Recitation 1
CS435: Introduction to Big Data

GTAs: Bibek R. Shrestha and Aaron Pereira
Email: cs435@cs.colostate.edu

August 31, 2018
Today...

- Discussion on how WordCount program works
- Introduction to Programming Assignment 1
Understanding MapReduce

Figure 1: Execution overview

Reference: MapReduce: Simplified Data Processing on Large Clusters (paper)
Data Flow in Hadoop

Understanding WordCount Example

Reference: https://wikis.nyu.edu/display/NYUHPC/Big+Data+Tutorial+1%3A+MapReduce
Understanding WordCount.java

http://www.cs.colostate.edu/~cs435/CS435Fall18/PA0/PA0-Code.pdf
Programming Assignment 1

- Introduction to Programming Assignment 1
Dataset

- ~1.5 million Wikipedia articles (summarized)
- ~1 GB size
- Format:

  Title_of_Article-1<=====>DocumentID<=====>Text_of_Article-1
  NEWLINE
  NEWLINE
  Title_of_Article-2<=====>DocumentID<=====>Text_of_Article-2
  NEWLINE
  NEWLINE
N-gram

- A contiguous sequence of N items from a given sequence of text or speech
- Example:
  “We, analyze, large, datasets”
- 1-grams (aka unigram)
- We, analyze, large, datasets
- 2-grams (aka bigram)
- (__, We), (We, analyze), (analyze, large), (large, datasets), (datasets, __)
Profile 1

- First 500 unigrams in the dataset
- Must be sorted in alphabetical order (ascending)
- Eliminate duplicates.
- Use of combiner to eliminate local duplicates can be important (one of the approaches that can be taken).
Profile 2

- A list of top 500 unigrams and their frequencies within each article
- Similar to word-count program
- Each article has unique integral Document ID.
- Output should be grouped by Document ID.
- Output files will correspond to the number of reducer used.
- Requires two jobs to be executed in a sequence
- Also, you may use the concept of CompositeKey in MapReduce.
Profile 3

- A list of top 500 unigrams and their frequencies in the corpus
- List should be sorted from most frequent unigrams to least frequent ones.
- Solution to generate profile 3 is a combination of profile 1 and 2.
- Here, we list unigram with its total occurrence in the complete dataset (of course, without considering per article construct as in profile 2).
Data Preprocessing

- Consider only alphabetic and numeric text.
- Convert upper cases to lower cases.
- Example (Original → processed → toLowerCase)
  
  U.S.A. → USA → usa
  west. → west → west
  (USA) → USA → usa
  D.C., → DC → dc
  9.8 → 98 → 98
Notes

- Before starting, make a design your system.
- Make sure you understand Word-Count program.
- Start writing program by making modification in given Word-Count program.
- Test your program in standalone/local mode with given sample file.
- Verify your output in even smaller text file, if required, before moving on to larger files.
- Parsing (preprocessing) the dataset will be an important part and will require considerable effort. (This is generally true for any big data application.) Spend some time to observe the dataset provided.
In next recitation...

- We will go through code snippets for each profile.
- We will go through common problems asked in piazza and in-person.