.49</td>
<td>81909.79</td>
<td></td>
</tr>
<tr>
<td>60</td>
<td>512</td>
<td>100000</td>
<td>4626.18</td>
<td>96823.95</td>
<td></td>
</tr>
</tbody>
</table>

See About IOPS.
Test Result of the General-Enhanced Instances

Figure 1-5 vCPU:Memory=1:2

![MySQL 5.7 vCPU:Memory=1:2](image1)

Figure 1-6 vCPU:Memory=1:4

![MySQL 5.7 vCPU:Memory=1:4](image2)
Figure 1-7 vCPU:Memory=1:8

MySQL 5.7 vCPU:Memory=1:8

<table>
<thead>
<tr>
<th>Memory Size</th>
<th>TPS</th>
<th>QPS</th>
</tr>
</thead>
<tbody>
<tr>
<td>1U8G</td>
<td>612.75</td>
<td>675.66</td>
</tr>
<tr>
<td>2U16G</td>
<td>1225.01</td>
<td>1351.18</td>
</tr>
<tr>
<td>4U32G</td>
<td>2976.06</td>
<td>2988.25</td>
</tr>
<tr>
<td>8U64G</td>
<td>5621.81</td>
<td>5610.79</td>
</tr>
<tr>
<td>16U128G</td>
<td>8190.79</td>
<td>4095.49</td>
</tr>
<tr>
<td>60U512G</td>
<td>9682.95</td>
<td>4626.13</td>
</tr>
</tbody>
</table>
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.

- The **postgis** plugin is supported. Its space application is excellent and meets the international standards. PostgreSQL offers functions similar to those of Oracle database systems, but is less expensive.
- 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
- **Elastic Cloud Server (ECS)**: c3.2xlarge.2 flavor with 8 vCPUs, 16 GB of memory, and CentOS7.4 64 bit image. Bind an elastic IP address (EIP) to the ECS to access the Internet.
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:

```bash
#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 command to log in to a database and create the test database loadtest:

```bash
psql -h<host> -p5432 "dbname=postgres user=root password=<password>" <<TEST
create database loadtest;
TEST
```
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 the load 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 Models

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;
     ```
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 Metrics

- 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.

2.2 PostgreSQL 9.6 Test Data

NOTICE

The values in the Connections (Workload Test Value) column in the following tables are the result of the RDS performance stress test. For services running on the live network, set the parameter max_connections parameter.

Test List of General-Enhanced Instances

Table 2-1 vCPU:Memory=1:2

<table>
<thead>
<tr>
<th>vCPUs</th>
<th>Memory (GB)</th>
<th>Connections (Workload Test Value)</th>
<th>TPS</th>
<th>QPS</th>
<th>IOPS</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2</td>
<td>300</td>
<td>150</td>
<td>3086</td>
<td>1200</td>
</tr>
<tr>
<td>2</td>
<td>4</td>
<td>500</td>
<td>263</td>
<td>5257</td>
<td>2000</td>
</tr>
<tr>
<td>4</td>
<td>8</td>
<td>1200</td>
<td>578</td>
<td>11554</td>
<td>6000</td>
</tr>
<tr>
<td>60</td>
<td>128</td>
<td>12000</td>
<td>1398</td>
<td>27965</td>
<td>20000</td>
</tr>
</tbody>
</table>

Table 2-2 CPU:Memory=1:4

<table>
<thead>
<tr>
<th>vCPUs</th>
<th>Memory (GB)</th>
<th>Connections (Workload Test Value)</th>
<th>TPS</th>
<th>QPS</th>
<th>IOPS</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>4</td>
<td>500</td>
<td>166</td>
<td>3312</td>
<td>1400</td>
</tr>
<tr>
<td>2</td>
<td>8</td>
<td>1200</td>
<td>517</td>
<td>10340</td>
<td>4000</td>
</tr>
<tr>
<td>4</td>
<td>16</td>
<td>2500</td>
<td>705</td>
<td>14093</td>
<td>8000</td>
</tr>
<tr>
<td>8</td>
<td>32</td>
<td>5000</td>
<td>821</td>
<td>16421</td>
<td>12000</td>
</tr>
<tr>
<td>16</td>
<td>64</td>
<td>10000</td>
<td>880</td>
<td>17590</td>
<td>16000</td>
</tr>
<tr>
<td>32</td>
<td>128</td>
<td>12000</td>
<td>1392</td>
<td>27832</td>
<td>20000</td>
</tr>
<tr>
<td>60</td>
<td>256</td>
<td>20000</td>
<td>2017</td>
<td>40336</td>
<td>20000</td>
</tr>
</tbody>
</table>

Table 2-3 CPU:Memory=1:8

<table>
<thead>
<tr>
<th>vCPUs</th>
<th>Memory (GB)</th>
<th>Connections (Workload Test Value)</th>
<th>TPS</th>
<th>QPS</th>
<th>IOPS</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>16</td>
<td>2500</td>
<td>554</td>
<td>11099</td>
<td>5000</td>
</tr>
<tr>
<td>4</td>
<td>32</td>
<td>5000</td>
<td>781</td>
<td>15614</td>
<td>7000</td>
</tr>
<tr>
<td>8</td>
<td>64</td>
<td>5000</td>
<td>931</td>
<td>18613</td>
<td>11000</td>
</tr>
</tbody>
</table>

Test Result of the General-Enhanced Instances

Figure 2-2 General-purpose
**Figure 2-3** Compute

![Compute Chart]

**Figure 2-4** Memory

![Memory Chart]
3.1 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 modes: yearly/monthly and pay-per-use.

Test Environment

- Region: CN North-Beijing1
- AZ: AZ1
- Elastic Cloud Server (ECS) specifications: hc2 flavor with 8 vCPUs, 16 GB of memory, SSD disk, 200 GB storage, and Windows Server 2012 R2 Standard 64bit image. The network type is VPC.

Test Tool

HammerDB is a graphical open source database load 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
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 an OLTP benchmark. It is different and more complex than TPC-A because of its multiple transaction types, more complex database, 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  Open HammerDB.
Figure 3-1 HammerDB started

![HammerDB interface]

**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 3-2 Benchmark Options

![Benchmark Options dialog box]
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 3-3** 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.
Figure 3-4 Build

![Figure 3-4 Build](image1)

Wait until the initialization is complete.

Figure 3-5 Initialization completed

![Figure 3-5 Initialization completed](image2)
Step 5  Click 🔄 to stop the execution.

Figure 3-6 Execution stopped

Step 6  Select SQL Server > TPC-C > Driver Script and double-click Options to ensure that the connection information is correct.
Figure 3-7 Checking the connection information

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

Figure 3-8 Load
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 TPM values become consistent. **Transactions Per Minute (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**.
Step 10  Choose Virtual User and double-click Run until the TPM peak value is stable.
3.2 Test Results

Concepts

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

Transactions per second (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 pressure test, instead of the maximum IOPS capability.

NOTICE

TPM reflects more comprehensive performance than TPS. IOPS indicates the disk read/write capability in the current pressure test, which is for reference only.
## Test Data

### Table 3-1 Test list

<table>
<thead>
<tr>
<th>DB Instance Type</th>
<th>Edition</th>
<th>vCPUs</th>
<th>Memory (GB)</th>
<th>TPM</th>
<th>TPS</th>
<th>IOPS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Primary/Standby</td>
<td>2008 R2 Enterprise</td>
<td>2</td>
<td>8</td>
<td>300000</td>
<td>5500</td>
<td>4000</td>
</tr>
<tr>
<td></td>
<td>2008 R2 Enterprise</td>
<td>4</td>
<td>16</td>
<td>530000</td>
<td>9700</td>
<td>7000</td>
</tr>
<tr>
<td></td>
<td>2008 R2 Enterprise</td>
<td>8</td>
<td>32</td>
<td>930000</td>
<td>17050</td>
<td>15000</td>
</tr>
<tr>
<td></td>
<td>2008 R2 Enterprise</td>
<td>16</td>
<td>64</td>
<td>1250000</td>
<td>23000</td>
<td>20000</td>
</tr>
<tr>
<td></td>
<td>2008 R2 Enterprise</td>
<td>2</td>
<td>16</td>
<td>290000</td>
<td>5300</td>
<td>4000</td>
</tr>
<tr>
<td></td>
<td>2008 R2 Enterprise</td>
<td>4</td>
<td>32</td>
<td>540000</td>
<td>9900</td>
<td>7000</td>
</tr>
<tr>
<td></td>
<td>2008 R2 Enterprise</td>
<td>8</td>
<td>64</td>
<td>960000</td>
<td>17600</td>
<td>15000</td>
</tr>
<tr>
<td></td>
<td>2008 R2 Enterprise</td>
<td>16</td>
<td>128</td>
<td>1350000</td>
<td>24750</td>
<td>20000</td>
</tr>
<tr>
<td>Single</td>
<td>2014 Enterprise and 2014 Standard</td>
<td>4</td>
<td>16</td>
<td>550000</td>
<td>10083</td>
<td>7000</td>
</tr>
<tr>
<td></td>
<td>2014 Enterprise and 2014 Standard</td>
<td>8</td>
<td>32</td>
<td>1100000</td>
<td>20166</td>
<td>16000</td>
</tr>
<tr>
<td></td>
<td>2014 Enterprise and 2014 Standard</td>
<td>16</td>
<td>64</td>
<td>1500000</td>
<td>27500</td>
<td>22000</td>
</tr>
<tr>
<td>Primary/Standby</td>
<td>2014 Enterprise and 2014 Standard</td>
<td>4</td>
<td>32</td>
<td>500000</td>
<td>9000</td>
<td>7000</td>
</tr>
<tr>
<td></td>
<td>2014 Enterprise and 2014 Standard</td>
<td>8</td>
<td>64</td>
<td>1000000</td>
<td>18333</td>
<td>16000</td>
</tr>
<tr>
<td>DB Instance Type</td>
<td>Edition</td>
<td>vCPUs</td>
<td>Memory (GB)</td>
<td>TPM</td>
<td>TPS</td>
<td>IOPS</td>
</tr>
<tr>
<td>------------------</td>
<td>---------</td>
<td>-------</td>
<td>-------------</td>
<td>-----</td>
<td>-----</td>
<td>------</td>
</tr>
<tr>
<td>2014 Enterprise and 2014 Standard</td>
<td>16</td>
<td>128</td>
<td>1400000</td>
<td>24000</td>
<td>21000</td>
<td></td>
</tr>
<tr>
<td>Single</td>
<td>2014 Web</td>
<td>4</td>
<td>16</td>
<td>550000</td>
<td>10000</td>
<td>6000</td>
</tr>
<tr>
<td></td>
<td>2014 Web</td>
<td>8</td>
<td>32</td>
<td>1100000</td>
<td>20000</td>
<td>12000</td>
</tr>
<tr>
<td></td>
<td>2014 Web</td>
<td>16</td>
<td>64</td>
<td>1500000</td>
<td>27000</td>
<td>18000</td>
</tr>
</tbody>
</table>