FREEMIND

Date

Page

# INTRODUCTION TO BIG DATA

## ASSIGNMENT SHEET - I

Submitted By : Vishal Singh

B.Tech C.S.E, 3rd yr.

Sec. F

Q.1 Imagine that u are uploading a file of 500mb into HDFS. 100mb of data is successfully uploaded into HDFS and another client wants to read the uploaded data while the upload is still in progress. What will happen in such a scenario, will the 100 mb of data that is uploaded will be displayed?

- The default block size of Hadoop-1 is 64MB and of Hadoop-2 is 128MB.

Let the block size be 100 MB, i.e., 5-blocks are to be replicated 3 times (the default replication factor).

If we have A, B, C, D & E be the five blocks for client, file, name-node and data-node. Then firstly, the client takes the Block A and approaches the name-node for the data-node location in order to store this current block and replicated copies of it.

Once the data-node information is available to the client, he will reach directly to the data-node and starts the copying of Block-A, which will at the same time gets replicated to second data-node. When the block gets copied and replicated to data-node, the confirmation of Block A storage will get to the client, then further, the client will re-start the same procedure for the next block, B.

Thus, if one is uploading a file of 500MB into the HDFS where 100MB of data has been successfully uploaded in the HDFS and the other client is about to read the uploaded data while the upload is still to be in progress then only the present block which is being written will not be visible to the readers.

Q.2 Map Reduce is a batch query processor. Explain this statement w.r.t processing time and data usage.

Ans. Batch processing in Hadoop is considered as processing huge amount of data. It is processed on a group of transactions that are collected over a period of time.

Hadoop works on Batch processing where -

- first, data is collected, entered and then processed.
- Later, it produces batch results.

Date
Page

MapReduce is a programming model that allows the user to write batch processing jobs with a small amount of code. It is flexible in the sense that you, the user, can write code to modify the behavior, but making complex data processing pipelines becomes cumbersome bcoz every MapReduce job has to be managed and scheduled on its own. The intermediate output of map tasks is written to a file which allows the framework to recover easily if a node has a failure. The stability comes at a cost of performance, as the data could have been forwarded to reduce tasks with a small buffer instead, creating a stream.
The flexibility of MapReduce allows the user to do more complex data processing such as table joins, page ranks, sorting and anything you can code inside the limitations of the framework.

Q.3 When decommissioning the nodes in a Hadoop Cluster, why should you stop all the task trackers?

One should be very well aware of the complete procedure of decommissioning the data-node in the Hadoop cluster, but it is to be taken care of when the task trackers runs the MapReduce jobs on the data-node which is decommissioned.

Unlike data-node, there is not any graceful way of decommissioning the task-tracker, where assumption is made as whenever the present task is to be moved to the another node then one should rely on task making process in order to stop from failure, and further it will be rescheduled on the cluster. Possibility is that when the final attempt of task runs on the task-tracker then the final failure will result on the entire failing of the job. So, the decommissioning stops the data-node, but in order to move the present task to the other node, one should manually stop the task tracker which is running on the decommissioning node.

Q.4 Define Speculative Execution in Hadoop?

In Hadoop, MapReduce breaks jobs into tasks and these tasks run parallel rather than sequential, thus reduces overall execution time. This model of execution is sensitive to slow tasks (even if they are few in numbers) as they slow down the overall execution of a job.
There may be various reasons for the slowdown of tasks, including hardware degradation or software misconfiguration, but it may be difficult to detect causes since the tasks still complete successfully, although more time is taken than the expected time. Hadoop doesn't try to diagnose and fix slow running tasks, instead, it tries

FREEMIND

Date

Page

to detect them and runs backup tasks for them. This is called speculative execution in Hadoop. These backup tasks are called Speculative tasks in Hadoop.

Q.5 What is LazyOutputFormat in Hadoop?

File OutputFormat subclasses will create output files (part-r-nnnn), even if they are empty. Some applications prefer not to create empty files, which is where LazyOutputFormat helps.

LazyOutputFormat is a wrapper OutputFormat. It makes sure that the output file should create only when it emit its first record for a given partition.
To use LazyOutputFormat, call its SetOutputFormatClass method with the JobConf.
To enable LazyOutputFormat, streaming ~~cmd~~ and pipes supports a - lazyOutput option.

Q.6 List main configuration parameters in a MapReduce program?

The main configuration parameters are:
- Input format of data.
- Job's input locations in the distributed file system.
- Output format of data.
- Job's output locations in the distributed file system.
- JAR file containing the mapper, reducer & driver class.
- Class containing the map function.
- Class containing the reduce function.

Date
Page

Q.7 Assume you have a cluster of machines with disks mounted to them. All files throughout the cluster have a unique path. The setup is sufficient to run Hadoop on the cluster, HDFS does not need to be installed. Discuss the advantages or/and disadvantages of running Hadoop without HDFS in such a setting?

- As we know, HDFS has distributed storage, parallel computation and Horizontal Scalability. It is a core component of Hadoop, designed to run on a cluster of computers and servers at cloud-based utility services.

- Data is distributed and processed over the cluster in MapReduce which increases the time and reduce processing speed.

- The overall performance will be slow since Hadoop does not process streamed data, it supports batch processing only.

- Hadoop is not so efficient for iterative processing, as Hadoop does not support cyclic data flow.

- Without HDFS, Hadoop is not easy to use and lacks many features while processing data. So, its better to install HDFS always while working with Hadoop.

Date
Page

Q.8 If 8TB is the available disk space per node (10 disks with 1TB, 2 disks for O.S etc. were excluded). Assuming initial data size is 600TB. How will you estimate the no. of data nodes (n)?

The actual data size to be stored is 600TB.
Default - 3x replica factor for the Hadoop.

For the overhead of the hardware machine, 2 disks were considered.

=> Given Replication factor = 3
Data size = 600TB
Intermediate data = 1
Requirements of total storage = 3+1 X 600 = 2400TB
Available disk-size = 8 TB

Total data-nodes required = 2400/8 = 300 machines

Now, Disk-space utilisation = 65%
Compression ratio = 2:3

Total requirement of storage:

2400 2.3 = 1043.5 TB

Available size of disk:

8 x 0.65 = 5.2 TB

total data-nodes req.: 1043.55.2 = 201 machines

Actual usable size of cluster (100%):

$201 \times 8 \times 2.34 = 925$ TB

* It has been predicted that there is 20% of the increase of data in quarter and we all need to predict is the new machines which is added in particular year.

Increase of data : 20% quaterly

Additional data:

I$^{st}$ quarter : $1043.5 \times 0.2 = 208.7$ TB

II$^{nd}$ quarter : $1043.5 \times 1.2 \times 0.2 = 250.44$ TB

III$^{rd}$ quarter : $1043.5 \times 1.2 \times 1.2 \times 0.2 = 300.5$ TB

IV$^{th}$ quarter : $1043.5 \times 1.2 \times 1.2 \times 1.2 \times 0.2 = 360.6$ TB

Additional data nodes:

I$^{st}$ quarter : $208.75/2 = 41$ machines

II$^{nd}$ " : $250.445/2 = 49$ machines

III$^{rd}$ " : $300.55/2 = 58$ machines

IV$^{th}$ " : $360.65/2 = 70$ machines