CS15-319 / 15-619
Cloud Computing

Recitation 5
February 11th & 13th, 2014
Quiz2 Bugs

• Question 14
  - The question did not specify read/write ratio
  - Read speed: 12x (theoretical)
  - Write Speed: 6x (theoretical)

• Question 22, part3
  - Rounding issue

• Question 23
  - Some students considered leap years

We will manually grade these questions
Project 2.1 Checkpoint Bugs

- Question 4 and Question 5 will be manually graded
  - This happens due to AWS variations in performance
  - We will manually grade Q4 & Q5
  - Please be patient, we will inform you once this process is completed
Announcements

• Do not cheat
  – We will find out
  – The penalties are severe

• Tag your instances

• Provide feedback on OLI

• Post on Piazza:
  – Private: a grading bug
  – Public: general questions
    – Search Piazza and the web before posting
Announcements

• Monitor AWS expenses regularly
  – EMR cost is “on top of” the EC2 cost of instance and EMR cost is fixed per instance type per hour
    • for example, m2.4xlarge EMR cost is $0.42 on-top-of the spot pricing ($0.14)
  – Suggestions
    • Terminate your instance when not in use
      – stop still costs money!
    • Use smaller instances to test your code
    • Use small sample dataset in EMR
    • Decrease the total number of requests when firing up the benchmark
CloudWatch Billing Alert

• You can set up an alert to be notified automatically via e-mail when estimated charges reach a threshold that you specify
• You can use up to 10 alarms and 1,000 e-mail notifications free each month
• **Demo**
• Billing Alarm HOWTO:
  https://piazza.com/class/hq77w6lddnb6wd?cid=502
Last Week

● Content
  – Unit 2: Data Centers
  – Quiz 2 completed
● EC2 and CloudWatch APIs
  – Amazon Command Line
  – AWS SDK for Java
  – AWS SDK for Python
● Vertical Scaling
  – Instance Capacity
Reflection on Last Week

Metric: bandwidth (avg responses/sec), CPU Utilization

- Launchpad instance (m1.medium)
- Freshly launched instance (m1.small/m1.medium/m1.large)
- CPU Utilization
- Benchmark script
- CloudWatch API
Piazza Questions

• mon-get-stats
  – Refused: The security token included in the request is invalid

• Solution:
  – Each time you use the CloudWatch tools (or Amazon EC2 CLI tools) with your instance, you must provide your identity

• How to make sure the instance is running?
  – DescribeInstanceRequest correct
  – instance.getState().getName() wrong
Piazza Questions

- And...you still need to wait for it to initialize
This Week

• UNIT 3: Virtualizing Resources for the Cloud
  – Module 6: Introduction and Motivation
  – Module 7: Virtualization
  – Module 8: Resource Virtualization - CPU
  – Module 9: Resource Virtualization - Memory
  – Module 10: Resource Virtualization – I/O
  – Module 11: Case Study
  – Quiz 3: Virtualizing Resources for the Cloud
This Week

• Introduction and APIs
  – Single Instance Benchmarks

• Elastic Load Balancing (2 modules)
  – Elastic Load Balancer
  – Static Load Benchmarking
Project Module

Auto Scaling Group

Users

Monitor (CloudWatch)
Vertical Scaling vs. Horizontal Scaling

• Vertical Scaling Limitations
  – Can only increase the capacity to a limit
  – When scaling, need to transfer data, have to reboot

• Solution: Horizontal Scaling (add more resources)
Horizontal Scaling

Uneven distribution of load!

- Server 1
- Server 2
- Server 3
- Server 4

- CPU utilization, memory utilization…
- Available capacity
If Server2 goes down, no fall back is configured.
If load goes down, we need to change the number of servers

- CPU utilization, memory utilization…
- Available capacity
What You Need

• Make sure that workload is even on each server
• Do not assign load to servers that are down
• Increase/Remove servers according to the changing load

How does AWS help solve these problems?

Managed group of servers
AWS Elastic Load Balancer (ELB)

- ELB is a gateway that acts as a router interface and sends incoming requests to multiple EC2 Instances sitting behind it
- Distribute requests from clients to all servers equally

ELB Features

• Using ELB, you can distribute incoming traffic across your Amazon EC2 instances in multiple Availability Zones (redundancy within the same region)

• ELB can detect the health of Amazon EC2 instances. When it detects unhealthy instances, it spreads the load to other healthy instances

• ELB can offer integration with Auto Scaling to ensure that you can meet varying levels of traffic levels without requiring manual intervention
ELB Case

- **Airbnb** is a community that allows property owners and travelers to connect with each other.

- Airbnb is using Elastic Load Balancing, which automatically distributes incoming traffic between multiple Amazon EC2 instances. As a result, it saves a lot of operation expenditures.

- Within 4 years, it now has hundreds of employees in nearly 25,000 cities in 192 countries.
When scaling and load balancing, Airbnb employs ELB!
Project Module

- Elastic Load Balancing
  - Elastic Load Balancer
  - Static Load Benchmarking
# Upcoming Deadlines

## Project 2:

<table>
<thead>
<tr>
<th>Project 2</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Introduction and APIs</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Single Instance Benchmarks</td>
<td>Checkpoint</td>
<td>Available Now Due 2/9/14 11:59 PM</td>
</tr>
<tr>
<td><strong>Elastic Load Balancing</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Elastic Load Balancer</td>
<td>Checkpoint</td>
<td>Due 2/16/14 11:59 PM</td>
</tr>
<tr>
<td>Static Load Benchmarking</td>
<td>Checkpoint</td>
<td>Due 2/16/14 11:59 PM</td>
</tr>
</tbody>
</table>

## Unit 3:

<table>
<thead>
<tr>
<th>UNIT 3: Virtualizing Resources for the Cloud</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Module 6: Introduction and Motivation</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Module 7: Virtualization</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Discussions

- Questions and Comments
Demo

• Create a new Load Balancer manually. (Project 2 Module 2)

• Create a new Load Balancer and attach the instance to ELB programmatically using Python or any other programming language of your own choice. (Project 2 Module 3)
  • Create an ELB
  • Provision an instance & attach it to ELB
  • Wait some time for it to be in service