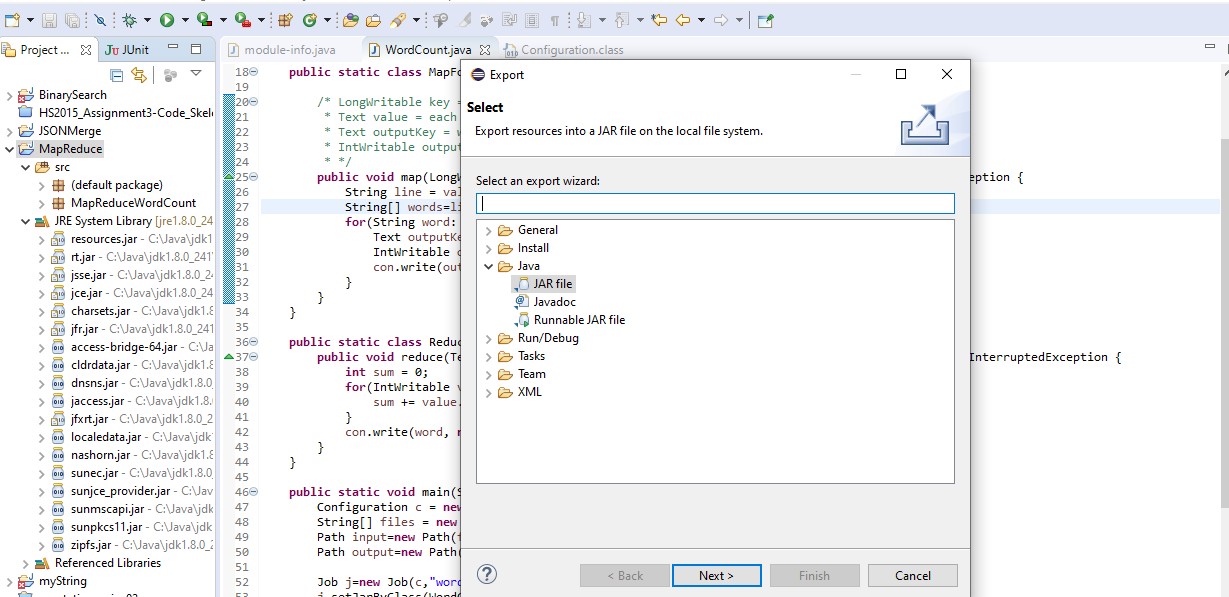
WordCount with MapReduce and Hadoop

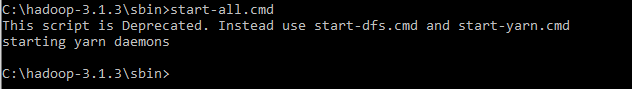
Khai Nguyen [khainguyen@temple.edu](mailto:khainguyen@temple.edu)

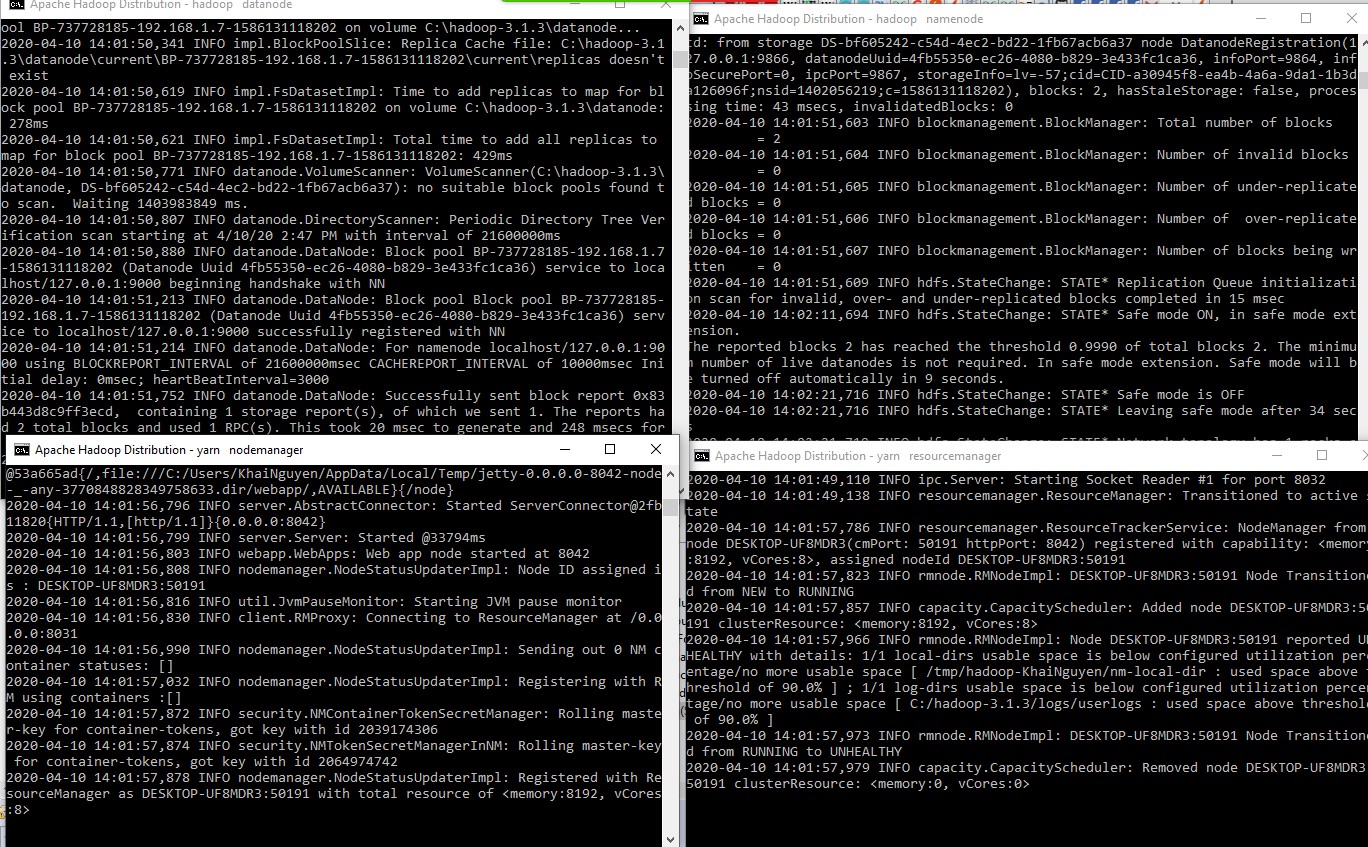
CIS 4517 Data Intensive and Cloud Computing

# Problem 1 & 2: Word Count

To produce Project > Export > JAR File > *Next*

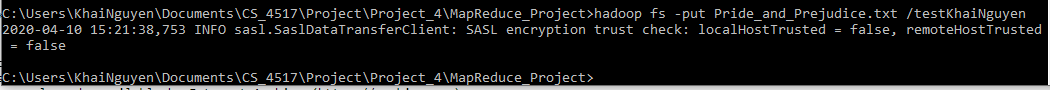






* Upload file to HDFS, store in /testKhaiNguyen

C:\Users\KhaiNguyen\Documents\CS\_4517\Project\Project\_4\MapReduce\_Project>hadoop fs -put Pride\_and\_Prejudice.txt /testKhaiNguyen



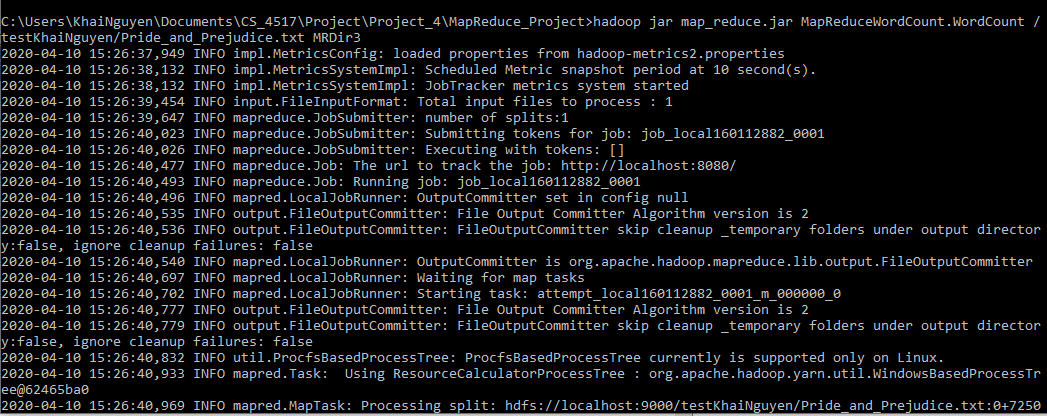
* Check if file is upload by viewing content C:\Users\KhaiNguyen\Documents\CS\_4517\Project\Project\_4\MapReduce\_Project>hadoop fs -cat

/testKhaiNguyen/Pride\_and\_Prejudice.txt

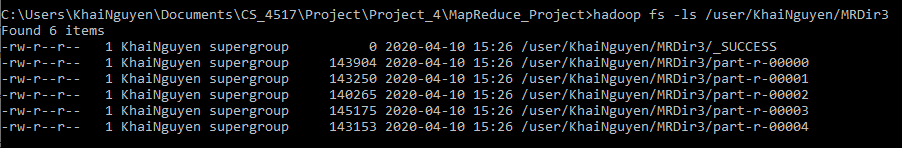


* Run MapReduce

C:\Users\KhaiNguyen\Documents\CS\_4517\Project\Project\_4\MapReduce\_Project>hadoop jar map\_reduce.jar MapReduceWordCount.WordCount /testKhaiNguyen/Pride\_and\_Prejudice.txt MRDir3



* View files created.



* Download files

hadoop fs -get /user/KhaiNguyen/MRDir3/part-r-\* .



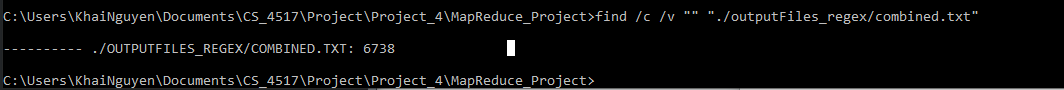
* Sort

hadoop fs -cat /user/KhaiNguyen/MRDir3/part-r-\* | sort > ./outputFiles/combined.txt

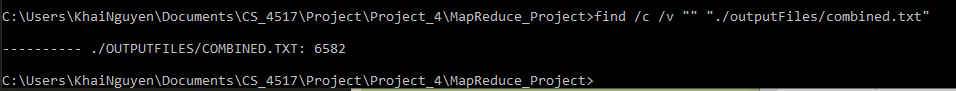


* View number of words OR number of lines since each line is designated for 1 word find /c /v "" "./outputFiles/combined.txt"

Problem 1:

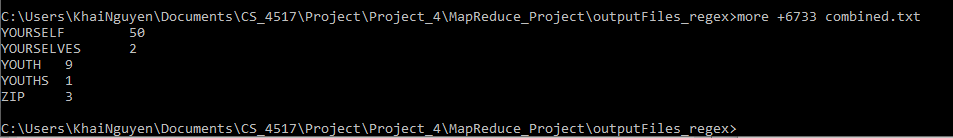


Problem 2:



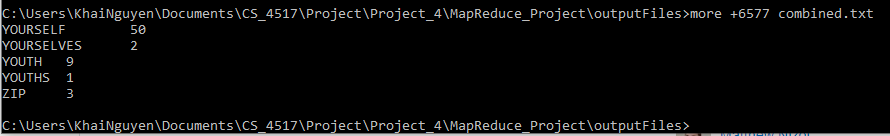
* Display last 5 lines. Problem 1:

Since there are 6738 lines, we can display the rest of the files starting at line 6733 to get the last 5 lines.

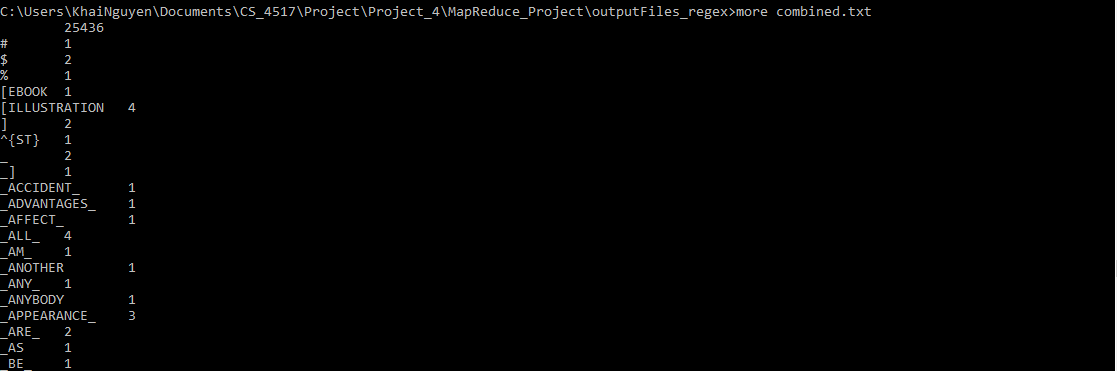


Problem 2:

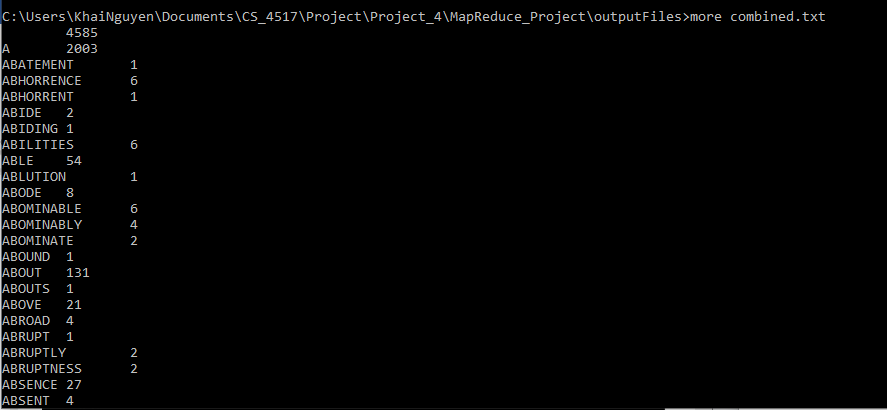
Since there are 6582 lines, we can display the rest of the files starting at line 6577 to get the last 5 lines.



* Display first line more combined.txt Problem 1:

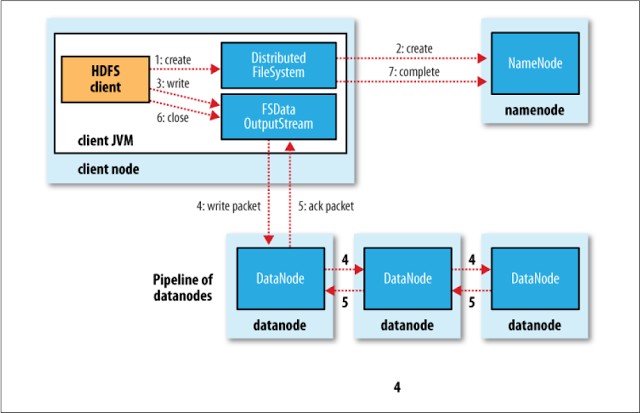


Problem 2:



# Homework 3

**(15 points) Problem 3:** The file sample.dat has **two** blocks A and B, explain in HDFS how this file is written to a Hadoop cluster with **one** namenode and **three** datanodes in a default configuration.



A

A

A

B

B

B

sample.dat

B

A

*Image from Lecture slides*

The client node will notify the namenode about the storing of file sample.data, saying how many blocks the file has. Given that each block needs to be replicated by 3 (default), the client node will write block A, and B on to the 3 datanode sequentially.

When *write* hits the final datanode, an *ack* message will be returned to the previously written nodes recursively, then to the clientnode. At the final step, the client node will notify the namenode to update the metadata of the nodes and the blocks contained.

**(10 points) Problem 5:** In Quorum Consensus algorithm, Reads go to a read quorum of size R and writes go to a write quorum of size W. For a group of **5 replicas**, explain and compare the following three possibilities:

|  |  |
| --- | --- |
| 1. R=5 and W=1    * Improves *writes* at the expense of *reads,* since *write*s can be performed at any one replica.    * Bad choice, since *writes* can be performed at 1 replica that later fails, leading to data loss. | 1. R=1 and W=5    * Improves *reads* at the expense of *writes,*   we can read from any replica.   * + Bad for *writes*. If one in the 5 replicas is down, *writes* have to wait until that replica recovers to read W=5 |
| 1. R=3 and W=3    * A good compromise, increasing the cost of reads and providing a reasonable availability of *writes* (W>1 at least) |  |