This exam is **closed-book** (no electronics allowed, except pocket calculators; no paper materials allowed).

Write your **name and student number** on the first page (below this box). In case any of the pages get detached from the rest, also write your student number in a header of each sheet of paper.

Each question is marked with a number of percentage points (for example, 10 %), and they add up to 100 %. Give your answers **on this paper** in the space provided. Design your answer on **scratch paper before** you start writing here, or you may run out of space on this paper. Handwrite neatly.

All questions require you to explain your answer. The explanation must be **correct and complete** in order to get the points for that question.

Name and student number: ________________________________

Question 1 *(12 %)*

On the Google File System (GFS):

1. Summarize what metadata is kept by the master node.
   Answer:

2. In what type of memory is the metadata stored on the master node, and why is this advantageous?
   Answer:

3. If two or more applications want to overwrite the same data chunk (starting at the same offset) concurrently, what does GFS guarantee about the resulting chunk data?
   Answer:

Question 2 *(6 %)*

Take a GFS computing cluster with 300 chunk servers, each with a free disk space of 10 TB, 10 GB of RAM available, a chunk size of 100 MB, the standard replication factor of 3. About 10 KB of metadata needs to be stored per chunk handler. You want to store a very big file on this cluster.

What is the maximum file size that is possible to store on this cluster, and why?

Answer:

Question 3 *(12 %)*

A company has a computing cluster with HDFS. This cluster has a total of 1 PB of free disk space, and 1 TB of free memory.

The company hasn't installed a big-data processing framework in this cluster yet. The company asks you to develop certain types of big-data processing software (listed below), and also to state which big-data processing framework should be installed on the computing cluster to best support this future software.

For each type of software below, encircle either **MapReduce** or **Spark**, and explain each answer briefly.

1. **MapReduce** **Spark** A distributed sort of 100 GB of input data from the company's weekly logs. The company will want to run this code over night, and pick up the results every morning.
   Why:

2. **MapReduce** **Spark** A word count over 10 TB of input data from their Web crawl. The company will want to run this code over night, and pick up the results every morning.
   Why:

3. **MapReduce** **Spark** Some in-depth statistics (aggregation and regression) over 10 GB of the latest log data. This software will be run every 2 hours, and should run as fast as possible.
   Why:

Question 4 *(18 %)*

Show your understanding of the Spark framework by answering the following questions succinctly:

1. What does *wide dependency* mean in Spark?
   Answer:

2. Give examples of Spark transformations which have a *wide dependency.*
   Answer:

3. What is a *partitioning function* (also called *partitioner*) in Spark?
   Answer:

4. Give 2 examples of *partitioning functions* commonly used by Spark. For each partitioning function, give at least one Spark method which uses it.
   Answer:

5. In Spark, in which type of memory are intermediate results stored?
   Answer:

6. What are the differences between RDDs and DataFrames?
   Answer:

Question 5 *(10 %)*

You had Spark jobs running on a cluster. Here are some practical questions about working with Spark.

1. If you run a Spark job in cluster mode, what is the *maximum amount of data* you should give as input to this job? You can state the answer in any terms you see fit (for example, the amount of resources on a cluster machine).
   Answer:

2. Consider the following four Spark programs:

   (a)
   ```
   text = sc.textFile("file.txt")
   map = text.flatMap(lambda line: line.split(" "))
             .flatMap(lambda word: (word, 1))
   reduce = map.reduce(lambda (a, b): a + b)
   ```
   (b)
   ```
   text = sc.textFile("file.txt")
   map = text.map(lambda line: line.split(" "))
             .map(lambda word: (word, 1))
   reduce = map.reduceByKey(lambda (a, b): a + b)
   ```
   (c)
   ```
   text = sc.textFile("file.txt")
   map = text.map(lambda line: line.split(" "))
             .flatMap(lambda word: (word, 1))
   reduce = map.reduce(lambda (a, b): a + b)
   ```
   (d)
   ```
   text = sc.textFile("file.txt")
   map = text.flatMap(lambda line: line.split(" "))
             .map(lambda word: (word, 1))
   reduce = map.reduceByKey(lambda (a, b): a + b)
   ```

   Which of the four is a correct *word count* program over the text file in the input, and why? If none is correct, describe a correct program.
   Answer:

Question 6 *(12 %)*

Here is a short quiz over the Bigtable framework:

1. What data types can one store in Bigtable cells?
   Answer:

2. Motivate briefly if `reading` a cell from a Bigtable is fast and why.
   Answer:

3. Motivate briefly if `writing` into a Bigtable is fast and why.
   Answer:

Question 7 *(14 %)*

Both Spark Streaming and Storm need to *transmit data* between worker machines in the cluster, during the running of the application. This can be an expensive operation.

For these two frameworks, compare how the programmer can specify which data goes where.

Answer:

Question 8 *(16 %)*

Design a Bigtable-based storage for the satellite layer of the Google Maps web service. A user of this service sees in their web browser a satellite image, overlayed with some vector objects (points of interest, road names, etc.). This satellite image is actually composed from *small, square image patches* seemlessly arranged in a grid:

The service should serve images at any *zoom level* requested by the user, quickly. The image patches will have the same absolute size at any zoom level (say, 400 pixels wide).

This Google Maps has its own API. This is built on top of the very basic *BigTable operations*, which `read` either from a single row, or from a number of consecutive rows. The Google Maps API has the following operations:

- `get_map(zoom, long1, lat1, long2, lat2)`
  Returns the satellite image patches for the geographic coordinates and zoom level given.
- `get_pois(zoom, long1, lat1, long2, lat2)`
  Returns all the point of interests for the region and zoom level given.

Give a schema and motivate why your design is suitable for the task.

Discuss, for each operation in the Google Maps API, what **Bigtable operation(s)** would be executed.

Answer:

Answer (continued):

Question 9 *(0 %)*

(Optional) Please give the teacher one good suggestion on what to change in order to improve this course next year.

Answer: