

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.leads4pass.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

Which one of the following statements describes the relationship between the NodeManager and the ApplicationMaster?

- A. The ApplicationMaster starts the NodeManager in a Container
- B. The NodeManager requests resources from the ApplicationMaster
- C. The ApplicationMaster starts the NodeManager outside of a Container
- D. The NodeManager creates an instance of the ApplicationMaster

Correct Answer: D

QUESTION 2

Which Two of the following statements are true about hdfs? Choose 2 answers

- A. An HDFS file that is larger than dfs.block.size is split into blocks
- B. Blocks are replicated to multiple datanodes
- C. HDFS works best when storing a large number of relatively small files
- D. Block sizes for all files must be the same size

Correct Answer: AB

QUESTION 3

Which project gives you a distributed, Scalable, data store that allows you random, realtime read/write access to hundreds of terabytes of data?

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

Correct Answer: A

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 ruby-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 4

You write MapReduce job to process 100 files in HDFS. Your MapReduce algorithm uses TextInputFormat: the mapper applies a regular expression over input values and emits key- values pairs with the key consisting of the matching text, and the value containing the filename and byte offset. Determine the difference between setting the number of reduces to one and settings the number of reducers to zero.

- A. There is no difference in output between the two settings.
- B. With zero reducers, no reducer runs and the job throws an exception. With one reducer, instances of matching patterns are stored in a single file on HDFS.
- C. With zero reducers, all instances of matching patterns are gathered together in one file on HDFS. With one reducer, instances of matching patterns are stored in multiple files on HDFS.
- D. With zero reducers, instances of matching patterns are stored in multiple files on HDFS. With one reducer, all instances of matching patterns are gathered together in one file on HDFS.

Correct Answer: D

Explanation: * It is legal to set the number of reduce-tasks to zero if no reduction is desired.

In this case the outputs of the map-tasks go directly to the FileSystem, into the output path set by `setOutputPath(Path)`. The framework does not sort the map-outputs before writing them out to the FileSystem.

* Often, you may want to process input data using a map function only. To do this, simply set `mapreduce.job.reduces` to zero. The MapReduce framework will not create any reducer tasks. Rather, the outputs of the mapper tasks will be the final output of the job.

Note:

Reduce

In this phase the `reduce(WritableComparable, Iterator, OutputCollector, Reporter)` method is called for each pair in the grouped inputs.

The output of the reduce task is typically written to the `FileSystem` via `OutputCollector.collect(WritableComparable, Writable)`.

Applications can use the `Reporter` to report progress, set application-level status messages and update Counters, or just indicate that they are alive.

The output of the Reducer is not sorted.

QUESTION 5

You want to populate an associative array in order to perform a map-side join. You've decided to put this information in a text file, place that file into the `DistributedCache` and read it in your Mapper before any records are processed.

Identify which method in the Mapper you should use to implement code for reading the file and populating the associative array?

- A. combine
- B. map
- C. init
- D. configure

Correct Answer: D

Reference: `org.apache.hadoop.filecache` , Class `DistributedCache`

QUESTION 6

Which one of the following statements regarding the components of YARN is FALSE?

- A. A Container executes a specific task as assigned by the `ApplicationMaster`
- B. The `ResourceManager` is responsible for scheduling and allocating resources
- C. A client application submits a YARN job to the `ResourceManager`
- D. The `ResourceManager` monitors and restarts any failed Containers

Correct Answer: D

QUESTION 7

You've written a MapReduce job that will process 500 million input records and generated 500 million key-value pairs. The data is not uniformly distributed. Your MapReduce job will create a significant amount of intermediate data that it needs to transfer between mappers and reduces which is a potential bottleneck. A custom implementation of which interface is most likely to reduce the amount of intermediate data transferred across the network?

- A. Partitioner
- B. OutputFormat
- C. WritableComparable
- D. Writable
- E. InputFormat
- F. Combiner

Correct Answer: F

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.

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

QUESTION 8

For each input key-value pair, mappers can emit:

- A. As many intermediate key-value pairs as designed. There are no restrictions on the types of those key-value pairs (i.e., they can be heterogeneous).
- B. As many intermediate key-value pairs as designed, but they cannot be of the same type as the input key-value pair.
- C. One intermediate key-value pair, of a different type.
- D. One intermediate key-value pair, but of the same type.
- E. As many intermediate key-value pairs as designed, as long as all the keys have the same types and all the values have the same type.

Correct Answer: E

Explanation: Mapper maps input key/value pairs to a set of intermediate key/value pairs.

Maps are the individual tasks that transform input records into intermediate records. The transformed intermediate records do not need to be of the same type as the input records. A given input pair may map to zero or many output

pairs.

Reference: Hadoop Map-Reduce Tutorial

QUESTION 9

You have just executed a MapReduce job. Where is intermediate data written to after being emitted from the Mapper's map method?

- A. Intermediate data is streamed across the network from Mapper to the Reduce and is never written to disk.
- B. Into in-memory buffers on the TaskTracker node running the Mapper that spill over and are written into HDFS.
- C. Into in-memory buffers that spill over to the local file system of the TaskTracker node running the Mapper.
- D. Into in-memory buffers that spill over to the local file system (outside HDFS) of the TaskTracker node running the Reducer
- E. Into in-memory buffers on the TaskTracker node running the Reducer that spill over and are written into HDFS.

Correct Answer: C

Explanation: The mapper output (intermediate data) is stored on the Local file system (NOT HDFS) of each individual mapper nodes. This is typically a temporary directory location which can be setup in config by the hadoop administrator. The intermediate data is cleaned up after the Hadoop Job completes.

Reference: 24 Interview Questions and Answers for Hadoop MapReduce developers, Where is the Mapper Output (intermediate key-value data) stored ?

QUESTION 10

Which TWO of the following statements are true regarding Hive? Choose 2 answers A. Useful for data analysts familiar with SQL who need to do ad-hoc queries

- B. Offers real-time queries and row level updates
- C. Allows you to define a structure for your unstructured Big Data
- D. Is a relational database

Correct Answer: AC

QUESTION 11

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

- A. The default partitioner assigns key-values pairs to reducers 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 even 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 partitions is then equal to the number of reduce tasks for the job.

Reference: Getting Started With (Customized) Partitioning

QUESTION 12

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 13

You want to run Hadoop jobs on your development workstation for testing before you submit them to your production cluster. Which mode of operation in Hadoop allows you to most closely simulate a production cluster while using a single machine?

- A. Run all the nodes in your production cluster as virtual machines on your development workstation.
- B. Run the hadoop command with the `-jt local` and the `-fs file:///options`.
- C. Run the DataNode, TaskTracker, NameNode and JobTracker daemons on a single machine.
- D. Run simldoop, the Apache open-source software for simulating Hadoop clusters.

Correct Answer: C

QUESTION 14

Which one of the following statements describes a Pig bag, tuple, and map, respectively?

- A. Unordered collection of maps, ordered collection of tuples, ordered set of key/value pairs
- B. Unordered collection of tuples, ordered set of fields, set of key value pairs
- C. Ordered set of fields, ordered collection of tuples, ordered collection of maps
- D. Ordered collection of maps, ordered collection of bags, and unordered set of key/value pairs

Correct Answer: B

QUESTION 15

Review the following data and Pig code.

M,38,95111

F,29,95060

F,45,95192

M,62,95102

F,56,95102

```
A = LOAD andapos;dataandapos; USING PigStorage(andapos;.andapos;) as (gender:Chararray, age:int,  
zlp:chararray);
```

```
B = FOREACH A GENERATE age;
```

Which one of the following commands would save the results of B to a folder in hdfs named myoutput?

- A. STORE A INTO andapos;myoutputandapos; USING PigStorage(andapos;;andapos;);
- B. DUMP B using PigStorage(andapos;myoutputandapos;);
- C. STORE B INTO andapos;myoutputandapos;;
- D. DUMP B INTO andapos;myoutputandapos;;

Correct Answer: C

[Latest APACHE-HADOOP-DEVELOPER Dumps](#)

[APACHE-HADOOP-DEVELOPER PDF Dumps](#)

[APACHE-HADOOP-DEVELOPER Study Guide](#)