Discrete Event Simulation Analysis of a Web Application

Shamik Kumar De (22m0822)
Prateek Gothwal(22m0813)

Preliminary Situation

We continue our study of the performance of a web application with our system assumption to be the same.
Thread-to-core affinity .
The different take here is that the study will be through Discrete Event Simulation.

Why Discrete Event Simulation?

The main purpose is to granulate the big picture into smaller chunks. To represent each event or process as a discrete event.
Thus we model the behaviour of the system as a sequence of events.
This is solely done to gain insights as well as the impact of different cases on the performance of the web application.
In the longer run, these help in improving and optimizing the prevalent systems.

What did we do?

We basically implemented a simulation logic in our code which represents real world entities which generates events and the processes them to calculate performance metrics.

To look at it from a broader perspective, we have created three classes - request, server and event which interfere with each other to perform the simulation.

Highlights of our code

Priority Queue

A priority queue is present to store all the events that will take place in the system.

It has options of various request execution time distributions, such as - constant, uniform, exponential.

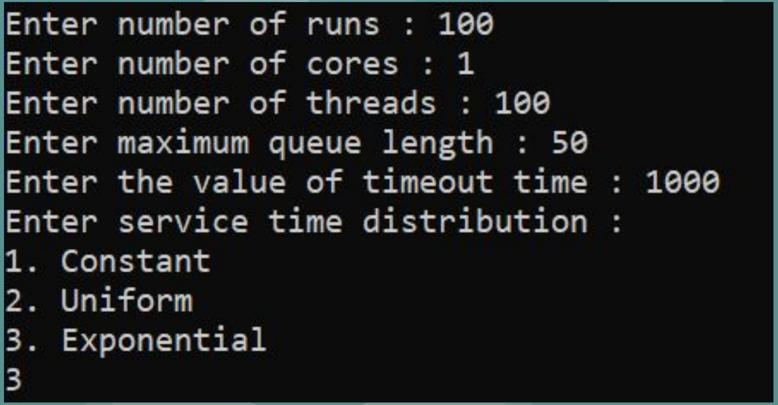
The main loop is running for the number of delays required times thus calling the arrival or departure handler based on the type of the next event in the priority queue.

Event Handlers

The arrival event handler is incorporated with the priority queue to assign a thread for the execution of events.
Some of the arrival events are placed in the server buffer till it crosses the max value at which request drops start occurring.
Soon after servicing , the corresponding depart events are pushed in the priority queue.
Request time out is checked before getting serviced, whenever a request is departing.

Working and graphs

We have tried to simulate the working of the closed load experiment that we did earlier. We have done the comparison between the Measured and Simulated values of Throughput and Utilization for different values of users.

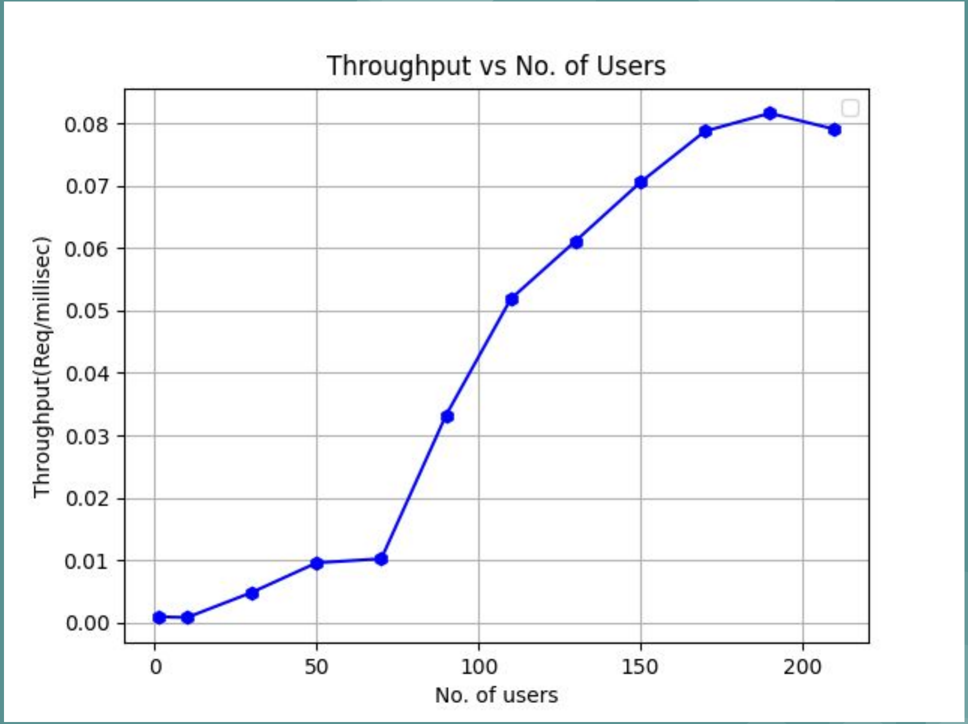


```
Enter number of runs : 100
Enter number of cores : 1
Enter number of threads : 100
Enter maximum queue length : 50
Enter the value of timeout time : 1000
Enter service time distribution :
1. Constant
2. Uniform
3. Exponential
3
```

System Specification

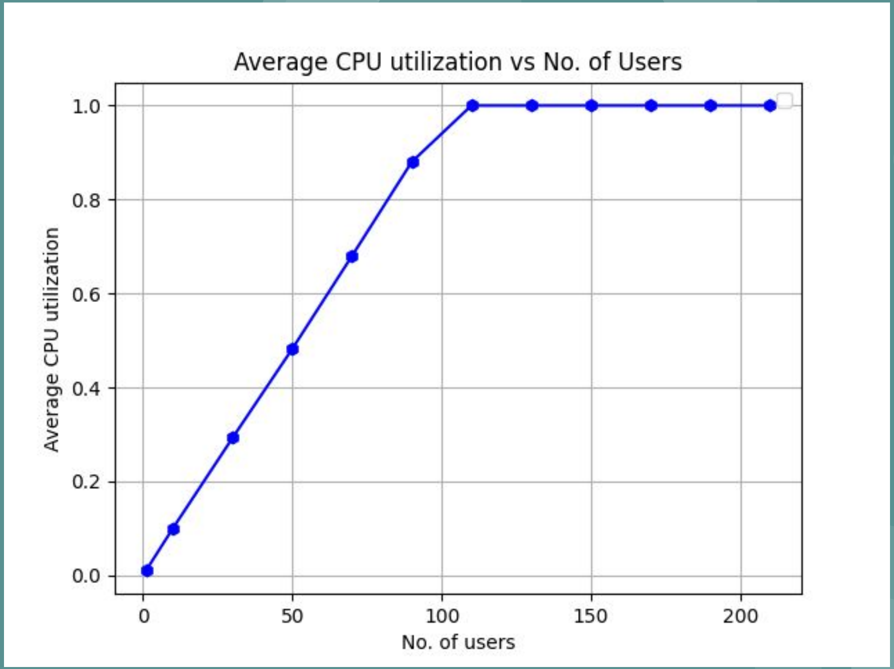
Working and graphs ...

The plot follows the confined rules and assumptions.

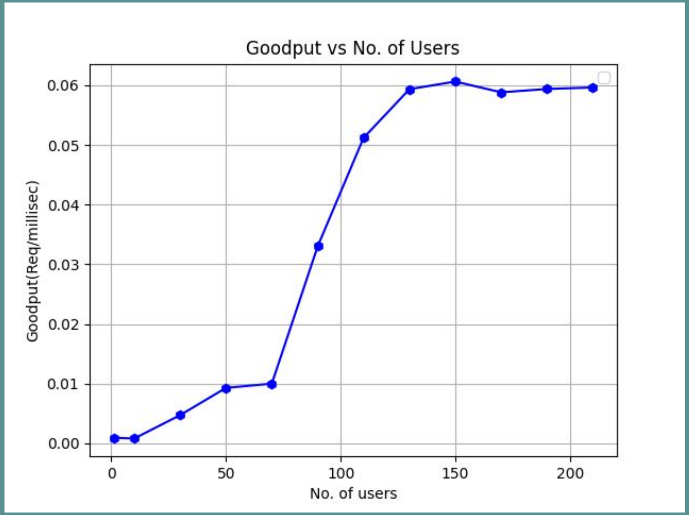


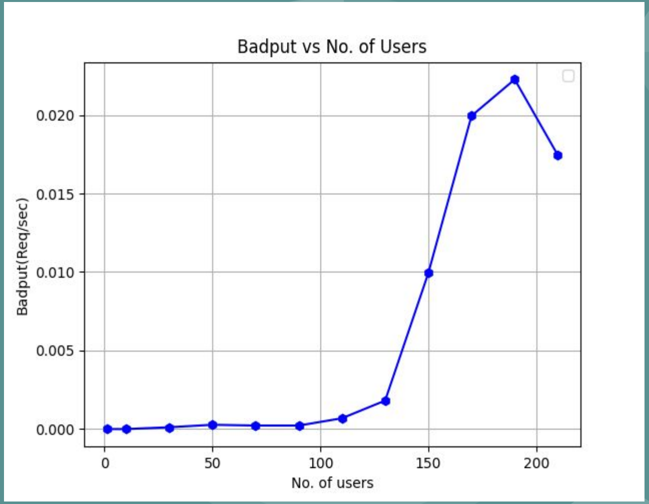
Working and graphs..

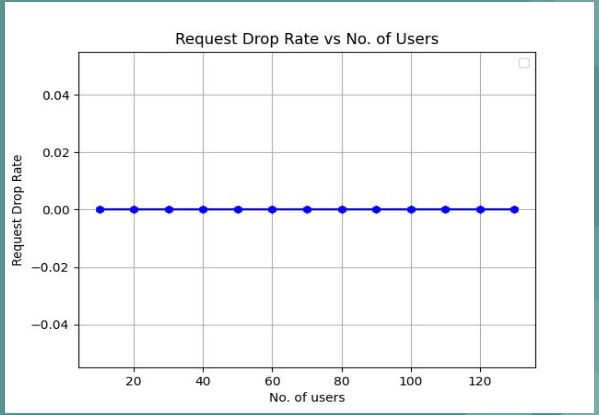
The utilization was simulated by finding out the total number of busy threads and the finding their fraction.



Working and graphs ...



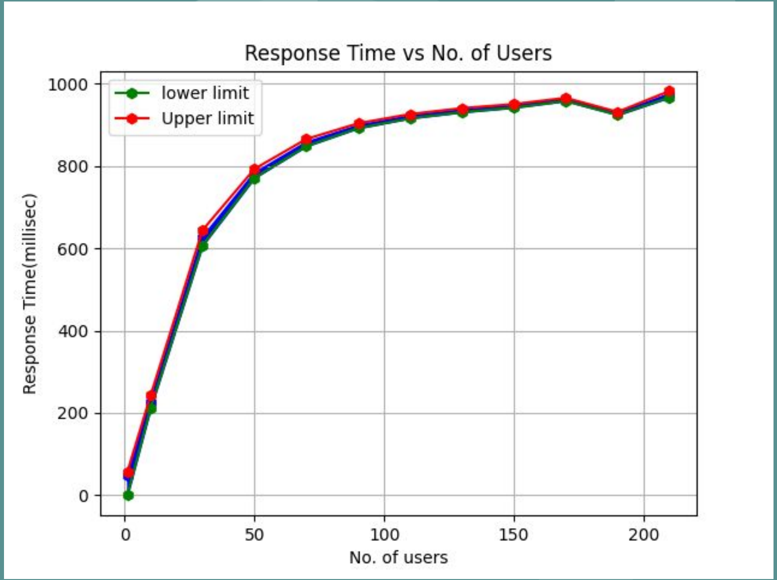




The goodput, badput and drop rate graphs are represented here.

Confidence Interval for Average response time

We needed confidence intervals of response time to get a idea where the actual value of response time might lie.
This shows that we have 95 % confidence of the original average response time for the specified number of users to lie in this interval.



THANK YOU