Information page for written examinations at Linköping University

Examination date	2019-08-22
Room (1)	TER3(20)
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	13
Teacher responsible/contact person during the exam time	Patrick Lambrix /Christoph Kessler / José Pena
Contact number during the exam time	26 05 (Q1-7) / 0703-666687 (Q8-10) / 16 51 (Q11-13)
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	

Information page for written examinations at Linköping University

Examination date	2019-08-22
Room (1)	TER1(9)
Time	8-12
Edu. code	TDDE31
Module	TEN1
Edu. code name **Module name**	Big Data Analytics (Big Data Analytics) Written exam (Skriftlig tentamen)
Department	IDA
Number of questions in the examination	13
Teacher responsible/contact person during the exam time	Patrick Lambrix / Christoph Kessler / José Pena
Contact number during the exam time	26 05 (Q1-7) / 0703-666687 (Q8-10) / 16 51 (Q11-13)
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

August 22, 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 (1p) **Topic: NoSQL motivation**

Describe one of the many new challenges for database systems that NoSQL systems aim to address.

Question 3 (1p) **Topic: Scalability**

Consider the following claim:

> *While read scalability can be achieved by scaling horizontally (scale out), it cannot be achieved by scaling vertically (scale up).*

Is this claim correct or wrong? Justify your answer!

Question 4 (1p) **Topic: Scalability**

Describe a *concrete* application / use case for which *data scalability* is important.

Question 5 (1p) **Topic: NoSQL database models**

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 names of all users that the user with ID `alice_in_se` likes.

Question 6 (2p) **Topic: NoSQL database models**

Describe the basic form of the wide-column database model and compare it to the relational data model (SQL databases).

Question 7 (2p) Topic: CAP theorem

Recall that the CAP theorem considers three possible properties of distributed systems: Consistency, Availability, and Partition Tolerance. Specify (a) what each of these properties means, and (b) what the CAP theorem says about them.

Question 8 (3p) 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 is/are the advantage(s) for the processing of a big-data computation over a distributed file compared to a traditional file? Be thorough! (1.5p)
(b) 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)
(c) 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 9 (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) "action" in Spark? Give also one example operation of an action. (0.5p)
(f) How does Spark-streaming differ from ordinary Spark? For what type of computations can it be suitably used? What is "windowing" in Spark streaming, and what is its (technical) purpose? (1.5p)

Question 10 (1p) Topic: Cluster resource management

Cluster resource management could be done either within the runtime system of a specific cluster programming framework (such as Spark or MapReduce) or be delegated to a separate cluster resource management layer, such as batch schedulers like Slurm or alternative approaches like MESOS or YARN.

(a) What is/are the main advantage(s) of delegating resource management to MESOS or YARN for the *owner* of a cluster resource (e.g., a data center)? Justify your answer. (0.5p)

(b) Why are resource management frameworks like MESOS and YARN more suitable for workloads consisting of (large) MapReduce jobs, *in comparison to batch schedulers*? (0.5p)

Question 11 (5 p) Topic: Machine Learing for Big Data

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 12 (4 p) Topic: Machine Learing for Big Data

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.

Question 13 (1 p) Topic: Machine Learing for Big Data

Describe in text how you would modify your implementation of the k-nearest neighbors algorithm to predict the class label of more than one point.