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.geekcert.com/ccd-410.html

100% Passing Guarantee
100% Money Back Assurance

Following Questions and Answers are all new published by Cloudera Official Exam Center

QUESTION 1

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 2

You have a directory named jobdata in HDFS that contains four files: _first.txt, second.txt, .third.txt and #data.txt. How many files will be processed by the FileInputFormat.setInputPaths () command when it\\'s given a path object representing this directory?

A. Four, all files will be processed

B. Three, the pound sign is an invalid character for HDFS file names

C. Two, file names with a leading period or underscore are ignored

D. None, the directory cannot be named jobdata

E. One, no special characters can prefix the name of an input file

Correct Answer: C

Files starting with \\'_\\' are considered \\'hidden\\' like unix files starting with \\'.\\'.

# characters are allowed in HDFS file names.

QUESTION 3

Which process describes the lifecycle of a Mapper?

A. The JobTracker calls the TaskTracker\\'s configure () method, then its map () method and finally its close () method.

B. The TaskTracker spawns a new Mapper to process all records in a single input split.

C. The TaskTracker spawns a new Mapper to process each key-value pair.

D. The JobTracker spawns a new Mapper to process all records in a single file.

Correct Answer: B

For each map instance that runs, the TaskTracker creates a new instance of your mapper. Note:

*

The Mapper is responsible for processing Key/Value pairs obtained from the InputFormat. The mapper may perform a number of Extraction and Transformation functions on the Key/Value pair before ultimately outputting none, one or many Key/Value pairs of the same, or different Key/Value type.

*

With the new Hadoop API, mappers extend the org.apache.hadoop.mapreduce.Mapper class. This class defines an \\'Identity\\' map function by default - every input Key/Value pair obtained from the InputFormat is written out.

Examining the run() method, we can see the lifecycle of the mapper:

/**

*

Expert users can override this method for more complete control over the

```
*
 execution of the Mapper.
*
 @param context
*
 @throws IOException
*/
public void run(Context context) throws IOException, InterruptedException {
setup(context);
while (context.nextKeyValue()) {
map(context.getCurrentKey(), context.getCurrentValue(), context);
}
cleanup(context);
}
```

setup(Context) - Perform any setup for the mapper. The default implementation is a no-op method.

map(Key, Value, Context) - Perform a map operation in the given Key / Value pair. The default

implementation calls Context.write(Key, Value)

cleanup(Context) - Perform any cleanup for the mapper. The default implementation is a no-op method.

Reference: Hadoop/MapReduce/Mapper

QUESTION 4

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

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 5

To process input key-value pairs, your mapper needs to lead a 512 MB data file in memory. What is the best way to accomplish this?

A. Serialize the data file, insert in it the JobConf object, and read the data into memory in the configure method of the mapper.

B. Place the data file in the DistributedCache and read the data into memory in the map method of the mapper.

C. Place the data file in the DataCache and read the data into memory in the configure method of the mapper.

D. Place the data file in the DistributedCache and read the data into memory in the configure method of the mapper.

Correct Answer: D