Université Libanaise
Faculté d'Information
Br I

الجامعة اللبنانية
كلية الاعلام
الفرع الاول

الامتحانات النهائية للفصل السادس
من العام الجامعي 2022-2023

المرحلة: الإجازة السنة المنهجية: الثالثة الاختصاص: علم البيانات - Data Science	المادة: Big Data المدة: ساعتين الأستاذ: د. حسين هزيمة

Documents are NOT authorized

Question 1: (20 pts) true-false questions:

Answer the following questions by true or false, and explain the false statements.

1. We can create more than two RDDs using the same RDD.	T	F	
2. Cluster manager can be replaced with YARN in standalone Spark installation.	T	F	
3. Hadoop hdfs can run on python.	T	F	
4. Spark supports large-scale graph analytics tasks.	T	F	
5. Spark supports machine learning but not data mining tasks.	T	F	
6. Kafka publish-subscribe mechanism is similar to the email massaging mechanism.	T	F	
7. Hadoop MapReduce is execution is faster than Spark.	T	F	
8. Data velocity means the generation of many different types of data.	T	F	
9. When installing Hadoop on Windows, four processes will run; if one of them is failed can we execute a MapReduce job?	T	F	
10. Hadoop can run normally without JVM?	T	F	

Question 2: (15 pts) Kafka and Spark

Assume we have the following data stored in a file named "data.txt".

```
In publishing and graphic design, Lorem
ipsum is a placeholder text commonly used
to demonstrate the visual form of a
document or a typeface without relying on
meaningful content.
```

The data is extracted from CNN, Newyork Times, and BBC websites.

1. How many topics do you think we must create for this data to be published in a Kafka broker?
2. If this data is streamed in real-time and stored on hdfs, can we run a Spark machine learning task on it? Why?
3. Write a Scala code to create an RDD for the input data in the above file using three different methods. Name each method and describe it.

4. Select and use an RDD from part 3) to write a code to create two transformations and their corresponding actions.
5. How many types of transformations exist, and which one would you choose to execute the transformations in question 5?

Question 3: (20 pts) Spark and Hadoop

Compare between Spark and Hadoop by filling the following table:

Features	Spark	Hadoop
Data processing engine		
Speed		
Resource management		
Storage engine		
MapReduce availability		
Written-in		
Supports machine learning		
Multi-language		
Graph processing		

Question 4: (30 pts) Hadoop

Assume that we have a Hadoop single-node cluster is configured on your machine. Write the necessary Hadoop commands to run the following MapReduce job.

1. Format your namenode.
2. What is the result of this command `start-dfs.cmd` ?
3. What is the result of the following command `start-yarn.cmd` ?
4. Create a directory on hdfs named "bigdata".
5. Upload a dataset named "big-dataset" from your pc to the created hdfs folder. The dataset size is 1.5 TBs.
6. Read the content of the dataset on your command line.
7. Run wordcount built-in MapReduce job.
8. How many blocks created for your dataset on hdfs? What is the size of each block?
9. What can you display on this address http://localhost:50070/?
10. Can we display the output result of MapReduce job in part 7) under the given address in part 9)?

Question 5: (15 pts) Big data architecture

Assume that you are a big data engineer and you are responsible to set-up a multi-node cluster.

Given the facts about your cluster:

- 1 master node
- 4 slaves
- 1 dataset of 5 blocks. Each block has 2 replicas.

Draw a diagram that shows all information about your cluster. You must label at least the following terms: namenode, datanode, resource manager, node manager, application master, replicas (free distribution).

Good Work

Big Data (2022, 2023)

Question 1:

1) True
2) True
3) True
5) False, Support both (MLlib)
4) True
6) True
7) False, Spark is faster due to in-memory processing
8) False, velocity: data generation speed
   variety: different types of data
9) False, all 4 processes (NameNode, DataNode, ResourceManager, NodeManager) are needed for proper MapReduce execution
10) False, Hadoop is written in Java and require JVM to run

Question 2:

1) 3 topics, one for each dataSource (CNN, BBC, Newyork Times) to organize data by source
2) Yes, Spark can read data directly from HDFS using Spark streaming, the apply ML algorithms using MLlib
3) 1st Method: From text file

```
val rdd1 = sc.textFile("data.txt")
```

Reads the file directly and creates an RDD where each line is a separate element

2nd method : From collection / parallelizing data

```
val dataSeq = Seq("In publishing ... content")
val rdd2 = sc.parallelize(dataSeq)
```

Convert a local dataset into a distributed RDD across nodes

The code assumes a pre-configured SparkContext named sc is available

3rd method : From existing RDD

```
val rdd3 = rdd1.FlatMap(line => line.split(" "))
```

creates a new RDD by applying a transformation (FlatMap) to an existing RDD (rdd1)

4) // using rdd1 from part 3

```
// Transformation 1: Convert each line to its length (nb. of chars
val lineLengthsRDD = rdd1.map(line => line.length)

// Action 1: Calculate total nb. of chars. in the doc.
val totalChars = lineLengthsRDD.reduce((a,b) => a+b)
println(s"Total characters in document : $totalChars")

// Transformation2: Keeps only lines containing "Lorem"
val FilteredRDD = rdd1.Filter(line => line.contains("Lorem"))

// Action 2: return the nb. of lines containing "Lorem"
val lineCount = FilteredRDD.count()
println(s" Lines with 'Lorem' : $lineCount")
```

5) 2 types of transformations exists. (Narrow and Wide) we used narrow t. in part 4 (map, Filter) which operates on single partitions because they're more efficient for simple text processing task (no shuffling is needed)

Question 3

Features	Spark	Hadoop
Data processing engine	In memory processing (uses RDD and DAG execution engine)	Disk-based processing (MapReduce)
Speed	10-100x Faster than MapReduce (due to in-memory)	Slower due to disk I/O operations
Resource Management	Can use YARN, Mesos or standalone	YARN
Storage	Uses Existing external storage (HDFS, S3, HBase..)	HDFS
MapReduce Availability	Supports optimized MapReduce (RDDs)	Native MapReduce support
written in	Scala (runs on JVM)	Java
Support ML	Yes (MLlib)	No (required external tools like Mahout)
Multi-Language	Scala, Java, Python, R	Java
Graph Processing	Yes (GraphX)	No (requires Giraph or other tools)

Question 4

1) $ hadoop namenode -format
2) start-dfs.cmd : starts the distributed file system daemons (NameNode and DataNode)
3) start-yarn.cmd : starts YARN daemons (ResourceManager and NodeManager)
4) $ hadoop fs -mkdir /bigdata
5) $ hadoop fs -put /local/path/big-dataset /bigdata/

/local/path/big-dataset: local source on pc

/bigdata/: Folder path on hdfs

6) $ hadoop fs -cat /bigdata/big-dataset

7) $ hadoop jar $HADOOP_HOME/share/hadoop/mapredu
hadoop-mapreduce-examples-*.jar wordcount
/bigdata/big-dataset /bigdata/output
input directory output directory

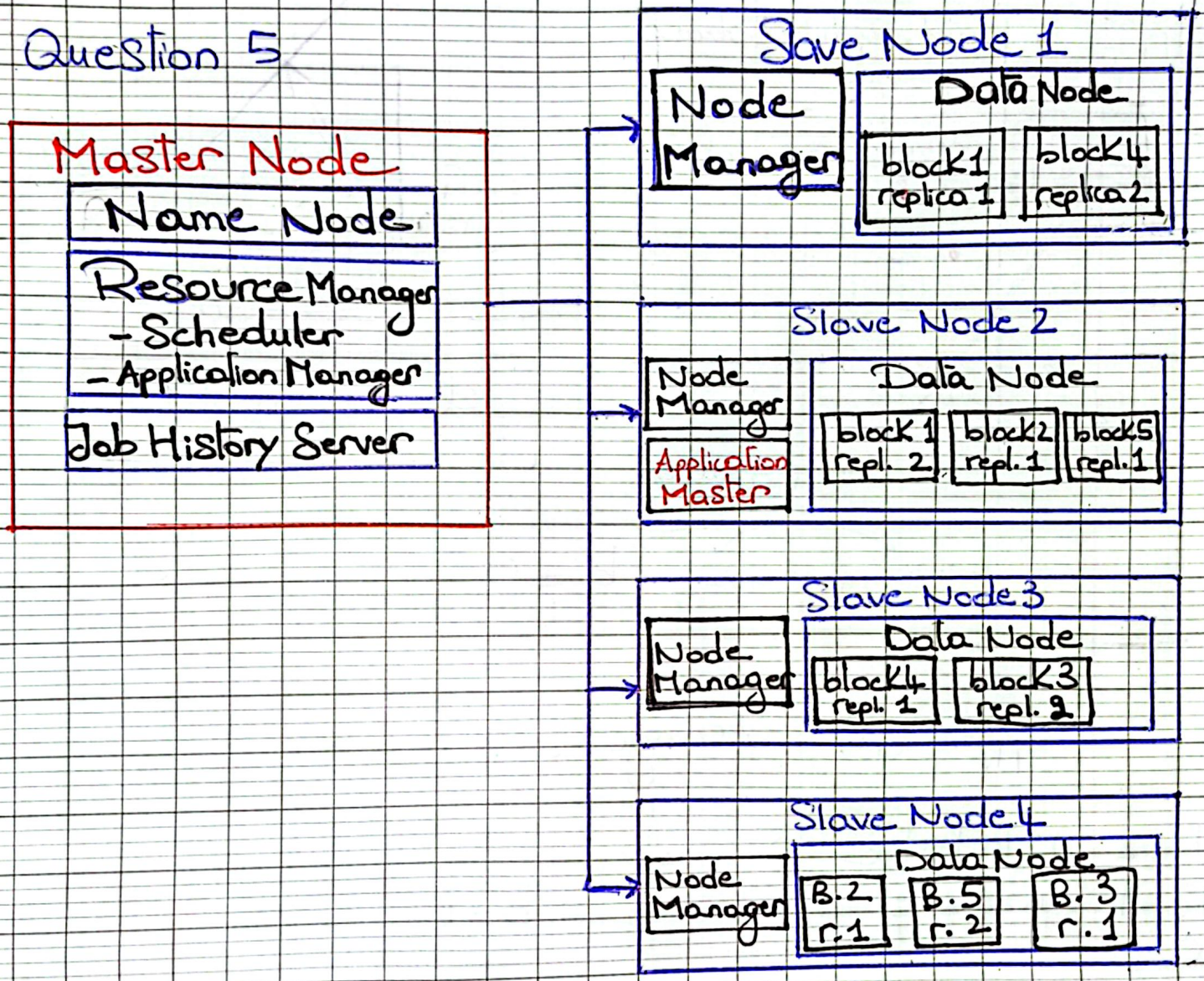
8) Default block size in Hadoop : 128 MB
=> Nb. of blocks : 1.5T x 1024 x 1024 = 1572 864 MB
1.5 T / 128MB = 1572 864MB / 128MB = 12,288 blocks
(of size 128 Mb)

9) http://localhost:50070 : This opens the HDFS NameNode Web UI which shows :
↳ Cluster info,
↳ File System namespace
↳ datanode info.
↳ memory usage
↳ cluster health status

=> other ports :
. 50075 → DataNode UI → DataNode status, block reports, disk usage
. 8088 → ResourceManager UI → view running apps, containers, cluster metrics
. 8042 → NodeManager UI → logs, container info on this node

10)

Question 5


Master Node
Name Node
Resource Manager
- Scheduler
- Application Manager
Job History Server
Slave Node 1
Node Manager
Data Node
block1 replica 1
block4 replica 2
Slave Node 2
Node Manager
Application Master
Data Node
block 1 repl. 2
block2 repl. 1
block5 repl. 1
Slave Node 3
Node Manager
Data Node
block4 repl. 1
block3 repl. 2
Slave Node 4
Node Manager
Data Node
B.2 r.1
B.5 r.2
B.3 r.1