Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Big Data for Engineers ETH Zurich
Spring Semester 2018
Lecturer(s): Ghislain Fourny Date: August 18, 2018
Assistant(s): Desislava Dimitrova, Konstantin Taranov, Zaheer Chothia, Can Berker Cikis, Alexandr Nigay

Exam

Answer Sheets

Name: _____ **Legi-Nr:** _____

Question:	A	B	C	D	E	F	G	H	I	J	K	Total
Points:	8	12	9	10	12	10	10	11	7	17	7	113
Score:												

Rules (please read carefully)

- You have 150 minutes for the exam.
- Write the answers on the stack of papers with the title "answer sheets".
- Write your name and Legi number on the cover page of the answer sheets.
- Please write your Legi number on all other pages of the answer sheets.
- Use blue or black ink, DO NOT USE red ink. DO NOT USE pencils.
- Write as clearly as possible and cross out everything that you do not consider to be part of your solution. Answers can be given in either English or German.
- Hand in your answer sheet. Do not hand in the sheets with the questions. You may keep the question sheets.

Remarks

- Unless noted otherwise, the multiple choice questions only have a single correct answer. If you cross more than one box of a question, you will get zero points for that question.
- There are no negative points.
- In a multiple choice question, if you picked an answer and then would like to change it, **make it very clear**. For example, you could cross out the old answer, tick the new answer, and draw a circle around your new answer.

Signature: _____

A Block storage

A.1: ◯ A ◯ B ◯ C ◯ D
A.2: ◯ A ◯ B ◯ C ◯ D
A.3: ◯ A ◯ B ◯ C ◯ D
A.4: ◯ A ◯ B ◯ C ◯ D
A.5: ◯ A ◯ B ◯ C ◯ D
A.6: ◯ A ◯ B ◯ C ◯ D
A.7: ◯ A ◯ B ◯ C ◯ D
A.8: ◯ A ◯ B ◯ C ◯ D

B Object storage

B.1: ◯ A ◯ B ◯ C ◯ D ◯ E ◯ F
B.2: ◯ A ◯ B ◯ C ◯ D ◯ E ◯ F
B.3: ◯ A ◯ B ◯ C ◯ D ◯ E
B.4: ◯ A ◯ B ◯ C ◯ D ◯ E

Question B.5:

B.6: ◯ A ◯ B ◯ C ◯ D
B.7.a: ◯ true ◯ false
B.7.b: ◯ true ◯ false
B.7.c: ◯ true ◯ false
B.7.d: ◯ true ◯ false
B.8.a: ◯ true ◯ false
B.8.b: ◯ true ◯ false
B.8.c: ◯ true ◯ false
B.8.d: ◯ true ◯ false

C HBase

C.1: ◯ A ◯ B ◯ C ◯ D
C.2: ◯ A ◯ B ◯ C ◯ D
C.3: ◯ A ◯ B ◯ C ◯ D
C.4: ◯ A ◯ B ◯ C ◯ D

Question C.5:

Query: `get 'phrases', '570'`			
Row key	**Column**	**Time-stamp**	**Value**

Query: `get 'phrases', '723'`			
Row key	**Column**	**Time-stamp**	**Value**

D Apache Spark

D.1.a: ◯ true ◯ false
D.1.b: ◯ true ◯ false
D.1.c: ◯ true ◯ false
D.1.d: ◯ true ◯ false
D.2.a: ◯ at most one ◯ may be many ◯ none
D.2.b: ◯ at most one ◯ may be many ◯ none
D.2.c: ◯ at most one ◯ may be many ◯ none
D.2.d: ◯ at most one ◯ may be many ◯ none
D.3.a: ◯ true ◯ false
D.3.b: ◯ true ◯ false
D.3.c: ◯ true ◯ false
D.3.d: ◯ true ◯ false
D.3.e: ◯ true ◯ false
D.3.f: ◯ true ◯ false

Question D.4:

E Document stores

E.1.a: ◯ true ◯ false
E.1.b: ◯ true ◯ false
E.1.c: ◯ true ◯ false
E.1.d: ◯ true ◯ false
E.1.e: ◯ true ◯ false
E.1.f: ◯ true ◯ false
E.2.a: ◯ true ◯ false
E.2.b: ◯ true ◯ false
E.2.c: ◯ true ◯ false
E.2.d: ◯ true ◯ false
E.3.a: ◯ yes ◯ no
E.3.b: ◯ yes ◯ no
E.3.c: ◯ yes ◯ no
E.3.d: ◯ yes ◯ no
E.3.e: ◯ yes ◯ no
E.3.f: ◯ yes ◯ no

Question E.4:

F Data models

F.1: ◯ A ◯ B ◯ C ◯ D

Question F.2:

F.3: ◯ A ◯ B ◯ C ◯ D
F.4: ◯ A ◯ B ◯ C ◯ D
F.5: ◯ A ◯ B ◯ C ◯ D
F.6: ◯ A ◯ B ◯ C ◯ D
F.7.a: ◯ valid ◯ not valid
F.7.b: ◯ valid ◯ not valid
F.7.c: ◯ valid ◯ not valid
F.7.d: ◯ valid ◯ not valid

G Querying

G.1: ◯ A ◯ B ◯ C ◯ D
G.2: ◯ A ◯ B ◯ C ◯ D
G.3: ◯ A ◯ B ◯ C ◯ D
G.4: ◯ A ◯ B ◯ C ◯ D
G.5: ◯ A ◯ B ◯ C ◯ D
G.6: ◯ A ◯ B ◯ C ◯ D
G.7: ◯ A ◯ B ◯ C ◯ D
G.8: ◯ A ◯ B ◯ C ◯ D
G.9: ◯ A ◯ B ◯ C ◯ D
G.10: ◯ A ◯ B ◯ C ◯ D

H Syntax

H.1: ◯ A ◯ B ◯ C ◯ D
H.2: ◯ A ◯ B ◯ C ◯ D
H.3: ◯ A ◯ B ◯ C ◯ D
H.4: ◯ A ◯ B ◯ C ◯ D
H.5: ◯ A ◯ B ◯ C ◯ D

Question H.6:

a.	
b.	
c.	
d.	
e.	
f.	

I Relational databases

I.1: ◯ A ◯ B ◯ C ◯ D
I.2: ◯ A ◯ B ◯ C ◯ D
I.3: ◯ A ◯ B ◯ C ◯ D
I.4: ◯ A ◯ B ◯ C ◯ D
I.5: ◯ A ◯ B ◯ C ◯ D
I.6: ◯ A ◯ B ◯ C ◯ D
I.7: ◯ A ◯ B ◯ C ◯ D

J MapReduce

J.1: ◯ A ◯ B ◯ C ◯ D
J.2.a: ◯ true ◯ false
J.2.b: ◯ true ◯ false
J.2.c: ◯ true ◯ false
J.2.d: ◯ true ◯ false
J.2.e: ◯ true ◯ false
J.2.f: ◯ true ◯ false
J.2.g: ◯ true ◯ false
J.2.h: ◯ true ◯ false
J.2.i: ◯ true ◯ false
J.2.j: ◯ true ◯ false
J.2.k: ◯ true ◯ false
J.2.l: ◯ true ◯ false
J.2.m: ◯ true ◯ false
J.2.n: ◯ true ◯ false

Question J.3:

Question J.4:

Question J.5:

```
function combine(key, values[])
% write your code here
```

```
function reduce(key, values[])
% write your code here
```

K General questions

Question K.1:

Question K.2:

K.3: ◯ A ◯ B ◯ C ◯ D
K.4: ◯ A ◯ B ◯ C ◯ D
K.5: ◯ A ◯ B ◯ C ◯ D
K.6: ◯ A ◯ B ◯ C ◯ D
K.7: ◯ A ◯ B ◯ C ◯ D

Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Big Data for Engineers ETH Zurich
Spring Semester 2018
Lecturer(s): Ghislain Fourny Date: August 18, 2018
Assistant(s): Desislava Dimitrova, Konstantin Taranov, Zaheer Chothia, Can Berker Cikis, Alexandr Nigay

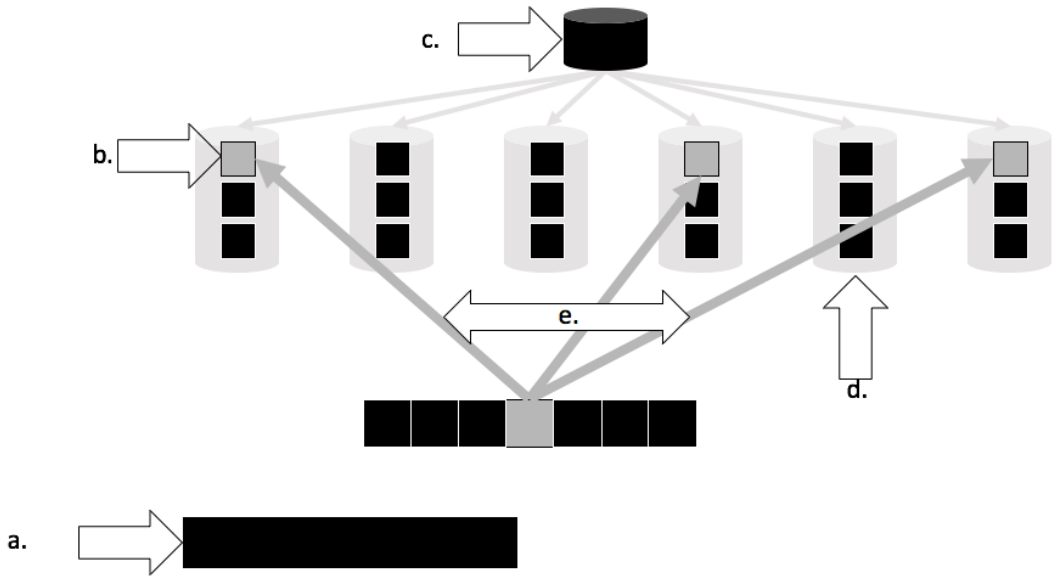
Exam

Exam Questions

Reminders

- Write all your answers on this stack of papers ("answer sheets").
- Write your name and Legi number on the cover page of the answer sheets.
- Put your signature on the cover page.
- Please write your Legi number on all other answer sheets.
- Do not separate the individual sheets of this stack. If there are pages missing, you will not receive any points for these pages.
- Hand in this stack of answer sheets.

A Block storage (8 points)



The above drawing sketches the architecture of HDFS, but with missing legend (a., b., c., d., e.).

(1) (1 point) What does `a.` correspond to?
- A. Block
- B. DataNode
- C. File
- D. NameNode

(2) (1 point) What does `b.` correspond to?
- A. Block
- B. DataNode
- C. File
- D. NameNode

(3) (1 point) What does `c.` correspond to?
- A. Block
- B. DataNode
- C. File
- D. NameNode

(4) (1 point) What does `d.` correspond to?
- A. Block
- B. DataNode
- C. File
- D. NameNode

(5) (1 point) What does `e.` correspond to (in HDFS terminology)?

A. Replicas
B. Shards
C. Splits
D. Partitions

(6) (1 point) What is the typical block size for HDFS, in the classical setting (typical network latency, typical network throughput) encountered in practice in clusters, and that we covered in the lecture?

A. 64 kB
B. 128 MB
C. 12 TB
D. There is no such typical block size. The block size does not play any major role in terms of performance in a distributed file system.

We are now going to look into alternate scenarios with a different environment in terms of network latency (time to wait until the data starts arriving) and network throughput (speed at which the data is transmitted). These scenarios are, for the purpose of this exam, imaginary, but only the future can tell how the Internet will look like in 20 years.

Note: In the following scenarios, we are interested only in the block size and the transfer of blocks over the network. We completely ignore here issues related to the management of block metadata, in the sense that you can assume that the NameNode gets scaled out seamlessly.

(7) (1 point) Prof. S., a famous networks professor in the ETH Computer Science department, has found a way to dramatically **reduce the latency of the Internet down to almost zero**, thanks to tachyons and top-secret satellites orbiting the Earth. However, **download throughput stays identical** to what we are used to in our clusters.

Prof. S. asked his PhD students to implement and adapt HDFS to his infrastructure, which they call TDFS. The PhD students are arguing about the size that a block should have. What is the *most likely* size that they will choose among the following choices, for this setting with very low latency and normal network throughput?

A. 64 kB
B. 128 MB
C. 12 TB
D. The block size does not play any major role in terms of performance in a distributed file system.

(8) (1 point) Prof. P. (Hogwarts University), a competitor of Prof. S., intends to attach newly-developed physical hard drives (1PB capacity with almost instant read/write) to pigeons and *literally* have the pigeons fly over his cluster *instead of the network cables* to transfer data between nodes: data is instantly copied from a source node onto a disk, a pigeon carries the disk to another node, the data is instantly copied to the destination node. He can thus achieve **much higher network throughputs**. Of course, he is aware that the **latency will be extremely high** and may involve several minutes.

Prof. P asked his PhD students to implement and adapt HDFS to his infrastructure, called PDFS. The PhD students are arguing about the size that a block should have. What is the *most likely* size that they will choose among the following choices, for this setting with high latency and very high network throughput?

A. 64 kB

B. 128 MB

C. 12 TB

D. The block size does not play any major role in terms of performance in a distributed file system.

B Object storage (12 points)

(1) (1 point) Order the following devices by their typical storage capacity (from smallest to largest):

1. HDD
2. SSD
3. DRAM

A. 1,2,3
B. 1,3,2
C. 2,3,1
D. 2,1,3
E. 3,2,1
F. 3,1,2

(2) (1 point) Order the following devices by their typical read bandwidth (from lowest to highest):

1. HDD
2. SSD
3. DRAM

A. 1,2,3
B. 1,3,2
C. 2,3,1
D. 2,1,3
E. 3,2,1
F. 3,1,2

(3) (1 point) What is the maximum read throughput in requests/second the object storage can achieve if its maximum read bandwidth is 600MB/s, and it stores objects ranging from 0.5MB to 20MB in their size, and its total capacity is 6000MB.

A. 300 req/sec
B. 10 req/sec
C. 1200 req/sec
D. 30 req/sec
E. 0.1 req/sec

(4) (1 point) A client sends the request to an object store (OS) to check whether an object exists there. The client gets the reply in 320ms that the object is present in the OS. Assuming that it takes 20 ms to process such request, how many milliseconds have passed between the client sending the request and the OS receiving it?

A. 320 ms
B. 300 ms
C. 170 ms
D. 150 ms
E. 130 ms

(5) (3 points) What do letters C,A,P stand for in the CAP theorem? What is the main statement of the CAP theorem?

(6) (1 point) What does the annual availability of 99.9% mean for an object storage (OS)?

A. There is 0.999 probability that a requested object is lost
B. Within a year the OS loses up to 0.001% of all objects
C. There is 0.001 likelihood that a requested object is not accessible
D. At most 0.001% of all objects can be inaccessible at the same time

(7) (2 points) Which of the following statements about object stores (OSs) are correct?

a. Probability of OS failure increases with the number of replicas
b. The write latency of OS decreases with the number of replicas
c. Reliability of OS increases with the number of replicas
d. Geo-replicated OS is more reliable than a locally-replicated one

(8) (2 points) Which of the following statements about Azure object storage are correct?

a. The object storage can store only JSON objects
b. Every object by default is replicated 3 times (2 additional copies)
c. The Azure object storage is not fault-tolerant
d. The Azure object storage cannot store objects larger than 8GB

C HBase (9 points)

(1) (1 point) What do HBase and relational databases have in common?

A. HBase is built out of many individual relational databases.
B. They are completely different.
C. They both use the same data shape—table.
D. They both support *joins*.

(2) (1 point) Why does HBase belong to "wide column stores"?

A. Because every several consecutive rows of data in HBase are actually stored as a single column.
B. Because each column in HBase can store very wide values.
C. Because column families need to be known in advance.
D. Because it stores data from the same column family together.

(3) (1 point) For which of the following use cases is HBase the most adequate tool compared to other Big-Data technologies?

A. Storing a map of all cities in the world with all roads between them, if this map is used in a navigation software.
B. Storing the catalog of all the stars in the Universe that we know about, which is used for statistical calculations based on subsets of their properties.
C. Storing grades for a class of one million students.
D. Implementing a large bank database, in which customer data, account data, and transaction data have to be kept in separate places due to legal reasons.

(4) (1 point) HBase requires *column families* to be specified in advance. Why does it not require *columns* to be specified in advance?

A. Because the concept of "columns" is not used by HBase—it only works with "column families".
B. Because the responsibility of keeping track of columns is delegated to the underlying HDFS.
C. Because HBase physically partitions data according to column families, not columns.
D. Because the column name ("qualifier") is not stored on disk at all—it is only saved in main memory.

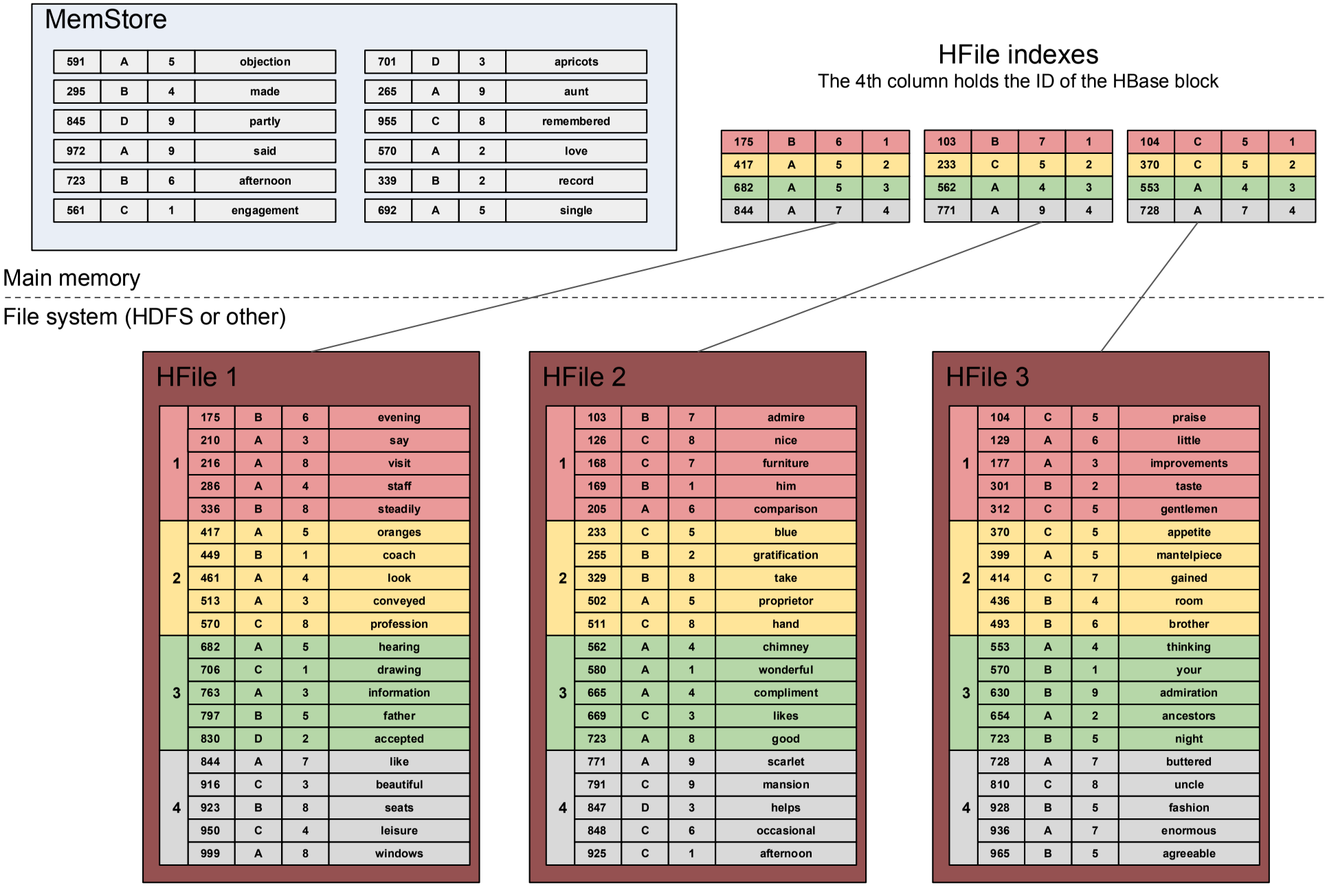
(5) (5 points) This task is similar to what we had in the exercise sheet. On the page after this introductory text you will see a diagram which presents a simplified representation of the internal state of an HBase RegionServer. You will be given a couple of queries, and your task will be to state what they return.

In this simplified representation, each HBase key-value pair consists of the following fields, in order: `row key, column qualifier, timestamp, value`. Column family here is omitted, because we assume that there is only one column family in the table. This HBase table is called `phrases`.

What you need to do: state what each of the following queries will return.

- `get 'phrases', '570'`
- `get 'phrases', '723'`

The queries are written in HBase shell syntax: `get <table name>, <row key>`. In provided spaces on the answer sheet, write down the data that will be returned by each query. Use as many lines as you need. Assume that the system is configured to return only the latest version of each cell.


MemStore
591 A 5 objection
295 B 4 made
845 D 9 partly
972 A 9 said
723 B 6 afternoon
561 C 1 engagement
701 D 3 apricots
265 A 9 aunt
955 C 8 remembered
570 A 2 love
339 B 2 record
692 A 5 single
HFile indexes
The 4th column holds the ID of the HBase block
175 B 6 1
417 A 5 2
682 A 5 3
844 A 7 4
103 B 7 1
233 C 5 2
562 A 4 3
771 A 9 4
104 C 5 1
370 C 5 2
553 A 4 3
728 A 7 4
Main memory
File system (HDFS or other)
HFile 1
1 175 B 6 evening
210 A 3 say
216 A 8 visit
286 A 4 staff
336 B 8 steadily
2 417 A 5 oranges
449 B 1 coach
461 A 4 look
513 A 3 conveyed
570 C 8 profession
3 682 A 5 hearing
706 C 1 drawing
763 A 3 information
797 B 5 father
830 D 2 accepted
4 844 A 7 like
916 C 3 beautiful
923 B 8 seats
950 C 4 leisure
999 A 8 windows
HFile 2
1 103 B 7 admire
126 C 8 nice
168 C 7 furniture
169 B 1 him
205 A 6 comparison
2 233 C 5 blue
255 B 2 gratification
329 B 8 take
502 A 5 proprietor
511 C 8 hand
3 562 A 4 chimney
580 A 1 wonderful
665 A 4 compliment
669 C 3 likes
723 A 8 good
4 771 A 9 scarlet
791 C 9 mansion
847 D 3 helps
848 C 6 occasional
925 C 1 afternoon
HFile 3
1 104 C 5 praise
129 A 6 little
177 A 3 improvements
301 B 2 taste
312 C 5 gentlemen
2 370 C 5 appetite
399 A 5 mantelpiece
414 C 7 gained
436 B 4 room
493 B 6 brother
3 553 A 4 thinking
570 B 1 your
630 B 9 admiration
654 A 2 ancestors
723 B 5 night
4 728 A 7 buttered
810 C 8 uncle
928 B 5 fashion
936 A 7 enormous
965 B 5 agreeable

D Apache Spark (10 points)

(1) (2 points) Which of the following statements about Spark are correct?

a. Spark starts actual processing when an action is called

b. Spark maintains a log of all updates to each RDD to recover the lost data by applying each update from the log

c. RDDs in Spark are always represented as key-value pairs

d. Transformations always return RDDs

(2) (2 points) Answer the following questions.

a. How many tasks are there in a stage? A. at most one B. may be many C. none

b. How many actions are there in a job? A. at most one B. may be many C. none

c. How many transformations are there in a job? A. at most one B. may be many C. none

d. How many stages are there in a job? A. at most one B. may be many C. none

(3) (3 points) Which of the following statements about Spark are correct?

a. Actions are functions that return another RDD

b. The `reduceByKey` transformation always induces a shuffle

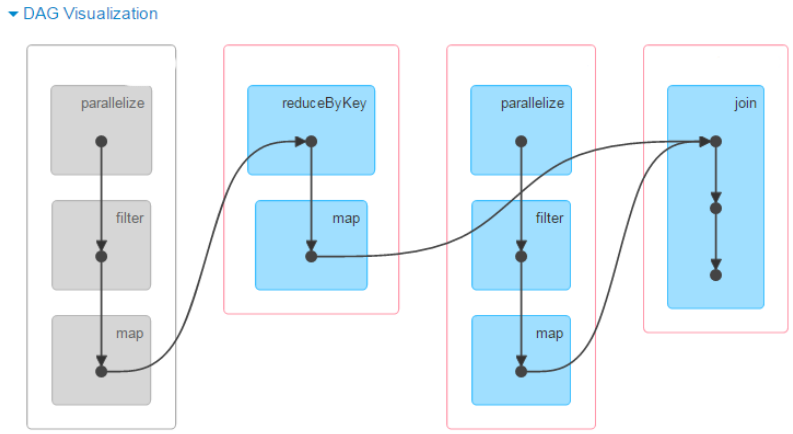
c. The `filter` transformation can be always computed without a shuffle

d. Shuffle time increases with the number of machines an RDD is partitioned on

e. It is impossible to choose the number of partitions of an RDD and it is determined automatically by Spark

f. In Spark after each transformation an action must be called

(4) (3 points) How many stages does the DAG from the Spark UI below have? How many stages have been actually computed? What are possible reasons for an RDD to be reused?



E Document stores (12 points)

(1) (3 points) Which of the following statements about Document stores and MongoDB are correct?

a. Document stores support key-value-like queries
b. MongoDB supports schema validation
c. MongoDB can serialize documents in the JSON-like format
d. MongoDB can store different documents in the same collection regardless of their content
e. In MongoDB all documents have the `_id` field
f. Document stores and object stores are completely the same

(2) (2 points) Consider the following requests to a newly created collection `company`:

```
1 db.company.insert({ "name": ["David"], "job":"Engineer" })
2 db.company.insert({ "room_number": 305, "type":"office" })
3 db.company.index({"salary" : 1})
```

Which of the following statements are correct?

a. MongoDB will complain about the request 1 as `"name"` cannot be an array
b. MongoDB will complain about the request 2, since 305 is not quoted
c. MongoDB will complain about the request 2, because the intended document for insertion has different schema from the document inserted with the request 1
d. MongoDB will complain about the request 3, since there is no documents with the field `"salary"` in the collection yet

(3) (3 points) Consider a collection of JSON documents of the following form:

```
{
  _id: "12345",
  publisher: "Allen and Unwin",
  books: [
              {
                title: "The Lord of the Rings",
                author: "John Ronald Reuel Tolkien"
              },
              {
                title: "The Hobbit",
                author: "John Ronald Reuel Tolkien"
              }
          ]
}
```

Which queries will use the following index: `.createIndex({ books.author : 1 })`

a. `find({books.title:"Boston"})`	A. yes	B. no
b. `find({books.author:"Dan Brown"})`	A. yes	B. no
c. `find({books.author:"Dan Brown"},{books.title:1})`	A. yes	B. no
d. `find({books.title:"The Hobbit"},{books.author:1})`	A. yes	B. no
e. `find({books.language:"English"})`	A. yes	B. no
f. `find({publisher:"Doubleday"})`	A. yes	B. no

(4) (4 points) Consider the following collection of JSON documents called `books`:

```
{
  _id: "1",
  title: "The Lord of the Rings",
  quantity : 5,
  genre : [ "fantasy ", "novel"]
}
```

```
{
  _id: "2",
  name: "The Hobbit",
  quantity : 1,
  genre : [ "fantasy ", "novel"]
}
```

```
{
  _id: "3",
  name: "Le Petit Prince",
  quantity : 4,
  genre : [ "novel"]
}
```

```
{
  _id: "4",
  name: "The Da Vinci Code",
  quantity : 19,
  genre : [ " mystery", "thriller "]
}
```

What does the following MongoDB query output? (Write what it would print in a console after execution.) *Hint: To get all points, the answer must exactly match the output of MongoDB.*

```
books.find({ "genre" : "fantasy" }, {"quantity" : 1} ).sort( { "quantity" : 1 } )
```

F Data models (10 points)

(1) (1 point) What is the purpose of XML Information Set?

A. It provides a tree-like data model for XML.

B. It outlines the well-formedness requirements for XML documents.

C. It extends XML with datatype `set`.

D. It describes an XML-based tool for drawing trees (in Computer Science sense).

(2) (1 point) Draw (in the space provided on the answer sheet) an XML Infoset tree for the following XML document. Show only the following information items: document, element, attribute, character (not every character separately but the whole string). There is no need to use colors.

```
<Controller>
  <Controls>
    <X-axis>34.7</X-axis>
    <Y-axis>-17.3</Y-axis>
    <Z-axis disabled="true"/>
  </Controls>
</Controller>
```

(3) (1 point) Under which conditions may an XML document be "valid against a certain schema" *and* "not well-formed"?

A. Only if the XML document uses no namespaces.

B. This is impossible.

C. Only if the schema is empty.

D. This is always the case, because one requires the other.

(4) (1 point) Why are the concepts of *XML well-formedness* and *XML validation* separate?

A. That is due to historic reasons—there is no practical reason for that.

B. That is because they refer to different qualities of an XML document.

C. Those terms are actually synonyms, so the concepts are not separate.

D. That is because those two concepts cannot be applied to the same document at the same time.

(5) (1 point) Which of the following *cannot* be expressed with XML Schema?

A. Requirements on the structure of an XML document.

B. List of permitted element names.

C. List of permitted attributes for each element.

D. Requirements on how the document should be read by software.

(6) (1 point) Imagine that we are using an XML Schema to check whether all elements called `<age>` in a certain XML document contain only integer values (because they are supposed to represent a person's age). Can we perform the same check but without using an XML Schema?

A. Yes, but only if the document is small.

B. Yes, we can do that in any case.

C. No, because that kind of check can only be done with an XML Schema.

D. No, because the base XML specification does not include datatypes.

(7) (4 points) Consider the following XML Schema instance. For each of the XML documents given below, state whether it will successfully be validated against this schema.

```
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="Book">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="Title" type="xs:string" minOccurs="1"/>
        <xs:element name="Author" type="xs:string" minOccurs="1"/>
        <xs:element name="Published" type="xs:string" minOccurs="0"/>
        <xs:element name="Edition" type="xs:integer" minOccurs="0"/>
        <xs:element name="FavouriteLine" type="xs:string" minOccurs="0"
                                          maxOccurs="unbounded"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
</xs:schema>
```

a.
```
<Book>
  <Title>Brave New World</Title>
  <Author>Aldous Huxley</Author>
  <Published>1932</Published>
  <Edition>1st</Edition>
  <FavouriteLine>But I don't want comfort.
  </FavouriteLine>
</Book>
```
A. valid B. not valid

b.
```
<Book>
  <Title>My new book</Title>
  <Author>Me, Myself, and I</Author>
  <Published>not yet</Published>
  <FavouriteLine>All of it!</FavouriteLine>
</Book>
```
A. valid B. not valid

c.
```
<Book>
  <Title>Point Counter Point</Title>
  <Author>Aldous Huxley</Author>
  <Published>1928</Published>
  <FavouriteLine> ... from a distance and
    theoretically he admired.
  </FavouriteLine>
</Book>
```
A. valid B. not valid

d.
```
<Book>
  <Title>1984</Title>
  <Author>George Orwell</Author>
  <Edition>1</Edition>
  <Published>1949</Published>
  <FavouriteLine>War is peace.</FavouriteLine>
</Book>
```
A. valid B. not valid

G Querying (10 points)

(1) (1 point) What is the purpose of XPath?

A. Checking if an XML document is located on a certain file-system path.

B. Checking if a JSON document is located on a certain file-system path

C. Presenting an XML document in a textual format which uses the "path" syntax.

D. Navigating within an XML document.

(2) (1 point) How can XQuery make use of XPath?

A. FLWOR expressions of XQuery can contain XPath expressions.

B. These are two names of the same technology.

C. By definition, XPath expressions always contain a FLWOR expression from XQuery.

D. XQuery must use XPath if no XML Schema is associated with the target XML document.

(3) (1 point) What prevents us from using XQuery with an XML document which is stored in HDFS?

A. XQuery can only work with XML documents permanently stored on the local disk.

B. XQuery was released earlier than HDFS, and because of this, the technologies are incompatible.

C. HDFS can only work with MapReduce, and XQuery cannot work with MapReduce.

D. Nothing.

(The exam continues on the next page)

The rest of the questions in this section will be about specific queries and will use the following XML document. Assume that the document is saved in a file named `books.xml`. The document is well-formed.

```
<bookstore>
    <book category="CHILDREN">
        <title>If Animals Kissed Good Night</title>
        <author>Ann Whitford Paul</author>
        <author>David Walker</author>
        <year>2014</year>
        <price>5.86</price>
    </book>
    <book category="CHILDREN">
        <title>Dragons Love Tacos</title>
        <author>Adam Rubin</author>
        <author>Daniel Salmieri</author>
        <year>2012</year>
        <price>10.79</price>
    </book>
    <book category="COOKING">
        <title>Cooking Under Pressure</title>
        <author>Lorna J. Sass</author>
        <year>2009</year>
        <price>16.31</price>
    </book>
    <book category="COOKING">
        <title>Culinary Reactions</title>
        <author>Simon Quellen Field</author>
        <year>2011</year>
        <price>11.74</price>
    </book>
</bookstore>
```

Hint: The atomization of an element with complex or mixed content is defined as the content of all descendant text nodes.

(The questions start on the next page)

For each of the following XPath or XQuery expressions, indicate what it will return when run on the document presented on the previous page.

(4) (1 point) `doc("books.xml")/bookstore/book/year`

A.
```
<year>2014</year>
<year>2012</year>
<year>2009</year>
<year>2011</year>
```

B.
```
<year>2014</year>
```

C.
```
2014
2012
2009
2011
```

D.
```
2014
```

(5) (1 point) `doc("books.xml")/bookstore/book[@category="CHILDREN"]/author`

A.
```
Ann Whitford Paul
Adam Rubin
```

B.
```
<author>Ann Whitford Paul</author>
<author>Adam Rubin</author>
```

C.
```
Ann Whitford Paul
David Walker
Adam Rubin
Daniel Salmieri
```

D.
```
<author>Ann Whitford Paul</author>
<author>David Walker</author>
<author>Adam Rubin</author>
<author>Daniel Salmieri</author>
```

(6) (1 point) `doc("books.xml")/bookstore/book[@year="2014"]/price`

A.
```
<price>5.86</price>
```

B.
```
5.86
```

C.
```
Syntax error
```

D.
```
Empty sequence
```

(7) (1 point)

```
for $x in doc("books.xml")/descendant::price
return $x
```

A.
```
<price>5.86</price>
<price>10.79</price>
<price>16.31</price>
<price>11.74</price>
```

B.
```
5.86
10.79
16.31
11.74
```

C.
```
Error
```

D.
```
Empty sequence
```

(8) (1 point)

```
for $x in doc("books.xml")/bookstore/book
let $y := $x/price
return $y + 1
```

A.
```
<price>6.86</price>
<price>11.79</price>
<price>17.31</price>
<price>12.74</price>
```

B.
```
6.86
11.79
17.31
12.74
```

C.
```
Error
```

D.
```
<price>5.86</price> + 1
<price>10.79</price> + 1
<price>16.31</price> + 1
<price>11.74</price + 1
```

(9) (1 point)

```
for $x in doc("books.xml")/bookstore/book
where $x = "Dragons Love Tacos"
return $x/year
```

A.
```
<year>2012</year>
```

B.
```
2012
```

C.
```
Error
```

D.
```
Empty sequence
```

(10) (1 point) Assume you are evaluating the given XQuery expression. Which of the following XPath expressions return the same result? Mark all applicable choices.

XQuery:
```
for $x in /bookstore/book[last()][@category='COOKING']
return $x/title
```

XPath:

A.
```
//book[position()>=3]/title
```

B.
```
/bookstore/book[4][@category='COOKING']/title
```

C.
```
/bookstore/book[@category='COOKING'][4]/title
```

D.
```
fn:reverse(/child::*/book)[position()<2]/descendant::*[1]
```

Additional hints:

1. *Sequences are indexed starting from one.*
2. *`fn:reverse` returns a sequence containing the items in reverse order. For example, `reverse(("ab", "cd", "ef"))` evaluates to `("ef", "cd", "ab")`.*

H Syntax (11 points)

(1) (1 point) What is the purpose of checking an XML document for well-formedness?

A. To make sure the document adheres to the rules defined by the user.

B. To make sure it can be written to a SQL database.

C. To make sure an XML language can be built from the document.

D. To make sure it adheres to the rules of XML language.

(2) (1 point) Among the following statements, only one is wrong. Which one?

A. XML elements cannot have multiple attributes with the same name.

B. XML comments can be put anywhere in the document.

C. An XML element can contain other elements and text.

D. An XML element is permitted to contain nothing between its opening and closing tags.

(3) (1 point) For which of the following use cases is XML a significantly better tool than JSON?

A. Designing a format for storing names and addresses of one million customers.

B. Storing a catalog of all the stars in the Universe that we know about (a document per star), containing a set of properties for each star.

C. Storing geographical information about all countries in the world.

D. Designing a format for describing a book in preparation for publishing.

(4) (1 point) How do XML and JSON fit into the topic of Big Data? Choose the option which fits best.

A. The transition from tabular data to tree-based data, which can be represented in XML and JSON, is central to all Big-Data technologies.

B. An entire class of databases is centered around the idea of storing data in XML or JSON or in a similar format.

C. All Big-Data databases can use XML well-formedness rules to check integrity of data.

D. XML and JSON do not fit the topic of Big Data directly and were given as a side topic instead.

(5) (1 point) What is the purpose of XML entities (`<`, `>`, `&apos;`, `&quot;`, `&`)?

A. The characters that these entities represent are prohibited to appear anywhere in an XML document and can only be used via these entities.

B. These entities must be used when characters they represent cannot be placed in a certain place of an XML document.

C. These entities make it possible to exchange XML documents between different computers which may be using incompatible character encodings.

D. These entities are remains of the language on which XML is based, and they do not carry any practical value.

(6) (6 points) For each of the following XML documents, determine whether it is well-formed or not. If the document is well-formed, write "*good*" in the corresponding field on the answer sheet. If the document is *not* well-formed, show where in the document the problem is, by writing the corresponding *line number(s)*.

a.
```
1 <Mountain>
2   <name>Piz Daint</name>
3   <height unit="metres">2968</height>
4 </mountain>
```

b.
```
1 <food name="bread">
2   <contents>
3     <flour type="wheat"/>
4   </contents>
5 </food>
```

c.
```
1 <BookFrontMatter>
2   <Title>Pride and Prejudice</Title>
3   <Author>Jane Austen</Author>
4   <Published>1813</Published>
5 </BookFrontMatter>
6 It is a truth universally acknowledged,
7 that a single man in possession of a good
8 fortune, must be in want of a wife...
```

d.
```
1 <CarInspection>
2   <Car headlights="functional">
3     <Interior>
4       <DriverSeat airbag="functional"/>
5       <PassengerSeat airbag="functional"/>
6     </Interior>
7   </Car taillights="functional">
8 </CarInspection>
```

e.
```
1 <History>
2   <Period what="Barack Obama's presidency" when="2009--2017">
3     <Period what="Theresa May as UK Prime Minister" when="2016--current">
4   </Period>
5     </Period>
6 </History>
```

f.
```
1 <z/>
```

I Relational databases (7 points)

(1) (1 point) Which of the following does not characterize relational databases from the 1970s?

A. The system runs on a single machine
B. Data has to be arborescent (shaped like trees).
C. Data always has a schema
D. Rows are homogeneous

(2) (1 point) Which one of the following relational algebra operators is considered the most expensive?

A. selection
B. projection
C. grouping
D. join

(3) (1 point) Which one of the following most adequately characterizes a table, when looking at it from a NoSQL perspective?

A. a collection of rows
B. a collection of items that can be atomic or structured
C. a collection of columns
D. a collection of atomic values

(4) (1 point) Which query performs a projection?

A. `SELECT * FROM students`
B. `SELECT name, birthday FROM students`
C. `SELECT * FROM students WHERE year = 2`
D. All of them

(5) (1 point) Which query performs a selection?

A. `SELECT * FROM students`
B. `SELECT name, birthday FROM students`
C. `SELECT * FROM students WHERE year = 2`
D. All of them

(6) (1 point) Which of the following characterizes the first normal form?

A. Values (intersection of a row with a column) must be atomic.
B. Values must be trees with a depth of at least 2.
C. Values may not be NULL.
D. The data must be valid against some schema.

(7) (1 point) Tables do not make any sense any more when scaling out on a cluster. True or false?

A. True.
B. False.
C. True, but only with a schema that strictly constrains columns.
D. False, except for a document store with denormalized data.

J MapReduce (17 points)

(1) (1 point) The map function takes inputs with the type `<k1, v1>`. Under what signature are the mapper's outputs produced?

A. `<k2, list(v2)>`
B. `list(<k2, v2>)`
C. `<list(k2), v2>`
D. `<k1, v1>`

(2) (7 points) Which of the following statements about MapReduce are correct?

a. MapReduce is designed to work only on specialized clusters.
b. MapReduce is designed to tolerate hardware failures.
c. Two different mappers might process pairs with the same key.
d. Two different reducers might process pairs with the same key.
e. The mapper in Hadoop can only produce key-value pairs as outputs.
f. The reducer in Hadoop can only produce key-value pairs as outputs.
g. In Hadoop the number of reducers can be chosen automatically.
h. In Hadoop the number of mappers can be chosen automatically.
i. In MapReduce any reduce function can be used as a combiner and provide the identical result.
j. The combiner function is executed by a reducer.
k. The intention of using a combiner function is to reduce communication.
l. A reducer starts processing only when all map tasks are finished.
m. Each data split is processed only by one mapper.
n. All reducers process the same number of key-value pairs.

(3) (2 points) Imagine you wrote a MapReduce job which was executed over 4 reducer nodes. After the execution you checked the job report and you noticed that only 1 out of 4 reducers worked, and other 3 were idle. Write possible reasons for such problem. What should you do to balance the job between reducers?

(4) (2 points) Consider the following data splits, and map and reduce functions,

Split 1	
Key	Value
21	123
4	456

Split 2	
Key	Value
4	734
13	623

```
function map(key, value)
   emit(key, value)

function reduce(key, values[])
  for value in values
      emit(key, value)
```

How many parts will the output file have, if the code was executed with two mappers and one reducer? Write down the content of the output file.

(5) (5 points) Consider the following map and reduce functions, which calculate sample variance of each key.

```
function map(key, value)
   emit(key, value)


function reduce(key, values[])
   n, sum_v, sum_sq = 0
   for value in values:
      sum_v += value
      sum_sq += value*value
      n += 1
   tmp = sum_sq / (n-1) - sum_v^2 / (n*(n-1))
   emit(key, tmp)
```

According to the code, most of the job is done by reducers, as a result the program above leads to a low degree of parallelization. Write a new combine and reduce functions to alleviate the problem.

Hint: you can emit multiple values with the same key using the notation `emit(key, (val1, val2, val3))` *and then retrieve individual values with* `value[0]`.

K General questions (7 points)

(1) (1 point) The prefix “peta” means how many zeros?

(2) (1 point) The prefix “exa” means how many zeros?

(3) (1 point) Which is the biggest?

- A. a petabyte
- B. an exabyte
- C. a zettabyte
- D. a yottabyte

(4) (1 point) Which one of the following bottlenecks motivates switching to a cluster of machines the *most* in practice for data-processing applications (Spark, MapReduce)?

- A. Bottleneck in CPU usage due to inefficient code
- B. Bottleneck in throughput of reading from/writing to disk
- C. Bottleneck in random memory access
- D. Bottleneck in network I/O

(5) (1 point) Which one of the following is a storage layer technology?

- A. MapReduce
- B. Spark
- C. HDFS
- D. YARN

(6) (1 point) Which one of the following is *not* syntax?

- A. YARN
- B. CSV
- C. JSON
- D. XML

(7) (1 point) Which one of the following is *not* a (widespread) data shape?

- A. Table
- B. Tree
- C. Sphere
- D. Graph

Empty Pages

You can use the following pages at your convenience.