CSE427 – Homework 2

M. Neumann

Due THU 02/04/2016 10am

Getting Started

Update your SVN repository. Find instructions on how to checkout, update, and commit to your SVN repository here: `http://sites.wustl.edu/neumann/resources/cse427s_resources/`

> When needed, you will find additional materials for *homework x* in the folder hwx. So, for the current assignment the folder is hw2.

Hint: You can **check your submission** to the SVN repository by viewing `https://svn.seas.wustl.edu/repositories/<yourwustlkey>/cse427s_sp16` in a web browser.

Indicating Group Work

Use the file `partners.txt` to indicate group work. Follow these instructions exactly, to ensure to get credit!

- `partners.txt` needs to include up to 2 wustlkeys in the first two lines (one line per wustlkey)
- **first line/wustlkey is the repository, where the solution is located**. We will **only** consider the submission in this repository!
- Every student in a group needs to have **the same** `partners.txt` in the hwx folder in their repository (indicating that the partnership are **mutually accepted**)!
- If you do not have a partner, try to find one. If you want to submit on your own, indicate your wusltkey in the first line of `partners.txt` and leave the second line blank.

Preparation (not for credit)

1. Download and set up the Cloudera VM. Find the pre-configured VM for this class here: `https://classes.cec.wustl.edu/cse427/`. Your username is `training` and the password (if you should need it) is `training`.
2. Checkout your SVN repository in the VM, then you can submit solutions directly from there.
3. if you haven't done so already, run the course setup script in a terminal window in your VM:

```
$ ~/scripts/developer/training_setup_dev.sh
```

Problem 1: HDFS (25%)

If you haven't done so, complete Lab 1 (Part I) as it prepares the data used in subsequent homework problems. Note, that you will loose credits if you are not using the correct input data in upcoming problems.

(a) If you haven't done so, remove the file named `glossary`. Then, list all the files in the `shakespeare` folder **in HDFS :**. Provide the command and result in the `hw2.txt` file.

(b) **In HDFS :** Display the first 16 lines (not more) in `poems`. Provide the command and result in the `hw2.txt` file.

(c) Assume you have a data file of size 640MB, the replication rate in the distributed file system is 2, the block size is 128MB, and the cluster consists of 6 nodes (indicate them by N1, N2, . . . , N6) on 3 racks (indicate them by RA, RB, RC). Now, consider the storage of this file in HDFS. You can assume that the cluster is empty. Provide the meta-data stored on master node for this file.

Submit your answers by editing the `hw2.txt` file in the hw2 folder in your SVN repository.

Problem 2: Running a MapReduce Job (45%)

If you haven't done so, complete Lab 1 (Part II). (Run a MapReduce job to count the number of occurrences of every word in the works of Shakespeare.) Now, we will look at the result which is stored in the file `part-r-00000` in the output directory you specified when submitting the job.

(a) How often do the following words occur:

- ADRIANO
- Whether
- love
- loves
- the
- whether
- we
- zodiac

(b) How many different words (you can consider every reduce() output key a word) occur in the `shakespeare` data? Provide the cpunt AND a **one line** unix command to retireve this number directly from the results file in HDFS . (Hint: on the course webpage under Resources and HowTos you will find a cheat sheet for useful terminal commands in Linux.)

(c) By looking at the results of this word-count implementation, give 2 suggestions on how to improve the algorithm if our goal is to use the results to analyze the sentiments in Shakespeare's work. You can read about sentiment analysis here: `https://en.wikipedia.org/wiki/Sentiment_analysis`.

Improvement can be in terms of both, efficiency (speed, memory), or quality of the result.

(d) Consider the MapReduce execution of word-count. The RecordReader is a system provided function used in each MapReduce program. What does it do? Describe RecordReader input and output.

Submit your answers by editing the `hw2.txt` file in the hw2 folder in your SVN repository.

Problem 3: Skew (30%)

Suppose we execute the word-count MapReduce program on a large repository of text data such as a copy of the English Wikipedia. Our cluster consists of 200 compute nodes and we shall use 100 Map tasks and some number of Reduce tasks.

(a) Do you expect there to be significant **skew** in the times taken by the various reducers (**reduce() functions**) to process their value list? Why or why not?
Note: we do not use a *combiner* at the Map tasks (if you do not know what a combiner is then ignore this note, we will learn about combiners later in this course).

(b) If we assign the reducers (reduce() functions) to a small number of Reduce tasks, say 10 tasks, at random, do you expect the **skew in the Reduce Tasks** to be significant? What happens if we instead combine the reducers into $10,000$ Reduce tasks?

Submit your answers by editing the `hw2.txt` file in the hw2 folder in your SVN repository.
Commit your solution, e.g. run:

```
$ svn commit -m 'hw2 submission' .
```

Bonus Problem (5% up to a max. of 100%) - no group work!

We will be doing a little experiment in the course. We will collect your **emotional** description for each homework assignment and hopefully, we will be able to use this data for sentiment analysis at the end of the semester!

Sentiment Analysis tries to assess the *emotion* or *mood* in a text document. Essentially it can be computed by comparing the set of words in each document to an existing dictionary of positive words, negative words, and neutral words. In our experiment, we give you the chance to voice your opinions on each homework by writing a couple of sentences as a review for the homework. You will not be graded on what your review says, but rather solely the completion of it. At the end of the year, given that we have enough data for each homework, you will perform sentiment analysis on this data to see which homework you and your peers regarded as "positive" or "negative".

So, please write a review for this homework and store it in the file `hw2_review.txt` provided in your SVN repository. This file should only include the review, no other information such as name, wustlkey, etc. Remember that you are not graded for the content of your review, solely it's completion.

You can only earn bonus points if you write **at least 50 words**. Bonus points are given to the **owner of the repository only** (no group work!).