

Document Description	Stress Test Report using sysbench	Date	2023.04.12
Document Filename	StressTestReport.pdf	Revision#	1.0
Document Author	House Hsu		
Author Email	house40105@gmail.com		

//To run a stress test on a MySQL database using sysbench

//Install MySQL and sysbench on system if it is not already installed

//if not

```
root@localhost ~]# rpm -qa | grep mysql
```

```
[root@localhost ~]# sudo yum install wget
```

```
[root@localhost ~]# wget -P /temp/ https://dev.mysql.com/get/mysql80-community-release-el7-4.noarch.rpm
```

//complete check

```
[root@localhost ~]# cd /temp
```

```
[root@localhost ~]# ls -la | grep mysql
```

```
[root@localhost ~]# rpm -vih /temp/mysql80-community-release-el7-4.noarch.rpm
```

```
[root@localhost ~]# yum repolist enabled | grep mysql
```

//install mysql

```
[root@localhost ~]#rpm --import https://repo.mysql.com/RPM-GPG-KEY-mysql-2022
```

```
[root@localhost ~]# yum install mysql-community-server
```

//To verify after installation

```
[root@localhost ~]# service mysqld start
```

```
[root@localhost ~]# service mysqld status
```

//Set password for root to log in mysql

```
[root@localhost ~]# cat /var/log/mysqld.log | grep 'temporary password'
```

```
[root@localhost ~]# mysql -uroot -p'*****'
```

//Create a test database and a test user with appropriate permissions to access the database

//Create a MySQL database with a table of the desired size. For this example, we will create a table with 1 million rows.

//Use the following command to create a database named sbtest

```
CREATE DATABASE sbtest;
```

```
USE sbtest;
```

Document Description	Stress Test Report using sysbench	Date	2023.04.12
Document Filename	StressTestReport.pdf	Revision#	1.0
Document Author	House Hsu		
Author Email	house40105@gmail.com		

```
CREATE TABLE sbtest (
    id INT UNSIGNED NOT NULL AUTO_INCREMENT,
    k INT UNSIGNED NOT NULL DEFAULT 0,
    c CHAR(120) NOT NULL DEFAULT "",
    pad CHAR(60) NOT NULL DEFAULT "",
    PRIMARY KEY (id),
    KEY k_1 (k)
) ENGINE=InnoDB;
```

//To change password authentication caching_sha2_password to mysql_native_password due to MySQL 8.0 version

```
ALTER USER 'root'@'localhost' IDENTIFIED WITH mysql_native_password BY '*****';
```

//Load the table with data using the following command

```
sysbench --db-driver=mysql --mysql-user=root --mysql-password=Happy8.0 \
--mysql-db=sbtest --table-size=1000000 /usr/share/sysbench/oltp_read_write.lua \
prepare
```

```
sysbench 1.0.17 (using system LuaJIT 2.0.4)
Creating table 'sbtest1'...
Inserting 1000000 records into 'sbtest1'
Creating a secondary index on 'sbtest1'...
```

//Run the stress test using the following command

//we are running the oltp_read_write benchmark with 64 threads for a duration of 300 seconds, and the report interval is set to 10 seconds

```
sysbench --db-driver=mysql --mysql-user=root --mysql-password=Happy8.0 \
--mysql-db=sbtest --table-size=1000000 /usr/share/sysbench/oltp_read_write.lua \
--threads=64 --time=300 --report-interval=10 run
```

//Once the test is completed, sysbench will generate a report showing various metrics such as transactions per second, latency, and CPU usage. An example report is shown below:

Document Description	Stress Test Report using sysbench	Date	2023.04.12
Document Filename	StressTestReport.pdf	Revision#	1.0
Document Author	House Hsu		
Author Email	house40105@gmail.com		

```

SQL statistics:
  queries performed:
    read:                1927100
    write:               550600
    other:               275300
    total:               2753000
  transactions:         137650 (458.75 per sec.)
  queries:              2753000 (9174.95 per sec.)
  ignored errors:        0      (0.00 per sec.)
  reconnects:           0      (0.00 per sec.)

General statistics:
  total time:           300.0550s
  total number of events: 137650

Latency (ms):
  min:                  5.55
  avg:                  139.49
  max:                  814.60
  95th percentile:     227.40
  sum:                  19201148.27

Threads fairness:
  events (avg/stddev):  2150.7812/20.45
  execution time (avg/stddev): 300.0179/0.01

```

//This report shows the number of queries performed, transactions, and latency. It also includes the total time taken to complete the test and the thread fairness of the benchmark. You can use this report to identify any bottlenecks or performance issues in your MySQL database.

//save the output to a file called report.txt for analysis and interpretation

```

sysbench --db-driver=mysql --mysql-user=root --mysql-password=Happy8.0 \
--mysql-db=sbtest --table-size=1000000 /usr/share/sysbench/oltp_read_write.lua \
--threads=64 --time=300 --report-interval=10 run> report.txt

```

//Clean up the test data by running the following command

```

sysbench --db-driver=mysql --mysql-user=root --mysql-password=Happy8.0 \
--mysql-db=sbtest --table-size=1000000 /usr/share/sysbench/oltp_read_write.lua \
Cleanup

```

When you use sysbench to test a MySQL database, it generates a report that includes various metrics to help you evaluate the system's performance and reliability.

Here are some of the key metrics in the report and what they mean:

Document Description	Stress Test Report using sysbench	Date	2023.04.12
Document Filename	StressTestReport.pdf	Revision#	1.0
Document Author	House Hsu		
Author Email	house40105@gmail.com		

Queries performed: The total number of queries executed during the test. This includes reads, writes, and other queries.

Transactions: The total number of transactions executed during the test. A transaction is a set of related operations that must be completed atomically, otherwise it is considered a failure.

Queries: The total number of queries executed during the test (including both transactional and non-transactional queries).

Ignored errors: The number of errors ignored during the test. If you encounter errors during the test, they will be recorded in the report.

Reconnects: The number of times MySQL reconnected during the test. If a connection is lost, sysbench will automatically attempt to reconnect to the MySQL database.

In addition to these metrics, the report provides information on latency and thread fairness.

Latency: Reports the latency during the test, including the minimum, average, and maximum latency, as well as the 95th percentile of latency. Latency refers to the time it takes to complete a transaction from the time it was submitted.

Threads fairness: Reports the average and standard deviation of the number of events each thread processed to evaluate thread fairness.

The report also provides information on CPU usage, I/O speed, and network traffic. These metrics can help you evaluate the system's load capacity and scalability.

Overall, the sysbench report provides rich metrics and information to help you evaluate the performance and reliability of a MySQL database system.