Problem 1: Running a MAPREDUCE Job (40%)

(a) How often do the following words occur:

- ADRIANO: 111
- Whether: 41
- love: 2221
- loves :203
- the: 25578
- whether: 79
- we: 2922
- zodiac: 1

(b) Number of different words: 29183

Command: hadoop fs –cat out_wc/part-r-00000 | wc –l

(c) Efficiency: we can Provide side data to be used in Mappers and Reducers such as look-up tables, lists and libraries. Then we can filter out irrelevant words such as stop words. WordCount can be more efficient because fewer words need to be processed.

Quality: we can separate the sentimental words into several groups according to their meaning and count their occurrences separately so that the sentiment analysis can be more accurate.

(d) A RecordReader is little more than an iterator over records, and the map task uses one to generate record key-value pairs, which it passes to the map function.

Input: raw input data.

Output: key-value pairs.

(e) Output writer is responsible for writing those files. It produces output folders and files in HDFS. Each file corresponds to the output of a Reducer.

Problem 2: Skew (30%)

(a) There will be significant skew. Because different words have different occurrences. Some of them can be large, such as *the*. While others can be rare. So different reducers take different amounts of time. If we make each reducer a separate Reduce task, then the tasks themselves will exhibit skew.

(b) If we assign the reduce() functions to 100 Reduce tasks, there will be skew but not so significant. We can reduce the impact of skew by using fewer Reduce tasks than there are reducers. If keys are sent randomly to Reduce tasks, we can expect that there will be some averaging of the total time required by the different Reduce tasks. So on the contrast, if we instead assign the reduce() functions to 10000 Reduce tasks, the skew will be significant.

(c) The skew of run time taken by various reduce function will not be significant because the combination will only occur in map phase. Combiners are mini reducers that run locally on Mapper output to reduce data sent to Reducers. So in that case we can use fewer reducer, which leads to smaller skew.

Problem 3: Job Execution (30%)

(a) 1) 4 types of daemons:

ResourceManager: job scheduling

ApplicationMaster: task progress monitoring

NodeManager: run tasks and send progress reports

JobHistory: archives job metrics and metadata

2) Input: When it is possible, Map tasks are executed local to the data on the node where the block of data to be processed is stored. If it is not possible, Map task transfers the data across the network.

Output: Map tasks store their output on local disk (not in HDFS). Reduce tasks store their output in HDFS

Intermediate data: There is not data locality for shuffle and sort and Reduce tasks. Intermediate data is transferred across the network. Reduce Tasks write directly to HDFS.

(b) Hadoop does its best to run the map task on a node where the input data resides in HDFS, because it doesn’t use valuable cluster bandwidth. This is called the data locality optimization. It is applicable in map phase.

If tasks cannot be executed local to the data, an off-rack node is used, which results in an inter-rack network transfer.