APACHE-HADOOP-DEVELOPER$^{Q\&As}$

Hadoop 2.0 Certification exam for Pig and Hive Developer

QUESTION 1

Which one of the following is NOT a valid Oozie action?

A. mapreduce

B. pig

C. hive

D. mrunit

Correct Answer: D

QUESTION 2

Which one of the following statements is true about a Hive-managed table?

A. Records can only be added to the table using the Hive INSERT command.

B. When the table is dropped, the underlying folder in HDFS is deleted.

C. Hive dynamically defines the schema of the table based on the FROM clause of a SELECT query.

D. Hive dynamically defines the schema of the table based on the format of the underlying data.

Correct Answer: B

QUESTION 3

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

Explanation: 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 4

You want to count the number of occurrences for each unique word in the supplied input data. You\\'ve decided to implement this by having your mapper tokenize each word and emit a literal value 1, and then have your reducer increment a counter for each literal 1 it receives. After successful implementing this, it occurs to you that you could optimize this by specifying a combiner. Will you be able to reuse your existing Reduces as your combiner in this case and why or why not?

A. Yes, because the sum operation is both associative and commutative and the input and output types to the reduce method match.

B. No, because the sum operation in the reducer is incompatible with the operation of a Combiner.

C. No, because the Reducer and Combiner are separate interfaces.

D. No, because the Combiner is incompatible with a mapper which doesn\\'t use the same data type for both the key and value.

E. Yes, because Java is a polymorphic object-oriented language and thus reducer code can be reused as a combiner.

Correct Answer: A

Explanation: 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. The execution of combiner is not guaranteed, Hadoop may or may not execute a combiner. Also, if required it may execute it more then 1 times. Therefore your MapReduce jobs should not depend on the combiners execution.

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

QUESTION 5

Given a directory of files with the following structure: line number, tab character, string: Example: 1abialkjfjkaoasdfjksdlkjhqweroij 2kadfjhuwqounahagtnbvaswslmnbfgy 3kjfteiomndscxeqalkzhtopedkfsikj You want to send each line as one record to your Mapper. Which InputFormat should you use to complete

the line: conf.setInputFormat (____.class) ; ?

A. SequenceFileAsTextInputFormat

B. SequenceFileInputFormat

C. KeyValueFileInputFormat

D. BDBInputFormat

Correct Answer: C

http://stackoverflow.com/questions/9721754/how-to-parse-customwritable-from-text-in- hadoop