Exam

TDDE31 and 732A54
Big Data Analytics

June 2, 2018, 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 (1p)

Explain and compare the concepts of vertical and horizontal scalability.

Question 3 (2p)

Give and explain the CAP theorem. Explain the notions in the CAP theorem.

Question 4 (3p)

P1, P2 and P3 are three distributed processes. The events following below have occurred during the processes and the values for their vector clocks are given:

```
P1: A (0,0,0); B (1,0,0); C (2,0,0); D (3,0,0); E (4,0,2)
P2: F (0,0,0); G (1,1,0); H (2,2,0); I (2,3,3)
P3: J (0,0,0); K (0,0,1); L (0,0,2); M (0,0,3)
```

Draw the temporal relationships between the events of the processes. Name the relationships between the following two pairs of events and explain (with the help of the respective formal rules) how you have determined them:
- B (1,0,0) and K (0,0,1)
- I (2,3,3) and E (4,0,2).

Question 5 (2p)

Compare HBase and Dynamo according to their data models. Expplain the data models.

Question 6 (1+1=2p)

Memory hierarchy
(a) Explain (including an annotated drawing) the principle of memory hierarchy as used in modern server computers.
(b) What is the purpose of memory hierarchy in computer architecture? What kind of programs are expected to benefit from memory hierarchy, and why?

Question 7 (1+1=2p)

Cluster computing
(a) 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?
(b) Why should servers in datacenters running I/O-intensive tasks (such as disk/DB accesses) get many more tasks to run than they have cores?

Question 8 (1+1+1+1+1=5p)

MapReduce and Spark
(a) Which parallel algorithmic design pattern (name and short explanation) is used for the *parallel reduction* algorithm?
(b) What properties do functions need to fulfill that are to be used in *Combine* or *Reduce* steps of MapReduce, and why?
(c) Which steps of MapReduce involve disk I/O, and for what purpose?
(d) 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.
(e) For what type of computations (structural property) will Spark outperform MapReduce considerably, and why?

Question 9 (1p)

Cluster Resource Management Systems
Which benefits do systems such as YARN and Mesos bring
(a) to the owner/operator of a cluster computer in a data center?
(b) to the user running MapReduce jobs?
Explain your answers.

Question 10 (1p)

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

Question 11 (4p)

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 12 (5p)

Implement in Spark (PySpark) the following naive Bayes classifier. Upper-case letters denote random variables and lower-case letters their values. There are two predictor variables (A and B) and one class variable (C). All the variables are binary. That is, A, B and C take values in the set $\{0, 1\}$. The naive Bayes classifier classifies according to the following posterior probability distribution

$$p(c|a,b) = \frac{p(a|c)p(b|c)p(c)}{p(a|c)p(b|c)p(c) + p(a|\overline{c})p(b|\overline{c})p(\overline{c})}$$

where $\overline{c} = 1 - c$. The equality above follows from the fact that the naive Bayes classifier naively assumes that the predictor variables are independent given the class variable, hence the name.

Since we do not have access to the probabilities in the equation above, we replace them with their maximum likelihood estimates. That is, we replace $p(a|c)$ with $\hat{p}(a|c) = N_{ac}/N_c$ where N_{ac} is the number of points in the learning data where A and C take values a and c simultaneously, and N_c is the number of points in the learning data where C takes value c. Similarly for $p(a|\overline{c})$, $p(b|c)$, $p(b|\overline{c})$, $p(c)$ and $p(\overline{c})$. The learning data is available in the file data.txt. Each row follows the format "a;b;c". That is, each row contains values for A, B and C separated by semicolons.

Implement the naive Bayes classifier described above to compute $p(0|0,0)$.