Vendor: Cloudera

Exam Code: CCA-410

Exam Name: Cloudera Certified Administrator for Apache Hadoop (CCAH) Exam

Version: DEMO

QUESTION 1
You observe that the number of spilled records from map tasks for exceeds the number of map output records. You child heap size is 1 GB and your io.sort.mb value is set to 100MB. How would you tune your io.sort.mb value to achieve maximum memory to disk I/O ratio?

A. Tune io.sort.mb value until you observe that the number of spilled records equals (or is as close to equals) the number of map output records.
B. Decrease the io.sort.mb value below 100MB.
C. Increase the IO.sort.mb as high you can, as close to 1GB as possible.
D. For 1GB child heap size an io.sort.mb of 128MB will always maximum memory to disk I/O.

Answer: A
Explanation:
Here are a few tradeoffs to consider.
1. the number of seeks being done when merging files. If you increase the merge factor too high, then the seek cost on disk will exceed the savings from doing a parallel merge (note that OS cache might mitigate this somewhat).
2. Increasing the sort factor decreases the amount of data in each partition. I believe the number is io.sort.mb / io.sort.factor for each partition of sorted data. I believe the general rule of thumb is to have io.sort.mb = 10 * io.sort.factor (this is based on the seek latency of the disk on the transfer speed, I believe. I'm sure this could be tuned better if it was your bottleneck. If you keep these in line with each other, then the seek overhead from merging should be minimized
3. you increase io.sort.mb, then you increase memory pressure on the cluster, leaving less memory available for job tasks. Memory usage for sorting is mapper tasks * io.sort.mb -- so you could find yourself causing extra GCs if this is too high
Essentially,
If you find yourself swapping heavily, then there's a good chance you have set the sort factor too high.
If the ratio between io.sort.mb and io.sort.factor isn't correct, then you may need to change io.sort.mb (if you have the memory) or lower the sort factor. If you find that you are spending more time in your mappers than in your reducers, then you may want to increase the number of map tasks and decrease the sort factor (assuming there is memory pressure).
Reference: How could I tell if my hadoop config parameter io.sort.factor is too small or too big?
http://stackoverflow.com/questions/8642566/how-could-i-tell-if-my-hadoop-config-parameter-io-sort-factor-is-too-small-or-to

QUESTION 2
Your Hadoop cluster has 25 nodes with a total of 100 TB (4 TB per node) of raw disk space allocated HDFS storage. Assuming Hadoop's default configuration, how much data will you be able to store?

A. Approximately 100TB
B. Approximately 25TB
C. Approximately 10TB
D. Approximately 33 TB

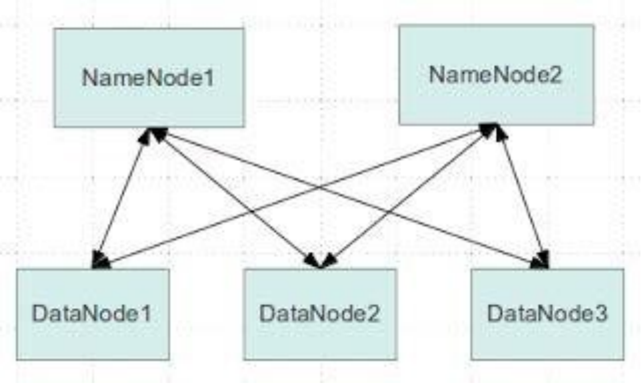
Answer: D
Explanation:
In default configuration there are total 3 copies of a datablock on HDFS, 2 copies are stored on datanodes on same rack and 3rd copy on a different rack. Reference: 24 Interview Questions & Answers for Hadoop MapReduce developers, How the HDFS Blocks are replicated?

QUESTION 3

You set up the Hadoop cluster using NameNode Federation. One NameNode manages the/users namespace and one NameNode manages the/data namespace. What happens when client tries to write a file to/reports/myreport.txt?

A. The file successfully writes to /users/reports/myreports/myreport.txt.
B. The client throws an exception.
C. The file successfully writes to /report/myreport.txt. The metadata for the file is managed by the first NameNode to which the client connects.
D. The file writes fails silently; no file is written, no error is reported.

Answer: C
Explanation:
Note:
* The current HDFS architecture allows only a single namespace for the entire cluster. A single Namenode manages this namespace. HDFS Federation addresses limitation of current architecture by adding support multiple Namenodes/namespaces to HDFS file system.
* HDFS Federation enables multiple NameNodes in a cluster for horizontal scalability of NameNode. All these NameNodes work independently and don't require any co-ordination. A DataNode can register with multiple NameNodes in the cluster and can store the data blocks for multiple NameNodes.



QUESTION 4
Identify two features/issues that MapReduce v2 (MRv2/YARN) is designed to address:

A. Resource pressure on the JobTrackr
B. HDFS latency.
C. Ability to run frameworks other than MapReduce, such as MPI.
D. Reduce complexity of the MapReduce APIs.
E. Single point of failure in the NameNode.
F. Standardize on a single MapReduce API.

Answer: AC
Explanation:
A: MapReduce has undergone a complete overhaul in hadoop-0.23 and we now have, what we call, MapReduce 2.0 (MRv2) or YARN.
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. The ResourceManager and per-node slave, the NodeManager (NM), form the data-computation

framework. The ResourceManager is the ultimate authority that arbitrates resources among all the applications in the system.
The per-application ApplicationMaster is, in effect, a framework specific library and is tasked with negotiating resources from the ResourceManager and working with the NodeManager(s) to execute and monitor the tasks.
C: YARN, as an aspect of Hadoop, has two major kinds of benefits:
The ability to use programming frameworks other than MapReduce. Scalability, no matter what programming framework you use.

QUESTION 5
The most important consideration for slave nodes in a Hadoop cluster running production jobs that require short turnaround times is:

A. The ratio between the amount of memory and the number of disk drives.
B. The ratio between the amount of memory and the total storage capacity.
C. The ratio between the number of processor cores and the amount of memory.
D. The ratio between the number of processor cores and total storage capacity.
E. The ratio between the number of processor cores and number of disk drives.

Answer: D

QUESTION 6
The failure of which daemon makes HDFS unavailable on a cluster running MapReduce v1 (MRv1)?

A. Node Manager
B. Application Manager
C. Resource Manager
D. Secondary NameNode
E. NameNode
F. DataNode

Answer: E
Explanation:
The NameNode is the centerpiece of an HDFS file system. It keeps the directory tree of all files in the file system, and tracks where across the cluster the file data is kept. It does not store the data of these files itself. There is only One NameNode process run on any hadoop cluster. NameNode runs on its own JVM process. In a typical production cluster its run on a separate machine. The NameNode is a Single Point of Failure for the HDFS Cluster. When the NameNode goes down, the file system goes offline.
Reference: 24 Interview Questions & Answers for Hadoop MapReduce developers, What is a NameNode? How many instances of NameNode run on a Hadoop Cluster?

QUESTION 7
Choose three reasons why should you run the HDFS balancer periodically?

A. To improve data locality for MapReduce tasks.
B. To ensure that there is consistent disk utilization across the DataNodes.
C. To ensure that there is capacity in HDFS tor additional data.
D. To ensure that all blocks in the cluster are 128MB in size.

E. To help HDFS deliver consistent performance under heavy loads.

Answer: ABE
Explanation:
The balancer is a tool that balances disk space usage on an HDFS cluster when some datanodes become full or when new empty nodes join the cluster. The tool is deployed as an application program that can be run by the cluster administrator on a live HDFS cluster while applications adding and deleting files.
DESCRIPTION
The threshold parameter is a fraction in the range of (0%, 100%) with a default value of 10%. The threshold sets a target for whether the cluster is balanced. A cluster is balanced if for each datanode, the utilization of the node (ratio of used space at the node to total capacity of the node) differs from the utilization of the (ratio of used space in the cluster to total capacity of the cluster) by no more than the threshold value. The smaller the threshold, the more balanced a cluster will become. It takes more time to run the balancer for small threshold values. Also for a very small threshold the cluster may not be able to reach the balanced state when applications write and delete files concurrently.
The tool moves blocks from highly utilized datanodes to poorly utilized datanodes iteratively. In each iteration a datanode moves or receives no more than the lesser of 10G bytes or the threshold fraction of its capacity. Each iteration runs no more than 20 minutes. At the end of each iteration, the balancer obtains updated datanodes information from the namenode. Reference: org.apache.hadoop.hdfs.server.balancer, Class Balancer

QUESTION 8
What additional capability does Ganglia provide to monitor a Hadoop?

A. Ability to monitor the amount of free space on HDFS.
B. Ability to monitor number of files in HDFS.
C. Ability to monitor processor utilization.
D. Ability to monitor free task slots.
E. Ability to monitor NameNode memory usage.

Answer: E
Explanation:
Ganglia itself collects metrics, such as CPU and memory usage; by using GangliaContext, you can inject Hadoop metrics into Ganglia.
Note:
Ganglia is an open-source, scalable and distributed monitoring system for large clusters. It collects, aggregates and provides time-series views of tens of machine-related metrics such as CPU, memory, storage, network usage.
Ganglia is also a popular solution for monitoring Hadoop and HBase clusters, since Hadoop (and HBase) has built-in support for publishing its metrics to Ganglia. With Ganglia you may easily see the number of bytes written by a particular HDSF datanode over time, the block cache hit ratio for a given HBase region server, the total number of requests to the HBase cluster, time spent in garbage collection and many, many others.
Hadoop and HBase use GangliaContext class to send the metrics collected by each daemon (such as datanode, tasktracker, jobtracker, HMaster etc) to gmonds.