**HW 10: Performance Testing**

**Objectives:**

Complete performance test and analysis of a Stevens web page (you can choose any Stevens webpage that you prefer) using Apache JMeter for this assignment.

**Assignment:**

Part 1 – Create simple workload model

Part 2 - Evaluate and generate report of performance testing

**Author:** Prateek Singh Chauhan

**Pledge:** “I pledge on my honor that I have not given or received any unauthorized assistance on this assignment/examination. I further pledge that I have not copied any material from a book, article, the Internet, or any other source except where I have expressly cited the source.”

**Application under test:** Stevens Institute of technology website.

**Tool used:** Apache J meter

**Workload model:**

**Stevens Institute of Technology website has 300 users load during normal hours and all the users browse Apply for Spring 2023 admissions information.**

* **Test Script Transactions**: Home Page, Apply
* **No. of users:** 300
* **No. of Iterations (assuming user will go to same page twice):** 2
* **Ramp Up Period (the amount of time it will take Apache JMeter™ to add all test users (threads) to a test execution):** 5 seconds
* **Think time:** 24ms (Assumed)

**J meter setup:**

Graphical user interface, text, application

Description automatically generated

Created a Thread group with config mentioned in workload model as:

* No. of Thread: 300
* Ramp up period: 5 seconds
* Loop Count: 2

For the Thread Group, created two Sampler, one for Stevens Home Page and another for Stevens Apply page performance.

* Steven’s homepage configuration:
* Graphical user interface, application

  Description automatically generated
* Stevens Apply Page configuration:
* Graphical user interface, application

  Description automatically generated

Also, Added Cache config element for sampler to clear for each iteration:

Graphical user interface, text, application

Description automatically generated

And Added Cookie config element for sampler to clear cookie for each iterations:

Graphical user interface, application

Description automatically generated

For comparison, we need to get the baseline results with 1 user and number of iterations as 2.

**Performance Results:**

* **Baseline Results: (To compare performance with)**

Config same as above but:

No of users: 1

No. of iterations: 2

Ramp up: 1 sec

**Home Page:**

**Average Response time: 146ms**

**Max Response Time: 208ms**

**Min Response Time: 146ms**

**Graphical user interface, application

Description automatically generated**

**Graphical user interface, application

Description automatically generated**

**Apply page:**

**Average Response time: 218ms**

**Max Response time: 278ms**

**Min Response time: 159ms**

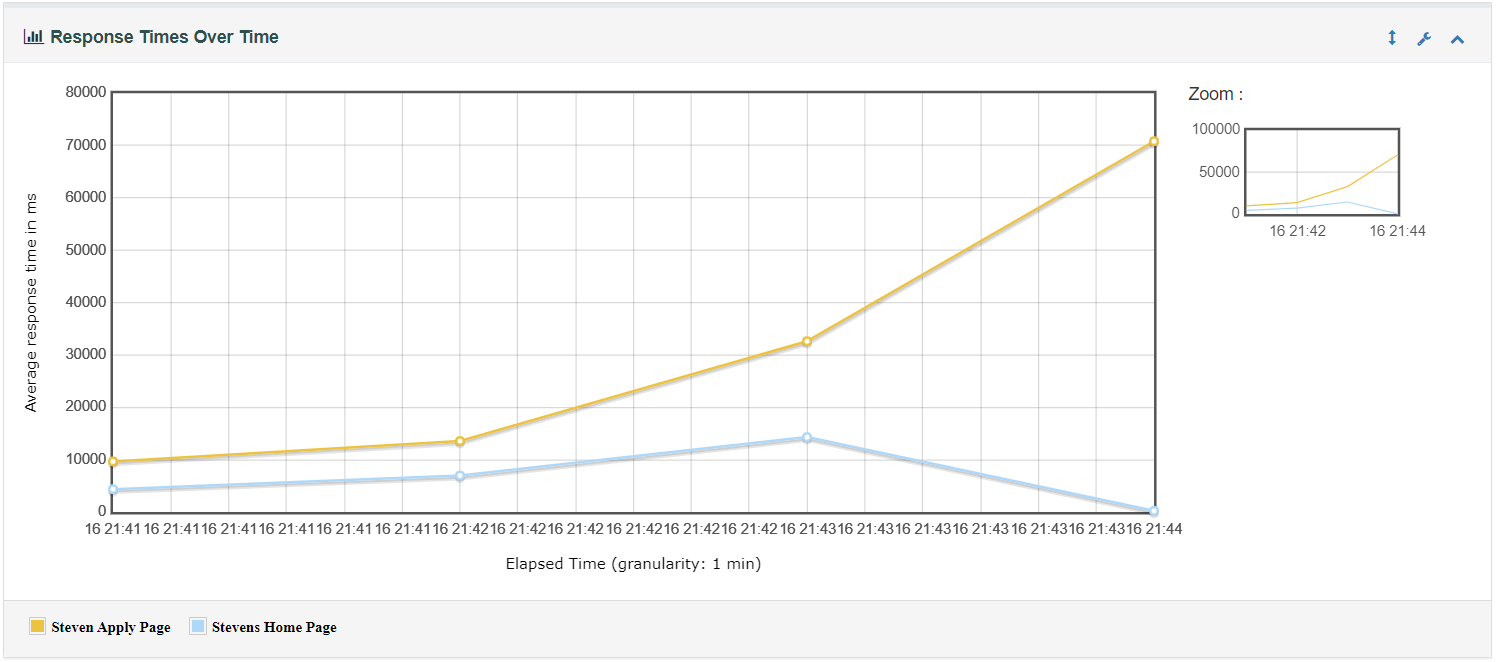
Graphical user interface, application

Description automatically generated

Graphical user interface, application

Description automatically generated

**Response time over time:**

****

**Active threads over time:**

**Chart, line chart

Description automatically generated**

**Response Time Overview:**

**Chart, bar chart

Description automatically generated**

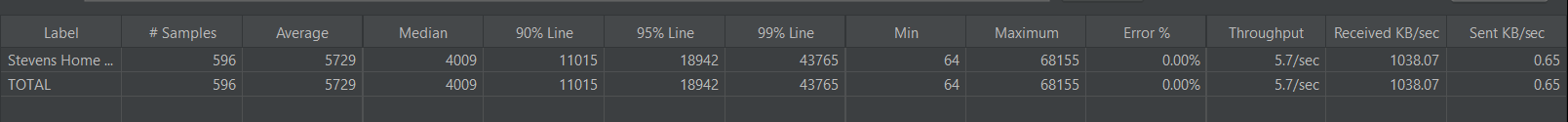
**Stevens Home Page Performance:**

**Average Response Time: 5729 ms**

**Median Response Time: 4009 ms**

**Min Response Time: 64 ms**

**Max Response Time: 68155 ms**

****

When comparing to single user baselined output, the system is much slower when the number of users are high. The expected average response time for the Stevens Home page will be 5729ms or 5.7 sec and in worst case scenario will be 1 min 13 seconds, if the number of users accessing will 300 within 5 seconds, which is very slow as compared to 208ms.

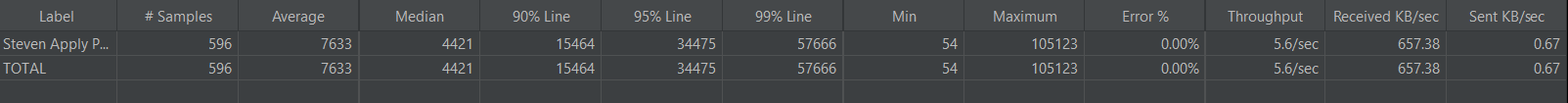
**Stevens Apply Page Performance:**

**Average Response Time: 7633 ms**

**Median Response Time: 4421 ms**

**Min Response Time: 54 ms**

**Max Response Time: 105123 ms**

****

When comparing to single user baselined output, the system is much slower when the number of users are high. The expected average response time for the Stevens Home page will be 4421ms or 4.4 sec and in worst case scenario will be 1 min 45 seconds, if the number of users accessing will 300 within 5 seconds, which is very slow as compared to 278ms or 2.7 sec.

For the above graphs, we understand that the average response time increases with the amount of load applied to system and when compared to based 1 user in system, the system is very slow. We also see that combined both apply and home page requests more than 800 requests took response time more than 1500 ms compared to 110 requests in the range of 500ms to 1500ms