AIM III – Large Scale Data Analysis and Data Mining

First Assignment

1. WordCount - „Hello World“ of MapReduce

We'll start with the classic MapReduce example of counting words. Your task is to complete the code in *de.tuberlin.dima.aim.exercises.one.FilteringWordCount*. The output of this job should be a textfile holding the following data per line: *word* < *tab* > *count*.

An additional requirement here is that stop words like „to“, „and“, „in“ or „the“ should be removed from the input data and all words should be lowercased.

2. A custom Writable

You will work on your first custom Writable object in this task. Have a look at *de.tuberlin.dima.aim.exercises.one.PrimeNumbersWritable*. This class models a collection of prime numbers. Writable classes need to be able to serialize to and deserialize from a binary representation. Enable that for our custom Writable by implementing *write*(*DataOutput out*) and *readFields*(*DataInput in*).

3. Average temperature per month

Have a look at the file *src/test/resources/one/temperatures.tsv*. It contains the output of a fictional temperature sensor, where each line denotes the year, the month and the temperature of a single recording. Additionally a quality parameter is included which expresses how „sure“ the sensor was of a single measurement: *year* < *tab* > *month* < *tab* > *temperature* < *tab* > *quality*

Your task is to implement a MapReduce program that computes the average temperature per month of year. It should ignore all records that are below a given minimum quality. The output of your program will be a textfile holding the following data per line: *year* < *tab* > *month* < *tab* > *average temperature* Use *de.tuberlin.dima.aim.exercises.one.AverageTemperaturePerMonth* as a starting point.

Deadline

Please send your solution as a patch file to *ssc@apache.org* with *isabel@apache.org* in CC.