Announcements

• Checkpoint Quiz Unit 5, due on:
  – Friday May 3\(^{rd}\) at midnight

• Project 4, Part c, due on:
  – Friday May 3\(^{rd}\) at midnight

• Last Recitation (#15):
  – Tuesday, April 30th
Announcements

• Open up S3 location of hand ins:
  – Give access to your S3 bucket to:
    • public
    • onlinecloudcomputingcourse@gmail.com
  – You could lose credit or be penalized otherwise
  – See Piazza Post on how to open up your handin directory

• Encounter a general bug:
  – Post on Piazza

• Encounter a grading bug:
  – Post Privately on Piazza

• Post feedback on OLI
Project 4 Student Progress

• Part b: Input Text Predictor: N-gram Generation
  – 97% Students Completed

• Stats:
  – Total n-grams generated from the Gutenberg Dataset:
    • Approximately 477 million
  – Fastest Computation
    • 16 minutes 48 seconds
    • 19 c1.xlarge @ $0.07 spot price
    • Cluster cost: $1.3 per hour
  – Slowest Computation
    • 4 m1.small (with 1000 reducers!)
    • 8 hours and 25 minutes
More MapReduce Tips

• Watch out for Whirr bugs
  – Failure to launch instances
    • Check AWS Management Console to verify
  – Failure to install and configure Hadoop correctly
    • Run `sudo jps` on Master node to verify that the Hadoop processes are running correctly. Test using example jobs or small data first.
  – Using different instance types for master and slave nodes may provision them in different zones
  – 32 bit AMIs will not work for larger instance types (m1.large – etc. need 64 bit)
New Modules

- Unit 5 – Distributed Programming and Analytics Engines for the Cloud
  - Introduction to Distributed Programming for the Cloud
  - Distributed Analytics Engines for the Cloud: MapReduce
  - Distributed Analytics Engines for the Cloud: Pregel
    - Pregel
    - The Computation and Architectural Models
    - The Data Structure and Storage
    - The Graph Flow and API
    - Architectural Model and Workflow
    - Fault Tolerance
Project 4, Part c

• Project 4, Part a
  – MapReduce
  – Project 4 Survey

• Project 4, Part b
  – Input Text Predictor: NGram Generation

• Project 4, Part c
  – Input Text Predictor: Language Model and User Interface
Recap Input Text Prediction

• Construct an Input Text Predictor

Google Suggest

WordLogic iKnowU keyboard
How to Construct an Input Text Predictor?

1. Given a language corpus
   - Project Gutenberg (2.5GB, already on S3)
   - English Language Wikipedia Articles (30GB, on S3 soon)

2. Construct an n-gram model of the corpus
   - An n-gram is a phrase with n 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 blue 60
     • this is a blue house 20
How to Construct an Input Text Predictor?

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

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

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

4. Store and index the words and their probabilities to use in an application
Discussions

• Your questions...
Upcoming Deadlines

• Unit 5:
  Unit 5: Distributed Programming and Analytics Engines for the Cloud
  Module 20: Distributed Analytics Engines for the Cloud: Pregel

• Project 4
  Project 4
  Module 34: Input Text Predictor: Language Model and User Interface
  Language Model Generation  Checkpoint  Available Now
  Due 5/3/13 11:59 PM