

QUESTIONS & ANSWERS

Kill your exam at first Attempt



Cloudera

CCA-410

*Cloudera Certified Administrator for Apache Hadoop
(CCA410)*

Reference:

CONFIGURATION PARAMETERS DFS.BLOCK.SIZE

QUESTION: 91

Which daemons instantiates JVM's to perform mapreduce processing in a cluster running mapreduce v1 (MRv1)?

- A. Nodemanager
- B. applicationmanager
- C. Applicationmaster
- D. Tasktracker
- E. Jobtracker
- F. Datanode
- G. Namenode
- H. Resourcemanager

Answer: D

Explanation:

A TaskTracker is a slave node daemon in the cluster that accepts tasks (Map, Reduce and Shuffle operations) from a JobTracker. There is only One Task Tracker process run on any hadoop slave node. Task Tracker runs on its own JVM process. Every TaskTracker is configured with a set of slots, these indicate the number of tasks that it can accept. The TaskTracker starts a separate JVM processes to do the actual work (called as Task Instance) this is to ensure that process failure does not take down the task tracker.

QUESTION: 92

You set `mapred.tasktracker.reduce.tasks.maximum` to a value of 4 on a cluster running mapreduce v1 (MRv1). How many reducers will run for any given job?

- A. Maximum of 4 reducers, but the actual number of reducers that run for any given job is based on the volume of input data
- B. a maximum of 4 reducer's, but the actual number of reducer's that run for any given job is based on the volume intermediate data.
- C. Four reducer's will run. Once set by the cluster administrator, this parameter can't be overridden
- D. The number of reducer's for any given job is set by the developer

Answer: B

Explanation:

mapred.tasktracker.reduce.tasks.maximum The maximum number of reduce tasks that will be run simultaneously by a task tracker. A reducer uses intermediate data as input.

Note:

* The mapper output (intermediate data) is stored on the Local file system (NOT HDFS) of each individual mapper nodes.

* Reducers start copying intermediate key-value pairs from the mappers as soon as they are available. The progress calculation also takes in account the processing of data transfer which is done by reduce process, therefore the reduce progress starts showing up as soon as any intermediate key-value pair for a mapper is available to be transferred to reducer.

QUESTION: 93

You configure your hadoop development cluster with both mapreduce frameworks, mapreduceV1 (MRv1) and mapreduce v2 (MRv2/YARN). You plan to run only one set of mapreduce daemons at a time in this development environment (running both simultaneously results in an unsuitable cluster but configure in both moving between them is fine). Which two mapreduce (capitation) daemons do you need to configure to run on your masternodes?

- A. Containermanager
- B. Nodemanager
- C. Journalnode
- D. Applicationmaster
- E. Resourcemanager
- F. Jobtracker

Answer: E, F

Explanation:

Jobtracker for mapreduce V1 (MRv1) and ResourceManager for mapreduce v2 (MRv2/YARN).

* Configure YARN daemons

If you have decided to run YARN, you must configure the following services: ResourceManager (on a dedicated host) and NodeManager (on every host where you plan to run MapReduce v2 jobs).

* JobTracker is the daemon service for submitting and tracking MapReduce V1 (MRv1) jobs in Hadoop.

Note:

* Make sure you are not trying to run MRv1 and YARN on the same set of nodes at the same time. This is not supported; it will degrade performance and may result in an unstable cluster deployment.

* MapReduce has undergone a complete overhaul and CDH4 now includes MapReduce 2.0 (MRv2). The fundamental idea of MRv2's YARN architecture is to split up the two primary responsibilities of the JobTracker — resource management and job scheduling/monitoring — into separate daemons: a global ResourceManager (RM) and per-application ApplicationMasters (AM).

Incorrect:

Not C: JournalNode is for Quorum implementation.

QUESTION: 94

Using hadoop's default settings, how much data will be able to store on your hadoop cluster if it has 12 nodes with 4TB raw disk space per node allocated to HDFS storage?

- A. Approximately 3TB
- B. Approximately 12TB
- C. Approximately 16TB
- D. Approximately 48TB

Answer: C

Explanation:

Default replication value is 3. The following amount of data can be stored: $12 \times 4 / 3 = 16$ TB

QUESTION: 95

When requesting a file, how does HDFS retrieve the blocks associated with that file?

- A. the client polls the datanodes for the block ID's
- B. the namenode requires the datanode for the block ID's
- C. The namenode reads the block ID's from memory
- D. the namenode reads the block ID's from disk

Answer: D

Explanation:

Here is how a client RPC request to the Hadoop HDFS NameNode flows through the NameNode. The Hadoop NameNode receives requests from HDFS clients in the form of Hadoop RPC requests over a TCP connection. Typical client requests include mkdir, getBlockLocations, create file, etc. Remember – HDFS separates metadata from actual file data, and that the NameNode is the metadata server. Hence, these requests are pure metadata requests – no data transfer is involved.

QUESTION: 96

What four functions do scheduling algorithms perform on hadoop cluster?

- A. Run jobs at periodic times of the day
- B. Reduce the job latencies in environment with multiple jobs of different sizes
- C. Allow multiple users to share clusters in a predictable policy-guided manner
- D. support the implementation of service-level agreements for multiple cluster users
- E. allow short jobs to complete even when large, long jobs (consuming a lot of resources are running)
- F. Reduce the total amount of computation necessary to complete a job.
- G. Ensure data locality by ordering map tasks so that they run on data local maps slots

Answer: A, B, E, F

Explanation:

Hadoop schedulers Since the pluggable scheduler was implemented, several scheduler algorithms have been developed for it. / FIFO scheduler The original scheduling algorithm that was integrated within the JobTracker was called FIFO. In FIFO scheduling, a JobTracker pulled jobs from a work queue, oldest job first. This schedule had no concept of the priority or size of the job, but the approach was simple to implement and efficient. / Fair scheduler The core idea behind the fair share scheduler was to assign resources to jobs such that on average over time, each job gets an equal share of the available resources. The result is that jobs that require less time are able to access the CPU and finish intermixed with the execution of jobs that require more time to execute. This behavior allows for some interactivity among Hadoop jobs and permits greater responsiveness of the Hadoop cluster to the variety of job types submitted. The fair scheduler was developed by Facebook. / Capacity scheduler The capacity scheduler shares some of the principles of the fair scheduler but has distinct differences, too. First, capacity scheduling was defined for large clusters, which may have multiple, independent consumers and target applications.

For this reason, capacity scheduling provides greater control as well as the ability to provide a minimum capacity guarantee and share excess capacity among users. The capacity scheduler was developed by Yahoo!. In capacity scheduling, instead of pools, several queues are created, each with a configurable number of map and reduce slots. Each queue is also assigned a guaranteed capacity (where the overall capacity of the cluster is the sum of each queue's capacity).

* The introduction of the pluggable scheduler was yet another evolution in cluster computing with Hadoop. The pluggable scheduler permits the use (and development) of schedulers optimized for the particular workload and application. The new schedulers have also made it possible to create multi-user data warehouses with Hadoop, given the ability to share the overall Hadoop infrastructure with multiple users and organizations.

QUESTION: 97

What happens when a map task crashes while running a mapreduce job?

- A. The job immediately fails
- B. The tasktracker closes the JVM instance and restarts
- C. The jobtracker attempts to re-run the task on the same node
- D. the jobtracket attempts to re-run the task on a different node

Answer: D

Explanation:

When the jobtracker is notified of a task attempt that has failed (by the tasktracker's heartbeat call), it will reschedule execution of the task. The jobtracker will try to avoid rescheduling the task on a tasktracker where it has previously failed.

For More exams visit <http://killexams.com> -



KILLEXAMS.COM

Kill your exam at First Attempt....Guaranteed!