**ZOHO - Summer Internship**

May 21, 2024

1. **Induction Program** 
   1. Login to zoho people
   2. Filled the basic details
2. **Went to team place**
   1. 7th floor , got laptop

**Demo Task 1 : Install Tomcat server**

Reference link : <https://anamul-haque.medium.com/how-to-install-tomcat-on-ubuntu-and-deploy-war-file-in-tomcat-24039e8c3278>

**Step 1: Install Java**

1. Check the java version by `**java -version**`
2. If not available download java
   1. **sudo apt-get update**
   2. **sudo apt-get install default-jdk**

**Step2 : Create Tomcat User and Group**

What is User Group ?

All Linux operating systems are designed as multi-user operating systems. This means that they provide the capabilities and related tools to create and handle multiple users within a system. One such tool is user groups. **A user group is simply a collection of users**

Why do i need to create a new user and group?

Ans **👇**

<https://serverfault.com/questions/563736/debian-and-tomcat-why-creating-a-dedicated-user>

Create a new user group and user.

1. ***sudo groupadd tomcat***
2. **sudo useradd -s /bin/false -g tomcat -d /opt/tomcat tomcat**
   1. ‘-s SHELL’ The name of a new user's login shell. This option sets the SHELL variable in /etc/default/useradd.
   2. ‘-g GROUP\_NAME’ The group name
   3. ‘-d DIR’ The new user will be created using DIR as the value for the user's login directory.

**Step 3: Download Tomcat 9**

1. Go to the directory where the new user created ‘**cd /opt**’
2. Wget ‘<https://dlcdn.apache.org/tomcat/tomcat-9/v9.0.89/bin/apache-tomcat-9.0.89.tar.gz>’
3. **sudo tar xzvf apache-tomcat-9\*tar.gz -** extract the file
4. **sudo mv apache-tomcat-9.0.89 tomcat**

**Step 4: Update Permissions**

***sudo chgrp -R tomcat /opt/tomcat***

***chmod 755 /opt/tomcat***

***sudo chown -R tomcat webapps/ work temp/ logs***

**Step 5: Create a systemd Service File**

—------------------------------------------

<https://www.youtube.com/watch?v=TNZuqEglH9Y>

Install apache-tomcat-10.1.24.tar.gz from website

sudo tar xzvf apache-tomcat-10.1.24.tar.gz -C /opt/tomcat --strip-components=1

sudo /opt/tomcat/bin/startup.sh

Task 1: Create a mysql automation performance/load tester using java

1. Analyse existing benchmark tools
   1. Sysbench - [How to Benchmark Performance of MySQL & MariaDB Using SysBench | Severalnines](https://severalnines.com/blog/how-benchmark-performance-mysql-mariadb-using-sysbench/)
   2. DBT2 - [Benchmarking MySQL with dbt2 | Cagesong on Earth](https://cagesongonearth.wordpress.com/2020/03/01/benchmarking-mysql-with-dbt2/)
   3. Mysqlslap - [MySQL 8.0 Reference Manual :: 6.5.8 mysqlslap — A Load Emulation Client](https://dev.mysql.com/doc/refman/8.0/en/mysqlslap.html)
   4. HammerDB - GUI - [3. Choosing a Database for running TPROC-H workloads](https://www.hammerdb.com/docs/ch11s03.html#d0e4326)
   5. Jmeter
2. Mysql configuration fine tuning

[Fine tuning MYSQL for best performance - Stack Overflow](https://stackoverflow.com/questions/70389860/fine-tuning-mysql-for-best-performance)

May 22, 2024

By sudalai sir

1. [PostgreSQL: Documentation: 16: pgbench](https://www.postgresql.org/docs/current/pgbench.html)
2. What is load/performance testing ? It is Purpose ?
3. Benchmark standards .. TPC
4. HammerDB
5. Postgresql Bench Mark
   1. Postgresql - an RDBMS, but have more features than MySQL, actually it is an object relational database , can make custom data types, like a class and access it, interestingly it also have json, key-value pair datatypes to store unstructured data.
   2. PG Bench mark - it runs same SQL query over and over and get the average transaction time, by default uses the TCP-B standard.
6. Load/Performance testing?
   1. [Performance Testing and It's Types With Practical Examples | Software Testing](https://www.youtube.com/watch?v=-dI_gKRePRo)
   2. Performance testing is nothing but, testing an application’s stability and response time by applying load,
      1. where stability is how much users can the application handle without crashing
      2. response time is the total time taken to send the request , process the request and send the response back to the user.
      3. Load is no of users using the application at a particular time.
   3. Types of Performance Testing
      1. Load Testing
         1. Testing the stability and response time of an application by applying load which is less than or equal to the designed load number [ test load <= designed load ]
      2. Stress Testing
         1. Testing the stability and response time of an application by applying load which is greater than the designed load number [ Test Load > designed load]
      3. Scalability Testing.
         1. Testing the stability and response time of an application by applying load which is greater than the designed load number and also track at which point the application crashes
      4. Volume Testing (flood testing).
         1. Testing the stability and response time of a database by transferring huge amounts of data to it.
         2. Ex: while creating a bank transaction application database, we need to check the volume that is able to handle huge data transfer from the old database.
      5. Soak Testing (Endurance Testing).
         1. Testing the stability and response time of an application by applying load continuously like 48 hours or 72 hours.
   4. Purpose of Performance Testing.
      1. To check where the application can handle the designed no.of users.
      2. To check the breaking point of the application
      3. To identify the response time and latency.

Discussed with veera anna May 22, 2024 11:12 AM

1. Performance of DB is not only based on the RAM, but also based on the program, the query plan (query execution plan) , and the sequence of steps used to execute the query.
2. Load Testing is not only based on the no.of users connected to a database, it mainly depends on how they are using it like inputs they are given.
3. There may be 100 connections in a DB but only 10 threads are actively running.
4. Threads and Connections are different ? don't know threads in db.

---------------------

1. Benchmark standards .. TPC
   1. TPC - Transaction Processing Performance Council
   2. It is a benchmark standard to test the database performance
   3. It simulates some of the user entry environment and querying the database to get the latency , transactions per second like things
2. HammerDB
   1. An open source Database benchmarking tool.
   2. Can be applied to popular databases like mysql, postgresql, mongodb,etc.
   3. Supported workloads,
      1. TPROC-C = OLTP -> for transactional apps
      2. TPROC-H = OLAP -> for analytics and reporting
   4. It is a parallel benchmarking software, concurrency control must be in the database.
   5. Programming Languages used in HammerDB
      1. Database commands in SQL
      2. Application logic in stored procedures
      3. Database interfaces in C
   6. Tcl as Glue Language(facilitate communication between different programming languages and software components)
      1. It provides truly parallel multithreading.
   7. HammerDB For Linux
      1. Install HammerDB.tar.gz latest version from the hammerDB website.
      2. Extract the tar file using `tar -xvf ./file\_name` cmd
      3. <https://www.hammerdb.com/blog/uncategorized/hammerdb-command-line-build-and-test-examples/>
      4. Create a new .tcl file to configure the hammerDB

```

#!/bin/tclsh

puts "SETTING CONFIGURATION"

dbset db mysql

diset connection mysql\_host 127.0.0.1

diset connection mysql\_port 3306

diset connection mysql\_socket /var/run/mysqld/mysqld.sock

diset tpcc mysql\_count\_ware 800

diset tpcc mysql\_dbase tpcc2

diset tpcc mysql\_partition true

diset tpcc mysql\_num\_vu 64

diset tpcc mysql\_storage\_engine innodb

print dict

buildschema

```

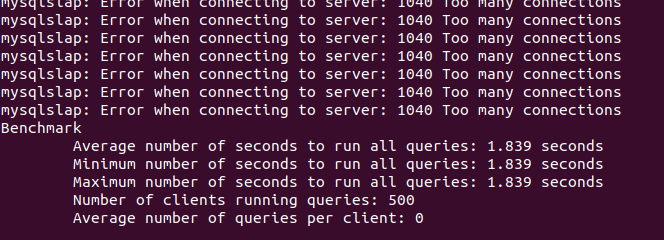
* + 1. /var/run/mysqld/mysqld.sock - this socket folder may vary from different distros
    2. Buildschema - this is the command which runs the build schema which has more users.
    3. To run the build
       1. ./hammerdbcli
       2. hammerdb> source filename.tcl
    4. I stuck in an error more than half a day, that Unexpected error - cant able to write on read-only database.
    5. I ran it first in gui only it is not clearly told the error, after running it in cli, it shows what file is read only for this user, i changed the write permission to root user from normal user, then also it is not worked, after restarting it works fine.
    6. … continuation on May 23, 2024

MySQL Slap :

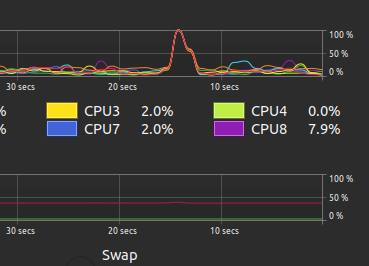
It is a diagnostic program designed to emulate client load for a MySQL server and to report the timing of each stage. It works as if multiple clients are accessing the server.

Cmd:

sudo mysqlslap --host=localhost --user=root --password=password --concurrency=500 --iterations=1 --number-int-cols=20 --number-char-cols=30 --auto-generate-sql

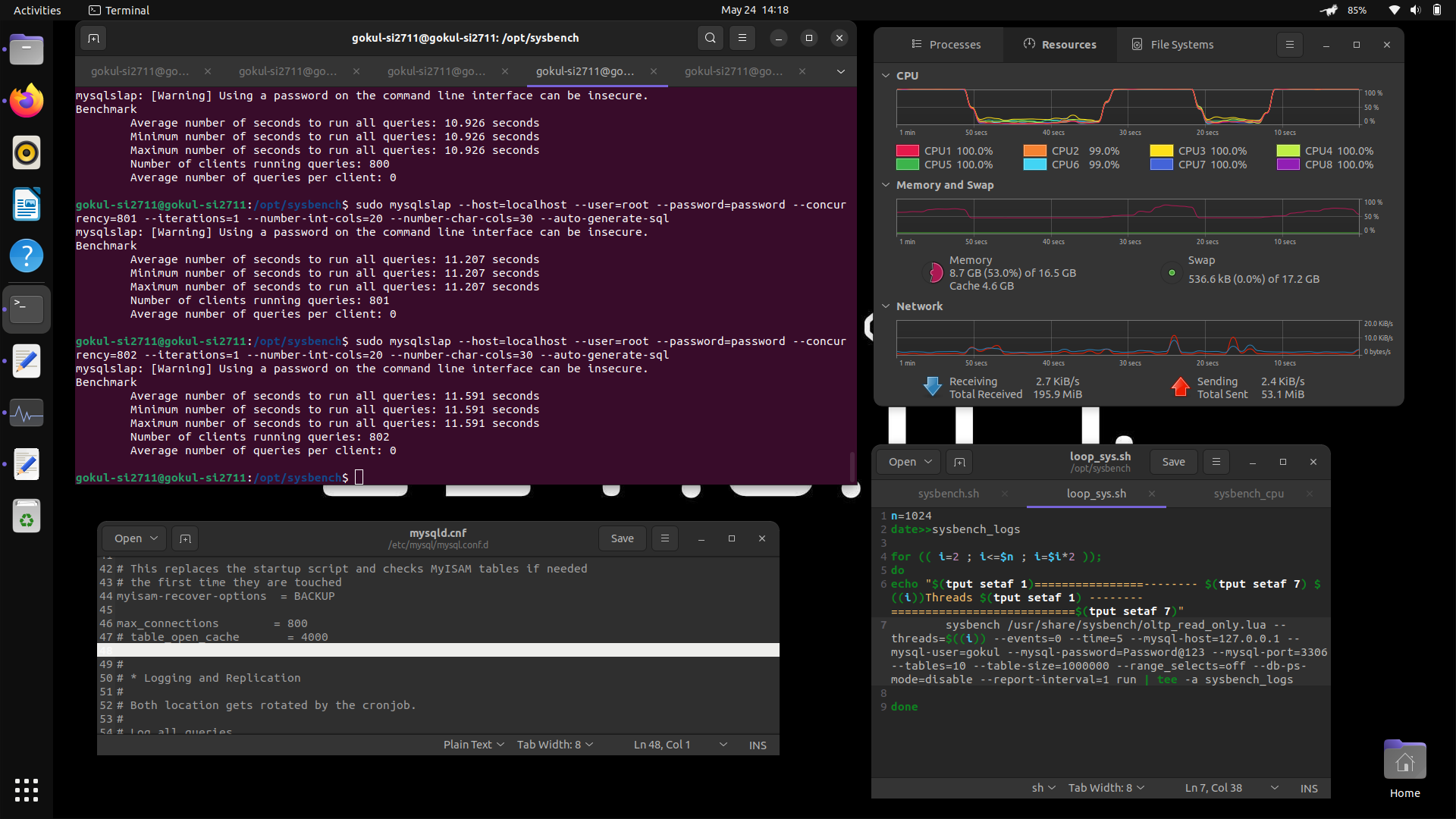


By Observation



When the number of the clients increases , usage of the RAM is not increasing the CPU usage is only increasing.

Increased the max\_connection in mysql server then run the 800 threads



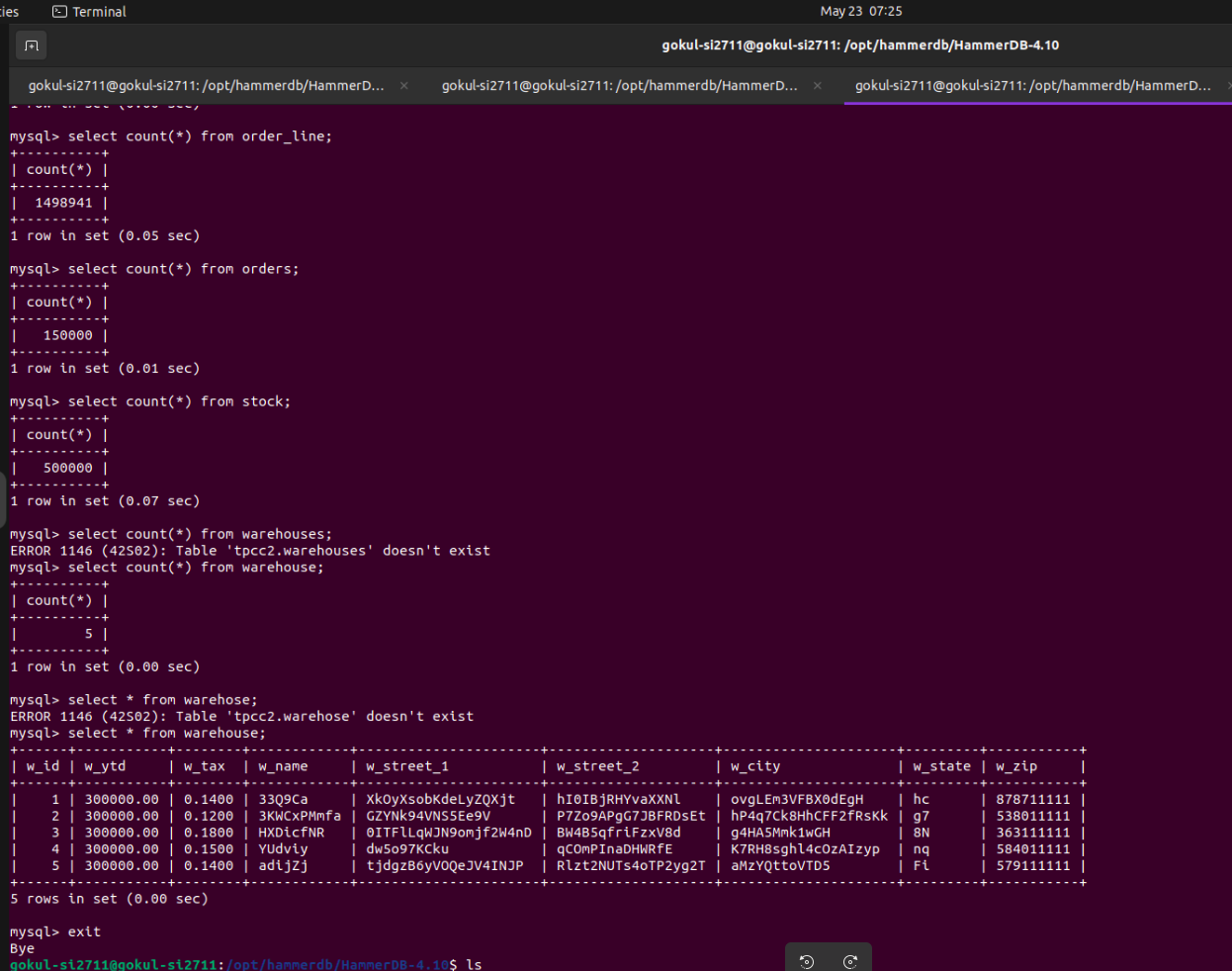
Purely 100% of the CPU used and RAM fluctuates…

May 23, 2024

4. HammerDB

g. HammerDB for linux

x. After running the script in the internet , i waited around half a day for the output but after 11pm the program terminated.BY running around 65 users and 800 warehouses, it takes around 68gb space to create almost 80% of the database but not completed fully after 5 hours of running the same program, due to insufficient space the program cant able to execute further. so chang the vusers and warehouse number.



* + 1. After some changes in the users and warhouses, it created some tables in the database during the build schema process.
    2. Running the next command from internet in cli gives error, so i ran it in the gui it works fine now.
    3. The HammerDB log file while running virtual users from GUI.

Hammerdb Log @ Thu May 23 09:55:37 IST 2024

+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-

Timestamp 1 @ Thu May 23 09:55:40 IST 2024

Vuser 1:Ssl\_cipher

Timestamp 1 @ Thu May 23 09:55:40 IST 2024

Vuser 1:Beginning rampup time of 2 minutes

Timestamp 2 @ Thu May 23 09:55:41 IST 2024

Vuser 2:Ssl\_cipher

Timestamp 2 @ Thu May 23 09:55:41 IST 2024

Vuser 2:Processing 10000000 transactions with output suppressed...

Timestamp 3 @ Thu May 23 09:55:41 IST 2024

Vuser 3:Ssl\_cipher

Timestamp 3 @ Thu May 23 09:55:41 IST 2024

Vuser 3:Processing 10000000 transactions with output suppressed...

Timestamp 1 @ Thu May 23 09:56:40 IST 2024

Vuser 1:Rampup 1 minutes complete ...

Timestamp 1 @ Thu May 23 09:57:40 IST 2024

Vuser 1:Rampup 2 minutes complete ...

Timestamp 1 @ Thu May 23 09:57:40 IST 2024

Vuser 1:Rampup complete, Taking start Transaction Count.

Timestamp 1 @ Thu May 23 09:57:40 IST 2024

Vuser 1:Timing test period of 1 in minutes

Timestamp 1 @ Thu May 23 09:58:40 IST 2024

Vuser 1:1 ...,

Timestamp 1 @ Thu May 23 09:58:40 IST 2024

Vuser 1:Test complete, Taking end Transaction Count.

Timestamp 1 @ Thu May 23 09:58:40 IST 2024

Vuser 1:2 Active Virtual Users configured

Timestamp 1 @ Thu May 23 09:58:40 IST 2024

Vuser 1:TEST RESULT : **System achieved 2969 NOPM from 6798 MySQL TPM**

NOPM - new orders per minute

"New orders" typically refers to requests or instructions received by a system

TPM - Transactions per minute

NOPM vs TPM

<https://github.com/TPC-Council/HammerDB/discussions/197>

"System achieved 2969 NOPM from 6798 MySQL TPM"

* NOPM stands for "Number of Operations Per Minute," indicating the number of database operations (such as reads, writes, updates, etc.) performed in a minute.
* MySQL TPM stands for "Transactions Per Minute" in the context of MySQL database operations, indicating the number of transactions processed in a minute.

So, the statement means that the system achieved 2969 database operations per minute while processing 6798 MySQL transactions per minute. It's a measure of the system's efficiency and throughput in handling database operations.

What can i get from the NOPM and TPM

**Throughput:** The system can handle a number of transactions per minute (6798 MySQL TPM)

**Operations per minute:** The total number of operations performed in all the transactions in a minute (2696 NOPM).

-> from the above metrics we can plan our database performance accordingly.

Results from CLI - f1,2,4,8 virtual users.

Hammerdb Log @ Thu May 23 10:09:16 IST 2024

+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-

Vuser 1:Ssl\_cipher

Vuser 1:Beginning rampup time of 2 minutes

Vuser 2:Ssl\_cipher

Vuser 2:Processing 10000000 transactions with output suppressed...

Vuser 1:Rampup 1 minutes complete ...

Vuser 1:Rampup 2 minutes complete ...

Vuser 1:Rampup complete, Taking start Transaction Count.

Vuser 1:Timing test period of 1 in minutes

Vuser 1:1 ...,

Vuser 1:Test complete, Taking end Transaction Count.

Vuser 1:1 Active Virtual Users configured

Vuser 1:TEST RESULT : System achieved 3083 NOPM from 7329 MySQL TPM

Hammerdb Log @ Thu May 23 10:12:28 IST 2024

+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-

Vuser 1:Ssl\_cipher

Vuser 1:Beginning rampup time of 2 minutes

Vuser 2:Ssl\_cipher

Vuser 2:Processing 10000000 transactions with output suppressed...

Vuser 3:Ssl\_cipher

Vuser 3:Processing 10000000 transactions with output suppressed...

Vuser 1:Rampup 1 minutes complete ...

Vuser 1:Rampup 2 minutes complete ...

Vuser 1:Rampup complete, Taking start Transaction Count.

Vuser 1:Timing test period of 1 in minutes

Vuser 1:1 ...,

Vuser 1:Test complete, Taking end Transaction Count.

Vuser 1:2 Active Virtual Users configured

Vuser 1:TEST RESULT : System achieved 2967 NOPM from 7043 MySQL TPM

Hammerdb Log @ Thu May 23 10:15:52 IST 2024

+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-

Vuser 1:Ssl\_cipher

Vuser 1:Beginning rampup time of 2 minutes

Vuser 2:Ssl\_cipher

Vuser 2:Processing 10000000 transactions with output suppressed...

Vuser 3:Ssl\_cipher

Vuser 3:Processing 10000000 transactions with output suppressed...

Vuser 4:Ssl\_cipher

Vuser 4:Processing 10000000 transactions with output suppressed...

Vuser 5:Ssl\_cipher

Vuser 5:Processing 10000000 transactions with output suppressed...

Vuser 1:Rampup 1 minutes complete ...

Vuser 1:Rampup 2 minutes complete ...

Vuser 1:Rampup complete, Taking start Transaction Count.

Vuser 1:Timing test period of 1 in minutes

Vuser 1:1 ...,

Vuser 1:Test complete, Taking end Transaction Count.

Vuser 1:4 Active Virtual Users configured

Vuser 1:TEST RESULT : System achieved 3759 NOPM from 8743 MySQL TPM

Hammerdb Log @ Thu May 23 10:19:10 IST 2024

+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-

Vuser 1:Ssl\_cipher

Vuser 1:Beginning rampup time of 2 minutes

Vuser 2:Ssl\_cipher

Vuser 2:Processing 10000000 transactions with output suppressed...

Vuser 3:Ssl\_cipher

Vuser 3:Processing 10000000 transactions with output suppressed...

Vuser 4:Ssl\_cipher

Vuser 4:Processing 10000000 transactions with output suppressed...

Vuser 5:Ssl\_cipher

Vuser 5:Processing 10000000 transactions with output suppressed...

Vuser 6:Ssl\_cipher

Vuser 6:Processing 10000000 transactions with output suppressed...

Vuser 7:Ssl\_cipher

Vuser 7:Processing 10000000 transactions with output suppressed...

Vuser 8:Ssl\_cipher

Vuser 8:Processing 10000000 transactions with output suppressed...

Vuser 9:Ssl\_cipher

Vuser 9:Processing 10000000 transactions with output suppressed...

Vuser 1:Rampup 1 minutes complete ...

Vuser 1:Rampup 2 minutes complete ...

Vuser 1:Rampup complete, Taking start Transaction Count.

Vuser 1:Timing test period of 1 in minutes

Vuser 1:1 ...,

Vuser 1:Test complete, Taking end Transaction Count.

Vuser 1:8 Active Virtual Users configured

Vuser 1:TEST RESULT : System achieved 3027 NOPM from 7016 MySQL TPM

From the Logs we can tell the results

Vuser 1:TEST RESULT : System achieved 3083 NOPM from 7329 MySQL TPM

Vuser 1:TEST RESULT : System achieved 2967 NOPM from 7043 MySQL TPM

Vuser 1:TEST RESULT : System achieved 3759 NOPM from 8743 MySQL TPM

Vuser 1:TEST RESULT : System achieved 3027 NOPM from 7016 MySQL TPM

On an average our configuration can handle 7532 Transaction Per Minute and 3209 New Orders Per Minute (new order == instructions).

Demo JDBC - CRUD.

1. Download Eclipse from snap store
   1. sudo apt install snapd
   2. sudo snap install --classic eclipse
2. Download MySql driver from official site and extract it to get the .jar file
   1. [Download MySQL Connector and Configure JDBC in Eclipse IDE](https://www.youtube.com/watch?v=y1IU65ffx7A)
3. Open eclipse, create a new class, right click on the JRE-> click modules-> click add external JAR file -> select the path -> apply and close.

package crud;

import java.sql.Connection;

import java.sql.DriverManager;

import java.sql.SQLException;

import java.sql.ResultSet;

import java.sql.Statement;

public class Jdbc {

public static void main(String[] args) {

String url = "jdbc:mysql://localhost:3306/crud";

String username = "gokul";

String password = "Password@123";

try (

Connection connection = DriverManager.getConnection(url, username, password);

Statement statement = connection.createStatement();

ResultSet resultSet = statement.executeQuery("SELECT \* FROM employee");

)

{

System.out.println("Connected to the database!");

while (resultSet.next()) {

int id = resultSet.getInt("eid");

String name = resultSet.getString("ename");

String office = resultSet.getString("eoffice");

System.out.println("ID: " + id + ", Name: " + name + ", Office: " + office);

}

} catch (SQLException e) {

System.err.println("Failed to connect to the database.");

e.printStackTrace();

}

}

}

* 1. Connection connection -> this line is responsible for the connection to database with the use
  2. Statement statement -> statement is the object which can able to execute the query we are giving using the connection
  3. ResultSet resultSet - > results set is the object which can able to give store the data fetched from the statement.executeQuery.
  4. try() {} -> is try-with-resources which can able to autoclose the objects such as connection, statement, resultset.
  5. statement.executeUpdate -> is to insert, update or delete returns either (1) the row count for SQL Data Manipulation Language (DML) statements or (2) 0 for SQL statements that return nothing.
  6. We can also connect with other systems if /etc/mysql/mysql.conf.d/mysqld.cnf -> bind address is 0.0.0.0 or the accessing system’s ip address.

1. Java Microbenchmark Harness (JMH):
   1. A java tool used to benchmark the databases.
   2. Using code provided by chatgpt , need to add jmh.core in model-info.java, convert the configuration to maven project, and add dependencies to pom.xml

You can add the JMH dependency to your project using Maven or Gradle. Here, I'll guide you through adding it with Maven.

Right-click on your project in the Project Explorer.

Go to Configure > Convert to Maven Project.

If you haven't configured Maven in Eclipse, you'll need to do so. Otherwise, proceed to the next step.

Open your pom.xml file.

Add the following JMH dependency within the **<dependencies>** section:

xml

**<dependency>**

**<groupId>org.openjdk.jmh</groupId>**

**<artifactId>jmh-core</artifactId>**

**<version>1.32</version>**

**</dependency>**

**Step 4: Write a Benchmark Class**

Create a new Java class that contains your benchmark methods. Here's a simple example:

java

import org.openjdk.jmh.annotations.Benchmark;

import org.openjdk.jmh.annotations.BenchmarkMode;

import org.openjdk.jmh.annotations.Mode;

import org.openjdk.jmh.annotations.OutputTimeUnit;

import java.util.concurrent.TimeUnit;

public class MyBenchmark {

@Benchmark

@BenchmarkMode(Mode.Throughput)

@OutputTimeUnit(TimeUnit.SECONDS)

public void myBenchmarkMethod() {

// Your benchmark code here

}

// Add more benchmark methods as needed

}

* 1. Importantly make the folder like

project

|-- src

|-- main

|-- java

|-- jmh

|-- Jmh.java

|-- com

|-- example

|-- MyBenchmark.java

|-- pom.xml

And the sample execution files like 1. Jmh.java.

package jmh;

import org.openjdk.jmh.runner.RunnerException;

import org.openjdk.jmh.runner.options.Options;

import org.openjdk.jmh.runner.options.OptionsBuilder;

import com.MyBenchmark;

import org.openjdk.jmh.runner.Runner;

public class Jmh {

public static void main(String[] args) throws RunnerException {

Options opt = new OptionsBuilder()

.include(MyBenchmark.class.getSimpleName())

.forks(1)

.build();

new Runner(opt).run();

} }

2. MyBenchMark.java

package com;

import org.openjdk.jmh.annotations.Benchmark;

public class MyBenchmark {

*@Benchmark*

public void testMethod() {

// Benchmark code here

}

}

3. Pom.xml

<project xmlns="http://maven.apache.org/POM/4.0.0"

xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 https://maven.apache.org/xsd/maven-4.0.0.xsd">

<modelVersion>4.0.0</modelVersion>

<groupId>jmh</groupId>

<artifactId>jmh</artifactId>

<version>0.0.1-SNAPSHOT</version>

<name>jmh</name>

<properties>

<maven.compiler.source>11</maven.compiler.source>

<maven.compiler.target>11</maven.compiler.target>

<maven.compiler.release>11</maven.compiler.release>

</properties>

<build>

<plugins>

<plugin>

<artifactId>maven-compiler-plugin</artifactId>

<version>3.8.1</version>

<configuration>

<source>${maven.compiler.source}</source>

<target>${maven.compiler.target}</target>

<release>${maven.compiler.release}</release>

<annotationProcessorPaths>

<path>

<groupId>org.openjdk.jmh</groupId>

<artifactId>jmh-generator-annprocess</artifactId>

<version>1.37</version>

</path>

</annotationProcessorPaths>

</configuration>

</plugin>

<plugin>

<groupId>org.codehaus.mojo</groupId>

<artifactId>exec-maven-plugin</artifactId>

<version>3.0.0</version>

<executions>

<execution>

<id>run-benchmarks</id>

<phase>integration-test</phase>

<goals>

<goal>exec</goal>

</goals>

<configuration>

<classpathScope>test</classpathScope>

<executable>java</executable>

<arguments>

<argument>-classpath</argument>

<classpath />

<argument>org.openjdk.jmh.Main</argument>

<argument>.\*</argument>

</arguments>

</configuration>

</execution>

</executions>

</plugin>

</plugins>

</build>

<dependencies>

<dependency>

<groupId>org.openjdk.jmh</groupId>

<artifactId>jmh-core</artifactId>

<version>1.37</version>

</dependency>

<dependency>

<groupId>org.openjdk.jmh</groupId>

<artifactId>jmh-generator-annprocess</artifactId>

<version>1.37</version>

<scope>provided</scope>

</dependency>

</dependencies>

</project>

* 1. Benchmark Mode Cnt Score Error Units

MyBenchmark.testMethod thrpt 25 2909302831.324 ± 131249245.670 ops/s

* 1. Using a simple database and fetch query the results wil be like the below
  2. Result "com.MyBenchmark.testMethod":

280.476 ±(99.9%) 72.835 ops/s [Average]

(min, avg, max) = (252.382, 280.476, 304.528), stdev = 18.915

CI (99.9%): [207.641, 353.311] (assumes normal distribution)

* 1. # Run complete. Total time: 00:01:40
  2. Benchmark Mode Cnt Score Error Units

MyBenchmark.testMethod thrpt 5 280.476 ± 72.835 ops/s

* 1. The import thing is you have to build it after making changes like changing the function name, because in the previous build , it stores the function name and if you run it it will run only the stored function code.

May 24, 2024

MySQL performance tuning…

<https://www.devart.com/dbforge/mysql/studio/mysql-performance-tips.html>

1. Hardware - Hardware level performance tuning includes, switching to SSD from HDD, increasing RAM, changing the CPU.
2. Software - Software Level Performance tuning involved MySQL indexing, Changing Storage Engine, MySQL query optimization, MySQL configurations.
   1. MySQL indexing - it is a concept of assigning indexes to a query that repeated used to reduce the fetching time.

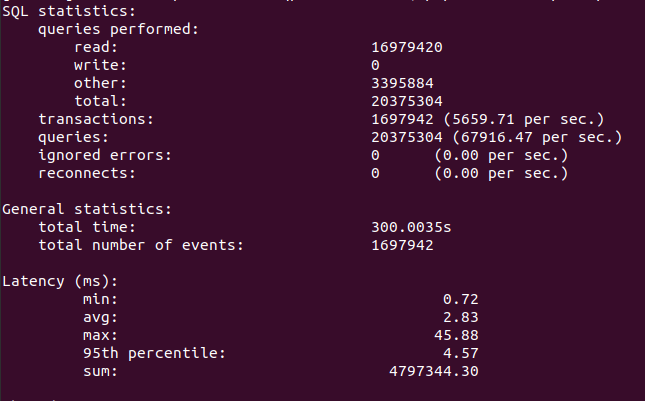
<https://www.digitalocean.com/community/tutorials/how-to-use-indexes-in-mysql>

* 1. Changing Storage Engine , so changing storage engine to innoDB, to make large storage database efficiently
  2. MySQL query optimization
     1. Use indexing on required places, dont use it too much due to changing it is a costly operation
     2. Use specific column name instead of select \* from table
     3. Use Inner Join than outer join, outer join causes more time , if you feel outer join is faster than inner join use it
     4. Avoid using LIKE “%xyz” it will search all the records, instead use it like “x%”.
     5. Use Distinct and Union wisely
  3. MySQL configurations.
     1. **Innodb\_buffer\_pool\_size -** The recommended value for this parameter is 70-80% of available memory. The larger your data sets are the larger the value should be
     2. **Max\_connection -** parameter defines the maximum permitted number of simultaneous client connections, default - 151.In order to avoid getting the "Too many connections" error, the value can be increased.
     3. **Query\_cache\_size -** This parameter sets the total amount of memory allocated to the query cache. The optimal value for it depends primarily on your working case and needs to be determined tentatively. The idea is to start very small—10MB—for example, then increase in small increments to 100-200MB. Adjusting query\_cache\_size, remember to enable query cache (query-cache-type ON). Note that large query cache size can result in a serious performance decrease.
     4. **Innodb\_io\_capacity -** This parameter specifies the number of I/O operations per second allowed for the tasks performed in the background and has the default value of 200. Generally, the value around 100 is suitable for average-level hard drives, while for faster and more modern storage devices higher values will be advantageous.
     5. **Innodb\_log\_file\_size -** This parameter specifies the size in bytes for each MySQL redo log file in a log group and has the default value of 134,217,728 (about 128 MB). The innodb\_log\_files\_in\_group parameter in its turn specifies the number of log files in the log group and has the default value of 2. In case the innodb\_log\_file\_size value is small for your workload and your application is write-intensive, we recommend increasing it. However, too large innodb\_log\_file\_size will increase the crash recovery time. So you will have to find its optimal size.

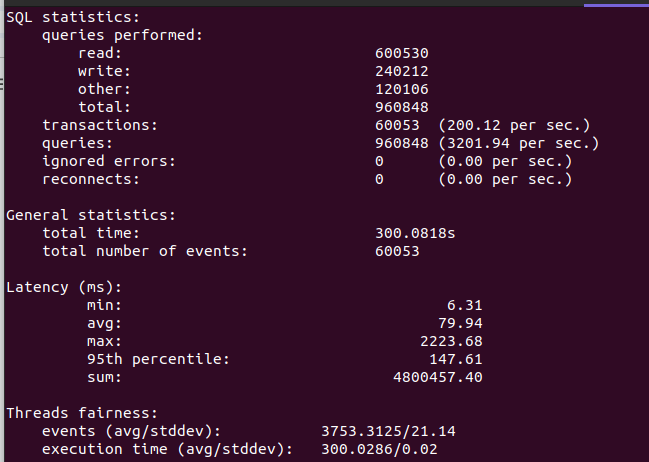
Sysbench

<https://severalnines.com/blog/how-benchmark-performance-mysql-mariadb-using-sysbench/>

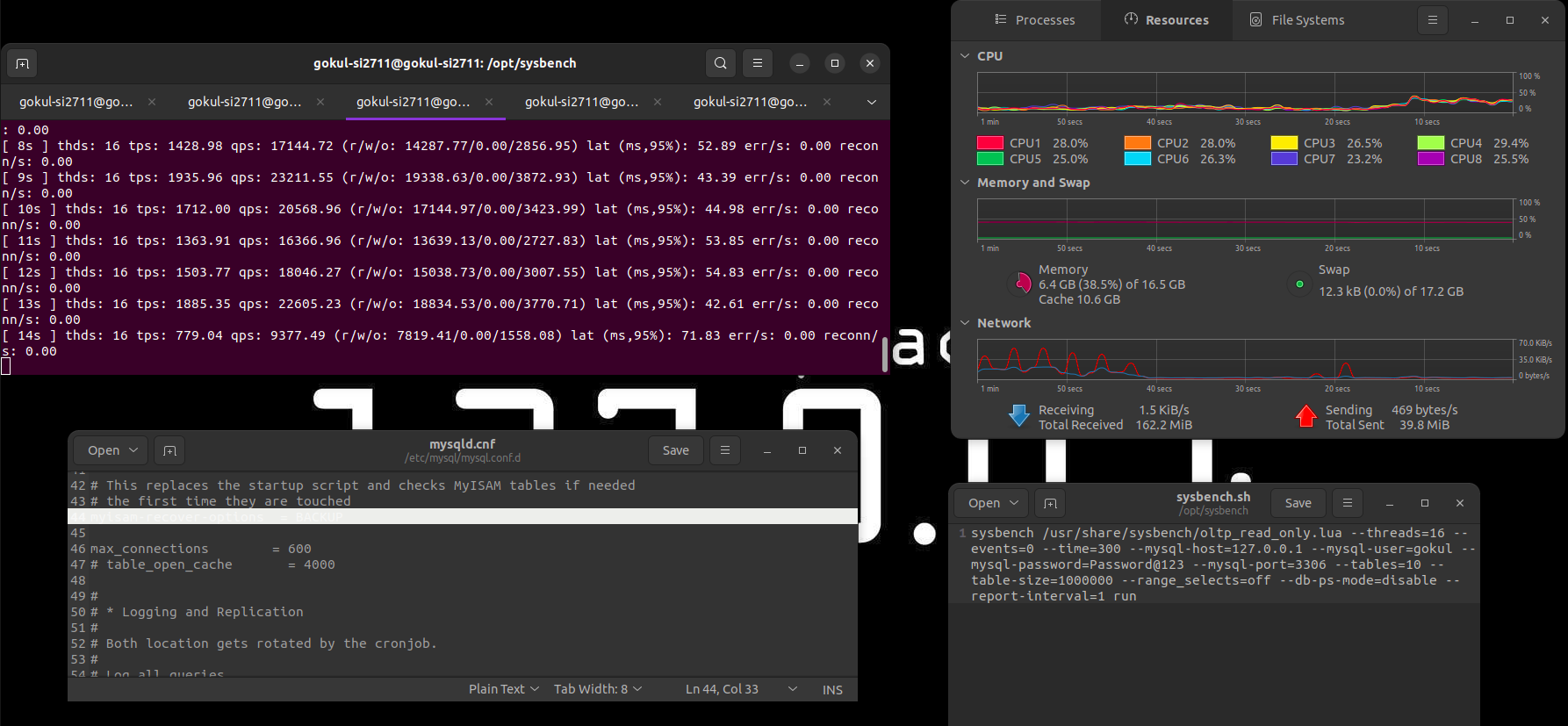
1. A open source bench mark tool
2. Install using - sudo apt install sysbench
3. First need to create database to test
   1. sysbench /usr/share/sysbench/oltp\_read\_only.lua --threads=4 --mysql-host=127.0.0.1 --mysql-user=gokul --mysql-password=Password@123 --mysql-port=3306 --tables=10 --table-size=1000000 **prepare**
   2. /usr/share/sysbench/oltp\_read\_only.lua - find where the location is and give.
   3. It will create 10 tables of rows 1million.
4. Second test it using any of the operation
   1. sysbench /usr/share/sysbench/oltp\_read\_write.lua --threads=16 --events=0 --time=300 --mysql-host=127.0.0.1 --mysql-user=gokul --mysql-password=Password@123 --mysql-port=3306 --tables=10 --table-size=1000000 --range\_selects=off --db-ps-mode=disable --report-interval=1 **run**
   2. We can able to run any of the operations like read\_only, read\_write, insert, update,delete etc..
5. Read only bench mark



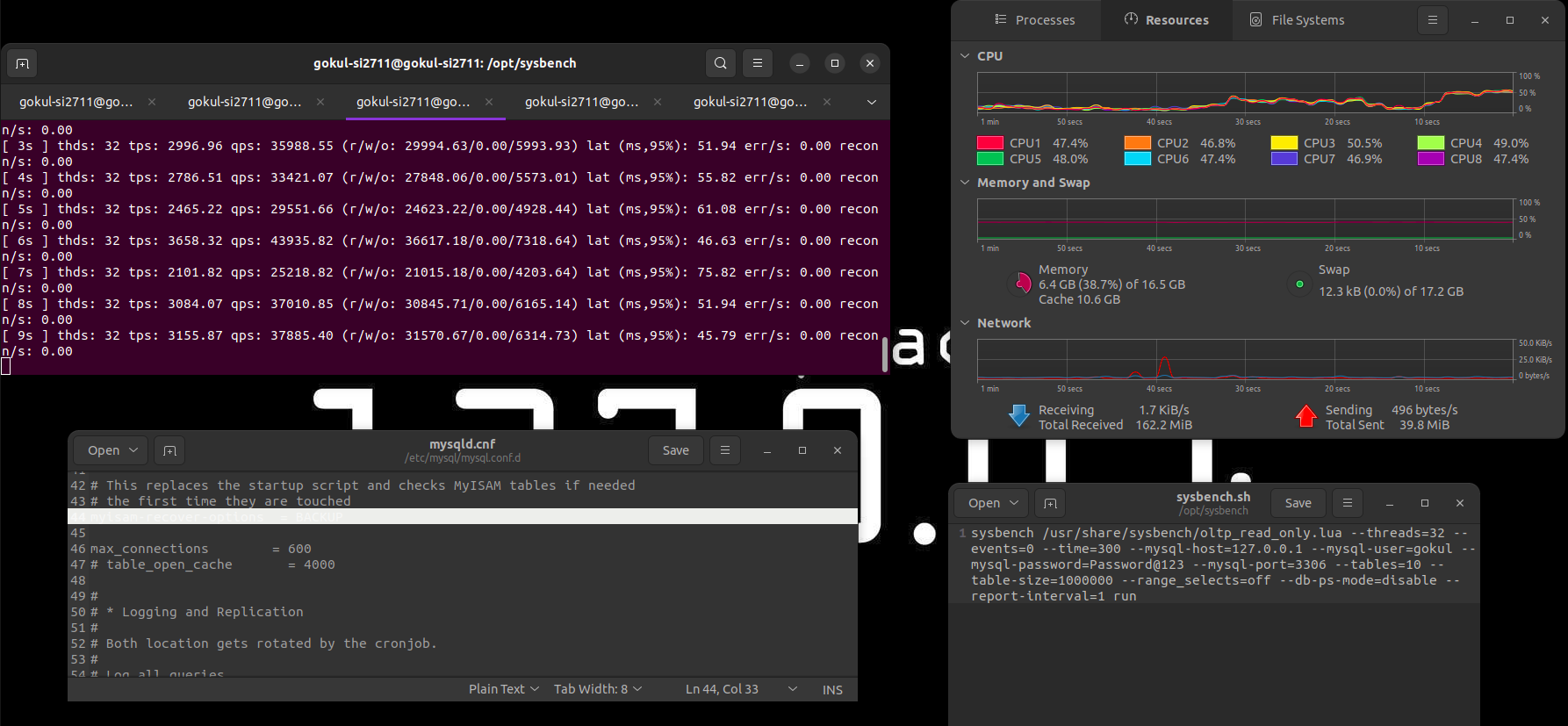
1. Read and Write bench mark



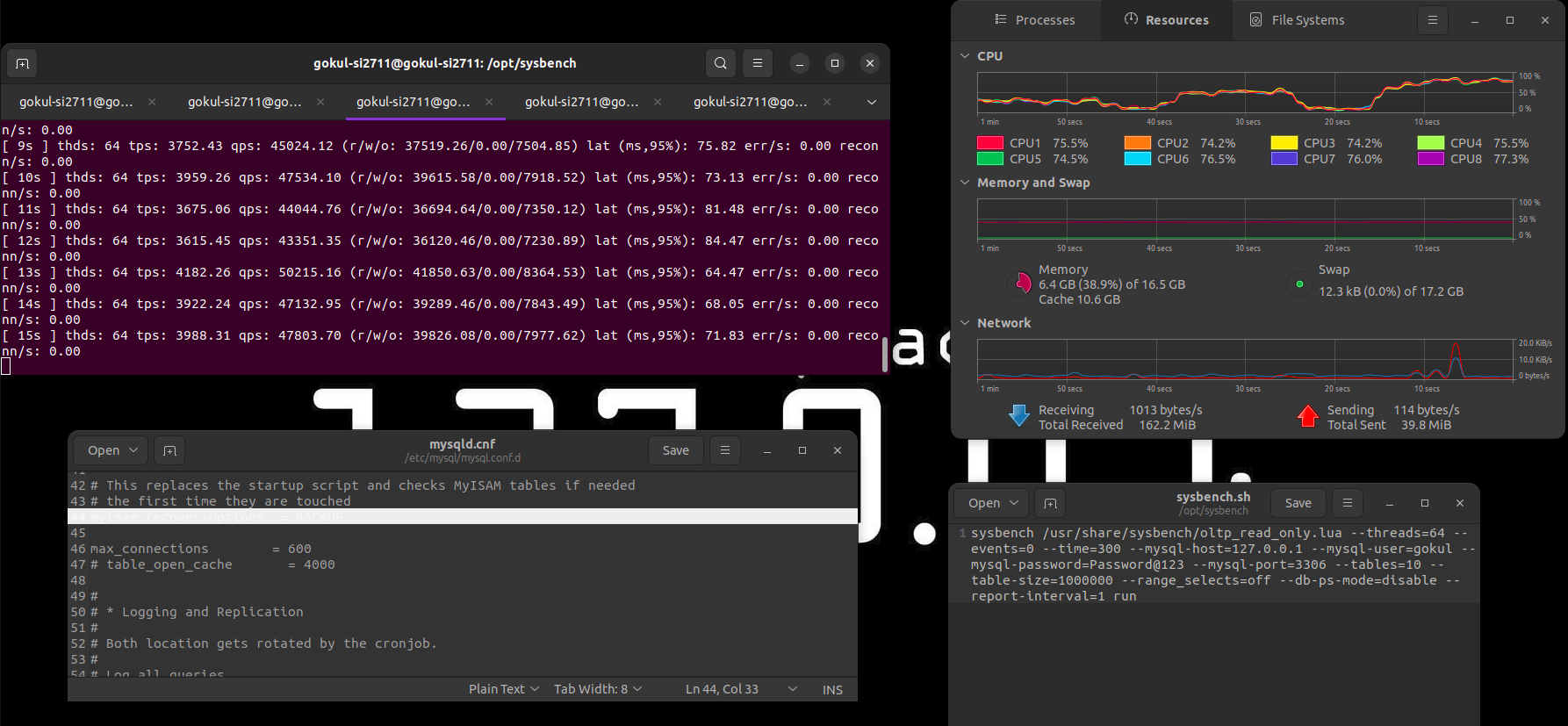
1. Sysbench threads is the no of users, for every thread one connection will be created, the maximum\_connection of mysql server by default is 151, if we give other than 152 the error occurs Too many connections
2. I have observed that if we are giving the same file to prepare and run it will take lot of cpu, ex: read\_only for both
   1. If we give like read\_only for prepare and refmad\_write for run means it will use less cpu which means the program needs to prepare and run the same file.
3. Cpu utilization test by 2 powers
   1. Cpu utilization while 16 threads - ~25% cpu



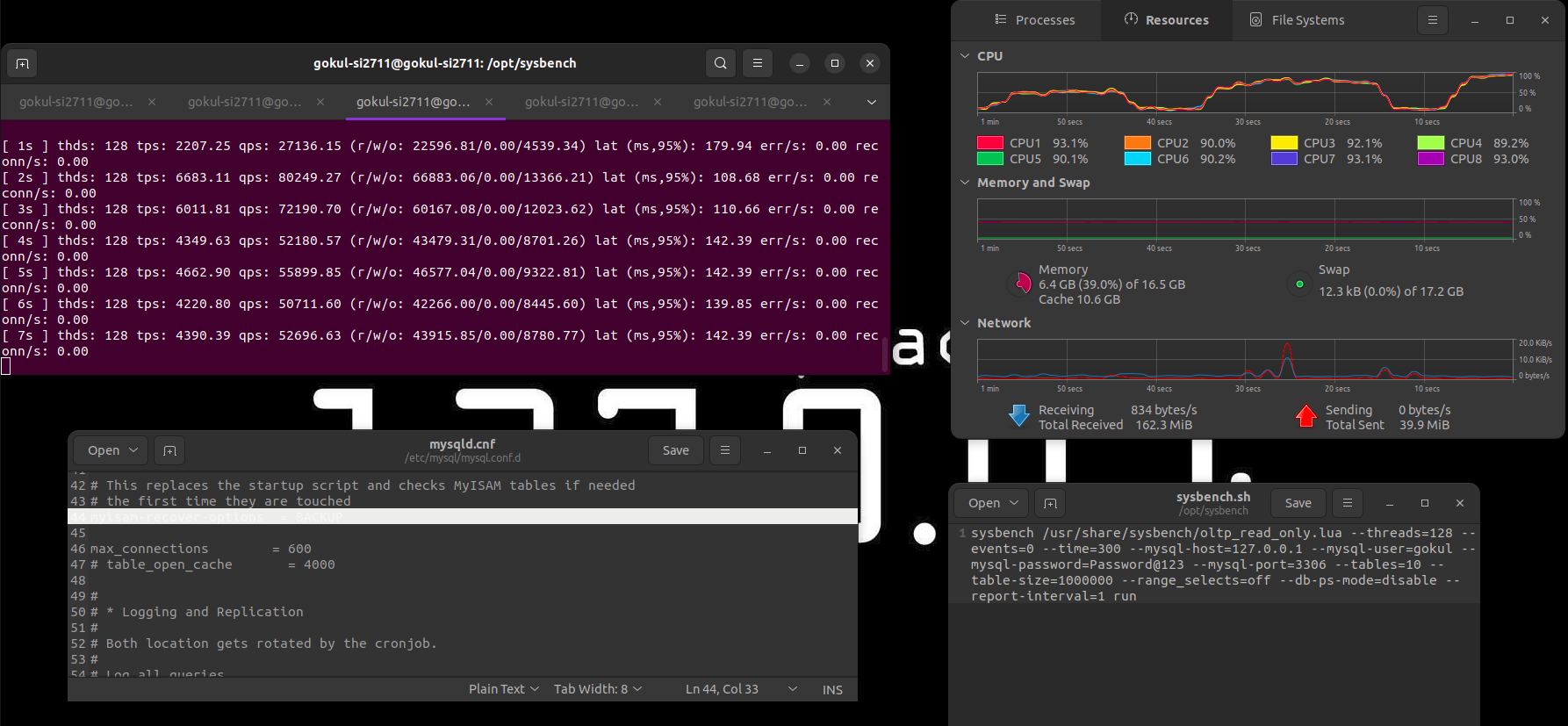
* 1. Cpu utilization while 32 threads - ~50%



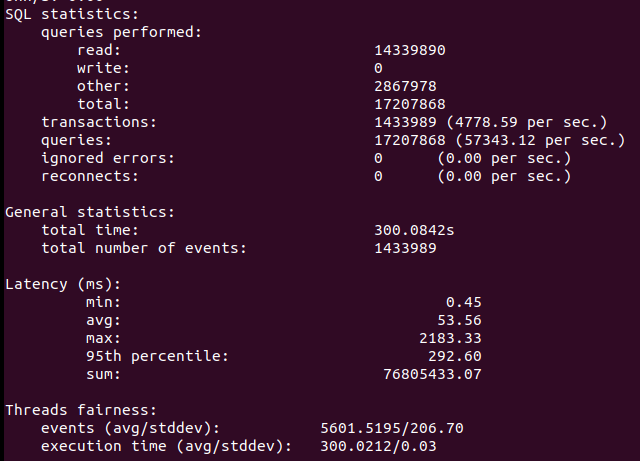
* 1. Cpu utilization while 64 threads - ~80%



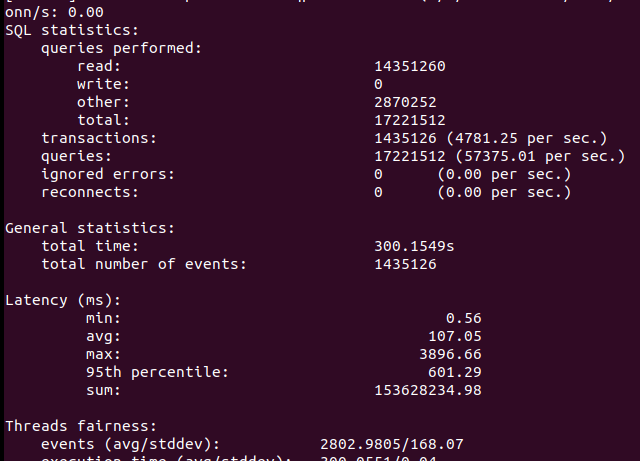
* 1. Cpu utilization while 128 threads - ~100%



* 1. Cpu utilization while 256 threads is still 100% but not crashed



* 1. Cpu utilization while 512 threads is still 100% but not crashed

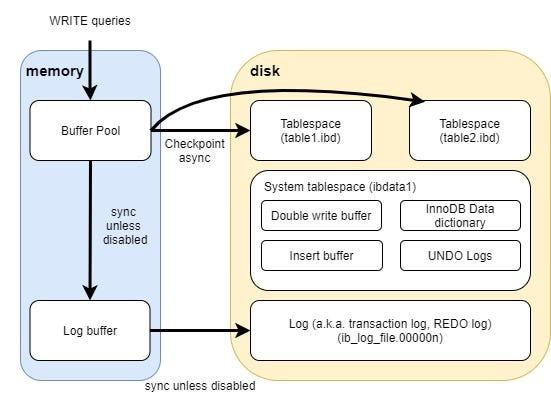
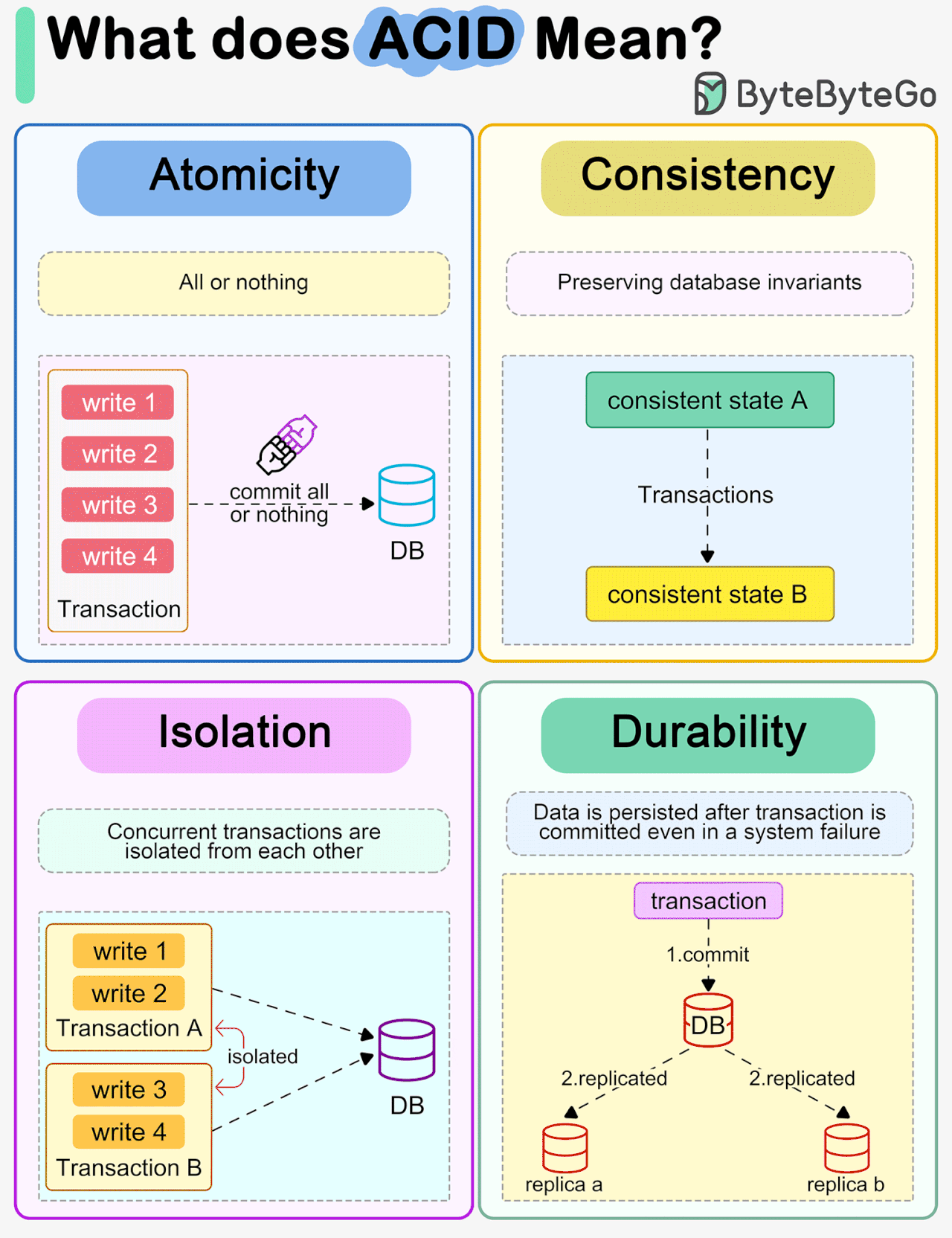


Latency increased by Half of the previous…

* 1. Ok, The system runs maximum of 651 users if it is 652 means the program will through connection failed error…, it is not like mysqlSlap because mysql slap uses small database and simple queries to run, but sysbench is not having the same parameters as mysqlSlap.
  2. The error showing -
     1. FATAL: unable to connect to MySQL server on host '127.0.0.1', port 3306, aborting…
     2. FATAL: error 2004: Can't create TCP/IP socket (24)
  3. This error may be not for the maximux user .
  4. Change the **ulimit -n 2048,**

InnoDB 🌍:

<https://hidetatz.medium.com/how-innodb-writes-data-on-the-disk-1b109a8a8d14>

* A storage engine that makes Database storage possible.
* MySQL uses this storage engine as default.
* This makes the database faster.
* So basically databases consist of 2 parts
  + 1. Server layer
  + 2. Storage layer
* Server layer responsible to connect the storage layer and the client the storage layer is the one who actually handles the data.
* InnoDB follows the ACID compliance
* 
* So basically if we insert in normal database it seams like.

When INSERT, UPDATE, DELETE

* A user executes a query
* The database server opens “data” file for the table
* The database server writes the change to the file
* The database server closes file
* Respond “OK” to the user

When SELECT

* A user executes a query
* The database server opens “data” file for the table
* The database server reads the file and gathers the result which should be returned
* The database server closes file
* While introducing cache on memory

When INSERT, UPDATE, DELETE

* A user executes a query
* The database server writes the change on the cache on memory
* Respond “OK” to the user
* Asynchronously, another thread writes the change on the file on the disk in the background

When SELECT

* + A user executes a query
  + The database server lookups the data on the cache on memory. If not found, reads the data file on the disk and get the result
  + Return the result to the user
* Others are detailed in the blog check it , it is just awesome.

Real First Task :

Use the benchmark tool , analyze the transactions per second, latency, cpu performance according to the no of threads, and give the concurrency rate where the TPS is not dropping. It gives the suggestion where the system can able to do well, and also give the breaking point.

May 25, 2024

Sysbench -> Loop the concurrency in file -> from the file , get the latency, transactions per second, cpu performance (sar cmd used to get the cpu performance over a time kill it after needed) -> give the dat to awk program to filter out the values.

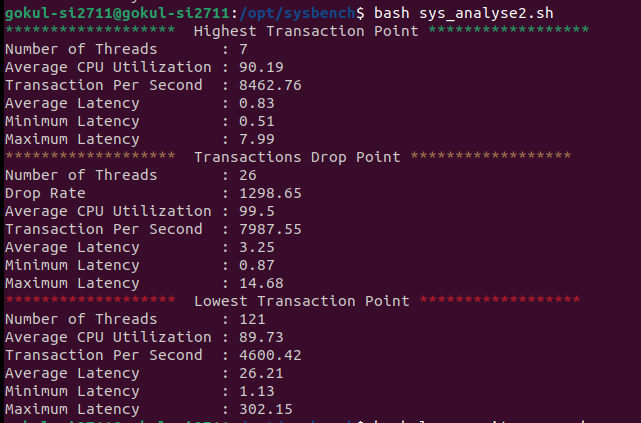
AWK - <https://www.geeksforgeeks.org/awk-command-unixlinux-examples/>

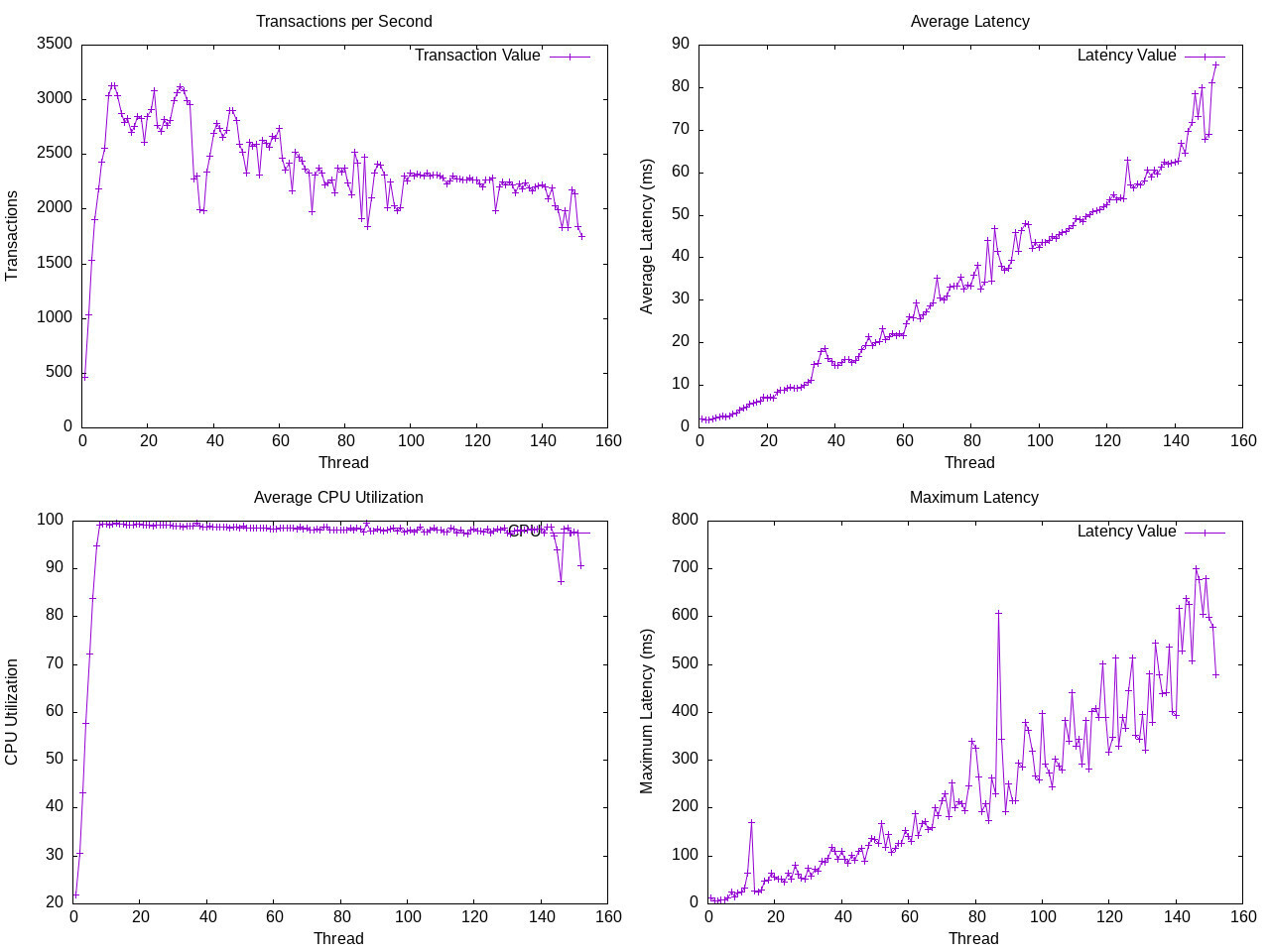
May 27, 2024

Almost completed the task given - to do bench mark for possible no of threads till the default value of connections in mysql program.

Learnt some new bash commands

1. Can able to assign a output of an external command to a bash variable
   1. Ex: var=$(date)
2. Can assign awk output to an array
   1. Ex: declare -a thread=($(awk '/Threads/ {print $3}' monitor\_log))
3. For loop in bash <https://www.geeksforgeeks.org/bash-scripting-for-loop/>
4. Gnu plot - a command to do graphs using numbers like “1 100 2 259 3 300”
   1. X values - 1 2 3
   2. Y values - 100 259 300
5. To delete something in a variable or value
   1. declare -a transactions=($(awk '/transactions/ {print $4}' monitor\_log |tr -d '()'))
   2. Tr -d ‘()’ [ (123) -> 123 ]
6. To merge images vertically and horizontally
   1. convert +append transactions.jpeg average\_latency.jpeg horizontal1.jpeg
   2. convert -append horizontal1.jpeg horizontal2.jpeg result.jpeg
7. To show image
   1. eog result.jpeg
8. Files are in /opt/sysbench/
   1. loop\_monitor\_sys.sh
   2. Sys\_analysis2.sh





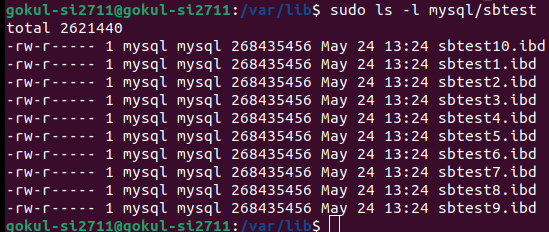
Discussed with sudalai sir.

Need to make some changes and add some.

1. What are we testing is not described in the report
2. Database size ?\*
3. buffer\_pool\_size=?\*
4. Can we use some dedicated cores for the sysbench program only?\*
5. Sleep between tests like 10 sec\*
6. Make steps to avoid long time\*
7. Make some gui
8. Range\_select ? -> understand the command deeply

May 28, 2024

1. **What is the database size used by sysbench - sbtest( db name )**



1. So the database have 10 tables each tables have 1 million records
2. All the tables are in the same size
3. 268435456 == 256MiB
4. Total size of the Database(sbtest) == 2560MiB == 2.5GiB
5. Sbtest10.ibd -> ibd?
   1. It is the mysql table file created by innoDB storage engine.
6. **Buffer\_pool\_size** - default 128MiB
   1. While go to the conf file /etc/mysql/mysql.conf.d/mysqld.conf
   2. There is no key like buffer\_pool\_size , but there is key\_buffer\_size
   3. By checking the size using
      1. **SHOW VARIABLES LIKE 'innodb\_buffer\_pool\_size';**
      2. +-------------------------+-----------+
      3. | Variable\_name | Value |
      4. +-------------------------+-----------+
      5. | innodb\_buffer\_pool\_size | 134217728 |
      6. +-------------------------+-----------+
   4. 134217728 bytes == 128 MB
   5. We can change it directly using
      1. **SET GLOBAL innodb\_buffer\_pool\_size = 2147483648;**
   6. Or in the conf file /etc/mysql/mysql.conf.d/mysqld.conf
      1. innodb\_buffer\_pool\_size = 1G
      2. By adding the above under [mysqld] section
7. **Can we able to dedicate specific core to the sysbench**
   1. Yes, that concept is not specified to sysbench
   2. But by using takeset command we can achieve this
   3. taskset -c 2,3 sysbench …
   4. It means dedicate core no 2 and 3 to sysbench program
   5. NOT WORKING
      1. Expected reason
      2. Sysbench itself might spawn multiple threads that are not respecting the CPU affinity settings, especially if it has been configured to manage threads independently of the parent process's CPU affinity.
8. **Understand the command**

### **events**

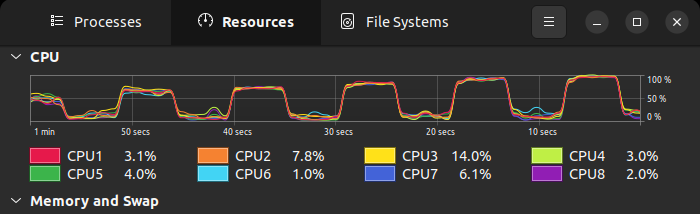
* + **Definition**: The --events parameter in Sysbench specifies the total number of events (transactions or queries) to be executed during the benchmark test.
  + **Usage**: In your command, --events=0 means that the test will run until the specified --time (5 seconds in this case) elapses, rather than stopping after a certain number of events.
  + **Example**: --events=10000 would mean that the benchmark should execute 10,000 events and then stop.

### **range\_selects**

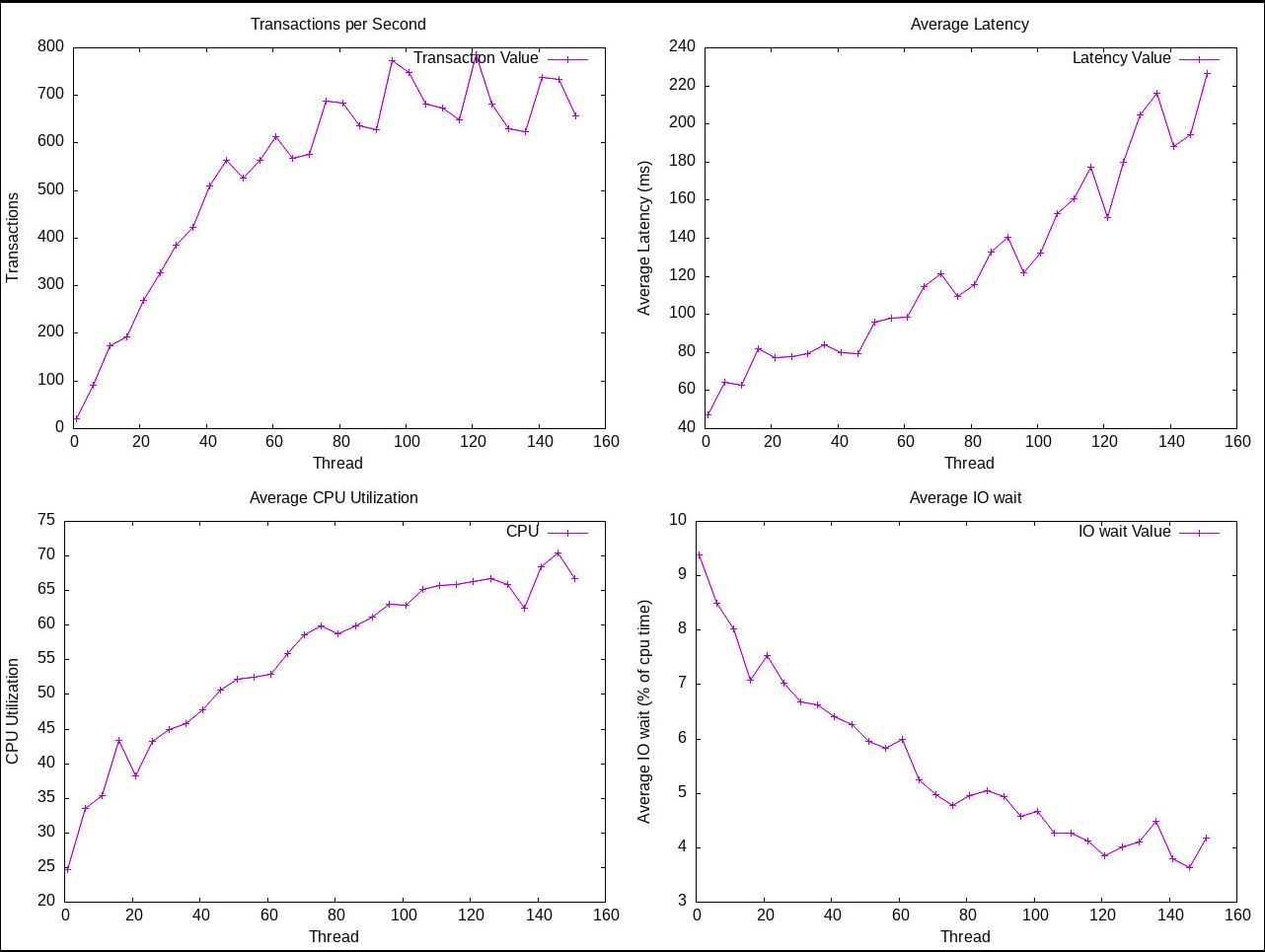
* **Definition**: The --range\_selects parameter controls whether Sysbench will perform range select queries during the OLTP benchmark.
* **Usage**: In your command, --range\_selects=off disables range select queries, meaning Sysbench will not perform these types of queries during the test.
* **Example**: --range\_selects=on would enable range select queries, allowing Sysbench to include these in its benchmark tests.

### **db-ps-mode**

* **Definition**: The --db-ps-mode parameter specifies whether to use prepared statements in the database operations during the benchmark.
* **Usage**: In your command, --db-ps-mode=disable disables the use of prepared statements, meaning Sysbench will not use prepared statements for database operations during the test.
* **Options**:
  + disable: Prepared statements are not used.
  + auto: Prepared statements are used if the database driver supports them.
  + force: Prepared statements are always used.
* **Example**: --db-ps-mode=auto would enable prepared statements if supported by the MySQL driver being used.
  + 1. What is prepared statements
       1. Prepared statements is the ones who can be used/reused when needed
  1. So what is the jargon in (r/w/o: 28329.81/0.00/5664.36) in the output
     1. It is nothing but r - read, w- write , o - other transactions like commit rollback, other than read and write
  2. **Strange observation**
     1. While using **read\_only**.lua from oltp it takes a **vigorous amount of cpu**, but while using **read\_write** operation , it uses **very less amount of cpu** ?
     2. May be read only operation is a direct operation that may uses cache to get the readily available data, it may take some more cpu
     3. May be while doing read and write operations, we can able to say a thing that while writing to the disk if the I/O wait(idle and waiting for a process to complete) time increased the CPU utilization may be decreased due to the waiting time, need to check
     4. **Read\_only**.lua
        1. Even though the threads is like 8 , the CPU goes like 100%
        2. Running 500 threads and monitoring IO wait time using sar -u
           1. So basically the average io write time is 0.20 totally
           2. Average CPU Utilization: 94.42 %
           3. Average IO wait :0.93 %
           4. Average Idle :0.00 %
           5. transactions per sec (1598.53)
           6. average latency (ms): 309.78
     5. **Read\_write**.lua
        1. 500 threads
           1. Average CPU Utilization: 86.71 %
           2. Average IO wait :1.37 %
           3. Average Idle :0.00 %
           4. transactions per sec (320.80)
           5. average latency (ms): 1503.06
     6. **write\_only**.lua
        1. While doing write only even though it is like 500 threads, the average CPU utilization is like 75%
           1. Average CPU Utilization: 78.8 %
           2. Average IO wait :2.97 %
           3. Average Idle :0.00 %
           4. transactions per sec (394.40)
           5. average latency (ms): 1216.74
     7. So from the observation of io wait , the CPU to wait more for I/O operations to complete, so while writing to the disk, cpu utilization decreases.
     8. It is calculated in percentage , the percentage represents the proportion of time the CPU is idle while waiting for I/O operations (disk, network, etc.) to complete

1. Some strange stores the binlog.0000 like files
   1. It is not strange actually it is called binlog
   2. The size vary from 100 bytes to 100 MiB
   3. It stores the record of all the changes made to mysql database
   4. Is it secure to remove binlog files
      1. Yes as long as the data stored in the slave node it is safe.
2. Applying Sleep to the Sysbench after bench mark 5 sec
   1. (read\_only.lua)
   2. 
3. Write only -> 1 to 152 threads , step 5
   1. Observation, while cpu threads increases cpu usage increases and io wait decreases, transaction increases

Changing the innoDB\_buffer\_pool\_size , checking for 500 threads



1. **Changing the** **innoDB\_buffer\_pool\_size**
   1. **read\_only.lua** 
      1. Buffer = 256 MiB, threads = 8, time 10 sec
      2. Restarting the mysql server, but not starting as expected -> just a stupid bug, go to the/var/log/mysql/error.log , it has every error log, it says - - name in the conf file, it is just a typo in the file. (45 min)
         1. Average CPU Utilization: 96.59 %
         2. Average IO wait :**1.02** %
         3. Average Idle :0.00 %
         4. transactions per sec (2859.06)
         5. minimum latency (ms): 1.71
         6. maximum latency (ms): 21.49
         7. average latency (ms): 2.79
         8. total latency (ms): 79924.62
      3. Buffer = 4096 MiB, threads = 8, time 10 sec
         1. Average CPU Utilization: 99.64 %
         2. Average IO wait :**0.01** %
         3. Average Idle :0.00 %
         4. transactions per sec (2239.87)
         5. minimum latency (ms): 1.72
         6. maximum latency (ms): 45.63
         7. average latency (ms): 3.56
         8. total latency (ms): 79951.68
   2. **Read\_write.lua**
      1. Buffer = 256 MiB, threads = 8, time 10 sec
         1. Average CPU Utilization: 37.12 %
         2. Average IO wait :**7.75** %
         3. Average Idle :0.00 %
         4. transactions per sec (310.86)
         5. minimum latency (ms): 8.44
         6. maximum latency (ms): 190.65
         7. average latency (ms): 25.63
         8. total latency (ms): 80142.37
      2. Buffer = 4096 MiB, threads = 8, time 10 sec
         1. Average CPU Utilization: 63.43 %
         2. Average IO wait :**4.96** %
         3. Average Idle :0.00 %
         4. transactions per sec (334.84)
         5. minimum latency (ms): 9.58
         6. maximum latency (ms): 161.88
         7. average latency (ms): 23.82
         8. total latency (ms): 80324.42
   3. **Write\_only.lua**
      1. Buffer = 256 MiB, threads = 8, time 10 sec
         1. Average CPU Utilization: 31.94 %
         2. Average IO wait :**8.29** %
         3. Average Idle :0.00 %
         4. transactions per sec (526.34)
         5. minimum latency (ms): 7.06
         6. maximum latency (ms): 101.84
         7. average latency (ms): 15.18
         8. total latency (ms): 80079.13
      2. Buffer = 4096 MiB, threads = 8, time 10 sec
         1. Average CPU Utilization: 37.7 %
         2. Average IO wait :**7.88** %
         3. Average Idle :0.00 %
         4. transactions per sec (469.07)
         5. minimum latency (ms): 7.62
         6. maximum latency (ms): 158.52
         7. average latency (ms): 17.03
         8. total latency (ms): 79985.29
   4. OBSERVATIONs of innoDB buffer pool size changing
      1. The innoDB buffer pool helps to get the data faster
      2. As the results says, the IO wait time, is reduced while the buffer pool size increases
      3. Read only operation is utilizing the buffer pool to get the data without go to the disk to read
      4. Read-write operation as the write operation depends on the disk writing so the write operation takes IO wait time
      5. Write-only operation is purely based on disk write so there is no big change in IO write.
   5. If the buffer pool increases the IO wait decreases (especially reading operations)
   6. If the write operation increases, IO wait increases and then cpu utilization decreases due to the waiting status to the CPU.
2. **Does the write operation really make the CPU wait?**
   1. Some behavior of innoDB write operation
      1. **Transaction Log:**
         1. Each write operation(insert,update,delete) must be logged to transaction log to maintain durability, this includes not only write in RAM but also in the disk also.
         2. The Log files will be essential to redo some operation, so it needs to be flushed into disk , in a time like commits.
      2. **Doublewrite Buffer**:
         1. While the innoDB ensures the durability of the data, it writes the data in an area called doublewrite buffer, and then it will flush to the final destination on the disk, this flush takes additional disk I/O
      3. **Checkpointing**:
         1. As the innoDB directly write the data to buffer pool first, and then in some checkpoints the data from the buffer pool should be write into disk, here large amount of data needs to be flushed, so it can take higher disk I/O and make the CPU wait.
   2. Conclusion of the analysis of the I/O wait for write operation
      1. Due to the innoDB needs to write the data to the disk from the buffer pool and log buffer, the CPU has to wait for the write so the I/O time increases.
      2. So what is the purpose of using buffer pool?
         1. It is meant for reduce the latency of the transaction from the client
         2. In the client end the response time will be optimized while the data is writing in the buffer pool, but after some time, the buffer pool needs to write in the disk which is in the server side, so CPU wait is concern for the only writing in the disk

May 29, 2024

**Checking if we can reduce the IO wait**

These are some of the parameters which may optimize the IO wait time.

1. **Innodb\_flush\_log\_at\_trx\_commit -**  it determines the frequency at which the log buffer data writes into the disk
   1. **Value 0**: the logs are flushed from the log buffer to log file for every second, which not having that much durability of the data
   2. **Value 1 (default)**: the log buffer flushed to the log file and the disk at every transaction commit, which gives more durability
   3. **Value 2:**  the log buffer is written to the log file at each commit , but flushed to the disk every second,still having some guarantee about durability.

**Using Write\_only test**

* 1. Buffer = 4096 MiB, threads = 8, time 10 sec , **innodb\_flush\_log\_at\_trx\_commit=1(default)**
     1. Average CPU Utilization: 26.8 %
     2. Average IO wait :**9.27 %**
     3. Average Idle :0.00 %
     4. transactions per sec **(458.29)**
     5. minimum latency (ms): 7.77
     6. maximum latency (ms): 148.74
     7. average latency (ms): 17.44
     8. total latency (ms): 80170.21
  2. Buffer = 4096 MiB, threads = 8, time 10 sec , **innodb\_flush\_log\_at\_trx\_commit=2**
     1. Average CPU Utilization: 32.19 %
     2. Average IO wait :**9.09 %**
     3. Average Idle :0.00 %
     4. transactions per sec **(620.67)**
     5. minimum latency (ms): 3.75
     6. maximum latency (ms): 221.46
     7. average latency (ms): 12.83
     8. total latency (ms): 80680.66
  3. Buffer = 4096 MiB, threads = 8, time 10 sec , **innodb\_flush\_log\_at\_trx\_commit=0**
     1. Average CPU Utilization: 32.43 %
     2. Average IO wait :**8.86 %**
     3. Average Idle :0.00 %
     4. transactions per sec **(677.78)**
     5. minimum latency (ms): 3.86
     6. maximum latency (ms): 192.92
     7. average latency (ms): 11.79
     8. total latency (ms): 80045.23
  4. By the above observation the **innodb\_flush\_log\_at\_trx\_commit** parameter does not change the performance of IO that much.
  5. But the transactions per second increase
     1. In the default value , every transaction commit need to flushes the value
     2. But in the value is 0, it makes multiple transactions into a batch and leads to a single flush, making the IO wait less and also , with fewer I/O operations, the system can handle more transactions per second, leading to higher throughput.
     3. NOTE: this is not a safe settings, due to if the server crashes in the last second of a batch needs two flushes the data, a multiple transaction data could be lost.

1. **Innodb\_log\_file\_size**
   1. It is the one where all the transactions are logged, Larger log files reduce the frequency of checkpoints, which may reduce the IO time
   2. Buffer = 4096 MiB, threads = 8, time 10 sec ,

**innodb\_log\_file\_size = 48MiB(default)**

* + 1. Average CPU Utilization: 29.96 %
    2. Average IO wait :**8.47** %
    3. Average Idle :0.00 %
    4. transactions per sec (567.09)
    5. minimum latency (ms): 6.26
    6. maximum latency (ms): 158.99
    7. average latency (ms): 14.09
    8. total latency (ms): 80027.19
  1. Buffer = 4096 MiB, threads = 8, time 10 sec ,

**innodb\_log\_file\_size = 256MiB**

* + 1. Average CPU Utilization: 26.65 %
    2. Average IO wait :**8.53** %
    3. Average Idle :0.00 %
    4. transactions per sec (630.89)
    5. minimum latency (ms): 6.46
    6. maximum latency (ms): 103.74
    7. average latency (ms): 12.67
    8. total latency (ms): 80016.21
  1. Buffer = 4096 MiB, threads = 8, time 10 sec ,

**innodb\_log\_file\_size = 1GiB**

* + 1. Average CPU Utilization: 28.61 %
    2. Average IO wait :**8.66** %
    3. Average Idle :0.00 %
    4. transactions per sec (601.85)
    5. minimum latency (ms): 6.32
    6. maximum latency (ms): 121.63
    7. average latency (ms): 13.28
  1. By the above observation, it is shown that changing the log file size is not changing the IO wait time.

1. **Innodb\_io\_capacity**
   1. It controls the number of IO operations per second, by default 200
   2. If it sets too low the background task may not handles the work load and make the IO wait time increase.
   3. If it sets too high, it may overwhelm the storage subsystem and leading to IO contention and reduces the performance of the foreground.
   4. Faster SSDs can handle higher IOPS, so a higher value may be appropriate. Slower HDDs may require a lower setting.
   5. The test running system is having 512GiB SSD.
   6. Buffer = 4096 MiB, threads = 8, time 10 sec ,

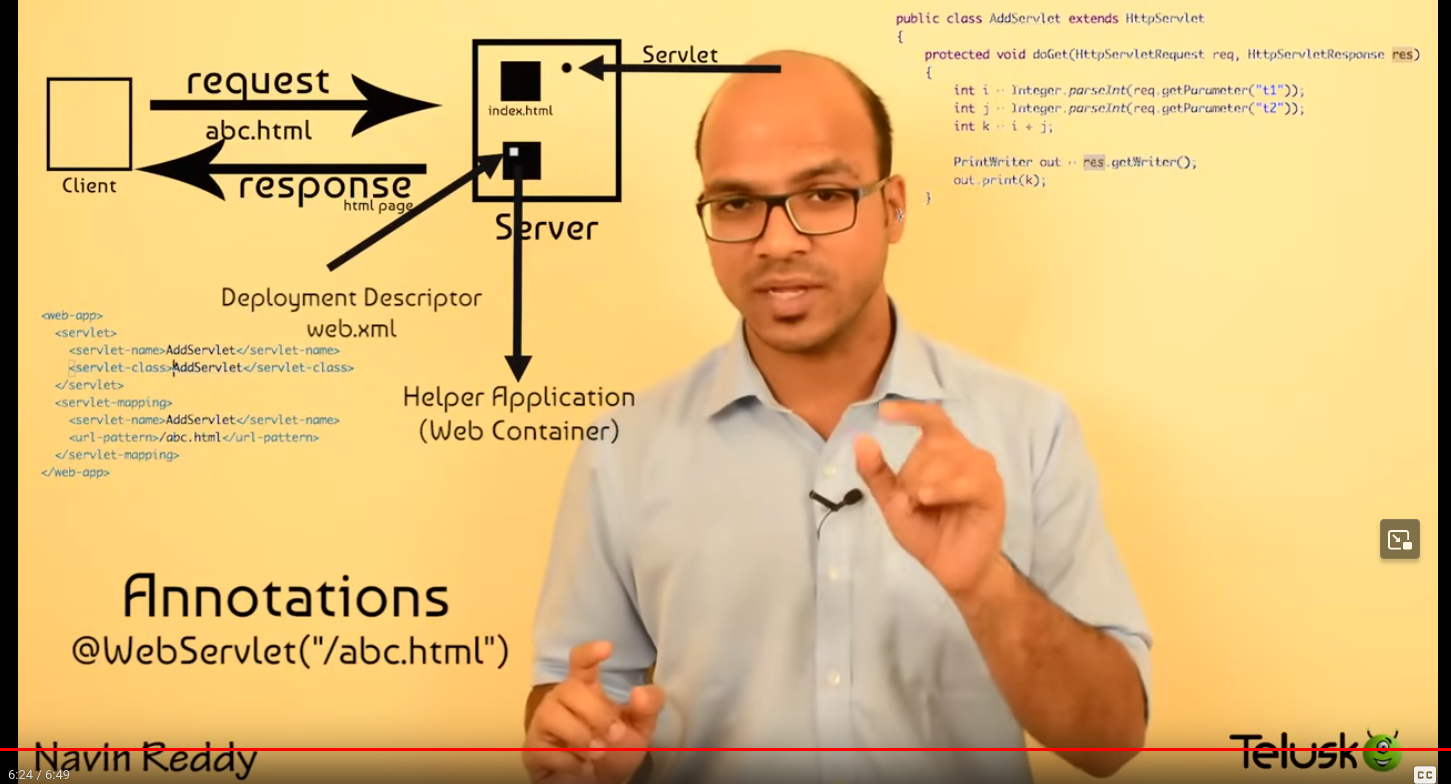
**Innodb\_io\_capacity = 200 (default)**

* + 1. Average CPU Utilization: 26.43 %
    2. Average IO wait :**8.96** %
    3. Average Idle :0.00 %
    4. transactions per sec (460.58)
    5. minimum latency (ms): 6.67
    6. maximum latency (ms): 155.79
    7. average latency (ms): 17.35
    8. total latency (ms): 80087.84
  1. Buffer = 4096 MiB, threads = 8, time 10 sec ,**Innodb\_io\_capacity = 2000**
     1. Average CPU Utilization: 30.73 %
     2. Average IO wait :**8.66** %
     3. Average Idle :0.00 %
     4. transactions per sec (400.49)
     5. minimum latency (ms): 8.91
     6. maximum latency (ms): 163.59
     7. average latency (ms): 19.95
     8. total latency (ms): 80099.28
  2. So from the above observation, we can conclude that , changing the **Innodb\_io\_capacity** value is not changing that much IO in the working system.

**Making UI**

**To make the UI need to have a basic knowledge in jsp and servlet, i need to learn**

**Watching Telusko videos**

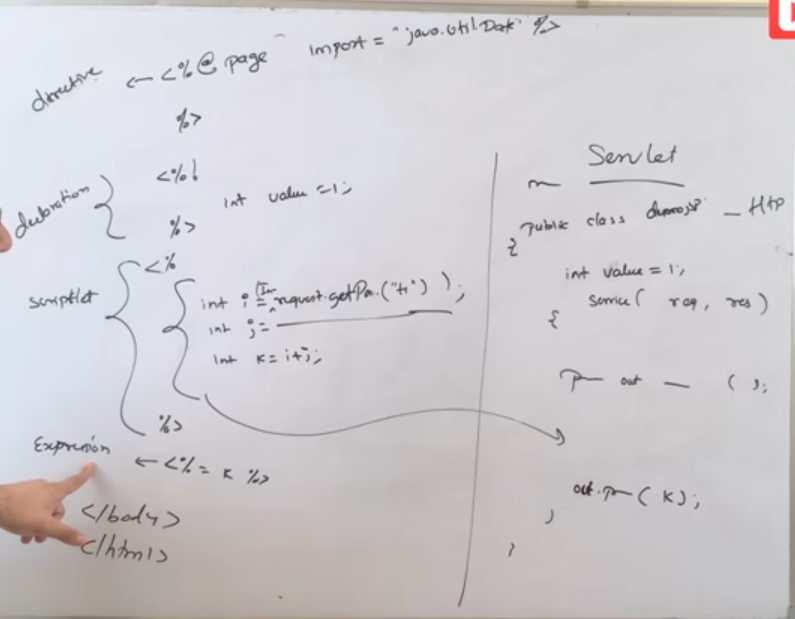
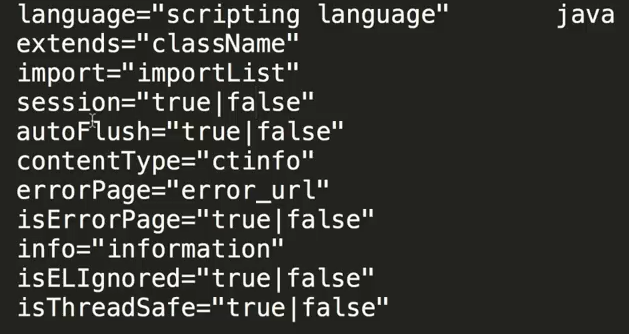
1. [Introduction to Servlets](https://www.youtube.com/watch?v=7TOmdDJc14s&list=PLsyeobzWxl7pUPF2xjjJiG4BKC9x_GY46&index=1)
   1. So basically servlets is used to create connection between front end and the backend like, flask
   2. So firstly user sends request to the server, server sends it to the web container, then the webcontainer search for which servlet need to call for this url in the deployment descriptor inside web.xml
   3. So we can directly give the mapping using Annotations in java without web.xml
   4. Then the request is sent to the mapped servlet.
   5. Then the response, sent by the servlet program as html,json…
   6. 
2. [#1 Servlet and JSP Tutorial | Eclipse Setup and Tour](https://www.youtube.com/watch?v=NrdGCk5Cwls&list=PLsyeobzWxl7pUPF2xjjJiG4BKC9x_GY46&index=2) - just eclipse download
3. [#2 Servlet and JSP Tutorial | Configure Tomcat in Eclipse](https://www.youtube.com/watch?v=R7DlcnrXfYE&list=PLsyeobzWxl7pUPF2xjjJiG4BKC9x_GY46&index=3) - configuring tomcat
4. [#3 Servlet and JSP Tutorial | Eclipse and Tomcat Setup on Windows](https://www.youtube.com/watch?v=9iHKCnxUWqQ&list=PLsyeobzWxl7pUPF2xjjJiG4BKC9x_GY46&index=4) - 🤷 again…
5. [#4 Servlet and JSP Tutorial | Creating Web Project in Eclipse](https://www.youtube.com/watch?v=EkacxN8gx08&list=PLsyeobzWxl7pUPF2xjjJiG4BKC9x_GY46&index=5)
   1. Create a new dynamic web project with web.xml file,
   2. create a html page all the html and jsp page should be inside the webapp folder,
   3. All the core java files needs to be in src folder
   4. right click and run as -> run on server, configure the class, then run.
   5. Run on eclipse or browser , go to window -> web browser -> internal or external.
   6. Do we really need the jsp here, no
   7. Suppose we need to add 2 numbers in the server
      1. Create a form , then action to a path
      2. Get the inputs and submit
      3. Page not found error , because we don’t create the file specified in the path yet will do it later.
6. [#5 Servlet and JSP Tutorial | Create Servlet and web.xml Config](https://www.youtube.com/watch?v=wty6OROO__8&list=PLsyeobzWxl7pUPF2xjjJiG4BKC9x_GY46&index=6)
   1. First create a normal java class , name the package like com.demoapp or anything could be the package name to be unique we are using the reverse of the url
   2. To make the java class into a servlet extend the java class from the httpservlet class.
   3. By default it is not installed the HttpServlet class you have to download the jar and add it to the project
   4. Can download from here [jakarta.servlet : jakarta.servlet-api : 5.0.0](https://repo1.maven.org/maven2/jakarta/servlet/jakarta.servlet-api/5.0.0/)
   5. Note that the servlet version and tomcat version should be matched check the tomcat documentation for correct requirements
   6. Then create a function called service with public and void return type, it is a method inside HttpServlet you have to overwrite this function to run it as a servlet.
   7. Give 2 arguments request and response, request can able to get the value from the form and response can send the value to the client.
   8. req.getParameter to get the value by name of the input field, it by default gives string , we need to parse it to get the respective datatypes. The data is get from the query parameter (?a=1&b=2)
   9. Now get the numbers and add it and print it using println , obviously it will print in the console only, will talk about it later.
   10. To connect the servlet and the page we need to add servlet and servlet-mapping in web.xml
       1. Inside servlet tag -> servlet name and servlet class
       2. Inside servlet mapping tag -> servlet name and url-pattern
   11. Ok now we have to send the response to the client, one of the way is to write the output in the html.
   12. res.getWriter, it will return a printwriter object, so use the object and do println or directly use the method to write it on the web page.
7. [#6 Servlet and JSP Tutorial | Get and Post](https://www.youtube.com/watch?v=_6ISkXSC1js&list=PLsyeobzWxl7pUPF2xjjJiG4BKC9x_GY46&index=7)
   1. HTTP has some methods like get,post,put,delete,options
   2. Get - used to fetch the data
   3. Post - used to insert the data
   4. In servlet by we can use service method or we can use separate methods for the get and post request
   5. doGet method to handle get request and doPost to handle post request.
   6. We can restrict user to send only post, not get, by implementing doPost method
   7. Even service method determines the request type and call doGet or doPost methods internally
      1. Why we need this, in get method you can able to see the variable values in the url , but in the post you cant
      2. A user can easily make the post to get in the url, the service method run it anyway
      3. But by restricting it to doPost method they cant able to do this hack
8. [#7 Servlet and JSP Tutorial | RequestDispatcher | Calling a Servlet from Servlet](https://www.youtube.com/watch?v=EiuKnHNFwRU&list=PLsyeobzWxl7pUPF2xjjJiG4BKC9x_GY46&index=9)
   1. To call a servlet from another servlet we can use 2 ways
      1. Request dispatcher
      2. Redirecting
   2. Request dispatcher an interface so , we need to get a class using req.getRequestDispatcher by passing the url path of the servlet
   3. First create a servlet class and add it paths to web.xml
   4. Then forward req and res using forward method.
   5. Now we need to pass the data to the servlet one way is using sessions, but later, another way is adding attributes to the request sent to the servlet.
   6. req.setAttribute("k", k);
9. [HttpServletRequest and HttpServletResponse Theory](https://www.youtube.com/watch?v=GbF_nBLHP6A&list=PLsyeobzWxl7pUPF2xjjJiG4BKC9x_GY46&index=10)
   1. Actually these 2 are interfaces and the object is created by the tomcat server and given to us
   2. This is the objects which is responsible for the request and response to the client.
10. [#8 Servlet and JSP Tutorial | RequestDispatcher and sendRedirect Theory](https://www.youtube.com/watch?v=kYzyXWmh37A&list=PLsyeobzWxl7pUPF2xjjJiG4BKC9x_GY46&index=11)
    1. As the picture below defines better
       1. Calling one servlet from another servlet using requestDispatcher, is directly by the servlet, and it uses the same req and res objects, and the user don't know what is happening because the url is not getting changed
       2. In the sendRedirect, if we want to call the servlet we need to send another request from the user, so totally 2 request, this will notify the user that the page is getting redirected.
    2. 
11. [#9 Servlet and JSP Tutorial | sendRedirect | URL Rewriting](https://www.youtube.com/watch?v=2HurrgxlTmg&list=PLsyeobzWxl7pUPF2xjjJiG4BKC9x_GY46&index=12)
    1. Ok by using res.sendRedirect function we can send the client a response telling to redirect to this url.
    2. We can able to send the data using session or cookies
    3. Another way is url rewriting from the parameter of sendRedirect we can able to send url parameters like sendRedirect(“?k=”+k);
12. [#10 Servlet and JSP Tutorial | HttpSession | Cookie](https://www.youtube.com/watch?v=5tLGwdyPGRY&list=PLsyeobzWxl7pUPF2xjjJiG4BKC9x_GY46&index=13)
    1. To send a servlet a value we can use sessions or cookies
    2. Sessions are the ones who stored in the server and have the value for some time
    3. Session is also given by tomcat server, we can use the variable by req.getSession and store it in a class called HttpSession.
    4. For the session variable we can set the attributes by setAttribute method.
    5. And we can get the value in another servlet by using another session variable. sess.getAttribute(“k”);
    6. Cookies it is the one in which server send it with the response and user add some data in it and send back to the server, it is like a token to the vegetable shop.
    7. Ok to do that we need to create a cookie object with the name and value, and send it to the user with the response. Cookie is a class with constructor of 2 attributes that is string type of name and value. And res.addCookie(object\_cookie);
    8. After getting that response, we are redirecting to another page, which the user sends all of the cookies it has as an array, we need to filter it out and get the value. Using the name of the cookie
13. Will continue….. On May 30, 2024

Making UI

1. Get the details required to connect in a database from a user.
   1. Create an html page to get the values such as
      1. IP address
      2. Port no of the server
      3. UserName
      4. Password
   2. I used a new nested css property it will be fun
2. Show the default configuration variable to the user.

May 30, 2024

Learning JSP continuations

1. [ServletConfig and ServletContext](https://www.youtube.com/watch?v=uKoBbSp0J3Y&list=PLsyeobzWxl7pUPF2xjjJiG4BKC9x_GY46&index=14)
   1. We can have some initial values for servlet like name value pair
   2. We can assign them inside web.xml, like inside <context-param>
      1. <param-name>
      2. <param-value>
   3. We can able to get the value using both servlet config and context, both are interfaces
   4. Servletcontext instance can be get by getServletContext a method inside HttpServlet, so from this object we can get the initParam using the name
   5. So then why we need ServletConfig, actually we are defining values outside the servlet tag which is common for all of the servlets, suppose we need to have different values for different servlets means, inside the servlet
      1. Inside <init-param>, have same thing like param name and param value
      2. Then make an instance for he ServletConfig to get the individual param of the servlet
2. [Servlet and JSP Tutorial | Servlet Annotation Configuration](https://www.youtube.com/watch?v=TMlXF1iW9rE&list=PLsyeobzWxl7pUPF2xjjJiG4BKC9x_GY46&index=15)
   1. Instead of web.xml mapping we can use annotations which makes our life easier
   2. @Webservlet(/path) above the servlet class thats it we dont need a web.xml at all to tell the server to map.
3. [Servlet and JSP Tutorial | Why JSP?](https://www.youtube.com/watch?v=5Sz8z1Yreow&list=PLsyeobzWxl7pUPF2xjjJiG4BKC9x_GY46&index=16)
   1. JSP - Java Server Pages
   2. In Servlet we can able to write html inside out.println file it will print the html code inside the web page, but we cant able to write the entire page inside a function
   3. To make it easy only we are using JSP which we can able to write java program inside HTML using the tag <% – Java Codes – %>
   4. One thing is we can directly say action=”file.jsp” no need for annotations
   5. To use the request response object , by default available in jsp also the ‘out’ is available. These objects are called implicit objects in jsp
4. [How JSP translated into Servlets?](https://www.youtube.com/watch?v=8M5fuIuhRHA&list=PLsyeobzWxl7pUPF2xjjJiG4BKC9x_GY46&index=17)
   1. So JSP can be easy why do we need servlets then?
   2. As we say before tomcat is a servlet container , it will under servlet but not jsp,
   3. So how then the jsp works, it will converted to servlet
      1. Name of the servlet -> it will be name of the file name
         1. Ex: add.jsp -> add\_jsp(classname)
      2. The class automatically extends the HttpServlet and create implicit objects like request,response, out, session,etc
      3. Anything that we write inside <% %> tag it will go inside the ‘service’ method
      4. Anything we write inside <%**!** %> which tells us not service, that means outside the service method but inside the class
      5. We need to import packages also so we have a tag like
         1. <%@page import = “java.util.Date” %>
         2. More than one class can be given as comma separated
      6. And also we have one more tag <%= %> anything inside it will goes to the out.print method
   4. Tags in JSP
      1. <% %> -> scriptlet
      2. <%! %> -> declaration
      3. <%@ page import=”” %> -> directive
      4. <%= %> -> expression
   5. 
5. [JSP to Servlet Conversion | Netbeans](https://www.youtube.com/watch?v=ji4SJYxv6GM&list=PLsyeobzWxl7pUPF2xjjJiG4BKC9x_GY46&index=18)
   1. We can able to see the JSP to servlet conversion in netbeans
   2. All the html tags we wrote it will converted to out.write(“”) ,each and every line of the tag will converted.
   3. Install netbeans, create a web application, create a jsp and right click to see the servlet of the JSP.
   4. 
6. [JSP Tags | Scriptlet | Declaration | Directive | Expression](https://www.youtube.com/watch?v=NQ7xKNULkTk&list=PLsyeobzWxl7pUPF2xjjJiG4BKC9x_GY46&index=19)
   1. Scriptlet
      1. The code we are writing inside <%%> tag will go into the service method
   2. Declaration
      1. To declare any variable outside the service method use <%! %>
   3. Directive
      1. To import anything use <%@page import =””%>
   4. Expression
      1. To print anything instead of using <% out.print%> use <%= expression%>
7. [JSP Directive | Page | Include | Taglib](https://www.youtube.com/watch?v=LlVcQhKGVbU&list=PLsyeobzWxl7pUPF2xjjJiG4BKC9x_GY46&index=20)
   1. Java Directives
      1. Page
         1. Syntax - <%@page attribute=”value” attribute=”value”%>
         2. 
         3. There are no of attributes and values we can use in @page directive
      2. Include - we can include another jsp or html files also
         1. Syntax -> <%@include file=”file\_name”%>
      3. Taglib - is to use external tags from other frameworks like springboot will see it later.
8. Will continue…

Making UI May 30, 2024 11:40 AM

1. Get the details required to connect in a database from a user.
   1. Create an html page to get the values such as
      1. IP address
      2. Port no of the server
      3. UserName
      4. Password
   2. I used a new nested css property it will be fun
   3. Login page completed
   4. Making JDBC connection and run the configuration queries
   5. While making JDBC make sure that jar for sql is inside the webapp>web-inf>lib
2. Show the default configuration variable to the user.
   1. Less configs in the top and all others needed?
3. Integrate the cli code with java using process builder, common error code 126, it is because permission denied to execute chmod +x cmd.sh which gives all the users execute permission

—------------------

May 30, 2024 3:23 PM

Talked with atthi sir,

1. He asked about what are you doing
2. I say benchmarking tool and stuff
3. He asked why sysbench instead of hammerDB
4. I said hammerDB gives only 2 parameters, sysbench gives me more like latency i say
5. He said like yes, if you are changing the configuration of the mysql in an application you need minute details so sysbench gives that , if you are changing just a harddisk in your system and want to check how many transactions that the harddisk handles that time go for hammerDB, after that he say good work.

—----------------

May 31, 2024

—-------------

May 31, 2024 10:31 AM

Discussed with preethi akka.

1. Asked about the process
2. Asked about the schema of the tables created
3. To see the schema of the table “SHOW CREATE TABLE table\_name;”
4. Then can we able to give our custom schema to the sysbench?
5. How many connections are made? I said for each thread each connections created
6. To view that there is a command “**SHOW FULL PROCESSLIST;**”

—--------------

1. **Can we able to give our custom schema inside sysbench?**
   1. **Yes,**  we can able to create our custom data that is need to be tested, by creating a .lua script.
   2. By watching the oltp\_common.lua script we can able to see that sysbench is created this files in the path /usr/share/sysbench/
   3. In that file they are creating a simple query to create a simple schema, we can create like this using that logic.

Testing the benchmarking tool locally and remotely

1. Remotely - connected to local network , speed ~2 to 5 MiBs
   1. **Readonly**, tables=10, records = 1million , time = 5sec,threads = 150, loop = off
      1. Average CPU Utilization: **45.95** %
      2. Average IO wait :0.05 %
      3. Average Idle :0.00 %
      4. transactions per sec (**457.46**)
      5. average latency (ms): **316.01**
   2. **Read write**
      1. Average CPU Utilization: **38.25** %
      2. Average IO wait :0.00 %
      3. Average Idle :0.00 %
      4. transactions per sec (**256.90**)
      5. average latency (ms): 553.08
   3. **Write only**
      1. Average CPU Utilization: **34.66** %
      2. Average IO wait :0.00 %
      3. Average Idle :0.00 %
      4. transactions per sec (**274.52**)
      5. average latency (ms): 313.44
2. Locally
   1. **Readonly**, tables=10, records = 1million , time = 5sec,threads = 150, loop = off
      1. Average CPU Utilization: **99.02** %
      2. Average IO wait :0.00 %
      3. Average Idle :0.00 %
      4. transactions per sec (**1690.77**)
      5. average latency (ms): **87.84**
   2. **Read write**
      1. Average CPU Utilization: **85.61** %
      2. Average IO wait :2.16 %
      3. Average Idle :0.00 %
      4. transactions per sec (**410.69**)
      5. average latency (ms): 352.27
   3. **Write only**
      1. Average CPU Utilization: **84.14** %
      2. Average IO wait :2.12 %
      3. Average Idle :0.00 %
      4. transactions per sec (**625.32**)
      5. average latency (ms): 233.26

Learning JSP continuation…

1. [Implicit Objects in JSP](https://www.youtube.com/watch?v=mhrDVzj8pQM&list=PLsyeobzWxl7pUPF2xjjJiG4BKC9x_GY46&index=21)
   1. Builtin Objects in JSP
      1. request(HttpServletRequest)
      2. respond(HttpServletResponse)
      3. pageContext(PageContext)
         1. It is like req.setAttribute, but it’s scope is by default with in the page. Ex: pageContext.setAttribute(“name”,”naveen”,PageContextx.SESSION\_SCOPE);
         2. We can also change the scope of the object to session or request, by adding a third parameter like above
      4. out(PrintWriter)
      5. session(HttpSession)
      6. application(ServletContext)
      7. config (ServletConfig)
2. [Exception Handling in JSP](https://www.youtube.com/watch?v=0LvRQmLX5eQ&list=PLsyeobzWxl7pUPF2xjjJiG4BKC9x_GY46&index=22)
   1. We can do all the things that we do in core java to handle the eros
   2. We can also have a dedicated page to show the error message
   3. In error occurring page add
      1. <%@ page errorPage=”error.jsp”%>
   4. In the error.jsp
      1. It will have the “exception” object implicitly so we can use that, only if we add a directive call isErrorPage=’’true’, when we enable this only we can able to get the “exception” object
3. [Exception Handling in JSP](https://www.youtube.com/watch?v=0LvRQmLX5eQ&list=PLsyeobzWxl7pUPF2xjjJiG4BKC9x_GY46&index=22)