Recitation 4
September 17th & 20th, 2013
Announcements

• Encounter a general bug:
  – Post on Piazza

• Encounter a grading bug:
  – Post Privately on Piazza

• Post feedback on OLI
Unit 2: Data Centers

- Data Centers Unit
  - Module 3: Data Center Trends
  - Module 4: Data Center Components

- Read and complete:
  - Module 5: Design Considerations
  - Unit 2: Checkpoint Quiz

- Timed Quiz (120 minutes), due 19th September, 2013
Project 1 Student Progress

• Introduction to Big Data:
  – Sequential Analysis: Average is: 98%
  – Elastic MapReduce: 96%
Piazza Questions

• Elastic MapReduce Billing Question
  – **Normalized Hours (Elastic MapReduce)**

<table>
<thead>
<tr>
<th>Date</th>
<th>Elapsed Time</th>
<th>Normalized Instance Hours</th>
</tr>
</thead>
<tbody>
<tr>
<td>11:59 EDT</td>
<td>1 hour 46 minutes</td>
<td>40</td>
</tr>
</tbody>
</table>

- 1 hour of m1.small = 1 hour normalized compute time
- 1 hour of m1.medium = 2 hours normalized compute time
- 1 hour of m1.large = 4 hours normalized compute time
- 1 hour of m1.xlarge = 8 hours normalized compute time
- 1 hour of c1.medium = 2 hours normalized compute time
Piazza Questions

- Elastic MapReduce Billing Question
  - [Elastic MapReduce Pricing](#)

<table>
<thead>
<tr>
<th>Region: US East (N. Virginia)</th>
<th>Amazon EC2 Price</th>
<th>Amazon Elastic MapReduce Price</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Standard On-Demand Instances</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Small (Default)</td>
<td>$0.06 per hour</td>
<td>$0.015 per hour</td>
</tr>
<tr>
<td>Medium</td>
<td>$0.12 per hour</td>
<td>$0.03 per hour</td>
</tr>
<tr>
<td>Large</td>
<td>$0.24 per hour</td>
<td>$0.06 per hour</td>
</tr>
<tr>
<td>Extra Large</td>
<td>$0.48 per hour</td>
<td>$0.12 per hour</td>
</tr>
</tbody>
</table>
Piazza Questions

- Elastic MapReduce Debugging

---

### Debug a Job Flow

**Job Flow:** My Job Flow (j-36Q4Q4B3WQYNF)

- **Job flow failed with reason:** Shut down as step failed

#### Steps → Jobs → Tasks → Task Attempts

<table>
<thead>
<tr>
<th>Step</th>
<th>Name</th>
<th>State</th>
<th>Start Time</th>
<th>Log Files</th>
<th>Actions</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Setup Hadoop Debugging</td>
<td>COMPLETED</td>
<td>2013-09-10 09:57 EDT</td>
<td>controller</td>
<td>stderr</td>
</tr>
<tr>
<td>2</td>
<td>Streaming Job</td>
<td>FAILED</td>
<td>2013-09-10 09:57 EDT</td>
<td>controller</td>
<td>stderr</td>
</tr>
</tbody>
</table>

---

### Debug a Job Flow

**Job Flow:** My Job Flow (j-36Q4Q4B3WQYNF)

View logs for steps, Hadoop jobs, tasks, and task attempts.

#### Steps → Jobs → Tasks → Task Attempts

<table>
<thead>
<tr>
<th>Job</th>
<th>Step</th>
<th>State</th>
<th>Start Time</th>
<th>Actions</th>
</tr>
</thead>
<tbody>
<tr>
<td>job_201309101355_0001</td>
<td>2</td>
<td>FAILED</td>
<td>2013-09-10 09:58 EDT</td>
<td>View Tasks</td>
</tr>
</tbody>
</table>
Piazza Questions

• Elastic MapReduce Debugging

Hadoop Job: job_201309101355_0001
Task Summary: 33 Total Tasks - 0 Completed, 0 Running, 5 Failed, 0 Pending, 28 Cancelled.

Steps → Jobs → Tasks → Task Attempts

<table>
<thead>
<tr>
<th>Task</th>
<th>Type</th>
<th>Job</th>
<th>State</th>
<th>Start Time</th>
<th>Elapsed Time</th>
<th>Actions</th>
</tr>
</thead>
<tbody>
<tr>
<td>m_000009</td>
<td>map</td>
<td>201309101355_0001</td>
<td>FAILED</td>
<td>2013-09-10 09:58 EDT</td>
<td>0 hours 1 minute</td>
<td>View Attempts</td>
</tr>
<tr>
<td>m_000008</td>
<td>map</td>
<td>201309101355_0001</td>
<td>FAILED</td>
<td>2013-09-10 09:58 EDT</td>
<td>0 hours 1 minute</td>
<td>View Attempts</td>
</tr>
<tr>
<td>m_000002</td>
<td>map</td>
<td>201309101355_0001</td>
<td>FAILED</td>
<td>2013-09-10 09:58 EDT</td>
<td>0 hours 1 minute</td>
<td>View Attempts</td>
</tr>
<tr>
<td>m_000001</td>
<td>map</td>
<td>201309101355_0001</td>
<td>FAILED</td>
<td>2013-09-10 09:58 EDT</td>
<td>0 hours 1 minute</td>
<td>View Attempts</td>
</tr>
</tbody>
</table>

View logs for steps, Hadoop jobs, tasks, and task attempts.

Steps → Jobs → Tasks → Task Attempts

<table>
<thead>
<tr>
<th>Attempt</th>
<th>Job</th>
<th>Task</th>
<th>Type</th>
<th>State</th>
<th>Start Time</th>
<th>Log Files</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>201309101355_0001</td>
<td>m_000000</td>
<td>map</td>
<td>FAILED</td>
<td>2013-09-10 09:59 EDT</td>
<td>stderr</td>
</tr>
<tr>
<td>2</td>
<td>201309101355_0001</td>
<td>m_000000</td>
<td>map</td>
<td>FAILED</td>
<td>2013-09-10 09:59 EDT</td>
<td>stderr</td>
</tr>
<tr>
<td>1</td>
<td>201309101355_0001</td>
<td>m_000000</td>
<td>map</td>
<td>FAILED</td>
<td>2013-09-10 09:58 EDT</td>
<td>stderr</td>
</tr>
<tr>
<td>0</td>
<td>201309101355_0001</td>
<td>m_000000</td>
<td>map</td>
<td>FAILED</td>
<td>2013-09-10 09:58 EDT</td>
<td>stderr</td>
</tr>
</tbody>
</table>
Piazza Questions

• Elastic MapReduce Debugging

```
/mnt/var/lib/hadoop/mapred/taskTracker/hadoop/jobcache/job_201309101355_0001/attempt_201309101355
/mnt/var/lib/hadoop/mapred/taskTracker/hadoop/jobcache/job_201309101355_0001/attempt_201309101355
unexpected token `(':
/mnt/var/lib/hadoop/mapred/taskTracker/hadoop/jobcache/job_201309101355_0001/attempt_201309101355
is_in_filtered_title(line_item_list):
java.lang.RuntimeException: PipeMapRed.waitOutputThreads(): subprocess failed with code 2
 at org.apache.hadoop.streaming.PipeMapRed.waitOutputThreads(PipeMapRed.java:372)
 at org.apache.hadoop.streaming.PipeMapRed.mapRedFinished(PipeMapRed.java:586)
 at org.apache.hadoop.streaming.PipeMapper.map(PipeMapper.java:125)
 at org.apache.hadoop.mapred.MapRunner.run(MapRunner.java:50)
 at org.apache.hadoop.streaming.PipeMapRunner.run(PipeMapRunner.java:36)
```
Piazza Traffic

Posts Per Day

Total Questions Asked
OLI Feedback

- Students want to know more about MapReduce
Scenario

- Online Photo Verification, *PixVerify*
Current Infrastructure

User

Application Server

Database Server
Cloud Infrastructure

Instance Types

Small
Medium
Large

Middleware (Virtualization)

Bare Metal Resources
Maximize Requests per Dollar

Instance Types

Small

Medium

Large

\[ \times \text{REQUESTS} \quad \div \text{COST A} \quad \times \text{REQUESTS} \quad \div \text{COST B} \quad \times \text{REQUESTS} \quad \div \text{COST C} \]

**GOAL:** MAXIMIZE (REQUESTS/DOLLAR)
Load Testing Request & Response Flow

Launchpad

Benchmark Script

Small

Medium

Large
Provisioning Instances

- Manual Provisioning: Online Interface

![Request Instances Wizard](image)

Provide the details for your instance(s). You may also decide whether you want to launch your instances as "on-demand" or "spot" instances.

**Number of Instances:** 1  
**Instance Type:** T1 Micro (t1.micro, 613 MiB)

Launch as an EBS-Optimized instance (additional charges apply):

- Not supported for this instance type

**Launch Instances**

EC2 Instances let you pay for compute capacity by the hour with no long term commitments. This transforms what are commonly large fixed costs into much smaller variable costs.

**Launch into:**
- EC2-Classic  
- EC2-VPC

**Availability Zone:** No Preference
Amazon APIs

• Provisioning & Monitoring:
  – EC2 API
  – CloudWatch API

• Supported APIs
  – Command Line Interface API Tools
  – AWS SDK for Java
  – AWS SDK for Python
Provisioning and Monitoring

Launchpad
(Simulate User Load)

Your Script/Program
- Provision Test Instance
- Run Benchmark Script
- CloudWatch to retrieve CPUUtilization
- Store Benchmark Output into a file
- Terminate Test Instance
- Repeat for Other Test Instances
Demo Outline

• 1. Launch an instance using command line
   – Set up command line
   – Launch an instance
   – CloudWatch command line

• 2. Launch an instance using Java API
   – Set up in eclipse
   – Amazon EC2 API
   – CloudWatch class
Setup ec2 command line tool

• Set JAVA_HOME environment variable
  – Slightly different in Mac OS if you would like to use the command line tool locally.
• Set EC2_HOME & system PATH variable
• Tell system your access key and secret key
• Set private key & certificate
• Verify
  – ec2-describe-regions
Launch an instance

• ec2-run-instances ami_id [-n instance count] [-k keypair] [--instance-type instance_type] [--availability-zone zone]

• http://docs.aws.amazon.com/AWSEC2/latest/CommandLineReference/ApiReference-cmd-RunInstances.html
Cloud Watch Command Line

• Download cloud watch

• Fill your access key and secret key in
  `$AWS_CLOUDWATCH_HOME/credential-file-path.template`

• You will have to find out how to use this command line
  – Hint: `mon-get-stats`
Set up AWS Toolkit for Eclipse

• Plug-in for the Eclipse Java IDE
• Help -> Install New Software
• Enter ‘http://aws.amazon.com/eclipse’ in ‘Work with’
• Select ‘AWS Toolkit for Eclipse’
• http://aws.amazon.com/eclipse/
Amazon EC2 API

- Load AWS Credentials
  - BasicAWSCredentials
- Create Amazon EC2 Client
  - AmazonEC2Client
- Create and Configure Instance Request
  - RunInstancesRequest
- Launch Instance
  - RunInstanceResult
Amazon CloudWatch API

• Load AWS Credentials
  – BasicAWSCredentials
• Create Amazon CloudWatch Client
  – AmazonCloudWatchClient
• Create and Configure Metric Request
  – GetMetricStatisticsRequest
• Get Metric Result
  – GetMetricStatisticsResult
• http://aws.amazon.com/sdkforjava/
Discussion

• Your questions...
# Upcoming Deadlines

- **Unit 2:** **Timed Quiz (120 minutes)**
  - **UNIT 2: Data Centers**
    - **Module 3:** Data Center Trends
    - **Module 4:** Data Center Components
    - **Module 5:** Design Considerations
  - **Quiz 2: Data Centers**
    - **Checkpoint**
  - **Available 9/16/13 12:01 AM**
  - **Due 9/19/13 11:59 PM**

- **Project 2:**
  - **Project 2**
    - **Introduction and APIs**
    - **Single Instance Benchmarks**
      - **Checkpoint**
      - **Available Now**
      - **Due 9/22/13 11:59 PM**