

ISLEVER

CCB-400

Cloudera Certified Specialist in Apache HBase

DEMO

<https://www.islever.com/ccb-400.html>

<https://www.islever.com/cloudera.html>

For the most up-to-date exam questions and materials, we recommend visiting our website, where you can access the latest content and resources.

QUESTION NO: 1

You have one primary HMaster and one standby. Your primary HMaster Falls fails and your client application needs to make a metadata change. Which of the following is the effect on your client application?

- A. The client will query ZooKeeper to find the location of the new HMaster and complete the metadata change.
- B. The client will make the metadata change regardless of the state of the HMaster.
- C. The new HMaster will notify the client and complete the metadata change.
- D. The client application will fail with a runtime error.

Answer: A

Explanation: the HBase master publishes its location to clients via Zookeeper. This is done to support multimaster operation (failover). So if the HBase master self-discovers its location as a localhost address, then it will publish that. Region servers or clients which go to Zookeeper for the master location will get back an address in that case only useful if they happen to be co-located with the master.

Note:

* HMaster is the implementation of the Master Server. The Master server is responsible for monitoring all RegionServer instances in the cluster, and is the interface for all metadata changes.

QUESTION NO: 2

You have an average key-value pair size of 100 bytes. Your primary access is random needs on the table. Which of the following actions will speed up random reading performance on your cluster?

- A. Turn off WAL on puts
- B. Increase the number of versions kept
- C. Decrease the block size
- D. Increase the block size

Answer: C

Explanation: Larger block size is preferred if files are primarily for sequential access. Smaller blocks are good for random access, but require more memory to hold the block index, and may be slower to create

Reference: Could I improve HBase performance by reducing the hdfs block size?

QUESTION NO: 3

The cells in a given row have versions that range from 1000 to 2000. You execute a delete specifying the value 3000 for the version. What is the outcome?

- A. The delete fails with an error.
- B. Only cells equal to the Specified version are deleted.
- C. The entire row is deleted.
- D. Nothing in the row is deleted.

Answer: C

Explanation: When performing a delete operation in HBase, there are two ways to specify the versions to be deleted

Delete all versions older than a certain timestamp

Delete the version at a specific timestamp

A delete can apply to a complete row, a complete column family, or to just one column. It is only in the last case that you can delete explicit versions. For the deletion of a row or all the columns within a family, it always works by deleting all cells older than a certain version.

Deletes work by creating tombstone markers. For example, let's suppose we want to delete a row. For this you can specify a version, or else by default the `currentTimeMillis` is used. What this means is "delete all cells where the version is less than or equal to this version". HBase never modifies data in place, so for example a delete will not immediately delete (or mark as deleted) the entries in the storage file that correspond to the delete condition. Rather, a so-called tombstone is written, which will mask the deleted values[17]. If the version you specified when deleting a row is larger than the version of any value in the row, then you can consider the complete row to be deleted.

Reference: Apache HBase, Delete

<http://archive.cloudera.com/cdh4/cdh/4/hbase/book.html#delete>(scroll below and see 5.8.1.5. Delete topic, read the last paragraph)

QUESTION NO: 4

You have an "Employees" table in HBase. The Row Keys are the employees' IDs. You would like to retrieve all employees who have an employee ID between 'user_100' and 'user_110'. The shell command you would use to complete this is:

- A. scan 'Employees', {STARTROW => 'user_100', STOPROW => 'user_111'}
- B. get 'Employees', {STARTROW => 'user_100', STOPROW => 'user_110'}
- C. scan 'Employees', {STARTROW => 'user_100', SLIMIT => 10}
- D. scan 'Employees', {STARTROW => 'user_100', STOPROW => 'user_110'}

Answer: D

Explanation: public Scan(byte[] startRow,
byte[] stopRow)

Create a Scan operation for the range of rows specified.

Parameters:

startRow - row to start scanner at or after (inclusive)

stopRow - row to stop scanner before (exclusive)

Reference: [org.apache.hadoop.hbase.client, Class Scan](#)

QUESTION NO: 5

Under default settings, which feature of HBase ensures that data won't be lost in the event of a RegionServer failure?

- A. All HBase activity is written to the WAL, which is stored in HDFS
- B. All operations are logged on the HMaster.
- C. HBase is ACID compliant, which guarantees that it is Durable.
- D. Data is stored on the local filesystem of the RegionServer.

Answer: A

Explanation: HBase data updates are stored in a place in memory called memstore for fast write. In the event of a region server failure, the contents of the memstore are lost because they have not been saved to disk yet. To prevent data loss in such a scenario, the updates are persisted in a WAL file before they are stored in the memstore. In the event of a region server failure, the lost contents in the memstore can be regenerated by replaying the updates (also called edits) from the WAL file.

Reference: [HBase Log Splitting](#)