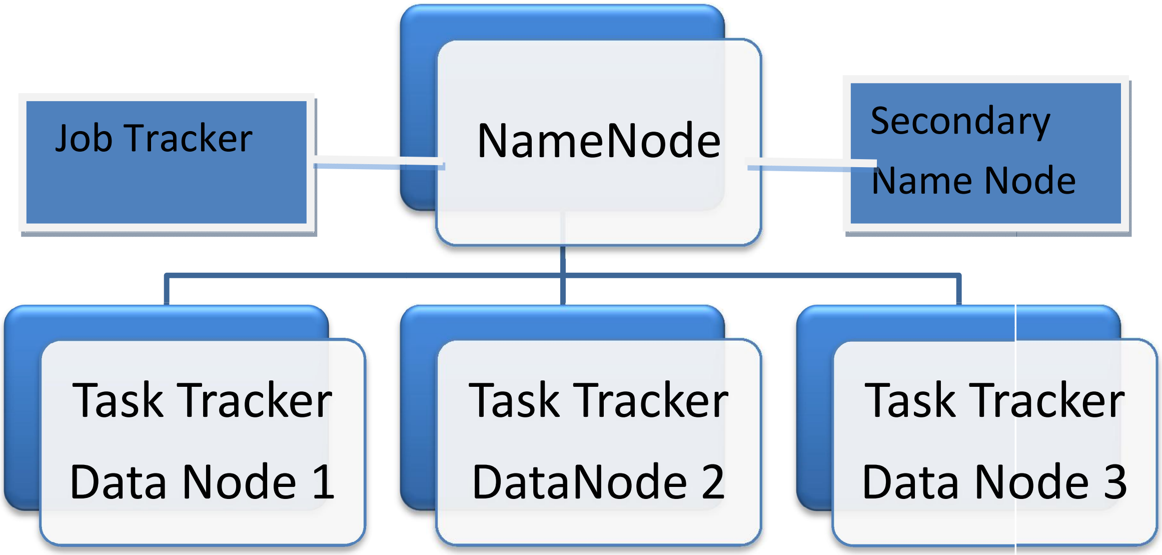
Hadoop Architecture



Name-Node: It controls complete Storage operations.
When you store a file,
1. Name Node will contact operating system and fetch file size.
2. From configuration file it takes information about blocksize and no of replication factor for each block.
3. Based on above information it splits the data file into blocks including replications.
Example: f1 is unix file and size is 100 mb, Number of replications configured is 3. Block size is 64 mb.
The required file size in hadoop is 64mb x 2 (blocks) x3 (replications).
4. Storage decision making for these blocks. it's done based on priority sequence.

- High Configured systems
- Nearest location
- Which had high available space?

5. After storing blocks including replications, the file block-meta data will be registered on Name-Node.
6. Other information what Name Node will maintain is Configuration of Nodes, physical address of the nodes, edit log (restoring purpose).
7. It serves all its information, when other daemon requested.

Data-Node: As per order of Name-Node the Data-Node will store file block in the local disc and sends acknowledgement to Name-Node.

Job Tracker: Job Tracker is responsible for controlling the job executions.

- Work allotment to the nodes
- Fault tolerance
- Automatic load balancing
- Health monitoring

Steps 1. When you submit a job, request is send to Job Tracker. It verifies file existence and takes the file block information such as how many blocks the file had and physical location for the blocks etc…(By requesting meta-data of the Name-Node).

2. Divide the job into small tasks (Example: each block is one smaller task).
3. Work assignment of this task to the Task Tracker of the Data-Nodes.
4. Giving orders to Task Trackers.
5. Taking care without fault tolerance.
6. after completion of Job Execution, acknowledgement to Name-Node.

Decision Process (Priority Sequence)

- high Configured System
- Nearest location.
- which is not busy
- If all the nodes are free then check health throughput ratio
- If all are busy, work allotment to non local system/ when current block is not available)

Task Tracker: As per order of Job Tracker, it accepts the work and process the current block. It acknowledges to Job-Tracker. It frequently sends heart-beat signals to Job-Tracker.

Seconday Name Node:

Name Node is single point of failure. If Name-Node is down all running jobs will be terminated and queued jobs cannot be submitted. New jobs can't be submitted. Storage services will be discarded ie total cluster is down.

The purpose of 2- Name-Node is at least running jobs to be continued even though Name-Node got down, because secondary Name-Node maintains meta-data of the files on which current jobs are running. If Name-Node down, the Meta-data information is served to JT by Secondary Name Node. After completion of all running jobs execution, Secondary Name-Node will be shutdown.

Note: for a Cluster Secondary Name-Node is optional.

How Job-Tracker monitor health of the System(Nodes):

Task Tracker sends heart-beat signals for each 10 seconds(Hear-Beat Interval and itcan't be configured).

Job Tracker health decision interval is 10 mins by default (can be configured) in this,

A Task Tracker is supposed to send 60 (10 x 6) beats to Job-Tracker.

- If Hear-Beat Signal is 60 (perfect health)
- If Hear-Beat Signal is <60 (but health)
- If Hear-Beat Signal is 0(node down)

What Admin has to do, if Name Node is down!

Replace the new machine for Name-Node.

Restore all the data into new machine.

Restart the edit log (if not restart, new node not recognizing the existing node)

Hadoop Streaming:

It is an internal sub system which converts non-java Map Reduce code into java MapReduce code (job).

Python Map Reduce→Java MapReduce

Ruby Map Reduce→Java MapReduce

C++ Map Reduce→Java MapReduce

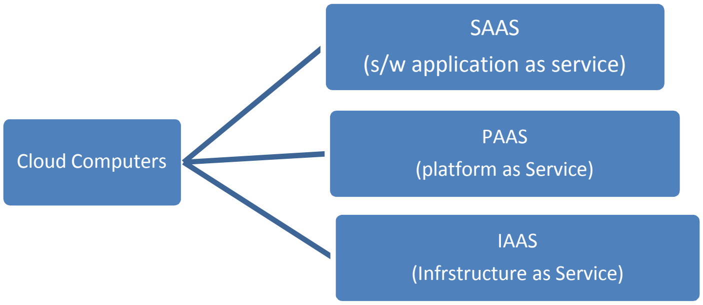
Hive Map Reduce→Java MapReduce

SQOOP Map Reduce→Java MapReduce

Hadoop Piping: to process biggest rows as per MR/hive/other eco systems. The max length of read is 100 Mb. If your record length is more than that using hadoop piping will be splited into smaller pieces and processed and merged while writing.

Example: 110 Mb→100Mb+10 Mb

Hadoop on Cloud:



Amazon AWS-→Hadoop

HDFS→SSS→S3(Simple storage service)

EMP→Elastic Map Reduce

EC2→(Elastic Computers)

Hadoop Software Distributions:

- Apache foundation
- Clodera Distribution for hadoop
- Map Reduce Distribution
- Hortonworks Distribution Platform

In real time we have pseudo distributed mode. In pseudo mode hadoop daemon will run on single JVM, In single mode each daemon will run separately.