Big Data Processing (Resit)
2014-2015 Q2
April 15, 2015
Lecturer: Claudia Hauff
Time Limit: 180 Minutes

Name: _____

Student number: _____

Answer the questions in the spaces provided on this exam. If you run out of room for an answer, continue on the back of the page. Mark clearly which question the answer refers to.

- Before you start, write down your name and student number on this page. On all the following pages, write your student number at the top of the page.
- This exam contains 16 pages (including this cover page) and 24 questions. Check to see if any pages are missing.
- The use of material (book, slides, laptop, etc.) during the exam is not allowed.
- The amount of points each question is worth is indicated.
- Write clearly! If your writing cannot be deciphered, it will not be considered for grading.
- Every multiple-choice question has just **one** correct answer. To select an answer, circle the letter.
- When you are asked to write pseudo-code, **do not forget** to also add specific configurations that your code might require to run correctly (Partitioner or Reducer setting, etc.).

- The total number of available points is 48.

Multiple-choice questions.

1. (1 point) **[Streaming]** A Bloom filter guarantees no ...
   A. false negatives
   B. false positives
   C. false positives or false negatives, depending on the Bloom filter type
   D. false positives and false negatives

2. (1 point) **[Hadoop]** Bob has a Hadoop cluster with 20 machines with the following Hadoop setup: replication factor 2, 128MB input split size. Each machine has 500GB of HDFS disk space. The cluster is currently empty (no job, no data). Bob intends to upload 4 Terabytes of plain text (in 4 files of approximately 1 Terabyte each), followed by running Hadoop's standard `WordCount`[1] job. What is going to happen?
   A. The data upload fails at the first file: it is too large to fit onto a DataNode
   B. The data upload fails at a later stage: the disks are full
   C. `WordCount` fails: too many input splits to process
   D. `WordCount` runs successfully

3. (1 point) **[Streaming]** The following stream is processed using the DGIM algorithm: 1 0 1 1 1 0 0 0 0 1 1 0 1 0 1 0 1 (the most recent bit appears to the right). Applying DGIM to this stream results in:
   A. [1 0 1 1 1 0 0 0 0 1 1] [0 1 0 1] [0 1]
   B. [1 0 1 1 1] 0 0 0 0 [1 1 0] [1 0 1 0] [1]
   C. [1 0 1 1 1] 0 0 0 0 [1 1 0 1] 0 [1 0 1]
   D. None of the above

4. (1 point) **[Streaming]** Which algorithm should be used to approximate the number of distinct elements in a data stream?
   A. Misra-Gries
   B. Alon-Matias-Szegedy
   C. DGIM
   D. None of the above are algorithms that approximate the number of distinct elements

[1] WordCount is a simple Hadoop job which counts the number of occurences of each word in a given input.

5. (1 point) **[GFS/HDFS]** The distributed file systems GFS and HDFS were devised with a number of use cases (data scenarios) in mind. Consider the following data storage scenarios:

   (S1) A global company dealing with the data of its one hundred million employees (salary, bonuses, age, performance, etc.)

   (S2) A Web search engine's query log (each search request by a user is logged)

   (S3) A hospital's medical imaging data generated during an MRI scan

   (S4) Data sent by the Hubble telescope to th Space Telescope Science Institute

   For which of these scenarios are GFS or HDFS a good choice?

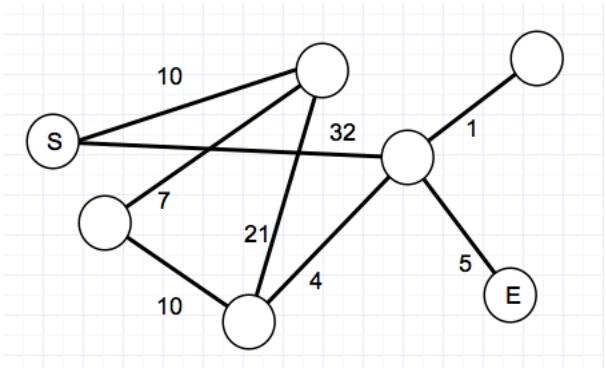
   A. Scenarios (S1) and (S4)

   B. Scenarios (S2) and (S3)

   C. Scenarios (S2) and (S4)

   D. Scenarios (S1) and (S3)

6. (1 point) **[Hadoop]** For graphs similar to the toy example shown below, the maximum number of iterations required to compute parallel breadth-first search in Hadoop depends on ...



   A. the diameter of the graph.

   B. the number of nodes in the graph.

   C. the number of edges in the graph.

   D. neither of the three aspects above.

7. (1 point) **[Hadoop]** In Hadoop, a Counter is attached to a specific ...
   A. Mapper
   B. Reducer
   C. machine in the cluster
   D. job

8. (1 point) **[Pig]** Pig is an execution engine that ...
   A. compiles Pig Latin scripts into HDFS
   B. replaces the MapReduce core in Hadoop
   C. compiles Pig Latin scripts into database queries
   D. utilizes the MapReduce core in Hadoop

9. (1 point) **[Hadoop]** The partitioner determines ...
   A. which values are assigned to the same key
   B. the order of key/value processing
   C. to which specific machine in the cluster a particular key/value pair is sent
   D. which keys are processed on the same machine

10. (1 point) **[ZooKeeper]** Using the ZooKeeper API, the following call is made:
    `create(/my_app/p, data, SEQUENTIAL)`
    What is (i) the name of the created znode and (ii) when will it be destroyed?
    A. (i) `my_app/p1`, (ii) at the end of the client's session
    B. (i) `my_app/p1`, (ii) it needs to be explicitly deleted through an API call
    C. (i) `my_app1/p`, (ii) at the end of the client's session
    D. (i) `my_app1/p`, (ii) after a failure of the client machine

11. (1 point) **[Giraph]** You are given the Giraph code shown in Figure 1. You can assume as input a directed graph (all edge weights are 1.0) which is encoded in adjacency list format (each node is encoded together with all its outgoing edges). What does this code compute?
    A. A node's inlink count.
    B. A node's outlink count.
    C. The shortest path between any two nodes in the graph.
    D. A node's PageRank score.

```java
public class BDP1 extends Vertex<
LongWritable, LongWritable, DoubleWritable, DoubleWritable> {

  @Override
  public void compute(Iterable<DoubleWritable> messages) {
    if (getSuperstep() == 0) {
      Iterable<Edge<LongWritable, DoubleWritable>> edges = getEdges();
      for (Edge<LongWritable, DoubleWritable> edge : edges) {
        sendMessage(edge.getTargetVertexId(), new DoubleWritable(1.0));
      }
    } else {
      long sum = 0;
      for (DoubleWritable message : messages) {
        sum++;
      }
      LongWritable vertexValue = getValue();
      vertexValue.set(sum);
      setValue(vertexValue);
      voteToHalt();
    }
  }
}
```

Figure 1: Giraph code extract for Question 11.

12. (1 point) **[Pig]** Assume you want to *join* two datasets within a Pig script.
    Data set 1 consists of all Wikipedia edits captured for all languages in one log file; one line contains the fields `[Unique ID, Wikipedia URL, Edit Timestamp, Editing UserID]`. The lines are unordered.
    Data set 2 consists of information about Wikipedia articles written in English (approximately 5 million articles overall): `[Unique ID, Wikipedia URL, Wikipedia Title]`. The lines are unordered.

    The join should be performed on `Wikipedia URL` and the generated data set should look as follows: `[Edit Timestamp, Wikipedia URL, Wikipedia Title]`.

    Which join is the most efficient one to use assuming a Hadoop cluster with 15 machines, each one with about 256MB of memory and 1TB of disk space?

    A. sort-merge join

    B. skew join

    C. fragment-replicate join

    D. default join

13. (1 point) **[HDFS]** Which of the following operations requires NO communication with the NameNode?

    A. A client deleting a file from HDFS.

    B. A client reading a file from HDFS.

    C. A client requesting the filename of a given block of data.

    D. All of the above require communication with the NameNode.

Free-form questions.

14. **[Streaming]** We have discussed two approaches in class to solve the following task: *Given a data stream of unknown length, randomly pick k elements from the stream so that each element has the same probability of being chosen.*

    (a) (2 points) Describe how both approaches work.

    (b) (2 points) For each approach, name one computational advantage and one disadvantage.

15. **[Streaming]** This question covers the concept of a Bloom filter.

(a) ($1\frac{1}{2}$ points) Assume a Bloom filter with k hash functions and a bit vector of size n (all bits are set to 0 initially). Further assume that every position in the bit vector can be selected by each hash function with equal probability. Compute the probability that a particular bit b_i in the bit vector is set after the insertion of x elements.

(b) (2 points) Lets create a Bloom filter W with $m = 11$, $k = 2$, $h_1(e) = e \ \textit{mod}\ 11$ and $h_2(e) = (2 \times e + 1) \ \textit{mod}\ 11$.
We now initialize W and add three elements to it. For each of the four operations defined below, write out the state of the Bloom filter <u>after</u> the conclusion of each operation.

1. ☐☐☐☐☐☐☐☐☐☐☐
initialize the filter

2. ☐☐☐☐☐☐☐☐☐☐☐
add element 14

3. ☐☐☐☐☐☐☐☐☐☐☐
add element 20

4. ☐☐☐☐☐☐☐☐☐☐☐
add element 9

(c) (1 point) Consider the final state of W (i.e. after the insertion of elements 14, 20 and 9). Lets now test whether the following three elements exist in W: 1, 15 and 20. Which of these elements are falsely identified as being in W?

16. (2 points) **[General]** Enterprise routers are the backbone of large-scale computer networks. They can forward millions of data packets per second based on the destination address of each packet. You are asked to **design an algorithm** that counts with *reasonable accuracy* and a *low memory footprint* the *unique* number of destination addresses of data packets passing through a router on a daily basis.

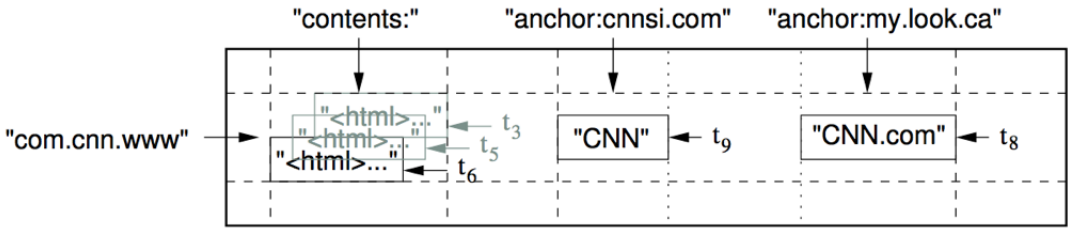
17. (2 points) **[Hadoop]** Two of the ideas behind the MapReduce paradigm can be summarized as (1) *Scale out, not up*, and, (2) *Move programs to data*.
For each idea, name one example of a particular aspect of the Hadoop implementation that encompasses it.

18. ($1\frac{1}{2}$ points) **[Hadoop]** Name three different purposes of Heartbeat messages in a Hadoop cluster.

19. **[HBase]** This question covers the HBase framework.

    (a) (1½ points) Name three differences between HBase and Hadoop.

    (b) (1 point) The figure below contains an example of a row in HBase.



    How many cells and column families (if any) are shown here?

    - Number of cells: _____
    - Name all column families: _____

20. (2 points) **[ZooKeeper]** Describe how to implement the following coordination task using the ZooKeeper API: a program (e.g. a distributed WebCrawler) is assigned a number of workers in the cluster at program startup. The workers regularly need to access a database. In order to not overwhelm the database, only two workers are allowed to access it at the same time.
    To help you with your task, listed below are the ZooKeeper API calls discussed in class:

```
* String create(path, data, flags)
* void delete(path, version)
* Stat exists(path, watch)
* (data, Stat) getData(path, watch)
* Stat setData(path, data, version)
* String[] getChildren(path, watch)
```

Write your answer to Question 20 here:

21. (2 points) **[Design Patterns]** Consider the Hadoop pseudo code shown below. It computes for each key, the *log average* across all values associated with that key. The formula for the log average is: *logAvg* $= exp(\frac{1}{n}\sum_{i=1}^{n}\log a_i)$.

```
map(string t, int r):
   EmitIntermediate(string t, int r)

reduce(string t, ints [r1, r2, r3, ..])
   float sum = 0.0;
   int count = 0;

   foreach int r in ints {
      sum += log(r);
      count++;
   }
   float logAvg = exp( 1/(float)count * sum );
   Emit(string t, float logAvg);
```

Rewrite the pseudo-code and **include a Combiner** to speed up the job. Make sure that both versions of the code (i.e. with and without the Combiner) achieve the same final result.

Write your answer to Question 21 here:

22. **[Design Patterns]** You are given two large-scale datasets containing information about users and their actions on the EdX platform - a platform where massive open online courses (MOOCs) are offered to anyone interested to join.
The first dataset *Users* contains the profiles of all users having registered on the platform. A number of example records are shown in Table 1 (ID is the primary key).

ID	FirstName	LastName	Email	Nationality	Age
4332	John	Walters	john@walters.com	US	27
4499	Klara	Sommer	klaras@yahoo.de	German	39
..	...	...	...	...	...

Table 1: Example records for dataset *Users*. The header (top row) is not included in the *Users* file.

The second dataset (*Log*) is a log of user actions recorded by the EdX platform. A number of example records are shown in Table 2. The ID uniquely identifies the user performing the action. The *Users* dataset contains less than 10 million records, the *Log* dataset contains several billion records. In extreme cases, for a single user millions of actions may have been recorded (e.g. a bot or a registered

ID	Activity	Target	Duration	Timestamp
4332	C	E4322		3432433224
4332	Q	E9800	12	3432433299
4499	C	V00233		3432435933
4332	Q	E345	78	3432430775
4499	P	P343	2390	3432434349
...	...	...	...	...

Table 2: Example rows for dataset *Log*. The header (top row) is not included in the *Log* file

Web crawler). Due to the large-scale nature of these files, the data does not reside in a database but on a Hadoop cluster; the files are stored as plain text files on HDFS in the format of one record per line.

You are asked to **join the two datasets**, based on the ID column. The final result is a single dataset which should have the format shown in Table 3.

ID	FirstName	LastName	Email	Nationality	Age	Activity	Target	Duration	Timestamp
4332	John	Walters	john@walters.com	US	27	C	E4322		3432433224
4332	John	Walters	john@walters.com	US	27	Q	E9800	12	3432433299
4499	Klara	Sommer	klaras@yahoo.de	German	39	C	V00233		3432435933
4332	John	Walters	john@walters.com	US	27	Q	E345	78	3432430775
4499	Klara	Sommer	klaras@yahoo.de	German	39	P	P343	2390	3432434349
...	...	...	...	...					

Table 3: Example output. The header (top row) should not be included in the output.

(a) ($1\frac{1}{2}$ points) Outline how you will solve this task with a single Hadoop job. Which design patterns are useful here?

(b) (3 points) Implement this task within a single Hadoop job. Write your answer down in **pseudo-code**.

23. **[Giraph]** IMDB (the Internet Movie database) has released a "who-worked-with-whom" actor graph: in this graph, every actor is a vertex and two vertices are connected via an undirected edge if the two actors appeared in a movie together. One popular actor metric is the so-called "Bacon number" $\mathfrak{B}$, i.e. the degrees of separation an actor has from the actor Kevin Bacon. The Bacon number is defined as follows:

- Kevin Bacon has $\mathfrak{B} = 0$.
- Actors that appeared with Kevin Bacon in a movie together have $\mathfrak{B} = 1$.
- If the lowest Bacon number of any actor with whom actor A appeared in a movie together is m, then A's Bacon number will be $\mathfrak{B} = m + 1$.

(a) (1 point) Argue which graph representation should be used to encode the graph.

(b) (3 points) How can the Bacon number of each vertex in this graph be computed in Giraph (or Pregel)? Explain superstep by superstep the local computation(s), messages send and votes to halt. Assume that the graph is encoded according to your chosen graph representation (previous sub-question).

(c) (2 points) Outline how you would solve this task with a Hadoop job. Describe what happens in each iteration (job) and in the map and reduce phase. You are not required to write pseudo-code.

24. **[Pig]** You have developed your own search engine and implemented a number of algorithms to bring the best search results to your users. Every time a user clicks on a search result, you log that event in the following format:

    1. search request the user submitted
    2. URL clicked by the user (a URL from the result ranking)
    3. Rank of the URL that was clicked in the result ranking
    4. algorithm used to compute the result ranking
    5. user who issued the search request and clicked a result
    6. time of the click in the format day-month-year;hours-minutes-seconds

    An example log is shown below:

```
jaguar  http://www.jaguar.com/login 1 algA user1 12-3-2009;05:03:55
jaguar  http://autoscout24.nl/jag 2 algA user1 12-3-2009;05:04:07
lottery ticket http://www.imdb.com/title/tt0979434/ 1 algB user243 12-3-2009;05:04:29
buy a car http://www.autotrader.com/index.html 7 algF user83 3-5-2009;07:25:18
.....
```

    In the example log, when `user1` issued the query “jaguar”, algorithm A was used to determine a ranking of search results and the user clicked first on the search result at rank 1 (line 1) and 12 seconds later on the search rseult shown at rank 2 (line 2). Note that you cannot assume any particular ordering of the lines in the log file.
    To investigate how popular your search engine is you need to analyze the log data.

    (a) (2 points) Write a Pig script, that determines for each user the number of *unique* search requests issued by him (as found in the log).

(b) (2 points) Write a Pig script, that determines the fraction of URLs (as found in the log), that were never ranked at rank 1 for any logged search request.

THE END.