CCD-410^{Q&As}

Cloudera Certified Developer for Apache Hadoop (CCDH)

Pass Cloudera CCD-410 Exam with 100% Guarantee

Free Download Real Questions & Answers **PDF** and **VCE** file from:

https://www.geekcert.com/ccd-410.html

100% Passing Guarantee
100% Money Back Assurance

Following Questions and Answers are all new published by Cloudera
Official Exam Center

⚙ **Instant Download** After Purchase

⚙ **100% Money Back** Guarantee

⚙ **365 Days** Free Update

⚙ **800,000+** Satisfied Customers

QUESTION 1

You use the hadoop fs put command to write a 300 MB file using and HDFS block size of 64 MB. Just after this command has finished writing 200 MB of this file, what would another user see when trying to access this life?

A. They would see Hadoop throw an ConcurrentFileAccessException when they try to access this file.

B. They would see the current state of the file, up to the last bit written by the command.

C. They would see the current of the file through the last completed block.

D. They would see no content until the whole file written and closed.

Correct Answer: C

QUESTION 2

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

Correct Answer: A

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 3

When is the earliest point at which the reduce method of a given Reducer can be called?

A. As soon as at least one mapper has finished processing its input split.

B. As soon as a mapper has emitted at least one record.

C. Not until all mappers have finished processing all records.

D. It depends on the InputFormat used for the job.

Correct Answer: C

In a MapReduce job reducers do not start executing the reduce method until the all Map jobs have completed. Reducers start copying intermediate key-value pairs from the mappers as soon as they are available. The programmer defined reduce method is called only after all the mappers have finished.

Note: The reduce phase has 3 steps: shuffle, sort, reduce. Shuffle is where the data is collected by the reducer from each mapper. This can happen while mappers are generating data since it is only a data transfer. On the other hand, sort and reduce can only start once all the mappers are done.

Why is starting the reducers early a good thing? Because it spreads out the data transfer from the mappers to the reducers over time, which is a good thing if your network is the bottleneck.

Why is starting the reducers early a bad thing? Because they "hog up" reduce slots while only copying data. Another job that starts later that will actually use the reduce slots now can\\'t use them.

You can customize when the reducers startup by changing the default value of mapred.reduce.slowstart.completed.maps in mapred-site.xml. A value of 1.00 will wait for all the mappers to finish before starting the reducers. A value of 0.0 will start the reducers right away. A value of 0.5 will start the reducers when half of the mappers are complete. You can also change mapred.reduce.slowstart.completed.maps on a job-by-job basis. Typically, keep mapred.reduce.slowstart.completed.maps above 0.9 if the system ever has multiple jobs running at once. This way the job doesn\\'t hog up reducers when they aren\\'t doing anything but copying data. If you only ever have one job running at a time, doing 0.1 would probably be appropriate.

Reference: 24 Interview Questions and Answers for Hadoop MapReduce developers, When is the reducers are started in a MapReduce job?

QUESTION 4

You write MapReduce job to process 100 files in HDFS. Your MapReduce algorithm uses TextInputFormat: the mapper applies a regular expression over input values and emits key- values pairs with the key consisting of the matching text, and the value containing the filename and byte offset. Determine the difference between setting the number of reduces to one and settings the number of reducers to zero.

A. There is no difference in output between the two settings.

B. With zero reducers, no reducer runs and the job throws an exception. With one reducer, instances of matching patterns are stored in a single file on HDFS.

C. With zero reducers, all instances of matching patterns are gathered together in one file on HDFS. With one reducer, instances of matching patterns are stored in multiple files on HDFS.

D. With zero reducers, instances of matching patterns are stored in multiple files on HDFS. With one reducer, all instances of matching patterns are gathered together in one file on HDFS.

Correct Answer: D

*

 It is legal to set the number of reduce-tasks to zero if no reduction is desired.

In this case the outputs of the map-tasks go directly to the FileSystem, into the output path set by setOutputPath(Path). The framework does not sort the map-outputs before writing them out to the FileSystem.

*

 Often, you may want to process input data using a map function only. To do this, simply set mapreduce.job.reduces to zero. The MapReduce framework will not create any reducer tasks. Rather, the outputs of the mapper tasks will be the final output of the job.

Note:

Reduce

In this phase the reduce(WritableComparable, Iterator, OutputCollector, Reporter) method is called for each pair in the grouped inputs.

The output of the reduce task is typically written to the FileSystem via OutputCollector.collect (WritableComparable, Writable).

Applications can use the Reporter to report progress, set application-level status messages and update Counters, or just indicate that they are alive.

The output of the Reducer is not sorted.

QUESTION 5

In a MapReduce job with 500 map tasks, how many map task attempts will there be?

A. It depends on the number of reduces in the job.

B. Between 500 and 1000.

C. At most 500.

D. At least 500.

E. Exactly 500.

Correct Answer: D

Explanation: From Cloudera Training Course: Task attempt is a particular instance of an attempt to execute a task There will be at least as many task attempts as there are tasks If a task attempt fails, another will be started by the JobTracker Speculative execution can also result in more task attempts than completed tasks

CCD-410 Practice Test   CCD-410 Study Guide   CCD-410 Exam Questions