Homework 8 (We Only Count 40 out of 70 Points!)

As for the previous homework, submit one .zip file containing four text files named Q1.txt to Q4.txt, each containing the answer to the corresponding question (or leave the file empty if you chose not to work on that question).

***** Important: since you have less time than usual for this last homework, you only need to obtain 40 points out of the 70 points offered for the best possible grade. *****

In the next three questions (Q1 to Q3), we describe an application scenario for data processing. We want you to recommend one of the systems or frameworks we have seen in class. Make sure that your proposal complies with all constraints specified in the description. Recommend the system you believe to perform best among all admissible alternatives. Write the name of your proposed system in the first line and one small paragraph with a justification in the following lines. Different solutions are acceptable for some of the questions.

Q1) (10 Points) We want to implement a globally accessible online interface for a bank. Users may issue money transfers and check their account balance over the interface. For an optimal user experience, we care about the following: if users issue a money transfer and inspect their account balance *immediately* afterwards, the account balance must already reflect the money transfer.

Q2) (10 Points) We want to analyze a large relational data set, containing for each customer hundreds of attributes in a single table (with millions of rows). We will be mainly running aggregation queries, counting customers with specific properties (e.g., queries for customers from specific geographical areas).

Q3) (10 Points) We want to run an iterative analysis algorithm on a large collection of text documents (which is too large to be processed by a single machine).

Q4) (40 Points) Write a MapReduce program (i.e., write pseudo-code for the map and for the reduce function) which takes a collection of text documents as input and counts for each word length the number of occurrences of words of that length appearing in the text documents.

The input to the MapReduce program is a set of key-value pairs where the key is the name of a document and the value is the text content of that document. The output of the MapReduce program is a set of key-value pairs where the key is the length of a word and the value is the number of occurrences of words of that length appearing in all the text documents.

Example Input:	<Doc1, “This is a test”>, <Do2, “This is also a test”>
Example Output:	<4,5>, <1,2>, <2,2>

For your pseudo-code, use a similar notation as seen in the lecture. You can use the auxiliary function Length(w) to calculate the length of a word w.