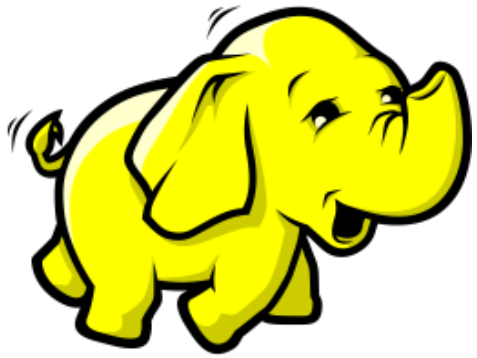
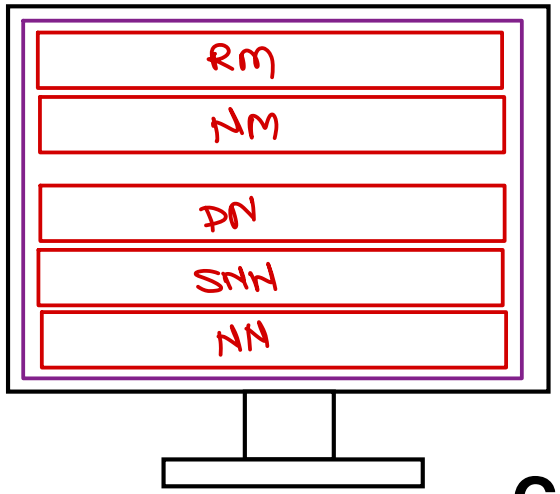


Big Data – Hadoop

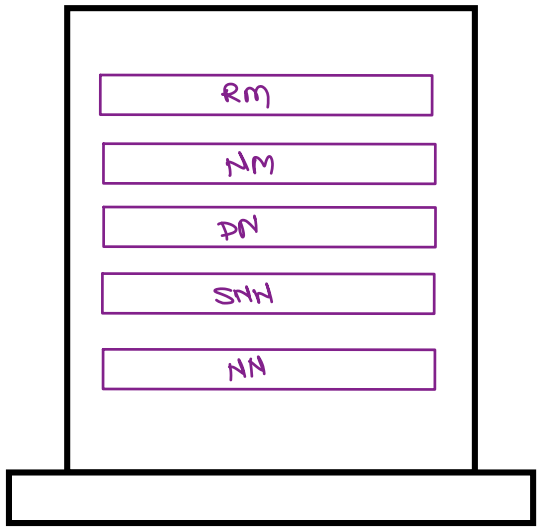
Trainer: Mr. Nilesh Ghule.

Hadoop installation modes & Configuration files

- Local mode



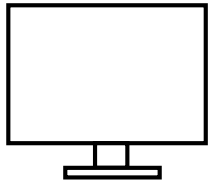
- Pseudo distribution mode

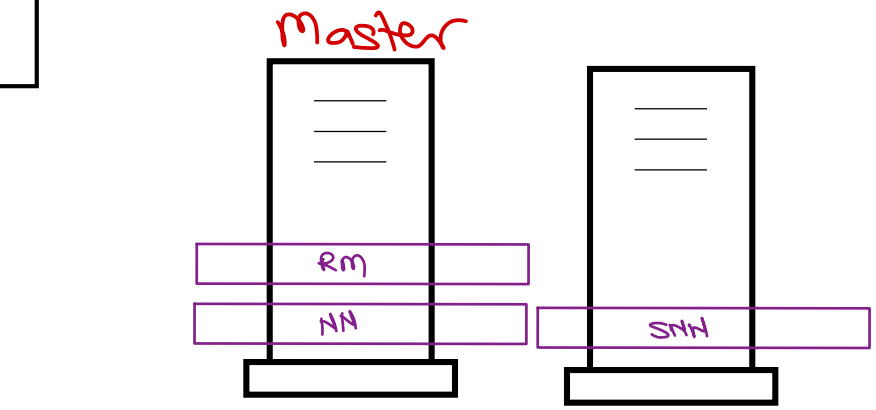


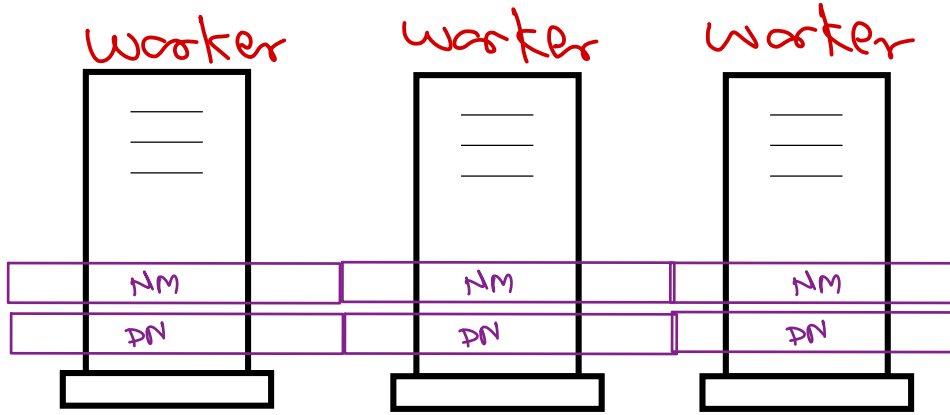
Config files

- hadoop-env.sh
- core-site.xml
- hdfs-site.xml
- mapred-site.xml
- yarn-site.xml
- ~/.bashrc

- Full distribution mode
  https://github.com/nilesh-g/hadoop-cluster-install







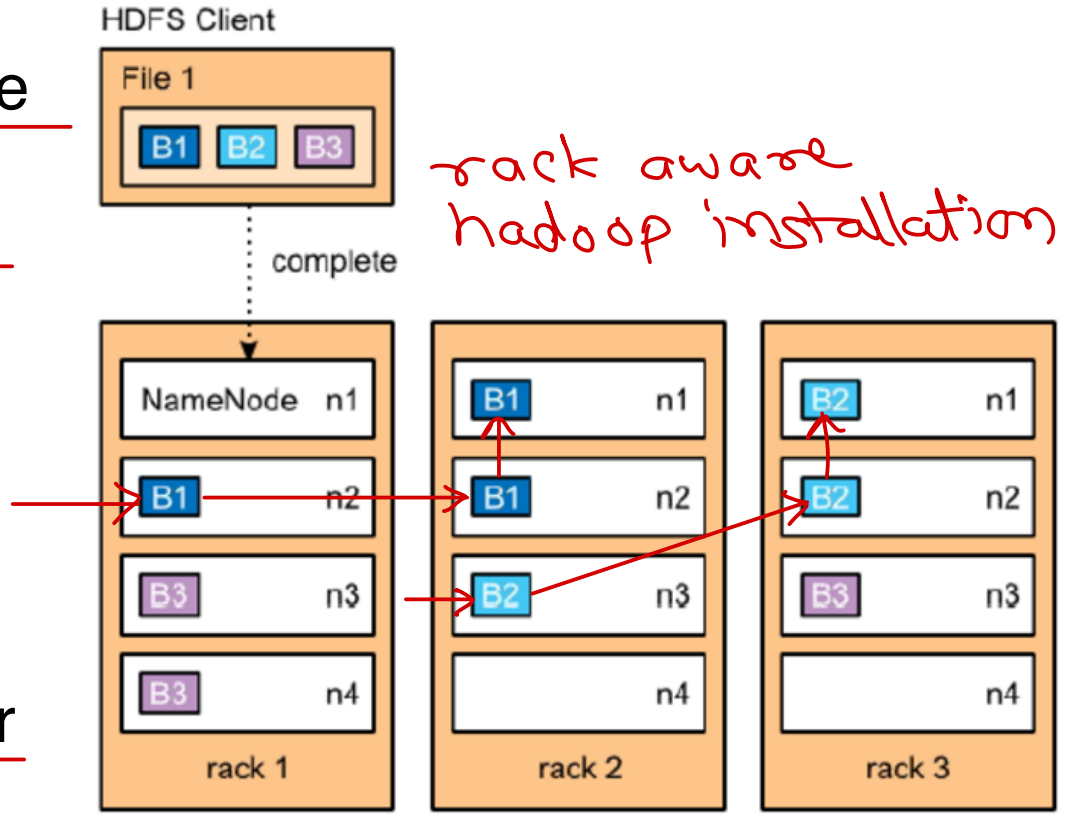
Using HDFS

- Before using, HDFS need to be formatted. It create first (empty) file system image on NameNode.
  - terminal> hdfs namenode –format
- Start all HDFS daemons & verify them
  - terminal> start-dfs.sh
  - terminal> jps
  - browser: http://localhost:50070
- While metadata is loaded into NameNode memory, HDFS is not ready for use. This state is safe mode.
- HDFS user commands
  - terminal> hadoop fs –help
  - syntax: hadoop fs genericoptions command
- Generic options
  - -conf, -fs, …
- HDFS user commands categories
  - ingestion/retrieval: put, get, getmerge
  - directory handling: ls, mkdir, rmdir
  - file data handling: cat, tail, rm, truncate, touchz, stat
  - metadata handling: chmod, chown, setrep
- HDFS admin commands
  - terminal> hdfs –help
  - terminal> hdfs dfsadmin –help

HDFS Replication

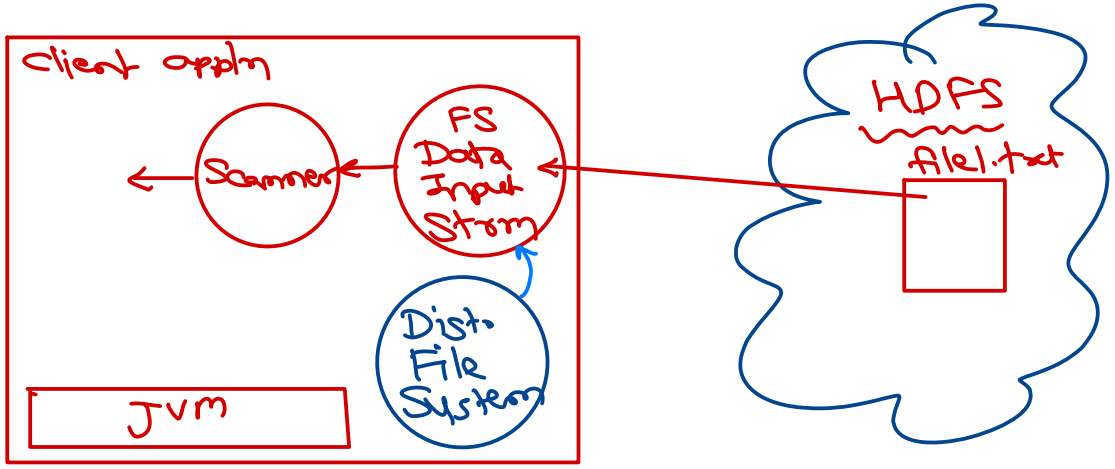
- Default replication factor for HDFS is 3.
  - hdfs-site.xml – hdfs-site.xml
- Each data block is copied on 3 different data nodes.
- Data nodes are stored across the racks for more reliability. Data nodes are chosen by name node considering load balancing.
- NameNode ensure availability of datanodes by the periodic heartbeat signal. (3 sec)
- If number of replicas are less than replication factor, it is under-replica. If number of replicas are more than replication factor, it is over-replica.
- Hadoop auto adjust replicas to the replication factor over the time by creating more replicas or deleting them depending on scenario.

- Replication is done while write operation. (pwr)
- If no replica is available while read operation, it fails.



HDFS Java API

- HDFS can be accessed or manipulated using Java API.
- DistributedFileSystem class represent HDFS, while LocalFileSystem class represent local file system.
- Mainly two types of APIs
  - FileSystem API
  - File-IO API
- FileSystem API
  - Deals with metadata & directories.
  - FileStatus object contains metadata of file or directory.
  - Most of FileSystem APIs don't need access to DataNode (as metadata is maintained on NameNode itself).
- File IO API
  - Deals with data of the files.
  - FSDataInputStream class for reading the file, while FSDataOututStream class for writing the files.
  - They provide abstraction like replication process, network access, etc.
- Write/Read text files



Thank you!

Nilesh Ghule <nilesh@sunbeaminfo.com>