

# HADOOP-PR000007<sup>Q&As</sup>

Hortonworks Certified Apache Hadoop 2.0 Developer (Pig and Hive Developer)

# Pass Hortonworks HADOOP-PR000007 Exam with 100% Guarantee

Free Download Real Questions & Answers **PDF** and **VCE** file from:

https://www.passapply.com/hadoop-pr000007.html

100% Passing Guarantee 100% Money Back Assurance

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

- Instant Download After Purchase
- 100% Money Back Guarantee
- 😳 365 Days Free Update
- 800,000+ Satisfied Customers





#### **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: 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 and Answers for Hadoop MapReduce developers, When is the reducers are started in a MapReduce job?

#### **QUESTION 2**

Examine the following Pig commands:

```
customers = LOAD 'data' USING PigStorage(',')
AS (name:chararray,age:int,zip:int,receipts:double);
s = SAMPLE customers 0.2;
DUMP s;
```

Which one of the following statements is true?



- A. The SAMPLE command generates an "unexpected symbol" error
- B. Each MapReduce task will terminate after executing for 0.2 minutes
- C. The reducers will only output the first 20% of the data passed from the mappers
- D. A random sample of approximately 20% of the data will be output

Correct Answer: D

## **QUESTION 3**

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: 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 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**

Which best describes what the map method accepts and emits?

A. It accepts a single key-value pair as input and emits a single key and list of corresponding values as output.

B. It accepts a single key-value pairs as input and can emit only one key-value pair as output.

C. It accepts a list key-value pairs as input and can emit only one key-value pair as output.

D. It accepts a single key-value pairs as input and can emit any number of key-value pair as output, including zero.

Correct Answer: D

Explanation: public class Mapper extends Object Maps input key/value pairs to a set of intermediate key/value pairs.

Maps are the individual tasks which transform input records into a intermediate records. The transformed intermediate records need not be of the same type as the input records. A given input pair may map to zero or many output pairs. Reference: org.apache.hadoop.mapreduce

**Class Mapper** 

Latest HADOOP-PR000007	HADOOP-PR000007 PDF
Dumps	Dumps

HADOOP-PR000007

**Braindumps**