Amazon Aurora Performance Assessment

This document outlines the steps to assess the performance of Amazon Relational Database Service (Amazon RDS) for Amazon Aurora using the Sysbench benchmarking tool and discusses use of the Amazon Machine Image (AMI) customized to make it easy to benchmark the performance of Amazon Aurora with the Sysbench benchmark and Transaction Processing Performance Council Benchmark C (TPC-C).

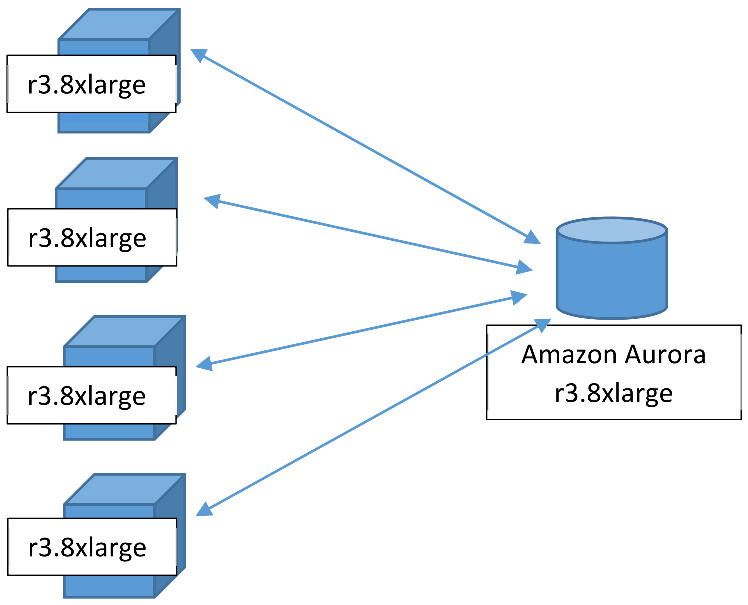
Assessing Amazon RDS for Aurora Performance Using Sysbench

This section outlines the steps to assess the performance of Amazon RDS for Aurora using the Sysbench tool. It describes running two different workloads to simulate a simple read-heavy workload (100 percent read), and a write-heavy workload (100 percent write) on Amazon RDS for Aurora.

The results displayed in this document are representative. The results of your testing might vary based on various environmental dependencies including instance configuration, network performance, and so on.

Setup

Our setup consists of four Amazon Elastic Compute Cloud (Amazon EC2) r3.8xlarge Linux instances running the Sysbench tool and querying an Amazon Aurora DB instance, also r3.8xlarge. All instances are created in an Amazon Virtual Private Cloud (Amazon VPC) with enhanced networking enabled to ensure that throughput numbers are not constrained by network bandwidth.



For this setup, perform the following steps:

1. Create a VPC. (Note that for the purposes of this test, all four EC2 r3.8xl instances running Sysbench were in same Availability Zone as the database instance.)
2. Ensure enhanced networking is enabled on all four r3.8xlarge instances. For more information, see http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/enhanced-networking.html.

3. Install Sysbench version 0.5 on r3.8xlarge instances. For detailed instructions, see Installing Sysbench Version 0.5 at the end of this section.
4. Apply the following network settings to the Sysbench client. These settings tell the Linux kernel to use all CPU cores to process packets, instead of the default of two, and to reduce context switching between cores. Both settings are intended to help drive throughput without the need for additional Sysbench clients.

   ```
   sudo sh -c 'for x in /sys/class/net/eth0/queues/rx-*; do echo ffffffff
   > $x/rps_cpus; done'
   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"
   ```

5. Launch a r3.8xlarge Amazon Aurora DB instance in the same VPC. When you do this, Amazon RDS automatically enables enhanced networking for you.

Steps for Performance Testing

To do the performance test, follow these steps:

1. Prepare the Amazon Aurora database by running the following command on any one of the Amazon EC2 r3.8xlarge instances. This command will create the test database.

   ```
   ./sysbench --test=tests/db/oltp.lua --mysql-host= <rds-aurora-instance-
   host-name> --mysql-port=3306 --mysql-user=<db-username> --mysql-
   password=<db-password> --mysql-db=<db-name> --mysql-table-engine=innodb --
   oltp-table-size=25000 --oltp-tables-count=250 --db-driver=mysql  prepare
   ```

2. To perform the simple read-heavy test, run the following commands on each of the four Sysbench clients in parallel.

   ```
   ./sysbench --test=tests/db/oltp.lua --mysql-host=<rds-aurora-instance-
   host-name> --oltp-tables-count=250 --mysql-user=<db-username> --mysql-
   password=<db-password> --mysql-port=3306 --db-driver=mysql --oltp-table-
   size=25000 --mysql-db=<db-name> --max-requests=0 --oltp_simple_ranges=0 --
   oltp-distinct-ranges=0 --oltp-sum-ranges=0 --oltp-order-ranges=0 --max-
   time=600 --oltp-read-only=on --num-threads=500 run
   ```

3. To perform the write-heavy test, run the following commands on each of the four Sysbench clients in parallel.

   ```
   ./sysbench --test=tests/db/oltp.lua --mysql-host=<rds-aurora-instance-
   host-name> --oltp-tables-count=250 --mysql-user=<db-username> --mysql-
   password=<db-password> --mysql-port=3306 --db-driver=mysql --oltp-table-
   size=25000 --mysql-db=<db-name> --max-requests=0 --max-time=600 --
   oltp_simple_ranges=0 --oltp-distinct-ranges=0 --oltp-sum-ranges=0 --oltp-
   order-ranges=0 --oltp-point-selects=0 --num-threads=1000 --rand-
   type=uniform run
   ```

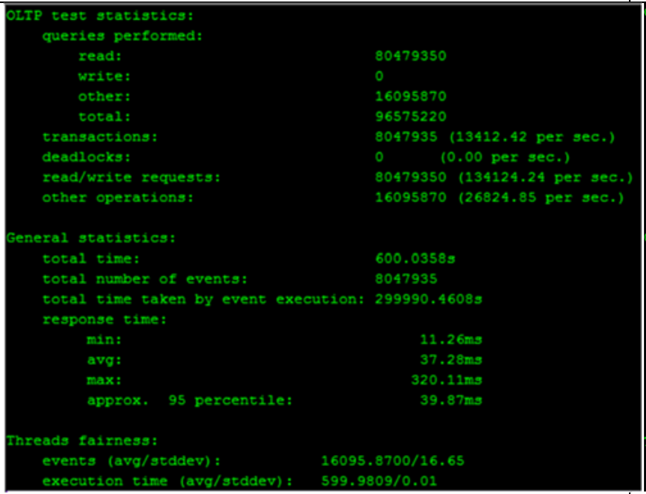
Results

The results of the test should be as follows.

Simple Read-Heavy (or Select) Workload

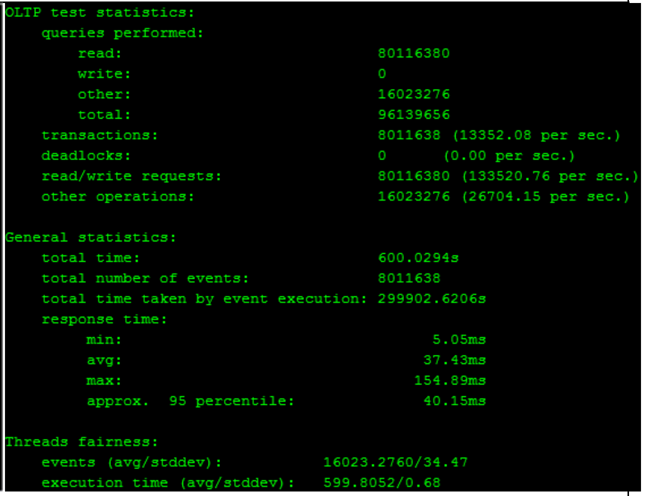
The following Sysbench output displays the results of our read tests when performed simultaneously on all four Amazon EC2 client machines running Sysbench. We ran this test for 600 seconds and observed a read performance of 535,164 read requests per second (the sum of the read/write requests per second for the four Amazon EC2 Sysbench client machines).

EC2 Sysbench client machine 1



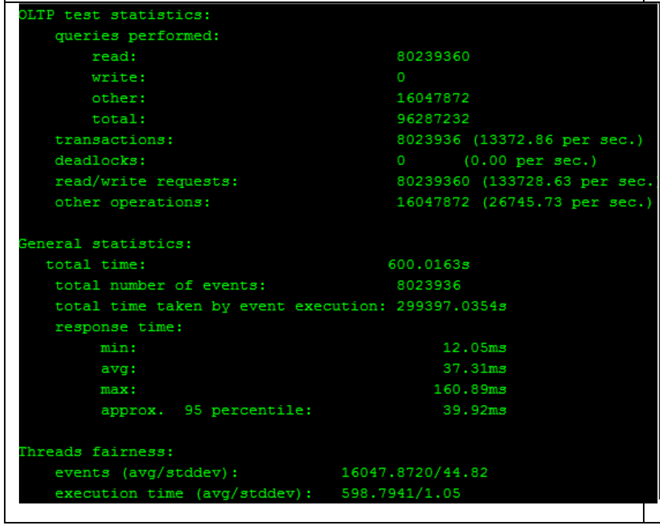
```
OLTP test statistics:
    queries performed:
        read:                            80479350
        write:                           0
        other:                           16095870
        total:                           96575220
    transactions:                        8047935 (13412.42 per sec.)
    deadlocks:                           0       (0.00 per sec.)
    read/write requests:                 80479350 (134124.24 per sec.)
    other operations:                    16095870 (26824.85 per sec.)

General statistics:
    total time:                          600.0358s
    total number of events:              8047935
    total time taken by event execution: 299990.4608s
    response time:
         min:                                 11.26ms
         avg:                                 37.28ms
         max:                                320.11ms
         approx.  95 percentile:              39.87ms

Threads fairness:
    events (avg/stddev):           16095.8700/16.65
    execution time (avg/stddev):   599.9809/0.01
```

EC2 Sysbench client machine 2



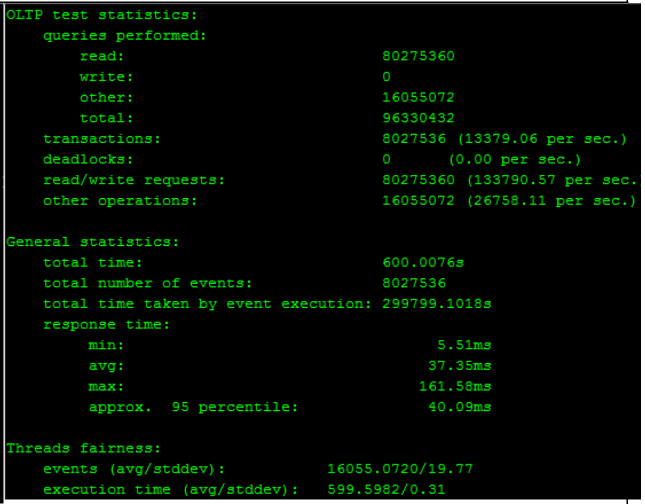
```
OLTP test statistics:
    queries performed:
        read:                            80116380
        write:                           0
        other:                           16023276
        total:                           96139656
    transactions:                        8011638 (13352.08 per sec.)
    deadlocks:                           0       (0.00 per sec.)
    read/write requests:                 80116380 (133520.76 per sec.)
    other operations:                    16023276 (26704.15 per sec.)

General statistics:
    total time:                          600.0294s
    total number of events:              8011638
    total time taken by event execution: 299902.6206s
    response time:
         min:                                  5.05ms
         avg:                                 37.43ms
         max:                                154.89ms
         approx.  95 percentile:              40.15ms

Threads fairness:
    events (avg/stddev):           16023.2760/34.47
    execution time (avg/stddev):   599.8052/0.68
```

EC2 Sysbench client machine 3



```
OLTP test statistics:
    queries performed:
        read:                            80239360
        write:                           0
        other:                           16047872
        total:                           96287232
    transactions:                        8023936 (13372.86 per sec.)
    deadlocks:                           0       (0.00 per sec.)
    read/write requests:                 80239360 (133728.63 per sec.
    other operations:                    16047872 (26745.73 per sec.)

General statistics:
   total time:                          600.0163s
    total number of events:              8023936
    total time taken by event execution: 299397.0354s
    response time:
         min:                                 12.05ms
         avg:                                 37.31ms
         max:                                160.89ms
         approx.  95 percentile:              39.92ms

Threads fairness:
    events (avg/stddev):           16047.8720/44.82
    execution time (avg/stddev):   598.7941/1.05
```

EC2 Sysbench client machine 4

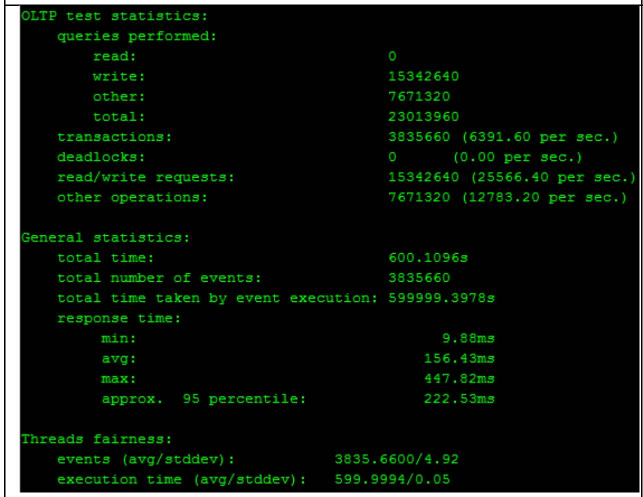


```
OLTP test statistics:
    queries performed:
        read:                            80275360
        write:                           0
        other:                           16055072
        total:                           96330432
    transactions:                        8027536 (13379.06 per sec.)
    deadlocks:                           0       (0.00 per sec.)
    read/write requests:                 80275360 (133790.57 per sec.
    other operations:                    16055072 (26758.11 per sec.)

General statistics:
    total time:                          600.0076s
    total number of events:              8027536
    total time taken by event execution: 299799.1018s
    response time:
         min:                                  5.51ms
         avg:                                 37.35ms
         max:                                161.58ms
         approx.  95 percentile:              40.09ms

Threads fairness:
    events (avg/stddev):           16055.0720/19.77
    execution time (avg/stddev):   599.5982/0.31
```

Write-Heavy Workload

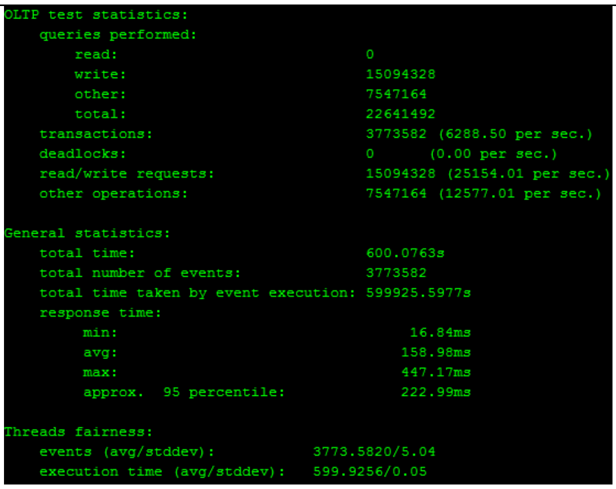
The following Sysbench output displays the results of our write-heavy tests when performed simultaneously on all four Amazon EC2 client machines running Sysbench. We ran this test for 600 seconds and observed a write performance of 101,386 write requests per second (the sum of the read/write requests per second for the four Amazon EC2 Sysbench client machines), while persisting several copies of data in three different data centers. During this test, we performed 25,329 transactions per second (the sum of the transactions per second for the four Amazon EC2 Sysbench client machines), involving inserts, index updates, non-index updates, and deletes.

EC2 Sysbench client machine 1



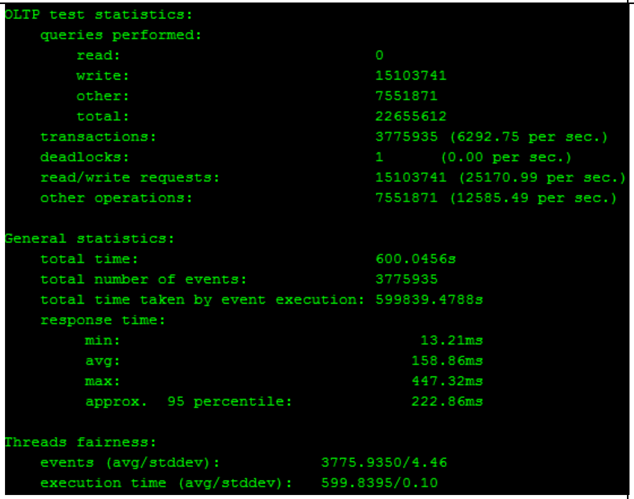
```
OLTP test statistics:
    queries performed:
        read:                            0
        write:                           15342640
        other:                           7671320
        total:                           23013960
    transactions:                        3835660 (6391.60 per sec.)
    deadlocks:                           0      (0.00 per sec.)
    read/write requests:                 15342640 (25566.40 per sec.)
    other operations:                    7671320 (12783.20 per sec.)

General statistics:
    total time:                          600.1096s
    total number of events:              3835660
    total time taken by event execution: 599999.3978s
    response time:
         min:                                  9.88ms
         avg:                                156.43ms
         max:                                447.82ms
         approx.  95 percentile:             222.53ms

Threads fairness:
    events (avg/stddev):           3835.6600/4.92
    execution time (avg/stddev):   599.9994/0.05
```

EC2 Sysbench client machine 2



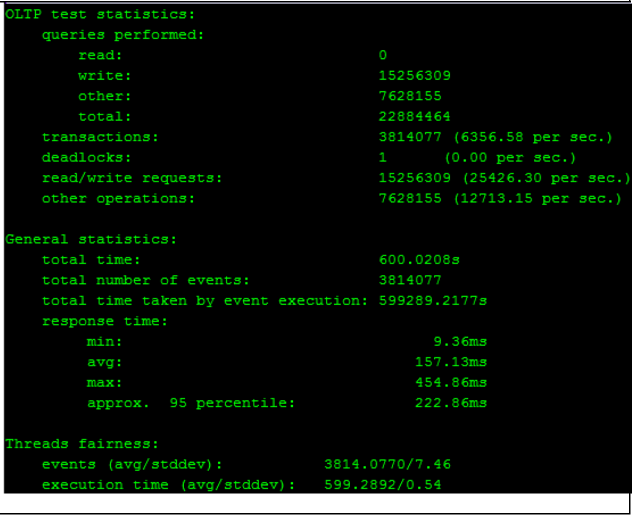
```
OLTP test statistics:
    queries performed:
        read:                            0
        write:                           15094328
        other:                           7547164
        total:                           22641492
    transactions:                        3773582 (6288.50 per sec.)
    deadlocks:                           0      (0.00 per sec.)
    read/write requests:                 15094328 (25154.01 per sec.)
    other operations:                    7547164 (12577.01 per sec.)

General statistics:
    total time:                          600.0763s
    total number of events:              3773582
    total time taken by event execution: 599925.5977s
    response time:
         min:                                 16.84ms
         avg:                                158.98ms
         max:                                447.17ms
         approx.  95 percentile:             222.99ms

Threads fairness:
    events (avg/stddev):           3773.5820/5.04
    execution time (avg/stddev):   599.9256/0.05
```

EC2 Sysbench client machine 3



```
OLTP test statistics:
    queries performed:
        read:                            0
        write:                           15103741
        other:                           7551871
        total:                           22655612
    transactions:                        3775935 (6292.75 per sec.)
    deadlocks:                           1      (0.00 per sec.)
    read/write requests:                 15103741 (25170.99 per sec.)
    other operations:                    7551871 (12585.49 per sec.)

General statistics:
    total time:                          600.0456s
    total number of events:              3775935
    total time taken by event execution: 599839.4788s
    response time:
         min:                                 13.21ms
         avg:                                158.86ms
         max:                                447.32ms
         approx.  95 percentile:             222.86ms

Threads fairness:
    events (avg/stddev):           3775.9350/4.46
    execution time (avg/stddev):   599.8395/0.10
```

EC2 Sysbench client machine 4



```
OLTP test statistics:
    queries performed:
        read:                            0
        write:                           15256309
        other:                           7628155
        total:                           22884464
    transactions:                        3814077 (6356.58 per sec.)
    deadlocks:                           1      (0.00 per sec.)
    read/write requests:                 15256309 (25426.30 per sec.)
    other operations:                    7628155 (12713.15 per sec.)

General statistics:
    total time:                          600.0208s
    total number of events:              3814077
    total time taken by event execution: 599289.2177s
    response time:
         min:                                  9.36ms
         avg:                                157.13ms
         max:                                454.86ms
         approx.  95 percentile:             222.86ms

Threads fairness:
    events (avg/stddev):           3814.0770/7.46
    execution time (avg/stddev):   599.2892/0.54
```

Installing Sysbench Version 0.5

To install Sysbench version 0.5, perform the following steps:

1. Install Bazaar using this command:
```
$ yum -y install bzr
```
2. Install Automake using this command:
```
$ yum -y install automake
```
3. Install Libtool using this command:
```
$ yum -y install libtool
```
4. Install MySQL client-side headers and libraries (mysql-devel for Red Hat, or libmysqlclient-dev for Debian/Ubuntu) using this command:
```
$ yum -y install mysql-devel
```
5. Download Sysbench using this command:
```
$ bzr branch lp:sysbench
```
6. Compile and install Sysbench using these commands:
```
$ cd sysbench
$ ./autogen.sh
$ ./configure
$ make
$ cd sysbench
```

Using the Aurora Benchmarking AMI

AWS has prepared an Amazon Machine Image (AMI) customized to make it easy to benchmark the performance of Amazon Aurora. Customers can launch an instance of the Aurora Benchmarking AMI in their own accounts and use the host name, user name, and password of their instance of Amazon Aurora to run any or all of the Sysbench or TPC-C tests included in the AMI.

Sysbench is the standard open-source benchmark discussed in the previous section, for which MySQL provides a tool here.

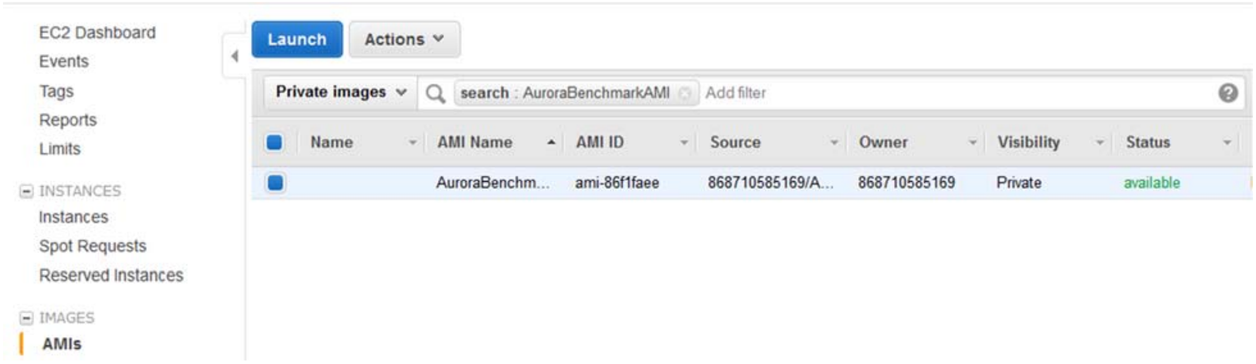
Transaction Processing Performance Council Benchmark C (TPC-C) is another standard benchmark, for which a test tool is available from Percona here. For the TPC-C test, we create tables and foreign keys and run the tpcc100.sql file to simulate a 100 warehouse data dump.

Launching the Aurora Benchmarking AMI

The Aurora Benchmarking AMI is available in three regions:

- US East (N. Virginia): ami-afe00ac4
- US West (Oregon): ami-0b6b6d3b
- EU (Ireland) : ami-f45d1983

To launch an instance of the benchmarking AMI, sign in to the AWS Management Console and select the Amazon EC2 service. In the left pane, choose **AMI**. At the top, type the search string AuroraBenchmarkAMI_2.0. Select the image you want, and choose **Launch** to launch the instance.



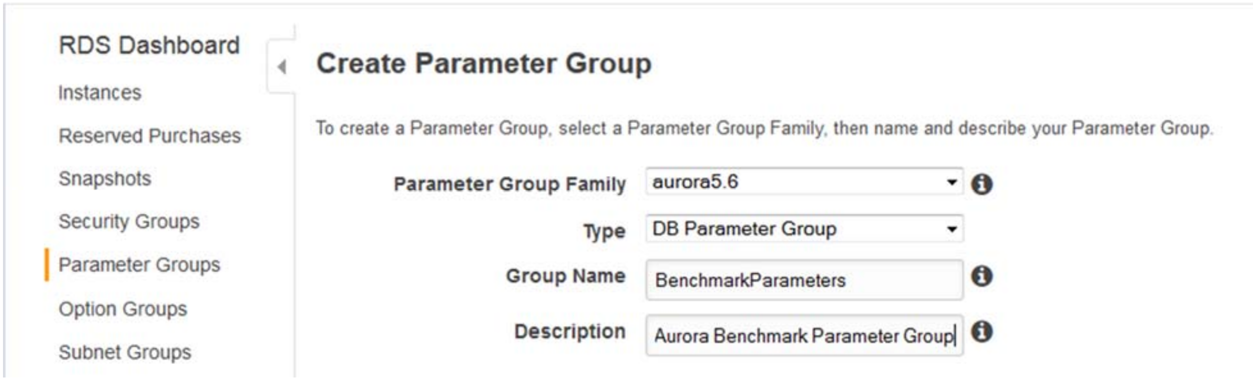
Name	AMI Name	AMI ID	Source	Owner	Visibility	Status
	AuroraBenchm...	ami-86f1faee	868710585169/A...	868710585169	Private	available

The Aurora Benchmarking AMI connects to an Aurora database on port 3306. An existing Aurora database running on port 3306 will work fine. Alternatively, create a new database using the instructions in Getting Started with Amazon Aurora in the *Amazon Relational Database Service User Guide.* Record the master user name, master password, and host name to use as parameters when you start the benchmark tests.

Note that in the command line to invoke the benchmarks, the database name should be a new database for the test, not the main database created with Aurora. For example, use **sysbench_test** not **mydb**.

Modifying the Parameter Group

To run the TPC-C test, you need to adjust parameters. To do so, in the Aurora Management Console create a new parameter group.

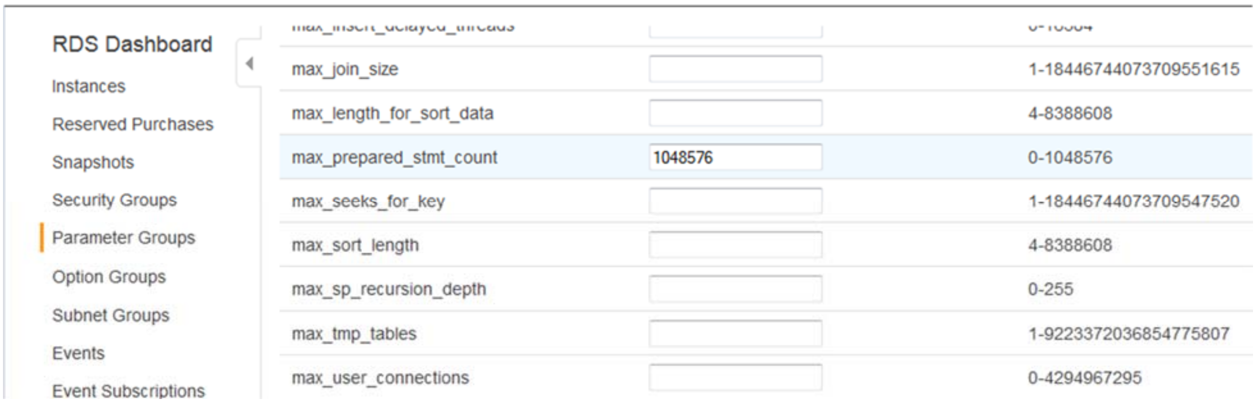


Create Parameter Group

To create a Parameter Group, select a Parameter Group Family, then name and describe your Parameter Group.

Parameter Group Family: aurora5.6
Type: DB Parameter Group
Group Name: BenchmarkParameters
Description: Aurora Benchmark Parameter Group

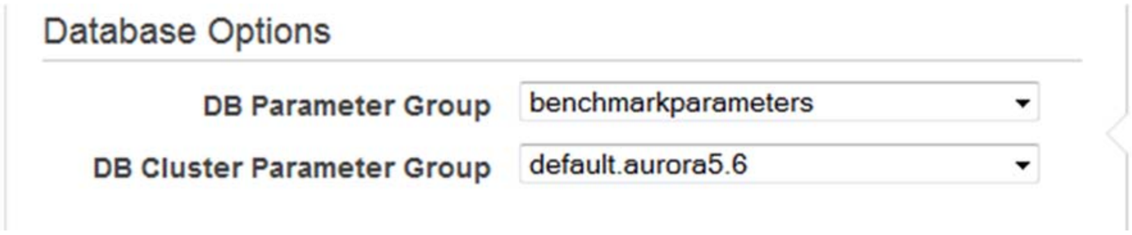
Once you have done so, in the list of parameter groups select the new **benchmarkparameters** group and choose **Edit Parameters**. Scroll to the **max_prepared_stmt_count** parameter and set the value to `1048576`. The TPC-C test simulates a warehouse workload, and we set this parameter very high to allow for a large volume of prepared statements without running into a bottleneck with the configuration.



max_join_size		1-18446744073709551615
max_length_for_sort_data		4-8388608
max_prepared_stmt_count	1048576	0-1048576
max_seeks_for_key		1-18446744073709547520
max_sort_length		4-8388608
max_sp_recursion_depth		0-255
max_tmp_tables		1-9223372036854775807
max_user_connections		0-4294967295

Scroll to the top of the page and choose **Save Changes**.

Next, choose **Instances** in the left pane, select your database and in the **Instance Actions** list choose Modify. For **DB Parameter Group**, choose **benchmarkparameters**, and then choose **Apply Immediately**.



DB Parameter Group: benchmarkparameters

DB Cluster Parameter Group: default.aurora5.6

Choose **Continue**, and then validate your change. When you are through, choose **Modify DB Instance**, and then restart the instance.

Running the Tests

Running the Sysbench Benchmark

Run the following command to perform the Sysbench benchmark test:

```
./db_benchmarks -test sysbench  -engine db_engine -host host_name  -u user -D
db_name -p password -type instance_type -t time
```

The test should take at least 20 seconds. Results are reported to the command line.

Running the TPC-C Benchmark

Run the following command to perform the TPC-C benchmark test:

```
. /db_benchmarks --test tpcc --host host_name -u user  -p password  -D
db_name
```

Results are reported to the command line.