**CLOUD COMPUTING ASSIGNMENT (AMAZON EC2)**

-Nitin Nayyar

-Raunak Deshmukh

100% Participation by both the team mates.

1. **Which configuration had the lowest latency for the most computationally expensive requests?**

Linux AMIm3.xlarge

1. **The least expensive requests?**

T2.Micro for x=10

1. **Did the more expensive configurations provide enough performance to offset the added cost?**

No I don’t think so since performance for m3.xlarge has increased only 15 times compared to t2.micro in comparison to 21 times increase in cost and for t2.small there is hardly any significant boost in performance where as the price is doubled.

1. **If you were running a simple site, which configuration would you choose?**

I would prefer t2.micro for small application where there are just a few much computations need to be done on the server.

1. **If you were running a Complex site, which configuration would you choose?**

For Complex site where more number of computations need to be performed I would prefer m3.xlarge since I need to provide better performance to the end users and will require fast machines to do so.

1. **If you have simple site was being hit hard constantly, how much would each request cost?**

Suppose t2.micro can take up to 1000 requests/sec and cost/sec will be .013/3600 dollars/sec and per request will be 0.013/3600\*1000

1. **Which AMI(s) did you use, and/or base your developed AMI off of?**

We used Amazon Linux AMI.

1. **Which HTTP web server did you use?  Did you change the configuration for better performance?  How?**

Apache Webserver and Increased the PHP memory limit from 64MB to 128MB.

1. **What region did you run your instances?  Where was your Apache JMeter installation?**

We used Oregon region for instances t2.Micro, t2.Small and m3.xLarge and N.Virginia for t2.medium. We installed apache j meter on our local machines at our home at 10mbps.

1. **A brief description of what your dynamic web page does.**

We are displaying Fibonacci series equal to the number entered by user in textbox. Post request is made to the server and server side code used to display numbers is coded in PHP.

Graphs:

|  |  |  |
| --- | --- | --- |
| Type | Throughput (Requests/sec) | Cost (Cents/Hr) |
| t2.micro(Oregon) | 20.12 | 1.3 |
| t2.small(Oregon) | 26.23 | 2.6 |
| t2.medium(N.Virginia) | 35.60 | 5.2 |
| m3.xLarge(Oregon) | 402.01 | 28 |

**Conclusion**: m3.xlarge should be preferred over others if our site is complex and serving large user base since its performance is exceptionally better than other instances. However for small applications cost effective instances such as micro or small instances should be preferred.

**The various URL’s if needed:**

t2.micro**:** ec2-54-201-55-219.us-west-2.compute.amazonaws.com

t2.small: ec2-54-201-189-21.us-west-2.compute.amazonaws.com

t2.medium: ec2-52-0-52-208.compute-1.amazonaws.com

m3.xlarge: ec2-54-186-59-240.us-west-2.compute.amazonaws.com

**The GitHub Commit ID:** 8850e0f5cffd211cd8d055ef0e04eca478cb64fb

**https://github.uc.edu/deshmurh/Cloud-Computing-Assgnmt1-RD-NN/tree/8850e0f5cffd211cd8d055ef0e04eca478cb64fb**