732A54 / TDDE31
Big Data Analytics

Exam
Part 2

June 2, 2020
9:50 – 12:00

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

Question 11 (1p)

Describe a *concrete* application / use case for which *data scalability* is important. Note that "scalability" is a key word in this question; it is *not* enough if your application / use case simply has to do with a huge amount of data.

To answer this question write a maximum of 200 words.

Question 12 (1 + 1 = 2p)

(a) Consider the following key-value database which contains four key-value pairs where the keys are user IDs and the values consist of a user name, the user's year of birth, and an array of IDs of users that the current user likes (for instance, Bob likes Charlie).

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

Describe how the types of queries typically implemented in a key-value store can be used to retrieve the birth years of all users named Alice.

(b) Describe how the given key-value database can be changed/extended such that retrieving the birth years of users named Alice becomes more efficient.

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

Question 13 (1p)

Assume we use *consistent hashing* to map keys to compute nodes in a distributed key-value store. Explain in about three to five sentences what happens in such a system if a compute node is removed.

Question 14 (0.5p)

Does a combiner function need to be commutative? Explain why or why not.

To answer this question write a maximum of 100 words.

Question 15 (0.5 + 1 = 1.5p)

(a) In the lecture, the MapReduce mechanism was referred to as the "Swiss Army Knife" of distributed parallel big-data computing over distributed files. Why?

To answer this question write a maximum of 50 words.

(b) Explain how the Spark programming interface differs, and what the benefit could be.

To answer this question write a maximum of 150 words.

Question 16 (1.5p)

Why is fault tolerance an important aspect in big-data computations? Compare the fault tolerance mechanisms in MapReduce and Spark to each other.
(Note that "compare" means compare, not just present each one for itself.)

To answer this question write a maximum of 300 words.

Question 17 (1.5p)

Consider the MapReduce wordcount program as in our lecture but *without* a Combiner. The user has configured the MapReduce framework to use $M > 1$ mapper tasks but only one reducer task. Assume for simplicity that mapper tasks and reducer tasks each take time proportional to their input size, i.e.,

$$time_{mapper}(N) = time_{reducer}(N) = C \cdot N \text{ seconds}$$

for some constant $C > 0$ and the amount N of data the task processes. Assume also for simplicity that a mapper reading N bytes of input will also produce N bytes of output, and that the amount of input data in HDFS is large enough to provide work for at least M mappers.

Derive a tight upper bound for the speedup that the user can achieve with M mapper tasks and 1 reducer task (even when using an arbitrarily large number of execution resources on the cluster), compared to a sequential scenario (i.e., 1 mapper and 1 reducer task). Justify your answer. A formal calculation is expected. (Hint: Which fundamental theorem of parallel computing can you apply here?)

To answer this question write a maximum of 200 words.

Question 18 (6p)

You are asked to implement in Spark (PySpark) the perceptron algorithm. This algorithm for binary classification is a predecessor of modern neural networks.

Specifically, consider a binary classification problem with class labels $t \in \{-1, +1\}$. Then, the class label assigned to a point $\boldsymbol{x}$ is given by

$$y(\boldsymbol{x}) = \begin{cases} +1 & \text{if } \boldsymbol{w}^T\boldsymbol{x} \geq 0 \\ -1 & \text{if } \boldsymbol{w}^T\boldsymbol{x} < 0. \end{cases}$$

To determine the parameter values $\boldsymbol{w}$, some learning data $\{(\boldsymbol{x}_1, t_1), \ldots, (\boldsymbol{x}_N, t_N)\}$ is available. We seek the parameter values that minimize the following error function:

$$E(\boldsymbol{w}) = -\sum_{n \in \mathcal{M}} \boldsymbol{w}^T \boldsymbol{x}_n t_n$$

where $\mathcal{M}$ denotes the set of all misclassified instances in the learning data, i.e., correctly classified instances give zero error. The error function above is minimized by using batch gradient descent, i.e., the weights are iteratively updated as follows:

$$\boldsymbol{w}^{(new)} = \boldsymbol{w}^{(old)} - \alpha \nabla E(\boldsymbol{w}^{(old)}) = \boldsymbol{w}^{(old)} + \alpha \sum_{n \in \mathcal{M}} \boldsymbol{x}_n t_n$$

where α is the learning rate. You can assume that α is given. You can also assume that $\boldsymbol{x}$ and $\boldsymbol{w}$ are of dimension two. Finally, you can assume that the learning data is linearly separable and, thus, the learning process will converge.

It is not required that your code actually compiles; nevertheless, it must be code rather than pseudocode, i.e., you have to use the transformations and actions properly.

To get full points you need to comment your code.