Bash, R Script

EAA CA2

X00115013

Tom Murray

EAA CA2

Introduction

Results.sh

Step 1: I put the files been read from and wrote to into local variables for ease of use.

Step 2: Printed the heading to the results.dat file.

Step 3: Started a for-loop to simulate fifty different users.

Step 4: I used a 10 seconds timeout to kill each call to the loadtest.c file and set them to run concurrently by setting them up as separate threads running in the back ground.

Step 5: Ran MPSTAT to produce ten reports of global statistics from the processor at 1 second intervals and wrote the results to a temp file.

Step 6: Calculated the total Co by doing a line count in the synthetic.dat file

Step 7: Retrieved the 12th column (Average) from MPSTAT output using AWK, which is the CPU average.

Step 8: I then appended the results to the results.dat file

Script.r

Step 1: I set up the path to the R file.

Step 2: I the read in data from results.dat

Step 3: At this point I put the time and percentage into local variables for ease of use (timeTest<- 10), (percent <- 100)

Step 4: I got the average utilization (avgutil <- 100 - perfdata$IDLE)

Step 5: The next step was to bring the users into the equation and write it to a local variable (n<- perfdata$N)

Step 6: I used functions to do my calculations as it seemed a cleaner method

1. I first calculated utilisation (ui <- avgutil/percent)
2. Then I calculated throughput (x0 <- perfdata$C0/timeTest)
3. Then I calculated the service demand (di <- calcUI()/calcXO())
4. Lastly I calculated the Response time (r <- n/calcXO())

Step 7: At this point it was time to set my graphs for:

1. Graph for utilisation Vs Users
2. Graph for Throughput Vs Users
3. Graph for Service Demand Vs Users
4. Graph for Response Time Vs Users

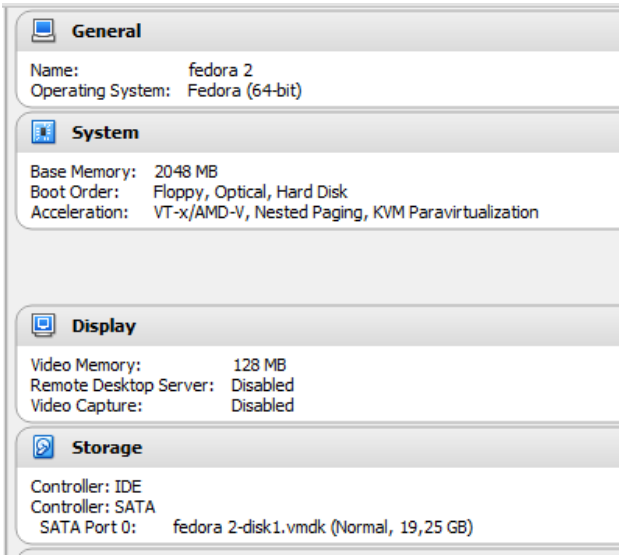
Step 8: Calculating a summary for each calculation

1. summary(calcUI())
2. summary(calcXO())
3. summary(calcDI())
4. summary(calcR())

Step 9: I ran the script.r file in R and the graphs it produced are below.

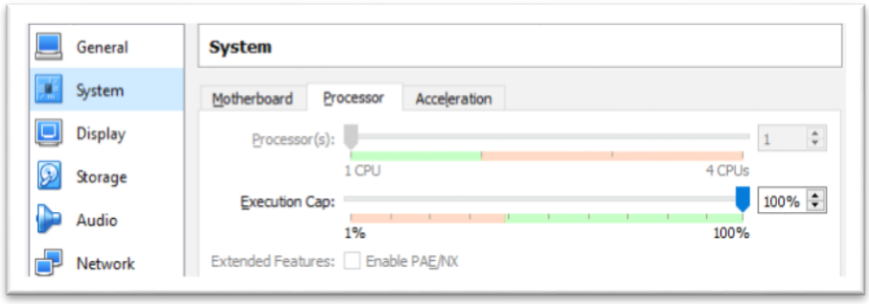
Hardware

The Operating System I tested is a Red Hat Linux OS version fedora. I ran this on my windows machine using a virtual machine to run my tests. The memory amount used is 2048 MB and with a simple SATA setup for the disk configurations with 20 GB of memory made available.

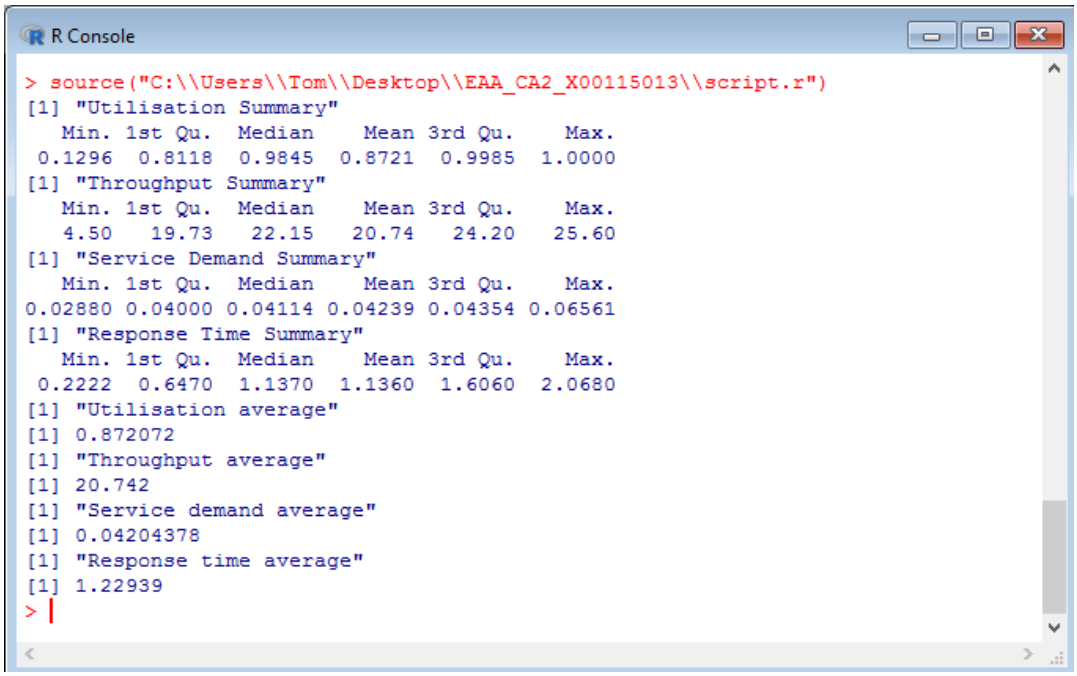


The CPU is a single core CPU

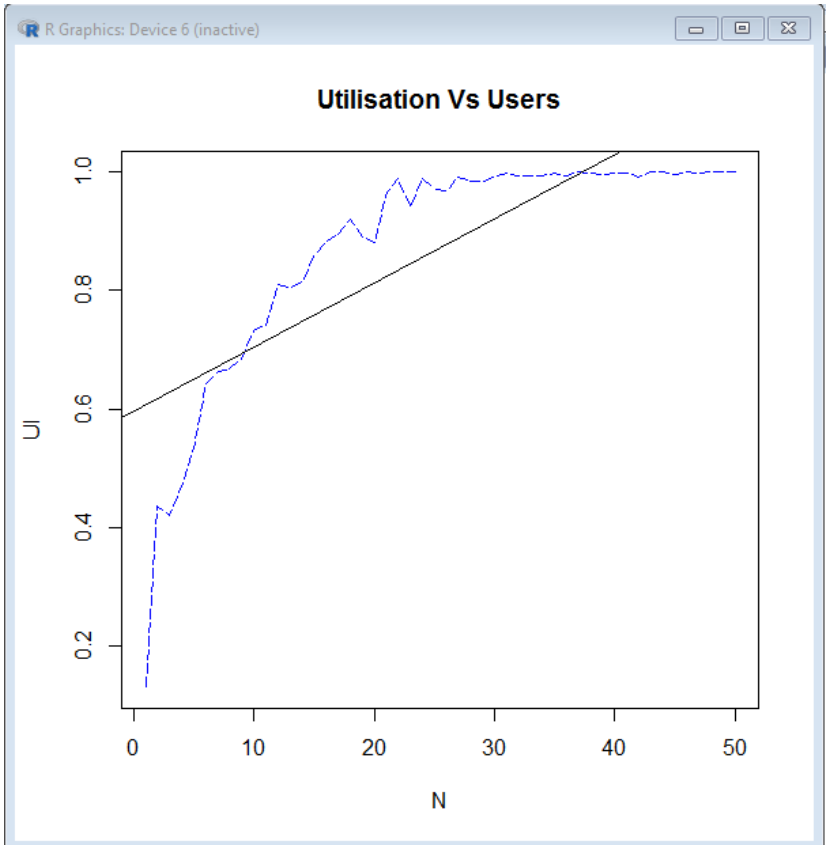
```
processor       : 0
vendor_id       : GenuineIntel
cpu family      : 6
model           : 61
model name      : Intel(R) Core(TM) i3-5010U CPU @ 2.10GHz
stepping        : 4
cpu MHz         : 2095.148
cache size      : 3072 KB
physical id     : 0
siblings        : 1
core id         : 0
cpu cores       : 1
```



Summary Results

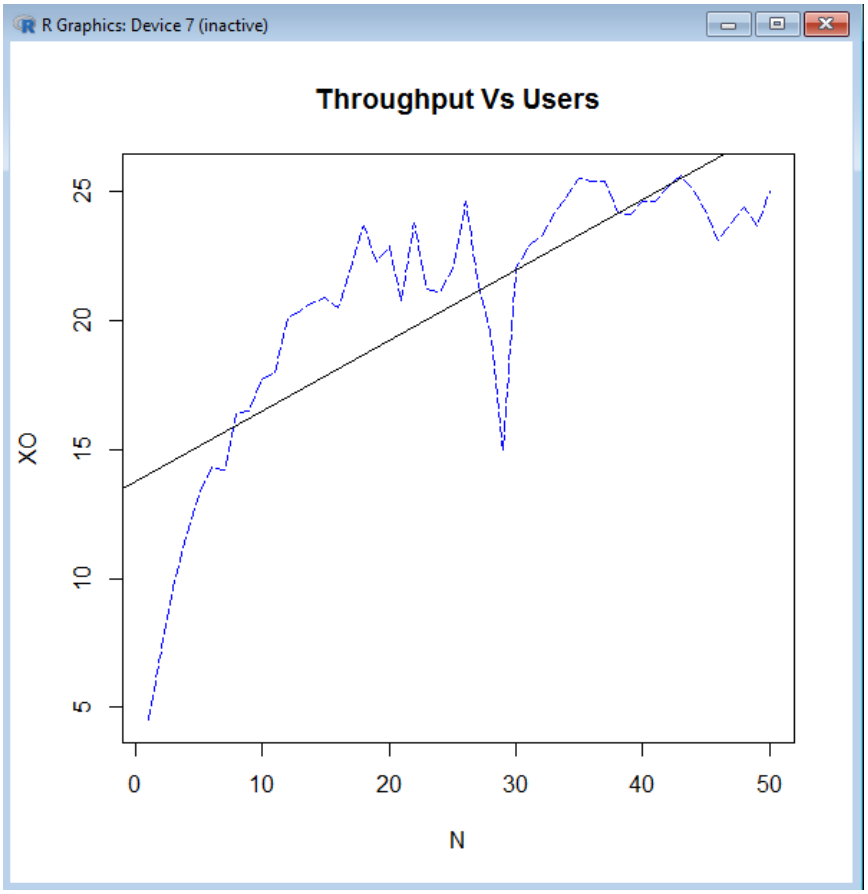


Utilisation Vs Users



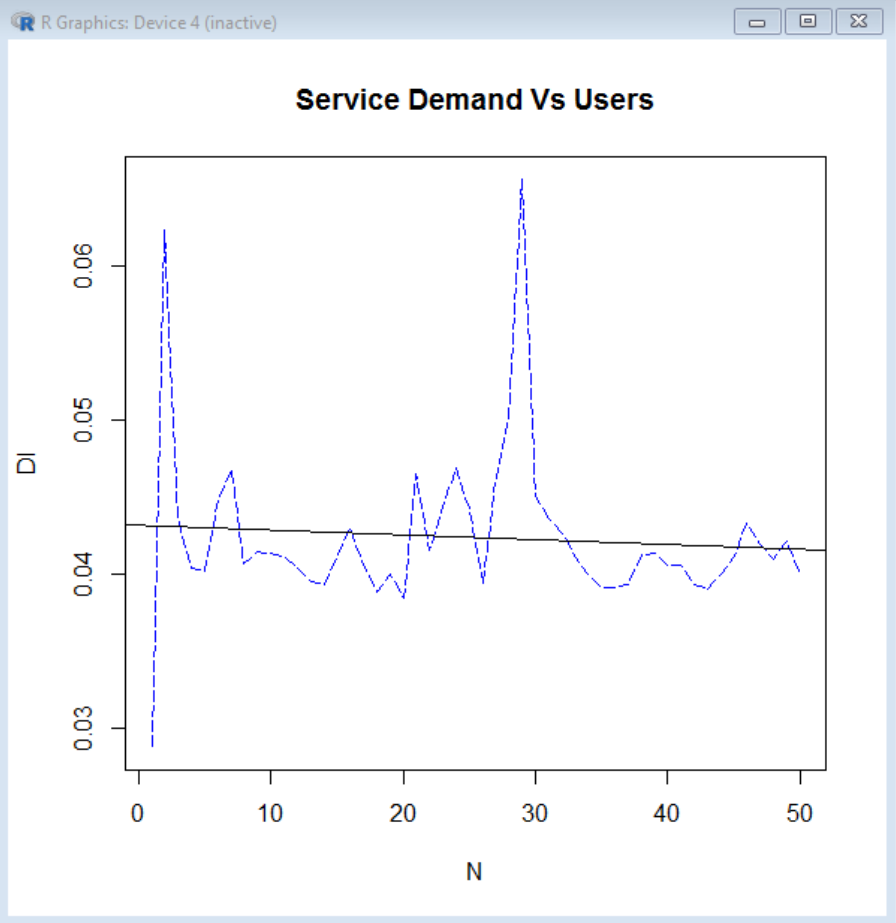
CPU utilization in this context is the proportion of the total available processor cycles that are consumed by each process. From this graph we can see that the CPU reaches utilization at about user 23-30. At this point the CPU has maxed out and the average CPU idle time went to zero. This graph suggests that the system may has insufficient CPU power.

Throughput Vs Users



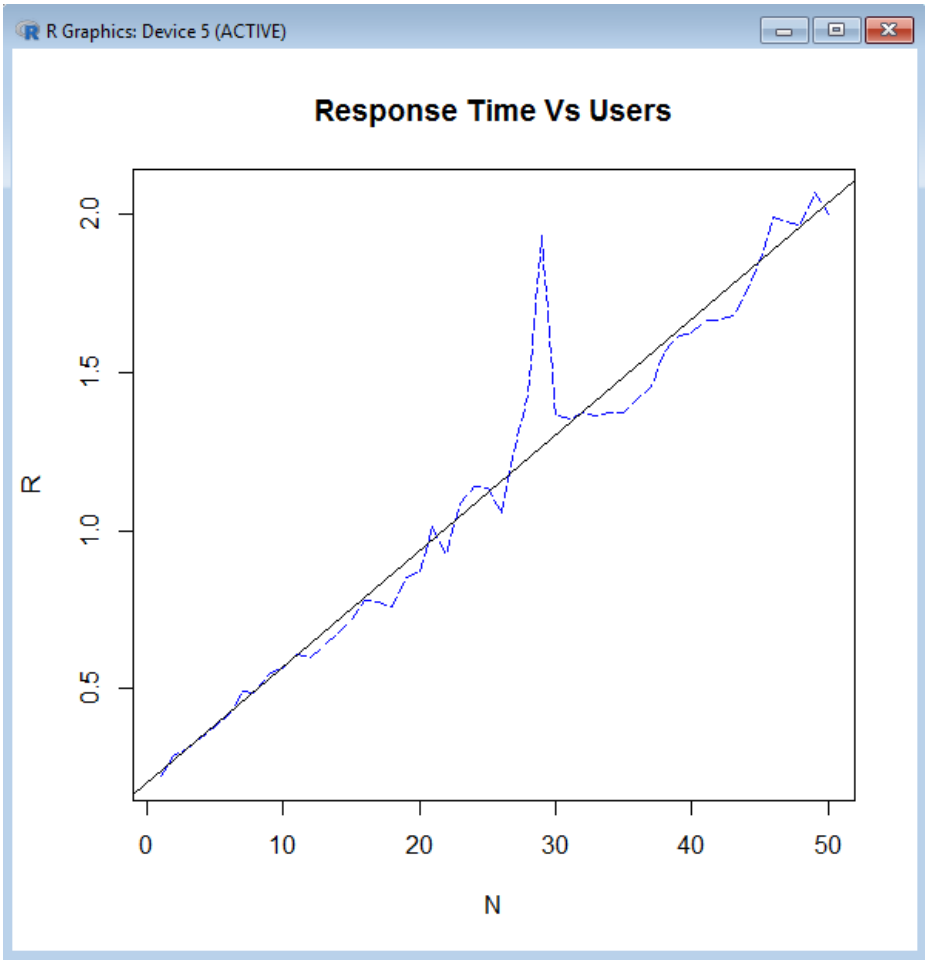
Throughput is a measure of how many units of information a system can process in a given amount of time. From this graph we can see that this system peeks at about 22-23 users and then struggles to manage the load and the speed of the throughput levels out. User 30's throughput drops and can be reflected in the service demand and response time graphs below.

Service demand Vs Users



Service demand is defined as the total average time spent by a transaction from the time of its first access of the CPU to the end of its cycle. A transaction may visit the CPU multiple times before it is completed. From this graph we can see that the time it takes each transaction to finish spikes at the start and then drops and stays steady as the CPU has a set time slot for each process. The difference between the competition times of these transactions is minimal in this case but with a larger set of tests I can see that this system would eventually become to slow and reach utilisation quickly. As for the spike at user 30 I can't explain what happened but as the system reached utilization around user 30 this might have something to do with it.

Response time Vs Users



Response time is the total amount of time it takes for a transaction to be on the system. The response time is the service time and wait time combined. From this graph we can see that as the transactions start to pile up the response time increases. What is showing is that the average wait time increases in a linear fashion the busier the CPU is dealing with other transactions. As for the spike at user 30 just like in the service demand I can't explain what happened. Unless it has something to do with CPU utilisation has been reached.