Enterprise Performance Architecture
CA2

Kevin Lambe — X00123430

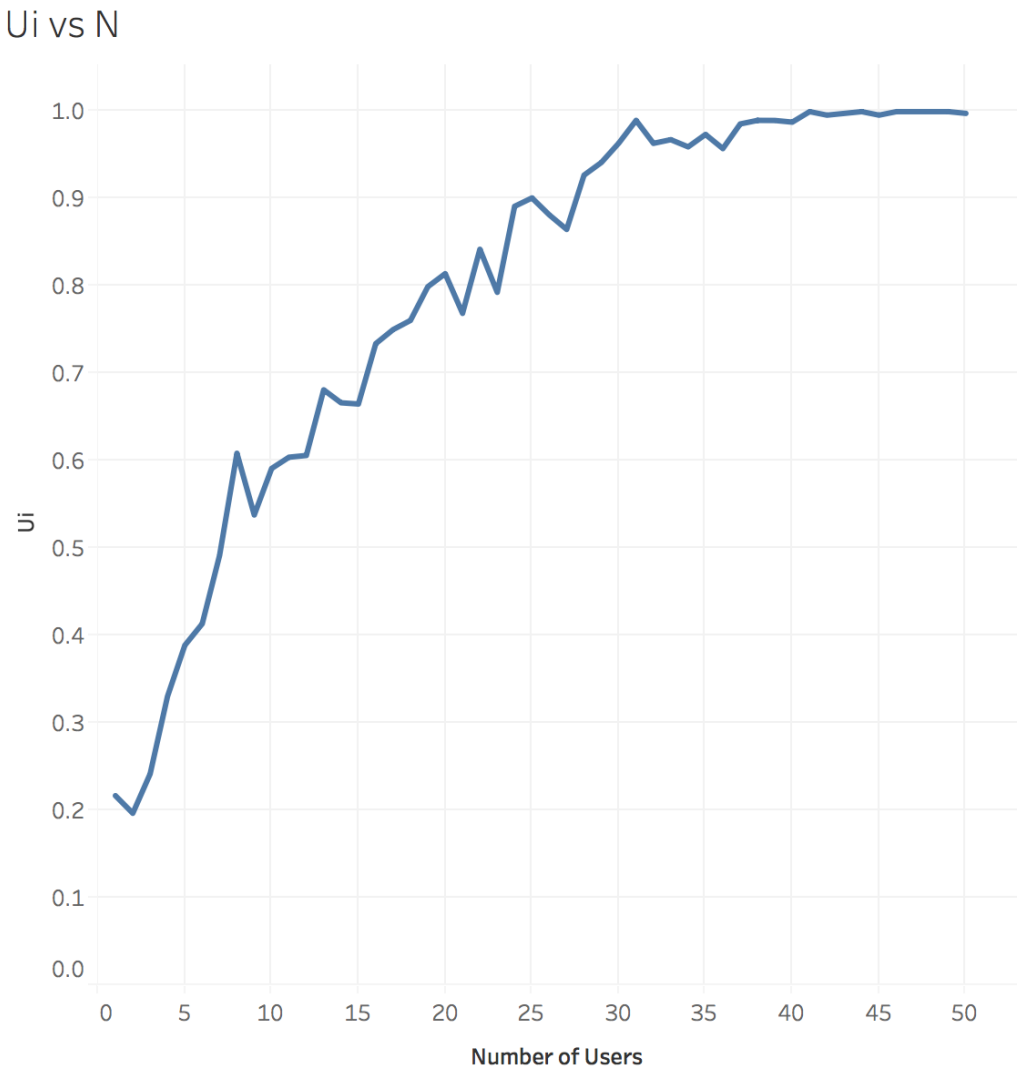
Contents

System Specifications

I ran my script in lab-233 for intervals of 5 seconds on a virtual machine with the fedora 26 image below is the full specifications of the system I used.

Number of Cpu's	1
Cpu Clock Speed	3.30GHz
Cpu model	i5-4590
Ram	5000mb
Operating system	Fedora 26
Hard drive	Hard Disk Drive 5400rpm

Part 1 Ui vs N



N vs. Ui.

Figure 1 Ui vs N

Characteristics and Trends Ui v N

As you can see in figure 1 I have plotted Ui vs N with Ui being the utilisation and N being the number of users of the system. I calculated busy time in my script by subtracting idle time from 100 to leave the percentage of cpu busy time as shown in the appendix.

N is in the range of 1 – 50 as this is part of the specification for the Ca but as you can see from the graph that when we reach 40-50 users the Ui never goes above 1.0 as this represents 100% CPU utilisation.

The graph shows CPU utilisation going from 20% utilisation for 1 user all the way up to 100% utilisation for 40 concurrent users as loadtest.c spins up threads for each user in order to stress test the system with concurrent requests.
The graph is always going to trend upwards and will always hit the limit of 1.0 and never go above, the system may need more concurrent users to reach this limit if the CPU is more powerful.

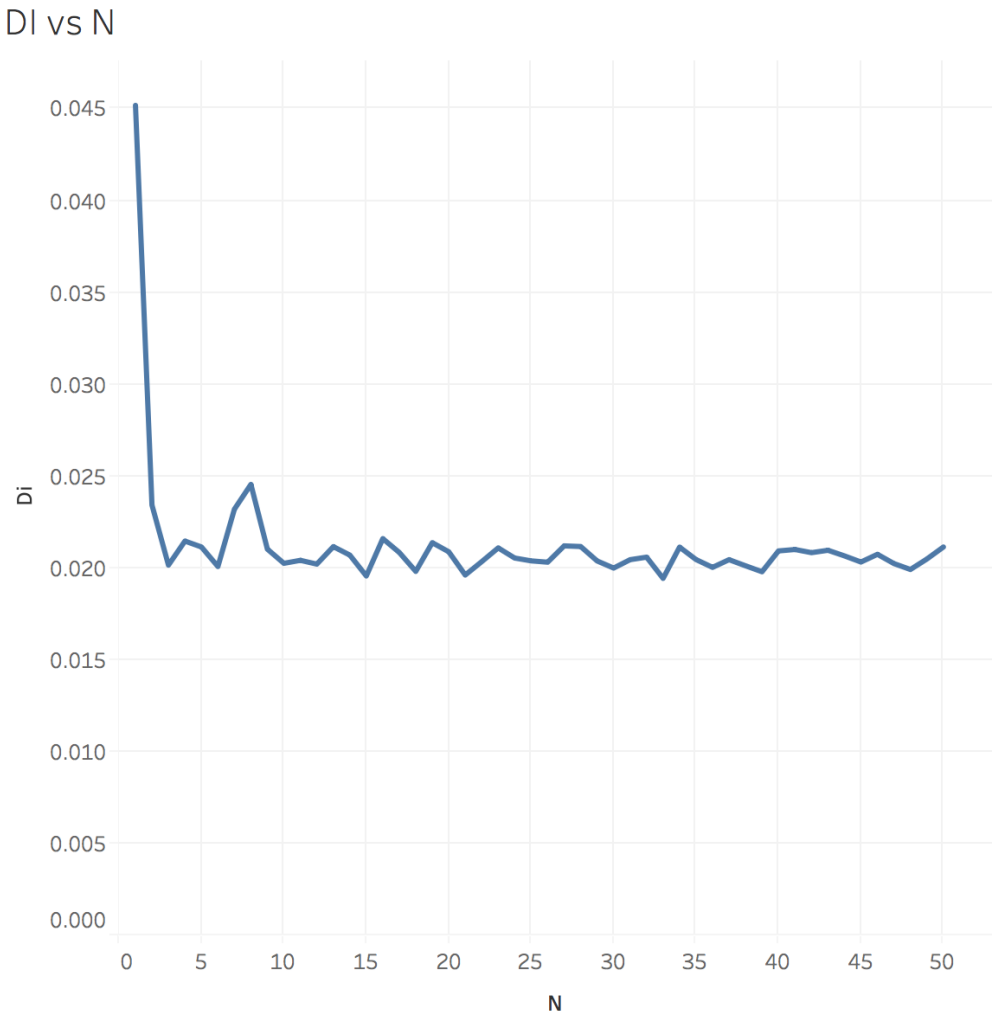
```
        Pearson's product-moment correlation

data:  corr1$ï..N and corr1$Ui
t = 15.441, df = 48, p-value < 2.2e-16
alternative hypothesis: true correlation is not equal to 0
95 percent confidence interval:
 0.8498462 0.9495690
sample estimates:
      cor
0.9123702
```

Figure 2 Ui v N Correlation

Correlation Done in R Ui v N

A Pearson's product moment correlation was carried out on the data to assess if the numbers of users correlated with Ui. Pearson product correlation indicated that there was a high positive correlation between the number of users and the Ui (Figure 2) (r = 0.91).
There was a significant p value p<0.05. our 95%CI [0.85 – 0.95] and we can clearly see that there is a high correlation between our two variables with a significant p value therefore we can say that 83% of one variable has an effect over the other, this does not explain the other 17% and there may be other co-variate factors affecting these two variables.

Part 2 Di vs N



N vs. Di.

Figure 3 Di vs N

Definition

Service Demand

$$D_i = \frac{U_i \times \mathcal{T}}{C_0} = \frac{U_i}{C_0/\mathcal{T}} = \frac{U_i}{X_0} \quad (3)$$

Figure 4 Service Demand

Characteristics and Trends Di v N

For part 2 we had to plot Di vs N with Di being the service demand and I calculated this by using the formula in figure 4 which is utilisation over throughput. As you

can see from the graph in figure 3 the trend from 3 – 50 users are very similar with the variation being the stopping and starting of the test. The mean for the test is .02124 but it would be expected that as utilisation goes up that di would also go up but this isn't the case because utilisation and throughput are tied together so when utilisation goes up throughput also goes up so that Di remains constant.

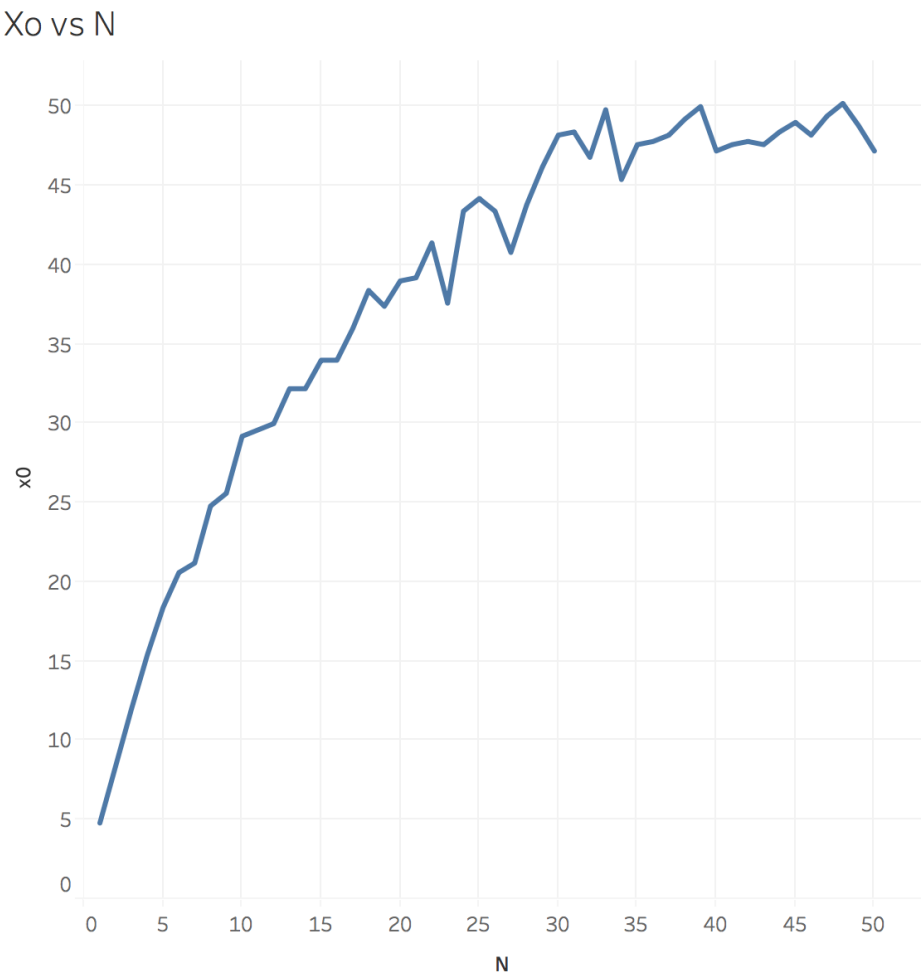
```
        Pearson's product-moment correlation

data:  corr5$ï..N and corr5$Di
t = -2.3453, df = 48, p-value = 0.0232
alternative hypothesis: true correlation is not equal to 0
95 percent confidence interval:
 -0.54990562 -0.04643233
sample estimates:
       cor
-0.320636
```

Figure 5 Correlation Di v N

Correlation Done in R Di v N

A Pearson's product moment correlation was carried out on the data to assess if the numbers of users correlated with Di.
Pearson product correlation indicated that there was a weak negative correlation between the number of users and the Di (Figure 5) (r = -0.32). therefore, 10% of the number of users can be influenced by Di.

Part 3 Xo vs N



N vs. x0.

Figure 6 Xo vs N

Characteristics and Trends Xo v N

For part 3 we had to plot Xo vs N with Xo being the system throughput. To calculate the throughput, I divided the number of competed transactions by the number of seconds the test was run for with the sample time being 5 seconds.
As you can see as the number of users go up so does the throughput for the system with an initial steep increase to a more gradual one and finally the system throughput will hit a ceiling when the CPU utilisation hits the 100% ceiling.
The reason for this is that as more threads are spun up to simulate the number of user's N the cpu utilisation goes up and so does the number of completed transactions with the time being the only constant and since throughput is Co/T then Xo will always increase until it hits a bottleneck with CPU utilisation being

the bottleneck in this case as we saw in figure 1 once N reached 40 users the utilisation reached 100% and the number of completed transactions were very similar.

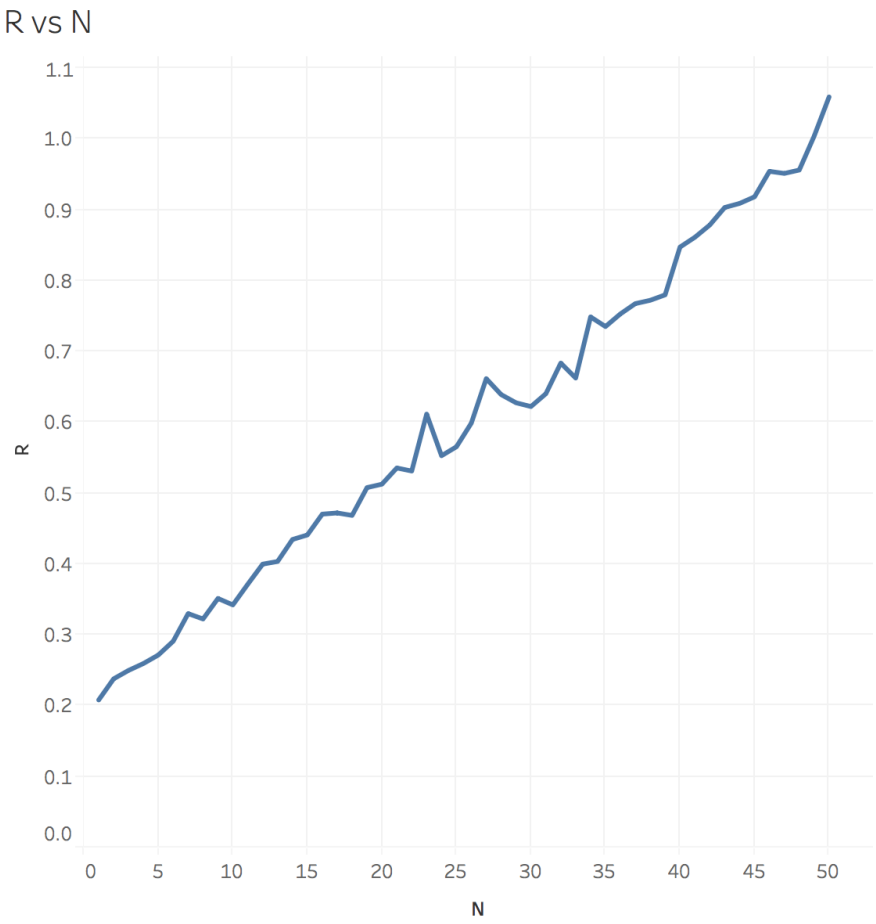
```
        Pearson's product-moment correlation

data:  NEW_DATA$x0 and NEW_DATA$ï..N
t = 14.685, df = 48, p-value < 2.2e-16
alternative hypothesis: true correlation is not equal to 0
95 percent confidence interval:
 0.8366761 0.9448851
sample estimates:
      cor
0.9044007

> |
```

Figure 7 Xo v N

Correlation Done in R Xo v N

A Pearson's product moment correlation was carried out on the data to assess if the numbers of users correlated with Xo.
Pearson product correlation indicated that there was a high positive correlation between the number of users and the Xo (Figure 7) (r = 0.90). there was a significant p value p<0.05. our 95%CI [0.84 – 0.94].
we can clearly see that there is a high correlation between our two variables with a significant p value therefore we can say that 94% of one variable has an effect over the other, this does not explain the other 6% and there may be other co-variate factors affecting these two variables.

Part 4 R vs N



N vs. R.

Figure 8 R vs N

Definition

Little's Law The average number of jobs inside a system is equal to the departure rate times the average time spent in the system.
$N = XR$

Figure 9 Littles Law

Characteristics and Trends R v N

For part 4 we had to plot R vs N with R being the average response time and can be calculated using Littles Law as shown in figure 9. When you get littles law in terms of R it is equal to N / Xo. As you can see from the graph as N increases the value of

R also increases, you may think that since N and Xo are increasing then we should get a similar graph to figure 3 were Di vs N levelled off but this is not the case for R vs N.

The number of users is increasing faster than the throughput of the system and because of this the queue size or R gets larger, as we reach 40 users the throughput of the system has almost hit the ceiling so the value for Xo will level off at 100% but if we were to keep increasing the number of users in the system then the queue time will continue to increase.

```
        Pearson's product-moment correlation

data:  corr6$ï..N and corr6$R
t = 67.398, df = 48, p-value < 2.2e-16
alternative hypothesis: true correlation is not equal to 0
95 percent confidence interval:
 0.9907332 0.9970375
sample estimates:
      cor
0.9947581
```

Figure 10 Correlation R vs N

Correlation Done in R of R v N

A Pearson's product moment correlation was carried out on the data to assess if the numbers of users correlated with R.

Pearson product correlation indicated that there was a very strong positive correlation between the number of users and the R (Figure 10) (r = 0.99).

There was a significant p value p<0.05. our 95%CI [0.99 – 0.99]. we can clearly see that there is a very strong correlation between our two variables with a significant p value therefore we can say that 99% of one variable has an effect over the other, this does not explain the other 1% and there may be other co-variate factors affecting these two variables however, we can clearly see that these two variables are highly correlated and have an effect over each other.

Appendix

```
#!/bin/bash

# x00123430 Kevin Lambe
echo "co    N    CPU usage" >results.dat
for i in {1..50}
do
# do something here
./loadtest $i &
cpu=`mpstat 5 1 -o JSON | jq '100 - .sysstat.hosts[0].statistics[0]."cpu-load"[0].idle'`


#sleep 5
pkill loadtest
cnt=`grep processor /proc/cpuinfo | wc -l`
num=`cat synthetic.dat |wc -l`
echo "Finished run $i co is $num num processors is $cnt"
echo "$num    $i    $cpu" >>results.dat
echo "$num    $i    $cpu"

done
```