

Enterprise Performance Architecture

[CA2]

Conor Griffin x00111602

## Contents

Introduction .....	2
Graphing the results .....	3
Ui vs N .....	3
Graph : Ui vs N .....	3
X0 vs N.....	4
Graph : X0 vs N.....	5
Di vs N .....	6
Graph : Di vs N.....	6
R vs N.....	7
Graph : R vs N.....	7
Appendix A.....	8
Bash Script runtest.sh .....	8

## Introduction

For the purpose of this CA I will be discussing the throughput, utilization, service demand and response time of a single CPU as the number of concurrent users increases from 1 user to 100 users. I decided to run the loadtest with up to 100 users to reach full saturation on the CPU.

I completed this CA using the fedora 26 (64 bit) Linux virtual machine.

- Linux Operating System.
- I allocated a single CPU to the OS.
- I allocated 2GB of memory (2048 MB)
- Virtual Hard Disk : Fedora 26.vdi
- Virtual Size 100 GB
- Actual Size 8.82 GB
- Dynamically allocated storage

I began by creating a bash script (runtest.sh) which looped through each number of concurrent users (N) on the CPU and ran the loadtest where i represented the number of users. While the loadtest was running I got the system stats using mpstat to get the idle time on the CPU. I also got the number of completed transactions from a synthetic .dat file which is produced each time the loadtest is invoked. When the script was complete, after each loadtest ran the results were appended to a results.dat file.

## Graphing the results

I used Tableau to load the results.dat file and use the results to build graphs representing CPU utilization (Ui), service demand (Di), throughput (X0) and response time (R).

### Ui vs N

To get the CPU utilization (Ui) I looked at the idle time column in my results.dat file which represented how much time as a % the CPU was not being used. When the CPU is at 100% execution cap the CPU is fully utilized so to get Ui I subtracted 100 from the idle time and divided this by 100 (because CPU is between 0 and 1).

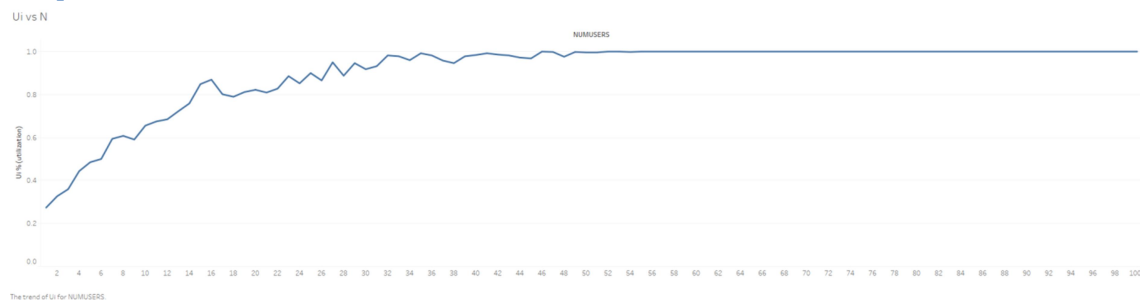
I used tableau to create a calculated field representing Ui. I then used Ui and N as my axis to create the graph Ui vs N.

1	C0	N	idle
2	23	1	72.64
3	38	2	67.34
4	51	3	64.06
5	60	4	55.65
6	70	5	51.51

Ui

$(100 - [\text{Idle}]) / 100$

### Graph : Ui vs N



From the graph you can identify that as the idle time on the CPU decreases, the utilization increases. This is because as the number of users N in the loadtest increases the more busy the CPU becomes and eventually reaches full saturation at around 50 users. Because this is a single core machine it's not surprising that the machine can't handle the load after 50 users and reaches full capacity.

## X0 vs N

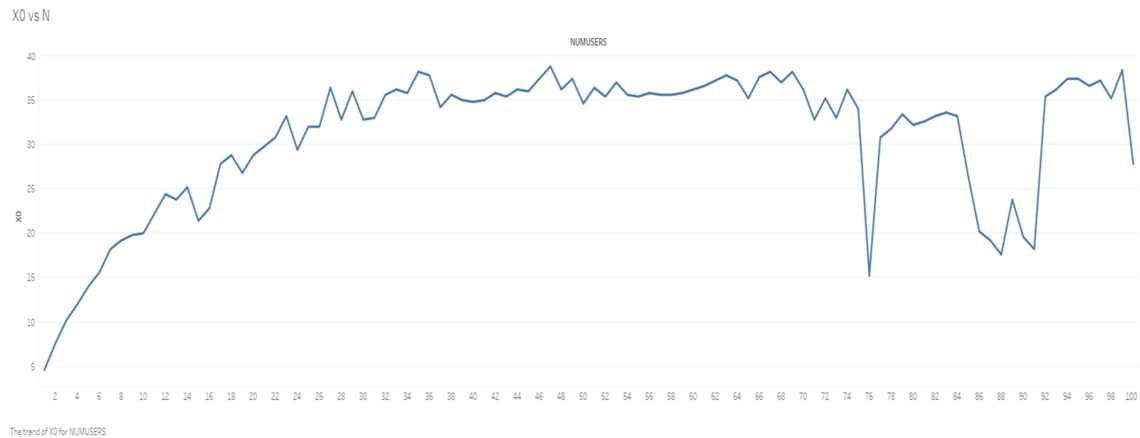
To calculate the system throughput (X0) I used the formula  $X0 = C0/T$  where C0 represented the number of completed transactions in T which could be obtained from the results.dat file in the C0 column. T represented the time each loadtest was ran for, I ran each loadtest for 5 seconds.

I began by creating a calculated field in tableau for T and then a calculated field for X0.



Once these metrics were calculated I loaded the results into a line graph where both X0 and N were my axis.

## Graph : X0 vs N



From the graph you can identify that as the number of users increase the average number of transactions per second also increases. At the start when N is low you can see that system throughput is also low, this is due to low amount of transactions being completed per second (C0). System throughput will remain low as the number of transactions coming in is low. Number of transactions start to increase because number of users N in the loadtest also increase generating more transactions, resulting in an increase in system throughput.

There are a couple of drops in system throughput at loadtest 77, 88, 90, and 91 this might be due to the fact that service demand (Di) on the CPU has increased also at these points. As service demand on the CPU increases, system throughput drops. As the average time spent by a request at loadtest 77, 88, 90, and 91 has increased, this has resulted in the number transactions per second (C0) dropping, resulting in low throughput.

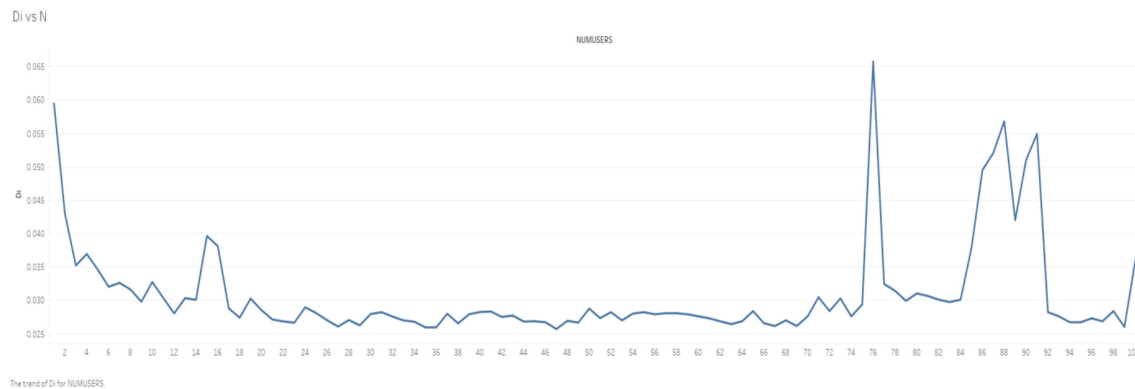
## Di vs N

To calculate the service demand (Di) on the CPU I used the service demand formula  $D_i = U_i / X_0$  where  $U_i$  represented utilization and  $X_0$  throughput. Service demand from the CPU can be generated for each loadtest by formulating the equation each time as the number of users (N) increases.

I began by creating a calculated field for Di on tableau.



## Graph : Di vs N



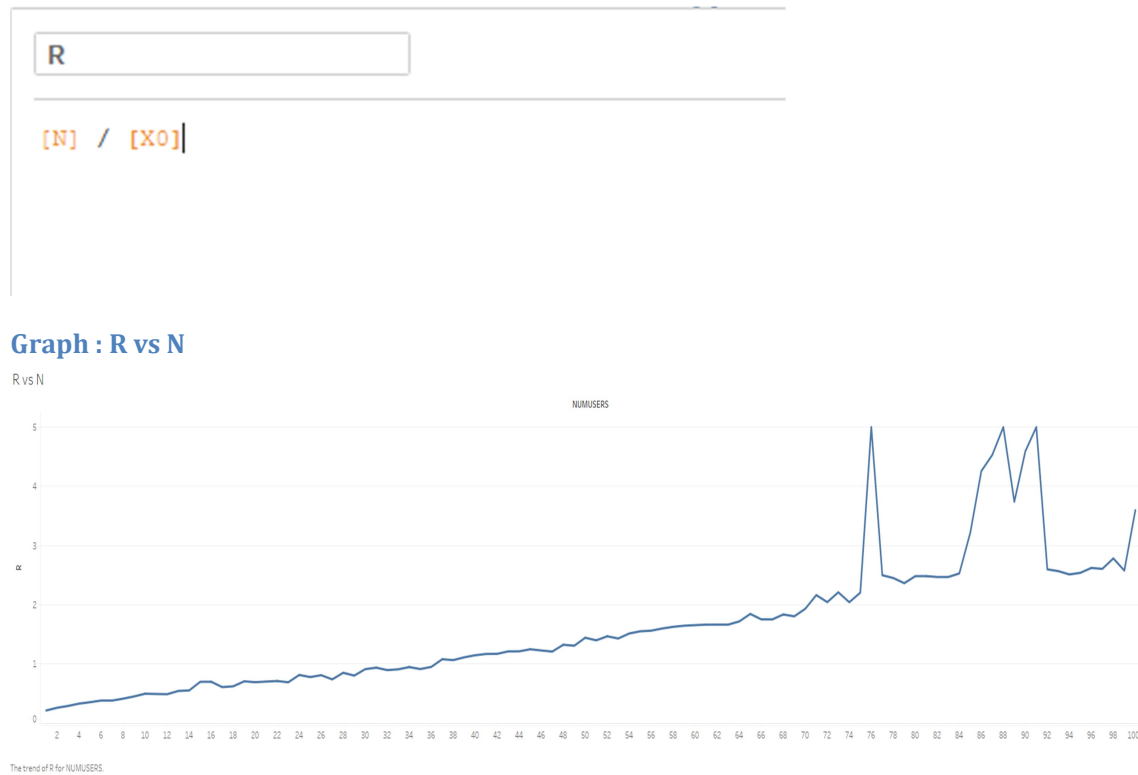
From the graph you can see that as the number of users (N) increase, the average time spent by a request obtaining service from CPU decreases. This is due to the fact that as CPU utilization increases, so too does throughput resulting in less time to complete a request and a low service demand.

The jump in service demand at loadtest 77, 88, 90, and 91 is due to the fact that as throughput drops and utilization has reached full saturation it has resulted in the average time spent by a request obtaining a service from the CPU to take longer i.e. service demand has increased.

## R vs N

To calculate the Response Time (R) I used the formula response time is equal to number of users divided by the throughput  $R = N / X0$ . The response time is equal to the time it takes for a request to complete.

I began by creating a calculated field for R using tableau. I then loaded R and N as my axis into tableau and this created a line graph representing R vs N.



From the graph you can see that the maximum response time of a request is 5 seconds because this is the time I allocated each loadtest to run for ( $T = 5$ ). You can also see from the graph that as N increases so too does the response time to complete each request, this is due to the fact that as the CPU becomes fully saturated the harder it is to handle the load and to complete the number of requests coming in from more users.

The jump in response time with 77, 88, 90, and 91 users could be due to the fact that service demand has also increased at these stages meaning that the longer it has taken for a request to obtain a service from the CPU the longer the response time for that request.



## Appendix A

### Bash Script runtest.sh

```
#!/bin/bash

#Conor Griffin X00111602

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

for i in {1..100}
do

# invoke the loadtest
echo "loadtest $i is running.."
./loadtest $i&

# sleep in background to allow further cmds to execute
echo "loadtest $i is sleeping.."

# while load test is running in background collect cpu utilization
idle=`mpstat 5 1 -o JSON | jq '.sysstat.hosts[0].statistics[0]."cpu-load"[0].idle'`
echo "$idle"

# kill the loadtest
echo "loadtest $i killed.."
pkill loadtest

# write the number of completions in synthetic.dat
C=`cat synthetic.dat | wc -l`

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

# remove synthetic.dat file each time in the loop for new file to be
generated
rm synthetic.dat

done
```