What is BIG DATA?

Big Data is nothing but an assortment of such a huge and complex data that it becomes very tedious to capture, store, process, retrieve and analyze it with the help of on-hand database management tools or traditional data processing techniques.

To know more about BIG DATA, browse through **The Hype Behind Big Data!**

Can you give some examples of Big Data?

There are many real life examples of Big Data! Facebook is generating 500+ terabytes of data per day, NYSE (New York Stock Exchange) generates about 1 terabyte of new trade data per day, a jet airline collects 10 terabytes of censor data for every 30 minutes of flying time. All these are day to day examples of Big Data!

Can you give a detailed overview about the Big Data being generated by Facebook?

As of December 31, 2012, there are 1.06 billion monthly active users on facebook and 680 million mobile users. On an average, 3.2 billion likes and comments are posted every day on Facebook. 72% of web audience is on Facebook. And why not! There are so many activities going on facebook from wall posts, sharing images, videos, writing comments and liking posts, etc. In fact, Facebook started using Hadoop in mid-2009 and was one of the initial users of Hadoop.

According to IBM, what are the three characteristics of Big Data?

According to IBM, the three characteristics of Big Data are:

Volume: Facebook generating 500+ terabytes of data per day**.**
Velocity: Analyzing 2 million records each day to identify the reason for losses.
Variety: images, audio, video, sensor data, log files, etc.

How Big is 'Big Data'?

With time, data volume is growing exponentially. Earlier we used to talk about Megabytes or Gigabytes. But time has arrived when we talk about data volume in terms of terabytes, petabytes and also zettabytes! Global data volume was around 1.8ZB in 2011 and is expected to be 7.9ZB in 2015. It is also known that the global information doubles in every two years!

How analysis of Big Data is useful for organizations?

Effective analysis of Big Data provides a lot of business advantage as organizations will learn which areas to focus on and which areas are less important. Big data analysis provides some early key indicators that can prevent the company from a huge loss or help in grasping a great opportunity with open hands! A precise analysis of Big Data helps in decision making! For

instance, nowadays people rely so much on Facebook and Twitter before buying any product or service. All thanks to the Big Data explosion.

Who are 'Data Scientists'?

Data scientists are soon replacing business analysts or data analysts. Data scientists are experts who find solutions to analyze data. Just as web analysis, we have data scientists who have good business insight as to how to handle a business challenge. Sharp data scientists are not only involved in dealing business problems, but also choosing the relevant issues that can bring value-addition to the organization.

What is Hadoop?

Hadoop is a framework that allows for distributed processing of large data sets across clusters of commodity computers using a simple programming model.

Click on **What Is Hadoop all about** to know more!

Why the name 'Hadoop'?

Hadoop doesn't have any expanding version like 'oops'. The charming yellow elephant you see is basically named after Doug's son's toy elephant!

Why do we need Hadoop?

Everyday a large amount of unstructured data is getting dumped into our machines. The major challenge is not to store large data sets in our systems but to retrieve and analyze the big data in the organizations, that too data present in different machines at different locations. In this situation a necessity for Hadoop arises. Hadoop has the ability to analyze the data present in different machines at different locations very quickly and in a very cost effective way. It uses the concept of MapReduce which enables it to divide the query into small parts and process them in parallel. This is also known as parallel computing.

The link **Why Hadoop** gives you a detailed explanation about why Hadoop is gaining so much popularity!

What are some of the characteristics of Hadoop framework?

Hadoop framework is written in Java. It is designed to solve problems that involve analyzing large data (e.g. petabytes). The programming model is based on Google's MapReduce. The infrastructure is based on Google's Big Data and Distributed File System. Hadoop handles large files/data throughput and supports data intensive distributed applications. Hadoop is scalable as more nodes can be easily added to it.

Give a brief overview of Hadoop history.

In 2002, Doug Cutting created an open source, web crawler project.
In 2004, Google published MapReduce, GFS papers.
In 2006, Doug Cutting developed the open source, Mapreduce and HDFS project.
In 2008, Yahoo ran 4,000 node Hadoop cluster and Hadoop won terabyte sort benchmark.
In 2009, Facebook launched SQL support for Hadoop.

Give examples of some companies that are using Hadoop structure?

A lot of companies are using the Hadoop structure such as Cloudera, EMC, MapR, Hortonworks, Amazon, Facebook, eBay, Twitter, Google and so on.

What is the basic difference between traditional RDBMS and Hadoop?

Traditional RDBMS is used for transactional systems to report and archive the data, whereas *Hadoop* is an approach to store huge amount of data in the distributed file system and process it. RDBMS will be useful when you want to seek one record from Big data, whereas, Hadoop will be useful when you want Big data in one shot and perform analysis on that later.

What is structured and unstructured data?

Structured data is the data that is easily identifiable as it is organized in a structure. The most common form of structured data is a database where specific information is stored in tables, *that is,* rows and columns. Unstructured data refers to any data that cannot be identified easily. It could be in the form of images, videos, documents, email, logs and random text. It is not in the form of rows and columns.

What are the core components of Hadoop?

Core components of Hadoop are HDFS and MapReduce. HDFS is basically used to store large data sets and MapReduce is used to process such large data sets.

What is HDFS?

HDFS is a file system designed for storing very large files with streaming data access patterns, running clusters on commodity hardware.

What are the key features of HDFS?

HDFS is highly fault-tolerant, with high throughput, suitable for applications with large data sets, streaming access to file system data and can be built out of commodity hardware.

What is Fault Tolerance?

Suppose you have a file stored in a system, and due to some technical problem that file gets destroyed. Then there is no chance of getting the data back present in that file. To avoid such situations, Hadoop has introduced the feature of fault tolerance in HDFS. In Hadoop, when we

store a file, it automatically gets replicated at two other locations also. So even if one or two of the systems collapse, the file is still available on the third system.

Replication causes data redundancy then why is is pursued in HDFS?

HDFS works with commodity hardware (systems with average configurations) that has high chances of getting crashed any time. Thus, to make the entire system highly fault-tolerant, HDFS replicates and stores data in different places. Any data on HDFS gets stored at atleast 3 different locations. So, even if one of them is corrupted and the other is unavailable for some time for any reason, then data can be accessed from the third one. Hence, there is no chance of losing the data. This replication factor helps us to attain the feature of Hadoop called Fault Tolerant.

Since the data is replicated thrice in HDFS, does it mean that any calculation done on one node will also be replicated on the other two?

Since there are 3 nodes, when we send the MapReduce programs, calculations will be done only on the original data. The master node will know which node exactly has that particular data. In case, if one of the nodes is not responding, it is assumed to be failed. Only then, the required calculation will be done on the second replica.

What is throughput? How does HDFS get a good throughput?

Throughput is the amount of work done in a unit time. It describes how fast the data is getting accessed from the system and it is usually used to measure performance of the system. In HDFS, when we want to perform a task or an action, then the work is divided and shared among different systems. So all the systems will be executing the tasks assigned to them independently and in parallel. So the work will be completed in a very short period of time. In this way, the HDFS gives good throughput. By reading data in parallel, we decrease the actual time to read data tremendously.

What is streaming access?

As HDFS works on the principle of '*Write Once, Read Many*', the feature of streaming access is extremely important in HDFS. HDFS focuses not so much on storing the data but how to retrieve it at the fastest possible speed, especially while analyzing logs. In HDFS, reading the complete data is more important than the time taken to fetch a single record from the data.

What is a commodity hardware? Does commodity hardware include RAM?

Commodity hardware is a non-expensive system which is not of high quality or high-availability. Hadoop can be installed in any average commodity hardware. We don't need super computers or high-end hardware to work on Hadoop. Yes, Commodity hardware includes RAM because there will be some services which will be running on RAM.

What is a Namenode?

Namenode is the master node on which job tracker runs and consists of the metadata. It maintains and manages the blocks which are present on the datanodes. It is a high-availability machine and single point of failure in HDFS.

Is Namenode also a commodity?

No. Namenode can never be a commodity hardware because the entire HDFS rely on it. It is the single point of failure in HDFS. Namenode has to be a high-availability machine.

What is a metadata?

Metadata is the information about the data stored in datanodes such as location of the file, size of the file and so on.

What is a Datanode?

Datanodes are the slaves which are deployed on each machine and provide the actual storage. These are responsible for serving read and write requests for the clients.

Why do we use HDFS for applications having large data sets and not when there are lot of small files?

HDFS is more suitable for large amount of data sets in a single file as compared to small amount of data spread across multiple files. This is because Namenode is a very expensive high performance system, so it is not prudent to occupy the space in the Namenode by unnecessary amount of metadata that is generated for multiple small files. So, when there is a large amount of data in a single file, name node will occupy less space. Hence for getting optimized performance, HDFS supports large data sets instead of multiple small files.

What is a daemon?

Daemon is a process or service that runs in background. In general, we use this word in UNIX environment. The equivalent of Daemon in Windows is “services” and in Dos is ” TSR”.

What is a job tracker?

Job tracker is a daemon that runs on a namenode for submitting and tracking MapReduce jobs in Hadoop. It assigns the tasks to the different task tracker. In a Hadoop cluster, there will be only one job tracker but many task trackers. It is the single point of failure for Hadoop and MapReduce Service. If the job tracker goes down all the running jobs are halted. It receives heartbeat from task tracker based on which Job tracker decides whether the assigned task is completed or not.

What is a task tracker?

Task tracker is also a daemon that runs on datanodes. Task Trackers manage the execution of individual tasks on slave node. When a client submits a job, the job tracker will initialize the job and divide the work and assign them to different task trackers to perform MapReduce tasks. While performing this action, the task tracker will be simultaneously communicating with job tracker by sending heartbeat. If the job tracker does not receive heartbeat from task tracker within specified time, then it will assume that task tracker has crashed and assign that task to another task tracker in the cluster.

Is Namenode machine same as datanode machine as in terms of hardware?

It depends upon the cluster you are trying to create. The Hadoop VM can be there on the same machine or on another machine. For instance, in a single node cluster, there is only one machine, whereas in the development or in a testing environment, Namenode and datanodes are on different machines.

What is a heartbeat in HDFS?

A heartbeat is a signal indicating that it is alive. A datanode sends heartbeat to Namenode and task tracker will send its heart beat to job tracker. If the Namenode or job tracker does not receive heart beat then they will decide that there is some problem in datanode or task tracker is unable to perform the assigned task.

Are Namenode and job tracker on the same host?

No, in practical environment, Namenode is on a separate host and job tracker is on a separate host.

What is a 'block' in HDFS?

A 'block' is the minimum amount of data that can be read or written. In HDFS, the default block size is 64 MB as contrast to the block size of 8192 bytes in Unix/Linux. Files in HDFS are broken down into block-sized chunks, which are stored as independent units. HDFS blocks are large as compared to disk blocks, particularly to minimize the cost of seeks.

If a particular file is 50 mb, will the HDFS block still consume 64 mb as the default size?

No, not at all! 64 mb is just a unit where the data will be stored. In this particular situation, only 50 mb will be consumed by an HDFS block and 14 mb will be free to store something else. It is the MasterNode that does data allocation in an efficient manner.

What are the benefits of block transfer?

A file can be larger than any single disk in the network. There's nothing that requires the blocks from a file to be stored on the same disk, so they can take advantage of any of the disks in the cluster. Making the unit of abstraction a block rather than a file simplifies the storage subsystem. Blocks provide fault tolerance and availability. To insure against corrupted blocks

and disk and machine failure, each block is replicated to a small number of physically separate machines (typically three). If a block becomes unavailable, a copy can be read from another location in a way that is transparent to the client.

If we want to copy 10 blocks from one machine to another, but another machine can copy only 8.5 blocks, can the blocks be broken at the time of replication?

In HDFS, blocks cannot be broken down. Before copying the blocks from one machine to another, the Master node will figure out what is the actual amount of space required, how many block are being used, how much space is available, and it will allocate the blocks accordingly.

How indexing is done in HDFS?

Hadoop has its own way of indexing. Depending upon the block size, once the data is stored, HDFS will keep on storing the last part of the data which will say where the next part of the data will be. In fact, this is the base of HDFS.

If a data Node is full how it's identified?

When data is stored in datanode, then the metadata of that data will be stored in the Namenode. So Namenode will identify if the data node is full.

If datanodes increase, then do we need to upgrade Namenode?

While installing the Hadoop system, Namenode is determined based on the size of the clusters. Most of the time, we do not need to upgrade the Namenode because it does not store the actual data, but just the metadata, so such a requirement rarely arise.

Are job tracker and task trackers present in separate machines?

Yes, job tracker and task tracker are present in different machines. The reason is job tracker is a single point of failure for the Hadoop MapReduce service. If it goes down, all running jobs are halted.

When we send a data to a node, do we allow settling in time, before sending another data to that node?

Yes, we do.

Does hadoop always require digital data to process?

Yes. Hadoop always require digital data to be processed.

On what basis Namenode will decide which datanode to write on?

As the Namenode has the metadata (information) related to all the data nodes, it knows which datanode is free.

Doesn't Google have its very own version of DFS?

Yes, Google owns a DFS known as "*Google File System (GFS)*" developed by Google Inc. for its own use.

Who is a 'user' in HDFS?

A user is like you or me, who has some query or who needs some kind of data.

Is client the end user in HDFS?

No, Client is an application which runs on your machine, which is used to interact with the Namenode (job tracker) or datanode (task tracker).

What is the communication channel between client and namenode/datanode?

The mode of communication is SSH.

What is a rack?

Rack is a storage area with all the datanodes put together. These datanodes can be physically located at different places. Rack is a physical collection of datanodes which are stored at a single location. There can be multiple racks in a single location.

On what basis data will be stored on a rack?

When the client is ready to load a file into the cluster, the content of the file will be divided into blocks. Now the client consults the Namenode and gets 3 datanodes for every block of the file which indicates where the block should be stored. While placing the datanodes, the key rule followed is "f*or every block of data, two copies will exist in one rack, third copy in a different rack*". This rule is known as "*Replica Placement Policy*".

Do we need to place 2nd and 3rd data in rack 2 only?

Yes, this is to avoid datanode failure.

What if rack 2 and datanode fails?

If both rack2 and datanode present in rack 1 fails then there is no chance of getting data from it. In order to avoid such situations, we need to replicate that data more number of times instead of replicating only thrice. This can be done by changing the value in replication factor which is set to 3 by default.

What is a Secondary Namenode? Is it a substitute to the Namenode?

The secondary Namenode constantly reads the data from the RAM of the Namenode and writes it into the hard disk or the file system. It is not a substitute to the Namenode, so if the Namenode fails, the entire Hadoop system goes down.

What is the difference between Gen1 and Gen2 Hadoop with regards to the Namenode?

In Gen 1 Hadoop, Namenode is the single point of failure. In Gen 2 Hadoop, we have what is known as Active and Passive Namenodes kind of a structure. If the active Namenode fails, passive Namenode takes over the charge.

What is MapReduce?

Map Reduce is the '*heart*' of Hadoop that consists of two parts – 'map' and 'reduce'. Maps and reduces are programs for processing data. 'Map' processes the data first to give some intermediate output which is further processed by 'Reduce' to generate the final output. Thus, MapReduce allows for distributed processing of the map and reduction operations.

Can you explain how do 'map' and 'reduce' work?

Namenode takes the input and divide it into parts and assign them to data nodes. These datanodes process the tasks assigned to them and make a key-value pair and returns the intermediate output to the Reducer. The reducer collects this key value pairs of all the datanodes and combines them and generates the final output.

What is 'Key value pair' in HDFS?

Key value pair is the intermediate data generated by maps and sent to reduces for generating the final output.

What is the difference between MapReduce engine and HDFS cluster?

HDFS cluster is the name given to the whole configuration of master and slaves where data is stored. Map Reduce Engine is the programming module which is used to retrieve and analyze data.

Is map like a pointer?

No, Map is not like a pointer.

Do we require two servers for the Namenode and the datanodes?

Yes, we need two different servers for the Namenode and the datanodes. This is because Namenode requires highly configurable system as it stores information about the location details

of all the files stored in different datanodes and on the other hand, datanodes require low configuration system.

Why are the number of splits equal to the number of maps?

The number of maps is equal to the number of input splits because we want the key and value pairs of all the input splits.

Is a job split into maps?

No, a job is not split into maps. Spilt is created for the file. The file is placed on datanodes in blocks. For each split, a map is needed.

Which are the two types of 'writes' in HDFS?

There are two types of writes in HDFS: *posted and non-posted write*. Posted Write is when we write it and forget about it, without worrying about the acknowledgement. It is similar to our traditional Indian post. In a Non-posted Write, we wait for the acknowledgement. It is similar to the today's courier services. Naturally, non-posted write is more expensive than the posted write. It is much more expensive, though both writes are asynchronous.

Why '*Reading*' is done in parallel and '*Writing*' is not in HDFS?

Reading is done in parallel because by doing so we can access the data fast. But we do not perform the *write* operation in parallel. The reason is that if we perform the *write* operation in parallel, then it might result in data inconsistency. For example, you have a file and two nodes are trying to write data into the file in parallel, then the first node does not know what the second node has written and vice-versa. So, this makes it confusing which data to be stored and accessed.

Can Hadoop be compared to NOSQL database like Cassandra?

Though NOSQL is the closet technology that can be compared to Hadoop, it has its own pros and cons. There is no DFS in NOSQL. Hadoop is not a database. It's a filesystem (HDFS) and distributed programming framework (MapReduce).

What is a JobTracker in Hadoop? How many instances of JobTracker run on a Hadoop Cluster?

JobTracker is the daemon service for submitting and tracking MapReduce jobs in Hadoop. There is only One Job Tracker process run on any hadoop cluster. Job Tracker runs on its own JVM process. In a typical production cluster its run on a separate machine. Each slave node is configured with job tracker node location. The JobTracker is single point of failure for the Hadoop MapReduce service. If it goes down, all running jobs are halted. JobTracker in Hadoop

performs following actions(from Hadoop Wiki:)

Client applications submit jobs to the Job tracker.
The JobTracker talks to the NameNode to determine the location of the data
The JobTracker locates TaskTracker nodes with available slots at or near the data
The JobTracker submits the work to the chosen TaskTracker nodes.
The TaskTracker nodes are monitored. If they do not submit heartbeat signals often enough, they are deemed to have failed and the work is scheduled on a different TaskTracker.

A TaskTracker will notify the JobTracker when a task fails. The JobTracker decides what to do then: it may resubmit the job elsewhere, it may mark that specific record as something to avoid, and it may may even blacklist the TaskTracker as unreliable.

When the work is completed, the JobTracker updates its status.
Client applications can poll the JobTracker for information.

How JobTracker schedules a task?
The TaskTrackers send out heartbeat messages to the JobTracker, usually every few minutes, to reassure the JobTracker that it is still alive. These message also inform the JobTracker of the number of available slots, so the JobTracker can stay up to date with where in the cluster work can be delegated. When the JobTracker tries to find somewhere to schedule a task within the MapReduce operations, it first looks for an empty slot on the same server that hosts the DataNode containing the data, and if not, it looks for an empty slot on a machine in the same rack.

What is a Task Tracker in Hadoop? How many instances of TaskTracker run on a Hadoop Cluster
A TaskTracker is a slave node daemon in the cluster that accepts tasks (Map, Reduce and Shuffle operations) from a JobTracker. There is only One Task Tracker process run on any hadoop slave node. Task Tracker runs on its own JVM process. Every TaskTracker is configured with a set of slots, these indicate the number of tasks that it can accept. The TaskTracker starts a separate JVM processes to do the actual work (called as Task Instance) this is to ensure that process failure does not take down the task tracker. The TaskTracker monitors these task instances, capturing the output and exit codes. When the Task instances finish, successfully or not, the task tracker notifies the JobTracker. The TaskTrackers also send out heartbeat messages to the JobTracker, usually every few minutes, to reassure the JobTracker that it is still alive. These message also inform the JobTracker of the number of available slots, so the JobTracker can stay up to date with where in the cluster work can be delegated.

What is a Task instance in Hadoop? Where does it run?
Task instances are the actual MapReduce jobs which are run on each slave node. The TaskTracker starts a separate JVM processes to do the actual work (called as Task Instance) this is to ensure that process failure does not take down the task tracker. Each Task Instance runs on its own JVM process. There can be multiple processes of task instance running on a slave node. This is based on the number of slots configured on task tracker. By default a new task instance JVM process is spawned for a task.

How many Daemon processes run on a Hadoop system?
Hadoop is comprised of five separate daemons. Each of these daemon run in its own JVM. Following 3 Daemons run on Master nodes NameNode - This daemon stores and maintains the metadata for HDFS. Secondary NameNode - Performs housekeeping functions for the NameNode. JobTracker - Manages MapReduce jobs, distributes individual tasks to machines running the Task Tracker. Following 2 Daemons run on each Slave nodes DataNode – Stores actual HDFS data blocks. TaskTracker - Responsible for instantiating and monitoring individual Map and Reduce tasks.

What is configuration of a typical slave node on Hadoop cluster? How many JVMs run on a slave node?

Single instance of a Task Tracker is run on each Slave node. Task tracker is run as a separate JVM process.

Single instance of a DataNode daemon is run on each Slave node. DataNode daemon is run as a separate JVM process.

One or Multiple instances of Task Instance is run on each slave node. Each task instance is run as a separate JVM process. The number of Task instances can be controlled by configuration. Typically a high end machine is configured to run more task instances.

What is the difference between HDFS and NAS ?
The Hadoop Distributed File System (HDFS) is a distributed file system designed to run on commodity hardware. It has many similarities with existing distributed file systems. However, the differences from other distributed file systems are significant. Following are differences between HDFS and NAS

In HDFS Data Blocks are distributed across local drives of all machines in a cluster. Whereas in NAS data is stored on dedicated hardware.

HDFS is designed to work with MapReduce System, since computation are moved to data. NAS is not suitable for MapReduce since data is stored seperately from the computations.

HDFS runs on a cluster of machines and provides redundancy usinga replication protocal. Whereas NAS is provided by a single machine therefore does not provide data redundancy.

How NameNode Handles data node failures?
NameNode periodically receives a Heartbeat and a Blockreport from each of the DataNodes in the cluster. Receipt of a Heartbeat implies that the DataNode is functioning properly. A Blockreport contains a list of all blocks on a DataNode. When NameNode notices that it has not recieved a hearbeat message from a data node after a certain amount of time, the data node is marked as dead. Since blocks will be under replicated the system begins replicating the blocks that were stored on the dead datanode. The NameNode Orchestrates the replication of data blocks from one datanode to another. The replication data transfer happens directly between datanodes and the data never passes through the namenode.

Does MapReduce programming model provide a way for reducers to communicate with each other? In a MapReduce job can a reducer communicate with another reducer?

Nope, MapReduce programming model does not allow reducers to communicate with each other. Reducers run in isolation.

Can I set the number of reducers to zero?
Yes, Setting the number of reducers to zero is a valid configuration in Hadoop. When you set the reducers to zero no reducers will be executed, and the output of each mapper will be stored to a separate file on HDFS. [This is different from the condition when reducers are set to a number greater than zero and the Mappers output (intermediate data) is written to the Local file system(NOT HDFS) of each mappter slave node.]
Where is the Mapper Output (intermediate kay-value data) stored ?

The mapper output (intermediate data) is stored on the Local file system (NOT HDFS) of each individual mapper nodes. This is typically a temporary directory location which can be setup in config by the hadoop administrator. The intermediate data is cleaned up after the Hadoop Job completes.

What are combiners? When should I use a combiner in my MapReduce Job?
Combiners are used to increase the efficiency of a MapReduce program. They are used to aggregate intermediate map output locally on individual mapper outputs. Combiners can help you reduce the amount of data that needs to be transferred across to the reducers. You can use your reducer code as a combiner if the operation performed is commutative and associative. The execution of combiner is not guaranteed, Hadoop may or may not execute a combiner. Also, if required it may execute it more then 1 times. Therefore your MapReduce jobs should not depend on the combiners execution.

What is Writable & WritableComparable interface?
org.apache.hadoop.io.Writable is a Java interface. Any key or value type in the Hadoop Map-Reduce framework implements this interface. Implementations typically implement a static read(DataInput) method which constructs a new instance, calls readFields(DataInput) and returns the instance.
org.apache.hadoop.io.WritableComparable is a Java interface. Any type which is to be used as a key in the Hadoop Map-Reduce framework should implement this interface. WritableComparable objects can be compared to each other using Comparators.

What is the Hadoop MapReduce API contract for a key and value Class?
The Key must implement the org.apache.hadoop.io.WritableComparable interface.
The value must implement the org.apache.hadoop.io.Writable interface.

What is a IdentityMapper and IdentityReducer in MapReduce ?
org.apache.hadoop.mapred.lib.IdentityMapper Implements the identity function, mapping inputs directly to outputs. If MapReduce programmer do not set the Mapper Class using JobConf.setMapperClass then IdentityMapper.class is used as a default value.
org.apache.hadoop.mapred.lib.IdentityReducer Performs no reduction, writing all input values directly to the output. If MapReduce programmer do not set the Reducer Class using JobConf.setReducerClass then IdentityReducer.class is used as a default value.

What is the meaning of speculative execution in Hadoop? Why is it important?
Speculative execution is a way of coping with individual Machine performance. In large clusters where hundreds or thousands of machines are involved there may be machines which are not performing as fast as others. This may result in delays in a full job due to only one machine not performaing well. To avoid this, speculative execution in hadoop can run multiple copies of same map or reduce task on different slave nodes. The results from first node to finish are used.

When is the reducers are started in a MapReduce job?
In a MapReduce job reducers do not start executing the reduce method until the all Map jobs have completed. Reducers start copying intermediate key-value pairs from the mappers as soon as they are available. The programmer defined reduce method is called only after all the mappers have finished.
If reducers do not start before all mappers finish then why does the progress on MapReduce job shows something like Map(50%) Reduce(10%)? Why reducers progress percentage is displayed when mapper is not finished yet?

Reducers start copying intermediate key-value pairs from the mappers as soon as they are available. The progress calculation also takes in account the processing of data transfer which is done by reduce process, therefore the reduce progress starts showing up as soon as any intermediate key-value pair for a mapper is available to be transferred to reducer. Though the reducer progress is updated still the programmer defined reduce method is called only after all the mappers have finished.

What is HDFS ? How it is different from traditional file systems?
HDFS, the Hadoop Distributed File System, is responsible for storing huge data on the cluster. This is a distributed file system designed to run on commodity hardware. It has many similarities with existing distributed file systems. However, the differences from other distributed file systems are significant.

HDFS is highly fault-tolerant and is designed to be deployed on low-cost hardware.
HDFS provides high throughput access to application data and is suitable for applications that have large data sets.
HDFS is designed to support very large files. Applications that are compatible with HDFS are those that deal with large data sets. These applications write their data only once but they read it one or more times and require these reads to be satisfied at streaming speeds. HDFS supports write-once-read-many semantics on files.

What is HDFS Block size? How is it different from traditional file system block size?
In HDFS data is split into blocks and distributed across multiple nodes in the cluster. Each block is typically 64Mb or 128Mb in size. Each block is replicated multiple times. Default is to replicate each block three times. Replicas are stored on different nodes. HDFS utilizes the local file system to store each HDFS block as a separate file. HDFS Block size can not be compared with the traditional file system block size.

What is a NameNode? How many instances of NameNode run on a Hadoop Cluster?
The NameNode is the centerpiece of an HDFS file system. It keeps the directory tree of all files

in the file system, and tracks where across the cluster the file data is kept. It does not store the data of these files itself. There is only One NameNode process run on any hadoop cluster. NameNode runs on its own JVM process. In a typical production cluster its run on a separate machine. The NameNode is a Single Point of Failure for the HDFS Cluster. When the NameNode goes down, the file system goes offline. Client applications talk to the NameNode whenever they wish to locate a file, or when they want to add/copy/move/delete a file. The NameNode responds the successful requests by returning a list of relevant DataNode servers where the data lives.

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

How the Client communicates with HDFS?
The Client communication to HDFS happens using Hadoop HDFS API. Client applications talk to the NameNode whenever they wish to locate a file, or when they want to add/copy/move/delete a file on HDFS. The NameNode responds the successful requests by returning a list of relevant DataNode servers where the data lives. Client applications can talk directly to a DataNode, once the NameNode has provided the location of the data.

How the HDFS Blocks are replicated?
HDFS is designed to reliably store very large files across machines in a large cluster. It stores each file as a sequence of blocks; all blocks in a file except the last block are the same size. The blocks of a file are replicated for fault tolerance. The block size and replication factor are configurable per file. An application can specify the number of replicas of a file. The replication factor can be specified at file creation time and can be changed later. Files in HDFS are write-once and have strictly one writer at any time. The NameNode makes all decisions regarding replication of blocks. HDFS uses rack-aware replica placement policy. In default configuration there are total 3 copies of a datablock on HDFS, 2 copies are stored on datanodes on same rack and 3rd copy on a different rack.
- See more at: http://www.aired.in/2013/04/interview-questions-and-answers-for.html#sthash.EgjZJXWJ.dpuf

Name the most common InputFormats defined in Hadoop? Which one is default ?
Following 3 are most common InputFormats defined in Hadoop
- TextInputFormat
- KeyValueInputFormat
- SequenceFileInputFormat

TextInputFormat is the hadoop default.

What is the difference between TextInputFormat and KeyValueInputFormat class?
TextInputFormat: It reads lines of text files and provides the offset of the line as key to the Mapper and

actual line as Value to the mapper

KeyValueInputFormat: Reads text file and parses lines into key, val pairs. Everything up to the first tab character is sent as key to the Mapper and the remainder of the line is sent as value to the mapper.

What is InputSplit in Hadoop?
When a hadoop job is run, it splits input files into chunks and assign each split to a mapper to process. This is called Input Split

How is the splitting of file invoked in Hadoop Framework ?
It is invoked by the Hadoop framework by running getInputSplit() method of the Input format class (like FileInputFormat) defined by the user

Consider case scenario: In M/R system,
- HDFS block size is 64 MB
- Input format is FileInputFormat
- We have 3 files of size 64K, 65Mb and 127Mb

then how many input splits will be made by Hadoop framework?
Hadoop will make 5 splits as follows
- 1 split for 64K files
- 2 splits for 65Mb files
- 2 splits for 127Mb file

What is the purpose of RecordReader in Hadoop?
The InputSplit has defined a slice of work, but does not describe how to access it. The RecordReader class actually loads the data from its source and converts it into (key, value) pairs suitable for reading by the Mapper. The RecordReader instance is defined by the InputFormat

After the Map phase finishes, the hadoop framework does "Partitioning, Shuffle and sort". Explain what happens in this phase?
- Partitioning
Partitioning is the process of determining which reducer instance will receive which intermediate keys and values. Each mapper must determine for all of its output (key, value) pairs which reducer will receive them. It is necessary that for any key, regardless of which mapper instance generated it, the destination partition is the same

- Shuffle
After the first map tasks have completed, the nodes may still be performing several more map tasks each. But they also begin exchanging the intermediate outputs from the map tasks to where they are required by the reducers. This process of moving map outputs to the reducers is known as shuffling.

- Sort
Each reduce task is responsible for reducing the values associated with several intermediate keys. The set of intermediate keys on a single node is automatically sorted by Hadoop before they are presented to the

Reducer

If no custom partitioner is defined in the hadoop then how is data partitioned before its sent to the reducer?
The default partitioner computes a hash value for the key and assigns the partition based on this result

What is a Combiner?
The Combiner is a "mini-reduce" process which operates only on data generated by a mapper. The Combiner will receive as input all data emitted by the Mapper instances on a given node. The output from the Combiner is then sent to the Reducers, instead of the output from the Mappers.

What is job tracker?
Job Tracker is the service within Hadoop that runs Map Reduce jobs on the cluster

What are some typical functions of Job Tracker?
The following are some typical tasks of Job Tracker
- Accepts jobs from clients
- It talks to the NameNode to determine the location of the data
- It locates TaskTracker nodes with available slots at or near the data
- It submits the work to the chosen Task Tracker nodes and monitors progress of each task by receiving heartbeat signals from Task tracker

What is task tracker?
Task Tracker is a node in the cluster that accepts tasks like Map, Reduce and Shuffle operations - from a JobTracker

Whats the relationship between Jobs and Tasks in Hadoop?
One job is broken down into one or many tasks in Hadoop.

Suppose Hadoop spawned 100 tasks for a job and one of the task failed. What will hadoop do ?
It will restart the task again on some other task tracker and only if the task fails more than 4 (default setting and can be changed) times will it kill the job

Hadoop achieves parallelism by dividing the tasks across many nodes, it is possible for a few slow nodes to rate-limit the rest of the program and slow down the program. What mechanism Hadoop provides to combat this ?
Speculative Execution

How does speculative execution works in Hadoop ?
Job tracker makes different task trackers process same input. When tasks complete, they announce this fact to the Job Tracker. Whichever copy of a task finishes first becomes the definitive copy. If other copies were executing speculatively, Hadoop tells the Task Trackers to abandon the tasks and discard their outputs. The Reducers then receive their inputs from whichever Mapper completed successfully, first.

Using command line in Linux, how will you
- see all jobs running in the hadoop cluster
- kill a job
- hadoop job -list
- hadoop job -kill jobid

What is Hadoop Streaming ?
Streaming is a generic API that allows programs written in virtually any language to be used as Hadoop Mapper and Reducer implementations

What is the characteristic of streaming API that makes it flexible run map reduce jobs in languages like perl, ruby, awk etc. ?
Hadoop Streaming allows to use arbitrary programs for the Mapper and Reducer phases of a Map Reduce job by having both Mappers and Reducers receive their input on stdin and emit output (key, value) pairs on stdout.

Whats is Distributed Cache in Hadoop ?
Distributed Cache is a facility provided by the Map/Reduce framework to cache files (text, archives, jars and so on) needed by applications during execution of the job. The framework will copy the necessary files to the slave node before any tasks for the job are executed on that node.

What is the benifit of Distributed cache, why can we just have the file in HDFS and have the application read it ?
This is because distributed cache is much faster. It copies the file to all trackers at the start of the job. Now if the task tracker runs 10 or 100 mappers or reducer, it will use the same copy of distributed cache. On the other hand, if you put code in file to read it from HDFS in the MR job then every mapper will try to access it from HDFS hence if a task tracker run 100 map jobs then it will try to read this file 100 times from HDFS. Also HDFS is not very efficient when used like this.

What mechanism does Hadoop framework provides to synchronize changes made in Distribution Cache during runtime of the application ?
This is a trick questions. There is no such mechanism. Distributed Cache by design is read only during the time of Job execution

Have you ever used Counters in Hadoop. Give us an example scenario ?
Anybody who claims to have worked on a Hadoop project is expected to use counters

Is it possible to provide multiple input to Hadoop? If yes then how can you give multiple directories as

input to the Hadoop job ?

Yes, The input format class provides methods to add multiple directories as input to a Hadoop job

Is it possible to have Hadoop job output in multiple directories. If yes then how ?

Yes, by using Multiple Outputs class

What will a hadoop job do if you try to run it with an output directory that is already present? Will it
- overwrite it
- warn you and continue
- throw an exception and exit

The hadoop job will throw an exception and exit.

How can you set an arbitrary number of mappers to be created for a job in Hadoop ?

This is a trick question. You cannot set it

How can you set an arbitary number of reducers to be created for a job in Hadoop ?

You can either do it progamatically by using method setNumReduceTasksin the JobConfclass or set it up as a configuration setting

How will you write a custom partitioner for a Hadoop job ?

To have hadoop use a custom partitioner you will have to do minimum the following three
- Create a new class that extends Partitioner class
- Override method getPartition
- In the wrapper that runs the Map Reducer, either
- add the custom partitioner to the job programtically using method setPartitionerClass or
- add the custom partitioner to the job as a config file (if your wrapper reads from config file or oozie)

How did you debug your Hadoop code ?

There can be several ways of doing this but most common ways are
- By using counters
- The web interface provided by Hadoop framework

Did you ever built a production process in Hadoop ? If yes then what was the process when your hadoop job fails due to any reason?

Its an open ended question but most candidates, if they have written a production job, should talk about some type of alert mechanisn like email is sent or there monitoring system sends an alert. Since Hadoop works on unstructured data, its very important to have a good alerting system for errors since unexpected

data can very easily break the job.

Did you ever ran into a lop sided job that resulted in out of memory error, if yes then how did you handled it ?

This is an open ended question but a candidate who claims to be an intermediate developer and has worked on large data set (10-20GB min) should have run into this problem. There can be many ways to handle this problem but most common way is to alter your algorithm and break down the job into more map reduce phase or use a combiner if possible.

What is HDFS?

HDFS, the Hadoop Distributed File System, is a distributed file system designed to hold very large amounts of data (terabytes or even petabytes), and provide high-throughput access to this information. Files are stored in a redundant fashion across multiple machines to ensure their durability to failure and high availability to very parallel applications

What does the statement "HDFS is block structured file system" means?

It means that in HDFS individual files are broken into blocks of a fixed size. These blocks are stored across a cluster of one or more machines with data storage capacity

What does the term "Replication factor" mean?

Replication factor is the number of times a file needs to be replicated in HDFS

What is the default replication factor in HDFS?

3

What is the default block size of an HDFS block?

64Mb

What is the benefit of having such big block size (when compared to block size of linux file system like ext)?

It allows HDFS to decrease the amount of metadata storage required per file (the list of blocks per file will be smaller as the size of individual blocks increases). Furthermore, it allows for fast streaming reads of data, by keeping large amounts of data sequentially laid out on the disk

Why is it recommended to have few very large files instead of a lot of small files in HDFS?

This is because the Name node contains the meta data of each and every file in HDFS and more files means more metadata and since namenode loads all the metadata in memory for speed hence having a lot of files may make the metadata information big enough to exceed the size of the memory on the Name node

True/false question. What is the lowest granularity at which you can apply replication factor in HDFS

- You can choose replication factor per directory

- You can choose replication factor per file in a directory
- You can choose replication factor per block of a file
- True
- True
- False

What is a datanode in HDFS?
Individual machines in the HDFS cluster that hold blocks of data are called datanodes

What is a Namenode in HDFS?
The Namenode stores all the metadata for the file system

What alternate way does HDFS provides to recover data in case a Namenode, without backup, fails and cannot be recovered?
There is no way. If Namenode dies and there is no backup then there is no way to recover data

Describe how a HDFS client will read a file in HDFS, like will it talk to data node or namenode ... how will data flow etc?
To open a file, a client contacts the Name Node and retrieves a list of locations for the blocks that comprise the file. These locations identify the Data Nodes which hold each block. Clients then read file data directly from the Data Node servers, possibly in parallel. The Name Node is not directly involved in this bulk data transfer, keeping its overhead to a minimum.

Using linux command line. how will you
- List the the number of files in a HDFS directory
- Create a directory in HDFS
- Copy file from your local directory to HDFS
hadoop fs -ls
hadoop fs -mkdir
hadoop fs -put localfile hdfsfile

Advantages of Hadoop?
• Bringing compute and storage together on commodity hardware: The result is blazing speed at low cost.
• Price performance: The Hadoop big data technology provides significant cost savings (think a factor of approximately 10) with significant performance improvements (again, think factor of 10). Your mileage may vary. If the existing technology can be so dramatically trounced, it is worth examining if Hadoop can complement or replace aspects of your current architecture.
• Linear Scalability: Every parallel technology makes claims about scale up.Hadoop has genuine scalability since the latest release is expanding the limit on the number of nodes to beyond 4,000.
• Full access to unstructured data: A highly scalable data store with a good parallel programming model, MapReduce, has been a challenge for the industry for some time. Hadoop programming model does not solve all problems, but it is a strong solution for many tasks.

Definition of Big data?

According to Gartner, Big data can be defined as high volume, velocity and variety information requiring innovative and cost effective forms of information processing for enhanced decision making.

How Big data differs from database ?

Datasets which are beyond the ability of the database to store, analyze and manage can be defined as Big. The technology extracts required information from large volume whereas the storage area is limited for a database.

Who are all using Hadoop? Give some examples?

- A9.com
- Amazon
- Adobe
- AOL
- Baidu
- Cooliris
- Facebook
- NSF-Google
- IBM
- LinkedIn
- Ning
- PARC
- Rackspace
- StumbleUpon
- Twitter
- Yahoo!

Pig for Hadoop - Give some points?

Pig is Data-flow oriented language for analyzing large data sets.
It is a platform for analyzing large data sets that consists of a high-level language for expressing data analysis programs, coupled with infrastructure for evaluating these programs. The salient property of Pig programs is that their structure is amenable to substantial parallelization, which in turns enables them to handle very large data sets.

At the present time, Pig infrastructure layer consists of a compiler that produces sequences of Map-Reduce programs, for which large-scale parallel implementations already exist (e.g., the Hadoop subproject). Pig language layer currently consists of a textual language called Pig Latin, which has the following key properties:

Ease of programming.
It is trivial to achieve parallel execution of simple, "embarrassingly parallel" data analysis tasks. Complex tasks comprised of multiple interrelated data transformations are explicitly encoded as data flow sequences, making them easy to write, understand, and maintain.

Optimization opportunities.
The way in which tasks are encoded permits the system to optimize their execution automatically, allowing the user to focus on semantics rather than efficiency.

Extensibility.
Users can create their own functions to do special-purpose processing.

Features of Pig:
– data transformation functions
– datatypes include sets, associative arrays, tuples
– high-level language for marshalling data
- developed at yahoo!

Hive for Hadoop - Give some points?

Hive is a data warehouse system for Hadoop that facilitates easy data summarization, ad-hoc queries, and the analysis of large datasets stored in Hadoop compatible file systems. Hive provides a mechanism to project structure onto this data and query the data using a SQL-like language called HiveQL. At the same time this language also allows traditional map/reduce programmers to plug in their custom mappers and reducers when it is inconvenient or inefficient to express this logic in HiveQL.

Keypoints:
• SQL-based data warehousing application
– features similar to Pig
– more strictly SQL-type
• Supports SELECT, JOIN, GROUP BY,etc
• Analyzing very large data sets
– log processing, text mining, document indexing
• Developed at Facebook

Map Reduce in Hadoop?

Map reduce :
it is a framework for processing in parallel across huge datasets usning large no. of computers referred to cluster, it involves two processes namely Map and reduce.

Map Process:
In this process input is taken by the master node,which divides it into smaller tasks and distribute them to the workers nodes. The workers nodes process these sub tasks and pass them back to the master node.

Reduce Process :
In this the master node combines all the answers provided by the worker nodes to get the results of the original task. The main advantage of Map reduce is that the map and reduce are performed in distributed mode. Since each operation is independent, so each map can be performed in parallel and hence reducing the net computing time.

What is a heartbeat in HDFS?

A heartbeat is a signal indicating that it is alive. A data node sends heartbeat to Name node and task tracker will send its heart beat to job tracker. If the Name node or job tracker does not receive heart beat then they will decide that there is some problem in data node or task tracker is unable to perform the assigned task.

What is a metadata?

Metadata is the information about the data stored in data nodes such as location of the file, size of the file and so on.

Is Namenode also a commodity?

No. Namenode can never be a commodity hardware because the entire HDFS rely on it.
It is the single point of failure in HDFS. Namenode has to be a high-availability machine.

Can Hadoop be compared to NOSQL database like Cassandra?

Though NOSQL is the closet technology that can be compared to Hadoop, it has its own pros and cons. There is no DFS in NOSQL. Hadoop is not a database. It's a filesystem (HDFS) and distributed programming framework (MapReduce).

What is Key value pair in HDFS?

Key value pair is the intermediate data generated by maps and sent to reduces for generating the final output.

What is the difference between MapReduce engine and HDFS cluster?

HDFS cluster is the name given to the whole configuration of master and slaves where data is stored. Map Reduce Engine is the programming module which is used to retrieve and analyze data.

What is a rack?

Rack is a storage area with all the datanodes put together. These datanodes can be physically located at different places. Rack is a physical collection of datanodes which are stored at a single location. There can be multiple racks in a single location.

How indexing is done in HDFS?

Hadoop has its own way of indexing. Depending upon the block size, once the data is stored, HDFS will keep on storing the last part of the data which will say where the next part of the data will be. In fact, this is the base of HDFS.

History of Hadoop?

Hadoop was created by Doug Cutting, the creator of Apache Lucene, the widely used text search library. Hadoop has its origins in Apache Nutch, an open source web search engine, itself a part of the Lucene project.

The name Hadoop is not an acronym; it's a made-up name. The project's creator, Doug Cutting, explains

how the name came about:
The name my kid gave a stuffed yellow elephant. Short, relatively easy to spell and pronounce, meaningless, and not used elsewhere: those are my naming criteria.

Subprojects and "contrib" modules in Hadoop also tend to have names that are unrelated to their function, often with an elephant or other animal theme ("Pig," for example). Smaller components are given more descriptive (and therefore more mundane) names. This is a good principle, as it means you can generally work out what something does from its name. For example, the jobtracker keeps track of MapReduce jobs.

What is meant by Volunteer Computing?

Volunteer computing projects work by breaking the problem they are trying to solve into chunks called work units, which are sent to computers around the world to be analyzed.
SETI@home is the most well-known of many volunteer computing projects.

How Hadoop differs from SETI (Volunteer computing)?

Although SETI (Search for Extra-Terrestrial Intelligence) may be superficially similar to MapReduce (breaking a problem into independent pieces to be worked on in parallel), there are some significant differences. The SETI@home problem is very CPU-intensive, which makes it suitable for running on hundreds of thousands of computers across the world. Since the time to transfer the work unit is dwarfed by the time to run the computation on it. Volunteers are donating CPU cycles, not bandwidth.

MapReduce is designed to run jobs that last minutes or hours on trusted, dedicated hardware running in a single data center with very high aggregate bandwidth interconnects. By contrast, SETI@home runs a perpetual computation on untrusted machines on the Internet with highly variable connection speeds and no data locality.

Compare RDBMS and MapReduce?

Data size:
RDBMS - Gigabytes
MapReduce - Petabytes
Access:
RDBMS - Interactive and batch
MapReduce - Batch
Updates:
RDBMS - Read and write many times
MapReduce - Write once, read many times
Structure:
RDBMS - Static schema
MapReduce - Dynamic schema
Integrity:
RDBMS - High
MapReduce - Low
Scaling:

RDBMS - Nonlinear
MapReduce - Linear

What is HBase?
A distributed, column-oriented database. HBase uses HDFS for its underlying storage, and supports both batch-style computations using MapReduce and point queries (random reads).

What is ZooKeeper?
A distributed, highly available coordination service. ZooKeeper provides primitives such as distributed locks that can be used for building distributed applications.

What is Chukwa?
A distributed data collection and analysis system. Chukwa runs collectors that store data in HDFS, and it uses MapReduce to produce reports. (At the time of this writing, Chukwa had only recently graduated from a "contrib" module in Core to its own subproject.)

What is Avro?
A data serialization system for efficient, cross-language RPC, and persistent data storage. (At the time of this writing, Avro had been created only as a new subproject, and no other Hadoop subprojects were using it yet.)

core subproject in Hadoop - What is it?
A set of components and interfaces for distributed filesystems and general I/O (serialization, Java RPC, persistent data structures).

What are all Hadoop subprojects?
Pig, Chukwa, Hive, HBase, MapReduce, HDFS, ZooKeeper, Core, Avro

What is a split?
Hadoop divides the input to a MapReduce job into fixed-size pieces called input splits, or just splits. Hadoop creates one map task for each split, which runs the userdefined map function for each record in the split.

Having many splits means the time taken to process each split is small compared to the time to process the whole input. So if we are processing the splits in parallel, the processing is better load-balanced.

On the other hand, if splits are too small, then the overhead of managing the splits and of map task creation begins to dominate the total job execution time. For most jobs, a good split size tends to be the size of a HDFS block, 64 MB by default, although this can be changed for the cluster

Map tasks write their output to local disk, not to HDFS. Why is this?
Map output is intermediate output: it's processed by reduce tasks to produce the final output, and once the job is complete the map output can be thrown away. So storing it in HDFS, with replication, would be overkill. If the node running the map task fails before the map output has been consumed by the reduce

task, then Hadoop will automatically rerun the map task on another node to recreate the map output.

MapReduce data flow with a single reduce task- Explain?

The input to a single reduce task is normally the output from all mappers.
The sorted map outputs have to be transferred across the network to the node where the reduce task is running, where they are merged and then passed to the user-defined reduce function. The output of the reduce is normally stored in HDFS for reliability.
For each HDFS block of the reduce output, the first replica is stored on the local node, with other replicas being stored on off-rack nodes.

MapReduce data flow with multiple reduce tasks- Explain?

When there are multiple reducers, the map tasks partition their output, each creating one partition for each reduce task. There can be many keys (and their associated values) in each partition, but the records for every key are all in a single partition. The partitioning can be controlled by a user-defined partitioning function, but normally the default partitioner.

MapReduce data flow with no reduce tasks- Explain?

It's also possible to have zero reduce tasks. This can be appropriate when you don't need the shuffle since the processing can be carried out entirely in parallel.
In this case, the only off-node data transfer is used when the map tasks write to HDFS

What is a block in HDFS?

Filesystems deal with data in blocks, which are an integral multiple of the disk block size. Filesystem blocks are typically a few kilobytes in size, while disk blocks are normally 512 bytes.

Why is a Block in HDFS So Large?

HDFS blocks are large compared to disk blocks, and the reason is to minimize the cost of seeks. By making a block large enough, the time to transfer the data from the disk can be made to be significantly larger than the time to seek to the start of the block. Thus the time to transfer a large file made of multiple blocks operates at the disk transfer rate.

File permissions in HDFS?

HDFS has a permissions model for files and directories.
There are three types of permission: the read permission (r), the write permission (w) and the execute permission (x). The read permission is required to read files or list the contents of a directory. The write permission is required to write a file, or for a directory, to create or delete files or directories in it. The execute permission is ignored for a file since you can't execute a file on HDFS.

What is Thrift in HDFS?

The Thrift API in the "thriftfs" contrib module exposes Hadoop filesystems as an Apache Thrift service, making it easy for any language that has Thrift bindings to interact with a Hadoop filesystem, such as HDFS.
To use the Thrift API, run a Java server that exposes the Thrift service, and acts as a proxy to the Hadoop filesystem. Your application accesses the Thrift service, which is typically running on the same machine

as your application.

How Hadoop interacts with C?

Hadoop provides a C library called libhdfs that mirrors the Java FileSystem interface.
It works using the Java Native Interface (JNI) to call a Java filesystem client.
The C API is very similar to the Java one, but it typically lags the Java one, so newer features may not be supported. You can find the generated documentation for the C API in the libhdfs/docs/api directory of the Hadoop distribution.

What is FUSE in HDFS Hadoop?

Filesystem in Userspace (FUSE) allows filesystems that are implemented in user space to be integrated as a Unix filesystem. Hadoop's Fuse-DFS contrib module allows any Hadoop filesystem (but typically HDFS) to be mounted as a standard filesystem. You can then use Unix utilities (such as ls and cat) to interact with the filesystem.
Fuse-DFS is implemented in C using libhdfs as the interface to HDFS. Documentation for compiling and running Fuse-DFS is located in the src/contrib/fuse-dfs directory of the Hadoop distribution.

Explain WebDAV in Hadoop?

WebDAV is a set of extensions to HTTP to support editing and updating files. WebDAV shares can be mounted as filesystems on most operating systems, so by exposing HDFS (or other Hadoop filesystems) over WebDAV, it's possible to access HDFS as a standard filesystem.

What is Sqoop in Hadoop?

It is a tool design to transfer the data between Relational database management system(RDBMS) and Hadoop HDFS.
Thus, we can sqoop the data from RDBMS like mySql or Oracle into HDFS of Hadoop as well as exporting data from HDFS file to RDBMS.
Sqoop will read the table row-by-row and the import process is performed in Parallel. Thus, the output may be in multiple files.
Example:
sqoop INTO "directory";
(SELECT * FROM database.table WHERE condition;)