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.passapply.com/ccd-410.html

100% Passing Guarantee
100% Money Back Assurance

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

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

QUESTION 1

Which describes how a client reads a file from HDFS?

A. The client queries the NameNode for the block location(s). The NameNode returns the block location

(s) to the client. The client reads the data directory off the DataNode(s).

B. The client queries all DataNodes in parallel. The DataNode that contains the requested data responds directly to the client. The client reads the data directly off the DataNode.

C. The client contacts the NameNode for the block location(s). The NameNode then queries the DataNodes for block locations. The DataNodes respond to the NameNode, and the NameNode redirects the client to the DataNode that holds the requested data block(s). The client then reads the data directly off the DataNode.

D. The client contacts the NameNode for the block location(s). The NameNode contacts the DataNode that holds the requested data block. Data is transferred from the DataNode to the NameNode, and then from the NameNode to the client.

Correct Answer: A

Reference: 24 Interview Questions and Answers for Hadoop MapReduce developers, How the Client communicates with HDFS?

QUESTION 2

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 simldooop, the Apache open-source software for simulating Hadoop clusters.

Correct Answer: C

QUESTION 3

MapReduce v2 (MRv2/YARN) is designed to address which two issues?

A. Single point of failure in the NameNode.

B. Resource pressure on the JobTracker.

C. HDFS latency.

D. Ability to run frameworks other than MapReduce, such as MPI.

E. Reduce complexity of the MapReduce APIs.

F. Standardize on a single MapReduce API.

Correct Answer: BD

YARN (Yet Another Resource Negotiator), as an aspect of Hadoop, has two major kinds of benefits:

*

(D) The ability to use programming frameworks other than MapReduce. / MPI (Message Passing Interface) was mentioned as a paradigmatic example of a MapReduce alternative

*

Scalability, no matter what programming framework you use. Note:

*

The fundamental idea of MRv2 is to split up the two major functionalities of the JobTracker, resource management and job scheduling/monitoring, into separate daemons. The idea is to have a global ResourceManager (RM) and per-application ApplicationMaster (AM). An application is either a single job in the classical sense of Map-Reduce jobs or a DAG of jobs.

*

(B) The central goal of YARN is to clearly separate two things that are unfortunately smushed together in current Hadoop, specifically in (mainly) JobTracker:

/ Monitoring the status of the cluster with respect to which nodes have which resources available. Under YARN, this will be global. / Managing the parallelization execution of any specific job. Under YARN, this will be done separately for each job. The current Hadoop MapReduce system is fairly scalable -- Yahoo runs 5000 Hadoop jobs, truly concurrently, on a single cluster, for a total 1.5 2 millions jobs/cluster/month. Still, YARN will remove scalability bottlenecks

Reference: Apache Hadoop YARN Concepts and Applications

QUESTION 4

You have written a Mapper which invokes the following five calls to the OutputColletor.collect method:

output.collect (new Text ("Apple"), new Text ("Red")) ;

output.collect (new Text ("Banana"), new Text ("Yellow")) ; output.collect (new Text ("Apple"), new Text

("Yellow")) ; output.collect (new Text ("Cherry"), new Text ("Red")) ;

output.collect (new Text ("Apple"), new Text ("Green")) ;

How many times will the Reducer\\'s reduce method be invoked?

A. 6

B. 3

C. 1

D. 0

E. 5

Correct Answer: B

reduce() gets called once for each [key, (list of values)] pair. To explain, let\\'s say you called:

out.collect(new Text("Car"),new Text("Subaru");

out.collect(new Text("Car"),new Text("Honda");

out.collect(new Text("Car"),new Text("Ford");

out.collect(new Text("Truck"),new Text("Dodge");

out.collect(new Text("Truck"),new Text("Chevy");

Then reduce() would be called twice with the pairs

reduce(Car,)

reduce(Truck,)

Reference: Mapper output.collect()?

QUESTION 5

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

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.

Failed tasks are tasks that error out.

*

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

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