EAA CA 2

Analyse metrics of a CPU

Abstract

The following paper analyzes the Utilization, Service Demand, System Throughput and Response Time of a single CPU as the number of concurrent users increases from 1 to 50

Michael Smith

X00107586

Contents

Introduction

For the second continuous assessment assignment, the objective was to find out how increasing the number of concurrent users would affect different aspects of a single core CPU. We would then take the data produced and display the results using R.

For the CA I used a fedora virtual machine running on a Fedora(64-bit) operating system with a single core CPU.

To begin with I created a bash script with a for loop that would increase the number of concurrent users by 1 each time that it ran the loop. Inside the loop I took note of the necessary values that we are trying to analyse. In this case, we were looking for

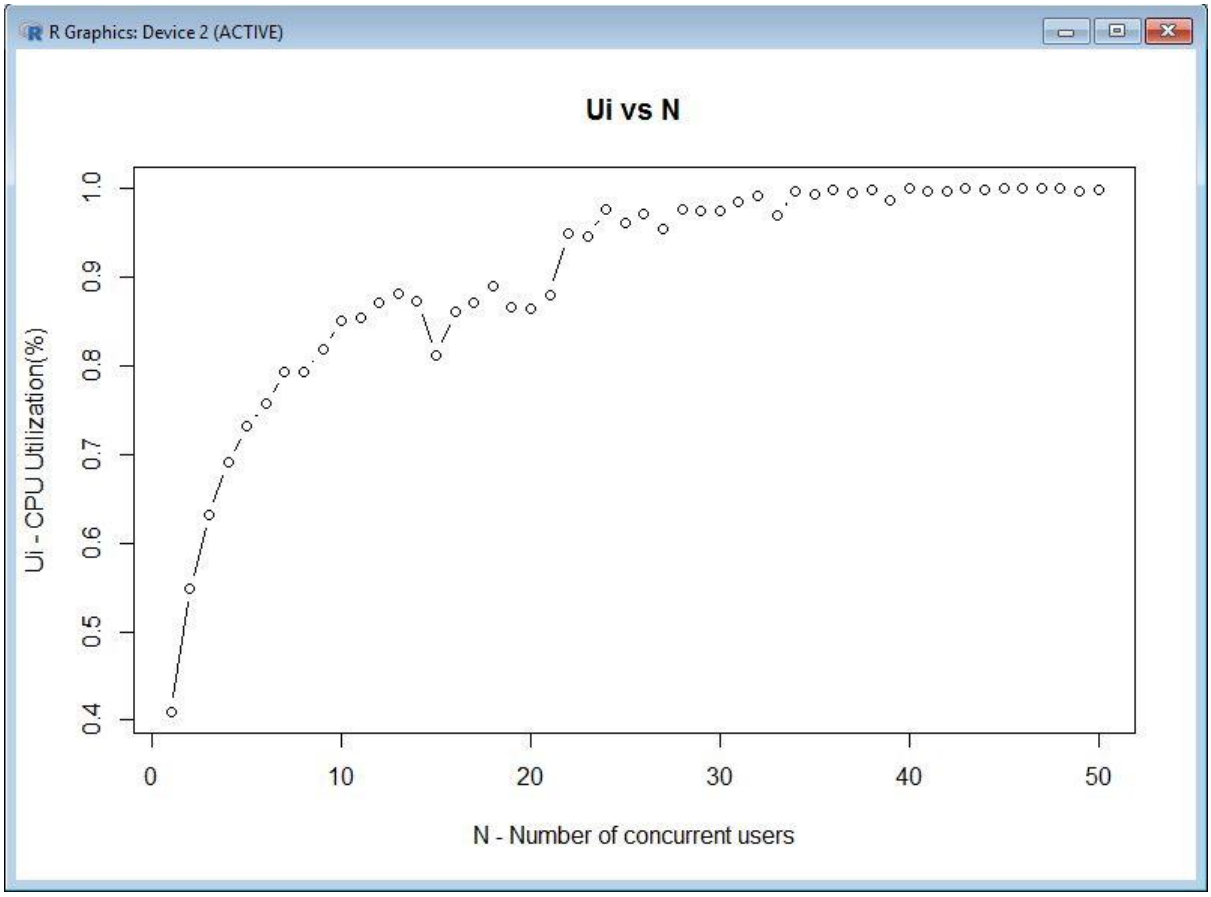
- CPU Utilization
- Service Demand
- System Throughput
- Response Time

To gather this information, I ran the bash script using command line on a single core virtual machine.

Once I had the necessary information I then read the file into R and created functions to calculate these metrics and then plot the resulting information in separate graphs.

From these graphs, I could see how increasing the number of concurrent users affected the CPU.

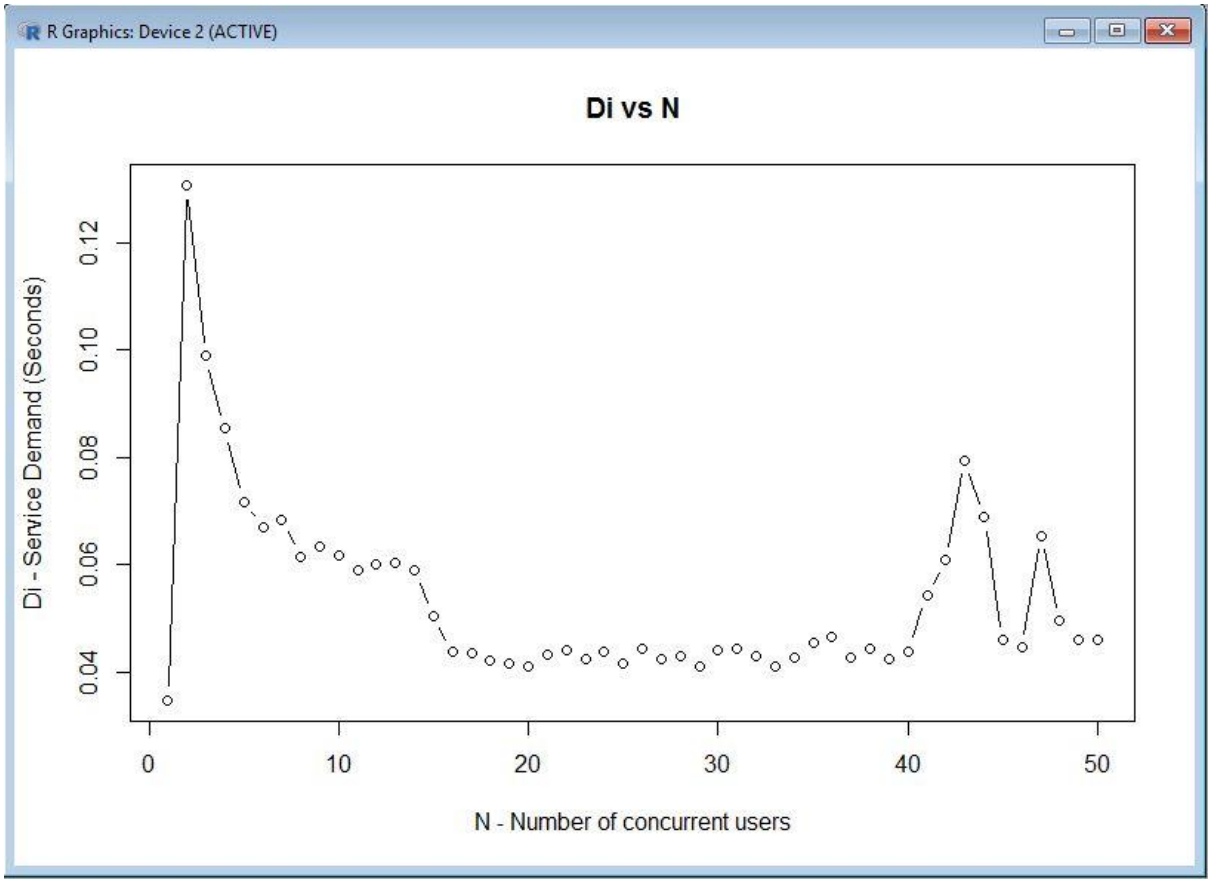
Graph 1 – Utilization - Ui vs N



The first graph that I plotted was to find out how the number of concurrent users would affect the Idle time of the CPU. To do this I had to find the CPU Utilization (Ui). To find Ui, I began by taking away the amount of time the CPU was idle for from 100, the resulting figures gives me the amount of time that the CPU was being utilized and I stored this in a variable called Usage. I then divided the result by 100 to get it to a percentage between 0 and 1.

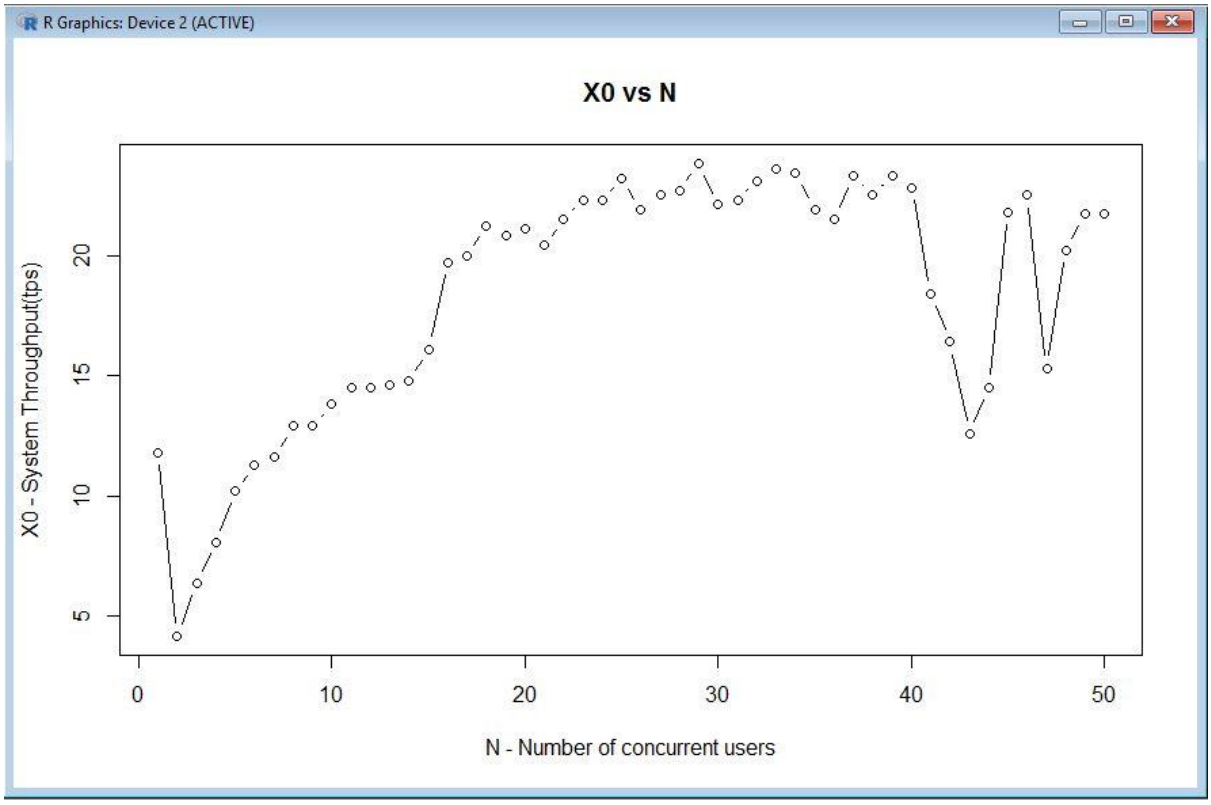
We can see from the above graph that when there was only a small number of concurrent users the CPU Utilization was very low meaning that there was a high level of idle time. But we can see that the CPU Utilization rapidly grows as more concurrent users are accessing the CPU, this is due to the fact that as more users are trying to access the CPUs resources this increases the workload for the CPU until it is at a point where it is being fully utilized. We can see in the graph that the CPU levels off towards the end, this is because the CPUs utilization is at a maximum 100%.

Graph 2 – Service Demand - Di vs N



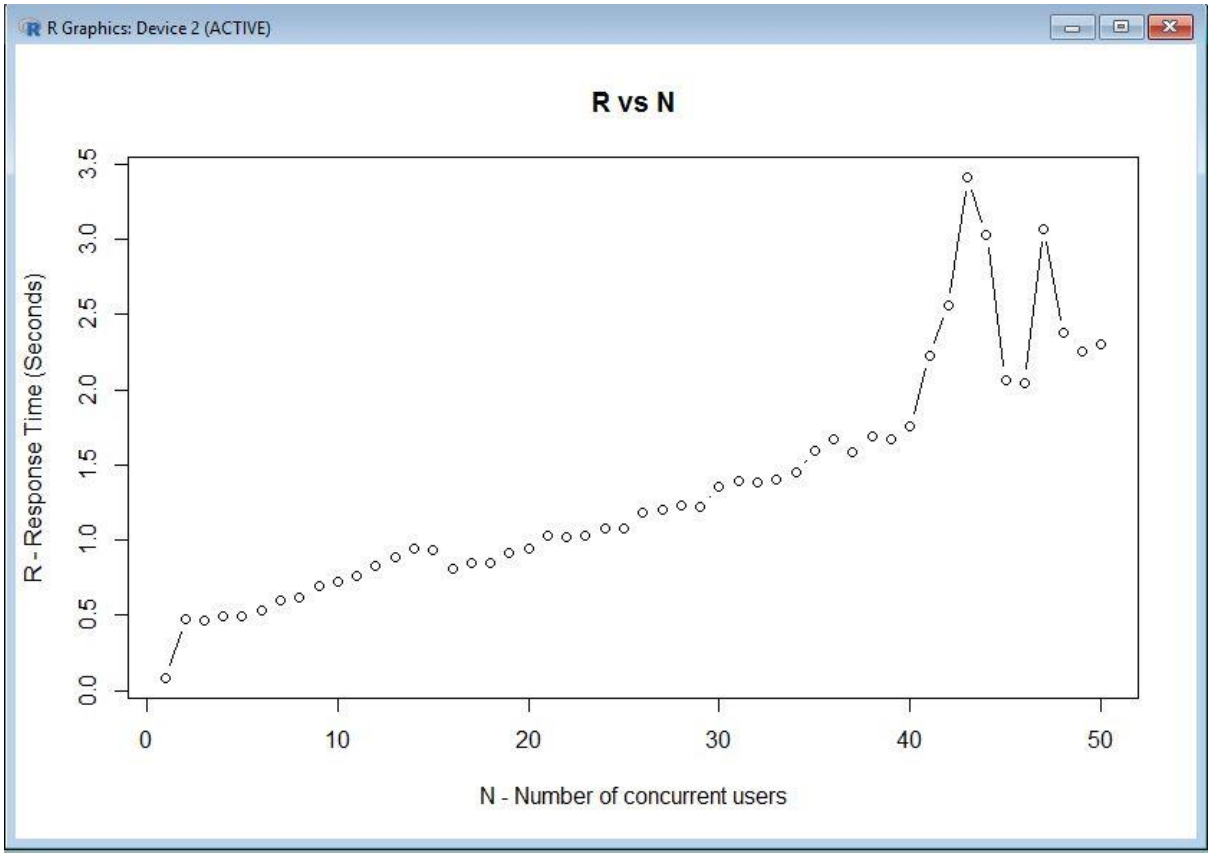
The next metric that I wanted to analyse was the service demand. Service demand measure the amount of time in seconds that is spent requesting a service from the CPU. Service demand is calculated by dividing the CPU utilization by the system throughput (Di = Ui / X0). We can see from the graph that the service demand starts off at a high level but immediately begins to drop as more concurrent users are added.

Graph 3 – System Throughput – X0 vs N



The next metric I calculated was the system throughput. System throughput is the number of transactions that the CPU can complete per second. System throughput is calculated by taking the total number of completions by the system in the observation time and dividing it by the observation time (X0 = C0 / T). We can see that as the number of concurrent users increases so does the amount of system throughput. The system throughput never rises above 23.80tps meaning that the maximum amount of throughput per second is 23.80 transactions.

Graph 4 – Response Time – R vs N



The final metric calculated was the response time in seconds of the CPU. The response time is calculated by rearranging Littles Law (N = RX0) so that we can find R by dividing the number of concurrent users by the system throughput (R = N / X0). The response time measures the amount of time that it takes the CPU to respond to a request by a user. We can see that as the number of concurrent users rises so does the amount of time that the CPU takes to respond to a request.

Summary

Below is a summary of each metric

```
> summary(Ui())
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
 0.4097  0.8614  0.9653  0.9026  0.9967  1.0000
> summary(Di())
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
0.03472 0.04296 0.04491 0.05339 0.06067 0.13060
> summary(X0())
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
   4.20   14.50   20.95   18.32   22.30   23.80
> summary(R())
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
0.08475 0.83300 1.13200 1.32600 1.67400 3.41300
> |
```

From this summary, we can find some interesting information. We can see the minimum, average and maximum values for each metric calculated and can see the difference that is made by increasing the number of concurrent users.