DUMPSARENA

Hortonworks Data Platform Certified Developer

Hortonworks HDPCD

Version Demo

Total Demo Questions: 10

Total Premium Questions: 108

Buy Premium PDF

https://dumpsarena.co

sales@dumpsarena.co

sales@dumpsarena.co
dumpsarena.co

QUESTION NO: 1

Given a directory of files with the following structure: line number, tab character, string:

Example:

1 abialkjfjkaoasdfjksdlkjhqweroij

2 kadfjhuwqounahagtnbvaswslmnbfgy

3 kjfteiomndscxeqalkzhtopedkfsikj

You want to send each line as one record to your Mapper. Which InputFormat should you use to complete the line: conf.setInputFormat (____.class) ; ?

A. SequenceFileAsTextInputFormat

B. SequenceFileInputFormat

C. KeyValueFileInputFormat

D. BDBInputFormat

ANSWER: C

Explanation:

http://stackoverflow.com/questions/9721754/how-to-parse-customwritable-from-text-in-hadoop

QUESTION NO: 2

Which Two of the following statements are true about hdfs? Choose 2 answers

A. An HDFS file that is larger than dfs.block.size is split into blocks

B. Blocks are replicated to multiple datanodes

C. HDFS works best when storing a large number of relatively small files

D. Block sizes for all files must be the same size

ANSWER: A B

QUESTION NO: 3

Which project gives you a distributed, Scalable, data store that allows you random, realtime read/write access to hundreds of terabytes of data?

A. HBase

B. Hue

C. Pig

D. Hive

E. Oozie

F. Flume

G. Sqoop

ANSWER: A

Explanation:

Use Apache HBase when you need random, realtime read/write access to your Big Data.

Note: This project's goal is the hosting of very large tables -- billions of rows X millions of columns -- atop clusters of commodity hardware. Apache HBase is an open-source, distributed, versioned, column-oriented store modeled after Google's Bigtable: A Distributed Storage System for Structured Data by Chang et al. Just as Bigtable leverages the distributed data storage provided by the Google File System, Apache HBase provides Bigtable-like capabilities on top of Hadoop and HDFS.

Features

Linear and modular scalability.

Strictly consistent reads and writes.

Automatic and configurable sharding of tables

Automatic failover support between RegionServers.

Convenient base classes for backing Hadoop MapReduce jobs with Apache HBase tables.

Easy to use Java API for client access.

Block cache and Bloom Filters for real-time queries.

Query predicate push down via server side Filters

Thrift gateway and a REST-ful Web service that supports XML, Protobuf, and binary data encoding options

Extensible jruby-based (JIRB) shell

Support for exporting metrics via the Hadoop metrics subsystem to files or Ganglia; or via JMX

Reference: http://hbase.apache.org/ (when would I use HBase? First sentence)

QUESTION NO: 4

MapReduce v2 (MRv2/YARN) is designed to address which two issues?

A. Single point of failure in the NameNode.

B. Resource pressure on the JobTracker.

C. HDFS latency.

D. Ability to run frameworks other than MapReduce, such as MPI.

E. Reduce complexity of the MapReduce APIs.

F. Standardize on a single MapReduce API.

ANSWER: A B

Explanation:

Reference: Apache Hadoop YARN – Concepts & Applications

QUESTION NO: 5

What data does a Reducer reduce method process?

A. All the data in a single input file.

B. All data produced by a single mapper.

C. All data for a given key, regardless of which mapper(s) produced it.

D. All data for a given value, regardless of which mapper(s) produced it.

ANSWER: C

Explanation:

Reducing lets you aggregate values together. A reducer function receives an iterator of input values from an input list. It then combines these values together, returning a single output value.

All values with the same key are presented to a single reduce task.

Reference: Yahoo! Hadoop Tutorial, Module 4: MapReduce

QUESTION NO: 6

You have just executed a MapReduce job. Where is intermediate data written to after being emitted from the Mapper's map method?

A. Intermediate data in streamed across the network from Mapper to the Reduce and is never written to disk.

B. Into in-memory buffers on the TaskTracker node running the Mapper that spill over and are written into HDFS.

C. Into in-memory buffers that spill over to the local file system of the TaskTracker node running the Mapper.

D. Into in-memory buffers that spill over to the local file system (outside HDFS) of the TaskTracker node running the Reducer

E. Into in-memory buffers on the TaskTracker node running the Reducer that spill over and are written into HDFS.

ANSWER: C

Explanation:

The mapper output (intermediate data) is stored on the Local file system (NOT HDFS) of each individual mapper nodes. This is typically a temporary directory location which can be setup in config by the hadoop administrator. The intermediate data is cleaned up after the Hadoop Job completes.

Reference: 24 Interview Questions & Answers for Hadoop MapReduce developers, Where is the Mapper Output (intermediate kay-value data) stored ?

QUESTION NO: 7

All keys used for intermediate output from mappers must:

A. Implement a splittable compression algorithm.

B. Be a subclass of FileInputFormat.

C. Implement WritableComparable.

D. Override isSplitable.

E. Implement a comparator for speedy sorting.

ANSWER: C

Explanation:

The MapReduce framework operates exclusively on pairs, that is, the framework views the input to the job as a set of pairs and produces a set of pairs as the output of the job, conceivably of different types.

The key and value classes have to be serializable by the framework and hence need to implement the Writable interface. Additionally, the key classes have to implement the WritableComparable interface to facilitate sorting by the framework.

Reference: MapReduce Tutorial

QUESTION NO: 8

In Hadoop 2.2, which TWO of the following processes work together to provide automatic failover of the NameNode? Choose 2 answers

A. ZKFailoverController

B. ZooKeeper

C. QuorumManager

D. JournalNode

ANSWER: A D

QUESTION NO: 9

MapReduce v2 (MRv2/YARN) splits which major functions of the JobTracker into separate daemons? Select two.

A. Heath states checks (heartbeats)

B. Resource management

C. Job scheduling/monitoring

D. Job coordination between the ResourceManager and NodeManager

E. Launching tasks

F. Managing file system metadata

G. MapReduce metric reporting

H. Managing tasks

ANSWER: B C

Explanation:

The fundamental idea of MRv2 is to split up the two major functionalities of the JobTracker, resource management and job scheduling/monitoring, into separate daemons. The idea is to have a global ResourceManager (RM) and per-application ApplicationMaster (AM). An application is either a single job in the classical sense of Map-Reduce jobs or a DAG of jobs.

Note:

The central goal of YARN is to clearly separate two things that are unfortunately smushed together in current Hadoop, specifically in (mainly) JobTracker:

/ Monitoring the status of the cluster with respect to which nodes have which resources available. Under YARN, this will be global.

/ Managing the parallelization execution of any specific job. Under YARN, this will be done separately for each job.

Reference: Apache Hadoop YARN – Concepts & Applications

QUESTION NO: 10

Which two of the following are true about this trivial Pig program' (choose Two)

```
$ pig
grunt>  ABC  =  LOAD 'myfile';
grunt>  DUMP ABC;
```

A. The contents of myfile appear on stdout

B. Pig assumes the contents of myfile are comma delimited

C. ABC has a schema associated with it

D. myfile is read from the user's home directory in HDFS

ANSWER: A D