CS15-319 / 15-619
Cloud Computing

Recitation 12
November 11\textsuperscript{th} and 13\textsuperscript{th}, 2014
Announcements

• Encounter a general bug:
  – Post on Piazza
• Encounter a grading bug:
  – Post Privately on Piazza
• Don’t ask if my answer is correct
• Don’t post code on Piazza
• Search before posting
• Post feedback on OLI
Piazza Questions

• Load balancer sending load only to one instance @2065

• Solution:
  1. Enable multiple threads for the YCSB benchmark;
  2. Add db.driver=com.mysql.jdbc.Driver to db.properties file. (Thanks to Garauv Khunger)
Project 3, Module 5 Reflections

• When to use DynamoDB:
  – Required throughput is determined (steady arrival rate);
  – Easier to implement and scale;
  – Enough budget
    • Charged by provisioned throughput capacity.

• When to use RDS:
  – Automated database backup and maintenance;
  – Software upgrades.

• When to use custom MySQL:
  – Lots of flexibility;
  – Low cost;
  – More prone to human errors.
Module to Read

• UNIT 5: Distributed Programming and Analytics Engines for the Cloud
  – Module 16: Introduction to Distributed Programming for the Cloud
  – Module 17: Distributed Analytics Engines for the Cloud: MapReduce
  – Module 18: Distributed Analytics Engines for the Cloud: Pregel
  – Module 19: Distributed Analytics Engines for the Cloud: GraphLab
Project 4

• MapReduce
  – Hadoop MapReduce

• Input Text Predictor: NGram Generation
  – NGram Generation

• Input Text Predictor: Language Model and User Interface
  – Language Model Generation
Google

• Inverted index
  – Word -> \{doc1, doc2, …\}
• PageRank
Google

• Google Instant
  – Input text predictor

  
  
  – Generate a list of phrases in a text corpus with their corresponding counts
  – Rank the probability
MapReduce Reflection on Project 1

• The idea of MapReduce

How many times does the word “Apple” appear in these books?

I heard 6 “Apple”s!
MapReduce Reflection on Project 1

• The idea of MapReduce

You Don’t!

How Do I know Who is the “Apple” Man?
MapReduce Reflection on Project 1

• The idea of MapReduce

Mapper

Map Phase

Orange,1
Blueberry,1
Blueberry,1
Apple,1

Apple,1
Apple,1
Apple,1
Orange,1

Apple,1
Apple,1
Orange,1
Blueberry,1

Reducer

Reduce Phase

Magic Box (Shuffle, sort, merge)

Orange?

Apple?

Blueberry?
MapReduce This Week

• The idea of MapReduce

Map Phase

Orange, 1
Blueberry, 1
Blueberry, 1
Apple, 1
Apple, 1
Apple, 1
Orange, 1
Apple, 1
Apple, 1
Orange, 1
Blueberry, 1

Reduce Phase

Black Box (Shuffle, sort, merge)

Jar instead of streaming

Orange?
Apple?
Blueberry?
MapReduce

• Mapper (default)
  – Input: **key-value pairs**
    • **Key:** byte offset of the line;
    • **Value:** the text content of the line;
  – Output: **key-value pairs**
    • **Key:** specified by your program;
    • **Value:** specified by your program based on what content you expect the reducer to receive as a list.

(k1,v1) -> Mapper -> (k2,v2)
MapReduce

• Reducer
  – Input: **key-value pairs**
    • A list of values for each key output from the mapper;
  – Output: **key-value pairs**
    • The desired result from your aggregation;

(k2,list(v2)) -> Reducer -> (k3,v3)
Hadoop

• Apache Hadoop
  – A framework for running applications on a large cluster of commodity hardware
  – Implements the MapReduce computational paradigm
  – Uses HDFS for data storage
  – Engineers with little knowledge of distributed computing can finish the code in a short period

• MapReduce
  – A programming model for processing large data sets using a parallel distributed algorithm
HDFS

• Hadoop Distributed File System
• Open source version of Google File System
HDFS

• Paper

• Purpose
  – Implemented for running Hadoop’s MapReduce applications with distributed storage
  – An open-source framework which can be used by different clients with different needs
Typical MapReduce Job

• Simplistic view of a MapReduce job

![Diagram of MapReduce process]

• You simply write code for the
  – Map task; and
  – Reduce task.
MapReduce and HDFS

- Detailed workflow
Cool things with MapReduce

- Chain of two MapReduce jobs;
  - Input: HDFS → Mapper1 → Reducer1 → Mapper2 → Reducer2 → Output: HDFS

- Load external data into your program;
  - Input: HDFS → Mapper1 → Reducer1 → Output: HDFS → Distributed Cache

- Modify the behavior of FileInputSplit;
Project 4 Module 1

• Write a MapReduce program that will build an inverted index of documents
  – Ex: you have 3 files - A, B and C. You need to know in which documents a particular word appear. The desired output is similar to this:
    Word1: A, B
    Word2: A, B, C
    Word3: A

• Have to use EMR Custom Jar
  – CANNOT use EMR streaming
Upcoming Deadlines

- **Project 4:**

<table>
<thead>
<tr>
<th>Project 4</th>
<th>MapReduce</th>
<th>Checkpoint</th>
<th>Available Now Due 11/16/14 11:59 PM</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hadoop MapReduce</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Custom Jar

- What is custom Jar
  - Customize your java MapReduce program

- Why custom Jar
  - More resources: HDFS/HBASE/S3
  - More job configuration flexibility
  - More control of how the resources are utilized
Recommendations

• Test for correctness with small datasets first
• DO NOT need to restart a new cluster
  – EMR will charge you one hour of usage for instances even though your EMR job failed to start
• Pay attention to your code efficiency
• Version of Hadoop
  – should match the version of your API
• Start early
TWITTER ANALYTICS: THE 619 PROJECT
Phase 2 Live Test

- WOW !!!

And the winners are...

<table>
<thead>
<tr>
<th>Query (MySQL)</th>
<th>Team</th>
<th>Throughput</th>
<th>% of requirement</th>
</tr>
</thead>
<tbody>
<tr>
<td>Q1</td>
<td>whatever</td>
<td>19057</td>
<td>136</td>
</tr>
<tr>
<td>Q2</td>
<td>CodeUncle</td>
<td>9906</td>
<td>196</td>
</tr>
<tr>
<td>Q3</td>
<td>Echo</td>
<td>14153</td>
<td>142</td>
</tr>
<tr>
<td>Q4</td>
<td>MJMCloud</td>
<td>13338</td>
<td>296</td>
</tr>
<tr>
<td>Mix-Q1</td>
<td>littleBobbyTables</td>
<td>13481</td>
<td>222</td>
</tr>
<tr>
<td>Mix-Q2</td>
<td>ZJers</td>
<td>3632</td>
<td>240</td>
</tr>
<tr>
<td>Mix-Q3</td>
<td>Penalty</td>
<td>4129</td>
<td>103</td>
</tr>
<tr>
<td>Mix-Q4</td>
<td>MJMCloud</td>
<td>3455</td>
<td>345</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Query (HBase)</th>
<th>Team</th>
<th>Throughput</th>
<th>% of requirement</th>
</tr>
</thead>
<tbody>
<tr>
<td>Q1</td>
<td>Originals</td>
<td>19124</td>
<td>137</td>
</tr>
<tr>
<td>Q2</td>
<td>NoLiFE</td>
<td>8300</td>
<td>166</td>
</tr>
<tr>
<td>Q3</td>
<td>FDU</td>
<td>12350</td>
<td>125</td>
</tr>
<tr>
<td>Q4</td>
<td>Amazombies</td>
<td>15867</td>
<td>275</td>
</tr>
<tr>
<td>Mix-Q1</td>
<td>spartans</td>
<td>10750</td>
<td>179</td>
</tr>
<tr>
<td>Mix-Q2</td>
<td>Darwin</td>
<td>2473</td>
<td>163</td>
</tr>
<tr>
<td>Mix-Q3</td>
<td>sudoCloud</td>
<td>4087</td>
<td>102</td>
</tr>
<tr>
<td>Mix-Q4</td>
<td>MJMCloud</td>
<td>4109</td>
<td>411</td>
</tr>
</tbody>
</table>

- Most competitive Live Test Ever!!!
- No clear winner?
### Unofficial rankings

#### MySQL:

<table>
<thead>
<tr>
<th>Team</th>
<th>Phase Score</th>
<th>Raw</th>
<th>Successful Queries (Max 8)</th>
</tr>
</thead>
<tbody>
<tr>
<td>CodeUncle</td>
<td>99</td>
<td>169</td>
<td>7</td>
</tr>
<tr>
<td>Darwin</td>
<td>99</td>
<td>150</td>
<td>7</td>
</tr>
<tr>
<td>FDU</td>
<td>98</td>
<td>151</td>
<td>7</td>
</tr>
<tr>
<td>sudoCloud</td>
<td>97</td>
<td>161</td>
<td>6</td>
</tr>
<tr>
<td>tw5566</td>
<td>97</td>
<td>128</td>
<td>6</td>
</tr>
<tr>
<td>EW2</td>
<td>97</td>
<td>119</td>
<td>7</td>
</tr>
<tr>
<td>Echo</td>
<td>93</td>
<td>141</td>
<td>6</td>
</tr>
<tr>
<td>SanYingZhanLvBu</td>
<td>90</td>
<td>143</td>
<td>6</td>
</tr>
</tbody>
</table>
### Unofficial rankings

**HBase:**

<table>
<thead>
<tr>
<th>Team</th>
<th>Phase Score</th>
<th>Raw</th>
<th>Successful Queries (Max 8)</th>
</tr>
</thead>
<tbody>
<tr>
<td>sudoCloud</td>
<td>98</td>
<td>151</td>
<td>6</td>
</tr>
<tr>
<td>Darwin</td>
<td>96</td>
<td>132</td>
<td>6</td>
</tr>
<tr>
<td>FDU</td>
<td>87</td>
<td>132</td>
<td>5</td>
</tr>
<tr>
<td>CodeUncle</td>
<td>82</td>
<td>125</td>
<td>4</td>
</tr>
</tbody>
</table>
Unofficial rankings

Combined:

<table>
<thead>
<tr>
<th>Team</th>
<th>MySQL Score</th>
<th>Hbase Score</th>
<th>Combined Raw</th>
</tr>
</thead>
<tbody>
<tr>
<td>sudoCloud</td>
<td>97</td>
<td>98</td>
<td>156</td>
</tr>
<tr>
<td>Darwin</td>
<td>99</td>
<td>96</td>
<td>141</td>
</tr>
<tr>
<td>FDU</td>
<td>98</td>
<td>87</td>
<td>141</td>
</tr>
<tr>
<td>CodeUncle</td>
<td>99</td>
<td>82</td>
<td>147</td>
</tr>
</tbody>
</table>

Hard work pays off.
Rewards? TBD, then TBA
Q5 & Q6

● Easiest queries of the 619 Project? Maybe…

● Question: “Why so many ETLs? It takes 12-18 hours (runtime) each phase?”

● Answer: By now ETL should not take more than a couple of hours per query. Maybe you need to rethink your ETL strategy?
Warnings

- Take care of duplicate tweets (count only once)
- Read the writeup. So many hints!!
- No more correctness files will be provided. Writeup contains all you need to know.
Samples

And a sample response, for $m=1001395310$ $n=100169406$ the output is:

<table>
<thead>
<tr>
<th></th>
<th></th>
<th>1001395310</th>
<th>100169406</th>
<th>WINNER</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>21</td>
<td>100169406</td>
<td>100169406</td>
<td></td>
</tr>
<tr>
<td>33</td>
<td>12</td>
<td>1001395310</td>
<td>1001395310</td>
<td></td>
</tr>
<tr>
<td>100</td>
<td>40</td>
<td>1001395310</td>
<td>1001395310</td>
<td></td>
</tr>
<tr>
<td>133</td>
<td>73</td>
<td>1001395310</td>
<td>1001395310</td>
<td></td>
</tr>
</tbody>
</table>

For Q6, a single sample should be enough:

$\begin{align*}
  m &= 0 \\
  n &= 99999999999 \\
  x &= 48856657
\end{align*}$
Phase 3 Report

[SUPER IMPORTANT]

- Start early
- Document your steps
- Identify and isolate the performance impact of each change you make

Use the new template!!!!: http://goo.gl/IBWVUH

MAKE A QUANTITATIVE, DATA-DRIVEN REPORT
Live Test

- No benefit of database pre-caching
  - (unless you’re really smart)

- 30 minute warm-up
- 3 hours Q1-Q6
- 30 minutes mix-Q1Q2Q3Q4Q5Q6
  - Avoid bottlenecks
Grading

● The report really matters!!
  ○ Dense, not long
  ○ What you tried and what you measured matters

● 60% of the grade of the 619 Project
  ○ 45% Live Test
  ○ 15% Report

● But, improve a lot and you can partially cover up for a poor Phase 1 or Phase 2
Stress? Worried about coursework?

I’m a little stressed right now...
(just turn around and leave quietly and no one gets hurt.)

TAs to the rescue
Stress? Worried about course load?

● Remember this slide for the last 4 weeks?

Suggestions / Improvements

● UI
● Favicon
● Design
● Report comments
● Features
● Bugs

Here’s what it contains

<table>
<thead>
<tr>
<th>Timestamp</th>
<th>Bug/Suggestion</th>
<th>Feedback:</th>
</tr>
</thead>
<tbody>
<tr>
<td>10/6/2014 13:36:28</td>
<td>Bug</td>
<td>This is a bug report test.</td>
</tr>
<tr>
<td>10/6/2014 13:36:43</td>
<td>Suggestion</td>
<td>This is a suggestion test.</td>
</tr>
<tr>
<td>10/6/2014 13:37:01</td>
<td>Bug</td>
<td>This is a bug report test.</td>
</tr>
</tbody>
</table>

- Help us help you
  - Use the feedback mechanisms provided
  - Use Office Hours
  - Attend recitation

- Help each other
  - Student answers on Piazza are negligible
  - We cannot help 300 students starting on the last day
Any questions?