EPA CA2

Ruairí Keogh – X00121581

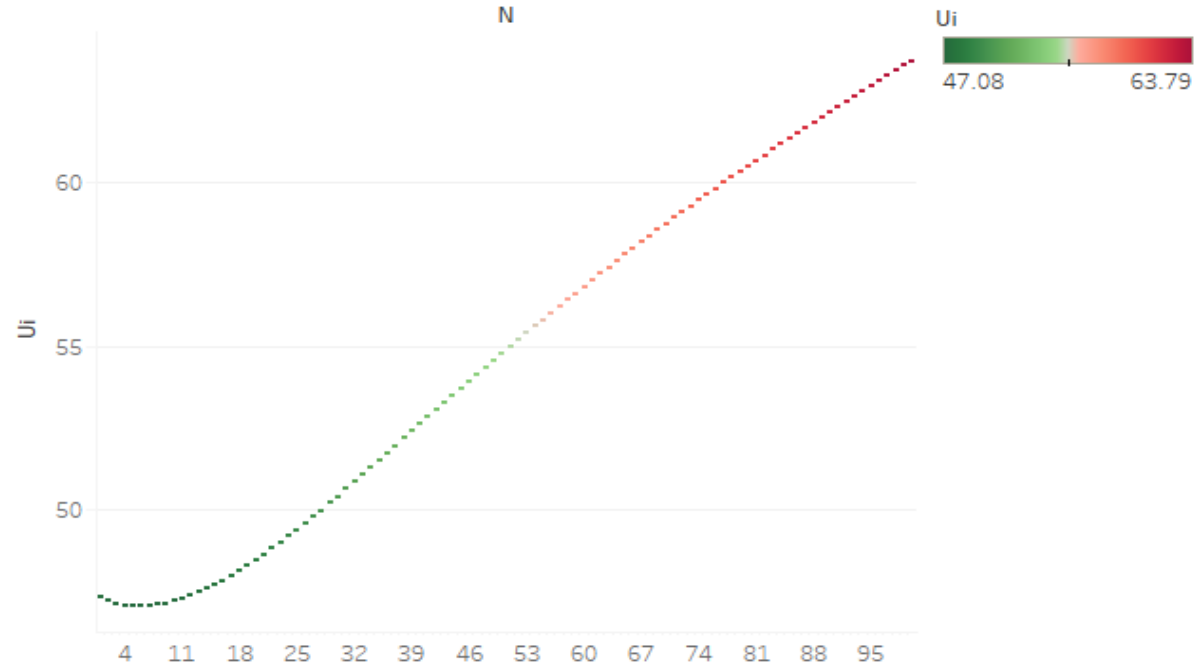
Table of Contents

Appendix A – Bash Script

```
#!/bin/bash

# Output for first line of Results

echo -e "C0 \t N \t idle" > results.dat



# For loop for each amount of users

for i in {1..100}

do

        # Run the loadtest script for each set of users

        ./loadtest $i&

        # Sleep

        sleep 5

        # Retreive C for based on the loadtest for i users

        C=`cat synthetic.dat | wc -l`

        # Retreive the Idle Percent for the loadtest

        Idle=`mpstat -o JSON | jq '.sysstat.hosts[0].statistics[0]."cpu-load"[0].idle'`

        # Kill the loadtest

        pkill loadtest

        # Output results

        echo -e "$C \t $i \t $Idle" >> results.dat

done
```

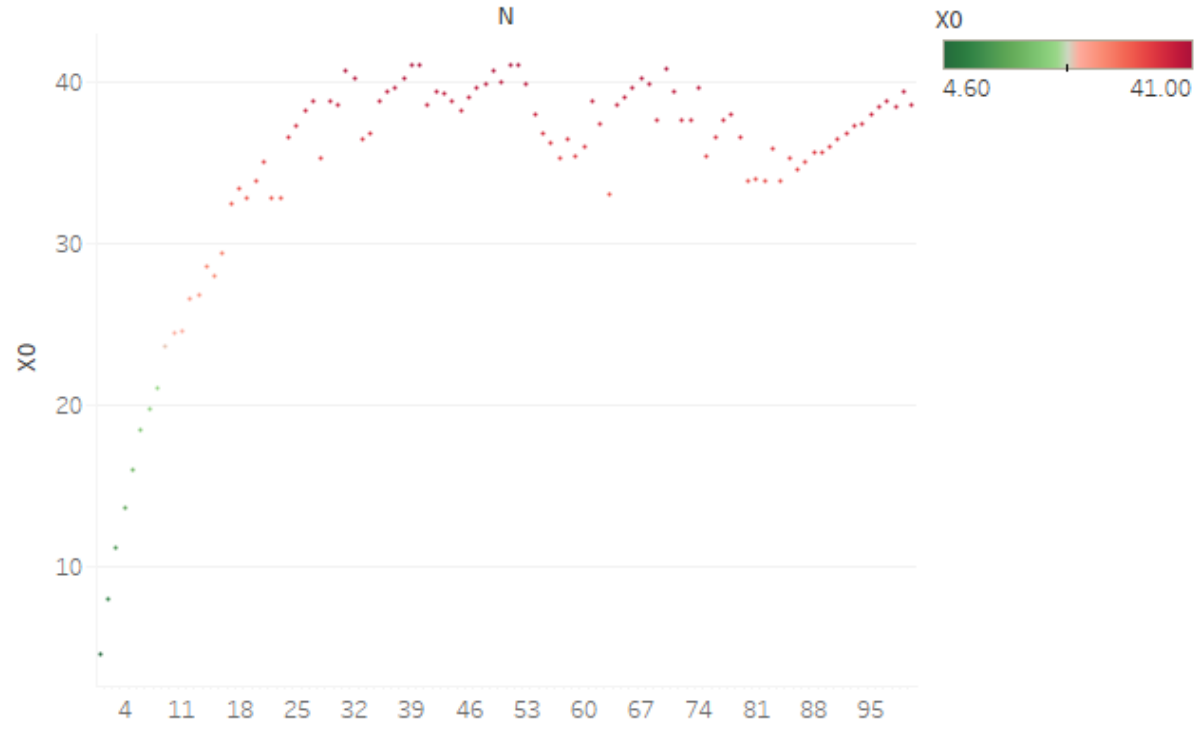
Appendix B - 4 Plots

Ui vs N



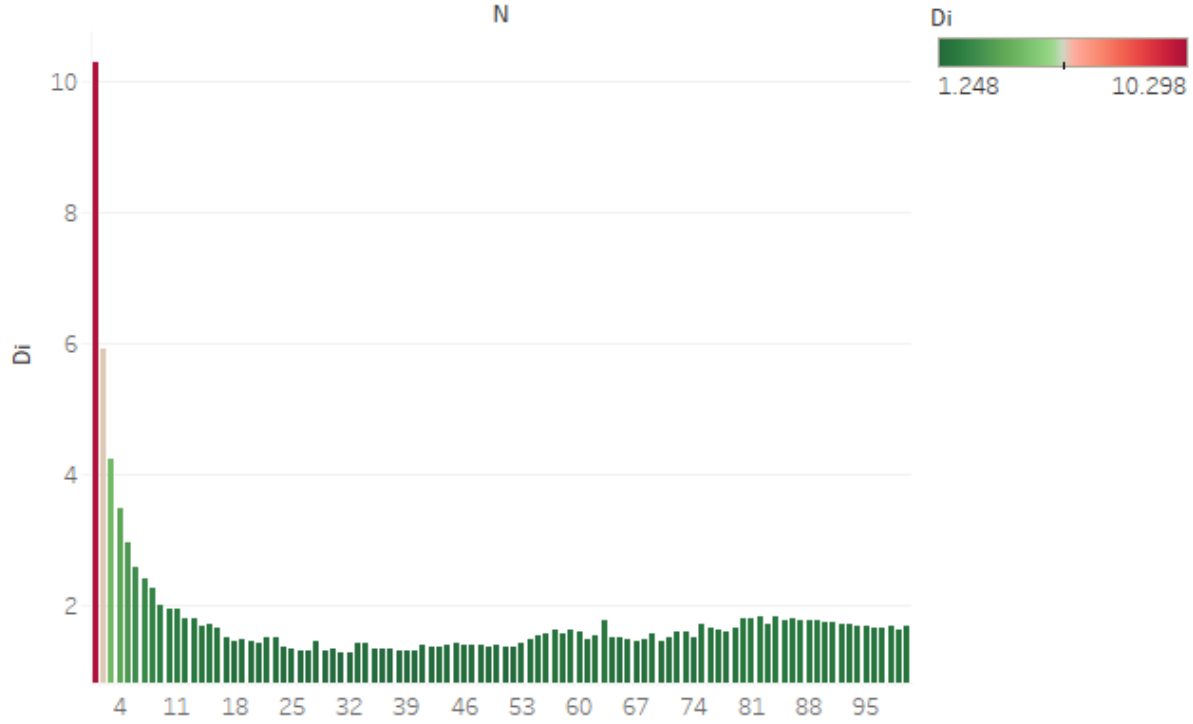
Ui for each N. Color shows details about Ui.

X0 vs N



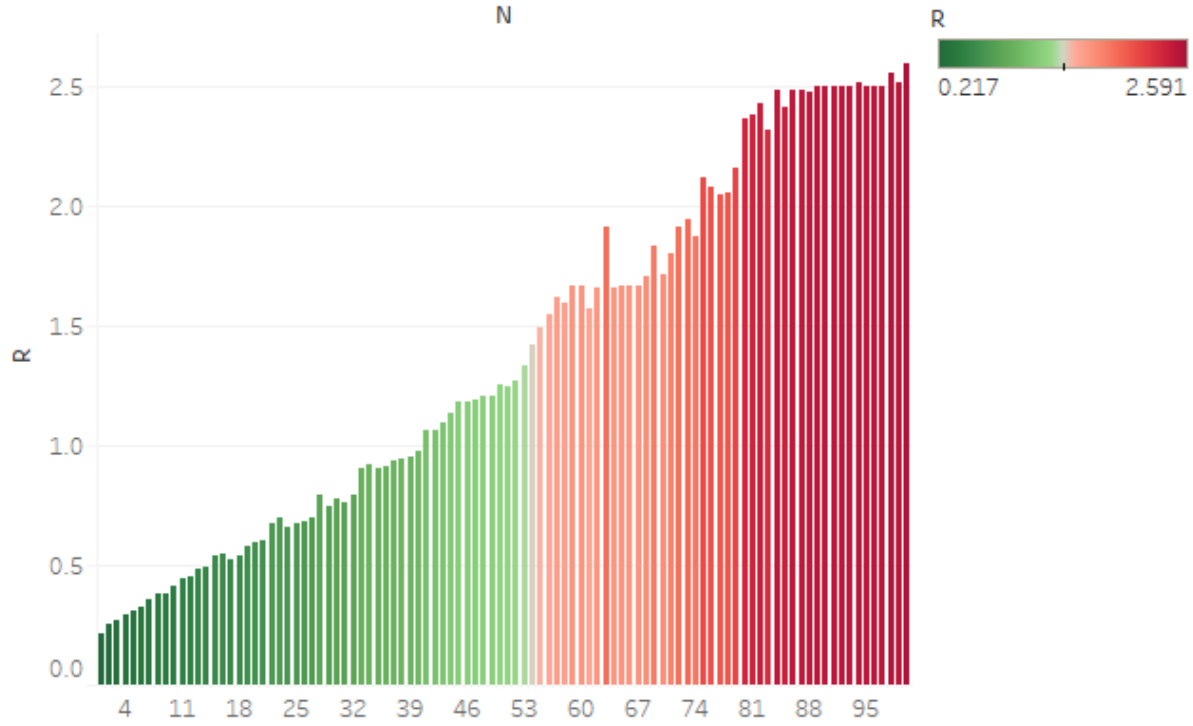
X0 for each N. Color shows details about X0.

Di vs N



Di for each N. Color shows details about Di.

R vs N



Sum of R for each N. Color shows sum of R.

Appendix C – Discussion

For each of these graphs it is clear to see that the performance of the machine is directly affected by the amount of users during each transaction. From the results of the bash script it was possible to graph the different aspects of the machine. The 4 graphs created are as follows, Ui vs N or Utilization vs the Number of Concurrent Users, Di vs N or Service Demand vs Number of Concurrent Users, X0 vs N or ThroughPut vs Number of Concurrent Users and R vs N or Response Time vs Number of Concurrent Users.

From the first graph (Ui vs N) it is clear that as there are more concurrent users the loadtest requires more CPU Utilization. When ran on the Fedorra Virtual Machine it was struggling to make it towards the end as the CPU usage was near 65% for the loadtest. Knowing this a user can limit or control the utilization of any such process so that there is a risk management measure to stop a process in execution when the system load exceeds a threshold.

Judging off the graph for Service Demand vs Number of concurrent users it is clear the service demand decreases as more concurrent users are added. This shows that there is a shorter amount of time spent for the request. From the graph I understand that the first request for a low amount of concurrent users needed the most time as it was a new request. After the number of concurrent users is increased the service demand decreases as it is processing the same requests.

The most interesting graph is probably the Throughput vs Number of Concurrent users as there is a steep incline at the beginning then multiple growths and falls for the majority of the graph. These growths and falls could indicate some stability for the machine as they are almost consistent with how they rise and fall. The most notable fall would be after the users reach 50. It is at this point that the rises and falls become more gradual and between about 65 and 80 users is a slow decrease for the throughput. After this then it begins to increase again until 100 users are reached. I feel like that if I had of used a

lower sample size of 50 concurrent users then the graph would have been characteristics that would be expected. This would be the normal gradual increase in throughput as the users increase with the peak coming somewhat towards the final few users e.g 45 onwards. As I had 100 users I feel the graph peaked too soon so the figures for the majority are based off the machine running with a large number of requests on the load. This in turn shows how the throughput has a direct correlation with the load and concurrent users at that time.

For the final graph based on Response Time vs Number of Concurrent Users there is a much smoother incline as the number of users increases. In the beginning with users from 1-50 the response times are actually really good with the times ranging from 0.217 to 1.25 secs. This can be compared to the poor time of 2.59 when there was 100 concurrent users at a time. With this poor timing with 100 users it is clear that the CPU will be limited in its actions. For instances such as this where the number of users is high there should be measures put in place to help the system such as increasing CPUs, Cores for processing and include hyper-threading.

Some overall conclusions I have gathered from completing this assignment are that I could have possible tried this load testing with concurrent users on multiple different operating systems with different CPU's, memory and disk configurations so that I had a broader range of results and could compare them to each other. From what I did gather I am happy with how my results were formed and it is clear they show the increased stress on the performance as the concurrent users increased.