CS15-319 / 15-619
Cloud Computing

Recitation 13
November 18th and 20th, 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
Last Week’s Project Reflection

• Provision your own Hadoop cluster
• Write a MapReduce program to construct inverted lists for the Project Gutenberg data
• Run your code from the master instance
• Piazza Highlights
  – Different versions of Hadoop API: Both old and new should be fine as long as your program is consistent
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
Input Text Predictor

• Suggest words based on letters already typed
An *n*-gram is a phrase with *n* contiguous words

**Example Phrase:** This is interesting because this is a cloud computing course

<table>
<thead>
<tr>
<th>#</th>
<th>1-gram</th>
<th>Count</th>
<th>2-gram</th>
<th>Count</th>
<th>3-gram</th>
<th>Count</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>this</td>
<td>2</td>
<td>this</td>
<td>2</td>
<td>this is interesting</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>is</td>
<td>2</td>
<td>is interesting</td>
<td>1</td>
<td>is interesting because</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td>interesting</td>
<td>1</td>
<td>interesting because</td>
<td>1</td>
<td>interesting because this</td>
<td>1</td>
</tr>
<tr>
<td>4</td>
<td>because</td>
<td>1</td>
<td>because this</td>
<td>1</td>
<td>because this is</td>
<td>1</td>
</tr>
<tr>
<td>5</td>
<td>a</td>
<td>1</td>
<td>is a</td>
<td>1</td>
<td>this is a</td>
<td>1</td>
</tr>
<tr>
<td>6</td>
<td>cloud</td>
<td>1</td>
<td>a cloud</td>
<td>1</td>
<td>is a cloud</td>
<td>1</td>
</tr>
<tr>
<td>7</td>
<td>computing</td>
<td>1</td>
<td>cloud computing</td>
<td>1</td>
<td>a cloud computing</td>
<td>1</td>
</tr>
<tr>
<td>8</td>
<td>course</td>
<td>1</td>
<td>computing course</td>
<td>1</td>
<td>cloud computing course</td>
<td>1</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>#</th>
<th>4-gram</th>
<th>Count</th>
<th>5-gram</th>
<th>Count</th>
<th>6-gram</th>
<th>Count</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>this is interesting because</td>
<td>1</td>
<td>this is interesting because this</td>
<td>1</td>
<td>this is interesting because this is</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>is interesting because this</td>
<td>1</td>
<td>is interesting because this is</td>
<td>1</td>
<td>is interesting because this is a</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td>interesting because this is</td>
<td>1</td>
<td>interesting because this is a</td>
<td>1</td>
<td>interesting because this is a cloud</td>
<td>1</td>
</tr>
<tr>
<td>4</td>
<td>because this is a</td>
<td>1</td>
<td>because this is a cloud</td>
<td>1</td>
<td>because this is a cloud computing</td>
<td>1</td>
</tr>
<tr>
<td>5</td>
<td>this is a cloud</td>
<td>1</td>
<td>this is a cloud computing</td>
<td>1</td>
<td>this is a cloud computing course</td>
<td>1</td>
</tr>
<tr>
<td>6</td>
<td>is a cloud computing</td>
<td>1</td>
<td>is a cloud computing course</td>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>a cloud computing</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>a cloud computing course</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
The result seems logical: the singular “is” becomes the dominant verb after the American Civil War.
“one nation under God” and “one nation indivisible.”

“under God” was signed into law by President Eisenhower in 1954.
How to Construct an Input Text Predictor?

1. **Given a language corpus**
   - Project Gutenberg (2.5 GB)
   - English Language Wikipedia Articles (30 GB)

2. **Construct an n-gram model of the corpus**
   - An n-gram is a phrase with n contiguous words
   - For example a set of 1,2,3,4,5-grams with counts:
     - this 1000
     - this is 500
     - this is a 125
     - this is a cloud 60
     - this is a cloud computing 20
How to Construct an Input Text Predictor?
(Next Week)

3. Build a statistical language model that contains the probability of a word appearing after a phrase

- $Pr(is|this) = \frac{Count(this\ is)}{Count(this)} = \frac{500}{1000} = 0.5$

- $Pr(a|this\ is) = \frac{Count(this\ is\ a)}{Count(this\ is)} = \frac{125}{500} = 0.25$

4. Store and index the words and their probabilities to use in an application
This Week’s Goal

Construct an n-gram model of the corpus

– An n-gram is a phrase with n contiguous words
– For example a set of 1,2,3,4,5-grams with counts:
  • this 1000
  • this is 500
  • this is a 125
  • this is a cloud 60
  • this is a cloud computing 20
Recommendation

• Use small text to test your code and debug before running the entire big dataset
• Optimize your code to accelerate MapReduce before seeking other optimization methods
• Start Early
• Reference:
Upcoming Deadlines

• Project 4:

<table>
<thead>
<tr>
<th>Project 4</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Input Text Predictor: NGram Generation</td>
<td></td>
</tr>
<tr>
<td>NGram Generation</td>
<td><strong>Checkpoint</strong></td>
</tr>
<tr>
<td></td>
<td>11:59PM</td>
</tr>
<tr>
<td></td>
<td>11/23/2014</td>
</tr>
</tbody>
</table>

• 15-619 Project:
  – Phase 3 (last phase) is due on November 20th
  – Live-test will begin at 12:00 am, November 21st
TWITTER ANALYTICS: THE 619 PROJECT
Important Dates

● Phase 3 dummy test, Nov. 18
● Phase 3 test submission due 23:00 ET, Nov. 20
● Phase 3 report due 23:59 ET Nov. 21
Phase 3 Report [VERY IMPORTANT]

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

MAKE A QUANTITATIVE, DATA-DRIVEN REPORT
Good reports contain Tables

<table>
<thead>
<tr>
<th>Idea</th>
<th>Expected Result</th>
<th>Submission id</th>
<th>Observed Result</th>
<th>Inference</th>
</tr>
</thead>
<tbody>
<tr>
<td>Use a Godzilla caching proxy server</td>
<td>200% improvement in throughput (based on reference[1])</td>
<td>10022, 10024</td>
<td>No improvement (8k rps)</td>
<td>Caching proxy did not help as the front end server we use already caches responses</td>
</tr>
<tr>
<td>Use ZoomZoom Web Server</td>
<td>1 million rps (based on YouTube video [2])</td>
<td>11211-11217</td>
<td>25k rps</td>
<td>LG is not fast enough</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>
Good reports contain Charts/Graphs

Illustrate the effectiveness of your optimization technique by tweaking parameters and plotting the delta
Live Test

● 4-hour live test
  ○ 30 minute warm-up
  ○ 3 hours Q1-Q6
  ○ 30 minutes mix-Q1Q2Q3Q4Q5Q6

● $1.6 per hour
  ○ EC2 instance, EBS, ELB
## Timeline (+/- delta) EST

<table>
<thead>
<tr>
<th>Start Time</th>
<th>End Time</th>
<th>What’s Happening</th>
</tr>
</thead>
<tbody>
<tr>
<td>0000</td>
<td>0030</td>
<td>Warmup</td>
</tr>
<tr>
<td>0030</td>
<td>0100</td>
<td>Q1</td>
</tr>
<tr>
<td>0100</td>
<td>0130</td>
<td>Q2</td>
</tr>
<tr>
<td>0130</td>
<td>0200</td>
<td>Q3</td>
</tr>
<tr>
<td>0200</td>
<td>0230</td>
<td>Q4</td>
</tr>
<tr>
<td>0230</td>
<td>0300</td>
<td>Q5</td>
</tr>
<tr>
<td>0300</td>
<td>0330</td>
<td>Q6</td>
</tr>
<tr>
<td>0330</td>
<td>0400</td>
<td>Mix</td>
</tr>
</tbody>
</table>
Leaderboard (as of 3 AM on Nov 18)

<table>
<thead>
<tr>
<th>Q1</th>
<th>Q2</th>
<th>Q3</th>
<th>Q4</th>
<th>Q5</th>
<th>Q6</th>
</tr>
</thead>
<tbody>
<tr>
<td>apt143</td>
<td>Transcendence</td>
<td>FDU</td>
<td>MJMCcloud</td>
<td>FDU</td>
<td>SVM</td>
</tr>
<tr>
<td>cloudreaper</td>
<td>CumulusNimbus</td>
<td>Cyan</td>
<td>wanbaoC2</td>
<td>cloudlol</td>
<td>cloudlol</td>
</tr>
<tr>
<td>FDU</td>
<td>FDU</td>
<td>161Santiago</td>
<td>Cyan</td>
<td>CMUETC</td>
<td>FDU</td>
</tr>
</tbody>
</table>

Exciting stuff !!!
How to do well in a Live Test?

● Don’t crash. If you do, recover fast!!!

● Make smart trade-offs (focus on your score)

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

● Hopefully you have simulated a Live Test
  ○ self-warmup
  ○ sequential 20 minutes
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
Suggestions / Improvements

- UI
- Favicon
- Design
- Report comments
- Features
- Bugs
- Feedback

Call for Teaching Assistants

• If you are interested in being evaluated for a TA position for S15 or F15
  - We will be releasing a TA application form soon
  - We will also be releasing the TA interview dates soon
    • Most likely around 11/24 & 11/25
• Being a TA is an excellent learning opportunity
  - Ask one of the TAs now
• You can work on a variety of teams
  - Project development
  - Testing system/scoreboard improvements
  - Cheat checking
  - Budget and tag checking
  - Grading and office hours
  - Etc…
Any questions?