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.

What are the default configuration files that are used in Hadoop
As of 0.20 release, Hadoop supported the following read-only default configurations
– src/core/core-default.xml
– src/hdfs/hdfs-default.xml
– src/mapred/mapred-default.xml

How will you make changes to the default configuration files
Hadoop does not recommends changing the default configuration files, instead it recommends making all site specific changes in the following files
– conf/core-site.xml
– conf/hdfs-site.xml
– conf/mapred-site.xml

Unless explicitly turned off, Hadoop by default specifies two resources, loaded in-order from the classpath:
– core-default.xml : Read-only defaults for hadoop.
– core-site.xml: Site-specific configuration for a given hadoop installation.

Hence if same configuration is defined in file *core-default.xml* and *src/core/core-default.xml* then the values in file *core-default.xml* (same is true for other 2 file pairs) is used.

Consider case scenario where you have set property mapred.output.compress totrue to ensure that all output files are compressed for efficient space usage on the cluster. If a cluster user does not want to compress data for a specific job then what will you recommend him to do ?
Ask him to create his own configuration file and specify
configuration *mapred.output.compress*to *false* and load this file as a resource in his job.

In the above case scenario, how can ensure that user cannot override the
configuration mapred.output.compress to false in any of his jobs
This can be done by setting the property *final* to *true* in the *core-site.xml* file

What of the following is the only required variable that needs to be set in file conf/hadoop-env.sh for hadoop to work

- HADOOP_LOG_DIR

- JAVA_HOME

- HADOOP_CLASSPATH
The only required variable to set is JAVA_HOME that needs to point to <java installation> directory

List all the daemons required to run the Hadoop cluster
– NameNode
– DataNode
– JobTracker
– TaskTracker

Whats the default port that jobtrackers listens to
50030

Whats the default port where the dfs namenode web ui will listen on
50070

Hadoop Distributed File Systems (HDFS) Interview Questions

Q1. 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

Q2. 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

Q3. What does the term "Replication factor" mean Replication factor is the number of times a file needs to be replicated in HDFS

Q4. What is the default replication factor in HDFS 3

Q5. What is the typical block size of an HDFS block 64Mb to 128Mb

Q6. 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

Q7. 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

Q8. True/false question. What is the lowest granularity at which you can apply replication factor in HDSF
- 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

Q9. What is a datanode in HDFS ndividual machines in the HDFS cluster that hold blocks of data are called datanodes

Q10. What is a Namenode in HDSF The Namenode stores all the metadata for the file system

Q11. 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

Q12. 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.

Q13. 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 HDSF

```
hadoop fs -ls
hadoop fs -mkdir
hadoop fs -put localfile hdfsfile
```

Q31. 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)

Q32. 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

Q33. 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.

Q34. 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.

Q22. 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.

Q23. 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.

Q.24 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

Q25. 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

Q26. 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

Q27. Is it possible to have Hadoop job output in multiple directories. If yes then how
Yes, by using Multiple Outputs class

Q28. 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.

Q29. How can you set an arbitary number of mappers to be created for a job in Hadoop
This is a trick question. You cannot set it

Q30. 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

==

1. What is the difference between a Hadoop database and Relational Database?
Structured data is data that is organized into entities that have a defined format, such as XML documents or database tables that conform to a particular predefined schema. This is the realm of the RDBMS. but, map reduce works with semi structured data (spread sheet), structured data (image files, plain text) because the input keys and values for MapReduce are not an intrinsic property of the data, but they are chosen by the person analyzing the data.

2. What is Hadoop framework?
• Solution for big data
– Deals with complexities of high volume, velocity & variety of data
• Set of open source projects
• Transforms commodity hardware into a service that:
– Stores petabytes of data reliably
– Allows huge distributed computations
• Key attributes:
– Redundant and reliable (no data loss)
– Extremely powerful
– Batch processing centric
– Easy to program distributed apps
– Runs on commodity hardware
– fault tolarrent
– scalable
– cost effective

3. On What concept the Hadoop framework works?
(Divide and concore) Works based on Map-Reduce , which splits the problem into small chunks and solved individually and finally results will consolidated at end.

4. 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.

HDFS is a block-structured file system: 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. Individual machines in the cluster are referred to as DataNodes.

5. what is MAP REDUCE?
MapReduce is a programming model for processing large data sets, and the name of an implementation of the model by Google. MapReduce is typically used to do distributed computing on clusters of computers.
The model is inspired by the map and reduce functions commonly used in functional programming, although their purpose in the MapReduce framework is not the same as their original forms.
MapReduce libraries have been written in many programming languages. A popular free implementation is Apache Hadoop.

Overview
MapReduce is a framework for processing embarrassingly parallel problems across huge datasets using a large number of computers (nodes), collectively referred to as a cluster (if all nodes are on the same local network and use similar hardware) or a grid (if the nodes are shared across geographically and administratively distributed systems, and use more heterogenous hardware). Computational processing can occur on data stored either in a filesystem (unstructured) or in a database (structured). MapReduce can take advantage of locality of data, processing data on or near the storage assets to decrease transmission of data.

"Map" step: The master node takes the input, divides it into smaller sub-problems, and distributes them to worker nodes. A worker node may do this again in turn, leading to a multi-level tree structure. The worker node processes the smaller problem, and passes the answer back to its master node.

"Reduce" step: The master node then collects the answers to all the sub-problems and combines them in some way to form the output – the answer to the problem it was originally trying to solve.

Source:http://en.wikipedia.org/wiki/MapReduce

6. What Mapper does?
Mapper reads a file line by line (streaming) way and takes the byte offset of each line as key and the whole line as value and define it as pairs. It passes these key value pairs to the intermediate step before a Mapper starts processing the key value pairs

7. What is the InputSplit in map reduce software?
An inputsplit is the slice of data to be processed by a single Mapper. It generally is of the block size which is stored on the datanode.
InputSplit represents the data to be processed by an individual Mapper.

8. what is meaning Replication factor?
Replication factor defines the number of times a given data block is stored in the cluster. The default replication factor is 3. This also means that you need to have 3times the amount of storage needed to store the data. Each file is split into data blocks and spread across the cluster.

9. Which interface needs to be implemented to create Mapper and Reducer for the Hadoop?
Mapper and Reducer are abstract class.
We just need to extend Mapper and implement map method for Mapper and extend Reducer class and implement reduce method. No interfaces needed.

10. What is the InputFormat ?
The default InputFormat is the TextInputFormat. This treats each line of each input file as a separate record, and performs no parsing. This is useful for unformatted data or line-based records like log files. A more interesting input format is the KeyValueInputFormat. This format also treats each line of input as a separate record. While the TextInputFormat treats the entire line as the value, the KeyValueInputFormat breaks the line itself into the key and value by searching for a tab character. This is particularly useful for reading the output of one MapReduce job as the input to another, as the default OutputFormat (described in more detail below) formats its results in this manner. Finally, the SequenceFileInputFormat reads special binary files that are specific to Hadoop. These files include many features designed to allow data to be rapidly read into Hadoop mappers. Sequence files are block-compressed and provide direct serialization and deserialization of several arbitrary data types (not just text). Sequence files can be generated as the output of other MapReduce tasks and are an efficient intermediate representation for data that is passing from one MapReduce job to anther.

11. what is a datanode?
There are a Number of datanodes, usually one per node in the cluster, which manage storage attached to the nodes. Internally , a file is split into one or more blocks and these blocks are stored in a set of datanodes. The Datanodes are responsible for serving read and write requests from the file system's clients.

12. what is namenode?
Name node is one of the daemon that runs in Master node and holds the meta info where particular chunk of data (ie. data node) resides.Based on meta info maps the incoming job to corresponding data node

13. How does master slave architecture in the Hadoop?
Hadoop Architecture :

It contains 5 software daemons:
1. Name Node
2. Secondary Name Node
3. Data Node
4. Job Tractor
5. Task Tractor
1.Name Node: It is a master node.It contains metadata. i.e it gives the information about the physical locations of the blocks. i.e in contains the blocks information,but not data.
2.Secondary Name Node: when ever the name node goes down the secondary name node comes into picture and does the house keeping activities. i.e the persistent data will be maintain in two files.
1. NamesSpaceImage : FSImage
2. EditLog
3.Data Node: It is a slave node,it stores the data in the form of blocks.
4.Job Tracker: It is a software daemon resides in name node. it is mainly responsible for distributing,monitoring and rescheduling the map reduce tasks to task trackers.
5. Task Tracker:It is also a software daemon,which will does the execution of the map reduce tasks.

14. What is compute and Storage nodes?
I do define Hadoop into 2 ways :
Distributed Processing : Map – Reduce
Distributed Storage : HDFS
Name Node holds Meta info and Data holds exact data and its MR program

15. What are the restriction to the key and value class ?
Key class should implement WritableComparable interface
While a value class should atleast implement Writable interface.

16. Explain how input and output data format of the Hadoop framework?
Fileinputformat, textinputformat, keyvaluetextinputformat, sequencefileinputformat, sequencefileasinputtextformat, wholefileformat, nullvalueOutputFormat, FileOutputFormat and keyValueOutputFormat are file formats in hadoop framewor

17. Which object can be used to get the progress of a particular job ?
JobClient can be useful to get status of running or in queue jobs, and the status of their mapper/reducer phase as well.
Alternately, you could use web UI task tracker to monitor the status of jobs.

18. How can we control particular key should go in a specific reducer?
By using a custom partitioner.

19. How many maps are there in a particular Job?
The number of mappers are usually driven by the distributed file systems block size. For instance if the input file is 2GB and the block size is 64MB, the number of maps will be set to 32 (2000MB / 64MB). The reason why this is agood solution is because each mapper task will get its own block to work on. We followed this approach when we did the WordCount experiments and suggests that each machine should execute between 10 and 100 mapper tasks.
Correct number of reducers must also be considered before executing a job. A function given as 1.75 (number of nodes mapred.tasktracker.reduce.tasks.maximum) will make the faster nodes finish their initial round of reduces first, when the initial round is done they immediately start a new round. Load-balancing will be provided by following this approach . We followed this approach in theWord-Count experiments.

20. What is the use of Combiner?
Combiner acts as a mini reducer on the mapper side. combiner is used for reducing the load for the reducer. Combiner can be written individually or with in the mapper program. combiner will run once if it is within the mapper program otherwise many times can be executed.

21. What is the Reducer used for?
Reducer is used to combine the multiple outputs of mapper to one

22. What are the primary phases of the Reducer?
Reducer has 3 primary phases: shuffle, sort and reduce.

23. Explain the shuffle?
Reducer has 3 primary phases: shuffle, sort and reduce.
Shuffle: Input to the Reducer is the sorted output of the mappers. In this phase the framework fetches the relevant partition of the output of all the mappers, via HTTP.

24. Explain the Reducer?s reduce phase?
Reducer has 3 primary phases: shuffle, sort and reduce.

Reduce: In this phase the reduce(MapOutKeyType, Iterable<MapOutValType>, Context) method is called for each <key, (list of values)> pair in the grouped inputs. The output of the reduce task is typically written to the FileSystem via Context.write(ReduceOutKeyType, ReduceOutValType). Applications can use the Context to report progress, set application-level status messages and update Counters, or just indicate that they are alive. The output of the Reducer is not sorted.

25. How many Reducers should be configured?
The right number of reduces seems to be 0.95 or 1.75 multiplied by (<no. of nodes> * mapreduce.tasktracker.reduce.tasks.maximum). With 0.95 all of the reduces can launch immediately and start transfering map outputs as the maps finish. With 1.75 the faster nodes will finish their first round of reduces and launch a second wave of reduces doing a much better job of load balancing. Increasing the number of reduces increases the framework overhead, but increases load balancing and lowers the cost of failures. The scaling factors above are slightly less than whole numbers to reserve a few reduce slots in the framework for speculative-tasks and failed tasks.

26. It can be possible that a Job has 0 reducers?
Yes. These kind of jobs are called map only jobs. The output from the map job is written directly into the HDFS in the specified output directory as set in the job.

27. Explain the core methods of the Reducer?
public void setup(context)
public void reduce(Key,Value,context)
public void cleabup(context)

28. What is next step after Mapper or MapTask?
After Map Task, sorting and shuffling phase coming to the picture, In this phase sorted and merge the data, once complete the sort and shuffling phase reducer will coming to the picture.
mapper output: (hi,1) (hi,1) (hi,1) (hi,1) (what,2) (what,3) (what,3)
Sort and Shuffle: (hi, {1,1,1,1,}),(what,{2,3,3,})................)
Reducer Input: (hi, {1,1,1,1,}),(what,{2,3,3,})................)

29. How can you add the arbitrary key-value pairs in your mapper?
job.getConfiguration().set(key,value);

30. What is the use of Context object?
context object is an output collector of an old API and the describes the whole job context until it ends.

31. What happens if you don?t override the Mapper methods and keep them as it is?
If we don't override the mapper method the default Identity Mapper will call automatically, we have to declare bellow two things.
import org.apache.hadoop.mapred.lib.IdentityMapper; and conf.setMapperClass(IdentityMapper.class)

32. How Mapper is instantiated in a running job?
first job it will come to job tracker ,then it assigne perticuler ID of job then initilized (key,value) in that way Mapper is instantiated in a running job

33. What alternate way does HDFS provides to recover data in case a Namenode, without backup, fails and cannot be recovered?
ame node is single point of failure. HDFS can do nothing..ther are 2 ways 1.keeping back up files OR maintaining secondary name node on other machine.

34. How will you write a custom partitioner for a Hadoop job?
During Initialization the Hadoop Framework will create an instance of your custom partitioner, when hadoop's reflection utilities instatiate a class,they check to see if that class is configurable. If it is,setconf() is called immedietely after creation. The private configuration will be initialized before the first call to get partition.

35. How did you debug your Hadoop code ?
By using counters and/or loggers.

===

Q1. What are the default configuration files that are used in Hadoop
As of 0.20 release, Hadoop supported the following read-only default configurations
– src/core/core-default.xml
– src/hdfs/hdfs-default.xml
– src/mapred/mapred-default.xml

Q2. How will you make changes to the default configuration files
Hadoop does not recommends changing the default configuration files, instead it recommends making all site specific changes in the following files
– conf/core-site.xml
– conf/hdfs-site.xml
– conf/mapred-site.xml

Unless explicitly turned off, Hadoop by default specifies two resources, loaded in-order from the classpath:
– core-default.xml : Read-only defaults for hadoop.
– core-site.xml: Site-specific configuration for a given hadoop installation.

Hence if same configuration is defined in file core-default.xml and src/core/core-default.xml then the values in file core-default.xml (same is true for other 2 file pairs) is used.

Q3. Consider case scenario where you have set property mapred.output.compress to true to ensure that all output files are compressed for efficient space usage on the cluster. If a cluster user does not want to compress data for a specific job then what will you recommend him to do ?
Ask him to create his own configuration file and specify configuration mapred.output.compress to false and load this file as a resource in his job.

Q4. In the above case scenario, how can ensure that user cannot override the configuration mapred.output.compress to false in any of his jobs
This can be done by setting the property final to true in the core-site.xml file

Q5. What of the following is the only required variable that needs to be set in file conf/hadoop-env.sh for hadoop to work

- HADOOP_LOG_DIR

- JAVA_HOME
– HADOOP_CLASSPATH
The only required variable to set is JAVA_HOME that needs to point to directory

Q6. List all the daemons required to run the Hadoop cluster
– NameNode
– secondary Namenode
– DataNode
– JobTracker
– TaskTracker

Q7. Whats the default port that jobtrackers listens to
50030

Q8. Whats the default port where the dfs namenode web ui will listen on
50070
Posted by Aman at 9:04 AM 0 comments
Email ThisBlogThis!Share to TwitterShare to Facebook
Sunday, November 28, 2010
Java interview questions for Hadoop developer Part 3
Q21. Explain difference of Class Variable and Instance Variable and how are they declared in Java
Class Variable is a variable which is declared with static modifier.
Instance variable is a variable in a class without static modifier.
The main difference between the class variable and Instance variable is, that first time, when class is loaded in to memory, then only memory is allocated for all class variables. That means, class variables do not depend on the Objets of that classes. What ever number of objects are there, only one copy is created at the time of class loding.

Q22. Since an Abstract class in Java cannot be instantiated then how can you use its non static methods
By extending it

Q23. How would you make a copy of an entire Java object with its state?
Have this class implement Cloneable interface and call its method clone().

Q24. Explain Encapsulation,Inheritance and Polymorphism
Encapsulation is a process of binding or wrapping the data and the codes that operates on the data into a single entity. This keeps the data safe from outside interface and misuse. One way to think about encapsulation is as a protective wrapper that prevents code and data from being arbitrarily accessed by other code defined outside the wrapper.
Inheritance is the process by which one object acquires the properties of another object.
The meaning of Polymorphism is something like one name many forms. Polymorphism enables one entity to be used as as general category for different types of actions. The specific action is determined by the exact nature of the situation. The concept of polymorphism can be explained as "one interface, multiple methods".

Q25. Explain garbage collection?
Garbage collection is one of the most important feature of Java.
Garbage collection is also called automatic memory management as JVM automatically removes the unused variables/objects (value is null) from the memory. User program cann't directly free the object from memory, instead it is the job of the garbage collector to automatically free the objects that are no longer referenced by a program. Every class inherits finalize() method from java.lang.Object, the finalize() method is called by garbage collector when it determines no more references to the object exists. In Java, it is good idea to explicitly assign null into a variable when no more in us

Q26. What is similarities/difference between an Abstract class and Interface?
Differences- Interfaces provide a form of multiple inheritance. A class can extend only one other class.
– Interfaces are limited to public methods and constants with no implementation. Abstract classes can have a partial implementation, protected parts, static methods, etc.
– A Class may implement several interfaces. But in case of abstract class, a class may extend only one abstract class.
– Interfaces are slow as it requires extra indirection to find corresponding method in in the actual class. Abstract classes are fast.
Similarities
– Neither Abstract classes or Interface can be instantiated

Q27. What are different ways to make your class multithreaded in Java
There are two ways to create new kinds of threads:
– Define a new class that extends the Thread class
– Define a new class that implements the Runnable interface, and pass an object of that class to a Thread's constructor.

Q28. What do you understand by Synchronization? How do synchronize a method call in Java? How do you synchonize a block of code in java ?
Synchronization is a process of controlling the access of shared resources by the multiple threads in such a manner that only one thread can access one resource at a time. In non synchronized multithreaded application, it is possible for one thread to modify a shared object while another thread is in the process of using or updating the object's value. Synchronization prevents such type of data corruption.
– Synchronizing a method: Put keyword synchronized as part of the method declaration
– Synchronizing a block of code inside a method: Put block of code in synchronized (this) { Some Code }

Q29. What is transient variable?
Transient variable can't be serialize. For example if a variable is declared as transient in a Serializable class and the class is written to an ObjectStream, the value of the variable can't be written to the stream instead when the class is retrieved from the ObjectStreamthe value of the variable becomes null.

Q30. What is Properties class in Java. Which class does it extends?
The Properties class represents a persistent set of properties. The Properties can be saved to a stream or loaded from a stream. Each key and its corresponding value in the property list is a string

Q31. Explain the concept of shallow copy vs deep copy in Java
In case of shallow copy, the cloned object also refers to the same object to which the original object refers as only the object references gets copied and not the referred objects themselves.
In case deep copy, a clone of the class and all all objects referred by that class is made.

Q32. How can you make a shallow copy of an object in Java
Use clone() method inherited by Object class

Q33. How would you make a copy of an entire Java object (deep copy) with its state?
Have this class implement Cloneable interface and call its method clone().

==

Q1. What are the default configuration files that are used in Hadoop
As of 0.20 release, Hadoop supported the following read-only default configurations
– src/core/core-default.xml
– src/hdfs/hdfs-default.xml
– src/mapred/mapred-default.xml

Q2. How will you make changes to the default configuration files
Hadoop does not recommends changing the default configuration files, instead it recommends making all site specific changes in the following files
– conf/core-site.xml
– conf/hdfs-site.xml
– conf/mapred-site.xml

Unless explicitly turned off, Hadoop by default specifies two resources, loaded in-order from the classpath:
– core-default.xml : Read-only defaults for hadoop.
– core-site.xml: Site-specific configuration for a given hadoop installation.

Hence if same configuration is defined in file core-default.xml and src/core/core-default.xml then the values in file core-default.xml (same is true for other 2 file pairs) is used.

Q3. Consider case scenario where you have set property mapred.output.compress to true to ensure that all output files are compressed for efficient space usage on the cluster. If a cluster user does not want to compress data for a specific job then what will you recommend him to do ?
Ask him to create his own configuration file and specify configuration mapred.output.compress to false and load this file as a resource in his job.

Q4. In the above case scenario, how can ensure that user cannot override the configuration mapred.output.compress to false in any of his jobs
This can be done by setting the property final to true in the core-site.xml file

Q5. What of the following is the only required variable that needs to be set in file conf/hadoop-env.sh for hadoop to work

- HADOOP_LOG_DIR

- JAVA_HOME
– HADOOP_CLASSPATH
The only required variable to set is JAVA_HOME that needs to point to <java installation> directory

Q6. List all the daemons required to run the Hadoop cluster
– NameNode
– SecondaryNameNode
– DataNode
– JobTracker
– TaskTracker

Q7. Whats the default port that jobtrackers listens to
50030

Q8. Whats the default port where the dfs namenode web ui will listen on
50070

Q21. Explain difference of Class Variable and Instance Variable and how are they declared in Java
Class Variable is a variable which is declared with static modifier.
Instance variable is a variable in a class without static modifier.
The main difference between the class variable and Instance variable is, that first time, when class is loaded in to memory, then only memory is allocated for all class variables. That means, class variables do not depend on the Objets of that classes. What ever number of objects are there, only one copy is created at the time of class loding.

Q22. Since an Abstract class in Java cannot be instantiated then how can you use its non static methods
By extending it

Q23. How would you make a copy of an entire Java object with its state?
Have this class implement Cloneable interface and call its method clone().

Q24. Explain Encapsulation,Inheritance and Polymorphism
Encapsulation is a process of binding or wrapping the data and the codes that operates on the data into a single entity. This keeps the data safe from outside interface and misuse. One way to think about encapsulation is as a protective wrapper that prevents code and data from being arbitrarily accessed by other code defined outside the wrapper.
Inheritance is the process by which one object acquires the properties of another object.
The meaning of Polymorphism is something like one name many forms. Polymorphism enables one entity to be used as as general category for different types of actions. The specific action is determined by the exact nature of the situation. The concept of polymorphism can be explained as "one interface, multiple methods".

Q25. Explain garbage collection?
Garbage collection is one of the most important feature of Java.
Garbage collection is also called automatic memory management as JVM automatically removes the unused variables/objects (value is null) from the memory. User program cann't directly free the object from memory, instead it is the job of the garbage collector to automatically free the objects that are no longer referenced by a program. Every class inherits finalize() method from java.lang.Object, the finalize() method is called by garbage collector when it determines no more references to the object exists. In Java, it is good idea to explicitly assign null into a variable when no more in us

Q26. What is similarities/difference between an Abstract class and Interface?
Differences- Interfaces provide a form of multiple inheritance. A class can extend only one other class.
– Interfaces are limited to public methods and constants with no implementation. Abstract classes can have a partial implementation, protected parts, static methods, etc.
– A Class may implement several interfaces. But in case of abstract class, a class may extend only one abstract class.
– Interfaces are slow as it requires extra indirection to find corresponding method in in the actual class. Abstract classes are fast.
Similarities
– Neither Abstract classes or Interface can be instantiated

Q27. What are different ways to make your class multithreaded in Java
There are two ways to create new kinds of threads:
– Define a new class that extends the Thread class
– Define a new class that implements the Runnable interface, and pass an object of that class to a Thread's constructor.

Q28. What do you understand by Synchronization? How do synchronize a method call in Java? How do you synchonize a block of code in java ?
Synchronization is a process of controlling the access of shared resources by the multiple threads in such a manner that only one thread can access one resource at a time. In non synchronized multithreaded application, it is possible for one thread to modify a shared object while another thread is in the process of using or updating the object's value. Synchronization prevents such type of data corruption.
– Synchronizing a method: Put keyword synchronized as part of the method declaration
– Synchronizing a block of code inside a method: Put block of code in synchronized (this) { Some Code }

Q29. What is transient variable?
Transient variable can't be serialize. For example if a variable is declared as transient in a Serializable class and the class is written to an ObjectStream, the value of the variable can't be written to the stream instead when the class is retrieved from the ObjectStreamthe value of the variable becomes null.

Q30. What is Properties class in Java. Which class does it extends?
The Properties class represents a persistent set of properties. The Properties can be saved to a stream or loaded from a stream. Each key and its corresponding value in the property list is a string

Q31. Explain the concept of shallow copy vs deep copy in Java
In case of shallow copy, the cloned object also refers to the same object to which the original object refers as only the object references gets copied and not the referred objects themselves.
In case deep copy, a clone of the class and all all objects referred by that class is made.

Q32. How can you make a shallow copy of an object in Java
Use clone() method inherited by Object class

Q33. How would you make a copy of an entire Java object (deep copy) with its state?
Have this class implement Cloneable interface and call its method clone().