732A54 Big Data Analytics

Exam
Part 2

June 1, 2021
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 requirements to pass the exam, then for grade D, you need at least 18 points (for both parts together); for grade C, you need at least 21 points; for grade B, you need at least 24 points; for grade A, you need at least 27 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 15–19, then email christoph.kessler@liu.se

If you need clarifications on Question 20, then email jose.m.pena@liu.se

If you need clarifications on Questions 11–14, 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/69035369731?pwd=U3h1N2taRk95d2JkY1JnMUttTmwvZz09
Meeting ID: 690 3536 9731
Passcode: 790863

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 11 (1p)

Describe a *concrete* application / use case in which *data scalability* is ***not*** important, and explain why data scalability is not important in this case.

To answer this question write about five to ten sentences.

Question 12 (1p)

Remember that key-value databases may easily be partitioned based on the keys. However, just by itself this idea of horizontal partitioning is not sufficient to enable distributed key-value stores to process query requests very efficiently. In contrast, there is another fundamental design decision for such systems that, in combination with horizontal partitioning, is the reason for the high query performance that these systems achieve. What is this other design decision and how is it related to achieving high query performance if the data is horizontally partitioned?

(Note, the answer to this question has *nothing* to do with consistency guarantees.)

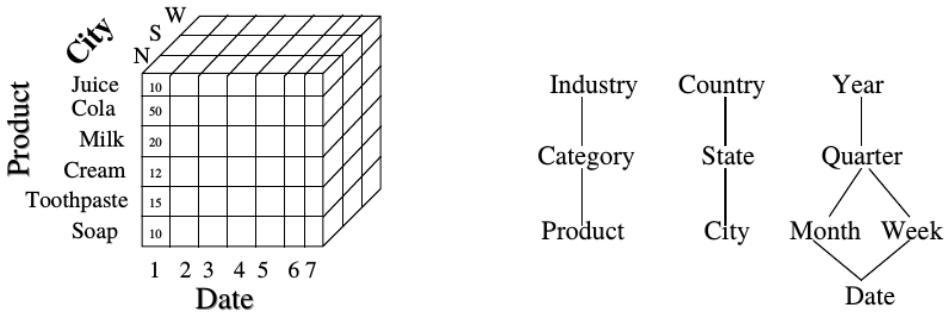
Question 13 (1p)

Consider a NoSQL system which does not provide strong consistency, but only some form of weak consistency. Remember that weak consistency means that, after updating the value of a data item in a database managed by such a system, there is no guarantee that all subsequent requests to retrieve the value of that data item will return the updated value. Explain in two to five sentences how it may happen that the old value will still be returned in response to such a request.

Question 14 (1p)

Recall that the data of a data warehouse may be represented using the multidimensional data model where numeric measures are captured in a multidimensional array and the attributes of some dimensions form hierarchies. Figure 1a illustrates an example of such a multidimensional array with three dimensions: a product dimensions, a city dimensions, and a date dimension. The numeric measures in this example are sales values; that is, any value in this array represents how often the corresponding product was sold in the corresponding city at the corresponding date. Figure 1b illustrates possible hierarchies for the three dimensions.

Considering the typical types of operations over multidimensional data (e.g., slicing, dicing, drill-down), specify which operation(s) has/have to be applied to this example data if you want to analyze and compare the total sales of a specific product, say toothpaste, in a specific city, say Linköping, for all the different years. While you do not have to define the operation(s) formally, try to be as concrete as possible. For instance, do not just write that drill-down has to be applied, but be



(a) (b)

Figure 1: Example of a multidimensional array with corresponding hierarchies for the dimensions.

concrete about the specific drill-down operation that you have in mind. If multiple operations have to be applied one after another, specify them in the order in which you would apply them.

Question 15 (1p)

What is the advantage for the *owner* of a cluster resource with many big-data computing users to use a system like Mesos or Yarn, compared to a batch reservation scheduler as known from HPC clusters? Explain your answer (technical reasons).

To answer this question write a maximum of 150 words.

Question 16 (1p)

Given is a distributed HDFS text file that contains the complete log of all LiU student course registrations up to now. The file consists of N lines that are each of the form `LiU-ID:coursecode:year`, e.g.,

```
...
liuid123:TDDE31:2020
abcde789:TDDE31:2021
liuid123:TDDE99:2020
...
```

Assume you would write a *MapReduce* program (you do *not* have to actually write it!) that produces an HDFS text file that contains one line for each course code, showing the number of registrations; for example:

```
...
TDDE30:174
TDDE31:123
...
TDDE99:27
...
```

Notice, as the HDFS file is not extremely large, the number of mapper tasks will be rather low but larger than one.

Do not write that program. Instead, answer the following question *and* motivate your answer: Will it be beneficial here to use a combiner or not?

To answer this question write a maximum of 100 words.

Question 17 (1p)

For HDFS with 64MB blocks and the usual replication scheme for fault tolerance, we run a single MapReduce step using only the mapping phase. The input is a large HDFS text file containing N lines. Each line contains 64 bytes of text consisting of a log message and a measured floating point value. The mapper user function leaves the log message unchanged, keeps positive values unchanged and changes negative values to zeroes.

How many HDFS block reads and how many HDFS block writes are performed in total for $N = 10^9$? Give a short explanation and calculation.

To answer this question write a maximum of 100 words.

Question 18 (1p)

On an HDFS cluster, according to which main criterion does the MapReduce scheduler in the master process assign mapper tasks to the workers? And why is this even more important if the cluster's interconnection network has very limited throughput capacity?

To answer this question write a maximum of 100 words.

Question 19 (1p)

On iterative computations that consist of many transformations, Spark significantly outperforms MapReduce. Why? Provide a technical explanation.

To answer this question write a maximum of 150 words.

Question 20 (6p)

Once we have a kernel method solution to the density estimation problem, it is easy to turn it into a probabilistic classifier. To see it, say that we have to predict the class label for a point $X = x$. Then, you can do the following:

1. Estimate $f(X = x|Y = 0)$ and $f(X = x|Y = 1)$ using kernel methods.
2. Estimate $p(Y = 0)$ and $p(Y = 1)$ using maximum likelihood, i.e. estimate these quantities as the proportions of training points labeled with $Y = 0$ and $Y = 1$, respectively.
3. Use Bayes theorem to compute $p(Y = 0|X = x)$ as

$$p(Y = 0|X = x) = \frac{f(X = x|Y = 0)p(Y = 0)}{f(X = x|Y = 0)p(Y = 0) + f(X = x|Y = 1)p(Y = 1)}$$

   and analogously for $p(Y = 1|X = x)$.

Your task is to implement in PySpark the kernel-based probabilistic classifier described above, in order to predict $p(Y = 0|X = 1)$.

To get full points you need to comment your code.