

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.pass2lead.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

You want to understand more about how users browse your public website, such as which pages they visit prior to placing an order. You have a farm of 200 web servers hosting your website.

How will you gather this data for your analysis?

- A. Ingest the server web logs into HDFS using Flume.
- B. Write a MapReduce job, with the web servers for mappers, and the Hadoop cluster nodes for reduces.
- C. Import all users' clicks from your OLTP databases into Hadoop, using Sqoop.
- D. Channel these clickstreams into Hadoop using Hadoop Streaming.
- E. Sample the weblogs from the web servers, copying them into Hadoop using curl.

Correct Answer: A

QUESTION 2

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

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 jruby-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 3

What data does a Reducer reduce method process?

- A. All the data in a single input file.
- B. All data produced by a single mapper.
- C. All data for a given key, regardless of which mapper(s) produced it.
- D. All data for a given value, regardless of which mapper(s) produced it.

Correct Answer: C

Reducing lets you aggregate values together. A reducer function receives an iterator of input values from an input list. It then combines these values together, returning a single output value.

All values with the same key are presented to a single reduce task.

Reference: Yahoo! Hadoop Tutorial, Module 4: MapReduce

QUESTION 4

Can you use MapReduce to perform a relational join on two large tables sharing a key? Assume that the two tables are formatted as comma-separated files in HDFS.

- A. Yes.
- B. Yes, but only if one of the tables fits into memory

- C. Yes, so long as both tables fit into memory.
- D. No, MapReduce cannot perform relational operations.
- E. No, but it can be done with either Pig or Hive.

Correct Answer: A

Note:

*

Join Algorithms in MapReduce A) Reduce-side join B) Map-side join C) In-memory join / Striped Striped variant variant / Memcached variant

*

Which join to use? / In-memory join > map-side join > reduce-side join / Limitations of each? In-memory join: memory
Map-side join: sort order and partitioning

Reduce-side join: general purpose

QUESTION 5

Your cluster's HDFS block size is 64MB. You have a directory containing 100 plain text files, each of which is 100MB in size. The InputFormat for your job is TextInputFormat. Determine how many Mappers will run?

- A. 64
- B. 100
- C. 200
- D. 640

Correct Answer: C

Each file would be split into two as the block size (64 MB) is less than the file size (100 MB), so 200 mappers would be running.

Note:

If you're not compressing the files then Hadoop will process your large files (say 10G), with a number of mappers related to the block size of the file.

Say your block size is 64M, then you will have ~160 mappers processing this 10G file ($160 * 64 \approx 10G$). Depending on how CPU intensive your mapper logic is, this might be an acceptable block size, but if you find that your mappers are executing in sub-minute times, then you might want to increase the work done by each mapper (by increasing the block size to 128, 256, 512M - the actual size depends on how you intend to process the data).

Reference: <http://stackoverflow.com/questions/11014493/hadoop-mapreduce-appropriate-input-files-size> (first answer, second paragraph)

[CCD-410 Study Guide](#)

[CCD-410 Exam Questions](#)

[CCD-410 Braindumps](#)