Information page for written examinations at Linköping University

Examination date	2017-01-18
Room (1)	TER1(29)
Time	8-12
Course code	732A54
Exam code	TENT
Course name **Exam name**	Big Data Analytics (Analys av Big data) Examination (Tentamen)
Department	IDA
Number of questions in the examination	14
Teacher responsible/contact person during the exam time	Christoph Kessler, Jose Pena, Valentina Ivanova (Patrick Lambrix)
Contact number during the exam time	2406 / 1651 / 2605
Visit to the examination room approximately	10:00
Name and contact details to the course administrator (name + phone nr + mail)	Annelie Almquist, 2934, annelie.almquist@liu.se
Equipment permitted	dictionary
Other important information	
Number of exams in the bag	

Exam

732A54 Big Data Analytics

January 18, 2017, 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 (4p)

Compare OLTP, OLAP, NoSQL and NewSQL systems according to their data models.

Question 2 (1p)

Explain and compare the concepts of vertical and horizontal scalability.

Question 3 (2p)

Explain (with the help of a figure) how versioning and reconciliation of divergent versions work in Dynamo DB.

Question 4 (2+1=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)
```

a) Illustrate the ordering of the events in each process and the communication between the processes (sent and received messages) on a figure where a separate axis represents the physical time for every process; the time increases from left to right.

b) Give an example of concurrent events, (write down a pair of events which are concurrent and explain why they are concurrent).

Question 5 (1p)

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 6 (1p)

The block size used in the Hadoop distributed file system is, by default, many Megabytes large (typically, 64MB). Considering the way how computer systems and the parallel computations with Hadoop MapReduce are organized, explain why a block size in this order of magnitude is usually a reasonable choice from a performance point of view, i.e., why one should not use very small block sizes (e.g. just a few dozen bytes).

Question 7 (1p)

Give an example of some computation of your choice (high-level description, pseudocode as appropriate, but no code details) that can be expressed in parallel with the MapReduce programming model but requires no Reduce functionality (give the justification).

Question 8 (3p)

The *Combiner* is optional in MapReduce. (a) What does the combiner do, (b) why could it be omitted from a correctness point of view, and (c) in what kind of scenarios can it be beneficial (and why?) to use a Combiner?

Hint: Be thorough. Your answer will show how well you have understood the MapReduce programming model.

Question 9 (1p)

What is a *Resilient Distributed Dataset (RDD)* in *Spark*?

Question 10 (2p)

For a Spark program consisting of 2 subsequent Map computations, show how Spark execution differs from Hadoop/MapReduce execution and explain why Spark execution is usually more efficient.

Question 11 (1p)

What is the purpose of systems like Mesos and YARN?

Question 12 (5p)

Implement in Spark (PySpark) the moving window classifier. This is a classifier that only considers the training points that are at most at a Euclidean distance h from the point to classify and, then, issues the majority class label. In other words, in a two class domain, the class label assigned to a point $\boldsymbol{x}$ is given by

$$y(\boldsymbol{x}) = \begin{cases} 0 & \text{if } \sum_{n=1}^{N} \mathbf{1}_{\{t_n=1, \boldsymbol{x}_n \in S(\boldsymbol{x},h)\}} \leq \sum_{n=1}^{N} \mathbf{1}_{\{t_n=0, \boldsymbol{x}_n \in S(\boldsymbol{x},h)\}} \\ 1 & \text{otherwise} \end{cases}$$

where $\{\boldsymbol{x}_n, t_n\}_{n=1}^{N}$ is the training data, t_n is the class label for the n-th training point, $S(\boldsymbol{x}, h)$ is a D-dimensional closed ball of radius h centered at $\boldsymbol{x}$, and the function $\mathbf{1}_{\{condition\}}$ returns 1 if the condition is satisfied and 0 otherwise.

Question 13 (4p)

Cross-validation is a technique to estimate the error of a classifier. It works as follows:

```
Split the training data into K folds of roughly equal size
For i in 1 : K
    Train the classifier on all the folds but the i-th
    Test the classifier on the i-th fold
Report the average of the K test errors
```

You are asked to implement in Spark (PySpark) cross-validation to estimate the error of the classifier built in the previous question. To run the classifier, simply call the function $MWC(x, d, h)$ where x is the point to classify, l is a list with the indexes of the folds to use as training data, and h is the radius of the ball.

Question 14 (1p)

What is the main feature that distinguishes the Spark framework from the MapReduce framework ? Show where this feature occurs in your solutions to the two previous questions.