**Hadoop Interview Question and Answer**

**What daemons are needed to run a Hadoop cluster?**

***Answer:*** DataNode, NameNode, TaskTracker, and JobTracker are required to run Hadoop cluster.

**What is structured unstructured and semi-structured data?**

***Answer:*** Any data which is on a tabular format is considered to be structured.

Any data which is in XML or a Log file is considered to be semi -structured.

However, any data such as an audio, video, or an image file is considered to be unstructured data.

**Which OS are supported by Hadoop deployment?**

***Answer:*** The main OS use for Hadoop is Linux. However, by using some additional software, it can be deployed on Windows platform.

**What are the common Input Formats in Hadoop?**

***Answer:*** Three widely used input formats are:

1. Text Input: It is default input format in Hadoop.

2. Key Value: It is used for plain text files

3. Sequence: Use for reading files in sequence

**What modes can Hadoop code be run in?**

***Answer:*** Hadoop can be deployed in

1. Standalone mode

2. Pseudo-distributed mode

3. Fully distributed mode.

**What is the main difference between RDBMS and Hadoop?**

***Answer:*** RDBMS is used for transactional systems to store and process the data whereas Hadoop can

be used to store the huge amount of data.

**What are the important hardware requirements for a Hadoop cluster?**

***Answer:*** There are no specific requirements for data nodes.

However, the namenodes need a specific amount of RAM to store filesystem image in memory.

This depends on the particular design of the primary and secondary namenode.

**How would you deploy different components of Hadoop in production?**

***Answer:*** You need to deploy jobtracker and namenode on the master node then deploy datanodes on multiple slave nodes.

**What is Big Data?**

***Answer:*** Big data is a term which describes the large volume of data. Big data can be used to make

better decisions and strategic business moves.

**OR**

**One of the perspective is according to oracle anything above 1 terabyte is a big data because we cannot process that data with the traditional database systems.**

**OR**

**Something that cannot be handled by the traditional database system is called Big data.**

**What is Hadoop and its components?**

***Answer:*** When “Big Data” emerged as a problem, Hadoop evolved as a solution for it. It is a framework

which provides various services or tools to store and process Big Data. It also helps to analyze

Big Data and to make business decisions which are difficult using the traditional method.

**What are the essential features of Hadoop?**

***Answer:*** Hadoop framework has the competence of solving many questions for Big Data analysis. It's

designed on Google MapReduce which is based on Google’s Big Data file systems.

**What are the network requirements to run Hadoop?**

***Answer:*** 1) SSH is required to run - to launch server processes on the slave nodes.

2) A password less SSH connection is required between the master, secondary machines and all the slaves.

**What do you need to do as Hadoop admin after adding new data nodes?**

***Answer:*** You need to start the balancer for redistributing data equally between all nodes so that Hadoop

cluster will find new data nodes automatically. To optimize the cluster performance, you should

start rebalance to redistribute the data between data nodes.

**Explain how you will restart a NameNode?**

***Answer:*** The easiest way of doing is to run the command to stop running shell script.

Just click on stop.all.sh. then restarts the NameNode by clocking on start-all-sh.

**What are the Hadoop shell commands can use for copy operation?**

***Answer:*** The copy operation command are:

fs –copyToLocal

fs –put

fs –copyFromLocal.

**What is the Importance of the namenode?**

***Answer:*** The role of namenonde is very crucial in Hadoop. It is the brain of the Hadoop. It is largely

responsible for managing the distribution blocks on the system. It also supplies the specific

Addresses for the data based when the client made a request.

**What happens when the NameNode is down?**

***Answer:*** If the NameNode is down, the file system goes offline.

**Is it possible to copy files between different clusters? If yes, how can you achieve this?**

***Answer:*** Yes, we can copy files between multiple Hadoop clusters. This can be done using distributed copy.

**Is there any standard method to deploy Hadoop?**

***Answer:*** No, there are now standard procedure to deploy data using Hadoop. There are few general

requirements for all Hadoop distributions. However, the specific methods will always different for each Hadoop admin.

**What is distcp?**

***Answer:*** Distcp is a Hadoop copy utility. It is mainly used for performing MapReduce jobs to copy data.

The key challenges in the Hadoop environment is copying data across various clusters, and distcp will also offer to provide multiple datanodes for parallel copying of the data.

**What is a checkpoint?**

***Answer:*** Checkpointing is a method which takes a FsImage. It edits log and compacts them into a newFsImage. Therefore, instead of replaying an edit log, the NameNode can be load in the final in-memory state directly from the FsImage.

This is surely more efficient operation which reduces NameNode startup time.

**What is rack awareness?**

***Answer:*** It is a method which decides how to put blocks base on the rack definitions. Hadoop will try tolimit the network traffic between datanodes which is present in the same rack. So that, it willonly contact remote.

**What is the use of 'jps' command?**

***Answer:*** The 'jps' command helps us to find that the Hadoop daemons are running or not. It also displays all the Hadoop daemons like namenode, datanode, node manager, resource manager, etc.which are running on the machine.

**Name some of the essential Hadoop tools for effective working with Big Data?**

***Answer:*** "Hive," HBase, HDFS, ZooKeeper, NoSQL, Lucene/SolrSee, Avro, Oozie, Flume, Clouds, and

SQL are some of the Hadoop tools that enhance the performance of Big Data.

**How many times do you need to reformat the namenode?**

***Answer:*** The namenode only needs to format once in the beginning. After that, it will never formatted. In fact, reformatting of the namenode can lead to loss of the data on entire the namenode.

**What Are The Job Resource Files?**

***Answer:*** job.xml and job.jar are core resources to process the Job. Job Client copy the resources to the HDFS.

**What do you mean by downtime?**

***Answer:*** **Downtime is when the cluster is not responding. This happens at times of configuration/error changes, Hardware error / upgrade. The -Ops responsibilities are to minimize the downtime so that the (revenue impact applications) are running all the time with no delay.**

**What is speculative execution?**

***Answer:*** If a node is executing a task slower than the master node. Then there is needs to redundantly execute one more instance of the same task on another node.

So the task finishes first will beaccepted and the other one likely to be killed. This process is known as “speculative execution.”

Hadoop run the process in commodity hardware, so it’s possible to fail the systems also has low memory. So if system failed, process also failed, it’s not recommendable.Speculative execution is a process performance optimization technique.Computation/logic distribute to the multiple systems and execute which system execute quickly. By default this value is true. Now even the system crashed, not a problem, framework choose logic from other systems.

E.g. logic distributed on A, B, C, D systems, completed within a time.

System A, System B, System C, System D systems executed 10 min, 8 mins, 9 mins 12 mins simultaneously. So consider system B and kill remaining system processes, framework take care to kill the other system process.

**Which is the best operating system to run Hadoop?**

***Answer:*** Ubuntu or Linux is the most preferred operating system to run Hadoop. Though Windows OS can also be used to run Hadoop but it will lead to several problems and is not recommended.

**Explain the differences between Hadoop 1.x and Hadoop 2.x**

***Answer:*** In Hadoop 1.x, MapReduce is responsible for both processing and cluster management whereas in Hadoop 2.x processing is taken care of by other processing models and YARN is responsible for cluster management.

Hadoop 2.x scales better when compared to Hadoop 1.x with close to 10000 nodes per cluster.

Hadoop 1.x has single point of failure problem and whenever the NameNode fails it has to be recovered manually. However, in case of Hadoop 2.x StandBy NameNode overcomes the SPOF problem and whenever the NameNode fails it is configured for automatic recovery.

Hadoop 1.x works on the concept of slots whereas Hadoop 2.x works on the concept of containers and can also run generic tasks.

**What are the core changes in Hadoop 2.0?**

***Answer:*** Hadoop 2.x provides an upgrade to Hadoop 1.x in terms of resource management, scheduling and the manner in which execution occurs.

In Hadoop 2.x the cluster resource management capabilities work in isolation from the MapReduce specific programming logic. This helps Hadoop to share resources dynamically between multiple parallel processing frameworks like Impala and the core MapReduce component.

Hadoop 2.x Hadoop 2.x allows workable and fine grained resource configuration leading to efficient and better cluster utilization so that the application can scale to process larger number of jobs.

**While I am reading the data the namenode goes down what will happen**

***Answer:*** There won’t be any impact as the location of the data blocks have already been conveyed to the client before the reading was started, hence the task will complete

**What is the main difference between an "Input Split" and "HDFS Block"?**

***Answer:*** "Input Split" is the logical division of the data while The "HDFS Block" is the physical division of the data.

**How will you decide whether you need to use the Capacity Scheduler or the Fair Scheduler?**

***Answer:*** **Fair Scheduling** is the process in which resources are assigned to jobs such that all jobs get to share equal number of resources over time. Fair Scheduler can be used under the following circumstances -

i) If you wants the jobs to make equal progress instead of following the FIFO order then you must use Fair Scheduling.

ii) If you have slow connectivity and data locality plays a vital role and makes a significant difference to the job runtime then you must use Fair Scheduling.

iii) Use fair scheduling if there is lot of variability in the utilization between pools.

**Capacity Scheduler** allows runs the hadoop mapreduce cluster as a shared, multi-tenant cluster to maximize the utilization of the hadoop cluster and throughput.Capacity Scheduler can be used under the following circumstances -

i) If the jobs require scheduler detrminism then Capacity Scheduler can be useful.

ii) CS's memory based scheduling method is useful if the jobs have varying memory requirements.

iii) If you want to enforce resource allocation because you know very well about the cluster utilization and workload then use Capacity Scheduler.

**How will you restart a NameNode?**

***Answer:***The easiest way of doing this is to run the command to stop running shell script i.e. click on stop-all.sh. Once this is done, restarts the NameNode by clicking on start-all.sh.

**Explain about the different schedulers available in Hadoop.**

***Answer:***

• FIFO Scheduler – This scheduler does not consider the heterogeneity in the system but orders the jobs based on their arrival times in a queue.

• COSHH- This scheduler considers the workload, cluster and the user heterogeneity for scheduling decisions.

• Fair Sharing-This Hadoop scheduler defines a pool for each user. The pool contains a number of map and reduce slots on a resource. Each user can use their own pool to execute the jobs.

**What are the important hardware considerations when deploying Hadoop in production environment?**

***Answer:***

• Memory-System’s memory requirements will vary between the worker services and management services based on the application.

• Operating System - a 64-bit operating system avoids any restrictions to be imposed on the amount of memory that can be used on worker nodes.

• Storage- It is preferable to design a Hadoop platform by moving the compute activity to data to achieve scalability and high performance.

• Capacity- Large Form Factor (3.5”) disks cost less and allow to store more, when compared to Small Form Factor disks.

• Network - Two TOR (Top of Rack) switches per rack provide better redundancy.

(It randomly choose the best path || earlier top three tier switch used to traffic jam which was affecting performance).

• Computational Capacity- This can be determined by the total number of MapReduce slots available across all the nodes within a Hadoop cluster.

**How many NameNodes can you run on a single Hadoop cluster?**

***Answer:*** Only one.

**What is the /conf/hadoop-env.sh file and which variable in the file should be set for Hadoop to work?**

***Answer:*** This file provides an environment for Hadoop to run and consists of the following variables-HADOOP\_CLASSPATH, JAVA\_HOME and HADOOP\_LOG\_DIR. JAVA\_HOME variable should be set for Hadoop to run.

**Apart from using the JPS command is there any other way that you can check whether the NameNode is working or not?**

***Answer:*** Use the command -/etc/init.d/hadoop-0.20-namenode status.

**In a MapReduce system, if the HDFS block size is 64 MB and there are 3 files of size 127MB, 64K and 65MB with FileInputFormat. Under this scenario, how many input splits are likely to be made by the Hadoop framework?**

***Answer:*** 2 splits each for 127 MB and 65 MB files and 1 split for the 64KB file.

**Which command is used to verify if the HDFS is corrupt or not?**

***Answer:*** Hadoop FSCK (File System Check) command is used to check missing blocks.

**List some use cases of the Hadoop Ecosystem?**

***Answer:*** Text Mining, Graph Analysis, Semantic Analysis, Sentiment Analysis, Recommendation Systems.

**How can you kill a Hadoop job?**

***Answer:***

1)hadoop job -kill job\_id OR

2)yarn application -kill application\_id

**I want to see all the jobs running in a Hadoop cluster. How can you do this?**

***Answer:*** Using the command Hadoop job –list, gives the list of jobs running in a Hadoop cluster.

**Is it possible to copy files across multiple clusters? If yes, how can you accomplish this?**

***Answer:*** Yes, it is possible to copy files across multiple Hadoop clusters and this can be achieved using distributed copy. DistCP command is used for intra or inter cluster copying.

**The mapred.output.compress property is set to true, to make sure that all output files are compressed for efficient space usage on the Hadoop cluster. In case under a particular condition if a cluster user does not require compressed data for a job. What would you suggest that he do?**

***Answer:*** If the user does not want to compress the data for a particular job then he should create his own configuration file and set the mapred.output.compress property to false. This configuration file then should be loaded as a resource into the job.

**What is the best practice to deploy a secondary NameNode?**

***Answer:*** It is always better to deploy a secondary NameNode on a separate standalone machine. When the secondary NameNode is deployed on a separate machine it does not interfere with the operations of the primary node.

**How often should the NameNode be reformatted?**

***Answer:*** The NameNode should never be reformatted. Doing so will result in complete data loss. NameNode is formatted only once at the beginning after which it creates the directory structure for file system metadata and namespace ID for the entire file system.

**If Hadoop spawns 100 tasks for a job and one of the job fails. What does Hadoop do?**

***Answer:*** The task will be started again on a new TaskTracker and if it fails more than 4 times which is the default setting (the default value can be changed), the job will be killed.

**What are the important properties of hdfs-site.xml?**

***Answer:*** - There are three important properties of hdfs-site.xml:-

A) data.dr:- Identify the location of the storage of data.

B) name.dr:- Identifies the location of metadata storage and specify whether DFS is located on disk or the remote location.

C) checkpoint.dr:- For Secondary NameNode.

**How can you add and remove nodes from the Hadoop cluster?**

***Answer:*** To add/remove new nodes to the HDFS cluster, the hostnames should be add/remove to the /etc/hosts on all machine & edit Namenode hdfs-site.xml file and edit Resource Manager yarn-site.xml after that DataNode and TaskTracker should be started on the new node by following commands,

1) Resource Manager Machine :- yarn rmadmin –refreshNodes

2) Namenode Machine :- hdfs dfsadmin –refreshNodes

**You increase the replication level but notice that the data is under replicated. What could have gone wrong?**

***Answer:*** Nothing could have actually wrong, if there is huge volume of data because data replication usually takes times based on data size as the cluster has to copy the data and it might take a few hours.

**Explain about the different configuration files and where are they located.**

***Answer:*** The configuration files are located in “conf” sub directory. Hadoop has 3 different Configuration files- hdfs-site.xml, core-site.xml and mapred-site.xml/yarn-site.xml.

**If 8TB is the available disk space per node (10 disks with 1 TB, 2 disk for operating system etc. were excluded)Assuming initial data size is 600 TB. How will you estimate the number of data nodes (n)?**

***Answer:***-

Estimating the hardware requirement is always challenging in Hadoop environment because we never know when data storage demand can increase for a business.

We must understand following factors in detail to come to a conclusion for the current scenario of adding right numbers to the cluster:

The actual size of data to store – 600 TB

At what pace the data will increase in the future (per day/week/month/quarter/year) – Data trending analysis or business requirement justification (prediction)

We are in Hadoop world, so replication factor plays an important role – default 3x replicas

Hardware machine overhead (OS, logs etc.) – 2 disks were considered

Intermediate mapper and reducer data output on hard disk - 1x

Space utilization between 60 % to 70 % - Finally, as a perfect designer we never want our hard drive to be full with their storage capacity.Compression ratio

Let’s do some calculation to find the number of data nodes required to store 600 TB of data:

**## Rough calculation:**

Data Size – 600 TB

Replication factor – 3

Intermediate data – 1

Total Storage requirement – (3+1) \* 600 = 2400 TB

Available disk size for storage – 8 TB

Total number of required data nodes (approx.): 2400/8 = 300 machines

**## Actual Calculation: Rough Calculation + Disk space utilization + Compression ratio**

Disk space utilization – 65 % (differ business to business)

Compression ratio – 2.3

Total Storage requirement – 2400/2.3 = 1043.5 TB

Available disk size for storage – 8\*0.65 = 5.2 TB

Total number of required data nodes (approx.): 1043.5/5.2 = 201 machines

Actual usable cluster size (100 %): (201\*8\*2.3)/4 = 925 TB

**## Case: Business has predicted 20 % data increase in a quarter and we need to predict the new machines to be added in a year**

Data increase – 20 % over a quarter

**Additional data:**

1st quarter: 1043.5 \* 0.2 = 208.7 TB

2nd quarter: 1043.5 \* 1.2 \* 0.2 = 250.44 TB

3rd quarter: 1043.5 \* (1.2)^2 \* 0.2 = 300.5 TB

4th quarter: 1043.5 \* (1.2)^3 \* 0.2 = 360.6 TB

**Additional data nodes requirement (approx.):**

1st quarter: 208.7/5.2 = 41 machines

2nd quarter: 250.44/5.2 = 49 machines

3rd quarter: 300.5/5.2 = 58 machines

4th quarter: 360.6/5.2 = 70 machines

With these numbers you can predict next year additional machines requirement for the cluster (last quarter + 24), (last quarter + 28) and so on.

**You have a directory Gabbar that has the following files – HadoopTraining.txt, \_SparkTraining.txt, #DataScienceTraining.txt, .SalesforceTraining.txt. If you pass the Gabbar directory to the Hadoop MapReduce jobs, how many files are likely to be processed?**

***Answer:***-

Only HadoopTraining.txt and #DataScienceTraining.txt will be processed for Mapreduce jobs because,

when we process a file (either in a directory or individual) in Hadoop using any FileInputFormat such as TextInputFormat, KeyValueInputFormat or SequenceFileInputFormat, we must confirm that none of files must have a hidden file prefix such as “\_” or “.” because,mapreduce FileInputFormat will by default uses hiddenFileFilter class to ignore all those files with these prefix in their name.

private static final PathFilter hiddenFileFilter = new PathFilter()

{

public boolean accept(Path p){

String name = p.getName();

return !name.startsWith("\_") && !name.startsWith(".");

}

};

However, we can set our own custom filter such as FileInputFormat.setInputPathFilter to eliminate such criteria but remember, hiddenFileFilter is always active.

**Imagine that you are uploading a file of 500MB into HDFS.100MB of data is successfully uploaded into HDFS and another client wants to read the uploaded data while the upload is still in progress.What will happen in such a scenario, will the 100 MB of data that is uploaded will it be displayed?**

***Answer:***-Although the default blocks size is 64 MB in Hadoop 1x and 128 MB in Hadoop 2x,whereas in such a scenario let us consider block size to be 100 MB which means that we are going to have 5 blocks replicated 3 times (default replication factor).

Let’s consider an example of how does a block is written to HDFS:We have 5 blocks (A/B/C/D/E) for a file, a client, a namenode and a datanode. So, first the client will take Block A and will approach namenode for datanode location to store this block and the replicated copies.

Once client is aware about the datanode information, it will directly reach out to datanode and start copying Block A which will be simultaneously replicated to other 2 datanodes.

Once the block is copied and replicated to the datanodes, client will get the confirmation about the Block A storage and then, it will initiate the same process for next block “Block B”.

So, during this process if 1st block of 100 MB is written to HDFS and the next block has been started by the client to store then 1st block will be visible to readers. Only the current block being written will not be visible by the readers.

**When decommissioning the nodes in a Hadoop Cluster, why should you stop all the task trackers?**

***Answer:***-

So but what about the task tracker running a MapReduce job on a datanode which is likely to be decommissioned. Unlike the datanode, there is no graceful way to decommission a tasktracker.

it’s always assumed that when we want to move the same task to other node then we need to rely on making the task process to stop for failure & need to rescheduled elsewhere on the cluster.

It is possible that a task on its final attempt is running on the tasktracker and that a final failure may result in the entire job failing. Unfortunately,

it’s not always possible to prevent this case from occurring. So, the idea behind decommissioning that it will stop your data node but to move the current task to another node, we need to manually stop the task tracker running on the decommissioned node.

**Did you ever run a lopsided job that resulted in out of memory ever? If yes, then how did you handle it?**

***Answer:***-

OutOfMemoryError” is the most common error in MapReduce jobs because data is growing with different sizes which makes a challenging environment for a developer to estimate-

the right amount of memory allocated for a job. In Hadoop world, it is not only an administrator job to look after the configuration but developer has also given an opportunity-

to manage their own jobs configuration. We must make sure that following properties must be set appropriately considering the available resources in the cluster-

to avoid out of memory error:

mapreduce.map.memory.mb: Maximum amount of memory used by a mapper within a container

mapreduce.map.java.opts: Maximum amount of heap size used by a mapper which must be less than the above

mapreduce.reduce.memory.mb: Maximum amount of memory used by a reducer within a container

mapreduce.reduce.java.opts: Maximum amount of heap size used by a reducer which must be less than the above

yarn.scheduler.maximum-allocation-mb: The maximum allocation size allowed for a container but require administrative privileges.

There are some other factors also which may impact your memory such as spilling data over disk which can be corrected using following configuration:

mapreduce.reduce.shuffle.input.buffer.percent

mapreduce.reduce.shuffle.memory.limit.percent

mapreduce.reduce.shuffle.parallel.copies

**What are the differences between -copyFromLocal and -put command & copyToLocal and -put command ?**

Basically, both put and copyFromLocal fulfill similar purposes, but there are some differences. First, see what both the command does-

-put: it can copy the file from source to destination

– copyFromLocal: It copies the file from local file system to Hadoop system

As you saw, put can do what copyFromLocal is doing but the reverse is not true. So the main difference between -copyFromLocal and -put commands is, in -copyFromLocal, the source has to be the local file system which is not mandatory for –put command.

**There are 100 map tasks that are running, of which 99 tasks have completed and one task is running very slow. The slow running map task is replicated on a different machine and the output is gathered from the first completed map task. All other map tasks are killed. What is this phenomenon referred to in Hadoop?**

***Answer:***-In the real world, user code is buggy, processes crash, and machines fail. One of the major benefits of using Hadoop is its ability to handle such failures and allow your job to complete.So, basically we will first understand the failure cases and causesTask Failure: (There are many cases wherein the tasks may fail)

**Case 1:** Child Task Failure

This happens when user code in the map or reduce task throws a runtime exception. If this happens, the child JVM reports the error back to its parent tasktracker, before it exits.The error ultimately makes it into the user logs. The tasktracker marks the task attempt as failed, freeing up a slot to run another task.

**Case 2:** For Streaming Task failures

For Streaming tasks, if the Streaming process exits with a nonzero exit code, it is marked as failed. This behavior is governed by the stream.non.zero.exit.is.failure property (the default is true).

**Case 3:** Sudden failure or exit of the task in JVM due to JVM bug

Another failure mode is the sudden exit of the child JVM—perhaps there is a JVM bug that causes the JVM to exit for a particular set of circumstances exposed by the Map-Reduce user code. In this case, the tasktracker notices that the process has exited, and marks the attempt as failed.

**Case 4:** Hanging tasks

These are dealt differently. The tasktracker notices that it hasn’t received a progress update for a while, and proceeds to mark the task as failed. The child JVM process will be automatically killed after this period.

\*The timeout period after which tasks are considered failed is normally 10 minutes, and can be configured on a per-job basis (or a cluster basis) by setting the mapred.task.timeout property to a value in milliseconds.Setting the timeout to a value of zero disables the timeout, so long-running tasks are never marked as failed. In this case, a hanging task will never free up its slot, and over time there may be cluster slowdown as a result.

This approach should be avoided, and always making sure that a task is reporting progress periodically.Hadoop’s way to handle such task failures:

**Approach 1:**

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. Furthermore, if a task fails more than four times, it will not be retried further.

This value is configurable: the maximum number of attempts to run a task is controlled by the mapred.map.max.attempts property for map tasks, and mapred.reduce.max.attempts for reduce tasks. By default, if any task fails more than four times (or whatever the maximum number of attempts is configured to), the whole job fails.

**Approach 2:**

For some applications it is undesirable to abort the job if a few tasks fail, as it may be possible to use the results of the job despite some failures. In this case, the maximum percentage of tasks that are allowed to fail without triggering job failure can be set for the job.

Map tasks and reduce tasks are controlled independently, using the mapred.max.map.failures.percent and mapred.max.reduce.failures.percent properties.

A task attempt may also be killed, which is different from it failing. A task attempt may be killed because it is a speculative duplicate, or because the tasktracker it was running on failed, and the jobtracker marked all the task attempts running on it as killed.

Killed task attempts do not count against the number of attempts to run the task (as set by mapred.map.max.attempts and mapred.reduce.max.attempts), since it wasn’t the task’s fault that an attempt was killed.Users may also kill or fail task attempts using the web UI or the command line (type hadoop job to see the options). Jobs may also be killed by the same mechanisms.

**When a job is run the properties file is copied to the distributed cache for the map jobs to access. How can you access the properties file?**

***Answer:***- ## Not exact answer but some what related read carefully ##

This is a common problem - the -files option works as an aside from the DistributedCache.

When you use -files, the GenericOptionsParser configures a job property called tmpfiles, while the DistributedCache uses a property called mapred.cache.files.

Also distributed cache expects the files to already be in HDFS and copies them down to the task nodes, where as -files copies to files to HDFS at job submission and then copies them to each task node.

In your case, to make your code work, just create a File object and name the file you passed in (obviously this requires you to know the filename of the local file, and hard code it into your mapper code). The file will be in the current working directory:

**Map Reduce**

**What Is Hadoop Mapreduce ?**

***Answer:*** MapReduce is a set of programs used to process or analyze vast of data over a Hadoop cluster. It process the vast amount of the datasets parallels across the clusters in a fault-tolerant manner across the Hadoop framework.

**Can You Elaborate Mapreduce Job Architecture?**

***Answer:*** First Hadoop programmer submit Mapreduce program to JobClient.

Job Client request the JobTracker to get Job id, Job tracker provide JobID, its’s in the form of Job\_HadoopStartedtime\_00001. It’s unique ID.

Once JobClient receive received Job ID copy the Job resources (job.xml, job.jar) to File System (HDFS) and submit job to JobTracker. JobTracker initiate Job and schedule the job.

Based on configuration, job split the input splits and submit to HDFS. TaskTracker retrieve the job resources from HDFS and launch Child JVM. In this Child JVM, run the map and reduce tasks and notify to the Job tracker the job status.

**Can You Elaborate About Mapreduce Job?**

***Answer:*** Based on the configuration, the MapReduce Job first splits the input data into independent chunks called Blocks. These blocks processed by Map() and Reduce() functions. First Map function process the data, then processed by reduce function. The Framework takes care of sorts the Map outputs, scheduling the tasks.

**How order of execution Map reduce Job?**

***Answer:***- The order of execution in map-reduce is,

1.Mapper

2.Combiner

3.Partitioner

4.Shuffling/Sorting

5.Reducer

**Why Compute Nodes And The Storage Nodes Are The Same?**

***Answer:*** Compute nodes for processing the data, Storage nodes for storing the data. By default Hadoop framework tries to minimize the network wastage, to achieve that goal Framework follows the Data locality concept. The Compute code execute where the data is stored, so the data node and compute node are the same.

**What Is The Configuration Object Importance In Mapreduce?**

***Answer:*** It’s used to set/get of parameter name & value pairs in XML file.It’s used to initialize values, read from external file and set as a value parameter. Parameter values in the program always overwrite with new values which are coming from external configure files. Parameter values received from Hadoop’s default values.

**Why Task Tracker Launch Child JVM?**

***Answer:*** Most frequently, Hadoop developer mistakenly submit wrong jobs or having bugs. If Task Tracker use existent JVM, it may interrupt the main JVM, so other tasks may influenced. Whereas child JVM if it’s trying to damage existent resources, TaskTracker kill that child JVM and retry or relaunch new child JVM.

**Why Job client, Job Tracker Submits Job Resources to File System?**

***Answer:*** Data locality. Move competition is cheaper than moving Data. So logic/ competition in Jar file and splits. So Where the data available, in File System Datanodes. So every resources copy where the data available.

**If I want to change default block size value, then which parameter and configuration files need to change?**

***Answer:*** - You have to make changes in dfs.block.size in hdfs.xml file.

**In Map Reduce Ideally how many maps should be configured on a slave?**

***Answer:***- By default, no of mappers are 2 on each slave-node. We can set/change this value using mapreduce.tasktracker.map.tasks.maximum parameter.

You need to set this parameter in mapred-site.xml file. We should not directly select random value to set the no of mappers.

**Where is the output of Mapper written?**

***Answer:***- The output of mappers are written on local disk rather than the HDFS Blocks. Because of the following reasons :

There are 2 levels of processing (Map and Reduce) involved to get the final desired outcome.

The result generated by mappers are just intermediate/temporary result which is intern result to the Reducers so writing this would be costly process and inefficient.

The final result (outcome of reducers) is stored on HDFS block.

**What is default mapper & reducer can we set as per job?**

***Answer:***-

Mapper:- There are 2 Default Mapper per Node or Slave and minimum mapper we can set as 1 per job.

Reducer:-There is 1 Default Reducer per job but minimum we can set as 0 in Hadoop and it is valid configuration.

Following are the way to set the reducer to 0 By setting

mapred.reduce.task=0

job.setNumReduceTasks(0);

Description:-When we set reducer to "0" in this situation whatever the output of mapper will be last output because there is no reducer to process.

No reduce phase gets executed and output from mapper is considered as final output and written in HDFS.

Ex:- If a data is coming from RDBMS then it is already sorted so some time client's project demand they only need mapper not reducer bcoz data is already in sorted structured form.

Generally we will do that in map-only job (there is no need of reducer job and output of mapper is considered as final output)

**In MapReduce where sorting is done on mapper node or reducer node?**

***Answer:***- The role of Mapper is to execute the business logic and produce the Key/Value pairs which are passed to Partitioner. The Partitioner sorts the mapper output Key/Value pairs.

The Partitioner instance runs in the SAME JVM as of Mapper and hence the same node.

Sorting in Map Reduce is on Both Mapper Side as well as Reducer side.

**A) Mapper side (Quick Sort):-** The Mapper process the input key value pair from Record Reader and generates output as per custom Business logic, Before writing the data the output is partitioned and sorted by key.

**B) Reducer side (Merge Sort):-** In Reducer phase the data from the Mapper is again sorted as per the key and sent to Reducer for processing data as per custom business logic.

The shuffling and sorting in Reducer phase happens simultaneously.

Merge sort is the default feature of Map Reduce. One cannot change the Map Reduce sorting method, the reason is that data comes from the different nodes to a single point,

So the best algorithm that can be used here is the merge sort.

**Can we pass output of one reducer as input to another mapper?**

***Answer:***- Yes we can pass output of one reducer to another mapper.

If you are using TextInputFormat to read the file then reducer output will be treated as key value pair for your mapper where each line offset from beginning of the file will be key and entire line will be value.

**How to compress mapper output?**

***Answer:***- The output of Mapper is also Known as intermediate output is written to the local disk.

To compress mapper output we should set conf.set("mapreduce.map.output.compress", true)

We also need to consider some other factors like, which codec to use and what should be the compression type.

Following are the properties for configuring the same:-

A) mapred.map.output.compression.codec

B) mapred.output.compression.type

**Why aggregation cannot be done in Mapper?**

***Answer:***- We cannot do aggregation (addition) in a mapper because, sorting is not done in a mapper. Sorting happens only on the reducer side. Mapper method initialization depends upon each input split. While doing aggregation, we will lose the value of the previous instance. For each row, a new mapper will get initialized.

For each row, input split again gets divided into mapper, thus we do not have a track of the previous row value.

Aggregation cannot be performed in Mapper side. Below are the reasons for the same:

1. Aggregation requires sorting of data, which happens only at Reducer side.

2. For aggregation, we require output from all the mappers, which cannot be possible during map phase, because map tasks will be running in different nodes, where data blocks are present.

3. Mapper is instantiated per InputSplit. Hence, once the InputSplit is processed, the data is lost from mapper and it is written as intermediate output to the local disk.Hence, there will not be previous data present in the mapper for aggregation.

4. If we try to aggregate in mapper, this requires movement of data from all the mapper outputs running in different machines, which increases network congestion.

**What will happen if in a Map-Reduce Job if there is no data/file in input directory? How many Mapper and Reducer will run in this case?**

***Answer:***- If input directory is empty, then no mappers or Reducer will run.

As a number of mappers depend upon the number of InputSplits, as no data no input splits hence no mappers.

Without any mapper, a number of the reducer is also 0.

If we try to run map/reduce job on Hadoop cluster without specifying any input file it will throw following

exception: java.io.IOException: No input paths specified in job

**Where Mapreduce Not Recommended?**

***Answer:*** Mapreduce is not recommended for Iterative kind of processing. It means repeat the output in a loop manner or cyclic data flow (i.e. a chain of stages in which each output of the previous stage is the input to the next stage).To process Series of Mapreduce jobs, MapReduce not suitable. each job persists data in local disk, then again load to another job. It’s costly operation and not recommended.

MapReduce cannot handle:-

Interactive Processing

Real-time (stream) Processing

Iterative (delta) Processing

In-memory Processing

Graph Processing

**What Is Namenode And It’s Responsibilities?**

***Answer:*** Namenode is a logical daemon name for a particular node. It’s heart of the entire Hadoop system. Which store the metadata in FsImage and get all block information in the form of Heartbeat.

**What Is Jobtracker’s Responsibility?**

***Answer:*** Scheduling the job’s tasks on the slaves. Slaves execute the tasks as directed by the JobTracker. Monitoring the tasks, if failed, re¬execute the failed tasks.

**What Are The Jobtracker & Tasktracker In Mapreduce?**

***Answer:*** MapReduce Framework consists of a single Job Tracker per Cluster, one Task Tracker per node. Usually A cluster has multiple nodes, so each cluster has single Job Tracker and multiple TaskTrackers.JobTracker can schedule the job and monitor the Task Trackers. If Task Tracker failed to execute tasks, try to re-execute the failed tasks.

TaskTracker follow the JobTracker’s instructions and execute the tasks. As a slave node, it report the job status to Master JobTracker in the form of Heartbeat.

**What Is Job Scheduling Importance In Hadoop Mapreduce?**

***Answer:*** Scheduling is a systematic procedure of allocating resources in the best possible way among multiple tasks. Hadoop task tracker performing many procedures, sometime a particular procedure should finish quickly and provide more priority, to do it few job schedulers come into the picture. Default Schedule is FIFO. Fair scheduling, FIFO and Capacity Scheduler are most popular Hadoop scheduling in Hadoop.

**When Used Reducer?**

***Answer:*** To combine multiple mapper’s output used reducer. Reducer has 3 primary phases sort, shuffle and reduce. It’s possible to process data without reducer, but used when the shuffle and sort is required.

**What Is Replication Factor?**

***Answer:*** A chunk of data is stored in different nodes with in a cluster called replication factor. By default replication value is 3, but it’s possible to change it. Automatically each file is split into blocks and spread across the cluster.

**Where the Shuffle And Sort Process Does?**

***Answer:*** After Mapper generate the output temporary store the intermediate data on the local File System. Usually this temporary file configured at core¬site.xml in the Hadoop file. Hadoop Framework aggregate and sort this intermediate data, then update into Hadoop to be processed by the Reduce function. The Framework deletes this temporary data in the local system after Hadoop completes the job.

**What Methods Can Control The Map And Reduce Function’s Output?**

***Answer:*** setOutputKeyClass() and setOutputValueClass()

If they are different, then the map output type can be set using the methods.

setMapOutputKeyClass() and setMapOutputValueClass()

**Java Is Mandatory To Write Mapreduce Jobs?**

***Answer:*** Rack Awareness No, By default Hadoop implemented in JavaTM, but MapReduce applications need not be written in Java. Hadoop support Python, Ruby, C++ and other Programming languages.

Hadoop Streaming API allows to create and run Map/Reduce jobs with any executable or script as the mapper and/or the reducer.Hadoop Pipes allows programmers to implement MapReduce applications by using C++ programs.

**What Happen If Number Of Reducer Is 0?**

***Answer:*** Number of reducer = 0 also valid configuration in MapReduce. In this scenario, No reducer will execute, so mapper output consider as output, Hadoop store this information in separate folder.

**What Is The Main Difference Between Mapper And Reducer?**

***Answer:*** Map method is called separately for each key/value have been processed. It process input key/value pairs and emits intermediate key/value pairs.

Reduce method is called separately for each key/values list pair. It process intermediate key/value pairs and emits final key/value pairs.

Both are initialize and called before any other method is called. Both don’t have any parameters and no output.

**What Is Difference between Mapside Join and Reduce Side Join? Or When We Goes To Mapside Join And Reduce Join?**

***Answer:*** Join multiple tables in mapper side, called map side join. Please note mapside join should has strict format and sorted properly. If dataset is smaller tables, goes through reducer phrase. Data should partitioned properly.

Join the multiple tables in reducer side called reduce side join. If you have large amount of data tables, planning to join both tables. One table is large amount of rows and columns, another one has few number of tables only, and goes through Reduce side join. It’s the best way to join the multiple tables.

**What Is Combiner?**

***Answer:*** It’s a logical aggregation of key and value pair produced by mapper. It’s reduces a lot amount of duplicated data transfer between nodes, so eventually optimize the job performance. The framework decides whether combiner runs zero or multiple times. It’s not suitable where mean function occurs.

When a Map Reduce(MR) job is run on a large dataset, Map task generates huge chunks of intermediate data, which is passed on to Reduce task.

During this phase, the output from Mapper has to travel over the network to the node where Reducer is running. This data movement may cause network congestion if the data is huge.

To reduce this network congestion, MR framework provides a function called 'Combiner', which is also called as 'Mini-Reducer'

The role of Combiner is to take the output of Mapper as it's input, process it and sends its output to the reducer.

Combiner reads each key-value pair, combines all the values for the same key, and sends this as input to the reducer, which reduces the data movement in the network.

Combiner works along with each Mapper. Combiner uses same class as Reducer.

With default value 3, combiner only runs if there are more than 3 spill files written to the disk.

As an example lets take part of data in

Mapper1 ((car,1), (bike,1), (car,1))

Mapper2 ((car,1),(bus,1),(bike,1),(bus,1)).

When the combiner in Mapper1 run it will produce ((car,2), (bike,1)) and combiner in Mapper2 will produce ((car,1), (bus,2), (bike,1)).

These o/P's from Combiners are passed on to reducers.

**When We Are Goes To Combiner? Why It Is Recommendable?**

***Answer:*** Mappers and reducers are independent they don’t talk each other. When the functions that are commutative (a.b = b.a) and associative {a.(b.c) = (a.b).c} we goes to combiner to optimize the mapreduce process. Many mapreduce jobs are limited by the bandwidth, so by default Hadoop framework minimizes the data bandwidth network wastage.

To achieve its goal, Mapreduce allows user defined “Combiner function” to run on the map output. It’s a MapReduce optimization technique, but it’s optional.

**What Is The Main Difference Between Mapreduce Combiner And Reducer?**

***Answer:*** Both Combiner and Reducer are optional, but most frequently used in MapReduce. There are three main differences such as:

A) Combiner will get only one input from one Mapper. While Reducer will get multiple mappers from different mappers.

B) If aggregation required used reducer, but if the function follows commutative (a.b=b.a) and associative a.(b.c)= (a.b).c law, use combiner.

C) Input and output keys and values types must same in combiner, but reducer can follows any type input, any output format.

**What Is Partition?**

***Answer:*** After combiner and intermediate map¬output the Partitioner controls the keys after sort and shuffle. Partitioner divides the intermediate data according to the number of reducers so that all the data in a single partition gets executed by a single reducer. It means each partition can executed by only a single reducer. If you call reducer, automatically partition called in reducer by automatically.

**When We Goes To Partition?**

***Answer:*** By default Hive reads entire dataset even the application have a slice of data. It’s a bottleneck for mapreduce jobs. So Hive allows special option called partitions. When you are creating table, hive partitioning the table based on requirement.

**What Are The Important Steps When You Are Partitioning Table?**

***Answer:***Don’t over partition the data with too small partitions, it’s overhead to the namenode.

if dynamic partition, atleast one static partition should exist and set to strict mode by using given commands.

SET hive.exec.dynamic.partition = true;

SET hive.exec.dynamic.partition.mode = nonstrict;

first load data into non¬partitioned table, then load such data into partitioned table. It’s not possible to load data from local to partitioned table.

insert overwrite table table\_name partition(year) select \* from non¬partition¬table;

**How Many Mappers And Reducers Can Run?**

***Answer:***By default Hadoop can run 2 mappers and 2 reducers in one datanode. also each node has 2 map slots and 2 reducer slots. It’s possible to change this default values in Mapreduce.xml in conf file.

**What Is Inputsplit?**

***Answer:*** A chunk of data processed by a single mapper called InputSplit. In another words logical chunk of data which processed by a single mapper called Input split, by default inputSplit = block Size.

**How to Configure The Split Value?**

***Answer:*** By default block size = 64mb, but to process the data, job tracker split the data. Hadoop architect use these formulas to know split size.

split size = min (max\_splitsize, max (block\_size, min\_split\_size));

split size = max(min\_split\_size, min (block\_size, max\_split, size));

by default split size = block size ,Always No of splits = No of mappers.

**How Much Ram Required To Process 64mb Data?**

***Answer:*** Let’s assume. 64 block size, system take 2 mappers, 2 reducers, so 64\*4 = 256 MB memory and OS take at least 30% extra space so at least 256 + 80 = 326MB Ram required to process a chunk of data. So in this way required more memory to process unstructured process.

**What Is Difference Between Block And Split?**

***Answer:*** Block: How much chunk data to stored in the memory called block.

Split: how much data to process the data called split.

**Why Hadoop Framework Reads A File Parallel Why Not Sequential?**

***Answer:*** To retrieve data faster, Hadoop reads data parallel, the main reason it can access data faster. While, writes in sequence, but not parallel, the main reason it might result one node can be overwritten by other and where the second node. Parallel processing is independent, so there is no relation between two nodes, if writes data in parallel, it’s not possible where the next chunk of data has. For example 100 MB data write parallel, 64 MB one block another block 36, if data writes parallel first block doesn’t know where the remaining data. So Hadoop reads parallel and write sequentially.

**If I Am Change Block Size From 64 To 128, Then What Happen?**

***Answer:*** Even you have changed block size not effect existent data. After changed the block size, every file chunked after 128 MB of block size. It means old data is in 64 MB chunks, but new data stored in 128 MB blocks.

**What Is splitable()?**

***Answer:*** By default this value is true. It is used to split the data in the input format. if unstructured data, it’s not recommendable to split the data, so process entire file as a one split. To do it first change is Splitable() to false.

**How Much Hadoop Allows Maximum Block Size And Minimum Block Size?**

***Answer:*** Minimum: 1MB is minimum block size limit. No one can decrease fewer than block size.

Maximum: Depends on environment. There is no upper bound.

**What’s The Mapreduce Job Consists?**

***Answer:*** MapReduce job is a unit of work that client wants to be performed. It consists of input data, MapReduce program in Jar file and configuration setting in XML files. Hadoop runs this job by dividing it in different tasks with the help of JobTracker

**What Is The Data Locality?**

***Answer:***Wherever the data is there process the data, computation/process the data where the data available, this process called data locality. “Moving Computation is Cheaper than Moving Data” to achieve this goal follow data locality. It’s possible when the data is splittable, by default it’s true.

Data Locality in Hadoop means moving computation close to data rather than moving data towards computation. Hadoop stores data in HDFS, which splits files into blocks and distribute among various data nodes. When a map Reduce job is submitted, it is divided into map jobs and reduce jobs.

A Map job is assigned to a datanode according to the availability of the data, i.e. it assigns the task to a datanode which is closer to or stores the data on its local disk.

Data locality refers the process of placing computation near to data , which helps in high throughput and faster execution of data.

**1. Data Local: -** If a map task is executing on a node which has the input block to be processed, its called data local.

**2. Intra- Rack: -** It’s always not possible to run map task on the same node where data is located due to network constraints. In that case, mapper runs on another machine, but on the same rack. So the data need to be moved between the nodes for execution.

**3. Inter-Rack:-** In certain cases Intra- Rack local is also not possible. In such cases, the mapper will execute from a different rack. In order to execute the mapper, the data need to be copied from the node which stores the data to the node which is executing the mapper between the racks.

**When We Goes To Reducer?**

***Answer:***When sort and shuffle is required then only goes to reducers otherwise no need partition. If filter, no need to sort and shuffle. So without reducer its possible to do this operation.

**What Is Chain Mapper?**

***Answer:*** Chain mapper class is a special mapper class sets which run in a chain fashion within a single map task. It means, one mapper input acts as another mapper’s input, in this way n number of mapper connected in chain fashion.

**OR**

Chain Mapper is a class which is defined in org.apache.hadoop.mapreduce.lib.chain.ChainMapper package. This class can be used to run multiple Mapper in a single map task.

All mappers are run in a chain fashioned, the output of the first mapper becomes the input of the second mapper, and the output of the second mapper becomes the input of the third mapper, and so on until the last mapper.

The last mapper output is written to intermediate files (Or in-memory if the complete data is less 80/100 MB).

**How To Do Value Level Comparison?**

***Answer:***Hadoop can process key level comparison only but not in the value level comparison.

**What Is Setup And Clean Up Methods?**

***Answer:***If you don’t no what is starting and ending point/lines, it’s much difficult to solve those problems. Setup and clean up can resolve it. N number of blocks, by default 1 mapper called to each split. Each split has one start and cleanup methods. N number of methods, number of lines.

Setup is initialize job resources. The purpose of cleanup is close the job resources. Map is process the data. Once last map is completed, cleanup is initialized. It improves the data transfer performance.

All these block size comparison can do in reducer as well. If you have any key and value, compare one key value to another key value use it.

If you compare record level used these setup and cleanup. It open once and process many times and close once. So it save a lot of network wastage during process.

**How Many Slots Allocate For Each Task?**

***Answer:*** By default each task has 2 slots for mapper and 2 slots for reducer. So each node has 4 slots to process the data.

**Why Tasktracker Launch Child Jvm To Do A Task? Why Not Use Existent JVM?**

***Answer:*** Sometime child threads currupt parent threads. It means because of programmer mistake entired MapReduce task distruped. So task tracker launch a child JVM to process individual mapper or tasker. If tasktracker use existent JVM, it might damage main JVM. If any bugs occur, tasktracker kill the child process and relaunch another child JVM to do the same task. Usually task tracker relaunch and retry the task 4 times.

**What Are The Main Components Of Mapreduce Job?**

***Answer:*** # Main Driver Class: providing job configuration parameters

# Mapper Class: must extend org.apache.hadoop.mapreduce.Mapper class and performs execution of map() method

# Reducer Class: must extend org.apache.hadoop.mapreduce.Reducer class

**What Main Configuration Parameters Are Specified In Mapreduce?**

***Answer:*** The MapReduce programmers need to specify following configuration parameters to perform the map and reduce jobs:

A) The input location of the job in HDFs.

B) The output location of the job in HDFS.

C) The input’s and output’s format.

D) The classes containing map and reduce functions, respectively.

E) The .jar file for mapper, reducer and driver classes

**What Is Partitioner And Its Usage?**

***Answer:*** Partitioner is yet another important phase that controls the partitioning of the intermediate map-reduce output keys using a hash function. The process of partitioning determines in what reducer, a key-value pair (of the map output) is sent. The number of partitions is equal to the total number of reduce jobs for the process.

Hash Partitioner is the default class available in Hadoop , which implements the following function.int getPartition(K key, V value, int numReduceTasks)

The function returns the partition number using the numReduceTasks is the number of fixed reducers.

**What Is Identity Mapper?**

***Answer:*** Identity Mapper is the default Mapper class provided by Hadoop. when no other Mapper class is defined, Identify will be executed. It only writes the input data into output and do not perform and computations and calculations on the input data. The class name is org.apache.hadoop.mapred.lib.IdentityMapper.

**OR**

Identity Mapper is the default Mapper class provided by Hadoop 1.x . This class will be picked automatically when no mapper is specified in MapReduce driver class.

Identity Mapper class implements identity function, which directly writes all its input key-value pair into output. You can check the implementation of Identity Mapper class in grepcode.com website

IdentityMapper class is defined in old mapreduce API (MR1) in org.apache.hadoop.mapred.lib package

From Hadoop 2.x (MR2 or YARN) onwards, it uses Mapper class which is defined in org.apache.hadoop.mapreduce package if you don't specify any Mapper class in MapReduce drive program

**Explain the complete flow of MapReduce.**

***Answer:***- The very first starting point in the execution of the MapReduce is InputFormat.

InputFormat defines how the data is split up logically. The default input format is TextInputFormat. There are other different InputFormat types as well depending upon the requirements e.g. FileInputFormat, KeyValueInputFormat, NLineInputFormat and TextInputFormat.

Once InputFormat is defined then a logical unit of data called as InputSplit in the context of MapReduce. Equivalent of Split in HDFS is block) is fed to next component which is RecordReader (RR).

One Split/block is processed by one mapper. The RR component reads records from Split and feeds the record to the map function of mapper in the form of Key/Value pair.

Each time a record is read, each time it is passed to map function of the mapper e.g. if the split is having 2000 records then map() will be called 2000 times.

After the record has been fed to map() function then your business logic defined in the map() function gets executed and the output of map() function (in the form of Key/Values) is passed to next component called as Partitioner.

The output of the Mapper (if reducers are set) is temporary and is NOT written to HDFS. Before the intermediate output is passed to Partitioner, an optional component is executed Combiner (for optimization, if set).

The idea behind Combiner is to reduce the data that need to be moved over the network and pre-process the data before it goes to Reducer. Generally, a combiner & reducer is having similar logic.

Partitioner is the most important component in the whole flow. The Partitioner is the component which decides where a given Key/Value pair will go to (which reducer).

Before passing on Mapper’s output to Reducer, framework sorts the output based on Key.

Then intermediate output is shuffled to Reducers. Shuffling is the actual movement of the data over the network. How does Partitioner decide which Key/Value to send to which Reducer – the decision is taken based on the hash value of the key.

The idea of using this hashing technique is purely for purpose of equal load distribution point of view e.g. hashcode 1-100 will go to redcuer1, 101 to 200 will go to reducer2 and so on.

Point to remember is that shuffling happens as soon as first mapper finishes. Once the reducer task receives the input (output of mapper) from all the mappers then the Reducer executes the custom business logic defined in the reduce function of Reducer.

Normally, the aggregation (e.g. summation, max, min etc) related functions are performed in Reducers.Sometimes, we want to sort the data by value than by keys.

This sorting on reducer is achieved by a technique called as Secondary Sorting at Reducer. The final output (output of reducer) is collected by Output Format & Record Writer.

The Record Writer combine the output and produces the final output in the form of Key/Value pair. The final output format comes in the form of OutputFormat. This final output is written to HDFS and hence gets replicated based on replication factor.

One must remember that it is only Reducer’s output that gets written to HDFS and not of Mapper.

The output of mapper is thrown away after shuffling.Below is the sequence of execution of different components: -

InputFormat --> Split --> Record Reader --> Mapper --> Combiner (Optional) --> Partitioner --> Sort and shuffle --> Reducers --> Record Writer --> OutputFormat --> Written to HDFS

**What Is Recordreader In A Map Reduce?**

***Answer:*** RecordReader is used to read key/value pairs form the InputSplit by converting the byte-oriented view and presenting record-oriented view to Mapper.

Record Reader communicates with the Input Split (created by Input Format) and converts the split into records.

Record Reader uses the data within the boundaries, defined by Input Split. At “start” Record Reader in Hadoop start generating key-value pairs and the “end” is where it should stop reading records.

By default it uses TextInputFormat for converting data into key-value pairs.

There are 2 types of Record Reader:

A) Line Record Reader- In Hadoop it is the default Record Reader that TextInputFormat provides. Hence, each line of the input file is the new value and a key is byte offset.

B)SequenceFileRecordReader- It reads data specified by the header of a sequence file.

**What Is Outputcommitter?**

***Answer:*** OutPutCommitter describes the commit of MapReduce task. FileOutputCommitter is the default available class available for OutputCommitter in MapReduce. It performs the following operations:

1. Create temporary output directory for the job during initialization.

2. Then, it cleans the job as in removes temporary output directory post job completion.

3. Sets up the task temporary output.

4. Identifies whether a task needs commit. The commit is applied if required.

5. JobSetup, JobCleanup and TaskCleanup are important tasks during output commit.

**What Is A “reducer” In Hadoop?**

***Answer:*** In Hadoop, a reducer collects the output generated by the mapper, processes it, and creates a final output of its own.

**What Are The Parameters Of Mappers And Reducers?**

***Answer:*** The four parameters for mappers are:

1. LongWritable (input)

2. text (input)

3. text (intermediate output)

4. IntWritable (intermediate output)

The four parameters for reducers are:

1. Text (intermediate output)

2. IntWritable (intermediate output)

3. Text (final output)

4. IntWritable (final output)

**What Is A “map” In Hadoop?**

***Answer:*** In Hadoop, a map is a phase in HDFS query solving. A map reads data from an input location, and outputs a key value pair according to the input type.

**Explain Job conf In Mapreduce?**

***Answer:*** It is a primary interface to define a map-reduce job in the Hadoop for job execution. JobConf specifies mapper, Combiner, partitioner, Reducer,InputFormat , OutputFormat implementations and other advanced job faets liek Comparators.

**Explain Job Scheduling Through Jobtracker?**

***Answer:*** JobTracker communicates with NameNode to identify data location and submits the work to TaskTracker node. The TaskTracker plays a major role as it notifies the JobTracker for any job failure.

It actually is referred to the heartbeat reporter reassuring the JobTracker that it is still alive.Later, the JobTracker is responsible for the actions as in it may either resubmit the job or mark a specific record as unreliable or blacklist it.

**What Is Sequence file inputformat?**

***Answer:*** A compressed binary output file format to read in sequence files and extends the FileInputFormat.It passes data between output-input (between output of one MapReduce job to input of another MapReduce job)phases of MapReduce jobs.

**How To Set Mappers And Reducers For Hadoop Jobs?**

***Answer:*** Users can configure JobConf variable to set number of mappers and reducers.

1.job.setNumMaptasks()

2.job.setNumreduceTasks()

**YARN**

**Assume you do not believe the information from the previous command is accurate. Which CLI command will update the node information at the ResourceManager?**

***Answer:*** Yarn rmadmin –refreshnodes

**True of False? When a cluster is managed by Ambari, manual changes to yarn-site.xml will go into effect after the service is restarted.**

***Answer:*** False. Manual changes will be overwritten by Ambari when the service restarts.

**What are the ports used to access ResourceManager?**

***Answer:*** Resource Manager : 8088

**What component of the YARN stack is not directly monitored for failure by the ResourceManager?**

***Answer:*** Job tasks – these are monitored only by the ApplicationMaster.

**Which YARN component is responsible for restarting a failed ApplicationMaster?**

***Answer:*** Resource Manager

**Which HDP feature centralizes YARN logging?**

***Answer:*** YARN LOG aggregation

**Is YARN a replacement of Hadoop MapReduce?**

***Answer:*** YARN is not a replacement of Hadoop but it is a more powerful and efficient technology that supports MapReduce and is also referred to as Hadoop 2.0 or MapReduce2.

**What are the additional benefits YARN brings in to Hadoop?**

***Answer:*** Effective utilization of the resources as multiple applications can be run in YARN all sharing a common resource.In Hadoop MapReduce there are seperate slots for Map and Reduce tasks whereas in YARN there is no fixed slot. The same container can be used for Map and Reduce tasks leading to better utilization.

YARN is backward compatible so all the existing MapReduce jobs.

Using YARN, one can even run applications that are not based on the MaReduce model.

**What is the relationship between datanode & node manager ?**

***Answer:*** The NodeManager is the YARN worker component while a DataNode is an HDFS worker component.

**Why yarn came into picture?**

***Answer:*** Before we understand how YARN came into the picture, we should understand how cluster resource management was done in Hadoop 1.0 and what the problem in that approach was.

Job Tracker, which does resource management & Scheduling/monitoring of job, is part of, MapReduce Framework.

In MapReduce framework, MapReduce job (MapReduce application) is divided between number of tasks called mappers and reducers. Each task runs on one of the machine (DataNode) of the cluster, and each machine has a limited number of predefined slots (map slot, reduce slot) for running tasks concurrently.

Here, JobTracker is responsible for both managing the cluster's resources and driving the execution of the MapReduce job. It reserves and schedules slots for all tasks, configures, runs and monitors each task, and if a task fails, it allocates a new slot and reattempts the task. After a task finishes, the job tracker cleans up temporary resources and releases the task's slot to make it available for other jobs.

Problems with this approach in Hadoop 1.0:

It limits scalability: JobTracker runs on single machine doing several task like Resource management, Job and task scheduling and Monitoring. Although there are so many machines (DataNode) available; they are not getting used. This limits scalability.

Availability Issue: In Hadoop 1.0, JobTracker is single Point of availability. This means if JobTracker fails, all jobs must restart.

Problem with Resource Utilization: In Hadoop 1.0, there is concept of predefined number of map slots and reduce slots for each TaskTrackers. Here the datanodes (resource) is sitting idle which is reserved for reduce slots even when there is immediate need for those resources to be used as mapper slots.

Limitation in running non-MapReduce Application: In Hadoop 1.0, Job tracker was tightly integrated with MapReduce and only supporting application that obeys MapReduce programming framework can run on Hadoop.

YARN took over the task of cluster management from MapReduce and MapReduce is streamlined to perform Data Processing only in which it is best.

Yarn does efficient utilization of the resource.

There are no more fixed map-reduce slots. YARN provides central resource manager. With YARN, you can now run multiple applications in Hadoop, all sharing a common resource.

Yarn can even run application that do not follow MapReduce model.

No more JobTracker and TaskTracker needed in Hadoop 2.0.

JobTracker and TaskTracker has totally disappeared. YARN splits the two major functionalities of theJobTracker i.e. resource management and job scheduling/monitoring into 2 separate daemons (components).

\*Resource Manager

\*Node Manager(node specific)

Central Resource Manager and node specific Node Manager together constitutes YARN.

**What exactly container contains?**

***Answer:*** It represents a collection of physical resources. Also could mean CPU cores, disk along with RAM.

**Which service manages cluster CPU and memory resources?**

***Answer:*** Yarn

**How many yarn daemons processes run on a Hadoop cluster?**

***Answer:*** Two:

Resource Manager – Master daemon process.

Node Manager - one slave daemon process per node in a cluster

**If Application Master went down, what will Happen?**

***Answer:***- RM will restart AM (yarn.resourcemanager.am.max-attempts=2) times by default)

**Have you ever configured Capacity Scheduler?**

***Answer:*** Yes I’ve done that, Using Ambari Views (YARN Queue Manager)

**Do you know Rack Awareness ?**

***Answer:*** Rack awareness is having the knowledge of cluster topology or more specifically how the different data nodes are distributed across the racks of a Hadoop cluster. Large Hadoop clusters are arranged in racks and network traffic between different nodes within the same rack is much more desirable than network traffic across the racks. In addition NameNode tries to place replicas of block on multiple racks for improved fault tolerance. Datanodes in the same rack will have more bandwidth and less latency whereas datanodes datanodes in separate racks will have less bandwidth & higher latency

Main purpose of rack awareness is (i) Better cluster performance (ii) Increasing the available of data block.

Configuring rack awareness : In Ambari click on hosts tab, then click on individual hosts and then click on hosts actions then click on set Rack and give the rack name and say OK. Do this for all the nodes in the cluster.

**Where can you quickly view the amount of utilized memory for the NodeManagers in your cluster on a per-machine basis?**

***Answer:***- Service > YARN > Heat Maps

**Which Web interface provides detailed information on running applications?**

***Answer:***- Resourmanager Web UI

**Assume you do not believe the information from the previous command is accurate. Which CLI command will update the node information at the ResourceManager?**

***Answer:***- Yarn rmadmin –refreshnodes

**What component of the YARN stack is not directly monitored for failure by the ResourceManager?**

***Answer:***- Job tasks – these are monitored only by the ApplicationMaster

**By default, how long does the ResourceManager wait until concluding that a NodeManager has failed?**

***Answer:***- 10 mins

**By default, how frequently does the NodeManager send a heartbeat to the ResourceManager if it is functioning properly?**

***Answer:***- Every one second

**What is UBER mode in Hadoop2?**

***Answer:***-

Normally mappers and reducers will run by ResourceManager (RM), RM will create separate container for mapper and reducer.

Uber configuration, will allow to run mapper and reducers in the same process as the ApplicationMaster (AM).

Uber jobs are jobs that are executed within the MapReduce ApplicationMaster. Rather than communicate with RM to create the mapper and reducer containers.

The AM runs the map and reduce tasks within its own process and avoided the overhead of launching and communicate with remote containers.

If you have a small dataset or you want to run MapReduce on small amount of data,

Uber configuration will help you out, by reducing additional time that MapReduce normally spends in mapper and reducers phase.

**Can I configure an Uber for all MapReduce job?**

***Answer:***- As of now, map-only jobs and jobs with one reducer are supported.

**Mistakenly User Deleted A File, How Hadoop Remote From Its File System? Can U Roll Back It?**

***Answer:*** -HDFS first renames its file name and place it in /trash directory for a configurable amount of time. In this scenario block might freed, but not file.

After this time, Namenode deletes the file from HDFS name-space and make file freed. It’s configurable as fs.trash.interval in core-site.xml.

By default its value is 1, you can set to 0 to delete file without storing in trash.

**What determines how many jobs you can run in parallel in YARN?**

***Answer:*** - With just the default queue. We could run 4 jobs in parallel.

Adding an extra queue did not allow us to run any more than 4 jobs.

We still were unable to run extra jobs in parallel. We can use the new queue but cannot get above 4 jobs running and the rest sit in ACCEPTED mode waiting on a job to stop.

**What are the ways to debug a failed Map Reduce job?**

***Answer:*** - Commonly there are two ways.

1) By using Map Reduce job counters

2) YARN Web UI for looking into syslogs for actual error messages or status.

**PIG**

**Compare Apache Pig And SQL?**

***Answer:*** **Apache Pig** differs from SQL in its usage for ETL, lazy evaluation, store data at any given point of time in the pipeline, support for pipeline splits and explicit declaration of execution plans. **SQL** is oriented around queries, which produce a single result. SQL has no in-built mechanism for splitting a data processing stream and applying different operators to each sub-stream.

**Apache Pig** allows user code to be included at any point in the pipeline whereas if **SQL** where to be used data needs to be imported to the database first and then the process of cleaning and transformation begins.

**Pig** **SQL**

• Pig is procedural (How) • SQL is declarative

• Nested relational data model • Flat relational data model

• Schema is optional • Schema is required

• Scan-centric analytic workloads • OLTP + OLAP workloads

• Limited query optimization • Significant opportunity for query optimization

**What is declarative language vs procedural language?**

***Answer:*** In a procedural language, you define the whole process and provide the steps how to do it. You just provide orders and define how the process will be served. Ex: - You have to describe How to make tea in detail. Ex: Pig

In a declarative language, you just set the command or order, and let it be on the system how to complete that order. You just need your result without digging into how it should be done. Ex: - Here you just order Tea, does not need to know how it is made just see the result. Ex: Hive, SQL.

**Explain The Need For Mapreduce While Programming In Apache Pig?**

***Answer:*** Apache Pig programs are written in a query language known as Pig Latin that is similar to the SQL query language. To execute the query, there is a need for an execution engine. The Pig engine converts the queries into MapReduce jobs and thus MapReduce acts as the execution engine and is needed to run the programs.

**Explain About The Bloommapfile.?**

***Answer:*** BloomMapFile is a class that extends the MapFile class. It is used in HBase table format to provide quick membership test for the keys using dynamic bloom filters.

**What Do You Mean By A Bag In Pig?**

***Answer:*** Collection of tuples is referred as a bag in Apache Pig.

**What Is The Usage Of Foreach Operation In Pig Scripts?**

***Answer:*** FOREACH operation in Apache Pig is used to apply transformation to each element in the data bag, so that respective action is performed to generate new data items.

Syntax- FOREACH data\_bagname GENERATE exp1, exp2.

**Explain About The Different Complex Data Types In Pig.?**

***Answer:***Apache Pig supports 3 complex data types:

Maps- These are key, value stores joined together using #.

Tuples- Just similar to the row in a table, where different items are separated by a comma. Tuples can have multiple attributes.

Bags- Unordered collection of tuples. Bag allows multiple duplicate tuples.

**What Does Flatten Do In Pig?**

***Answer:*** Sometimes there is data in a tuple or a bag and if we want to remove the level of nesting from that data, then flatten modifier in Pig can be used. Flatten un-nests bags and tuples. For tuples, the Flatten operator will substitute the fields of a tuple in place of a tuple, whereas un-nesting bags is a little complex because it requires creating new tuples.

**How Do Users Interact With The Shell In Apache Pig?**

***Answer:*** Using Grunt i.e. Apache Pig’s interactive shell, users can interact with HDFS or the local file system. To start Grunt, users should invoke Apache Pig with no command:

Executing the command “pig –x local” will result in the prompt, also interact with Local file system,

grunt >

Just type the command pig and enter will result in the prompt, also interact with HDFS file system,

This is where Pig Latin scripts can be run either in local mode or in cluster mode by setting the configuration in PIG\_CLASSPATH.

To exit from grunt shell, press CTRL+D or just type exit.

**What Do You Know About The Case Sensitivity Of Apache Pig?**

***Answer:*** It is difficult to say whether Apache Pig is case sensitive or case insensitive. For instance, user defined functions, relations and field names in pig are case sensitive i.e. the function COUNT is not the same as function count or X=load ‘foo’ is not same as x=load ‘foo’. On the other hand, keywords in Apache Pig are case insensitive i.e. LOAD is same as load.

**What Are The Debugging Tools Used For Apache Pig Scripts?**

***Answer:*** Describe and explain are the important debugging utilities in Apache Pig.

Explain utility is helpful for Hadoop developers, when trying to debug error or optimize Pig Latin scripts. Explain can applied on a particular alias in the script or it can applied to the entire script in the grunt interactive shell. Explain utility produces several graphs in text format, which can printed to a file.

Describe debugging utility is helpful to developers when writing Pig scripts as it shows the schema of a relation in the script. For beginners who are trying to learn Apache Pig can use the describe utility to understand how each operator makes alterations to data. A pig script can have multiple describes.

**What Is Illustrate Used For In Apache Pig?**

***Answer:*** Executing pig scripts on large data sets, usually takes a long time. To tackle this, developers run pig scripts on sample data but there is possibility that the sample data selected, might not execute your pig script properly.

For instance, if the script has a join operator there should be at least a few records in the sample data that have the same key, otherwise the join operation will not return any results. To tackle these kind of issues, illustrate is used. illustrate takes a sample from the data and whenever it comes across operators like join or filter that remove data, it ensures that only some records pass through and some do not, by making modifications to the records such that they meet the condition. Illustrate just shows the output of each stage but does not run any MapReduce task.

**Explain About The Execution Plans Of A Pig Script?<br> Or<br> Differentiate Between The Logical And Physical Plan Of An Apache Pig Script?**

***Answer:*** Logical and Physical plans are created during the execution of a pig script. Pig scripts are based on interpreter checking. Logical plan is produced after semantic checking and basic parsing and no data processing takes place during the creation of a logical plan. For each line in the Pig script, syntax check is performed for operators and a logical plan is created. Whenever an error is encountered within the script, an exception is thrown and the program execution ends, else for each statement in the script has its own logical plan. A logical plan contains collection of operators in the script but does not contain the edges between the operators.

After the logical plan is generated, the script execution moves to the physical plan where there is a description about the physical operators, Apache Pig will use, to execute the Pig script. A physical plan is more or less like a series of MapReduce jobs but then the plan does not have any reference on how it will be executed in MapReduce. During the creation of physical plan, co-group logical operator converted into three physical operators namely –Local Rearrange, Global Rearrange and Package. Load and store functions usually get resolved in the physical plan.

**What Are Some Of The Apache Pig Use Cases You Can Think Of?**

***Answer:*** Apache Pig big data tools, is used in particular for iterative processing, research on raw data and for traditional ETL data pipelines. As Pig can operate in circumstances where the schema is not known, inconsistent or incomplete- it is widely used by researchers who want to make use of the data before it is cleaned and loaded into the data warehouse.

To build behavior prediction models, for instance, it can be used by a website to track the response of the visitors to various types of ads, images, articles, etc.

**Differentiate Between Piglatin And Hiveql?**

***Answer:*** It is necessary to specify the schema in HiveQL, whereas it is optional in PigLatin.

HiveQL is a declarative language, whereas PigLatin is procedural.

HiveQL follows a flat relational data model, whereas PigLatin has nested relational data model.

**Is Pig Latin A Strongly Typed Language? If Yes, Then How Did You Come To The Conclusion?**

***Answer:*** In a strongly typed language, the user has to declare the type of all variables upfront. In Apache Pig, when you describe the schema of the data, it expects the data to come in the same format you mentioned.

However, when the schema is not known, the script will adapt to actually data types at runtime. So, it can be said that PigLatin is strongly typed in most cases but in rare cases it is gently typed, i.e. it continues to work with data that does not live up to its expectations.

**What Do You Understand By An Inner Bag And Outer Bag In Pig?**

***Answer:*** A relation inside a bag is referred to as inner bag and outer bag is just a relation in Pig.

**Differentiate Between Group And Co-group Operators?**

***Answer:*** Both GROUP and COGROUP operators are identical and can work with one or more relations. GROUP operator is generally used to group the data in a single relation for better readability, whereas COGROUP can be used to group the data in 2 or more relations. COGROUP is like a combination of GROUP and JOIN, i.e., it groups the tables based on a column and then joins them on the grouped columns. It is possible to co-group up to 127 relations at a time.

**Explain The Difference Between Count Star And Count Functions In Apache Pig?**

***Answer:*** COUNT function does not include the NULL value when counting the number of elements in a bag, whereas COUNT\_STAR (0 function includes NULL values while counting.

**What Are The Various Diagnostic Operators Available In Apache Pig?**

***Answer:*** Dump Operator- It is used to display the output of pig Latin statements on the screen, so that developers can debug the code.

**How Will You Merge The Contents Of Two Or More Relations And Divide A Single Relation Into Two Or More Relations?**

***Answer:*** This can be accomplished using the UNION and SPLIT operators.

**I Have A Relation R. How Can I Get The Top 10 Tuples From The Relation R?**

***Answer:*** TOP () function returns the top N tuples from a bag of tuples or a relation. N is passed as a parameter to the function top () along with the column whose values are to be compared and the relation R.

**What Are The Commonalities Between Pig And Hive?**

***Answer:*** HiveQL and Pig Latin both convert the commands into MapReduce jobs.

They cannot be used for OLAP transactions as it is difficult to execute low latency queries.

**What Are The Different Types Of UDF In Java Supported By Apache Pig?**

***Answer:*** Algebraic, Eval and Filter functions are the various types of UDF’s supported in Pig.

**You Have A File Employee.txt In The HDFS Directory With 100 Records. You Want To See Only The First 10 Records From The Employee.txt File. How Will You Do This?**

***Answer:*** The first step would be to load the file employee.txt into with the relation name as Employee. The first 10 records of the employee data can be obtained using the limit operator -

Result= limit employee 10.

**Explain About The Scalar Datatypes In Apache Pig?**

***Answer:*** integer, float, double, long, byte array and char array are the available scalar datatypes in Apache Pig.

**How Do Users Interact With HDFS In Apache Pig?**

***Answer:*** Using the grunt shell.

**What Is The Use Of Having Filters In Apache Pig?**

***Answer:*** Just like the where clause in SQL, Apache Pig has filters to extract records based on a given condition or predicate. The record is passed down the pipeline if the predicate or the condition turn to true. Predicate contains various operators like ==, <=,! =, >=.

Example:-

X= load ‘inputs’ as(name,address)

Y = filter X by symbol matches ‘Mr.\*’;

**What Is A UDF In Pig?**

***Answer:*** If the in-built operators do not provide some functions then programmers can implement those functionalities by writing user defined functions using other programming languages like Java, Python, Ruby, etc. These User Defined Functions (UDF’s) can then be embedded into a Pig Latin Script.

**Can You Join Multiple Fields In Apache Pig Scripts?**

***Answer:*** Yes, it is possible to join multiple fields in PIG scripts because the join operations takes records from one input and joins them with another input. This can be achieved by specifying the keys for each input and the two rows will be joined when the keys are equal.

**Does Pig Support Multi-line Commands?**

***Answer:*** Yes.

**Spark**

**What Is Apache Spark?**

***Answer:*** Spark is a fast, easy-to-use and flexible data processing framework. It has an advanced execution engine supporting cyclic data flow and in-memory computing. Spark can run on Hadoop, standalone or in the cloud and is capable of accessing diverse data sources including HDFS, HBase, Cassandra and others.

**What Is Shark?**

***Answer:*** Most of the data users know only SQL and are not good at programming. Shark is a tool, developed for people who are from a database background - to access Scala MLib capabilities through Hive like SQL interface. Shark tool helps data users run Hive on Spark - offering compatibility with Hive metastore, queries and data.

**What Is A Sparse Vector?**

***Answer:*** A sparse vector has two parallel arrays –one for indices and the other for values. These vectors are used for storing non-zero entries to save space.

**List Some Use Cases Where Spark Outperforms Hadoop In Processing?**

***Answer:*** Sensor Data Processing –Apache Spark’s ‘In-memory computing’ works best here, as data is retrieved and combined from different sources.

Spark is preferred over Hadoop for real time querying of data

Stream Processing – For processing logs and detecting frauds in live streams for alerts, Apache Spark is the best solution.

**What Is RDD?**

***Answer:*** RDDs (Resilient Distributed Datasets) are basic abstraction in Apache Spark that represent the data coming into the system in object format. RDDs are used for in-memory computations on large clusters, in a fault tolerant manner. RDDs are read-only portioned, collection of records that are

Immutable – RDDs cannot be altered.

Resilient – If a node holding the partition fails the other node takes the data.

**Explain About Transformations And Actions In The Context Of RDD?**

***Answer:*** Transformations are functions executed on demand, to produce a new RDD. All transformations are followed by actions. Some examples of transformations include map, filter and reduceByKey.

Actions are the results of RDD computations or transformations. After an action is performed, the data from RDD moves back to the local machine. Some examples of actions include reduce, collect, first, and take.

**What Are The Languages Supported By Apache Spark For Developing Big Data Applications?**

***Answer:*** Scala, Java, Python, R and Clojure

**Can You Use Spark To Access And Analyse Data Stored In Cassandra Databases?**

***Answer:*** Yes, it is possible if you use Spark Cassandra Connector.

**Is It Possible To Run Apache Spark On Apache Mesos?**

***Answer:*** Yes, Apache Spark can be run on the hardware clusters managed by Mesos.

**Explain About The Different Cluster Managers In Apache Spark?**

***Answer:*** The 3 different clusters managers supported in Apache Spark are:

1.YARN

2.Apache Mesos -Has rich resource scheduling capabilities and is well suited to run Spark along with other applications. It is advantageous when several users run interactive shells because it scales down the CPU allocation between commands.

3.Standalone deployments – Well suited for new deployments which only run and are easy to set up.

**How Can Spark Be Connected To Apache Mesos?**

***Answer:*** To connect Spark with Mesos:

Configure the spark driver program to connect to Mesos. Spark binary package should be in a location accessible by Mesos. (or)

Install Apache Spark in the same location as that of Apache Mesos and configure the property ‘spark.mesos.executor.home’ to point to the location where it is installed.

**How Can You Minimize Data Transfers When Working With Spark?**

***Answer:*** Minimizing data transfers and avoiding shuffling helps write spark programs that run in a fast and reliable manner.

The various ways in which data transfers can be minimized when working with Apache Spark are:

Using Broadcast Variable- Broadcast variable enhances the efficiency of joins between small and large RDDs.Using Accumulators – Accumulators help update the values of variables in parallel while executing.

The most common way is to avoid operations ByKey, repartition or any other operations which trigger shuffles.

**Why Is There A Need For Broadcast Variables When Working With Apache Spark?**

***Answer:*** These are read only variables, present in-memory cache on every machine. When working with Spark, usage of broadcast variables eliminates the necessity to ship copies of a variable for every task, so data can be processed faster. Broadcast variables help in storing a lookup table inside the memory which enhances the retrieval efficiency when compared to an RDD lookup ().

**Is It Possible To Run Spark and Mesos Along With Hadoop?**

***Answer:*** Yes, it is possible to run Spark and Mesos with Hadoop by launching each of these as a separate service on the machines. Mesos acts as a unified scheduler that assigns tasks to either Spark or Hadoop.

**What Is Lineage Graph?**

***Answer:*** The RDDs in Spark, depend on one or more other RDDs. The representation of dependencies in between RDDs is known as the lineage graph. Lineage graph information is used to compute each RDD on demand, so that whenever a part of persistent RDD is lost, the data that is lost can be recovered using the lineage graph information.

**How Can You Trigger Automatic Clean-ups In Spark To Handle Accumulated Metadata?**

***Answer:*** You can trigger the clean-ups by setting the parameter ‘spark.cleaner.ttl’ or by dividing the long running jobs into different batches and writing the intermediary results to the disk.

**Explain About The Major Libraries That Constitute The Spark Ecosystem?**

***Answer:*** Spark MLib- Machine learning library in Spark for commonly used learning algorithms like clustering, regression, classification, etc.

Spark Streaming – This library is used to process real time streaming data.

Spark GraphX – Spark API for graph parallel computations with basic operators like joinVertices, subgraph, aggregateMessages, etc.

Spark SQL – Helps execute SQL like queries on Spark data using standard visualization or BI tools.

**What Are The Benefits Of Using Spark With Apache Mesos?**

***Answer:*** It renders scalable partitioning among various Spark instances and dynamic partitioning between Spark and other big data frameworks.

**What Is The Significance Of Sliding Window Operation?**

***Answer:*** Sliding Window controls transmission of data packets between various computer networks. Spark Streaming library provides windowed computations where the transformations on RDDs are applied over a sliding window of data. Whenever the window slides, the RDDs that fall within the particular window are combined and operated upon to produce new RDDs of the windowed DStream.

**What Is A Dstream?**

***Answer:*** Discretized Stream is a sequence of Resilient Distributed Databases that represent a stream of data. DStreams can be created from various sources like Apache Kafka, HDFS, and Apache Flume.

DStreams have two operations: –

1. Transformations that produce a new DStream.

2. Output operations that write data to an external system.

**When Running Spark Applications, Is It Necessary To Install Spark On All The Nodes Of Yarn Cluster?**

***Answer:*** Spark need not be installed when running a job under YARN or Mesos because Spark can execute on top of YARN or Mesos clusters without affecting any change to the cluster.

**What Is Catalyst Framework?**

***Answer:*** Catalyst framework is a new optimization framework present in Spark SQL. It allows Spark to automatically transform SQL queries by adding new optimizations to build a faster processing system.

**Name A Few Companies That Use Apache Spark In Production.?**

***Answer:*** Pinterest, Conviva, Shopify, Open Table

**Which Spark Library Allows Reliable File Sharing At Memory Speed Across Different Cluster Frameworks?**

***Answer:*** Tachyon

**Why Is Blinkdb Used?**

***Answer:*** BlinkDB is a query engine for executing interactive SQL queries on huge volumes of data and renders query results marked with meaningful error bars. BlinkDB helps users balance ‘query accuracy’ with response time.

**How Can You Compare Hadoop And Spark In Terms Of Ease Of Use?**

***Answer:*** Hadoop MapReduce requires programming in Java which is difficult, though Pig and Hive make it considerably easier. Learning Pig and Hive syntax takes time. Spark has interactive APIs for different languages like Java, Python or Scala and also includes Shark i.e. Spark SQL for SQL lovers - making it comparatively easier to use than Hadoop.

**What Are The Common Mistakes Developers Make When Running Spark Applications?**

***Answer:*** Developers often make the mistake of:-

Hitting the web service several times by using multiple clusters.

Run everything on the local node instead of distributing it.

Developers need to be careful with this, as Spark makes use of memory for processing.

**How Can You Remove The Elements With A Key Present In Any Other Rdd?**

***Answer:*** Use the subtractByKey () function.

**What Is The Advantage Of A Parquet File?**

***Answer:*** Parquet file is a columnar format file that helps:

\*Limit I/O operations

\*Consumes less space

\*Fetches only required columns.

**What Are The Various Data Sources Available In Sparksql?**

***Answer:*** \*Parquet file

\*JSON Datasets

\*Hive tables

**What Are The Key Features Of Apache Spark That You Like?**

***Answer:*** Spark provides advanced analytic options like graph algorithms, machine learning, streaming data, etc.

It has built-in APIs in multiple languages like Java, Scala, Python and R

It has good performance gains, as it helps run an application in the Hadoop cluster ten times faster on disk and 100 times faster in memory.

**What Do You Understand By Pair RDD?**

***Answer:*** Special operations can be performed on RDDs in Spark using key/value pairs and such RDDs are referred to as Pair RDDs. Pair RDDs allow users to access each key in parallel. They have a reduceByKey () method that collects data based on each key and a join () method that combines different RDDs together, based on the elements having the same key.

**Explain About The Different Types Of Transformations On Dstreams?**

***Answer:*** \*Stateless Transformations:- Processing of the batch does not depend on the output of the previous batch.

Examples: map (), reduceByKey (), filter ().

\*Stateful Transformations:- Processing of the batch depends on the intermediary results of the previous batch.

Examples: Transformations that depend on sliding windows.

**Explain About The Popular Use Cases Of Apache Spark?**

***Answer:*** Apache Spark is mainly used for:

1. Iterative machine learning.

2. Interactive data analytics and processing.

3. Stream processing

4. Sensor data processing

**Is Apache Spark A Good Fit For Reinforcement Learning?**

***Answer:*** No. Apache Spark works well only for simple machine learning algorithms like clustering, regression, classification.

**What Is Spark Core?**

***Answer:*** It has all the basic functionalities of Spark, like - memory management, fault recovery, interacting with storage systems, scheduling tasks, etc.

**What Is The Difference Between Persist() And Cache()?**

***Answer:*** persist () allows the user to specify the storage level where as cache () uses the default storage level.

**What Are The Various Levels Of Persistence In Apache Spark?**

***Answer:*** Apache Spark automatically persists the intermediary data from various shuffle operations, however it is often suggested that users call persist () method on the RDD in case they plan to reuse it. Spark has various persistence levels to store the RDDs on disk or in memory or as a combination of both with different replication levels.

The various storage/persistence levels in Spark are:

MEMORY\_ONLY

MEMORY\_ONLY\_SER

MEMORY\_AND\_DISK

MEMORY\_AND\_DISK\_SER, DISK\_ONLY

OFF\_HEAP

**How Spark Handles Monitoring And Logging In Standalone Mode?**

***Answer:*** Spark has a web based user interface for monitoring the cluster in standalone mode that shows the cluster and job statistics. The log output for each job is written to the work directory of the slave nodes.

**Does Apache Spark Provide Check Pointing?**

***Answer:*** Lineage graphs are always useful to recover RDDs from a failure but this is generally time consuming if the RDDs have long lineage chains. Spark has an API for check pointing i.e. a REPLICATE flag to persist. However, the decision on which data to checkpoint - is decided by the user. Checkpoints are useful when the lineage graphs are long and have wide dependencies.

**How Can You Launch Spark Jobs Inside Hadoop Mapreduce?**

***Answer:*** Using SIMR (Spark in MapReduce) users can run any spark job inside MapReduce without requiring any admin rights.

**How Spark Uses Akka?**

***Answer:*** Spark uses Akka basically for scheduling. All the workers request for a task to master after registering. The master just assigns the task. Here Spark uses Akka for messaging between the workers and masters.

**How Can You Achieve High Availability In Apache Spark?**

***Answer:*** \*Implementing single node recovery with local file system

\*Using StandBy Masters with Apache ZooKeeper.

**Hadoop Uses Replication To Achieve Fault Tolerance. How Is This Achieved In Apache Spark?**

***Answer:*** Data storage model in Apache Spark is based on RDDs. RDDs help achieve fault tolerance through lineage. RDD always has the information on how to build from other datasets. If any partition of a RDD is lost due to failure, lineage helps build only that particular lost partition.

**Explain About The Core Components Of A Distributed Spark Application?**

***Answer:*** Driver: The process that runs the main () method of the program to create RDDs and perform transformations and actions on them.

Executor: The worker processes that run the individual tasks of a Spark job.

Cluster Manager: A pluggable component in Spark, to launch Executors and Drivers. The cluster manager allows Spark to run on top of other external managers like Apache Mesos or YARN.

**What Do You Understand By Lazy Evaluation?**

***Answer:*** Spark is intellectual in the manner in which it operates on data. When you tell Spark to operate on a given dataset, it heeds the instructions and makes a note of it, so that it does not forget - but it does nothing, unless asked for the final result.

When a transformation like map () is called on a RDD-the operation is not performed immediately. Transformations in Spark are not evaluated till you perform an action. This helps optimize the overall data processing workflow.

**Define A Worker Node?**

***Answer:*** A node that can run the Spark application code in a cluster can be called as a worker node. A worker node can have more than one worker which is configured by setting the SPARK\_ WORKER\_INSTANCES property in the spark-env.sh file. Only one worker is started if the SPARK\_ WORKER\_INSTANCES property is not defined.

**What Do You Understand By Schemardd?**

***Answer:*** An RDD that consists of row objects (wrappers around basic string or integer arrays) with schema information about the type of data in each column.

**What Are The Disadvantages Of Using Apache Spark Over Hadoop Mapreduce?**

***Answer:*** Apache spark does not scale well for compute intensive jobs and consumes large number of system resources. Apache Spark’s in-memory capability at times comes a major roadblock for cost efficient processing of big data. Also, Spark does have its own file management system and hence needs to be integrated with other cloud based data platforms or apache Hadoop.

**Is It Necessary To Install Spark On All The Nodes Of A Yarn Cluster While Running Apache Spark On Yarn ?**

***Answer:*** No , it is not necessary because Apache Spark runs on top of YARN.

**What Do You Understand By Executor Memory In A Spark Application?**

***Answer:*** Every spark application has same fixed heap size and fixed number of cores for a spark executor. The heap size is what referred to as the Spark executor memory which is controlled with the spark.executor.memory property of the –executor-memory flag.

Every spark application will have one executor on each worker node. The executor memory is basically a measure on how much memory of the worker node will the application utilize.

**What Does The Spark Engine Do?**

***Answer:*** Spark engine schedules, distributes and monitors the data application across the spark cluster.

**What Makes Apache Spark Good At Low-latency Workloads Like Graph Processing And Machine Learning?**

***Answer:*** Apache Spark stores data in-memory for faster model building and training. Machine learning algorithms require multiple iterations to generate a resulting optimal model and similarly graph algorithms traverse all the nodes and edges.

These low latency workloads that need multiple iterations can lead to increased performance. Less disk access and controlled network traffic make a huge difference when there is lots of data to be processed.

**HIVE**

**What Are The Different Types Of Tables Available In Hive?**

***Answer:*** There are two types. Managed table and external table. In managed table both the data and schema in under control of hive but in external table only the schema is under control of Hive.

**Is Hive Suitable To Be Used For Oltp Systems? Why?**

***Answer:*** No Hive does not provide insert and update at row level. So it is not suitable for OLTP system.

**Can A Table Be Renamed In Hive?**

***Answer:*** ALTER TABLE table\_name RENAME TO new\_name

Ex: ALTER TABLE student RENAME TO Employee

**Can We Change The Data Type Of A Column In A Hive Table?**

***Answer:*** Using REPLACE column option Ex: ALTER TABLE table\_name REPLACE COLUMNS ……

**OR** ALTER TABLE employee CHANGE salary new\_salary BIGINT;

**What Is A Metastore In Hive?**

***Answer:*** It is like a relational database storing the metadata of hive tables, partitions, schemas etc.

**What Is The Need For Custom Serde?**

***Answer:*** Depending on the nature of data the user has, the inbuilt SerDe may not satisfy the format of the data. SO users need to write their own java code to satisfy their data format requirements.

**Why Do We Need Hive?**

***Answer:*** Hive is a tool in Hadoop ecosystem which provides an interface to organize and query data in a database like fashion and write SQL like queries. It is suitable for accessing and analyzing data in Hadoop using SQL syntax.

**What Is the Default Location Where Hive Stores Table Data?**

***Answer:*** hdfs://namenode\_server/user/hive/warehouse

**Is There A Date Data Type In Hive?**

***Answer:*** Yes. The TIMESTAMP data types stores date in java.sql.timestamp format

**What Are The Three Different Modes In Which Hive Can Be Run?**

***Answer:*** 1. Local mode

2. Distributed mode

3. Pseudo distributed mode

**What Are Collection Data Types In Hive?**

***Answer:*** There are three collection data types in Hive.

1. ARRAY

2. MAP

3. STRUCT

**Can We Run Unix Shell Commands From Hive? Give Example?**

***Answer:*** Yes, using the ! mark just before the command.

For example !pwd at hive prompt will list the current directory.

**What Is A Hive Variable? What For We Use It?**

***Answer:*** The hive variable is variable created in the Hive environment that can be referenced by Hive scripts. It is used to pass some values to the hive queries when the query starts executing.

**Can Hive Queries Be Executed From Script Files? How?**

***Answer:*** Using the source command.

Example: Hive> source /path/to/file/file\_with\_query.hql

**What Are The Default Record And Field Delimiter Used For Hive Text Files?**

***Answer:*** The default record delimiter is − n

And the field delimiters are − 01,02,03

**What is the .hiverc File & Its Importance?**

***Answer:*** It is a file-containing list of commands needs to run when the hive CLI starts.

**OR** It is a file that is executed when you launch the hive shell - making it an ideal place for adding any hive configuration/customization you want set, on start of the hive shell. This could be:

- Setting column headers to be visible in query results

- Making the current database name part of the hive prompt

- Adding any jars or files

- Registering UDFs

**.hiverc file location**

The file is loaded from the hive conf directory.

I have the CDH4.2 distribution and the location is: /etc/hive/conf.cloudera.hive1

If the file does not exist, you can create it.

It needs to be deployed to every node from where you might launch the Hive shell.

**What Do You Mean By Schema On Read?**

***Answer:*** The schema is validated with the data when reading the data and not enforced when writing data.

**How Do You List All Databases Whose Name Starts With P?**

***Answer:*** SHOW DATABASES LIKE ‘p.\*’

**What Does The “use” Command In Hive Do?**

***Answer:*** With the use command you fix the database on which all the subsequent hive queries will run.

**How Can You Delete The Dbproperty In Hive?**

***Answer:*** There is no way you can delete the DBPROPERTY.

**Which Java Class Handles The Input Record Encoding Into Files Which Store The Tables In Hive?**

***Answer:*** org.apache.hadoop.mapred.TextInputFormat

**What Is the Significance Of The Line Set Hive.mapred.mode = Strict;**

***Answer:*** It sets the mapreduce jobs to strict mode.By which the queries on partitioned tables can not run without a WHERE clause. This prevents very large job running for long time.

**How Do You Check If A Particular Partition Exists?**

***Answer:*** This can be done with following query

SHOW PARTITIONS table\_name PARTITION(partitioned\_column=’partition\_value’)

**Which Java Class Handles The Output Record Encoding Into Files Which Result From Hive Queries?**

***Answer:*** org.apache.hadoop.hive.ql.io.HiveIgnoreKeyTextOutputFormat

**What Is The Significance Of ‘if Exists” Clause While Dropping A Table?**

***Answer:*** When we issue the command DROP TABLE IF EXISTS table\_name

Hive throws an error if the table being dropped does not exist in the first place.

**When You Point A Partition Of A Hive Table To A New Directory, What Happens To The Data?**

***Answer:*** The data stays in the old location. It has to be moved manually.

**Write A Query To Insert A New Column(new\_col Int) Into A Hive Table (htab) At A Position Before An Existing Column (x\_col)**

***Answer:*** ALTER TABLE table\_name

CHANGE COLUMN new\_col INT

BEFORE x\_col

**Does The Archiving Of Hive Tables Give Any Space Saving In Hdfs?**

***Answer:*** No. It only reduces the number of files which becomes easier for namenode to manage.

**How Can You Stop A Partition Form Being Queried?**

***Answer:*** By using the ENABLE OFFLINE clause with ALTER TABLE statement.

**While Loading Data Into A Hive Table Using The Load Data Clause, How Do You Specify It Is A Hdfs File And Not A Local File ?**

***Answer:*** By Omitting the LOCAL CLAUSE in the LOAD DATA statement.

**If You Omit The Overwrite Clause While Creating A Hive Table,what Happens To File Which Are New Files And Which are Already Exist?**

***Answer:*** The new incoming files are just added to the target directory and the existing files are simply overwritten. Other files whose name does not match any of the incoming files will continue to exist.

If you add the OVERWRITE clause then all the existing data in the directory will be deleted before new data is written.

**What Does The Following Query Do? Insert Overwrite Table Employees Partition (country, State) Select ..., Se.cnty, Se.st From Staged\_employees Se;**

***Answer:*** It creates partition on table employees with partition values coming from the columns in the select clause. It is called Dynamic partition insert.

**What Is A Table Generating Function in Hive?**

***Answer:*** A table generating function is a function which takes a single column as argument and expands it to multiple column or rows. Example exploe()

**How Can Hive Avoid Mapreduce?**

***Answer:*** If we set the property hive.exec.mode.local.auto to true then hive will avoid mapreduce to fetch query results.

**What Is the Difference Between Like And Rlike Operators In Hive?**

***Answer:*** The LIKE operator behaves the same way as the regular SQL operators used in select queries.

Example − street\_name RLIKE ‘.\*(Chi|Oho).\*’ which will select any word which has either chi or oho in it.

**OR** It helps to examine the two substring i.e if the substring of A matches with B then it define to True.

**Is It Possible To Create Cartesian Join Between 2 Tables, Using Hive?**

***Answer:*** No. As this kind of Join cannot be implemented in mapreduce

**As Part Of Optimizing The Queries In Hive, What Should Be The Order Of Table Size In A Join Query?**

***Answer:*** In a join query the smallest table to be taken in the first position and largest table should be taken in the last position.

**What Is The Usefulness Of The Distributed By Clause In Hive?**

***Answer:*** It controls how the map output is reduced among the reducers. It is useful in case of streaming data.

**How Will You Convert The String ’51.2’ To A Float Value In The Price Column?**

***Answer:*** Select cast(price as FLOAT)

**What Will Be The Result When You Do Cast(‘abc’ As Int)?**

***Answer:*** Hive will return NULL

**Can The Name Of A View Be Same As The Name Of A Hive Table?**

***Answer:*** No. The name of a view must be unique when compared to all other tables and views present in the same database.

**Can We Load Data Into A View?**

***Answer:*** No. A view cannot be the target of a INSERT or LOAD statement.

**What Types Of Costs Are Associated In Creating Index On Hive Tables?**

***Answer:*** Indexes occupies space and there is a processing cost in arranging the values of the column on which index is created.

**Give The Command To See The Indexes On A Table?**

***Answer:*** SHOW INDEX ON table\_name

This will list all the indexes created on any of the columns in the table table\_name.

**What Is Bucketing?**

***Answer:*** The values in a column are hashed into a number of buckets which is defined by user. It is a way to avoid too many partitions or nested partitions while ensuring optimizes query output.

**What Does /\*streamtable(table\_name)\*/ Do?**

*Answer:* It is query hint to stream a table into memory before running the query. It is a query optimization Technique.

**Can A Partition Be Archived? What Are The Advantages And Disadvantages?**

*Answer:* Yes. A partition can be archived. Advantage is it decreases the number of files stored in namenode and the archived file can be queried using hive. The disadvantage is it will cause less efficient query and does not offer any space savings.

**What Is A Generic Udf In Hive?**

***Answer:*** It is a UDF which is created using a java program to server some specific need not covered under the existing functions in Hive. It can detect the type of input argument programmatically and provide appropriate response.

**The Following Statement Failed To Execute. What Can Be The Cause? Load Data Local Inpath ‘${env:home}/country/state/’ Overwrite Into Table Address;**

***Answer:*** The local in path should contain a file and not a directory. The $env:HOME is a valid variable available in the hive environment.

**How Do You Specify The Table Creator Name When Creating A Table In Hive?**

***Answer:*** The TBLPROPERTIES clause is used to add the creator name while creating a table.

The TBLPROPERTIES is added like: TBLPROPERTIES(‘creator’= ‘Joan’)

**IMPALA**

**How Do I Try Impala Out?**

***Answer:*** To look at the core features and functionality on Impala, the easiest way to try out Impala is to download the Cloudera Quick Start VM and start the Impala service through Cloudera Manager, then use impala-shell in a terminal window or the Impala Query UI in the Hue web interface.

To do performance testing and try out the management features for Impala on a cluster, you need to move beyond the Quick Start VM with its virtualized single-node environment. Ideally, download the Cloudera Manager software to set up the cluster, then install the Impala software through Cloudera Manager.

**Does Cloudera Offer A Vm For Demonstrating Impala?**

***Answer:*** Cloudera offers a demonstration VM called the QuickStart VM, available in VMWare, VirtualBox, and KVM formats. For more information, see the Cloudera QuickStart VM. After booting the QuickStart VM, many services are turned off by default; in the Cloudera Manager UI that appears automatically, turn on Impala and any other components that you want to try out.

**Where Can I Find Impala Documentation?**

***Answer:*** Starting with Impala 1.3.0, Impala documentation is integrated with the CDH 5 documentation, in addition to the standalone Impala documentation for use with CDH 4. For CDH 5, the core Impala developer and administrator information remains in the associated Impala documentation portion. Information about Impala release notes, installation, configuration, startup, and security is embedded in the corresponding CDH 5 guides.

\*New features

\*Known and fixed issues

\*Incompatible changes

\*Installing Impala

\*Upgrading Impala

\*Configuring Impala

\*Starting Impala

\*Security for Impala

\*CDH Version and Packaging Information

**Where Can I Get Sample Data To Try?**

***Answer:*** You can get scripts that produce data files and set up an environment for TPC-DS style benchmark tests from this Github repository. In addition to being useful for experimenting with performance, the tables are suited to experimenting with many aspects of SQL on Impala: they contain a good mixture of data types, data distributions, partitioning, and relational data suitable for join queries.

**How Much Memory Is Required?**

***Answer:*** Although Impala is not an in-memory database, when dealing with large tables and large result sets, you should expect to dedicate a substantial portion of physical memory for the impalad daemon. Recommended physical memory for an Impala node is 128 GB or higher. If practical, devote approximately 80% of physical memory to Impala.

The amount of memory required for an Impala operation depends on several factors:

\* The file format of the table. Different file formats represent the same data in more or fewer data files. The compression and encoding for each file format might require a different amount of temporary memory to decompress the data for analysis.

\* Whether the operation is a SELECT or an INSERT. For example, Parquet tables require relatively little memory to query, because Impala reads and decompresses data in 8 MB chunks. Inserting into a Parquet table is a more memory-intensive operation because the data for each data file (potentially hundreds of megabytes, depending on the value of the PARQUET\_FILE\_SIZE query option) is stored in memory until encoded, compressed, and written to disk.

\* Whether the table is partitioned or not, and whether a query against a partitioned table can take advantage of partition pruning.

\* Whether the final result set is sorted by the ORDER BY clause. Each Impala node scans and filters a portion of the total data, and applies the LIMIT to its own portion of the result set. In Impala 1.4.0 and higher, if the sort operation requires more memory than is available on any particular host, Impala uses a temporary disk work area to perform the sort.

The intermediate result sets are all sent back to the coordinator node, which does the final sorting and then applies the LIMIT clause to the final result set.

For example, if you execute the query:

select \* from giant\_table order by some\_column limit 1000;

and your cluster has 50 nodes, then each of those 50 nodes will transmit a maximum of 1000 rows back to the coordinator node. The coordinator node needs enough memory to sort (LIMIT \* cluster\_size) rows, although in the end the final result set is at most LIMIT rows, 1000 in this case.

Likewise, if you execute the query:

select \* from giant\_table where test\_val > 100 order by some\_column;

then each node filters out a set of rows matching the WHERE conditions, sorts the results (with no size limit), and sends the sorted intermediate rows back to the coordinator node. The coordinator node might need substantial memory to sort the final result set, and so might use a temporary disk work area for that final phase of the query.

\* Whether the query contains any join clauses, GROUP BY clauses, analytic functions, or DISTINCT operators. These operations all require some in-memory work areas that vary depending on the volume and distribution of data. In Impala 2.0 and later, these kinds of operations utilize temporary disk work areas if memory usage grows too large to handle.

\* The size of the result set. When intermediate results are being passed around between nodes, the amount of data depends on the number of columns returned by the query. For example, it is more memory-efficient to query only the columns that are actually needed in the result set rather than always issuing SELECT \*.

\* The mechanism by which work is divided for a join query. You use the COMPUTE STATS statement, and query hints in the most difficult cases, to help Impala pick the most efficient execution plan.

**What Are The Main Features Of Impala?**

***Answer:*** \* A large set of SQL statements, including SELECT and INSERT, with joins, Subqueries in Impala SELECT Statements, and Impala Analytic Functions. Highly compatible with HiveQL, and also including some vendor extensions. For more information.

\* Distributed, high-performance queries.

\* Using Cloudera Manager, you can deploy and manage your Impala services. Cloudera Manager is the best way to get started with Impala on your cluster.

\* Using Hue for queries.

\* Appending and inserting data into tables through the INSERT statement.

\* ODBC: Impala is certified to run against MicroStrategy and Tableau, with restrictions. For more information, see Configuring Impala to Work with ODBC.

\* Querying data stored in HDFS and HBase in a single query.

\* In Impala 2.2.0 and higher, querying data stored in the Amazon Simple Storage Service (S3).

\* Concurrent client requests. Each Impala daemon can handle multiple concurrent client requests. The effects on performance depend on your particular hardware and workload.

\* Kerberos authentication. For more information.

\* Partitions. With Impala SQL, you can create partitioned tables with the CREATE TABLE statement, and add and drop partitions with the ALTER TABLE statement. Impala also takes advantage of the partitioning present in Hive tables.

**What Features From Relational Databases Or Hive Are Not Available In Impala?**

***Answer:*** \* Querying streaming data.

\* Deleting individual rows. You delete data in bulk by overwriting an entire table or partition, or by dropping a table.

\* Indexing (not currently). LZO-compressed text files can be indexed outside of Impala, as described in Using LZO-Compressed Text Files.

\* Full text search on text fields. The Cloudera Search product is appropriate for this use case.

\* Custom Hive Serializer/Deserializer classes (SerDes). Impala supports a set of common native file formats that have built-in SerDes in CDH.

\* Check pointing within a query. That is, Impala does not save intermediate results to disk during long-running queries. Currently, Impala cancels a running query if any host on which that query is executing fails.

When one or more hosts are down, Impala reroutes future queries to only use the available hosts, and Impala detects when the hosts come back up and begins using them again. Because a query can be submitted through any Impala node, there is no single point of failure.

In the future, we will consider adding additional work allocation features to Impala, so that a running query would complete even in the presence of host failures.

\* Encryption of data transmitted between Impala daemons.

\* Hive indexes.

\* Non-Hadoop data stores, such as relational databases.

**Does Impala Support Generic JDBC?**

***Answer:*** Impala supports the HiveServer2 JDBC driver.

**Is Avro Supported?**

***Answer:*** Yes, Avro is supported. Impala has always been able to query Avro tables. You can use the Impala LOAD DATA statement to load existing Avro data files into a table. Starting with Impala 1.4, you can create Avro tables with Impala. Currently, you still use the INSERT statement in Hive to copy data from another table into an Avro table.

**How Do I Know How Many Impala Nodes Are In My Cluster?**

***Answer:*** The Impala statestore keeps track of how many impala nodes are currently available. You can see this information through the statestore web interface. For example, at the URL http://statestore\_host:25010/metrics you might see lines like the following:

statestore.live-backends:3

statestore.live-backends.list:[host1:22000, host1:26000, host2:22000]

The number of impalad nodes is the number of list items referring to port 22000, in this case two. (Typically, this number is one less than the number reported by the statestore.live-backends line.) If an impalad node became unavailable or came back after an outage, the information reported on this page would change appropriately.

**Are Results Returned As They Become Available, Or All At Once When A Query Completes?**

***Answer:*** Impala streams results whenever they are available, when possible. Certain SQL operations (aggregation or ORDER BY) require all of the input to be ready before Impala can return results.

**Why Does My Select Statement Fail?**

***Answer:*** When a SELECT statement fails, the cause usually falls into one of the following categories:

\* A timeout because of a performance, capacity, or network issue affecting one particular node.

\* Excessive memory use for a join query, resulting in automatic cancellation of the query.

\* A low-level issue affecting how native code is generated on each node to handle particular WHERE clauses in the query. For example, a machine instruction could be generated that is not supported by the processor of a certain node. If the error message in the log suggests the cause was an illegal instruction, consider turning off native code generation temporarily, and trying the query again.

\* Malformed input data, such as a text data file with an enormously long line, or with a delimiter that does not match the character specified in the FIELDS TERMINATED BY clause of the CREATE TABLE statement.

**Why Does My Insert Statement Fail?**

***Answer:*** When an INSERT statement fails, it is usually the result of exceeding some limit within a Hadoop component, typically HDFS.

\* An INSERT into a partitioned table can be a strenuous operation due to the possibility of opening many files and associated threads simultaneously in HDFS. Impala 1.1.1 includes some improvements to distribute the work more efficiently, so that the values for each partition are written by a single node, rather than as a separate data file from each node.

\* Certain expressions in the SELECT part of the INSERT statement can complicate the execution planning and result in an inefficient INSERT operation. Try to make the column data types of the source and destination tables match up, for example by doing ALTER TABLE ... REPLACE COLUMNS on the source table if necessary. Try to avoid CASE expressions in the SELECT portion, because they make the result values harder to predict than transferring a column unchanged or passing the column through a built-in function.

\* Be prepared to raise some limits in the HDFS configuration settings, either temporarily during the INSERT or permanently if you frequently run such INSERT statements as part of your ETL pipeline.

\* The resource usage of an INSERT statement can vary depending on the file format of the destination table. Inserting into a Parquet table is memory-intensive, because the data for each partition is buffered in memory until it reaches 1 gigabyte, at which point the data file is written to disk. Impala can distribute the work for an INSERT more efficiently when statistics are available for the source table that is queried during the INSERT statement.

**Does Impala Performance Improve As It Is Deployed To More Hosts In A Cluster In Much The Same Way That Hadoop Performance Does?**

***Answer:*** Yes. Impala scales with the number of hosts. It is important to install Impala on all the DataNodes in the cluster, because otherwise some of the nodes must do remote reads to retrieve data not available for local reads. Data locality is an important architectural aspect for Impala performance.

**Is The Hdfs Block Size Reduced To Achieve Faster Query Results?**

***Answer:*** No. Impala does not make any changes to the HDFS or HBase data sets.

The default Parquet block size is relatively large (256 MB in Impala 2.0 and later; 1 GB in earlier releases). You can control the block size when creating Parquet files using the PARQUET\_FILE\_SIZE query option.

**Does Impala Use Caching?**

***Answer:*** Impala does not cache table data. It does cache some table and file metadata. Although queries might run faster on subsequent iterations because the data set was cached in the OS buffer cache, Impala does not explicitly control this.

Impala takes advantage of the HDFS caching feature in CDH 5. You can designate which tables or partitions are cached through the CACHED and UNCACHED clauses of the CREATE TABLE and ALTER TABLE statements. Impala can also take advantage of data that is pinned in the HDFS cache through the hdfscacheadmin command.

**What Are Good Use Cases For Impala As Opposed To Hive Or Mapreduce?**

***Answer:*** Impala is well-suited to executing SQL queries for interactive exploratory analytics on large data sets. Hive and MapReduce are appropriate for very long running, batch-oriented tasks such as ETL.

**Is Mapreduce Required For Impala? Will Impala Continue To Work As Expected If Mapreduce Is Stopped?**

***Answer:*** Impala does not use MapReduce at all.

**Can Impala Be Used For Complex Event Processing?**

***Answer:*** For example, in an industrial environment, many agents may generate large amounts of data. Can Impala be used to analyze this data, checking for notable changes in the environment?

Complex Event Processing (CEP) is usually performed by dedicated stream-processing systems. Impala is not a stream-processing system, as it most closely resembles a relational database.

**Is Impala Intended to Handle Real Time Queries In Low-latency Applications Or Is It For Ad Hoc Queries For The Purpose Of Data Exploration?**

***Answer:*** Ad-hoc queries are the primary use case for Impala. We anticipate it being used in many other situations where low-latency is required. Whether Impala is appropriate for any particular use-case depends on the workload, data size and query volume.

**How Does Impala Compare To Hive And Pig?**

***Answer:*** Impala is different from Hive and Pig because it uses its own daemons that are spread across the cluster for queries. Because Impala does not rely on MapReduce, it avoids the startup overhead of MapReduce jobs, allowing Impala to return results in real time.

**Can I Do Transforms or Add New Functionality?**

***Answer:*** Impala adds support for UDFs in Impala 1.2. You can write your own functions in C++, or reuse existing Java-based Hive UDFs. The UDF support includes scalar functions and user-defined aggregate functions (UDAs). User-defined table functions (UDTFs) are not currently supported.

Impala does not currently support an extensible serialization-deserialization framework (SerDes), and so adding extra functionality to Impala is not as straightforward as for Hive or Pig.

**Can Any Impala Query Also Be Executed In Hive?**

***Answer:*** Yes. There are some minor differences in how some queries are handled, but Impala queries can also be completed in Hive. Impala SQL is a subset of HiveQL, with some functional limitations such as transforms.

**Can I Use Impala To Query Data Already Loaded Into Hive And Hbase?**

***Answer:*** There are no additional steps to allow Impala to query tables managed by Hive, whether they are stored in HDFS or HBase. Make sure that Impala is configured to access the Hive metastore correctly and you should be ready to go. Keep in mind that impalad, by default, runs as the impala user, so you might need to adjust some file permissions depending on how strict your permissions are currently.

**Is Hive An Impala Requirement?**

***Answer:*** The Hive metastore service is a requirement. Impala shares the same metastore database as Hive, allowing Impala and Hive to access the same tables transparently.

Hive itself is optional, and does not need to be installed on the same nodes as Impala. Currently, Impala supports a wider variety of read (query) operations than write (insert) operations; you use Hive to insert data into tables that use certain file formats.

**Is Impala Production Ready?**

***Answer:*** Impala has finished its beta release cycle, and the 1.0, 1.1, and 1.2 GA releases are production ready. The 1.1.x series includes additional security features for authorization, an important requirement for production use in many organizations. The 1.2.x series includes important performance features, particularly for large join queries. Some Cloudera customers are already using Impala for large workloads.

The Impala 1.3.0 and higher releases are bundled with corresponding levels of CDH 5. The number of new features grows with each release.

**How Do I Configure Hadoop High Availability (ha) For Impala?**

***Answer:*** You can set up a proxy server to relay requests back and forth to the Impala servers, for load balancing and high availability.

**What Happens If There Is An Error In Impala?**

***Answer:*** There is not a single point of failure in Impala. All Impala daemons are fully able to handle incoming queries. If a machine fails however, all queries with fragments running on that machine will fail. Because queries are expected to return quickly, you can just rerun the query if there is a failure.

The longer Answer Impala must be able to connect to the Hive metastore. Impala aggressively caches metadata so the metastore host should have minimal load. Impala relies on the HDFS NameNode, and, in CDH4, you can configure HA for HDFS. Impala also has centralized services, known as the statestore andcatalog services, that run on one host only. Impala continues to execute queries if the statestore host is down, but it will not get state updates. For example, if a host is added to the cluster while the statestore host is down, the existing instances of impalad running on the other hosts will not find out about this new host. Once the statestore process is restarted, all the information it serves is automatically reconstructed from all running Impala daemons.

**What Is The Maximum Number Of Rows In A Table?**

***Answer:*** There is no defined maximum. Some customers have used Impala to query a table with over a trillion rows.

**On Which Hosts Does Impala Run?**

***Answer:*** Cloudera strongly recommends running the impalad daemon on each DataNode for good performance. Although this topology is not a hard requirement, if there are data blocks with no Impala daemons running on any of the hosts containing replicas of those blocks, queries involving that data could be very inefficient. In that case, the data must be transmitted from one host to another for processing by "remote reads", a condition Impala normally tries to avoid.

**How Are Joins Performed In Impala?**

***Answer:*** By default, Impala automatically determines the most efficient order in which to join tables using a cost-based method, based on their overall size and number of rows. (This is a new feature in Impala 1.2.2 and higher.) The COMPUTE STATS statement gathers information about each table that is crucial for efficient join performance. Impala chooses between two techniques for join queries, known as "broadcast joins" and "partitioned joins".

**How Does Impala Process Join Queries For Large Tables?**

***Answer:*** Impala utilizes multiple strategies to allow joins between tables and result sets of various sizes. When joining a large table with a small one, the data from the small table is transmitted to each node for intermediate processing. When joining two large tables, the data from one of the tables is divided into pieces, and each node processes only selected pieces.

**What Is Impala's Aggregation Strategy?**

***Answer:*** Impala currently only supports in-memory hash aggregation. In Impala 2.0 and higher, if the memory requirements for a join or aggregation operation exceed the memory limit for a particular host, Impala uses a temporary work area on disk to help the query complete successfully.

**How Is Impala Metadata Managed?**

***Answer:*** Impala uses two pieces of metadata: the catalog information from the Hive metastore and the file metadata from the NameNode. Currently, this metadata is lazily populated and cached when an impalad needs it to plan a query.

The REFRESH statement updates the metadata for a particular table after loading new data through Hive. The INVALIDATE METADATA Statement statement refreshes all metadata, so that Impala recognizes new tables or other DDL and DML changes performed through Hive.

In Impala 1.2 and higher, a dedicated catalogd daemon broadcasts metadata changes due to Impala DDL or DML statements to all nodes, reducing or eliminating the need to use the REFRESH and INVALIDATE METADATAstatements.

**What Load Do Concurrent Queries Produce On The Namenode?**

***Answer:*** The load Impala generates is very similar to MapReduce. Impala contacts the NameNode during the planning phase to get the file metadata (this is only run on the host the query was sent to). Every impaladwill read files as part of normal processing of the query.

**How Does Impala Achieve Its Performance Improvements?**

***Answer:*** These are the main factors in the performance of Impala versus that of other Hadoop components and related technologies.

Impala avoids MapReduce. While MapReduce is a great general parallel processing model with many benefits, it is not designed to execute SQL. Impala avoids the inefficiencies of MapReduce in these ways:

\* Impala does not materialize intermediate results to disk. SQL queries often map to multiple MapReduce jobs with all intermediate data sets written to disk.

\* Impala avoids MapReduce start-up time. For interactive queries, the MapReduce start-up time becomes very noticeable. Impala runs as a service and essentially has no start-up time.

\* Impala can more naturally disperse query plans instead of having to fit them into a pipeline of map and reduce jobs. This enables Impala to parallelize multiple stages of a query and avoid overheads such as sort and shuffle when unnecessary.

Impala uses a more efficient execution engine by taking advantage of modern hardware and technologies:

\* Impala generates runtime code. Impala uses LLVM to generate assembly code for the query that is being run. Individual queries do not have to pay the overhead of running on a system that needs to be able to execute arbitrary queries.

\* Impala uses available hardware instructions when possible. Impala uses the supplemental SSE3 (SSSE3) instructions which can offer tremendous speedups in some cases. (Impala 2.0 and 2.1 required the SSE4.1 instruction set; Impala 2.2 and higher relax the restriction again so only SSSE3 is required.)

\* Impala uses better I/O scheduling. Impala is aware of the disk location of blocks and is able to schedule the order to process blocks to keep all disks busy.

\* Impala is designed for performance. A lot of time has been spent in designing Impala with sound performance-oriented fundamentals, such as tight inner loops, inlined function calls, minimal branching, better use of cache, and minimal memory usage.

**What Happens When The Data Set Exceeds Available Memory?**

***Answer:*** Currently, if the memory required to process intermediate results on a node exceeds the amount available to Impala on that node, the query is cancelled. You can adjust the memory available to Impala on each node, and you can fine-tune the join strategy to reduce the memory required for the biggest queries. We do plan on supporting external joins and sorting in the future.

Keep in mind though that the memory usage is not directly based on the input data set size. For aggregations, the memory usage is the number of rows after grouping. For joins, the memory usage is the combined size of the tables excluding the biggest table, and Impala can use join strategies that divide up large joined tables among the various nodes rather than transmitting the entire table to each node.

**What Are the Most Memory-intensive Operations?**

***Answer:*** If a query fails with an error indicating "memory limit exceeded", you might suspect a memory leak. The problem could actually be a query that is structured in a way that causes Impala to allocate more memory than you expect, exceeded the memory allocated for Impala on a particular node. Some examples of query or table structures that are especially memory-intensive are:

\* INSERT statements using dynamic partitioning, into a table with many different partitions. (Particularly for tables using Parquet format, where the data for each partition is held in memory until it reaches the full block size in size before it is written to disk.)

Consider breaking up such operations into several different INSERT statements, for example to load data one year at a time rather than for all years at once.

\* GROUP BY on a unique or high-cardinality column. Impala allocates some handler structures for each different value in a GROUP BY query. Having millions of different GROUP BY values could exceed the memory limit.

\* Queries involving very wide tables, with thousands of columns, particularly with many STRING columns. Because Impala allows a STRING value to be up to 32 KB, the intermediate results during such queries could require substantial memory allocation.

**When Does Impala Hold On To Or Return Memory?**

***Answer:*** Impala allocates memory using tcmalloc, a memory allocator that is optimized for high concurrency. Once Impala allocates memory, it keeps that memory reserved to use for future queries. Thus, it is normal for Impala to show high memory usage when idle. If Impala detects that it is about to exceed its memory limit (defined by the -mem\_limit startup option or the MEM\_LIMIT query option), it deallocates memory not needed by the current queries.

When issuing queries through the JDBC or ODBC interfaces, make sure to call the appropriate close method afterwards. Otherwise, some memory associated with the query is not freed.

**Is There An Update Statement?**

***Answer:*** Impala does not currently have an UPDATE statement, which would typically be used to change a single row, a small group of rows, or a specific column. The HDFS-based files used by typical Impala queries are optimized for bulk operations across many megabytes of data at a time, making traditional UPDATEoperations inefficient or impractical.

You can use the following techniques to achieve the same goals as the familiar UPDATE statement, in a way that preserves efficient file layouts for subsequent queries:

\* Replace the entire contents of a table or partition with updated data that you have already staged in a different location, either using INSERT OVERWRITE, LOAD DATA, or manual HDFS file operations followed by a REFRESH statement for the table. Optionally, you can use built-in functions and expressions in the INSERTstatement to transform the copied data in the same way you would normally do in an UPDATE statement, for example to turn a mixed-case string into all uppercase or all lowercase.

\* To update a single row, use an HBase table, and issue an INSERT ... VALUES statement using the same key as the original row. Because HBase handles duplicate keys by only returning the latest row with a particular key value, the newly inserted row effectively hides the previous one.

**Can Impala Do User-defined Functions (udfs)?**

***Answer:*** Impala 1.2 and higher does support UDFs and UDAs. You can either write native Impala UDFs and UDAs in C++, or reuse UDFs (but not UDAs) originally written in Java for use with Hive.

**Why Do I Have To Use Refresh And Invalidate Metadata, What Do They Do?**

***Answer:*** In Impala 1.2 and higher, there is much less need to use the REFRESH and INVALIDATE METADATA statements:

\* The new impala-catalog service, represented by the catalogd daemon, broadcasts the results of Impala DDL statements to all Impala nodes. Thus, if you do a CREATE TABLE statement in Impala while connected to one node, you do not need to do INVALIDATE METADATA before issuing queries through a different node.

\* The catalog service only recognizes changes made through Impala, so you must still issue a REFRESHstatement if you load data through Hive or by manipulating files in HDFS, and you must issue an INVALIDATE METADATA statement if you create a table, alter a table, add or drop partitions, or do other DDL statements in Hive.

\* Because the catalog service broadcasts the results of REFRESH and INVALIDATE METADATA statements to all nodes, in the cases where you do still need to issue those statements, you can do that on a single node rather than on every node, and the changes will be automatically recognized across the cluster, making it more convenient to load balance by issuing queries through arbitrary Impala nodes rather than always using the same coordinator node.

Why Is Space Not Freed Up When I Issue Drop Table?

***Answer:*** Impala deletes data files when you issue a DROP TABLE on an internal table, but not an external one. By default, the CREATE TABLE statement creates internal tables, where the files are managed by Impala. An external table is created with a CREATE EXTERNAL TABLE statement, where the files reside in a location outside the control of Impala. Issue a DESCRIBE FORMATTED statement to check whether a table is internal or external.

The keyword MANAGED\_TABLE indicates an internal table, from which Impala can delete the data files. The keyword EXTERNAL\_TABLE indicates an external table, where Impala will leave the data files untouched when you drop the table.

Even when you drop an internal table and the files are removed from their original location, you might not get the hard drive space back immediately. By default, files that are deleted in HDFS go into a special trashcan directory, from which they are purged after a period of time (by default, 6 hours). For background information on the trashcan mechanism.

**Is There A Dual Table?**

***Answer:*** You might be used to running queries against a single-row table named DUAL to try out expressions, built-in functions, and UDFs. Impala does not have a DUAL table. To achieve the same result, you can issue a SELECTstatement without any table name:

select 2+2;

select substr('hello',2,1);

select pow(10,6);45.

**How Do I Load A Big Csv File Into A Partitioned Table?**

***Answer:*** To load a data file into a partitioned table, when the data file includes fields like year, month, and so on that correspond to the partition key columns, use a two-stage process. First, use the LOAD DATA or CREATE EXTERNAL TABLE statement to bring the data into an unpartitioned text table. Then use an INSERT ... SELECT statement to copy the data from the unpartitioned table to a partitioned one. Include a PARTITION clause in the INSERTstatement to specify the partition key columns.

**Can I Do Insert ... Select \* Into A Partitioned Table?**

***Answer:*** When you use the INSERT ... SELECT \* syntax to copy data into a partitioned table, the columns corresponding to the partition key columns must appear last in the columns returned by the SELECT \*. You can create the table with the partition key columns defined last. Or, you can use the CREATE VIEW statement to create a view that reorders the columns: put the partition key columns last, then do the INSERT ... SELECT \* from the view.

**What Kinds Of Impala Queries Or Data Are Best Suited For Hbase?**

***Answer:*** HBase tables are ideal for queries where normally you would use a key-value store. That is, where you retrieve a single row or a few rows, by testing a special unique key column using the = or IN operators.

HBase tables are not suitable for queries that produce large result sets with thousands of rows. HBase tables are also not suitable for queries that perform full table scans because the WHERE clause does not request specific values from the unique key column.

Use HBase tables for data that is inserted one row or a few rows at a time, such as by the INSERT ... VALUESsyntax. Loading data piecemeal like this into an HDFS-backed table produces many tiny files, which is a very inefficient layout for HDFS data files.

If the lack of an UPDATE statement in Impala is a problem for you, you can simulate single-row updates by doing an INSERT ... VALUES statement using an existing value for the key column. The old row value is hidden; only the new row value is seen by queries.

HBase tables are often wide (containing many columns) and sparse (with most column values NULL). For example, you might record hundreds of different data points for each user of an online service, such as whether the user had registered for an online game or enabled particular account features. With Impala and HBase, you could look up all the information for a specific customer efficiently in a single query. For any given customer, most of these columns might be NULL, because a typical customer might not make use of most features of an online service.

**Ambari**

**What Are The Three Layers Where The Hadoop Components Are Actually Supported By Ambari?**

***Answer:*** The three layers that are supported by Ambari are below:

\* Core Hadoop

\* Essential Hadoop

\* Hadoop Support

**What Is Apache Ambari?**

***Answer:*** The Apache Ambari is nothing but a project which is solely focused to make life simple while using Hadoop management system.

This software helps or provides comfort zone in terms of the following aspect:

\* Provisioning

\* Managing

\* Monitoring Hadoop clusters

\* Provides intuitive interface

\* It is backed up RESTful API’s.

\* Provides an easy to use Hadoop management web UI

**What Are The Areas Where Ambari Helps The System Administrators To Do?**

***Answer:*** With the help of Ambari, system administrators will be able to do the following easily, they are:

\* Provision of Hadoop Cluster

\* Manage a Hadoop cluster

\* Monitor a Hadoop Cluster

**What Bit Version That Ambari Needs And Also List Out The Operating Systems That Are Compatible?**

***Answer:*** The Apache Ambari is compatible with 64-bit version and the following are the operating systems that go well with Ambari implementation:

\* Debian 7

\* Ubuntu 12 and 14

\* SLES (Suse Linux Enterprise Server) 11

\* OEL (Oracle Enterprise Linux 6) and 7

\* CentOS 6 and 7

\* RHEL ( Redhat Enterprise Linux) 6 and 7

**What Is The Latest Version Of Ambari That Is Available In The Market And What Is The Feature That They Have Added In It?**

***Answer:*** The latest version of Ambari that is available in the market is Ambari 2.5.2. Within, this version they have added a feature called: Cross stack upgrade support.

**What Is Repository?**

***Answer:*** A repository is nothing but space where it hosts the software packages which can be used for download and plus install.

What Is Yum?

***Answer:*** The Yum is nothing but a package manager which actually fetches the software packages from the repository.

On RHEL/CentOS, typically “yum”,

ON SLES, typically “Zipper”.

**What Is A Local Repository And When It Is Useful While Using Ambari Environment?**

***Answer:*** A local repository is nothing but a hosted space in the local environment. Usually, when the machines don't have an active internet connection or have restricted or very limited network access a local repository should be set up. With this setup, the user will be able to obtain Ambari and HDP software packages.

**How Many Types Of Ambari Repositories Are Available?**

***Answer:*** The types of Ambari Repositories are listed below:

\* Ambari: This is for Ambari server, Ambari agent and other monitoring software packages

\* HDP: This is used to host Hadoop Stack packages

\* HDP-UTILS: All the utility packages for Ambari and HDP are available

\* EPEL: It stands for “Extra Packages for Enterprise Linux. It has a set of additional packages for the Enterprise Linux

**What Are The Different Methods To Set Up Local Repositories?**

***Answer:*** There are two ways to deploy the local repositories. It actually depends on your active Internet connection and based on that we can execute it.

\* First of all mirror the packages to the local repository

\* If the first method doesn’t work out good for you then download all the Repository Tarball and start building the Local repository

**How to Set Up Local Repository Manually?**

***Answer:*** This process is only used when there is no active internet connection is not available.

So to set up a local repository, please follow the below steps:

\* First and foremost, set up a host with Apache httpd

\* Next is to download Tarball copy for every repository entire contents

\*Once it is downloaded, one has to extract the contents

**What Are The Tools That Are Needed To Build Ambari?**

***Answer:*** The following tools are needed to build Ambari:

\* If you are using Mac then you have to download Xcode from the apple store.

\* JDK 7

\* Apache Maven 3.3.9 or later

\* Python 2.6 or later

\* Node JS

\* G++

**What Are The Independent Extensions That Are Contributed To The Ambari Codebase?**

***Answer:*** The independent extensions that are contributed to the Ambari Codebase are as follows:

\* Ambari SCOM Management Pack

\* Apache Slider View

**Is Ambari Python Client Can Be Used to Make Good Use of Ambari Api’s?**

***Answer:*** Yes, Ambari Python client can be used to make good use of Ambari API’s.

**What Is The Process Of Creating Ambari Client?**

***Answer:*** The following code will do help you to create an Ambari client:

from ambari\_client.ambari\_api import AmbariClient

headers\_dict={'X-Requested-By':'mycompany'} #Ambari needs X-Requested-By header

client = AmbariClient("localhost", 8080, "admin", "admin", version=1,http\_header=headers\_dict)

print client.version

print client.host\_url

print"n"

**How Can We See All The Clusters That Are Available In Ambari?**

***Answer:***all\_clusters = client.get\_all\_clusters()

print all\_clusters.to\_json\_dict()

print all\_clusters

**How Can We See All The Hosts That Are Available In Ambari?**

***Answer:***all\_hosts = client.get\_all\_hosts()

print all\_hosts

print all\_hosts.to\_json\_dict()

print"n"

**What Does Ambari Shell Can Provide?**

***Answer:*** The Ambari shell can provide an interactive and handy command line tool which actually supports the following:

\* All the available functionality in Ambari Web-app

\* All the context-aware command availability

\* Tab completion

\* Any required parameter support if needed

**What Are The Core Benefits For Hadoop Users By Using Apache Ambari?**

***Answer:*** The Apache Ambari is a great gift for individuals who use Hadoop in their day to day work life.

With the use of Ambari, Hadoop users will get the core benefits:

\* Installation process is simplified

\* Configuration and overall management is simplified

\* It has a centralized security setup process

\* It gives out full visibility in terms of Cluster health

\* It is extensively extendable and has an option to customize if needed.

**What Are The Different Life Cycle Commands In Ambari?**

***Answer:*** The Ambari has a defined life cycle commands and they are as follows:

\* Start

\* Stop

\* Status

\* Install

\* Configure

**What Are The Tools That Are Used In Ambari Monitoring?**

***Answer:*** Ambari Monitoring tools actually use two different open source projects for its monitoring purposes,

they are as follows:

o Ganglia

o Nagios

**What Is Ganglia Is Used For In Ambari?**

***Answer:*** It is one of the tools that is used in Ambari, it is mainly used for the following purpose:

\* Monitoring

\* Identifying trending patterns

\* Metrics collection in the clusters

\* It also supports detailed heatmaps

**What Is Nagios Is Used In Ambari?**

***Answer:*** It is one of the tools that is used in Ambari, it is mainly used for the following purpose:

\* First and foremost it is used for health checking and alerts purpose

\* The alert emails can be one of notifications type, service type, host address etc

**What Are The Other Components Of Ambari That Are Important For Automation And Integration?**

***Answer:*** The other components of Ambari that are imported for Automation and Integration are actually divided into three pieces of information:

\* Ambari Stacks

\* Ambari Blueprints

\* Ambari API

Actually, Ambari is built from scratch to make sure that it deals with Automation and Integration problems carefully.

**In Which Language Is The Ambari Shell Is Developed?**

***Answer:*** The shell is developed in Java and it actually based on Ambari REST client and the spring shell framework.

**Before Deploying The Hadoop Instance, What Are The Checks That An Individual Should Do?**

***Answer:*** The following is the list of items that need to be checked before actually deploying the Hadoop instance:

\* Check for existing installations

\* Set up password less SSH

\* Enable NTP on the clusters

\* Check for DNS

\* Disable the SELinux

\* Disable iptables

**List Out The Commands That Are Used To Start, Check The Progress And Stop The Ambari Server?**

***Answer:*** The following are the commands that are used to do the following activities:

To start the Ambari server

**ambari-server start**

To check the Ambari server processes

**ps -ef | grep Ambari**

To stop the Ambari server

**ambari-server stop**

**FLUME**

**What Is Flume?**

***Answer:*** Flume is a distributed, reliable, and available service for efficiently collecting, aggregating, and moving large amounts of log data. It has a simple and flexible architecture based on streaming data flows. It is robust and fault tolerant with tunable reliability mechanisms and many fail over and recovery mechanisms. It uses a simple extensible data model that allows for online analytic application.

**What Is Apache Flume?**

***Answer:*** Apache Flume is a distributed, reliable, and available system for efficiently collecting, aggregating and moving large amounts of log data from many different sources to a centralized data source. Review this Flume use case to learn how Mozilla collects and analyze the Logs using Flume and Hive.

Flume is a framework for populating Hadoop with data. Agents are populated throughout ones IT infrastructure – inside web servers, application servers and mobile devices, for example – to collect data and integrate it into Hadoop.

**Which Is The Reliable Channel In Flume To Ensure That There Is No Data Loss?**

***Answer:*** FILE Channel is the most reliable channel among the 3 channels JDBC, FILE and MEMORY.

**How Can Flume Be Used With Hbase?**

***Answer:*** Apache Flume can be used with HBase using one of the two HBase links:

\* HBaseSink (org.apache.flume.sink.hbase.HBaseSink) supports secure HBase clusters and also the novel HBase IPC that was introduced in the version HBase 0.96.

\* AsyncHBaseSink (org.apache.flume.sink.hbase.AsyncHBaseSink) has better performance than HBase sink as it can easily make non-blocking calls to HBase.

**Working of the HBaseSink:**

In HBaseSink, a Flume Event converted into HBase Increments or Puts. Serializer implements the HBaseEventSerializer which is then instantiated when the sink starts. For every event, sink calls the initialize method in the serializer which then translates the Flume Event into HBase increments and puts to be sent to HBase cluster.

**Working of the AsyncHBaseSink:**

AsyncHBaseSink implements the AsyncHBaseEventSerializer. The initialize method is called only once by the sink when it starts. Sink invokes the setEvent method and then makes calls to the getIncrements and getActions methods just similar to HBase sink. When the sink stops, the cleanUp method is called by the serializer.

**What Is An Flume Agent?**

***Answer:*** A Flume agent is JVM process that hosts flume components such as sources, channels and sinks, and thus has the ability to receive, store and forward events to their destination known as “Agent”. A Flume agent runs on port number 41414 or not sure about (4545).

**Is It Possible To Leverage Real Time Analysis On The Big Data Collected By Flume Directly? If Yes, Then Explain How?**

***Answer:*** Data from Flume can be extracted, transformed and loaded in real-time into Apache Solr server’s usingMorphlineSolrSink.

**What Is A Channel? How Many event entries does Flume stores in channel?**

***Answer:*** It stores events in a queue and the events are delivered to the channel via sources within the agent. An event stays in the channel until a sink removes it for further transport.

A Flume will store a maximum of 100 entries in the two channels.

**Explain About The Different Channel Types In Flume. Which Channel Type Is Faster?**

***Answer:*** The 3 different built in channel types available in Flume are:

\* MEMORY Channel – Events are read from the source into memory and passed to the sink.

\* JDBC Channel – JDBC Channel stores the events in an embedded Derby database.

\* FILE Channel –File Channel writes the contents to a file on the local file system after reading the event from a source. The file is deleted only after the contents are successfully delivered to the sink.

\* MEMORY Channel is the fastest channel among the three however has the risk of data loss. The channel that you choose completely depends on the nature of the big data application and the value of each event. When an agent process dies then events cannot recovered from memory channel because everything was on RAM.

**Explain About The Replication And Multiplexing Selectors In Flume?**

***Answer:*** Channel Selectors are used to handle multiple channels. Based on the Flume header value, an event can be written just to a single channel or to multiple channels. If a channel selector is not specified to the source then by default it is the Replicating selector. Using the replicating selector, the same event is written to all the channels in the source’s channels list. Multiplexing channel selector is used when the application has to send different events to different channels.

**Does Apache Flume Provide Support For Third Party Plug-ins?**

***Answer:*** Most of the data analysts use Apache Flume has plug-in based architecture as it can load data from external sources and transfer it to external destinations.

**OR** Yes, Flume has 100% plugin-based architecture, it can load and ships data from external sources to external destination which separately from Flume. SO that most of the big data analysis use this tool for streaming data.

**Differentiate Between Filesink And Filerollsink?**

***Answer:*** The major difference between HDFS FileSink and FileRollSink is that HDFS File Sink writes the events into the Hadoop Distributed File System (HDFS) whereas File Roll Sink stores the events into the local file system (LFS).

**Why We Are Using Flume?**

***Answer:*** Most often Hadoop developer use this too to get data from social media sites. Cloudera develops it for aggregating and moving very large amount of data. The primary use is to gather log files from different sources and asynchronously persist in the Hadoop cluster.

**What Is FlumeNG (Next Generation)?**

***Answer:*** A real time loader for streaming your data into Hadoop. It stores data in HDFS and HBase. You’ll want to get started with FlumeNG, which improves on the original flume.

**Explain What Are The Tools Used In Big Data?**

***Answer:*** Tools used in Big Data includes

\*Hadoop

\*Hive

\*Pig

\*Flume

\*Mahout

\*Sqoop

**What Are The Complicated Steps In Flume Configurations?**

***Answer:*** Flume can processing streaming data. So if started once, there is no stop/end to the process. Asynchronously it can flows data from source to HDFS via agent. First of all agent should know individual components how they are connected to load data. Therefore, configuration is trigger to load streaming data. For example consumerkey, consumersecret accessToken and accessTokenSecret are key factor to download data from twitter.

**What Are The Data Extraction Tools In Hadoop?**

***Answer:*** Sqoop can be used to transfer data between RDBMS and HDFS. Flume can be used to extract the streaming data from social media, web log etc. and store it on HDFS.

**What Are Flume Core Components?**

***Answer:*** Source, Channels and sink are core components in Apache Flume. When Flume source receives event from external source, it stores the event in one or multiple channels. Flume channel is temporarily store and keep the event until consumed by the Flume sink. It act as Flume repository.

Flume Sink removes the event from channel and put into an external repository like HDFS, Hbase, Solr, AWS S3 or Move to the next flume.

A source can send data to multiple channels, but a single or multiple sink can fetch data from same single channels.

**Does Flume Provide 100% Reliability To The Data Flow?**

***Answer:*** Yes, Apache Flume provides end to end reliability because of its transactional approach in data flow.

**Tell Any Two Features Of Flume?**

***Answer:*** Fume collects data efficiently, aggregate and moves large amount of log data from many different sources to centralized data store.

Flume is not restricted to log data aggregation and it can transport massive quantity of event data including but not limited to network traffic data, social-media generated data , email message and pretty much any data storage.

**What Are Interceptors?**

***Answer:*** Interceptors are used to filter the events between source and channel, channel and sink. These channels can filter un-necessary or targeted log files. Depends on requirements you can use n number of interceptors.

Interceptor is a point in the data flow where you can inspect the events and alter the events.

**Why Flume?**

***Answer:*** Flume is not limited to collect logs from distributed systems, but it is capable of performing other use cases such as

\* Collecting readings from array of sensors

\* Collecting impressions from custom apps for an ad network

\* Collecting readings from network devices in order to monitor their performance.

\* Flume is targeted to preserve the reliability, scalability, manageability and extensibility while it serves maximum number of clients with higher QoS

**What Is Flume Event? What does Event consist of?**

***Answer:*** A unit of data with set of string attribute called Flume event. The external source like web-server send events to the source. Internally Flume has inbuilt functionality to understand the source format.

Each log file is consider as an event. Each event has header and value sectors, which has header information and appropriate value that assign to articular header.

An Event consist of **body and headers**. The event body is a **byte array** and the header are map (grouped) with **string and values**. Headers are for routing purpose {Ex: - hostname=abc.com} which is used for tracking the event which is being sent

A number of events together build transactions, which has a unique ID.

**What Is Sink Processors?**

***Answer:*** Sink processors is mechanism by which you can create a fail-over path for sink and load balancer for multiple flume events across multiple sinks.

**How Multi-hop Agent Can Be Setup In Flume?**

***Answer:*** Avro RPC Bridge mechanism is used to setup Multi-hop agent in Apache Flume.

**Can Flume Can Distribute Data To Multiple Destinations?**

***Answer:*** Yes. It support multiplexing flow. The event flows from one source to multiple channel and multiple destinations, it is achieved by defining a flow multiplexer.

**Can You Explain About Configuration Files?**

***Answer:*** The agent configuration is stored in local configuration file. It comprises of each agents source, sink and channel information.

**What Are The Similarities And Differences Between Apache Flume And Apache Kafka?**

***Answer:*** Flume pushes messages to their destination via its Sinks. With Kafka you need to consume messages from Kafka Broker using a Kafka Consumer API.

**Explain Reliability And Failure Handling In Apache Flume?**

***Answer:*** Flume NG uses channel-based transactions to guarantee reliable message delivery. When a message moves from one agent to another, two transactions are started, one on the agent that delivers the event and the other on the agent that receives the event. In order for the sending agent to commit its transaction, it must receive success indication from the receiving agent.

The receiving agent only returns a success indication if its own transaction commits properly first. This ensures guaranteed delivery semantics between the hops that the flow makes.

**What is collector?**

***Answer:*** The agent receive the data from the data generators. A node known as collector will collect the data from this agent and write it to HDFS.

The flume sink for each agent node send data to collector nodes. The collector node grouped data from multiple agent and write it to HDFS further it can analyzed with pig.

**How the events can removed from the channel? What happens if any failure occurs during transaction?**

***Answer:*** Source writes an events to a channels and event stay in the channels unit, a sink remove the events from the channel through a transaction on HDFS.

If there is a network failure, the events are queued in the channels until the sink can write them to HDFS.

**What is Multi-Hop Flow?**

***Answer:*** An event may travel through multiple agent this known as Multi-Hop Flow.

**What is Fan-Out Flow?**

***Answer:*** The data flow from one source to multiple channel known as Fan-Out Flow.

**What is Fan-in Flow?**

***Answer:*** The data flow in which the data transferred from many source to one channel known as Fan-in Flow.

**Types of Source, Channel, Sink.**

***Answer:*** Source: 1. Avro Source 2. Netcat 3. Exec 4. Spooldir 5. HTTP

Channel: 1. File Channel 2. Memory Channel 3. JDBC Channel

Sink: 1. Logger 2. Avro 3. HDFS

**How event and messages copied to Flume agent Body?**

***Answer:*** Text message converted to byte array and copied to Flume agent body, the UTF-8 the default converter is used.

**What should be a message size in Flume?**

***Answer:*** Flume cannot convert more than 2GB of data per message. For Best performance, Cloudera recommends to configure your application to send message smaller than 2MB in size through flume.

**What is a default size of Event?**

***Answer:*** The default size of Event is 2kb in Size. Thus, large events such as video and audio events will not works well with flume.

**SQOOP**

**How to import RDBMS table in Hadoop using Sqoop when the table doesn’t have a primary key column?**

***Answer:*** Usually, we import an RDBMS table in Hadoop using Sqoop Import when it has a primary key column. If it doesn’t have the primary key column, **it will give you the below error-**

If your table does not have the primary key column, you need to specify -m 1 option for importing the data, or you have to provide –split-by argument with some column name.

Here are the scripts which you can use to import an RDBMS table in Hadoop using Sqoop when you don’t have a primary key column.

sqoop import -- connect jdbc:mysql://localhost/dbname –username root –password root –table user –target-dir /user/root/user\_data –columns “first\_name, last\_name, created\_date” -m 1

**OR**

sqoop import --connect jdbc:mysql://localhost/ dbname –username root –password root –table user –target-dir /user/root/user\_data –columns “first\_name, last\_name, created\_date” –split-by created\_date

**By default, Sqoop assumes that it's working with space-separated fields and that each record is terminated by a newline. True or false?**

***Answer:*** False

**Sqoop uses MapReduce jobs to import and export data, and you can configure the number of Mappers used. True or false?**

***Answer:*** True

**What is the default file format to import data using Apache Sqoop?**

Sqoop allows data to be imported using two file formats

i) Delimited Text File Format

This is the default file format to import data using Sqoop. This file format can be explicitly specified using the –as-textfile argument to the import command in Sqoop. Passing this as an argument to the command will produce the string based representation of all the records to the output files with the delimited characters between rows and columns. Ex:- Different columns of First Name, Last Name, City, Phone

ii) Sequence File Format

It is a binary file format where records are stored in custom record-specific data types which are shown as Java classes. Sqoop automatically creates these data types and manifests them as java classes.

**I have around 300 tables in a database. I want to import all the tables from the database except the tables named Table298, Table 123, and Table299. How can I do this without having to import the tables one by one?**

This can be accomplished using the import-all-tables import command in Sqoop and by specifying the exclude-tables option with it as follows-

sqoop import-all-tables --connect –username –password --exclude-tables Table298, Table 123, Table 299

**Does Apache Sqoop have a default database?**

Yes, MySQL is the default database.

**How can I import large objects (BLOB and CLOB objects) in Apache Sqoop?**

Sqoop in case of BLOB and CLOB objects, sqoop does not support the direct import function. Hence, if you have to import large purposes, you can use JDBC based imports. This can done without introducing the direct argument of the import utility.

**How can you execute a free form SQL query in Sqoop to import the rows in a sequential manner?**

This can be accomplished using the –m 1 option in the Sqoop import command. It will create only one MapReduce task which will then import rows serially.

**How will you list all the columns of a table using Apache Sqoop?**

Unlike sqoop-list-tables and sqoop-list-databases, there is no direct command like sqoop-list-columns to list all the columns. The indirect way of achieving this is to retrieve the columns of the desired tables and redirect them to a file, which can viewed manually containing the column names of a particular table.

**What is the difference between Sqoop and DistCP command in Hadoop?**

Both distCP (Distributed Copy in Hadoop) and Sqoop transfer data in parallel but the only difference is that distCP command can transfer any kind of data from one Hadoop cluster to another whereas Sqoop transfers data between RDBMS and other components in the Hadoop ecosystem like HBase, Hive, HDFS, etc.

**What is Sqoop metastore?**

Sqoop metastore is a shared metadata repository for remote users to define and execute saved jobs created using sqoop job defined in the metastore. The sqoop –site.xml should be configured to connect to the metastore.

In addition, with the –meta-connect argument Clients must be configured to connect to the metastore in sqoop-site.xml.

**During sqoop import, you use the clause –m or –numb-mappers to specify the number of mappers as 8 so that it can run eight parallel MapReduce tasks, however, sqoop runs only four parallel MapReduce tasks. Why?**

Hadoop MapReduce cluster is configured to run a maximum of 4 parallel MapReduce tasks and the sqoop import can be configured with number of parallel tasks less than or equal to 4 but not more than 4.

**What is the significance of using –split-by clause for running parallel import tasks in Apache Sqoop?**

--Split-by clause is used to specify the columns of the table that are used to generate splits for data imports. This clause specifies the columns that will be used for splitting when importing the data into the Hadoop cluster. —split-by clause helps achieve improved performance through greater parallelism. Apache Sqoop will create splits based on the values present in the columns specified in the –split-by clause of the import command. If the –split-by clause is not specified, then the primary key of the table is used to create the splits while data import. At times the primary key of the table might not have evenly distributed values between the minimum and maximum range. Under such circumstances –split-by clause can be used to specify some other column that has even distribution of data to create splits so that data import is efficient.

**You use –split-by clause but it still does not give optimal performance then how will you improve the performance further.**

Using the –boundary-query clause. Generally, sqoop uses the SQL query select min (), max () from to find out the boundary values for creating splits. However, if this query is not optimal then using the –boundary-query argument any random query can be written to generate two numeric columns.

**You successfully imported a table using Apache Sqoop to HBase but when you query the table, it is found that the number of rows is less than expected. What could be the likely reason?**

If the imported records have rows that contain null values for all the columns, then probably those records might have been dropped off during import because HBase does not allow null values in all the columns of a record.

**The incoming value from HDFS for a particular column is NULL. How will you load that row into RDBMS in which the columns are defined as NOT NULL?**

Using the –input-null-string parameter, a default value can be specified so that the row is inserted with the default value for the column that it has a NULL value in HDFS.

**If the source data gets updated every now and then, how will you synchronise the data in HDFS that is imported by Sqoop?**

Data can be synchronised using incremental parameter with data import –

--Incremental parameter can be used with one of the two options-

i) **append**-If the table is getting updated continuously with new rows and increasing row id values then incremental import with append option should be used where values of some of the columns are checked (columns to be checked are specified using –check-column) and if it discovers any modified value for those columns then only a new row will be inserted.

ii) **lastmodified** – In this kind of incremental import, the source has a date column which is checked for. Any records that have been updated after the last import based on the lastmodifed column in the source, the values would be updated.

**Below command is used to specify the connect string that contains hostname to connect MySQL with local host and database name as test\_db –**

**–connect jdbc: mysql: //localhost/test\_db**

**Is the above command the best way to specify the connect string in case I want to use Apache Sqoop with a distributed hadoop cluster?**

When using Sqoop with a distributed Hadoop cluster the URL should not be specified with localhost in the connect string because the connect string will be applied on all the DataNodes with the Hadoop cluster. So, if the literal name localhost is mentioned instead of the IP address or the complete hostname then each node will connect to a different database on their localhosts. It is always suggested to specify the hostname that can be seen by all remote nodes.

**What is Sqoop Import? Explain its purpose.**

Ans. While it comes to import tables from RDBMS to HDFS we use Sqoop Import tool. Generally, we can consider that each row in a table is a record in HDFS. Also, when we talk about text files all records are there as text data. However, when we talk about Avro and sequence files all records are there as binary data here. To be more specific, it imports individual tables from RDBMS to HDFS.

**Name a few import control commands. How can Sqoop handle large objects?**

Ans. To import RDBMS data, we use import control commands

Append: Append data to an existing dataset in HDFS.

–append

Columns: columns to import from the table.

–columns

<col,col……> •

Where: where clause to use during import. —

Where the common large objects are Blob and Clob. Suppose the object is less than 16 MB, it is stored inline with the rest of the data. If there are big objects, they are temporarily stored in a subdirectory with the name \_lob. Those data are then materialized in memory for processing. If we set lob limit as ZERO (0) then it is stored in external memory.

**How can we import data from particular row or column? What is the destination types allowed in Sqoop import command?**

Ans. Basically, on the basis of where clause, Sqoop allows to Export and Import the data from the data table. So, the syntax is

–columns

<col1,col2……> –where

–query

For Example:

sqoop import –connect jdbc:mysql://db.one.com/corp –table INTELLIPAAT\_EMP –where “start\_date> ’2016-07-20’ ”

sqoopeval –connect jdbc:mysql://db.test.com/corp –query “SELECT \* FROM intellipaat\_emp LIMIT 20”

sqoop import –connect jdbc:mysql://localhost/database –username root –password aaaaa –columns “name,emp\_id,jobtitle”

However, into following services Sqoop supports data imported:

HDFS, Hive, Hbase, Hcatalog, Accumulator

**When to use –target-dir and when to use –warehouse-dir while importing data?**

Ans. Basically, we use –target-dir to specify a particular directory in[HDFS](https://data-flair.training/blogs/hadoop-hdfs-tutorial/). Whereas we use –warehouse-dir to specify the parent directory of all the sqoop jobs. Therefore, in this case under the parent directory sqoop will create a directory with the same name as the table.

**What is the process to perform an incremental data load in Sqoop?**

Ans. In Sqoop, the process to perform incremental data load is to synchronize the modified or updated data (often referred as delta data) from RDBMS to [Hadoop](https://data-flair.training/blogs/hadoop-2-x-vs-hadoop-3-x-comparison/). Moreover, in Sqoop the delta data can be facilitated through the incremental load command.

In addition, by using Sqoop import command we can perform incremental load. Also, by loading the data into the hive without overwriting it. However, in Sqoop the different attributes that need to be specified during incremental load are

1) Mode (incremental)

It shows how Sqoop will determine what the new rows are. Also, it has value as Append or Last Modified.

2) Col (Check-column)

Basically, it specifies the column that should be examined to find out the rows to be imported.

3) Value (last-value)

It denotes the maximum value of the check column from the previous import operation.

**What is the significance of using –compress-codec parameter?**

Ans. However, we use the –compress -code parameter to get the out file of a sqoop import in formats other than .gz like .bz2.

**Can free-form SQL queries be used with Sqoop import command? If yes, then how can they be used?**

Ans. In Sqoop, we can use SQL queries with the import command. Basically, we should use import command with the –e and – query options to execute free-form SQL queries. But note that the –target dir value must be specified While using the –e and –query options with the import command.

**What is the importance of eval tool?**

Ans. Basically,[Sqoop Eval](https://data-flair.training/blogs/sqoop-eval/) helps to run sample SQL queries against Database as well as preview the results on the console. Moreover, it helps to know what data we can import or that desired data is imported or not.

**How can you import only a subset of rows from a table?**

Ans. In the sqoop import statement, by using the WHERE clause we can import only a subset of rows.

**What are the limitations of importing RDBMS tables into Hcatalog directly?**

Ans. By making use of –hcatalog –database option with the –hcatalog –table, we can import RDBMS tables into Hcatalog directly. However, there is one limitation to it is that it does not support several arguments like –as-Avro file, -direct, -as-sequencefile, -target-dir , -export-dir.

**What is the advantage of using –password-file rather than -P option while preventing the display of password in the sqoop import statement?**

Ans.  Inside a sqoop script, we can use the –password-file option. Whereas the -P option reads from standard input, preventing automation.

**What do you mean by Free Form Import in Sqoop?**

Ans. By using, any SQL Sqoop can import data from a relational database query rather than only using table and column name parameters.

**What is the role of JDBC driver in Sqoop?**

Ans. Basically, sqoop needs a connector to connect to different relational databases. Since, as a JDBC driver, every DB vendor makes this connector available which is specific to that DB. Hence, to interact with Sqoop needs the JDBC driver of each of the database it needs.

**Is JDBC driver enough to connect sqoop to the databases?**

Ans. No. to connect to a database Sqoop needs both JDBC and connector.

**What is InputSplit in Hadoop?**

Ans. Input Split is defined as while a Hadoop job runs, it splits input files into chunks also assign each split to a mapper to process.

**What is the work of Export in Hadoop sqoop?**

Ans. Export tool transfer the data from HDFS to RDBMS

**Use of Codegen command in Hadoop sqoop?**

Ans. Basically, Codegen command generates code to interact with database records

**Use of Help command in Hadoop sqoop?**

Ans. Help command in Hadoop sqoop generally list available commands

**How can you schedule a sqoop job using Oozie?**

Ans. However, Oozie has in-built sqoop actions inside which we can mention the sqoop commands to be executed.

**What is the importance of — the split-by clause in running parallel import tasks in sqoop?**

Ans. In Sqoop, it mentions the column name based on those value the data will be divided into groups of records. Further, by the [MapReduce](https://data-flair.training/blogs/how-hadoop-mapreduce-works/)tasks, these group of records will be read in parallel.

**What is the purpose of sqoop-merge?**

Ans. The merge tool combines two datasets where entries in one dataset should overwrite entries of an older dataset preserving only the newest version of the records between both the data sets.

**How can you see the list of stored jobs in sqoop metastore?**

Ans. sqoop job –list

**Which database the sqoop metastore runs on?**

Ans. Basically, on the current machine running sqoop-metastore launches, a shared HSQLDB (Hyper SQL Database) instance.

**Where can the metastore database be hosted?**

Ans. Anywhere, it means we can host metastore database within or outside of the [Hadoop cluster](https://data-flair.training/blogs/install-hadoop-1-x-on-multi-node-cluster/).

**Give the sqoop command to see the content of the job named myjob?**

Ans. Sqoop job –show myjob

**How can you control the mapping between SQL data types and Java types?**

Ans. we can configure the mapping between by using the –map-column-java property.

For example:

$ sqoop import … –map-column-java id = String, value = Integer

**Is it possible to add a parameter while running a saved job?**

Ans. Yes, by using the –exec option we can add an argument to a saved job at runtime.

sqoop job -exec jobname - -newparameter

**What is the usefulness of the options file in sqoop.**

Ans. To specify the command line values in a file and use it in the sqoop commands we use the options file in sqoop.

For example

The –connect parameter’s value and –user name value scan be stored in a file and used again and again with different sqoop commands.

**How can you avoid importing tables one-by-one when importing a large number of tables from a database?**

Ans. Using the command

sqoop import-all-tables –connect -usrename -password -exclude-tables table1,table2 ..

Basically, this will import all the tables except the ones mentioned in the exclude-tables clause.

**How can you control the number of mappers used by the sqoop command?**

Ans. To control the number of mappers executed by a sqoop command we use the parameter –num-mappers. Moreover, we should start with choosing a small number of map tasks and then gradually scale up as choosing high number of mappers initially may slow down the performance on the database side.

**What is the default extension of the files produced from a sqoop import using the –compress parameter?**

Ans. .gz

**What is the significance of using –compress-codec parameter?**

Ans. We use the –compress -code parameter to get the out file of a sqoop import in formats other than .gz like .bz2.

**What is a disadvantage of using –direct parameter for faster data load by sqoop?**

Ans. The native utilities used by databases to support faster laod do not work for binary data formats like SequenceFile.

**How will you update the rows that are already exported?**

Ans. Basically, to update existing rows we can use the parameter –update-key. Moreover, in it, a comma-separated list of columns is used which uniquely identifies a row. All of these columns are used in the WHERE clause of the generated UPDATE query. All other table columns will be used in the SET part of the query.

**What are the basic commands in Apache Sqoop and its uses?**

Ans. The basic commands of Apache Sqoop are:

[Codegen](https://data-flair.training/blogs/sqoop-codegen/), Create-hive-table, [Eval](https://data-flair.training/blogs/sqoop-eval/), [Export](https://data-flair.training/blogs/sqoop-export/), Help,[Import](https://data-flair.training/blogs/sqoop-import/), [Import-all-tables](https://data-flair.training/blogs/sqoop-import-all-tables/),[List-databases](https://data-flair.training/blogs/sqoop-list-databases/), [List-tables](https://data-flair.training/blogs/sqoop-list-tables/), Versions.

Moreover, uses of Apache Sqoop basic commands are:

Codegen- It helps to generate code to interact with database records.

Create- hive-table- It helps to Import a table definition into a hive

Eval- It helps to evaluate SQL statement and display the results

Export- It helps to export an HDFS directory into a database table

Help- It helps to list the available commands

Import- It helps to import a table from a database to HDFS

Import-all-tables- It helps to import tables from a database to HDFS

List-databases- It helps to list available databases on a server

List-tables- It helps to list tables in a database

Version- It helps to display the version information

**How Sqoop word came? Sqoop is which type of tool and the main use of sqoop?**

Ans. Sqoop word came from SQL+HADOOP=SQOOP.

Basically, it is a data transfer tool. We use Sqoop to import and export a large amount of data from RDBMS to HDFS and vice versa.

**What is Sqoop Validation?**

Ans. It means to validate the data copied. Either import or export by comparing the row counts from the source as well as the target post copy. Likewise, we use this option to compare the row counts between source as well as the target just after data imported into HDFS. Moreover, while during the imports, all the rows are deleted or added, Sqoop tracks this change. Also updates the log file.

**What is Purpose to Validate in Sqoop?**

Ans. In Sqoop to validate, the data copied is Validation main purpose. Either Sqoop import or Export by comparing the row counts from the source as well as the target post copy.

**What is Sqoop Job?**

Ans. To perform an incremental import if a saved job is configured, then state regarding the most recently imported rows is updated in the saved job. That allows the job to continually import only the newest rows.

**What is Sqoop Import Mainframe Tool and its Purpose?**

Ans. A tool that we use to import all sequential datasets in a partitioned dataset (PDS) on a mainframe to HDFS is Sqoop Import Mainframe. In addition, A PDS is similar to a directory on the open systems. Likewise, in a dataset, the records can only contain character data. Moreover, here records will be stored as a single text field with the entire record.

**What is the purpose of Sqoop List Tables?**

Ans. The main purpose of sqoop-list-tables will list down tables present in a database.

**Difference Between Apache Sqoop vs Flume.**

Ans. So, let us discuss all the differences based on features.  
**A. Data Flow**  
Apache Sqoop – Basically, Sqoop works with any type of relational database system (RDBMS) that has the basic JDBC connectivity. In addition, Sqoop can import data from NoSQL databases like MongoDB, Cassandra and along with it. Moreover, it allows data transfer to Apache Hive or HDFS. [Apache Flume](https://data-flair.training/blogs/apache-flume-tutorial/) – Likewise, Flume works with streaming data sources those generated continuously in Hadoop environments. Like log files.

**B. Type of Loading**  
Apache Sqoop – Basically, Sqoop load is not driven by events. Apache Flume – Here, data loading is completely event-driven.

**C. When to use**  
Apache Sqoop – However, if the data is being available in Teradata, Oracle, MySQL, PostgreSQL or any other JDBC compatible database it considered an ideal fit. Apache Flume – While we move bulk of streaming data from sources likes JMS or spooling directories, it is the best choice.

**D. Link to HDFS**  
Apache Sqoop – Basically, for importing data in Apache Sqoop, HDFS is the destination. Apache Flume – In Apache Flume, data generally flow to HDFS through channels.

**E. Architecture**   
Apache Sqoop – Basically, it has connector-based architecture. However, that means the connectors know a great deal in connecting with the various data sources. Also to fetch data correspondingly. Apache Flume – However, it has agent-based architecture. It means code written in Flume is we call agent that may responsible for fetching the data.

**Zookeeper**

**What is the model of zookeeper cluster ?**

***Answer:***- Leader and Follower.

**What are the key elements in Zookeeper Architecture?**

***Answer:*** - The key elements in the Zookeeper architecture are:

Node: The systems installed on the cluster

ZNode: The nodes where the status is updated by other nodes in cluster

Client Applications: The tools that interact with the distributed applications

Server Applications: Allows the client applications to interact using a common interface

**Zookeeper essentially mirrors the \_\_\_\_\_\_\_ functionality exposed in the Linux kernel.**

***Answer:***- inotify

**Explanation:** A client can request for Zookeeper to generate the node name to avoid collisions.

**What is the znodes?**

***Answer:***- znodes are Zookeeper data nodes. Zookeeper has a file system-like data model composed of znodes.

**OR**

Znodes maintain a state structure that includes version numbers for data changes, ACL changes, and timestamps, to allow cache validations and coordinated updates.

Each time a znode’s data changes, the version number increases. For instance, whenever a client retrieves data it also receives the version of the data.

The data stored at each znode in a namespace is read and written atomically. Reads get all the data bytes associated with a znode and a write replaces all the data.

Each node has an Access Control List (ACL) that restricts who can do what.

**Explain The Types Of Znodes?**

***Answer:***- There are 3 Types of znodes: ephemeral, persistent and sequential.

**A) Ephemeral znodes:-**

Ephemeral znodes are active until the client is alive. When a client gets disconnected from the ZooKeeper ensemble, then the ephemeral znodes get deleted automatically.

For this reason, ephemeral znodes are not allowed to have a children further. Ephemeral znodes play an important role in Leader election.

**B) Persistence znodes:-**

Persistence znode is alive even after the client, who created that particular znode, is disconnected.

By default, all znodes are persistent unless otherwise specified.

**C) Sequential znodes:-**

Sequential znodes can be either persistent or ephemeral.

When a new znode is created as a sequential znode, then Zookeeper sets the path of the znode by attaching a 10-digit sequence number to the original name.

For example, let’s say client created a cznode. In the ZooKeeper server, the cznode will be named like this: cznode0000000001 If client creates another sequential znode,

it would bear the next number in a sequence. So the next sequential znode will be called <znode-name>0000000002. Sequential znodes play an important role in Locking and Synchronization.

**A \_\_\_\_\_\_\_\_\_\_\_ server is a machine that keeps a copy of the state of the entire system and persists this information in local log files.**

***Answer:***- Zookeeper

**The Zookeeper Data Directory contains files which are \_\_\_\_\_\_\_\_\_ copy of the znodes stored by a particular serving ensemble.**

***Answer:***- persistent

**Explanation:** As changes are made to the znodes these changes are appended to a transaction log, occasionally, when a log grows large, a snapshot of the current state of all znodes will be written to the file system.

**The location where Zookeeper will store the in-memory database snapshots?**

***Answer:***- Data Directory

**What Are The Benefits Of Distributed Applications?**

***Answer:***- A) Reliability: Failure of a single or a few systems does not make the whole system to fail.

B) Scalability: Performance can be increased as and when needed by adding more machines with minor change in the configuration of the application with no downtime.

C) Transparency: Hides the complexity of the system and shows itself as a single entity / application.

**Explain The CLI In Zookeeper?**

***Answer:***- Zookeeper Command Line Interface (CLI) is used to interact with the ZooKeeper ensemble for development purpose which is also useful for debugging.

To perform ZooKeeper CLI operations, first turn on your ZooKeeper server (“bin/zkServer.sh start”) and then, ZooKeeper client (“bin/zkCli.sh”).

**How Can We Create Znodes?**

***Answer:***- At the time of Znode creation The flag argument specifies whether the created znode will be ephemeral, persistent, or sequential. By default, all znodes are persistent.

A) If no flags are specified, then the znode is considered as persistent.

create /path /data

B) To create a Sequential znode, add -s flag as shown below.

create -s /path /data

C) To create an Ephemeral Znode, add -e flag as shown below.

create -e /path /data

**How Can We Create Children / Sub-znode?**

***Answer:***- Creating children is similar to creating new znodes. The only difference is that the path of the child znode will have the parent path as well.

create /parent/path/subnode/path /data

If you want to create a child or znode Monday, while creating it you have mention its full path,

create /year/month/week/days/Monday

**How Can We Remove A Znode?**

***Answer:*** Removes a specified znode and recursively all its children. This would happen only if such a znode is available.

rmr /year/month

**What Are The Basics Of Zookeeper Api?**

***Answer:***

# Client Connect to the ZooKeeper ensemble. ZooKeeper ensemble assign a Session ID for the client.

# Client Send heartbeats to the server periodically. Otherwise, the ZooKeeper ensemble expires the Session ID and the client needs to reconnect.

# Client Get / Set the znodes as long as a session ID is active.

# Client Disconnect from the ZooKeeper ensemble, once all the tasks are completed. If the client is inactive for a prolonged time, then the ZooKeeper ensemble will automatically disconnect the client.

**What is the zookeeper daemon name?**

***Answer:*** quorumpeermain

QuorumPeerMain is the name of the main class that should be refer to start the zookeeper.

**What additional monitoring should I put in place?**

***Answer:***- You should add monitoring on each host that runs a NameNode to ensure that the ZKFC remains running. In some types of ZooKeeper failures,

for example, the ZKFC may unexpectedly exit, and should be restarted to ensure that the system is ready for automatic failover.

Additionally, you should monitor each of the servers in the ZooKeeper quorum. If ZooKeeper crashes, then automatic failover will not function.

**What are watches?**

***Answer:***- Accessing a znode every time a client needs to know its content would be very expensive.

Zookeeper has an event system referred to as watch which can be set on Znode to trigger an event whenever it is removed, altered or any new children are created below it.

Clients register with ZooKeeper to receive notifications of changes to znodes by setting a watch.

**OR**

ZooKeeper supports the concept of watches. Clients can set a watch on a znodes. A watch will be triggered and removed when the znode changes.

When a watch is triggered the client receives a packet saying that the znode has changed.

And if the connection between the client and one of the ZooKeeper servers is broken, the client will receive a local notification.

**What is ZooKeeper Atomic Broadcast (ZAB) protocol?**

***Answer:***- The ZooKeeper Atomic Broadcast (ZAB) protocol is the core of the system. A ZAB, which elects a leader, synchronizes the nodes, and performs broadcasts of updates from the leader.

The ZooKeeper Atomic Broadcast algorithm, which is the protocol that manages atomic updates to the replicas. It is responsible for agreeing on a leader in the ensemble, synchronizing the replicas, managing update transactions to be broadcast, as well as recovering from a crashed state to a valid state.

**Is it important that I start the ZKFC and NameNode daemons in any particular order?**

***Answer:***- No. On any given node you may start the ZKFC before or after its corresponding NameNode.

**What happens if ZooKeeper goes down?**

***Answer:***- If the ZooKeeper cluster crashes, no automatic failovers will be triggered. However, HDFS will continue to run without any impact.

When ZooKeeper is restarted, HDFS will reconnect with no issues.

**Can I designate one of my NameNodes as primary/preferred?**

***Answer:***- No. Currently, this is not supported. Whichever NameNode is started first will become active. You may choose to start the cluster in a specific order such that your preferred node starts first.

**How can I initiate a manual failover when automatic failover is configured?**

***Answer:***- Even if automatic failover is configured, you may initiate a manual failover using the same hdfs haadmin command. It will perform a coordinated failover.

**What is org.apache.jute package?**

***Answer:***- org.apache.jute – Hadoop record I/O contains classes and a record description language translator for simplifying serialization and, deserialization of records in a language-neutral manner.

**What are barriers?**

***Answer:*** - Distributed systems use barriers to block processing of a set of nodes until a condition is met at which time all the nodes are allowed to proceed.

Barriers are implemented in ZooKeeper by designating a barrier node. The barrier is in place if the barrier node exists. Here's the pseudo code:

A) Client calls the ZooKeeper API's exists () function on the barrier node, with watch set to true.

B) If exists () returns false, the barrier is gone and the client proceeds

C) Else, if exists () returns true, the clients wait for a watch event from ZooKeeper for the barrier node.

D) When the watch event is triggered, the client reissues exists ( ) call, again waiting until the barrier node is removed.

Double barriers enable clients to synchronize the beginning and the end of a computation.

When enough processes have joined the barrier, processes start their computation and leave the barrier once they have finished.

The barrier node as b. Every client process p registers with the barrier node on entry and unregisters when it is ready to leave.

A node registers with the barrier node via the Enter procedure below, it waits until x client process register before proceeding with the computation.

**What are Producer-Consumer Queues?**

***Answer:***- A producer-consumer queue is a distributed data structure that are group of processes use to generate and consume items.

Producer processes create new elements and add them to the queue. Consumer processes remove elements from the list, and process them.

**What is CONNECTION\_LOSS error?**

***Answer:***- CONNECTION\_LOSS means the link between the client and server was broken. It doesn’t necessarily mean that the request failed.

If you are doing a create request and the link was broken after the request reached the server and before the response was returned, the create request will succeed.

If the link was broken before the packet went onto the wire, the create request failed. Unfortunately, there is no way for the client library to know, so it returns CONNECTION\_LOSS.

**How should you handle SESSION\_EXPIRED?**

***Answer:***- Session expiration is managed by the ZooKeeper cluster itself, not by the client. When the ZK client establishes a session with the cluster it provides a "timeout" value.

This value is used by the cluster to determine when the client's session expires.

Expirations happens when the cluster does not hear from the client within the specified session timeout period (i.e. no heartbeat).

At session expiration the cluster will delete any/all ephemeral nodes owned by that session and immediately notify any/all connected clients of the change (anyone watching those znodes).

At this point the client of the expired session is still disconnected from the cluster,

It will not be notified of the session expiration until/unless it is able to re-establish a connection to the cluster.

The client will stay in disconnected state until the TCP connection is re-established with the cluster, at which point the watcher of the expired session will receive the "session expired" notification.

Example state transitions for an expired session as seen by the expired session's watcher:

A) ‘Connected’: session is established and client is communicating with cluster (client/server communication is operating properly)

B) .... client is partitioned from the cluster

C) ‘Disconnected’: client has lost connectivity with the cluster

D) .... time elapses, after 'timeout' period the cluster expires the session, nothing is seen by client as it is disconnected from cluster

E) .... time elapses, the client regains network level connectivity with the cluster

F) ‘Expired’: eventually the client reconnects to the cluster; it is then notified of the expiration.

**Is there an easy way to expire a session for testing?**

***Answer:***- Yes, a ZooKeeper handle can take a session id and password. This constructor is used to recover a session after total application failure.

For example, an application can connect to ZooKeeper, save the session id and password to a file, terminate, restart, read the session id and password, and reconnect to ZooKeeper without losing the session and the corresponding ephemeral nodes.

It is up to the programmer to ensure that the session id and password isn’t passed around to multiple instances of an application, otherwise problems can result.

**What are the options-process for upgrading ZooKeeper?**

***Answer:***- There are two primary ways of doing this; 1) full restart or 2) rolling restart.

**In the full restart case** you can stage your updated code/configuration/etc…, stop all of the servers in the ensemble, switch code/configuration, and restart the ZooKeeper ensemble.

If you do this programmatically (scripts typically, i.e. not by hand) the restart can be done on order of seconds.

As a result the clients will lose connectivity to the ZooKeeper cluster during this time, however it looks to the clients just like a network partition.

All existing client sessions are maintained and re-established as soon as the ZooKeeper ensemble comes back up.

Obviously, one drawback to this approach is that if you encounter any issues (it’s always a good idea to test/stage these changes on a test harness), the cluster may be down for longer than expected.

The second option, preferable for many users, is to do a “rolling restart”.

**In the rolling restart case** you upgrade one server in the ZooKeeper ensemble at a time; bring down the server, upgrade the code/configuration/etc…, then restart the server.

The server will automatically rejoin the quorum, update its internal state with the current ZK leader, and begin serving client sessions.

As a result of doing a rolling restart, rather than a full restart, the administrator can monitor the ensemble as the upgrade progresses, perhaps rolling back if any issues are encountered.

**Can I run an ensemble cluster behind a load balancer?**

***Answer:***- There are two types of servers failures in a distributed system from socket I/O perspective.

A) Server down due to hardware failures and OS panic/hang, Zookeeper daemon hang, temporary/permanent network outage, network switch anomaly, etc.

Client cannot figure out failures immediately since there is no responding entities. As a result, zookeeper clients must rely on timeout to identify failures.

B) Dead zookeeper process (daemon): since OS will respond to closed TCP port, client will get “connection refused” upon socket connect or “peer reset” on socket I/O.

Client immediately notice that the other end failed. Here’s how ZK clients respond to servers in each case.

ZK client rely on heartbeat algorithm. In this scenario, ZK client will immediately detect failure, and will retry connecting every second assuming only one ensemble IP is given.

If multiple ensemble IP is given (most installation falls into this category), ZK client retries next IP immediately.

**What happens to ZK sessions while the cluster is down?**

***Answer:***- Imagine that a client is connected to ZK with a 5 second session timeout, and the administrator brings the entire ZK cluster down for an upgrade.

The cluster is down for several minutes, and then is restarted.

In this scenario, the client is able to reconnect and refresh its session.

Because session timeouts are tracked by the leader, the session starts counting down again with a fresh timeout when the cluster is restarted.

So, as long as the client connects within the first 5 seconds after a leader is elected, it will reconnect without an expiration, and any ephemeral nodes it had prior to the downtime will be maintained.

The same behavior is exhibited when the leader crashes and a new one is elected.

In the limit, if the leader is flip-flopping back and forth quickly, sessions will never expire since their timers are getting constantly reset.

**What are the different ZKClientBindings?**

***Answer:***- ZooKeeper ships with C, Java, Perl and Python client bindings, here are a list of client bindings that are available from the community are,

Scala, C#, Node.js, Twisted/Python, Python (no C dependency), Erlang, Haskell, Ruby, Go, Lua.

**Can you list some useful Zookeeper tools?**

***Answer:***-

zkconf – generate configuration for a ZooKeeper ensemble

zk-smoketest – smoketest or latencytest a ZooKeeper ensemble (uses zkpython)

zookeeper\_dashboard – web dashboard for ZooKeeper ensemble (uses zkpython & django)

zktop – monitor ZooKeeper in realtime

zkexamples – phunt’s “random examples of useful bits of ZooKeeper ephemera”

SPM for ZooKeeper – Performance Monitoring and Alerting for ZooKeeper

**What is Cages?**

***Answer:***- Cages is a distributed synchronization library for Zookeeper. Cages is a Java library of distributed synchronization primitives that uses the Apache ZooKeeper system.

If you can run a ZooKeeper machine or cluster, then you can use Cages to synchronize and coordinate data access, data manipulation and data processing, configuration change and more secret things like cluster membership across multiple machines.

**What is BookKeeper?**

***Answer:***- Bookkeeper is a system which dependably on streams of records. It is designed to store write ahead logs, such as those found in database or database like applications.

In fact, the Hadoop NameNode inspired BookKeeper. The NameNode logs changes to the in-memory namespace data structures to the local disk before they are applied in memory.

However logging the changes locally means that if the NameNode fails the log will be inaccessible.

We found that by using BookKeeper, the NameNode can log to distributed storage devices in a way that yields higher availability and performance.

Although it was designed for the NameNode, BookKeeper can be used for any application that needs strong durability guarantees with high performance and has a single writer.

**What are the bookkeeper elements and concepts?**

***Answer:***- BookKeeper uses four basic elements:

A) Ledger

B) BookKeeper Client

C) Bookie

D) Metadata Storage Service

**What is a Ledger in BookKeeper?**

***Answer:***- Ledger : A ledger is a sequence of entries, and each entry is a sequence of bytes. **OR** BookKeeper clients create logs called ledgers.

**What is a BookKeeper client in BookKeeper?**

***Answer:***- BookKeeper client : A client runs along with a BookKeeper application, and it enables applications to execute operations on ledgers, such as creating a ledger and writing to it;

**What is a Bookie in BookKeeper?**

***Answer:***- Bookie : A bookie is a BookKeeper storage server. Bookies store the content of ledgers.

**What is Metadata storage service in BookKeeper?**

***Answer:***- Metadata storage service : BookKeeper requires a metadata storage service to store information related to ledgers and available bookies. We currently use ZooKeeper for such a task.

**Why not just use zookeeper for everything?**

***Answer:***- There are a number of reasons:

1. Zookeeper’s log is only exposed through a tree like interface. It can be hard to shoehorn your application into this.

2. A zookeeper ensemble of multiple machines is limited to one log. You may want one log per resource, which will become expensive very quickly.

3. Adding extra machines to a zookeeper ensemble does not increase capacity nor throughput.

Bookkeeper can be viewed as a means of exposing zookeeper’s replicated log to applications in a scalable fashion. However, we still use zookeeper to maintain consistency guarantees.

**How do I size a ZooKeeper ensemble (cluster)?**

***Answer:***- In general when determining the number of ZooKeeper serving nodes to deploy (the size of an ensemble) you need to think in terms of reliability, and not performance.

**Reliability:**

A single ZooKeeper server (standalone) is essentially a coordinator with no reliability (a single serving node failure brings down the ZK service).

A 3 server ensemble (you need to jump to 3 and not 2 because ZK works based on simple majority voting) allows for a single server to fail and the service will still be available.

So if you want reliability go with at least 3. We typically recommend having 5 servers in "online" production serving environments.

This allows you to take 1 server out of service (say planned maintenance) and still be able to sustain an unexpected outage of one of the remaining servers w/o interruption of the service.

**Performance:**

Write performance actually \_decreases\_ as you add ZK servers, while read performance increases modestly: http://bit.ly/9JEUju

See this page for a survey Patrick Hunt (http://twitter.com/phunt) did looking at operational latency with both standalone server and an ensemble of size 3.

You'll notice that a single core machine running a standalone ZK ensemble (1 server) is still able to process 15k requests per second.

This is orders of magnitude greater than what most applications require

(if they are using ZooKeeper correctly - ie as a coordination service, and not as a replacement for a database, filestore, cache, etc...)

**Describe how HBase uses ZooKeeper?**

***Answer:***- HBase clients find the cluster to connect to by asking zookeeper. The only configuration a client needs is the zk quorum to connect to.

Masters and hbase slave nodes (regionservers) all register themselves with zk. If their znode evaporates, the master or regionserver is consided lost and repair begins.

**Can Apache Kafka be used without Zookeeper & Explain about ZooKeeper in Kafka ?**

***Answer:***- It is not possible to use Apache Kafka without Zookeeper because if the Zookeeper is down Kafka cannot serve client request.

Apache Kafka uses ZooKeeper to be a highly distributed and scalable system.

Zookeeper is used by Kafka to store various configurations and use them across the hadoop cluster in a distributed manner.

To achieve distributed-ness, configurations are distributed and replicated throughout the leader and follower nodes in the ZooKeeper ensemble.

We cannot directly connect to Kafka by bye-passing ZooKeeper because if the ZooKeeper is down it will not be able to serve the client request.

**Leader Election**

***Answer:***A simple way of doing leader election with ZooKeeper is to use the SEQUENCE|EPHEMERAL flags when creating znodes that represent "proposals" of clients.

The idea is to have a znode, say "/election", such that each znode creates a child znode "/election/n\_" with both flags SEQUENCE|EPHEMERAL. With the sequence flag, ZooKeeper automatically appends a sequence number that is greater that any one previously appended to a child of "/election".

The process that created the znode with the smallest appended sequence number is the leader.That's not all, though. It is important to watch for failures of the leader, so that a new client arises as the new leader in the case the current leader fails.

A trivial solution is to have all application processes watching upon the current smallest znode, and checking if they are the new leader when the smallest znode goes away (note that the smallest znode will go away if the leader fails because the node is ephemeral).

But this causes a herd effect: upon of failure of the current leader, all other processes receive a notification, and execute getChildren on "/election" to obtain the current list of children of "/election".

If the number of clients is large, it causes a spike on the number of operations that ZooKeeper servers have to process. To avoid the herd effect, it is sufficient to watch for the next znode down on the sequence of znodes.

If a client receives a notification that the znode it is watching is gone, then it becomes the new leader in the case that there is no smaller znode.

Note that this avoids the herd effect by not having all clients watching the same znode.

Here's the pseudo code: Let ELECTION be a path of choice of the application. To volunteer to be a leader:

Create znode z with path "ELECTION/n\_" with both SEQUENCE and EPHEMERAL flags;

Let C be the children of "ELECTION", and i be the sequence number of z;

Watch for changes on "ELECTION/n\_j", where j is the smallest sequence number such that j < i and n\_j is a znode in C;

Upon receiving a notification of znode deletion:

Let C be the new set of children of ELECTION;

If z is the smallest node in C, then execute leader procedure;

Otherwise, watch for changes on "ELECTION/n\_j", where j is the smallest sequence number such that j < i and n\_j is a znode in C;

Note that the znode having no preceding znode on the list of children does not imply that the creator of this znode is aware that it is the current leader.

Applications may consider creating a separate to znode to acknowledge that the leader has executed the leader procedure.

**Zookeeper QuorumPeerMain Configuration file**

dataDir - The directory where the ZooKeeper data is stored.

dataLogDir - The directory where the ZooKeeper transaction log is stored.

clientPort - The port used to communicate with clients.

tickTime - The duration of a tick in milliseconds. This is the basic unit of time in ZooKeeper.

initLimit - The maximum number of ticks that a follower will wait to initially synchronize with a leader.

syncLimit - The maximum number of ticks that a follower will wait for a message (including heartbeats) from the leader.

server.id - This is the host:port[:port] that the server with the given id will use for the quorum protocol.

**Mixed**

**What if my namenode is down and standby namenode is not coming up, what can be the issue?**

There might be so many possibilities reasons of above case, but further to clarify just sharing one of incident.

**One of Reason: -** Standby namenode and journal node configurations were in a corrupted state, so that when the cluster tried to switch to the standby, SNN was not building fsimage.

**Error: -** 2016-05-20 18:53:58,322 FATAL namenode.NameNode (NameNode.java:main(1512)) - Failed to start namenode. java.io.IOException: There appears to be a gap in the edit log.

**Solution:-**

1. [Put Active NN in safemode](https://community.hortonworks.com/questions/34636/unable-to-restrat-standby-namenode.html)

sudo -u hdfs hdfs dfsadmin -safemode enter

2. Do a savenamespace operation on Active NN

sudo -u hdfs hdfs dfsadmin -saveNamespace

3. Leave Safemode

sudo -u hdfs hdfs dfsadmin -safemode leave

4. Login to Standby NN

5. Run below command on Standby namenode to get latest fsimage that we saved in above steps.

sudo -u hdfs hdfs namenode -bootstrapStandby -force

**Setting the Proper Threshold Value for the Balancer**

You can run the balancer command without any parameters, as shown here:

$ sudo –u hdfs hdfs balancer

This balancer command uses the default threshold of 10 percent. This means that the balancer will balance data by moving blocks from over-utilized to under-utilized nodes,until each DataNode’s disk usage differs by no more than plus or minus 10 percent of the average disk usage in the cluster.

Sometimes, you may wish to set the threshold to a different level—for example, when free space in the cluster is getting low and you want to keep the used storage levels on the individual DataNodes within a smaller range than the default of plus or minus 10 percent.

You can do so by specifying the threshold parameter, as shown here:

$ hdfs balancer –threshold 5

**Tip:** How long the balancer will run depends on the size of the cluster and how unbalanced the data is. When you run the balancer for the very first time, or you schedule it infrequently, as well as when you run it after adding a set of DataNodes, it will run for a very long time—often several days.

The amount of data moved around during rebalancing depends on the value of the threshold parameter.

If you use the default value of 10 and the average DFS usage across the cluster is, for example, 70 percent, the balancer will ensure that that each DataNode’sDFS usage lies somewhere between 60 and 80 percent of that DataNode's storage capacity,once the balancing of the HDFS data is completed.

When you run the balancer, it looks at two key HDFS usage values in your cluster:

\* Average DFS used percentage: The average DFS used percentage in the cluster

can be derived by performing the following computation:

Average DFS Used = DFS Used \* 100/Present Capacity

\* A Node’s used DFS percentage: This measure shows the percentage of DFS used per node.

The balancer will balance a DataNode only if the difference between a DataNode’s

used DFS percentage and the average DFS used (by the cluster) is greater than the

threshold value. Otherwise, it won’t rebalance the cluster.

As noted previously, if you run the balancer without specifying a threshold value, it’ll use the default value of 10 as the threshold.

In our case, it won’t perform any balancing,ending up as shown here (assuming all the DataNodes have a similar DFS usage as that of Node10):

$ hdfs balancer

15/05/04 12:56:36 INFO balancer.Balancer: namenodes = [hdfs://hadoop01-ns]

15/05/04 12:56:36 INFO balancer.Balancer: parameters = Balancer

.Parameters[BalancingPolicy.Node, threshold=10.0, number of nodes to be excluded = 0,

number of nodes to be included = 0]

Time Stamp Iteration# Bytes Already Moved Bytes Left To Move

Bytes Being Moved

,,,

The cluster is balanced. Exiting...

May 4, 2015 12:56:37 PM Balancing took 1.47 seconds

$

The balancer ran, but it wound things up pretty quickly, because it found that all

nodes in the cluster have a usage that’s within the threshold value—the cluster is already balanced!

In our case, for balancing to occur, you must specify a threshold value that’s <=2.

Here’s one way to run it:

$ nohup su hdfs –c "hdfs balancer –threshold 2" > /tmp/balancer.log/stdout.log

2>/tmp/balancer.log/stderr.log &

Specifying nohup and & will run the job in the background and get back control of the shell. Since a balancer job can run for quite a long time in a cluster, it is a good idea to run it in this way.