Exam

TDDE31 and 732A54
Big Data Analytics

May 29, 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)

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

Question 2 (2p)

(a) Define the notions of *read scalability* and *write scalability*. (1p)
(b) Describe an example use case / application for which read scalability is important but write scalability is not. (1p)

Question 3 (3p)

Consider the following relational database which consists of two relations (Project and Report). Notice that the attribute finalreport in the relation Project is a foreign key that references the primary key (attribute id) in the relation Report. Notice also that multiple projects may have the same final report.

Project

name	budget	finalreport
UsMis	1,000,000	391
AMee3	3,700,000	391
Bee	1,300,000	121

Report

id	pages	location
121	70	http://acme.com/beerep
391	350	http://acme.com/r391
699	100	http://acme.com/Other

Capture all the data in this relational database as
(a) a key-value database, (1p)
(b) a document database, (1p)
(c) a graph database (using the Property Graph model). (1p)

Question 4 (2p)

Describe i) the types of queries and ii) the form of data partitioning implemented in key-value stores (1p), and iii) explain how these things are related to achieving horizontal scalability (1p).

Question 5 (1p)

Assume a distributed database system in which every data item is stored on 3 nodes. For such a system to report to an application that a write operation as been finished, what is the number of nodes that are required to complete the write successfully if the system aims to achieve the consistency property as per the CAP theorem.

Question 6 (3p)

Cluster computing
(a) How does a distributed file (in a distributed file system like HDFS) differ from a traditional file, and what is the advantage for the processing of a big-data computations over a distributed file compared to a traditional file? (1p)
(b) Describe how modern hybrid clusters used for distributed parallel big-data processing are organized. In particular, specify and explain their control structure and memory structure. (1.5p)
(c) Why is it important to consider (operand) data locality when scheduling tasks (e.g., mapper tasks of a MapReduce program) to the nodes a cluster? (0.5p)

Question 7 (7p)

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) Which substeps of the MapReduce construct involve disk I/O, and for what purpose? (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 a RDD in Spark? Be thorough! (1p)
(e) Spark classifies its functions on RDDs into two main categories: "Transformations" and "Actions". Describe the main difference between those, and give one example operation for each category. (1p)
(f) How can Spark be used with input data that arrives in a continuous stream (e.g., from external sensors over the network)? In particular, how can such a data stream be structured for processing by Spark? (1p)
(g) Describe the execution model of Spark programs. In particular, there exist 2 different kinds of processes, driver and workers. Explain in general which operations of a Spark program are executed by each of them. (1p)

Question 8 (5 p)

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

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

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

Question 9 (4 p)

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 10 (1 p)

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.