# Comparisons of shortest, longest and average execution time for both 50 and 100 samples

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Web applications | Samples | Shortest(ms) | Longest(ms) | Average(ms) |
| https:\cse.buet.ac.bd\moodle | 50 | 5 | 768 | 110 |
| 100 | 4 | 749 | 121 |
| https:\cse.buet.ac.bd\home | 50 | 4 | 878 | 82 |
| 100 | 4 | 589 | 86 |
| https:\cse.buet.ac.bd\home\notice | 50 | 19 | 2160 | 294 |
| 100 | 17 | 17065 | 1805 |

This table shows the comparisons of the shortest, longest, and average execution times for various web applications with 50 and 100 samples.

# Comparisons of average throughput, average data reception and average data transmission for both 50 and 100 samples

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Web applications | Samples | Avg Throughput  (req/s) | Avg Data reception  (kB/s) | Avg Data transmission  (kB/s) |
| https:\cse.buet.ac.bd\moodle | 50 | 16.6 | 586.28 | 15.08 |
| 100 | 32.9 | 1159.74 | 29.83 |
| https:\cse.buet.ac.bd\home | 50 | 15.8 | 735.82 | 11.76 |
| 100 | 31.5 | 1461.26 | 23.38 |
| https:\cse.buet.ac.bd\home\notice | 50 | 6.3 | 2539.85 | 6.15 |
| 100 | 9.9 | 4020.57 | 9.73 |

The execution times vary significantly between the different applications, and this data helps in understanding their performance characteristics.

Here I have generate the two above tables using the 'Total' row of the 'Summary report' .

### How JMeter Calculates These Averages:

* JMeter computes these metrics based on all threads and samples in the test run.
* The **"Total" row** reflects the cumulative data for all requests across all threads, divided by the total runtime.

From the aggregation report , moodle-50 users-1700 samples total , moodle-100 users-3400 samples total, home-50 users-1600 samples total , home-100 users-3200 samples total , notice-50 users-650 samples total, notice-100 users-1300 samples total

# Load Testing Using Apache JMeter

Here I have used the Apache JMeter framework for load testing the given web applications using 50 and 100 samples with a ramp-up time of 100 seconds. The following steps were followed to complete the assignment:

## Steps:

1. At first, I opened Apache JMeter, where a Test Plan is available by default.
2. Under the Test Plan, I created an HTTP(S) Test Script Recorder.
3. I created three Recording Controllers for Moodle, Home, and Notice.
4. After setting up the proxy in my browser, I started navigating through the web applications, and the corresponding HTTP links were recorded.
5. Then, after recording for all three web applications, I ran my Test Plan.
6. I used assertions for the web applications as follows:  
    - Moodle: Duration - 500 milliseconds  
    - Home: Duration - 1000 milliseconds  
    - Notice: Duration - 500 milliseconds.
7. Next, I disabled the Recording Controllers for Notice and Home, ran the Test Plan for Moodle, and obtained the results and report using the command line.
8. I repeated the process for Notice and Home by disabling the other controllers sequentially.
9. The report for these are individually generated (2105123\_Home\_50\_html, 2105123\_Home\_100\_html, 2105123\_Moodle\_50\_html, 2105123\_Moodle\_100\_html, 2105123\_Notice\_50\_html, 2105123\_Notice\_100\_html).
10. And the result are individually generated and merge into one csv file (2105123\_results.csv).
11. Again enabling the recorder of the three websites at a time , I have generated the result for 100 and 50 users (2105123\_Overall\_50.csv, 2105123\_Overall\_100.csv) and report (2105123\_Overall\_50\_html, 2105123\_Overall\_100\_html).
12. Finally , I have submitted the 2105123.zip,2105123\_50\_html.zip,2105123\_100\_html.zip,2105123\_Overall\_html.zip.
13. In 2105123.zip file there is the result(results file for individual 6 cases, result file for 100 users in 3 websites and result file for 50 users in 3 websites ),testplan and assessment file , and the other zip files contains the reports.

As three web applications were tested with recorded navigations, Apache JMeter effectively simulated high user loads by sending HTTP requests to the web server. This allowed us to measure its performance and responsiveness under varying user loads.