ISLEVER

CCD-410

Cloudera Certified Developer for Apache Hadoop (CCDH)

DEMO

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

https://www.islever.com/cloudera.html

QUESTION NO: 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.

Answer: C
Explanation: 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 & Answers for Hadoop MapReduce developers, When is the reducers are started in a MapReduce job?

QUESTION NO: 2

Which describes how a client reads a file from HDFS?

A. The client queries the NameNode for the block location(s). The NameNode returns the block location(s) to the client. The client reads the data directory off the DataNode(s).
B. The client queries all DataNodes in parallel. The DataNode that contains the requested data responds directly to the client. The client reads the data directly off the DataNode.
C. The client contacts the NameNode for the block location(s). The NameNode then queries the DataNodes for block locations. The DataNodes respond to the NameNode, and the NameNode redirects the client to the DataNode that holds the requested data block(s). The client then reads the data directly off the DataNode.
D. The client contacts the NameNode for the block location(s). The NameNode contacts the DataNode that holds the requested data block. Data is transferred from the DataNode to the NameNode, and then from the NameNode to the client.

Answer: A
Explanation: 8.2.4. HDFS ClientUser applications access the filesystem using the HDFS client, a library that exports the HDFS filesystem interface.
Like most conventional filesystems, HDFS supports operations to read, write and delete files, and operations to create and delete directories. The user references files and directories by paths in the namespace. The user application does not need to know that filesystem metadata and storage are on different servers, or that blocks have multiple replicas.
When an application reads a file, the HDFS client first asks the NameNode for the list of DataNodes that host replicas of the blocks of the file. The list is sorted by the network topology distance from the client. The client contacts a DataNode directly and requests the transfer of the desired block. When a client writes, it first asks the NameNode to choose DataNodes to host replicas of the first block of the file. The client organizes a pipeline from node-to-node and sends the data. When the first block is filled, the client requests new DataNodes to be chosen to host replicas of the next block. A new pipeline is organized, and the client sends the further bytes of the file. Choice of DataNodes for each block is likely to be different.
Reference:

http://www.aosabook.org/en/hdfs.html

QUESTION NO: 3

You are developing a combiner that takes as input Text keys, IntWritable values, and emits Text keys, IntWritable values. Which interface should your class implement?

A. Combiner <Text, IntWritable, Text, IntWritable>

B. Mapper <Text, IntWritable, Text, IntWritable>
C. Reducer <Text, Text, IntWritable, IntWritable>
D. Reducer <Text, IntWritable, Text, IntWritable>
E. Combiner <Text, Text, IntWritable, IntWritable>

Answer: D
Explanation:

QUESTION NO: 4

Indentify the utility that allows you to create and run MapReduce jobs with any executable or script as the mapper and/or the reducer?

A. Oozie
B. Sqoop
C. Flume
D. Hadoop Streaming
E. mapred

Answer: D
Explanation: Hadoop streaming is a utility that comes with the Hadoop distribution. The utility allows you to create and run Map/Reduce jobs with any executable or script as the mapper and/or the reducer.

Reference: http://hadoop.apache.org/common/docs/r0.20.1/streaming.html (Hadoop Streaming, second sentence)

QUESTION NO: 5

How are keys and values presented and passed to the reducers during a standard sort and shuffle phase of MapReduce?

A. Keys are presented to reducer in sorted order; values for a given key are not sorted.
B. Keys are presented to reducer in sorted order; values for a given key are sorted in ascending order.
C. Keys are presented to a reducer in random order; values for a given key are not sorted.
D. Keys are presented to a reducer in random order; values for a given key are sorted in ascending order.