TDDE31 Big Data Analytics

Exam
Part 2

August 20, 2020
9:30 – **12:00**

Instructions: See https://www.ida.liu.se/~732A54/exam/distanceexam.en.shtml

Grades: You can get up to 15 points for this second part of the exam. Together with the max 14 points for the first part, you thus may get an overall of max 29 points. To pass the exam (grade 3 or E) you have to meet both of the following two conditions: First, you need to achieve at least 7 of the 14 points that can be achieved in the first part of the exam. Second, for both parts together, you need to achieve at least 14.5 of the 29 points that can be achieved overall. If you do not meet the first condition, your second part will *not* be considered for grading.

After fulfilling the aforementioned requirements to pass the exam, then for grade 4, you need at least 20 points (for both parts together), and for grade 5, you need at least 26 points.

Questions: If you have clarification questions regarding some of the exercises in the exam, please do the following depending on the exercise.

If you need clarifications on Questions 13–16, then email christoph.kessler@liu.se

If you need clarifications on Question 17, then email jose.m.pena@liu.se

If you need clarifications on Questions 10–12, or about something more general related to the exam, the examiner will be available in the following Zoom meeting room throughout the whole time of the exam.

https://liu-se.zoom.us/j/61055128986?pwd=SDlpOE04U2tEblltMHRGMWx3Vi9IZz09
Meeting ID: 610 5512 8986
Password: 588119

Notice that this Zoom meeting room has been set up using the waiting room feature of Zoom. Hence, when you enter, you will be put into the waiting room and, from there, you will then be admitted to the meeting room to ask your question.

Question 10 (1 + 1 = 2p)

Recall Question 1 of the first part of this exam where, for a given key-value database (copied again below), you had to describe how the types of queries typically implemented in a key-value store can be used for a particular data retrieval request (namely, to retrieve the names of all users liked by a given user, specified by his/her ID).

```
"alice_in_se" → "Alice, 1987, [bob95 charlie]"
      "bob95" → "Bob, 1995, [charlie]"
    "charlie" → "Charlie, 1996, []"
     "selaya" → "Alice, 1974, [charlie]"
```

(a) Describe how the given key-value database can be changed/extended such that the same data retrieval request can be done more efficiently (note: your solution still has to be a key-value database). Your description must also say explicitly how the data retrieval request can be done now after the change of the database.

(b) Discuss the pros and, in particular, the cons of your solution to change/extend the database.

For each of these two tasks, write a maximum of 200 words, respectively.

Question 11 (1p)

Describe a *concrete* application / use case for which *read scalability is important* but *data scalability is not important*.

To answer this question write a maximum of 200 words.

Question 12 (1p)

Assume we use *consistent hashing* to map keys to cluster nodes in a distributed key-value store. Consider the following claim:

> *When nodes are added or removed, every key has to be remapped.*

Is this claim correct or wrong? Justify your answer in two to four sentences.

Question 13 (1.5p)

In MapReduce, sometimes workers might be temporarily slowed down (e.g. due to repeated disk read errors) without being broken. Such workers could delay the completion of an entire MapReduce computation considerably. How could the master process speed up the overall MapReduce processing if it observes (how?) that some worker is late?

To answer this question write a maximum of 200 words.

Question 14 (1.5p)

Reconsider the *geometric mean* MapReduce program of Part 1 of this exam, applied to an overall (HDFS) input of N numbers. What is (i) the *work* (as defined in the lecture) performed by this MapReduce program on already distributed input of size N numbers, and what is (ii) the (parallel) *time* if executed as M mapper and R reducer tasks on $P > 1$ cluster nodes? Derive parametric formulas for work and time (use big-O notation where applicable) and justify your answer.

Assume for simplicity that a cluster node runs one task at a time, that reading/writing a single number from/to HDFS takes constant time, communicating K numbers costs time $a \cdot K + b$ for constants $a, b > 0$, and that additions and multiplications of two numbers take constant time too. If you need to make any further assumptions, state them carefully.

To answer this question write a maximum of 200 words.

Question 15 (1p)

From a performance point of view, is it better to have long lineages or short ones in Spark programs? Motivate your answer (technical explanation).

To answer this question write a maximum of 100 words.

Question 16 (1p)

We know that Spark offers support for *stream computing*, i.e., computing on very long or even infinite data streams. Which *fundamental* property of stream computations makes it possible to overlap computation with data transfer?

To answer this question write a maximum of 100 words.

Question 17 (6p)

You are asked to implement in Spark (PySpark) bootstrapping to estimate the expectation of the weight vector in logistic regression. In slide 12 of lecture 11, you can find a PySpark implementation of logistic regression. Of course, the weight vector $\boldsymbol{w}$ returned depends on the training database available. So, you may wonder what the expected value of $\boldsymbol{w}$ across all possible training databases is. However, you do not have access to all possible training databases. So, you cannot compute the expected value. You do not have either access to a sample of training databases to compute an (approximate) estimate of the expected value. You just have access to one training dataset. However, you can get a reasonable estimate with the help of boostrapping. This technique samples (for simplicity, with no replacement) a number of times (say, 100) the training database available to construct resampled databases. Then, logistic regression is run in each resampled database. Finally, the mean of the weight vectors obtained from the resampled databases is reported.

To get full points you need to comment your code.