Cloudera CCD-333

Cloudera Certified Developer for Apache Hadoop

Version: 5.6

QUESTION NO: 1

What is a SequenceFile?

A. A SequenceFile contains a binary encoding of an arbitrary number of homogeneous writable objects.
B. A SequenceFile contains a binary encoding of an arbitrary number of heterogeneous writable objects.
C. A SequenceFile contains a binary encoding of an arbitrary number of WritableComparable objects, in sorted order.
D. A SequenceFile contains a binary encoding of an arbitrary number key-value pairs. Each key must be the same type. Each value must be same type.

Answer: D
Explanation: SequenceFile is a flat file consisting of binary key/value pairs.

There are 3 different SequenceFile formats:

Uncompressed key/value records.
Record compressed key/value records - only 'values' are compressed here.
Block compressed key/value records - both keys and values are collected in 'blocks' separately and compressed. The size of the 'block' is configurable.

Reference:http://wiki.apache.org/hadoop/SequenceFile

QUESTION NO: 2

Given a directory of files with the following structure: line number, tab character, string:

Example:

1. abialkjfjkaoasdfjksdlkjhqweroij

2. kadf jhuwqounahagtnbvaswslmnbfgy

3. kjfteiomndscxeqalkzhtopedkfslkj

You want to send each line as one record to your Mapper. Which InputFormat would you use to complete the line: setInputFormat (_____.class);

A. BDBInputFormat
B. KeyValueTextInputFormat

C. SequenceFileInputFormat
D. SequenceFileAsTextInputFormat

Answer: C

Explanation: Note:
The output format for your first MR job should be SequenceFileOutputFormat - this will store the Key/Values output from the reducer in a binary format, that can then be read back in, in your second MR job using SequenceFileInputFormat.

Reference:http://stackoverflow.com/questions/9721754/how-to-parse-customwritable-from-text-in-hadoop(see answer 1 and then see the comment #1 for it)

QUESTION NO: 3

In a MapReduce job, you want each of you input files processed by a single map task. How do you configure a MapReduce job so that a single map task processes each input file regardless of how many blocks the input file occupies?

A. Increase the parameter that controls minimum split size in the job configuration.
B. Write a custom MapRunner that iterates over all key-value pairs in the entire file.
C. Set the number of mappers equal to the number of input files you want to process.
D. Write a custom FileInputFormat and override the method isSplittable to always return false.

Answer: D

Explanation: Note:
*// Do not allow splitting.
protected boolean isSplittable(JobContext context, Path filename) {
return false;
}
*InputSplits: An InputSplit describes a unit of work that comprises a single map task in a MapReduce program. A MapReduce program applied to a data set, collectively referred to as a Job, is made up of several (possibly several hundred) tasks. Map tasks may involve reading a whole file; they often involve reading only part of a file. By default, the FileInputFormat and its descendants break a file up into 64 MB chunks (the same size as blocks in HDFS). You can control this value by setting the mapred.min.split.size parameter in hadoop-site.xml, or by overriding the parameter in the JobConf object used to submit a particular MapReduce job. By processing a file in chunks, we allow several map tasks to operate on a single file in parallel. If the file is very large, this can improve performance significantly through parallelism. Even more importantly, since the various blocks that make up the file may be spread across several different nodes in the cluster, it allows tasks to be scheduled on each of these different nodes; the

individual blocks are thus all processed locally, instead of needing to be transferred from one node to another. Of course, while log files can be processed in this piece-wise fashion, some file formats are not amenable to chunked processing. By writing a custom InputFormat, you can control how the file is broken up (or is not broken up) into splits.

QUESTION NO: 4

Which of the following best describes the workings of TextInputFormat?

A. Input file splits may cross line breaks. A line that crosses tile splits is ignored.
B. The input file is split exactly at the line breaks, so each Record Reader will read a series of complete lines.
C. Input file splits may cross line breaks. A line that crosses file splits is read by the RecordReaders of both splits containing the broken line.
D. Input file splits may cross line breaks. A line that crosses file splits is read by the RecordReader of the split that contains the end of the broken line.
E. Input file splits may cross line breaks. A line that crosses file splits is read by the RecordReader of the split that contains the beginning of the broken line.

Answer: D
Explanation: As the Map operation is parallelized the input file set is first split to several pieces called FileSplits. If an individual file is so large that it will affect seek time it will be split to several Splits. The splitting does not know anything about the input file's internal logical structure, for example line-oriented text files are split on arbitrary byte boundaries. Then a new map task is created per FileSplit.

When an individual map task starts it will open a new output writer per configured reduce task. It will then proceed to read its FileSplit using the RecordReader it gets from the specified InputFormat. InputFormat parses the input and generates key-value pairs. InputFormat must also handle records that may be split on the FileSplit boundary. For example TextInputFormat will read the last line of the FileSplit past the split boundary and, when reading other than the first FileSplit, TextInputFormat ignores the content up to the first newline.

Reference:How Map and Reduce operations are actually carried out

http://wiki.apache.org/hadoop/HadoopMapReduce(Map, second paragraph)

QUESTION NO: 5

Which of the following statements most accurately describes the relationship between MapReduce and Pig?

A. Pig provides additional capabilities that allow certain types of data manipulation not possible with MapReduce.
B. Pig provides no additional capabilities to MapReduce. Pig programs are executed as MapReduce jobs via the Pig interpreter.
C. Pig programs rely on MapReduce but are extensible, allowing developers to do special-purpose processing not provided by MapReduce.
D. Pig provides the additional capability of allowing you to control the flow of multiple MapReduce jobs.

Answer: D
Explanation: In addition to providing many relational and data flow operators Pig Latin provides ways for you to control how your jobs execute on MapReduce. It allows you to set values that control your environment and to control details of MapReduce such as how your data is partitioned.

Reference:http://ofps.oreilly.com/titles/9781449302641/advanced_pig_latin.html(topic: controlling execution)

QUESTION NO: 6

You need to import a portion of a relational database every day as files to HDFS, and generate Java classes to Interact with your imported data. Which of the following tools should you use to accomplish this?

A. Pig
B. Hue
C. Hive
D. Flume
E. Sqoop
F. Oozie
G. fuse-dfs

Answer: E
Explanation: Sqoop (“SQL-to-Hadoop”) is a straightforward command-line tool with the following capabilities:

Imports individual tables or entire databases to files in HDFS
Generates Java classes to allow you to interact with your imported data

Provides the ability to import from SQL databases straight into your Hive data warehouse

Note:
Data Movement Between Hadoop and Relational Databases
Data can be moved between Hadoop and a relational database as a bulk data transfer, or relational tables can be accessed from within a MapReduce map function.

Note:
*Cloudera's Distribution for Hadoop provides a bulk data transfer tool (i.e., Sqoop) that imports individual tables or entire databases into HDFS files. The tool also generates Java classes that support interaction with the imported data. Sqoop supports all relational databases over JDBC, and Quest Software provides a connector (i.e., OraOop) that has been optimized for access to data residing in Oracle databases.

Reference:http://log.medcl.net/item/2011/08/hadoop-and-mapreduce-big-data-analytics-gartner/(Data Movement between hadoop and relational databases, second paragraph)

QUESTION NO: 7

You have an employee who is a Date Analyst and is very comfortable with SQL. He would like to run ad-hoc analysis on data in your HDFS duster. Which of the following is a data warehousing software built on top of Apache Hadoop that defines a simple SQL-like query language well-suited for this kind of user?

A. Pig
B. Hue
C. Hive
D. Sqoop
E. Oozie
F. Flume
G. Hadoop Streaming

Answer: C
Explanation: Hive defines a simple SQL-like query language, called QL, that enables users familiar with SQL to query the data. At the same time, this language also allows programmers who are familiar with the MapReduce framework to be able to plug in their custom mappers and reducers to perform more sophisticated analysis that may not be supported by the built-in capabilities of the language. QL can also be extended with custom scalar functions (UDF's), aggregations (UDAF's), and table functions (UDTF's).

Reference:https://cwiki.apache.org/Hive/(Apache Hive, first sentence and second paragraph)

QUESTION NO: 8

Workflows expressed in Oozie can contain:

A. Iterative repetition of MapReduce jobs until a desired answer or state is reached.
B. Sequences of MapReduce and Pig jobs. These are limited to linear sequences of actions with exception handlers but no forks.
C. Sequences of MapReduce jobs only; no Pig or Hive tasks or jobs. These MapReduce sequences can be combined with forks and path joins.
D. Sequences of MapReduce and Pig. These sequences can be combined with other actions including forks, decision points, and path joins.

Answer: D
Reference:http://incubator.apache.org/oozie/docs/3.1.3/docs/WorkflowFunctionalSpec.html(workflow definition, first sentence)

QUESTION NO: 9

You need a distributed, scalable, data Store that allows you random, realtime read/write access to hundreds of terabytes of data. Which of the following would you use?

A. Hue
B. Pig
C. Hive
D. Oozie
E. HBase
F. Flume
G. Sqoop

Answer: E
Explanation: 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 NO: 10

Which of the following utilities 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

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)