MySQL slap test and expectations

Bobby Brown - 2014-10-16 - 0 Comments - in Frequently Asked Questions

What is MySQLSlap?

It is a diagnostic program designed to emulate client load for a MySQL server and to report the timing of each stage. It works as if multiple clients are accessing the

server. mysqlslap is available as of MySQL 5.1.4.

This article is not an endorsement of any particular tool, but does describe typical testing

scenarios which demonstrate a reasonable testing platform.

How does MySQLslap works ?

MySQLslap allows you to fire set of sql queries to targetted database at defined frequency,

defined concurrency, for defined no of iterations, with defined no of queries per client etc.

There are quiet a few options we can mention while using MySQLslap.

Check the following URL for more information:

http://dev.mysql.com/doc/refman/5.1/en/mysqlslap.html

Option --auto-generate-sql

The --auto-generate-sql switch tells mysqlslap to automatically generate and execute SQL

statements, monitor how fast MySQL performs this task, and display the result.

The --auto-generate-sql switch creates a schema "mysqlslap", creates a table, executes an INSERT query and saves dummy data to it, executes a SELECT query to retrieve the dummy

data, and then drops the table and schema.

You can refer to following writeup on stress test MySQL with mysqlslap.

http://www.techrepublic.com/blog/how-do-i/how-do-i-stress-test-mysql-with-mysqlslap/

A typical test can be carried out by executing following command.

[Note: Adjust mysql variable "max_connections" according to the load you are going to generate with mysqlslap. It should be greater than the value of the --concurrency

parameter you are going to set.]

```
# mysqlslap --host=10.0.65.70 --port=3306 --user=root -p --auto-generate-sql --
```

concurrency=300 --number-of-queries=10000 -vv

Benchmark

Average number of seconds to run all queries: 5.268 seconds

Minimum number of seconds to run all queries: 5.268 seconds

Maximum number of seconds to run all queries: 5.268 seconds

Number of clients running queries: 300

Average number of queries per client: 33

Mysqlslap with ScaleArc

Issues with mysqlslap's auto-generate-sql

Let us pass the same load to same database server through ScaleArc.

Only change we have to make in command is to use cluster IP and port instead of host

address and port (if non-default port has been used)

```
# mysqlslap --host=10.0.65.5 --port=3306 --user=root -p --auto-generate-sql --
```

concurrency=300 --number-of-queries=10000 -vv

Now that you have seen how your database has perform, its time to see what difference

ScaleArc would make with its feature.

Connecting to ScaleArc.

Use --host option to pointing mysqlslap to connect databases through ScaleArc.

```
# mysqlslap --host=<cluster IP> --port=<cluster port> --user=<database user> -p  --create-schema=<logical
```

database name> --query="<sql query>" --concurrency=100 --iterations=5 --number-of-queries=10000 -vv

Here is an example with the output.

```
# mysqlslap --host=10.0.65.5 --port=3306 --user=root -p  --create-schema=cms --query="select * from table1;"  --
```

concurrency=100 --iterations=5 --number-of-queries=10000 -vv

Parsing engines to use.

Enter password:

Starting Concurrency Test

Generating primary key list

Generating primary key list

Generating primary key list

Generating primary key list

Generating primary key list

Generating stats

Benchmark

Average number of seconds to run all queries: 6.785 seconds

Minimum number of seconds to run all queries: 6.264 seconds

Maximum number of seconds to run all queries: 7.413 seconds

Number of clients running queries: 100

Average number of queries per client: 100

After enabling caching for select pattern.

```
# mysqlslap --host 10.0.65.5 --port=3308 --user=root -p  --create-schema=cms --
query="select * from table1;"  --concurrency=100 --iterations=5 --number-of-
```

queries=10000 -vv

Parsing engines to use.

Enter password:

Starting Concurrency Test

Generating primary key list

Generating primary key list

Generating primary key list

Generating primary key list

Generating primary key list

Generating stats

Benchmark

Average number of seconds to run all queries: 2.322 seconds

Minimum number of seconds to run all queries: 1.993 seconds

Maximum number of seconds to run all queries: 2.732 seconds

Number of clients running queries: 100

Average number of queries per client: 100

Advantage of ScaleArc caching is clearly visible with following comparison.

	Without Cache	With Cache
Average number of seconds to run all queries:	6.785 seconds	2.322 seconds
Minimum number of seconds to run all queries	6.264 seconds	1.993 seconds
Maximum number of seconds to run all queries	7.413 seconds	2.732 seconds

Permalink:
https://support.scalearc.com/kb/articles/114