Noah Davis, Jonathan DeFreeuw, Jordan Holland, Jisu Park, Dylan Zeigler, Bowei Zhao
Assignment 6 Phase 3

Team 2 Performance Testing

User Scenario 1

- Create a problem
- Delete the problem
- From Ashburn, US

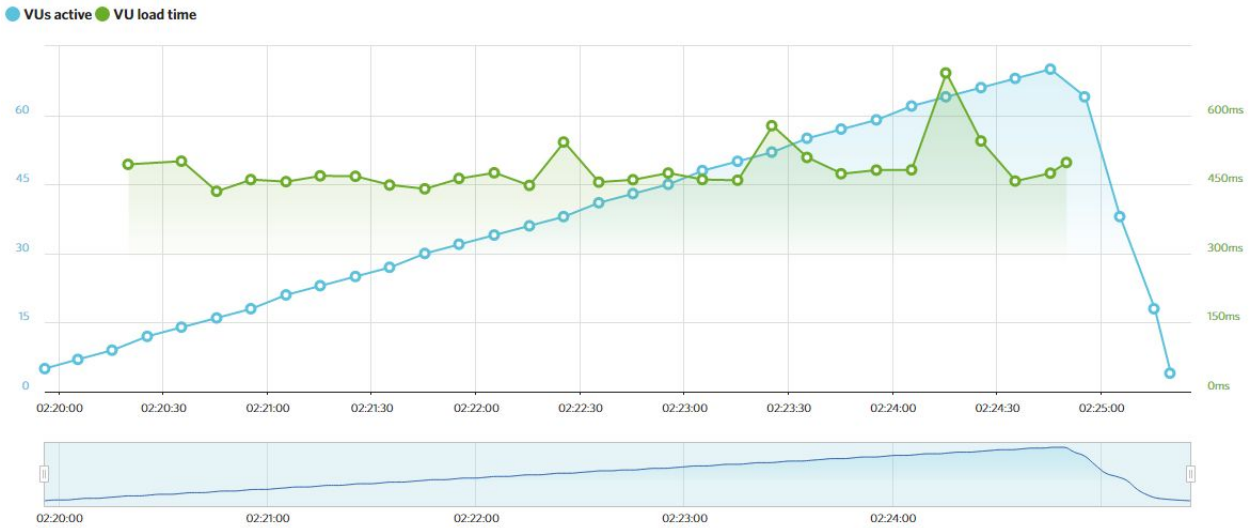

Figure 1. Overall Performance test for User Scenario 1

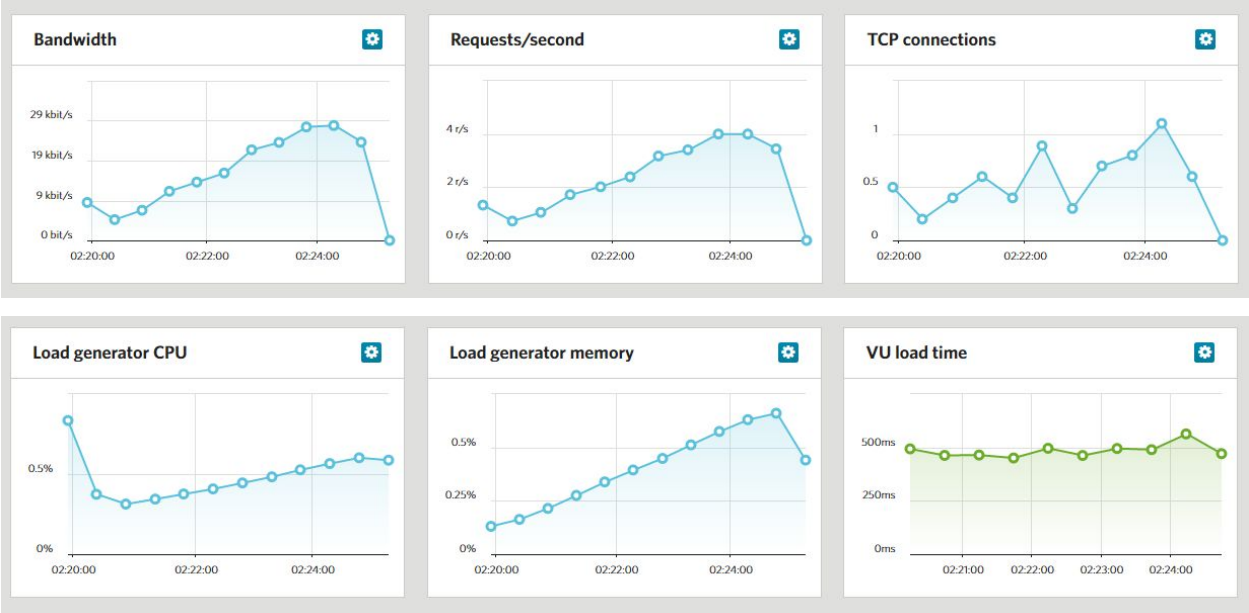

Figure 2. Individual Performance Data for User Scenario 1

The first scenario tests how a high number of concurrent users will affect the performance of the server. Test scenarios were kept simple to focus on the number of concurrent users. *Figure 1* shows the relationship between the active number of virtual users and load time of requests. The first irregularity of load time occurred at 38 virtual users (542 ms), and the first major irregularity of load time occurred at 64 virtual users (692 ms). Overall, the server handles the load test well enough that even with 70 concurrent virtual users (474 ms) the load time remained fairly constant.

User Scenario 2

- Create a problem
- Delete the problem
- From Ashburn, US / Montreal, CA / Tokyo, JP / London, GB / Seoul, KR

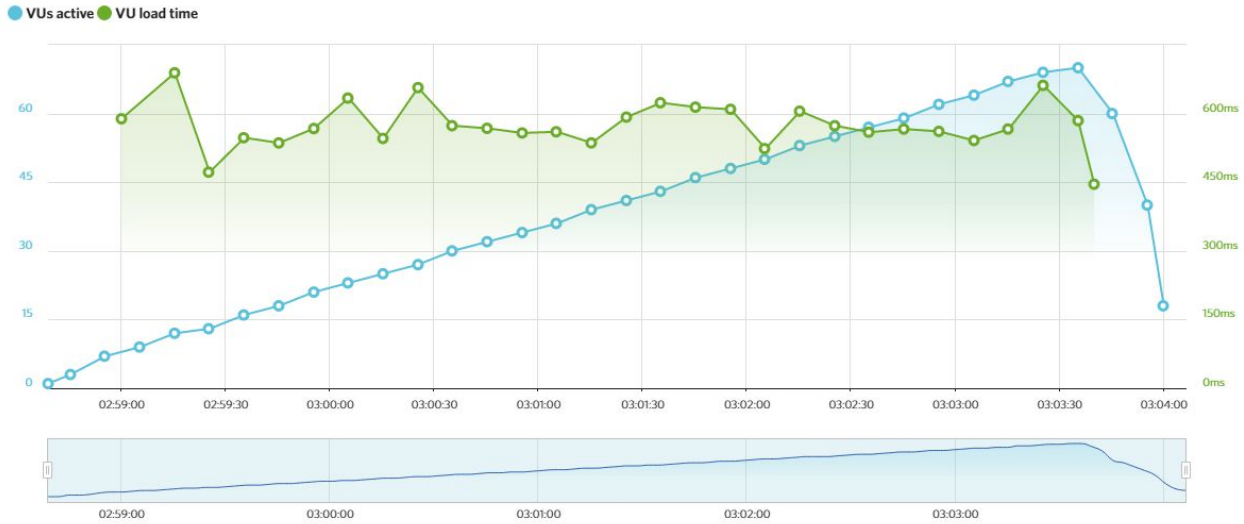

Figure 3. Overall Performance test for User Scenario 2

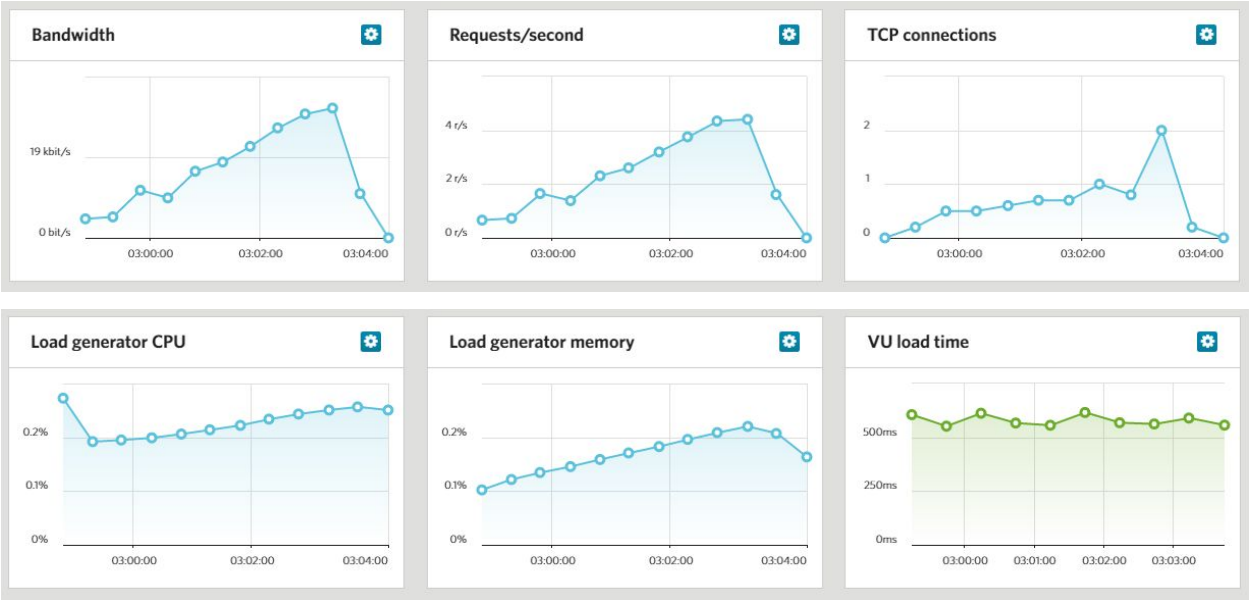

Figure 4. Individual Performance Data for User Scenario 2

The second test assessed whether location of the user had any effect on load time. Comparing the test results of the scenario 1 and scenario 2, two main observations can be made. Firstly, when the server was being used from different parts of the world at the same time, the load time increased by about 100 ms. Secondly, more frequent irregularities existed in load time when the server was being accessed from different locations.

User Scenario 3

- Posts problem 1 and problem 2 then deletes problem 2
- Gets current boundary then Puts new boundary data
- Post obstacle 1 and obstacle 2 then Deletes obstacle 2
- Gets all the obstacles
- Puts update for obstacle 1 and Gets obstacle 1
- Post robot 1 and robot 2 then Deletes robot 2 and default robot
- Puts update for robot 1 then Gets Robot 1 data
- Gets default goal then Put update for default goal
- Gets Path then Deletes problem 1

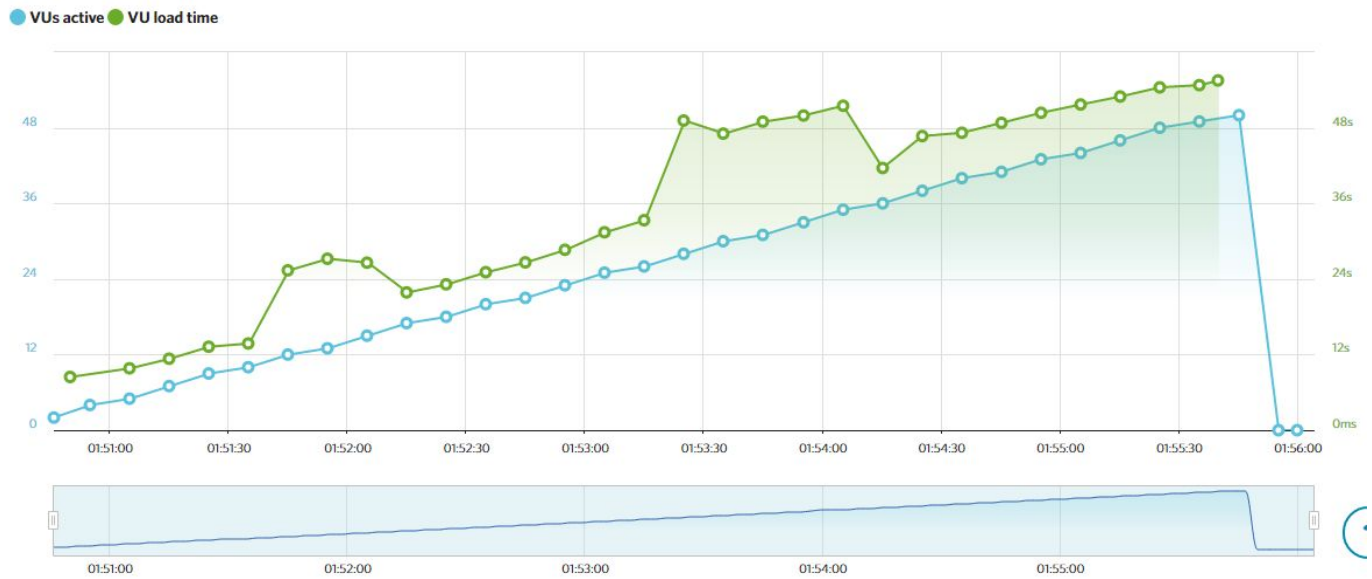

Figure 5. Overall Performance test for User Scenario 3

The purpose of the third test was to see the impact of complex requests on load time. In *Figure 5*, we see the overall performance test results. The blue line indicates the number of concurrent users while the green line indicates the response time. As one can see, a fairly linear trend in the response time is displayed in this user scenario. Considering that the user is requesting a path for each client, our response time is fairly sustainable. At 50 users one sees about a 55 sec response time indicating that it takes the bottleneck pathfinding service about a second per user added to response time.

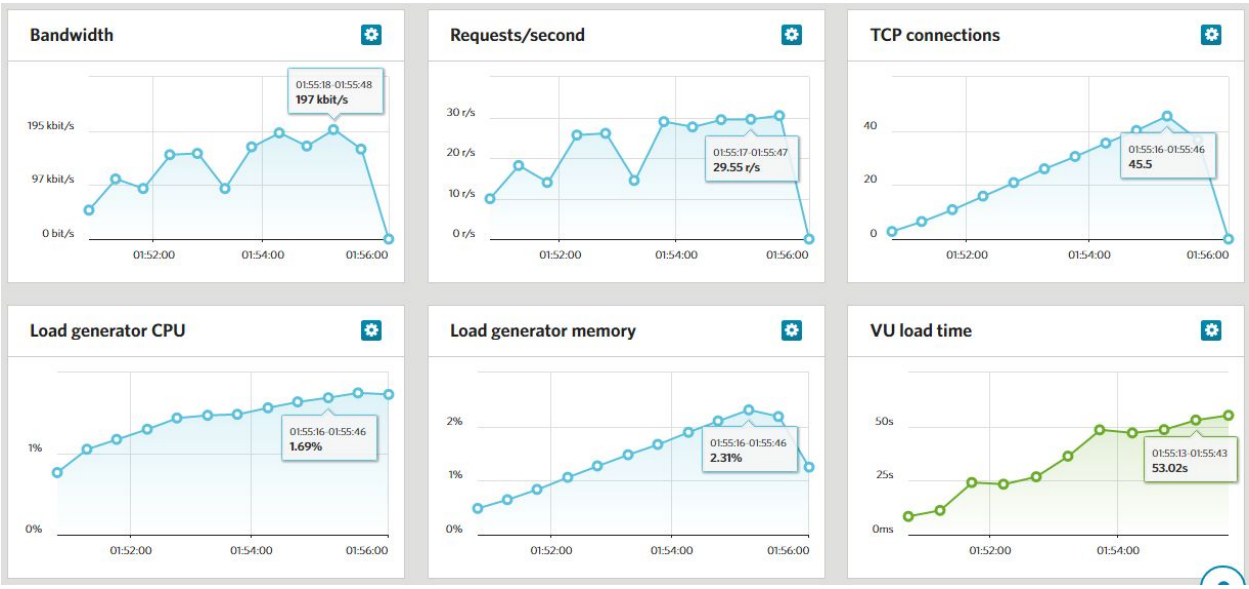

Figure 6. Individual Performance Data for User Scenario 3

If one compares *Figure 1* and *Figure 5*, one can see a bigger linear increase in the response time as the number of concurrent users increases. This is due to the fact that user scenario 3 had more complex tests from a simulated client. These tests included getting obstacles, changing the boundary, deleting robots and getting a path. The most extensive request is retrieving a path which in turn causes the most delay when accessing the server.

To replicate the performance test, use *Load Impact* online testing tool (https://loadimpact.com/) and the provided user scenario code in our github (https://github.com/bowei437/Round3_Team2/tree/master//PerformanceTesting/Scenario3.txt).

In conclusion, the server seems to have a linear trend for varied requests. Through various tests, it is observed how number of users, location of users, and complexity of the requests impact the performance of the server. Having a better loading technique on the server side could average out our response times to give faster results.