CSE 487/587 Assignment 2: Big Data Processing with Hadoop

The objective of this assignment is to get started with big data processing with Hadoop. The goals of the assignment are to implement basic text processing tasks from scratch on the Hadoop framework

PART1 - Setup and WordCount - 5 Points

- Get familiar with the VM and the Hadoop framework
- In the folder gutenberg, located in your home directory, there are 3 documents. Use necessary commands to transfer data to Hadoop distributed file system
- Implement a MapReduce algorithm to produce count of every word in the document

PART 2 - N-grams- 10 Points

- Using the same gutenberg dataset, implement a MapReduce algorithm that will produce modified tri-grams around the key words, after replacing the key word with '$'.

  Example:
  cat was sitting on a roof ---> if the key word was 'sitting' ---> the modified tri-grams would be
  cat_was_$, was_$_on,$_on_a,

- The key words to look for in the gutenberg dataset are 'science', 'sea' , 'fire'.

- The algorithm after producing these modified tri-grams, should return the 10 most occurred modified tri-gram in the dataset.

PART 3 - Inverted Index - 5 Points

- Using the gutenberg dataset, implement a MapReduce algorithm to produce inverted index for the whole dataset.
- A small explanation of what inverted index is can be found in the link Inverted index

PART 4 - Relational Join - 5 Points

- Using the Dataset provided along with the assignment, Implement a MapReduce algorithm to join two datasets using a primary key
- The assumed primary key is the 'Employee ID'

BONUS: K-Nearest Neighbour - 5 Points

- Using the train and test set provided along with the assignment, Implement KNN algorithm using MapReduce.
- You can assume the test set is small.
- The algorithm should return the corresponding predicted label for each test instance

Submission Instructions

You will submit Assignment2.zip or Assignment2.tar.gz, a compressed archive file containing the following files:

- MapReduce code for each part of the assignment.
- The code should exclusively follow the MapReduce programming model
- The code can be in Java or Python
- A video (or link to a video) of you running each part of the assignment.
- A report explaining the logic that you have used to implement each part of the assignment (make sure that you write names of your team members on the report)

Submission is due 04/18/2020, Saturday, 11:59 PM EST. Please use the submit_cse487 or submit_cse587 script in Timberlake to submit your assignment.

PS: Part of this assignment was inspired by the distributed systems course at cmu fall 2010.