Enterprise Application Architecture

Continuous Assessment

X00114388

Dean Flood

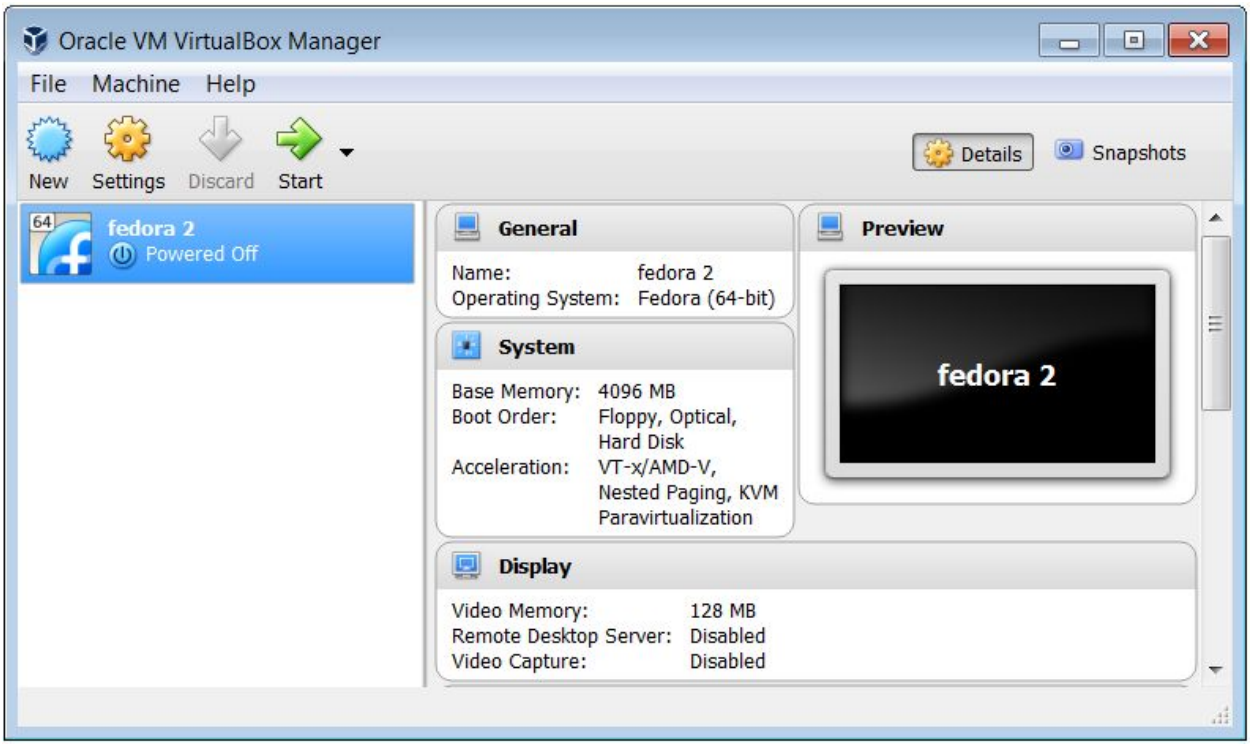
Table of Contents

Github Repo

The source code for this project is hosted at github.com/deanflood/EAA_CA2

Operating System Under Test

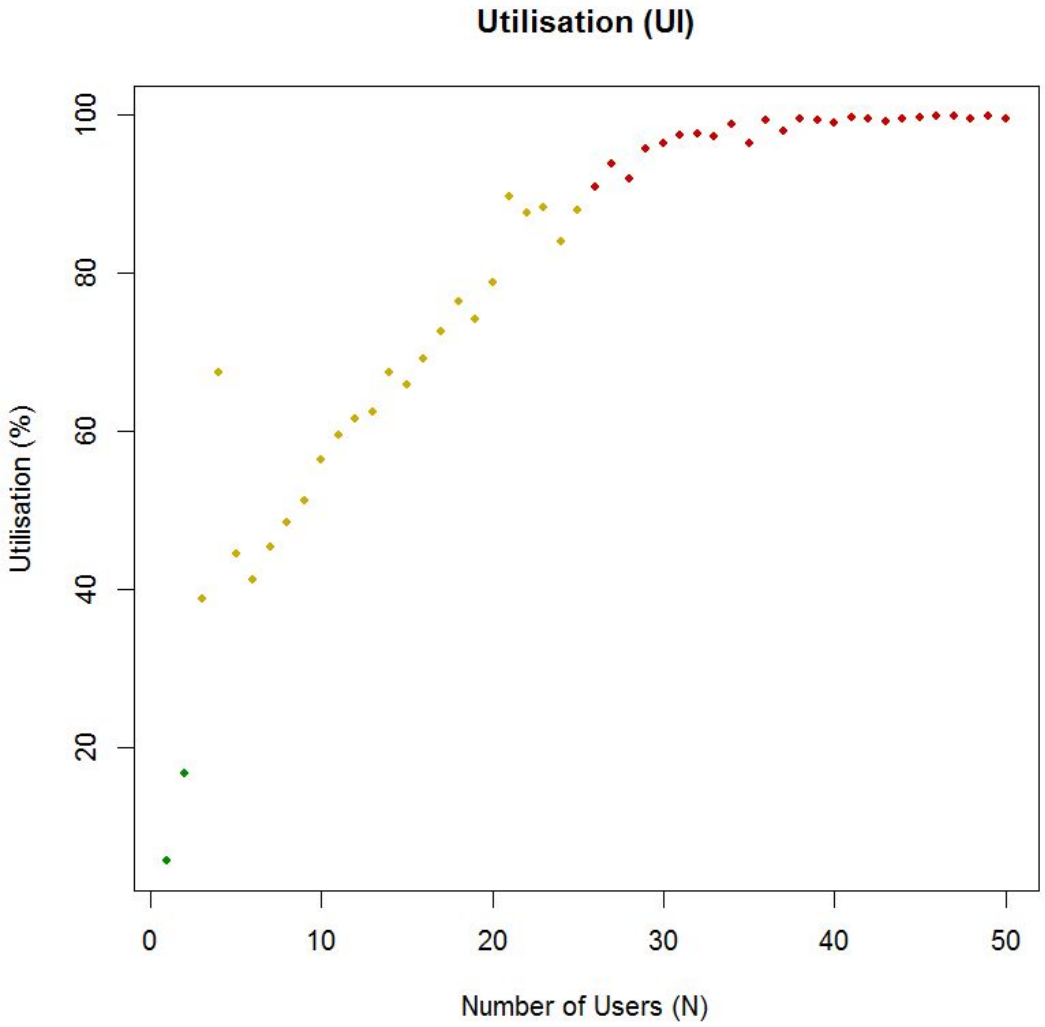
The operating system the tests in this report were ran on was a UNIX based Fedora image running inside Oracle VM VirtualBox. As per the requirements of the CA the image was configured to run on a single core processor with 4096MB of Base Memory. As the image was only able to execute on a single core, it meant that it was easier to maximise the CPU utilisation, allowing the results to be generated easier.



Graph 1 - Utilisation (UI)

This graph shows the expected “knee curve” one would expect when a CPU is put under load. The curve continues to grow steadily until the CPU is under 100% utilisation. As it is not possible to utilise any resource more than 100%, it maxes out, meaning the growth stops and continues at 100% utilisation until the loadtest is stopped. The growth is caused by the increase of the number of users (N) in loadtest.C. This value is fed into the program by the runtest.sh bash script. I have colour coded the nodes to indicate how much of a load the CPU is under.

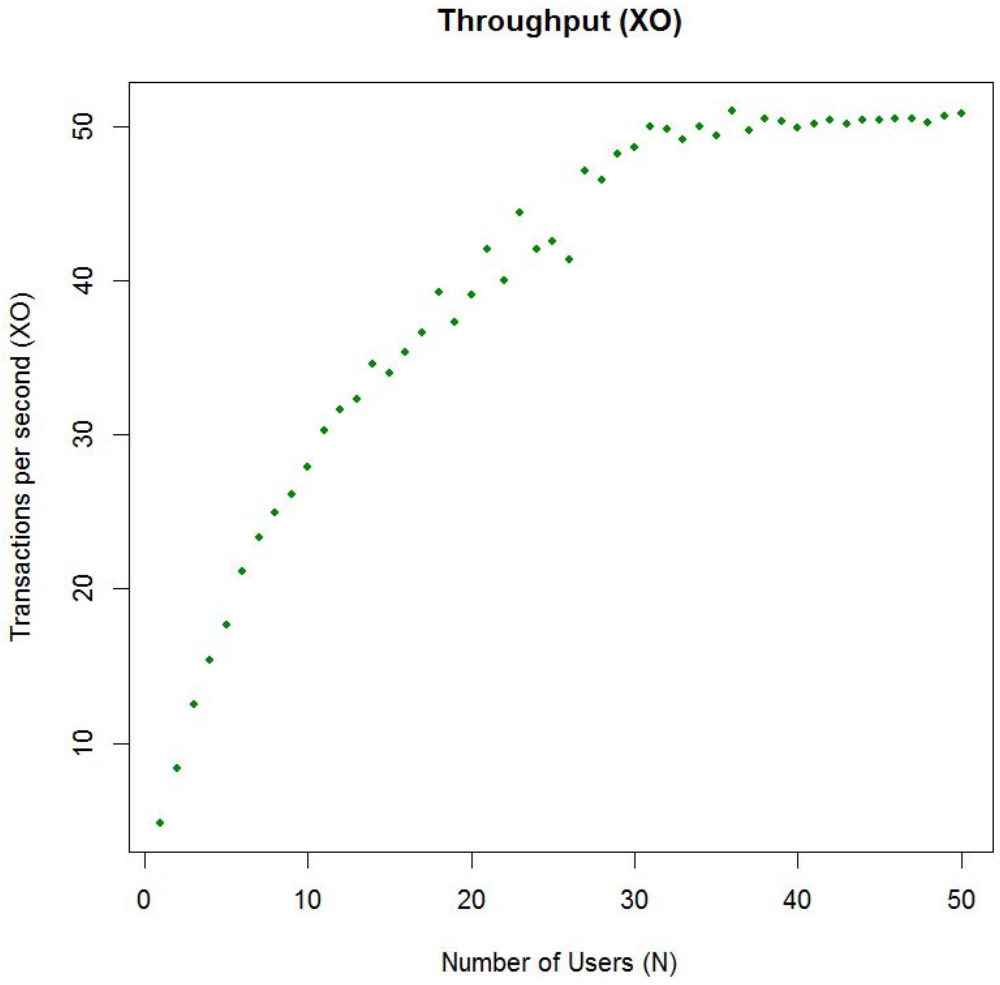
```
summary(UI)
 Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
 5.73   66.20   90.20   79.72   98.98   99.80
```



System Throughput (XO)

System throughput is a measure of the number of transactions completed per second by the system. This metric can be calculated by getting the number of completed transactions and dividing them by the timeframe for which you are measuring them (Co / T). As a result, like Utilisation, the expected trend would be growth as the number of transactions are completed, and then a plateau as the system's throughput is maximised. The results in the expected “knee curve” as below.

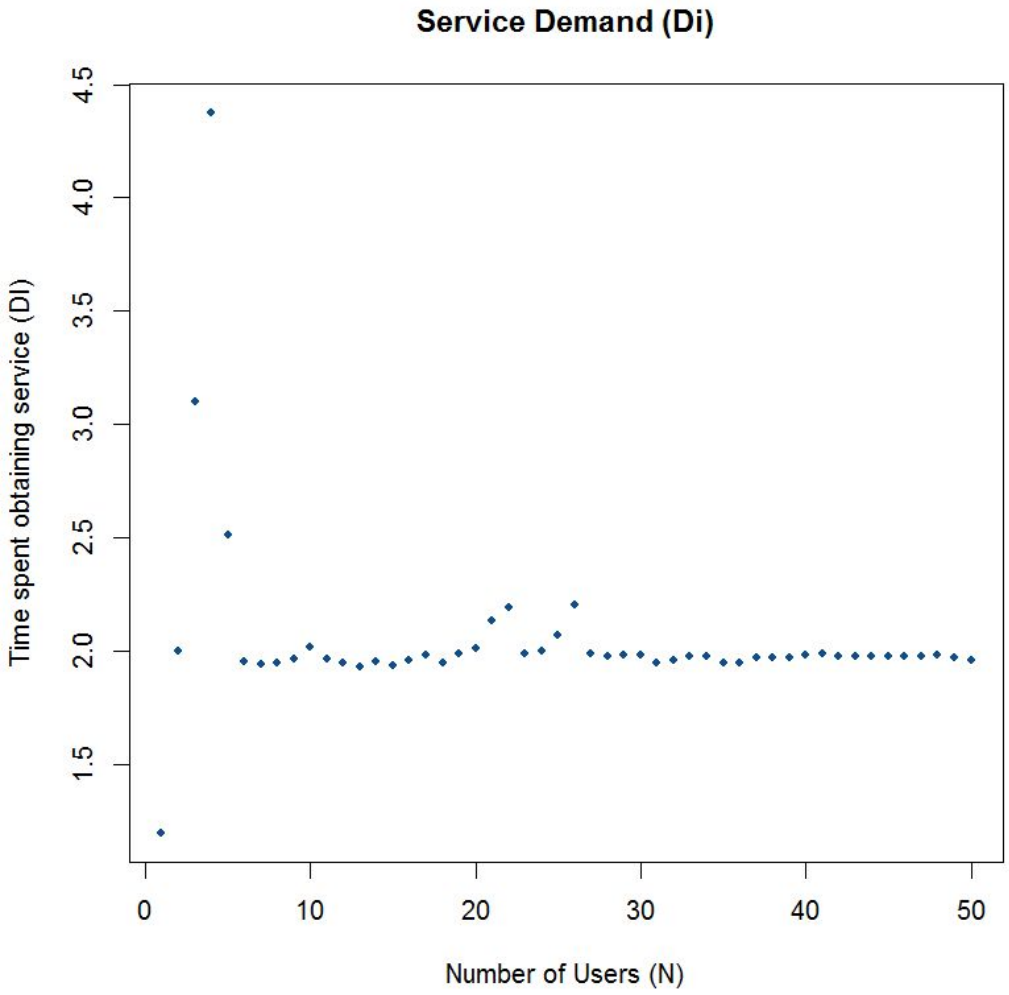
```
summary(XO)
 Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
 4.80   32.72   43.45   39.57   50.08   51.00
```



Service Demand (Di)

The service demand of a resource is defined by the average time spent by a transaction obtaining service from a resource. The interesting thing with this trend is that, unlike the other graphs, the service demand remains static, regardless of how many users are in the load balance test. A few outliers aside, one would expect Di to remain constant, as the time a transaction requires on the CPU remains consistent. While a CPU may have more transactions to process, that does not affect the speed at which it would process a single transaction. As such, the below graph is in line with expectations.

```
summary(DI)
 Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
1.194   1.958   1.976   2.050   1.988   4.377
```



Response Time (R)

Response time is defined by the total amount of time a resource takes to respond to a user's request. The below graph shows that as the number of users increase, the time it takes for the resource (CPU) to respond to that specific request increases. This is in line with expectations, as it one would expect a busy CPU to take longer to respond to a request, as it is more likely to be bust with another request.

```
summary(R)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
0.2083  0.4030  0.5807  0.5822  0.7505  0.9843
```