Information page for written examinations at Linköping University

Examination date	2019-11-01
Room (1)	U3(16)
Time	8-12
Edu. code	732A54
Module	TENT
Edu. code name **Module name**	Big Data Analytics (Analys av Big data) Examination (Tentamen)
Department	IDA
Number of questions in the examination	10
Teacher responsible/contact person during the exam time	Patrick Lambrix / Christoph Kessler / José Pena
Contact number during the exam time	26 05 (Q1-4) / 0703-666687 (Q5-6) / 16 51 (Q7-10)
Visit to the examination room approximately	10:00
Name and contact details to the course administrator (name + phone nr + mail)	
Equipment permitted	dictionary
Other important information	
Number of exams in the bag	

Exam

TDDE31 and 732A54
Big Data Analytics

November 1, 2019, 8-12

Grades: For a pass grade you need to obtain 50% of the total points.

Instructions:
In addition to the instructions on the cover page:

- Write clearly.
- Start the answers to a question on a new page.
- If you make assumptions that are not given in a question, then clearly describe these assumptions. (Of course, these assumptions cannot change the exercise.)
- Give relevant answers to the questions. Points can be deducted for answers that are not answers to the question.
- Answer in English.

Question 1 (2p) **Topic: Big data properties**

Give and explain 4 V's (big data properties) and give an example for each.

Question 2 (3p) **Topic: Scalability**

(a) Describe difference between *vertical scalability* and *horizontal scalability*. (1p)

(b) Describe the difference between *read scalability* and *write scalability*. (1p)

(c) 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. (1p)

Question 3 (3p) **Topic: NoSQL database models**

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

```
"alice_in_se" → "Alice, [bob95 charlie]"
      "bob95" → "Bob, [charlie]"
    "charlie" → "Charlie, []"
```

Describe how the types of queries typically implemented in a key-value store can be used to retrieve the user IDs of users named Alice. (1p)

(b) Describe how the given key-value database can be changed/extended such that retrieving the user IDs of users named Alice is more efficient. (1p)

(c) Identify two differences between the key-value database model and the document database model that was introduced in class. (1p)

Question 4 (2p) **Topic: BASE properties**

Specify what the BASE properties are. (Simply writing down the names of these properties is not enough and does not earn you any points.)

Question 5 (4p) **Topic: Cluster computing**

(a) How does a *distributed file* (in a distributed file system like HDFS) differ from a traditional file (with respect to how is it technically and logically structured, stored and accessed), and what are the two main advantages for the processing of a big-data computation over a distributed file compared to a traditional file? Be thorough! (1.5p)

(b) Define and shortly explain the following terms: (1p)

- Parallel programming model (0.5p)
- Algorithmic skeleton (0.5p)

Be general and thorough. An example is not a definition, but can illustrate a definition.

(c) Describe (by an annotated drawing and text) the *hardware* structure of modern hybrid clusters (used for both HPC and distributed parallel big-data processing). In particular, specify and explain their *memory structure* and how the different parts are *connected* to each other. (1p)

(d) Why is it important to consider (operand) data locality when scheduling tasks (e.g., mapper tasks of a MapReduce program) to nodes in a cluster? (0.5p)

Question 6 (6p) **Topic: MapReduce and Spark**

(a) What (mathematical) properties do functions need to fulfill that are to be used in *Combine* or *Reduce* steps of MapReduce, and why? (1p)

(b) The MapReduce construct is very powerful and consists of 7 substeps as presented in the lecture. Which one(s) of these substeps may involve *network* I/O, and for what purpose? Be thorough! (1p)

(c) Why and in what situations can it be beneficial for performance to use a Combiner in a MapReduce instance? (1p)

(d) What is the *RDD lineage graph* and how is it used in Spark for the efficient execution of Spark programs? (1p)

(e) What is an (RDD) "transformation" in Spark? Give also one example operation of a transformation. (0.5p)

(f) What does the *collect* operation in Spark do with its operand RDD? (0.5p)

(g) What is streaming, in general? For what type of computations can it be suitably used? And how does Spark support streaming? (1p)

Question 7 (1p)

Why is Spark more suitable than MapReduce for implementing many machine learning algorithms ?

Question 8 (3p)

Implement in Spark (PySpark) the following k-means algorithm.

```
Assign each point to a cluster at random
Compute the cluster centroids as the averages of the points assigned to each cluster
Repeat the following lines l times
    Assign each point to the cluster with the closest centroid
    Update the cluster centroids as the averages of the points assigned to each cluster
```

You can use the functions `randint(A,B)` which produces a random integer in the given interval, and `distance(A,B)` which returns the distance between two points.

Question 9 (3p)

Implement in Spark (PySpark) logistic regression. Recall from the lectures that we consider a binary classification problem with class labels $t \in \{-1, +1\}$ and a model $y(\boldsymbol{x}) = \boldsymbol{w}^T \boldsymbol{x}$ so that the posterior class distribution is

$$p(t = +1|\boldsymbol{x}) = \sigma(y(\boldsymbol{x})) = \frac{1}{1 + \exp(-y(\boldsymbol{x}))}$$

$$p(t = -1|\boldsymbol{x}) = 1 - \sigma(y(\boldsymbol{x})) = \frac{1}{1 + \exp(y(\boldsymbol{x}))}.$$

Thus, given some training data $\{(\boldsymbol{x}_1, t_1), \ldots, (\boldsymbol{x}_N, t_N)\}$, the negative log-likelihood becomes

$$L(\boldsymbol{w}) = \sum_{n=1}^{N} \log(1 + \exp(-t_n y(\boldsymbol{x}_n)))$$

whose gradient is given by

$$-\sum_{n=1}^{N} t_n (1 - 1/(1 + \exp(-t_n \boldsymbol{w}^T \boldsymbol{x}_n)))\boldsymbol{x}_n.$$

Specifically, you are asked to implement gradient descent to find the maximum likelihood estimates of $\boldsymbol{w}$.

Question 10 (3p)

Implement in Spark (PySpark) the k-nearest neighbors algorithm for classification. This algorithm receives as input a point to classify, finds the k closest points in the training data, and assigns the input point to the majority class of the k closest points.

Assume that the training data is available to you in the RDD `mydata`. This is a key-value pairs RDD. The key is the class label, and the value is a tuple with the predictive attribute values. Assume that we are dealing with binary classification, and that the class labels are 0 and 1.

You may want to use the transformation `sortBy(lambda x: x[i])` to sort the RDD ascending according to the column `x[i]`, and the action `take(n)` to return the first `n` rows in the RDD. You can also use the function `distance(A,B)` which returns the distance between two points.