

HOMEWORK 3

M. Neumann

Due: THU 20 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 hw3.

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.

²It 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.

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:
- 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 retrieve 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 (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 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 (<i>avoid this!</i>)	<code>map()</code> function
Group & Sort	Shuffle & Sort
Reduce task	Reducer
reducer (<i>avoid this!</i>)	<code>reduce()</code> function

Problem 3: Job Execution (30%)

- (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.

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

Write a review for this homework and store it in the file `hw3_review.txt` provided in your SVN repository (and commit your changes). 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.

Provide the star rating for hw3 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 **hw3_review.txt** provided in the hw3 folder in your SVN repository. To commit the file run:

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

Take 1 minute to provide a star rating for this homework – your star rating should match