Assignment 4

1

We used MapReduce function of Hadoop in Java for this question. We first passed the timestamp and the number of request time in the Mapper and then found the maximum of the request time in the Reducer. Since, we just had to consider "en" pages and ignore the title "Main_Page" and those with title starting with "Special:", we applied these conditions at the Mapper itself to eliminate the unnecessary data. We used the following commands to compile, run and view the output of the program-

```
> ${JAVA_HOME}/bin/javac -classpath '/usr/local/hadoop/bin/hadoop classpath'
WikipediaPopular.java
> ${JAVA_HOME}/bin/jar cf wordcount.jar WikipediaPopular*.class
> /usr/local/hadoop/bin/yarn jar wordcount.jar WikipediaPopular ~/Desktop/wc/dataset
~/Desktop/wc/output
> /usr/local/hadoop/bin/hadoop dfs -cat ~/Desktop/wc/output/*
```

We changed the value of mapreduce.map.cpu.vcores and mapreduce.reduce.cpu.vcores in the mapred-site.xml file to change the number of cores. The execution time with single (low) core was 1 min 56.899 secs and with multi (high) core was 1 min 45.239 secs. Hence the difference is 11.66 secs.

2

For Spark, we used Python. Our process involved first splitting all the lines into a list, filtering out the entries we don't want, mapping everything into a tuple, reducing by key so we had the highest view per time period, then finally sorting the entries by time period and formatting them for output.

For execution time, we found an ~1.2x speedup from using a more multi-core machine. Unfortunately we didn't have a good set of computers to test on, so we wouldn't expect this to be truly accurate to what increasing the core count will do to the execution time.

3

3.1 What are the advantages of Spark over MapReduce?

Performance- Spark is far ahead of MapReduce in performance. MapReduce performs processing in a stepbystep mode and Spark operates on the entire data set as a single entity. With batch processing, Spark outperforms MapReduce by a factor of ten, and when analyzing in memory- by a hundred.

Convinience- Spark is more convinient to use when we need to analyze streaming data from sensors in production rooms, or applications that require multiple operations. Typical tasks solved with the help

of Spark Apache include realtime marketing campaigns, prompt issuance of product recommendations, cybersecurity analysis and monitoring of machine logs

3.2 How does Spark DataFrame speed up computation over Spark RDDs?

DataFrame utilizes the power of SQL and relational databases to organize data in a different form than with RDD's. This can provide a computation speed increase due to the large amount of research and development that's been put into speeding up SQL database systems.

3.2.1 Do Spark over MapReduce seem to take good advantage of more cores?

This is hard to tell from the data we've collected, due to the limited set of systems to test on. The expectation would be that Spark would be much faster due to the specialized nature of its system, as compared to MapReduce.