Exam Number/Code: CCD-410

Exam Name: Cloudera Certified Developer for Apache Hadoop (CCDH)

Version: Demo

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: C

Explanation:
The Client communication to HDFS happens using Hadoop HDFS API. Client applications talk to the NameNode whenever they wish to locate a file, or when they want to add/copy/move/delete a file on HDFS. The NameNode responds the successful requests by returning a list of relevant DataNode servers where the data lives. Client applications can talk directly to a DataNode, once the NameNode has provided the location of the data.
Reference: 24 Interview Questions & Answers for Hadoop MapReduce developers, How the Client communicates with HDFS?

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

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.

Answer: A

Explanation: Reducer has 3 primary phases:
1. Shuffle
The Reducer copies the sorted output from each Mapper using HTTP across the network.
2. Sort
The framework merge sorts Reducer inputs by keys (since different Mappers may have output the same key).
The shuffle and sort phases occur simultaneously i.e. while outputs are being fetched they are merged.
SecondarySort
To achieve a secondary sort on the values returned by the value iterator, the application should extend the key with the secondary key and define a grouping comparator. The keys will be sorted using the entire key, but will be grouped using the grouping comparator to decide which keys and values are sent in the same call to reduce.
3. Reduce
In this phase the reduce(Object, Iterable, Context) method is called for each <key, (collection of values)> in the sorted inputs.
The output of the reduce task is typically written to a RecordWriter via
TaskInputOutputContext.write(Object, Object).
The output of the Reducer is not re-sorted.

Reference: org.apache.hadoop.mapreduce, Class Reducer<KEYIN,VALUEIN,KEYOUT,VALUEOUT>