CSE427S

HOMEWORK 3

M. Neumann

Due: THU 19 SEPT 2019 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 `hw3`.

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
- start every problem on a *new page*
- **FOR GROUPS**: make a **group submission** on GRADESCOPE and provide names and student IDs for **all group members** on every page of your `pdf` file.

CODE:

- No code submission required for this homework.

Problem 1: Running a MAPREDUCE Job (40%)

If you haven't done so, complete Lab 2 (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:

[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.

- 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 count AND a **one line** unix command to retireve this number directly from the results file in HDFS.
HINT: Lab0 and the cheat sheet linked on the course webpage under Resources and HowTos provide a documentation of useful Unix terminal commands.

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

Improvement can be in terms of **efficiency** (w.r.t. runtime or 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.

(e) In your output folder in HDFS you find one file with results (`part-r-00000`). In class we learned that this folder could contain several such results file. What do those files correspond to and which process is responsible for writing those files?

Problem 2: Skew (40%)

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 runtime taken by the various **reduce() functions** to process their value list? Why or why not?
Note: we do not use a *combiner* at the Map tasks.
HINT: read [MMDS] Ch2.2.5-6 pp 28-30 to learn more about *skew* in the MAPREDUCE job execution.

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

(c) Suppose we do use a combiner at the 100 Map tasks. Do you expect the skew in the runtime taken by the various **reduce() functions** to be significant? Why or why not?

Note: we use the following **terminology** (mind the capitalization):

- Map/Reduce Task or Mapper/Reducer
- map()/reduce() function
- Group & Sort or Shuffle & Sort

Notation used in the books:

GENERAL MAPREDUCE (MMDS)	HADOOP MAPREDUCE (HTDG)
Map task	Mapper
mapper (*avoid this!*)	map() function
Group & Sort	Shuffle & Sort
Reduce task	Reducer
reducer (*avoid this!*)	reduce() function

Problem 3: Job Execution (20%)

(a) Describe the HADOOP MAPREDUCE job execution using the YARN scheduler.

- Include the names of all daemons/processes that run on master and worker nodes.
- Describe storage locations of input, output, and intermediate data.

(b) When executing MAPREDUCE jobs, HADOOP performs *data locality optimization*. Describe what is meant by this term and indicate in which phases it is applicable. Also include a discussion on what happens if tasks can not be executed local to the data.

Reflection (Bonus Problem for 5% up to a max. of 100%)

Reflect on your homework experience! Write a paragraph of at least 50 words to express your experiences and feelings when working on this assignment. Answer at least 2 of the following questions:

- What did you like/dislike about the assignment and why?
- What is the most important thing you learned and why do you think so?
- What surprised you, and why?
- Assuming you could start over again (with working on the assignment), what would you do differently and why?

Do not include/copy and past the questions into your reflection!

Submission Instructions

Store your reflection in the `hw3_reflection.txt` file provided in the `hw3`folder in your SVN repository and commit it.

This file should only include the reflection, **no other personal information** such as name, wustlkey, etc. reflections are not graded based on the content, but solely for completion.

To submit your reflection `cd` into the `hw3` folder and run:

```
$ svn commit -m 'hw3 reflection submission' .
```

Take 1 minute to provide an overall **star rating** for this homework.
Submit it via this link: `https://wustl.az1.qualtrics.com/jfe/form/SV_bIRKP1xobpO8yIB`.

Grading - no group work!

You can only earn bonus points if you write a *meaningful* **reflection** of **at least 50 words** answering at least 2 of the prompted questions and provide the corresponding **star rating**. You will **not** be graded on what your reflection says and the number of stars you assign, but rather solely the completion of it.
Bonus points are given to the **owner of the repository only**. No group work!.