Last Week’s Project Module

• 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
Input Text Predictor

• Suggest words based on letters already typed
**n-gram**

- An *n*-gram is a phrase with *n* contiguous words

<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</td>
<td>1</td>
<td>this is interesting because</td>
<td>1</td>
<td>this is interesting because this is</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>is interesting</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</td>
<td>1</td>
<td>interesting because this a</td>
<td>1</td>
<td>interesting because this is a</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</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</td>
<td>1</td>
<td>is a cloud computing course</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>a cloud computing course</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td></td>
<td></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

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
Upcoming Deadlines

• Project 4:
  
  | Project 4 |
  |---|---|
  | Input Text Predictor: NGram Generation | Checkpoint |
  | Ngram Generation | 11:59PM 04/20/2014 |

• 15-619 Project:
  – Phase 3 (last phase) is due on April 22\textsuperscript{nd}
  – Live-test will be announced
15-619 Project

- Live test for phase 2 is completed
- You should have received feedback
- Phase 3 is ongoing!
  - 75% of the total grade
  - Pick one between MySQL and HBase
  - 6 queries in total
  - 4 hour live test at the end to determine your performance and the winning team!
15-619 Project: Phase 3

• Q4: Text of tweets
  – A tweet may contain multiple lines

• Q5: Find tweets by location and during a particular time range
  – The text of tweet contains a given place
  – All possible places come from “place” object in the data set
  – Text of tweet needs pre-processing (see write up)

• Q6: Number of tweets
  – The number of tweets in a given data set
15-619 Project:
Rumors and the Truth

- EMR cost is for cluster: **No!**
  - EMR cost is per instance per hour. A cluster of 9 m1.large will consume $0.044 \times 9 = $0.396

- Budget is only development cost: **No!**
  - $75 is for the whole phase including live test. Please intelligently plan how to spend

- We can start until this weekend: **No!**
  - The amount of data you will process will be larger than last phase, leading to increased risks for ETL
  - You may need more time to optimize your design: new queries tend to be more difficult to achieve a good score
  - You should be doing ETL now
15-619 Project: How We Test

- We use JMeter
- Multiple threads (up to 50) keep issuing requests to your IP address
- Your responses are compared to the correct responses
- Requests are not ordered: they are generated randomly based on some rules to fully explore your throughput
- For q5 and q6, expect large ranges (such as 100K userids in q6)
- For q4, expect many responses that are large in size
15-619 Project: What You Should Know

- Set port configuration of ELB as TCP 80 -> TCP 80 instead of HTTP 80 -> HTTP 80
- Do not use the same connection for every request: significant negative effects
- Table design is not the whole world: find the bottleneck in your system
- Re-test q1 – q3 in phase 3: Something may be different
Check AWS Services Charges
Check AWS Services Charges
Demo Outline

1. Hadoop Commands
   - Hadoop fs -help
   - hadoop fs -put
   - hadoop fs -get
   - hadoop distcp

2. N-Gram Generation
   - Google Instant
   - Input Text Predictor
   - N-Gram Generation
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:
  3. Amazon AWS EMR Best Practices (link posted on Piazza)