



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.passapply.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

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 2

Table metadata in Hive is:

- A. Stored as metadata on the NameNode.
- B. Stored along with the data in HDFS.
- C. Stored in the Metastore.
- D. Stored in ZooKeeper.

Correct Answer: C



By default, hive use an embedded Derby database to store metadata information. The metastore is the "glue" between Hive and HDFS. It tells Hive where your data files live in HDFS, what type of data they contain, what tables they belong to, etc.

The Metastore is an application that runs on an RDBMS and uses an open source ORM layer called DataNucleus, to convert object representations into a relational schema and vice versa. They chose this approach as opposed to storing this information in hdfs as they need the Metastore to be very low latency. The DataNucleus layer allows them to plugin many different RDBMS technologies.

Note:

*

By default, Hive stores metadata in an embedded Apache Derby database, and other client/server databases like MySQL can optionally be used.

*

features of Hive include:

Metadata storage in an RDBMS, significantly reducing the time to perform semantic checks during query execution.

Reference: Store Hive Metadata into RDBMS

QUESTION 3

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 is 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.

Correct Answer: C

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 and Answers for Hadoop MapReduce developers, Where is the Mapper Output (intermediate key-value data) stored ?

QUESTION 4



What types of algorithms are difficult to express in MapReduce v1 (MRv1)?

- A. Algorithms that require applying the same mathematical function to large numbers of individual binary records.
- B. Relational operations on large amounts of structured and semi-structured data.
- C. Algorithms that require global, sharing states.
- D. Large-scale graph algorithms that require one-step link traversal.
- E. Text analysis algorithms on large collections of unstructured text (e.g, Web crawls).

Correct Answer: C

See 3) below.

Limitations of Mapreduce where not to use Mapreduce While very powerful and applicable to a wide variety of problems, MapReduce is not the answer to every problem. Here are some problems I found where MapReudce is not suited and some papers that address the limitations of MapReuce.

1.

Computation depends on previously computed values

If the computation of a value depends on previously computed values, then MapReduce cannot be used. One good example is the Fibonacci series where each value is summation of the previous two values. i.e., $f(k+2) = f(k+1) + f(k)$. Also, if the data set is small enough to be computed on a single machine, then it is better to do it as a single `reduce(map(data))` operation rather than going through the entire map reduce process.

2.

Full-text indexing or ad hoc searching

The index generated in the Map step is one dimensional, and the Reduce step must not generate a large amount of data or there will be a serious performance degradation. For example, CouchDB's MapReduce may not be a good fit for full-text indexing or ad hoc searching. This is a problem better suited for a tool such as Lucene.

3.

Algorithms depend on shared global state

Solutions to many interesting problems in text processing do not require global synchronization. As a result, they can be expressed naturally in MapReduce, since map and reduce tasks run independently and in isolation. However, there are many examples of algorithms that depend crucially on the existence of shared global state during processing, making them difficult to implement in MapReduce (since the single opportunity for global synchronization in MapReduce is the barrier between the map and reduce phases of processing)

Reference: Limitations of Mapreduce where not to use Mapreduce

QUESTION 5

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 isSplittable.
- E. Implement a comparator for speedy sorting.

Correct Answer: C

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

[CCD-410 Practice Test](#)

[CCD-410 Exam Questions](#)

[CCD-410 Braindumps](#)