

APACHE-HADOOP-DEVELOPER^{Q&As}

Hadoop 2.0 Certification exam for Pig and Hive Developer

Pass Hortonworks APACHE-HADOOP-DEVELOPER Exam with 100% Guarantee

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

https://www.pass4itsure.com/apache-hadoop-developer.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

You have user profile records in your OLPT database, that you want to join with web logs you have already ingested into the Hadoop file system. How will you obtain these user records?

- A. HDFS command
- B. Pig LOAD command
- C. Sqoop import
- D. Hive LOAD DATA command
- E. Ingest with Flume agents
- F. Ingest with Hadoop Streaming

Correct Answer: C

Reference: Hadoop and Pig for Large-Scale Web Log Analysis

QUESTION 2

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 3

Analyze each scenario below and indentify which best describes the behavior of the default partitioner?

A. The default partitioner assigns key-values pairs to reduces based on an internal random number generator.

B. The default partitioner implements a round-robin strategy, shuffling the key-value pairs to each reducer in turn. This ensures an event partition of the key space.

C. The default partitioner computes the hash of the key. Hash values between specific ranges are associated with different buckets, and each bucket is assigned to a specific reducer.

D. The default partitioner computes the hash of the key and divides that value modulo the number of reducers. The result determines the reducer assigned to process the key-value pair.

E. The default partitioner computes the hash of the value and takes the mod of that value with the number of reducers. The result determines the reducer assigned to process the key-value pair.

Correct Answer: D

Explanation: The default partitioner computes a hash value for the key and assigns the partition based on this result.

The default Partitioner implementation is called HashPartitioner. It uses the hashCode() method of the key objects modulo the number of partitions total to determine which partition to send a given (key, value) pair to.

In Hadoop, the default partitioner is HashPartitioner, which hashes a record\\'s key to determine which partition (and thus which reducer) the record belongs in. The number of partition is then equal to the number of reduce tasks for the job.

Reference: Getting Started With (Customized) Partitioning

QUESTION 4

The Hadoop framework provides a mechanism for coping with machine issues such as faulty configuration or impending hardware failure. MapReduce detects that one or a number of machines are performing poorly and starts more copies of a map or reduce task. All the tasks run simultaneously and the task finish first are used. This is called:

- A. Combine
- B. IdentityMapper
- C. IdentityReducer
- D. Default Partitioner
- E. Speculative Execution

Correct Answer: E

Explanation: Speculative execution: One problem with the Hadoop system is that by dividing the tasks across many nodes, it is possible for a few slow nodes to rate-limit the rest of the program. For example if one node has a slow disk controller, then it may be reading its input at only 10% the speed of all the other nodes. So when 99 map tasks are already complete, the system is still waiting for the final map task to check in, which takes much longer than all the other



nodes. By forcing tasks to run in isolation from one another, individual tasks do not know where their inputs come from. Tasks trust the Hadoop platform to just deliver the appropriate input. Therefore, the same input can be processed multiple times in parallel, to exploit differences in machine capabilities. As most of the tasks in a job are coming to a close, the Hadoop platform will schedule redundant copies of the remaining tasks across several nodes which do not have other work to perform. This process is known as speculative execution. When tasks complete, they announce this fact to the JobTracker. Whichever copy of a task finishes first becomes the definitive copy. If other copies were executing speculatively, Hadoop tells the TaskTrackers to abandon the tasks and discard their outputs. The Reducers then receive their inputs from whichever Mapper completed successfully, first.

Reference: Apache Hadoop, Module 4: MapReduce

Note:

*

Hadoop uses "speculative execution." The same task may be started on multiple boxes. The first one to finish wins, and the other copies are killed.

*

There are a few reasons Hadoop can kill tasks by his own decisions:

Failed tasks are tasks that error out.

a) Task does not report progress during timeout (default is 10 minutes)

b) FairScheduler or CapacityScheduler needs the slot for some other pool (FairScheduler) or queue (CapacityScheduler).

c) Speculative execution causes results of task not to be needed since it has completed on other place.

Reference: Difference failed tasks vs killed tasks

QUESTION 5

Which YARN component is responsible for monitoring the success or failure of a Container?

A. ResourceManager

- B. ApplicationMaster
- C. NodeManager
- D. JobTracker

Correct Answer: A

Latest APACHE-HADOOP-DEVELOPER Dumps APACHE-HADOOP-DEVELOPER VCE Dumps APACHE-HADOOP-DEVELOPER Study Guide