CSE427S

HOMEWORK 2

M. Neumann

Due: THU 13 SEPT 2018 4PM

Getting Started

Update your SVN repository.

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_fl18` in a web browser.

SUBMISSION INSTRUCTIONS

WRITTEN:

- all written work needs to be submitted electronically in *pdf format*[1] via GRADESCOPE
- provide the following information on *every* page of your `pdf` file:
  - name
  - student ID
  - wustlkey (your `wustlkey` indicates the location (SVN repository) for your code submissions; without this information we will not be able to grade your code!)
- start every problem on a *new page*
- **FOR GROUPS**: make a **group submission** on GRADESCOPE and provide the following information for **all group members** on every page of your `pdf` file: names, student IDs, and location of code submissions (**one** student's `wustlkey`)[2]

CODE:

- code needs to be submitted via SVN repository commit (detailed submission instructions are provided whenever code submissions are required)
- make sure to always use the required *file name(s)* and *submission format(s)*
- **comment your code** to receive maximum credit

[1] Please, **type your solutions** or use **clear hand-writing**. If we cannot read your answer, we cannot give you credit nor will we be able to meet any regrade requests concerning your writing.

[2] It is sufficient to commit code to one SVN repository. If you do not specify the repository for your group's code submission clearly, we will only check the first repository, sorting wustlkeys alphabetically.

Preparation (this is essentially lab0)

1. Download and set up the course VM. Find instructions on the course webpage under Resources and HowTos.
2. Checkout your SVN repository **in the VM**, then you can commit code submissions directly from there. You will need to install subversion in your VM. To do so run:

```
$ sudo yum install subversion
```

3. If you haven't done so already, run the course setup script in a terminal window in your VM:

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

Problem 1: HDFS (50%)

If you haven't done so, complete Lab 1 as it prepares the data used in this and subsequent homework problems. Note, that you will loose credit 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 this command and the result in the `hw2.pdf` file.

Display the first 16 lines (not more) of the `poems` data stored **in HDFS**. Provide the command and result in the `hw2.pdf` file.

(b) The default **replication factor** in HDFS is 3. The optimal number of redundant block copies depends on the following factors: *cost of replication*, *cost of data loss*, and the *probability of failure*.

- What is the downside of a large replication factor with respect to the storage capacity of the data nodes in the cluster?
- What is the downside of a large replication factor with respect to the amount of memory of the cluster's NameNode[3] in your answer.

(c) The default **block size** in HDFS is 128MB, which is fairly large. Discuss one benefit and one disadvantage of large block sizes. (HINT: read HTDG pp. 43-45.)

(d) Assume you have a data file of size 642MB, the replication factor in the distributed file system is set to 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). Nodes are evenly distributed among racks and you can assume that the cluster is empty. Now, consider the cluster after storing this file in HDFS. Sketch the cluster state (file/block locations) and provide the meta-data stored on the master node for this example file.

(e) Explain two disadvantages of a distributed files system such as HDFS (HINT: read HTDG pp. 43-45).

(f) Assume you want to download data from HDFS to your local client. Provide the command for this process and briefly describe how the data is located and transferred.

[3] NOTE: the NameNode keeps the filesystem metadata **in memory** for fast access.

Problem 2: MapReduce I (25%)

Suppose the input data to a MapReduce operation consists of integer values (the input keys are not important). The map function takes an integer i and produces the list of pairs (p, i) such that p is a prime divisor of i. For example, $map(12) = [(2, 12), (3, 12)]$. The reduce function is addition. That is, $reduce(p, [i_1, i_2, ..., i_k])$ is $(p, i_1 + i_2 + ... + i_k)$.

Provide the intermediate data and final result of a MAPREDUCE execution for the following integer input $i = \{15, 21, 24, 30, 49\}$. Include all **Mapper inputs**, **Mapper outputs**, **Reducer inputs**, and **Reducer outputs** in your answer.

Problem 3: MapReduce II (25%)

Given the following **input data**:

```
2013-03-15 12:39 - 74.125.226.230 /common/logo.gif 1200ms - 2326
2013-03-15 12:39 - 157.166.255.18 /catalog/cat1.html 900ms - 1211
2013-03-15 12:40 - 65.50.196.141 /common/logo.gif 1900ms - 1198
2013-03-15 12:41 - 64.69.4.150 /common/promoex.jpg 4000ms - 2326
2013-03-15 12:44 - 157.166.255.18 /catalog/cat2.html 1100ms - 1451
```

Consider a MAPREDUCE program that analyzes the log data provided above to retrieve the *average processing time for each file type*.

(a) Compute the Mapper output, Reducer input, and Reducer output for this particular example input data.

(b) Which data type do you use to represent the keys? Which data type do you use to represent the values?

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.

To be able to evaluate your sentiment analysis approach we need the ground truth. So, in addition to your textual review, take 1 minute to provide a star rating for `hw2` via this link: `https://wustl.az1.qualtrics.com/jfe/form/SV_aaAc0ILmBleyKzj`.

You can only earn bonus points if you write a meaningful review of **at least 50 words** and provide the corresponding star rating. Bonus points are given to the **owner of the repository only** (no group work!).

Submit your review in the file **`hw2_review.txt`** provided in the `hw2` folder in your SVN repository. **To commit the file run:**

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