Vendor: Cloudera

Exam Code: CCD-410

Exam Name: Cloudera Certified Developer for Apache Hadoop

Version: Demo

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
Explanation

Explanation/Reference:
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 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.

Correct Answer: A
Explanation

Explanation/Reference:
Reference: 24 Interview Questions & Answers for Hadoop MapReduce developers, How the Client communicates with HDFS?

QUESTION 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>

Correct Answer: D
Explanation

Explanation/Reference:

QUESTION 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

Correct Answer: D
Explanation

Explanation/Reference:
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 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.

Correct Answer: A
Explanation

Explanation/Reference:
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>

QUESTION 6
Assuming default settings, which best describes the order of data provided to a reducer's reduce method:

A. The keys given to a reducer aren't in a predictable order, but the values associated with those keys always are.
B. Both the keys and values passed to a reducer always appear in sorted order.
C. Neither keys nor values are in any predictable order.
D. The keys given to a reducer are in sorted order but the values associated with each key are in no predictable order

Correct Answer: D
Explanation

Explanation/Reference:
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>

QUESTION 7
You wrote a map function that throws a runtime exception when it encounters a control character in input data. The input supplied to your mapper contains twelve such characters totals, spread across five file splits. The first four file splits each have two control characters and the last split has four control characters.

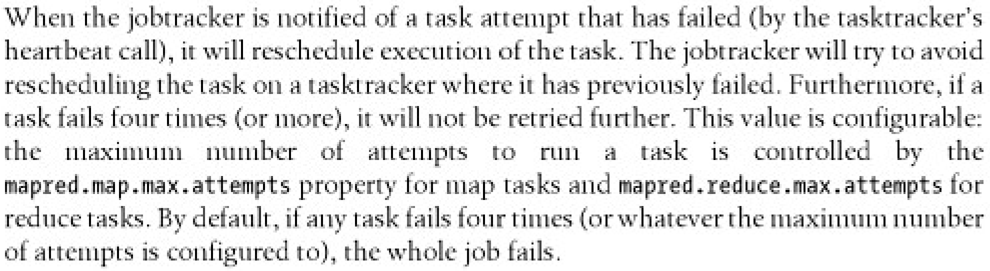
Indentify the number of failed task attempts you can expect when you run the job with mapred.max.map.attempts set to 4:

A. You will have forty-eight failed task attempts
B. You will have seventeen failed task attempts
C. You will have five failed task attempts
D. You will have twelve failed task attempts
E. You will have twenty failed task attempts

Correct Answer: E
Explanation

Explanation/Reference:
There will be four failed task attempts for each of the five file splits.

Note:



When the jobtracker is notified of a task attempt that has failed (by the tasktracker's heartbeat call), it will reschedule execution of the task. The jobtracker will try to avoid rescheduling the task on a tasktracker where it has previously failed. Furthermore, if a task fails four times (or more), it will not be retried further. This value is configurable: the maximum number of attempts to run a task is controlled by the `mapred.map.max.attempts` property for map tasks and `mapred.reduce.max.attempts` for reduce tasks. By default, if any task fails four times (or whatever the maximum number of attempts is configured to), the whole job fails.

QUESTION 8
You want to populate an associative array in order to perform a map-side join. You've decided to put this information in a text file, place that file into the DistributedCache and read it in your Mapper before any records are processed.

Indentify which method in the Mapper you should use to implement code for reading the file and populating the associative array?

A. combine
B. map
C. init
D. configure

Correct Answer: B
Explanation

Explanation/Reference:
Reference: org.apache.hadoop.filecache , Class DistributedCache

QUESTION 9
You've written a MapReduce job that will process 500 million input records and generated 500 million key-

value pairs. The data is not uniformly distributed. Your MapReduce job will create a significant amount of intermediate data that it needs to transfer between mappers and reduces which is a potential bottleneck. A custom implementation of which interface is most likely to reduce the amount of intermediate data transferred across the network?

A. Partitioner
B. OutputFormat
C. WritableComparable
D. Writable
E. InputFormat
F. Combiner

Correct Answer: F
Explanation

Explanation/Reference:
Combiners are used to increase the efficiency of a MapReduce program. They are used to aggregate intermediate map output locally on individual mapper outputs. Combiners can help you reduce the amount of data that needs to be transferred across to the reducers. You can use your reducer code as a combiner if the operation performed is commutative and associative.

Reference: 24 Interview Questions & Answers for Hadoop MapReduce developers, What are combiners? When should I use a combiner in my MapReduce Job?

QUESTION 10
Can you use MapReduce to perform a relational join on two large tables sharing a key? Assume that the two tables are formatted as comma-separated files in HDFS.

A. Yes.
B. Yes, but only if one of the tables fits into memory
C. Yes, so long as both tables fit into memory.
D. No, MapReduce cannot perform relational operations.
E. No, but it can be done with either Pig or Hive.

Correct Answer: A
Explanation

Explanation/Reference:
Note:
* Join Algorithms in MapReduce

A) Reduce-side join

B) Map-side join

C) In-memory join

/ Striped Striped variant variant

/ Memcached variant

* Which join to use?

/ In-memory join > map-side join > reduce-side join

/ Limitations of each?

In-memory join: memory

Map-side join: sort order and partitioning

Reduce-side join: general purpose

QUESTION 11
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.

Correct Answer: C
Explanation

Explanation/Reference:
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 12
You want to understand more about how users browse your public website, such as which pages they visit prior to placing an order. You have a farm of 200 web servers hosting your website.
How will you gather this data for your analysis?

A. Ingest the server web logs into HDFS using Flume.
B. Write a MapReduce job, with the web servers for mappers, and the Hadoop cluster nodes for reduces.
C. Import all users' clicks from your OLTP databases into Hadoop, using Sqoop.
D. Channel these clickstreams inot Hadoop using Hadoop Streaming.
E. Sample the weblogs from the web servers, copying them into Hadoop using curl.

Correct Answer: A
Explanation

Explanation/Reference:

QUESTION 13
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.

Correct Answer: BD
Explanation

Explanation/Reference:

YARN (Yet Another Resource Negotiator), as an aspect of Hadoop, has two major kinds of benefits:
* (D) The ability to use programming frameworks other than MapReduce. / MPI (Message Passing Interface) was mentioned as a paradigmatic example of a MapReduce alternative
* Scalability, no matter what programming framework you use.
Note:
* 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.

* (B) 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.
The current Hadoop MapReduce system is fairly scalable -- Yahoo runs 5000 Hadoop jobs, truly concurrently, on a single cluster, for a total 1.5 2 millions jobs/cluster/month. Still, YARN will remove scalability bottlenecks

Reference: Apache Hadoop YARN Concepts & Applications

QUESTION 14
You need to run the same job many times with minor variations. Rather than hardcoding all job configuration options in your drive code, you've decided to have your Driver subclass org.apache.hadoop.conf.Configured and implement the org.apache.hadoop.util.Tool interface. Indentify which invocation correctly passes.mapred.job.name with a value of Example to Hadoop?

A. hadoop "mapred.job.name=Example" MyDriver input output
B. hadoop MyDriver mapred.job.name=Example input output
C. hadoop MyDrive D mapred.job.name=Example input output
D. hadoop setproperty mapred.job.name=Example MyDriver input output
E. hadoop setproperty ("mapred.job.name=Example") MyDriver input output

Correct Answer: C
Explanation

Explanation/Reference:
Configure the property using the -D key=value notation:

-D mapred.job.name='My Job'
You can list a whole bunch of options by calling the streaming jar with just the -info argument

Reference: Python hadoop streaming : Setting a job name

QUESTION 15
You are developing a MapReduce job for sales reporting. The mapper will process input keys representing the year (IntWritable) and input values representing product indentifies (Text). Indentify what determines the data types used by the Mapper for a given job.

A. The key and value types specified in the JobConf.setMapInputKeyClass and JobConf.setMapInputValuesClass methods
B. The data types specified in HADOOP_MAP_DATATYPES environment variable
C. The mapper-specification.xml file submitted with the job determine the mapper's input key and value types.
D. The InputFormat used by the job determines the mapper's input key and value types.

Correct Answer: D
Explanation

Submit A Ticket