

KillTest

Higher Quality, Better Service!



Q&A

<http://www.killtest.com>

We offer free update service for one year.

Exam : **CCA-500**

Title : Cloudera Certified
Administrator for Apache
Hadoop (CCAHA)

Version : DEMO

1. Your cluster's `mapred-start.xml` includes the following parameters

```
<name>mapreduce.map.memory.mb</name>
```

```
<value>4096</value>
```

```
<name>mapreduce.reduce.memory.mb</name>
```

```
<value>8192</value>
```

And any cluster's `yarn-site.xml` includes the following parameters

```
<name>yarn.nodemanager.vmen-pmen-ration</name>
```

```
<value>2.1</value>
```

What is the maximum amount of virtual memory allocated for each map task before YARN will kill its Container?

- A. 4 GB
- B. 17.2 GB
- C. 8.9GB
- D. 8.2 GB
- E. 24.6 GB

Answer: D

2. Assuming you're not running HDFS Federation, what is the maximum number of NameNode daemons you should run on your cluster in order to avoid a "split-brain" scenario with your NameNode when running HDFS High Availability (HA) using Quorum-based storage?

- A. Two active NameNodes and two Standby NameNodes
- B. One active NameNode and one Standby NameNode
- C. Two active NameNodes and one Standby NameNode
- D. Unlimited. HDFS High Availability (HA) is designed to overcome limitations on the number of NameNodes you can deploy

Answer: B

3. Table schemas in Hive are:

- A. Stored as metadata on the NameNode
- B. Stored along with the data in HDFS
- C. Stored in the Metadata
- D. Stored in ZooKeeper

Answer: B

4. For each YARN job, the Hadoop framework generates task log file. Where are Hadoop task log files stored?

- A. Cached by the NodeManager managing the job containers, then written to a log directory on the NameNode
- B. Cached in the YARN container running the task, then copied into HDFS on job completion
- C. In HDFS, in the directory of the user who generates the job
- D. On the local disk of the slave node running the task

Answer: D

5. You have a cluster running with the fair Scheduler enabled. There are currently no jobs running on the

cluster, and you submit a job A, so that only job A is running on the cluster. A while later, you submit Job B. now Job A and Job B are running on the cluster at the same time. How will the Fair Scheduler handle these two jobs?

- A. When Job B gets submitted, it will get assigned tasks, while job A continues to run with fewer tasks.
- B. When Job B gets submitted, Job A has to finish first, before job B can gets scheduled.
- C. When Job A gets submitted, it doesn't consumes all the task slots.
- D. When Job A gets submitted, it consumes all the task slots.

Answer: B