Question No : 1

You configure Hadoop cluster with both MapReduce frameworks, MapReduce v1 (MRv1) and MapReduce v2 (MRv2/YARN). Which two MapReduce (computational) daemons do you need to configure to run on your master nodes?

A. JobTracker
B. ResourceManager
C. ApplicationMaster
D. JournalNode
E. NodeManager

Answer: B,C

Explanation: 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.

Note:
* MRV2 maintains API compatibility with previous stable release (hadoop-0.20.205). This means that all Map-Reduce jobs should still run unchanged on top of MRv2 with just a recompile.

Reference: Apache Hadoop NextGen MapReduce (YARN)

Question No : 2

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: A,B,E

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 No : 3

Assuming a large properly configured multi-rack Hadoop cluster, which scenario should not result in loss of HDFS data assuming the default replication factor settings?

A. Ten percent of DataNodes simultaneously fail.
B. All DataNodes simultaneously fail.
C. An entire rack fails.
D. Multiple racks simultaneously fail.
E. Seventy percent of DataNodes simultaneously fail.

Answer: A

Reference: http://stackoverflow.com/questions/12399197/in-a-large-properly-configured-multi-rack-hadoop-cluster-which-scenarios-will-b

Question No : 4

What happens if a Mapper on one node goes into an infinite loop while running a MapReduce job?

A. After a period of time, the JobTracker will restart the TaskTracker on the node on which the map task is running
B. The Mapper will run indefinitely; the TaskTracker must be restarted to kill it
C. The job will immediately fail.
D. After a period of time, the TaskTracker will kill the Map Task.

Answer: D

Explanation: * The TaskTracker nodes are monitored. If they do not submit heartbeat signals often enough, they are deemed to have failed and the work is scheduled on a different TaskTracker.
* A TaskTracker will notify the JobTracker when a task fails. The JobTracker decides what to do then: it may resubmit the job elsewhere, it may mark that specific record as something to avoid, and it may may even blacklist the TaskTracker as unreliable.

Question No : 5

You have a cluster running with the FIFO scheduler enabled. You submit a large job A to the cluster which you expect to run for one hour. Then, you submit job B to the cluster,

which you expect to run a couple of minutes only. Let's assume both jobs are running at the same priority.

How does the FIFO scheduler execute the jobs? (Choose 3)

A. The order of execution of tasks within a job may vary.
B. When a job is submitted, all tasks belonging to that job are scheduled.
C. Given jobs A and B submitted in that order, all tasks from job A will be scheduled before all tasks from job B.
D. Since job B needs only a few tasks, if might finish before job A completes.

Answer: A,B,C

Reference: http://seriss.com/rush-current/rush/rush-priority.html#FIFO%20Scheduling (see fifo scheduling)

Question No : 6

For each job, the Hadoop framework generates task log files. Where are Hadoop's task log files stored?

A. Cached on the local disk of the slave node running the task, then purged immediately upon task completion.
B. Cached on the local disk of the slave node running the task, then copied into HDFS.
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: C

Explanation: Job Statistics

These logs are created by the jobtracker. The jobtracker runtime statistics from jobs to thesefiles. Those statistics include task attempts, time spent shuffling, input splits given to task attempts, start times of tasks attempts and other information.

The statistics files are named:

<hostname>_<epoch-of-jobtracker-start>_<job-id>_<job-name>
where <hostname> is the hostname of the machine creating these logs, <epoch-of-jobtracker-start> is the number of milliseconds that had elapsed since Unix Epoch when the jobtracker daemon was started, <job-id> is the job ID, and <job-name> is the name of the job.

For example:

ec2-72-44-61-184.compute-1.amazonaws.com_1250641772616_job_200908190029_0002_hadoop_test-mini-mr
These logs are not rotated.You can clear these logs periodically without affecting Hadoop. However, consider archiving the logs if they are of interest in the job development process. Make sure you do not move or delete a file that is being written to by a running job.

Individual statistics logs are created for each job that is submitted to the cluster. The size of each log file varies. Jobs with more tasks produce larger files.

Reference: Apache Hadoop Log Files: Where to find them in CDH, and what info they contain

Question No : 7

Which three processes does HDFS High Availability (HA) enable on your cluster?

A. Automatically 'fail over' between NameNodes if one goes down
B. Write data to two clusterssimultaneously
C. Shut one NameNode down for maintenance without halting the cluster
D. Manually 'fail over' between NameNodes
E. Configure unlimited hot standby NameNode.

Answer: A,C,D

Explanation: The HDFS High Availability feature addresses the above problems by providing the option of running two redundant NameNodes in the same cluster in an Active/Passive configuration with a hot standby. This allows a fast failover to a new NameNode in the case that a machine crashes, or a graceful administrator-initiated failover for the purpose of planned maintenance.

Question No : 8

You has a cluster running with the Fail Scheduler enabled. There are currently no jobs running on the cluster 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 al the same time. How will the Fair' Scheduler handle these two Jobs?

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

Answer: D

Explanation: Fair scheduling is a method of assigning resources to jobs such that all jobs get, on average, an equal share of resources over time. When there is a single job running, that job uses the entire cluster. When other jobs are submitted, tasks slots that free up are assigned to the new jobs, so that each job gets roughly the same amount of CPU time. Unlike the default Hadoop scheduler, which forms a queue of jobs, this lets short jobs finish in reasonable time while not starving long jobs. It is also a reasonable way to share a cluster between a number of users. Finally, fair sharing can also work with job priorities - the priorities are used as weights to determine the fraction of total compute time that each job should get.

Reference: Hadoop, Fair Scheduler Guide

Question No : 9

You've configured your cluster with HDFS Federation. One NameNode manages the /data namesapace and another Name/Node manages the /reports namespace. How do you configure a client machine to access both the /data and the /reports directories on the cluster?

A. Configure the client to mount the /data namespace. As long as a single namespace is mounted and the client participates in the cluster, HDFS grants access to all files in the cluster to that client.
B. Configure the client to mount both namespaces by specifying the appropriate properties in the core-site.xml
C. You cannot configure a client to access both directories in the current implementation of HDFS Federation.
D. You don’t need to configure any parameters on the client machine. Access is controlled

by the NameNodes managing the namespace.

Answer: C

Explanation: HDFS Federation

In order to scale the name service horizontally, federation uses multiple independent namenodes/namespaces. The namenodes are federated, that is, the namenodes are independent and don't require coordination with each other. The datanodes are used as common storage for blocks by all the namenodes. Each datanode registers with all the namenodes in the cluster. Datanodes send periodic heartbeats and block reports and handles commands from the namenodes.

Note: HDFS Federation improves the existing HDFS architecture through a clear separation of namespace and storage, enabling generic block storage layer. It enables support for multiple namespaces in the cluster to improve scalability and isolation. Federation also opens up the architecture, expanding the applicability of HDFS cluster to new implementations and use cases.

Reference: Hortonworks, An Introduction to HDFS Federation

Question No : 10

You have a cluster running with the Fair in 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.

Which of the following describes how the Fair Scheduler operates? (Choose 2)

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

Answer: C,D

Reference: http://hadoop.apache.org/common/docs/r0.20.2/fair_scheduler.html (introduction, first paragraph)

Question No : 11

What determines the number of Reduces that run a given MapReduce job on a cluster running MapReduce v1 (MRv1)?

A. It is set by the Hadoop framework and is based on the number of InputSplits of the job.
B. It is set by the developer.
C. It is set by the JobTracker based on the amount of intermediate data.
D. It is set and fixed by the cluster administrator in mapred-site.xml. The number set always run for any submitted job.

Answer: B

Explanation: Number of Reduces

The right number of reduces seems to be 0.95 or 1.75 * (nodes * mapred.tasktracker.tasks.maximum). At 0.95 all of the reduces can launch immediately and start transfering map outputs as the maps finish. At 1.75 the faster nodes will finish their first round of reduces and launch a second round of reduces doing a much better job of load balancing.

Currently the number of reduces is limited to roughly 1000 by the buffer size for the output files (io.buffer.size * 2 * numReduces << heapSize). This will be fixed at some point, but until it is it provides a pretty firm upper bound.

The number of reduces also controls the number of output files in the output directory, but usually that is not important because the next map/reduce step will split them into even smaller splits for the maps.

The number of reduce tasks can also be increased in the same way as the map tasks, via JobConf's conf.setNumReduceTasks(int num).

Reference: org.apache.hadoop.mapred

Class JobConf

Question No : 12

Identify which two daemons typically run each slave node in a Hadoop cluster running MapReduce v1 (MRv1)

A. NodeManager
B. TaskTracker
C. DataNode
D. NameNode
E. Secondary NameNode
F. JobTracker

Answer: B,C

Explanation: A TaskTracker is a slave node daemon in the cluster that accepts tasks (Map, Reduce and Shuffle operations) from a JobTracker.

The following two daemons run on each Slave nodes:
* DataNode – Stores actual HDFS data blocks.
* TaskTracker - Responsible for instantiating and monitoring individual Map and Reduce tasks.

Reference: 24 Interview Questions & Answers for Hadoop MapReduce developers, How many Daemon processes run on a Hadoop system?

Question No : 13

What two processes must you do if you are running a Hadoop cluster with a single NameNode and six DataNodes, and you want to change a configuration parameter so that it affects all six DataNodes.

A. You must restart the NameNode daemon to apply the changes to the cluster
B. You must restart all six DataNode daemons to apply the changes to the cluster.
C. You don't need to restart any daemon, as they will pick up changes automatically.
D. You must modify the configuration files on each of the six DataNode machines.
E. You must modify the configuration files on only one of the DataNode machine
F. You must modify the configuration files on the NameNode only. DataNodes read their configuration from the master nodes.

Answer: A,F

Explanation: Note:Typically one machine in the cluster is designated as the NameNode and another machine the as JobTracker, exclusively. These are

the masters. The rest ofthe machines in the cluster actas both DataNode and TaskTracker. These are the slaves.

Question No : 14

Your Hadoop cluster contains nodes in three racks. Choose which scenario results if you leave the dfs.hosts property in the NameNode's configuration file empty (blank)?

A. The NameNode will update dfs.hosts property to include machines running the DataNode daemon on the next NameNode reboot or with a dfsadmin –refreshNodes.
B. Any machine running the DataNode daemon can immediately join the cluster.
C. Presented with a blank dfs.hosts property, the NameNode will permit DataNodes specified in mapred.hosts to join the cluster.
D. No new can be added to the cluster until you specify them in the dfs.hosts file.

Answer: D

Explanation: The dfs.hosts and mapred.hosts properties allow an administrator to supply a file containing an approved list of hostnames. If a machine is not in this list, it will be denied access to the cluster. This can be used to enforce policies regarding which teams of developers have access to which MapReduce sub-clusters. These are configured in exactly the same way as the excludes file.

Reference: Apache Hadoop , Module 7: Managing a Hadoop Cluster

Question No : 15

What metadata is stored on a DataNode when a block is written to it?

A. None. Only the block itself is written.
B. Checksums for the data in the block, as a separate file.
C. Information on the file's location in HDFS.
D. Node location of each block belonging to the same namespace.

Answer: D

Explanation: Each DataNode keeps a small amount of metadata allowing it to identify the cluster it participates in. If this metadata is lost, then the DataNode cannot participate in an HDFS instance and the data blocks it stores cannot be reached.

When an HDFS instance is formatted, the NameNode generates a unique namespace id for the instance. When DataNodes first connect to the NameNode, they bind to this namespace id and establish a unique “storage id” that identifies that particular DataNode in the HDFS instance. This data as well as information about what version of Hadoop was used to create the block files, is stored in a filed named VERSION in the ${dfs.data.dir}/current directory.

Note: Administrators of HDFS clusters understand that the HDFS metadata is some of the most precious bits they have. While you might have hundreds of terabytes of information stored in HDFS, the NameNode’s metadata is the key that allows this information, spread across several million “blocks” to be reassembled into coherent, ordered files.

Reference: Protecting per-DataNode Metadata

Question No : 16

Your cluster implements HDFS High Availability (HA). You two NameNodes are named nn01 and nn02. What occurs when you execute the command:

Hdfs haadmin -failover nn01 nn02

A. nn02 becomes the standby NameNode and nn02 becomes the active NameNode
B. Nn01 is fenced, and nn01 becomes the active NameNode
C. Nn01 is fenced, and nn02 becomes the active NameNode
D. Nn01 becomes the standby NameNode and nn02 becomes the active NameNode

Answer: C

Explanation: Failover- initiate a failover between two NameNodes

This subcommand causes a failover from the first provided NameNode to the second. If the first NameNode is in the Standby state, this command simply transitions the second to the Active state without error. If the first NameNode is in the Active state, an attempt will be made to gracefully transition it to the Standby state. If this fails, the fencing methods (as

configured by dfs.ha.fencing.methods) will be attempted in order until one of the methods succeeds. Only after this process will the second NameNode be transitioned to the Active state. If no fencing method succeeds, the second NameNode will not be transitioned to the Active state, and an error will be returned.

Reference: HDFS High Availability Administration, HA Administration using the haadmin command

Question No : 17

What is the standard configuration of slave nodes in a Hadoop cluster?

A. Each slave node either runs a TaskTracker or a DataNode daemon, but not both.
B. Each slave node runs a JobTracker and a DataNode daemon.
C. Each slave node runs a TaskTracker and a DataNode daemon.
D. Each slave node runs a DataNode daemon, but only a fraction of the slave nodes run TaskTrackers.
E. Each slave node runs a TaskTracker, but only a fraction of the slave nodes run DataNode daemons.

Answer: C

Reference: http://bradhedlund.com/2011/09/10/understanding-hadoop-clusters-and-the-network/ (second paragraph on the page)

Question No : 18

Which two updates occur when a client application opens a stream to begin a file write on a cluster running MapReduce v1 (MRv1)?

A. Once the write stream closes on the DataNode, the DataNode immediately initiates a black report to the NameNode.
B. The change is written to the NameNode disk.
C. The metadata in the RAM on the NameNode is flushed to disk.
D. The metadata in RAM on the NameNode is flushed disk.
E. The metadata in RAM on the NameNode is updated.

F. The change is written to the edits file.

Answer: D,F

Explanation: Note: Namenode stores modifications to the filesystem as a log appended to a native filesystem file (edits). When a Namenode starts up, it reads HDFS state from an image file (fsimage) and then applies edits from edits log file. It then writes new HDFS state to (fsimage) and starts normal operation with an empty edits file. Since namenode merges fsimage and edits files only during start up, edits file could get very large over time on a large cluster. Another side effect of larger edits file is that next restart of Namenade takes longer.

The secondary namenode merges fsimage and edits log periodically and keeps edits log size with in a limit. It is usually run on a different machine than the primary Namenode since its memory requirements are on the same order as the primary namemode. The secondary namenode is started by bin/start-dfs.sh on the nodes specified in conf/masters file.

Question No : 19

In HDFS, you view a file with rw-r--r-- set as its permissions. What does this tell you about the file?

A. The file cannot be deleted by anyone but the owner
B. The file cannot be deleted by anyone
C. The file cannot be run as a MapReduce job
D. The file's contents can be modified by the owner, but no-one else
E. As a Filesystem in Userspace (FUSE), HDFS files are available to all user's on a cluster regardless of their underlying POSIX permissions.

Answer: A

Explanation: The owner has rw- permissions (read and write). Only he can deletethe file.

Note:
Starting with Hadoop 0.16.1, HDFS has included a rudimentary file permissions system. This permission system is based on the POSIX model, but does not provide strong security for HDFS files. The HDFS permissions system is designed to prevent accidental corruption of data or casual misuse of information within a group of users who share access to a cluster. It is not a strong security model that guarantees denial of access to unauthorized

parties.

HDFS security is based on the POSIX model of users and groups. Each file or directory has 3 permissions (read, write and execute) associated with it at three different granularities: the file's owner, users in the same group as the owner, and all other users in the system. As the HDFS does not provide the full POSIX spectrum of activity, some combinations of bits will be meaningless. For example, no file can be executed; the +x bits cannot be set on files (only directories). Nor can an existing file be written to, although the +w bits may still be set.

Security permissions and ownership can be modified using the bin/hadoop dfs -chmod, -chown, and -chgrp operations described earlier in this document; they work in a similar fashion to the POSIX/Linux tools of the same name.

Reference: HDFS Permissions and Security

Question No : 20

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 No : 21

Compare the hardware requirements of the NameNode with that of the DataNodes in a Hadoop cluster running MapReduce v1 (MRv1):

A. The NameNode requires more memory and requires greater disk capacity than the DataNodes.
B. The NameNode and DataNodes should the same hardware configuration.
C. The NameNode requires more memory and no disk drives.
D. The NameNode requires more memory but less disk capacity.
E. The NameNode requires less memory and less disk capacity than the DataNodes.

Answer: D

Explanation: Note:
* 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. Client applications talk to the NameNode whenever they wish to locate a file, or when they want to add/copy/move/delete a file. The NameNode responds the successful requests by returning a list of relevant DataNode servers where the data lives.

* A DataNode stores data in the Hadoop File System HDFS. There is only One DataNode process run on any hadoop slave node. DataNode runs on its own JVM process. On startup, a DataNode connects to the NameNode. DataNode instances can talk to each other, this is mostly during replicating data.

Reference: 24 Interview Questions & Answers for Hadoop MapReduce developers

Question No : 22

Which of the following statements is the most accurate about the choice of operating systems to run on a Hadoop cluster?

A. Linux and Solaris/OpenSolaris are preferable to Windows. Solaris running on SPARC hardware is the preferred Hadoop slave node configuration.
B. Linux is preferable to Windows and Solaris/OpenSolaris. Some Linux distributions are intended for cluster environments more than others.