Assignment 1

Om Shri Prasath

February 2020

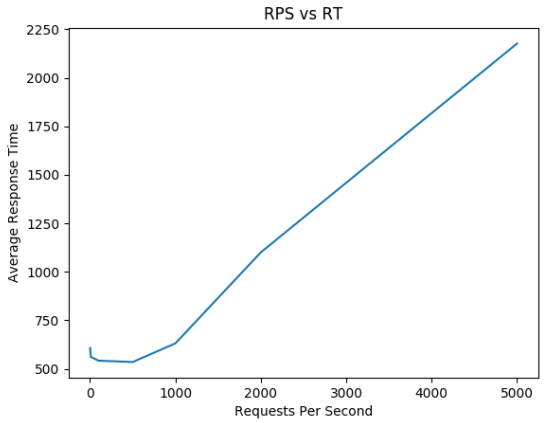
Introduction

The assignment consists of load testing the AWS server with and without auto-scaling enabled and looking at the difference

Without Auto Balancing

First, we give a non CPU-intensive task (reversal of string) to the server and test its response time as the rate of requests increase.

The requests per second versus response time graph is given below :

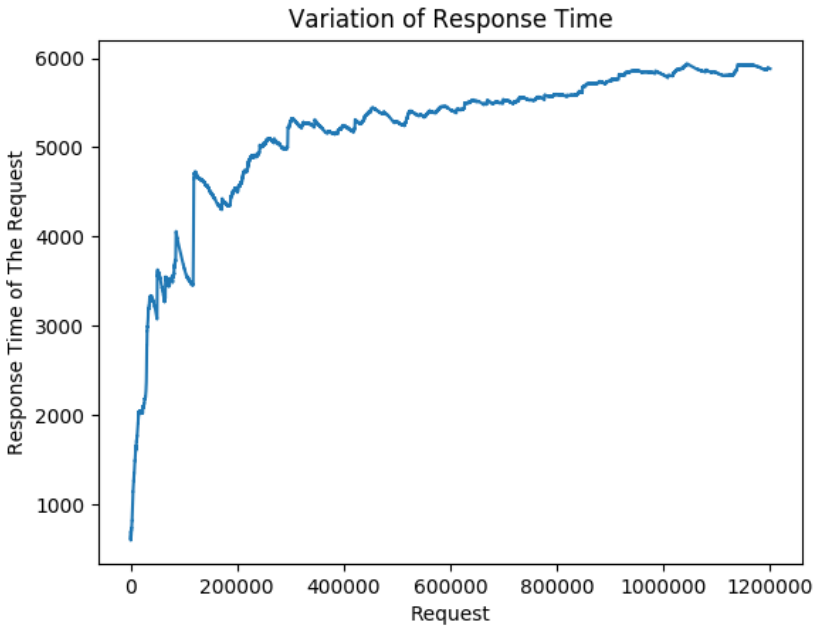


We see that as the number of requests per second increases, the response time also increases. At 5000 requests per second, the server started to send bad responses after waiting for some time. Thus the maximum capacity the server could handle was around 5000 requests per second

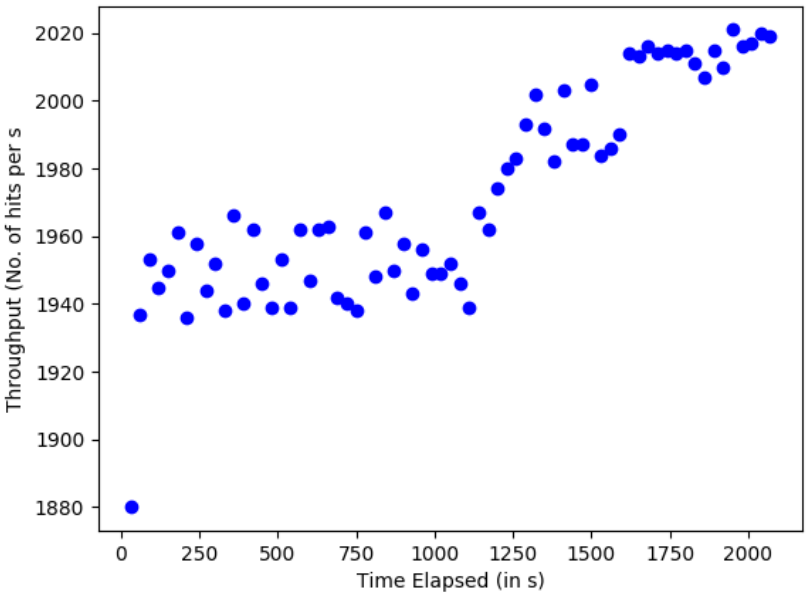
With Autoscaling

Now we set a CPU intensive task (finding the Fibonacci of a random number between 150 and 200), to test the load capacity and adaptability of auto-scaler.

Auto-scaler was set to increase the instances at 10% of CPU usage. Load Balancer (Classic) was set to listen to this Auto-scaling Group We set the request rate per second as 3000 requests per second (to prevent any crashes in load tester). The initial load for the first instance reached 10% quickly, thus a new instance was created. Since the creation of a new instance required CPU Utilization more than 10% CPU, the auto-scaler automatically created another instance, thus the load was balanced between the three instances. The average response time as the requests were made was as follows :



We see that the average response time does not change much, but when we plot the throughput at 30s interval, we can see the following :



As the instances increased, the throughput also increased similarly. This shows that auto-scaling helps in increasing the throughput of the system.

The CPU Utilization, Network Input and Network Output varied as follows :

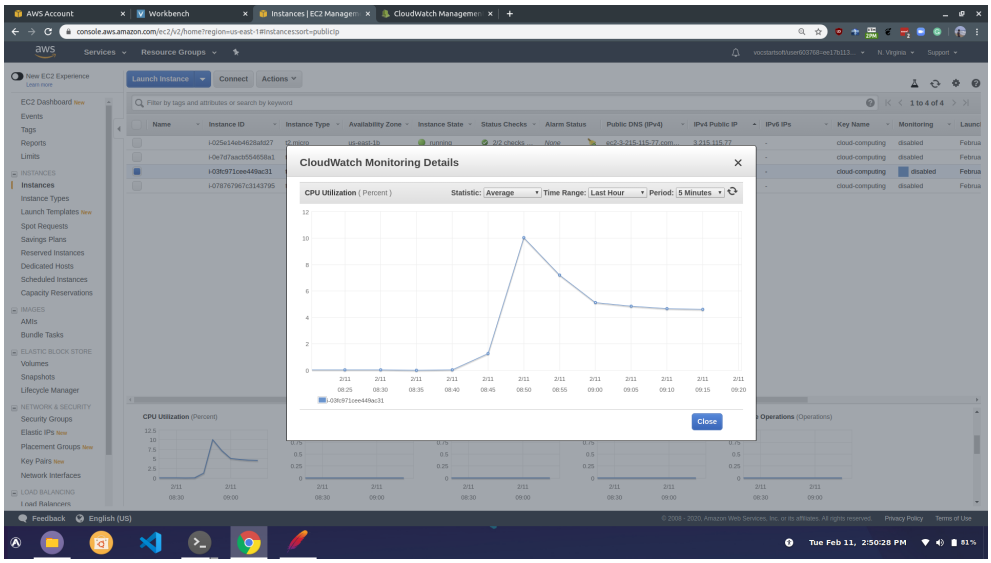


Figure 1: CPU Utilization

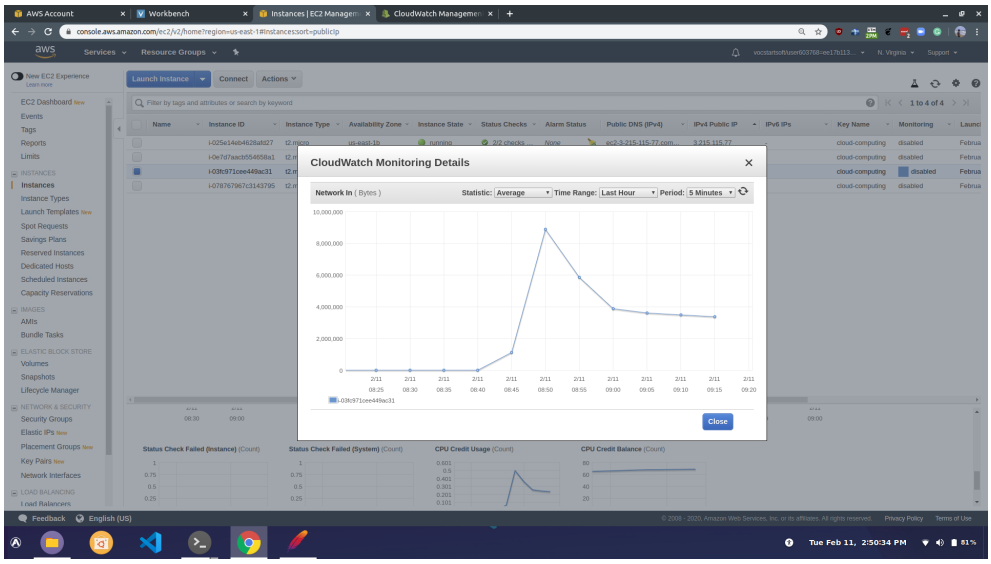


Figure 2: Network Input

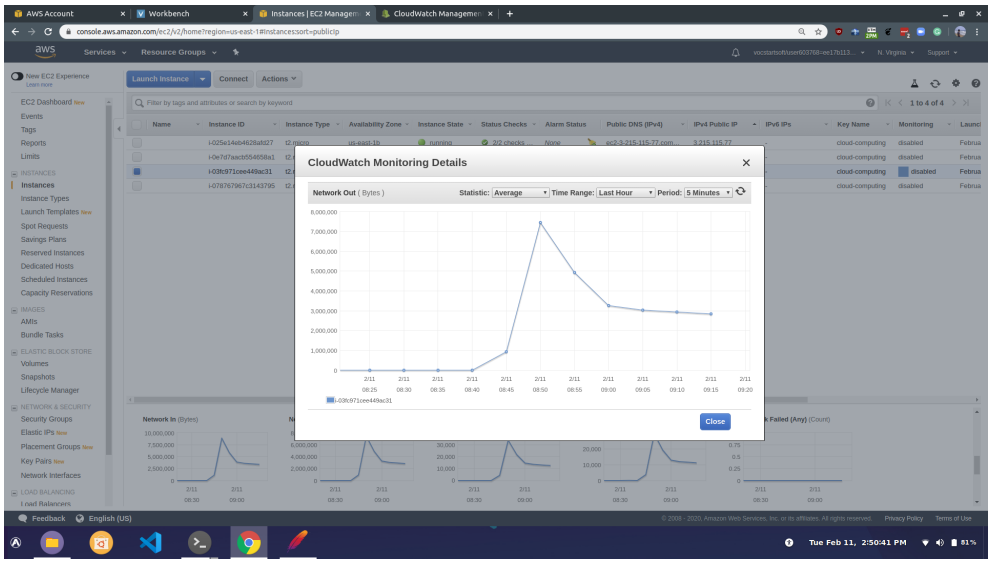


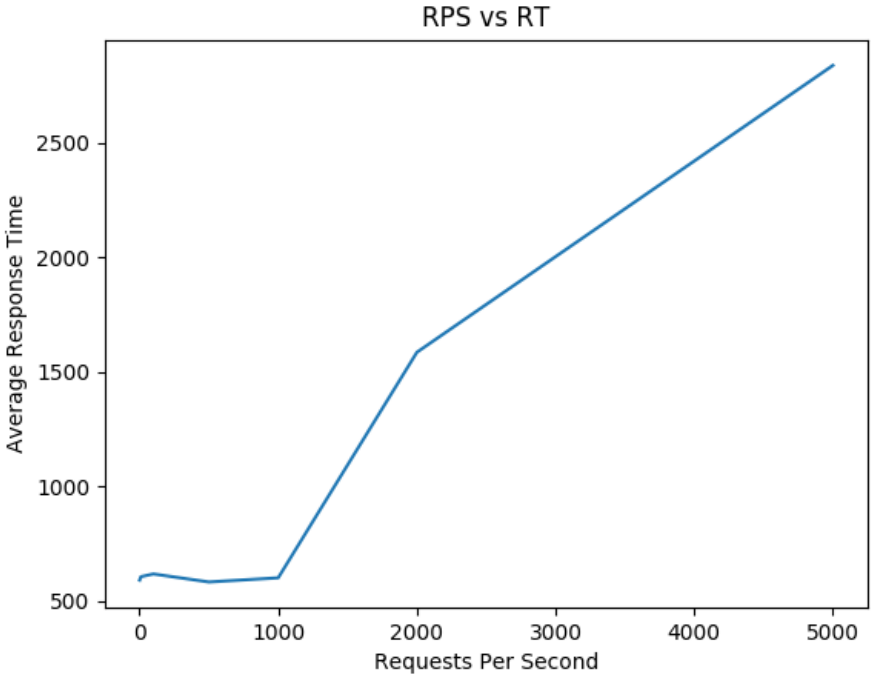
Figure 3: Network Output

We can see that the three graphs follow a similar trend, which can be verified by the fact that the CPU Utilization depends on the amount of requests handled by the computer, which is reflected in the network I/O

Comparison of AWS with GCP

The above requests and testing were run on Amazon AWS cloud service. Now, we have to compare the above with Google Cloud Platform (GCP)

- GCP offers a similar service as Amazon for creating virtual machines called Compute Engine. There is also an option to create machine for specialized tasks(with pre-installed features called App Engine.
- GCP also offers CLI service to control creation of instances unlike AWS. This service is provided through the Google Cloud SDK
- Creation of GCP instance is similar in process to creation of AWS instance, with no major differences.
- One good advantage in GCP is SSH instance can be instantiated directly in browser itself.
- The non-autoscaling part of he test showed the particular result as below :
- We can see that it follows a similar trend to AWS Instances.
- Creating a load balancer(which could be done using the Network Services Option) along with auto-scaling groups (which can be set up in Instance Groups) are also present in GCP, albeit with reduced number of options as compared to AWS.



Conclusion

- Response time increases with request rate per second.
- Auto scaling with load balancer helps reduce load on server by rerouting the requests to similar instances, thus improving throughput.
- CPU Utilization, Network Input and Network Output follow similar trends usually.
- An comparison between GCP and AWS was also made.