HDFS BASICS

HDFS : Hadoop distributed file system (BASIC HDFS)

Pre-requisites

- Basic understanding of what file system means
- Practical working knowledge of UNIX file system basics

Agenda

Architecture of HDFS

How does HDFS store the file internally

Failure handling and recovery mechanism

Rack awareness

Role of name node and secondary name node

When to use HDFS and when not to use it

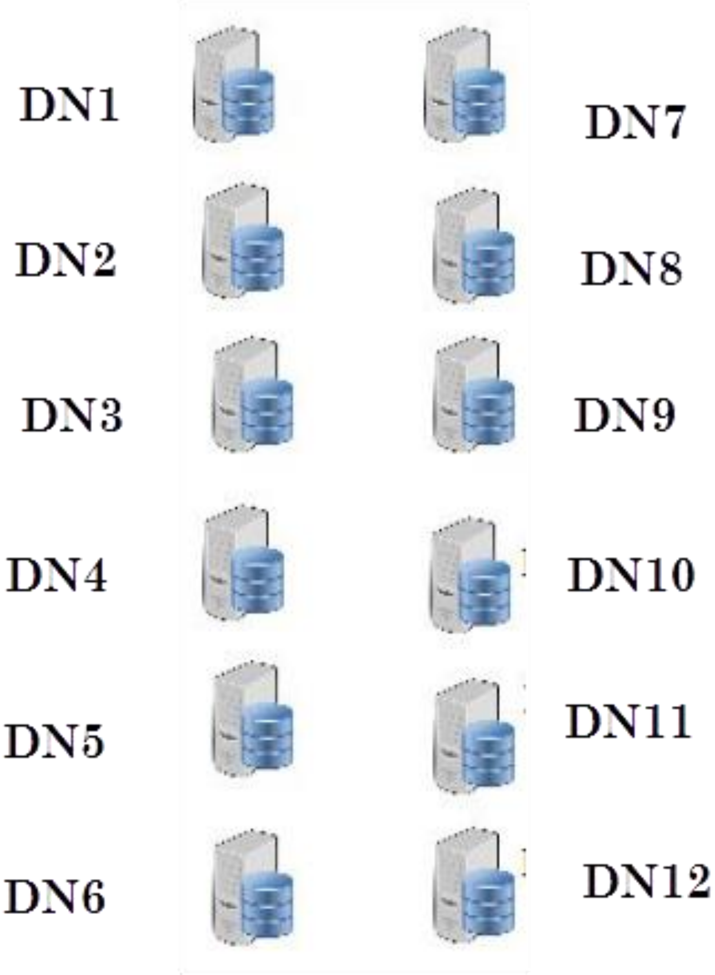
HDFS – The Storage Layer in Hadoop

Input file of 200 MB (“Newfile.txt”)

File format : Text file
Each line has some integers separated by space
100 200 237 65 67 0 9 56
200001 342342 9809 08734 …..

Client machine (not a part of cluster)

Name node (is a part of Hadoop cluster)



Splitting of file into blocks in HDFS

Default split size is **64MB(It can be changed)**

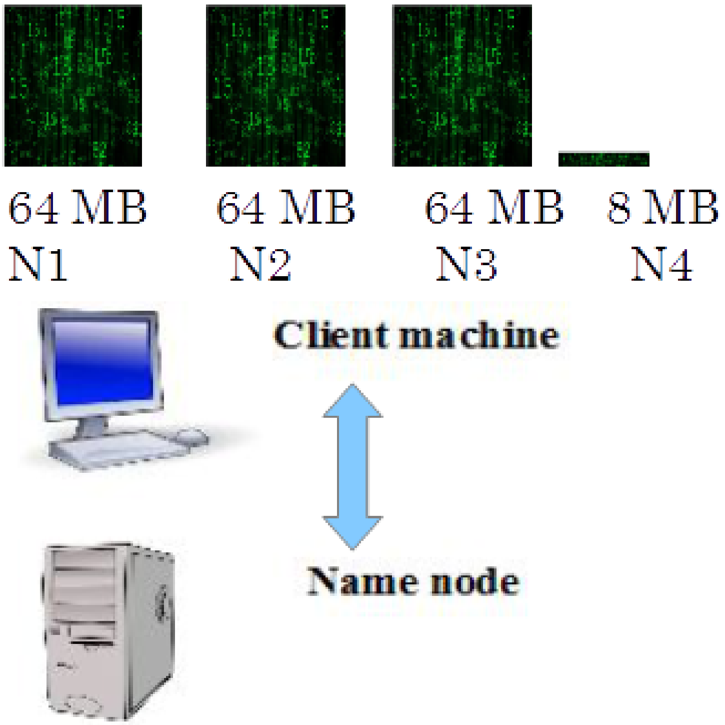
Original File size is 200MB. 200Mb file is

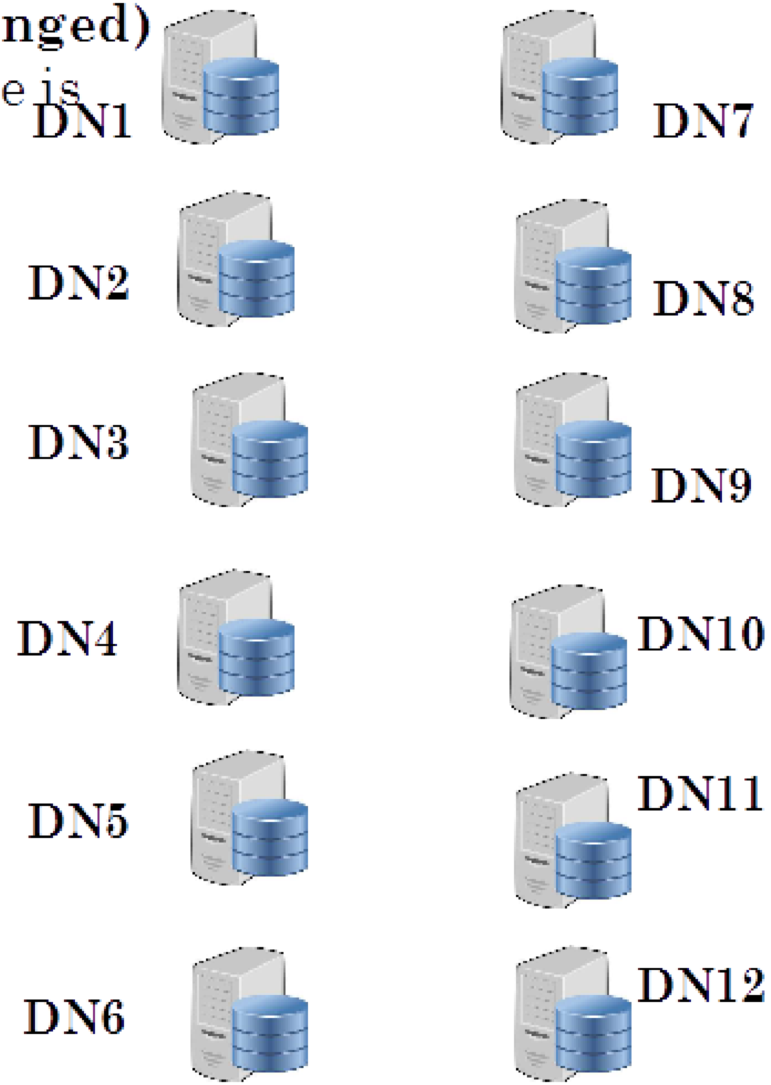
Split into 4 blocks of N1, N2, N3 and N4

The block N4 is just 8 MB (200-64*3)

Each block is now a separate FILE &

N1, N2, N3 and N4 are file names.



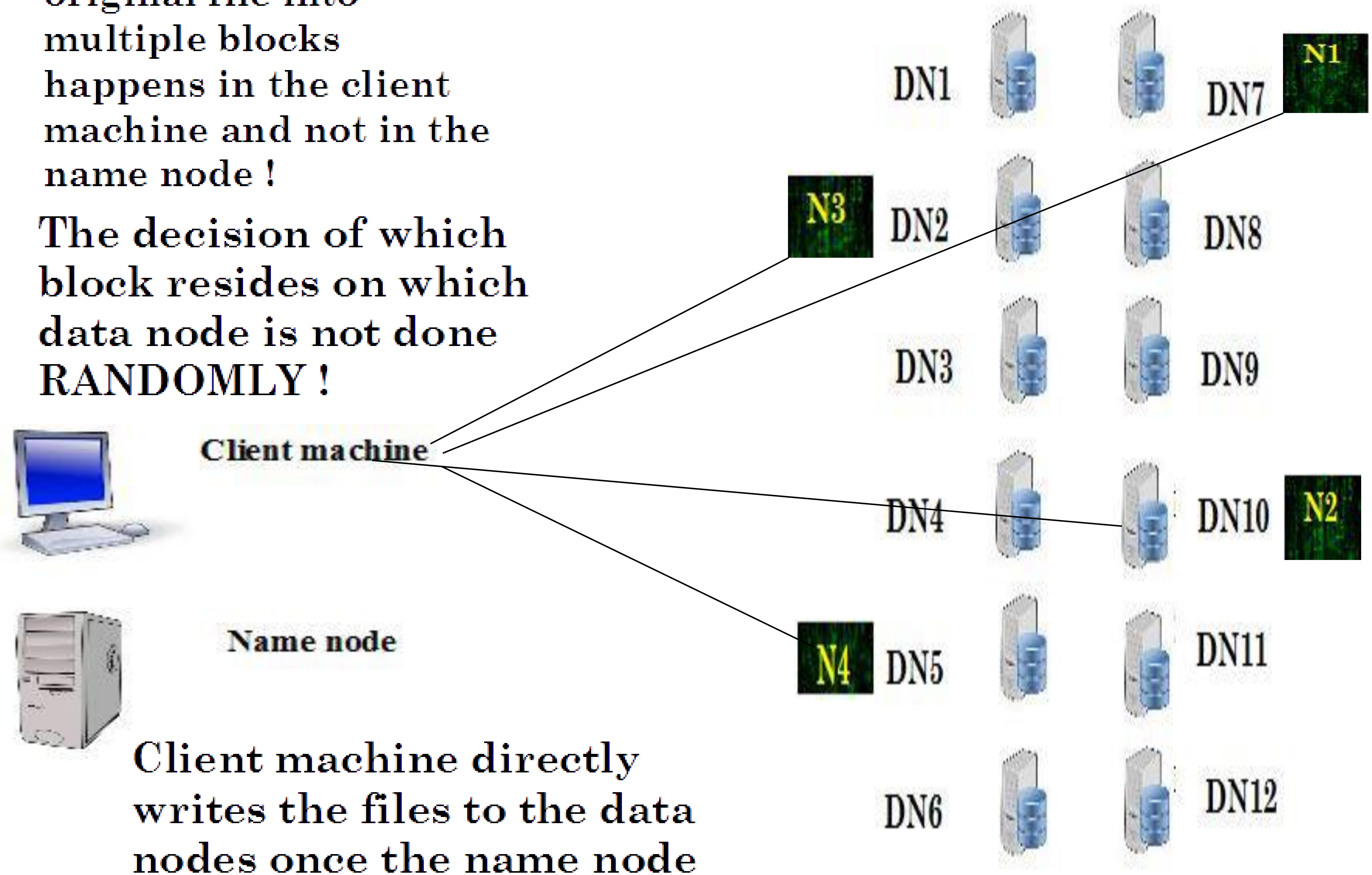


File Storage in HDFS

greatlearning
Learning for Life

Breaking up of the original file into multiple blocks happens in the client machine and not in the name node !

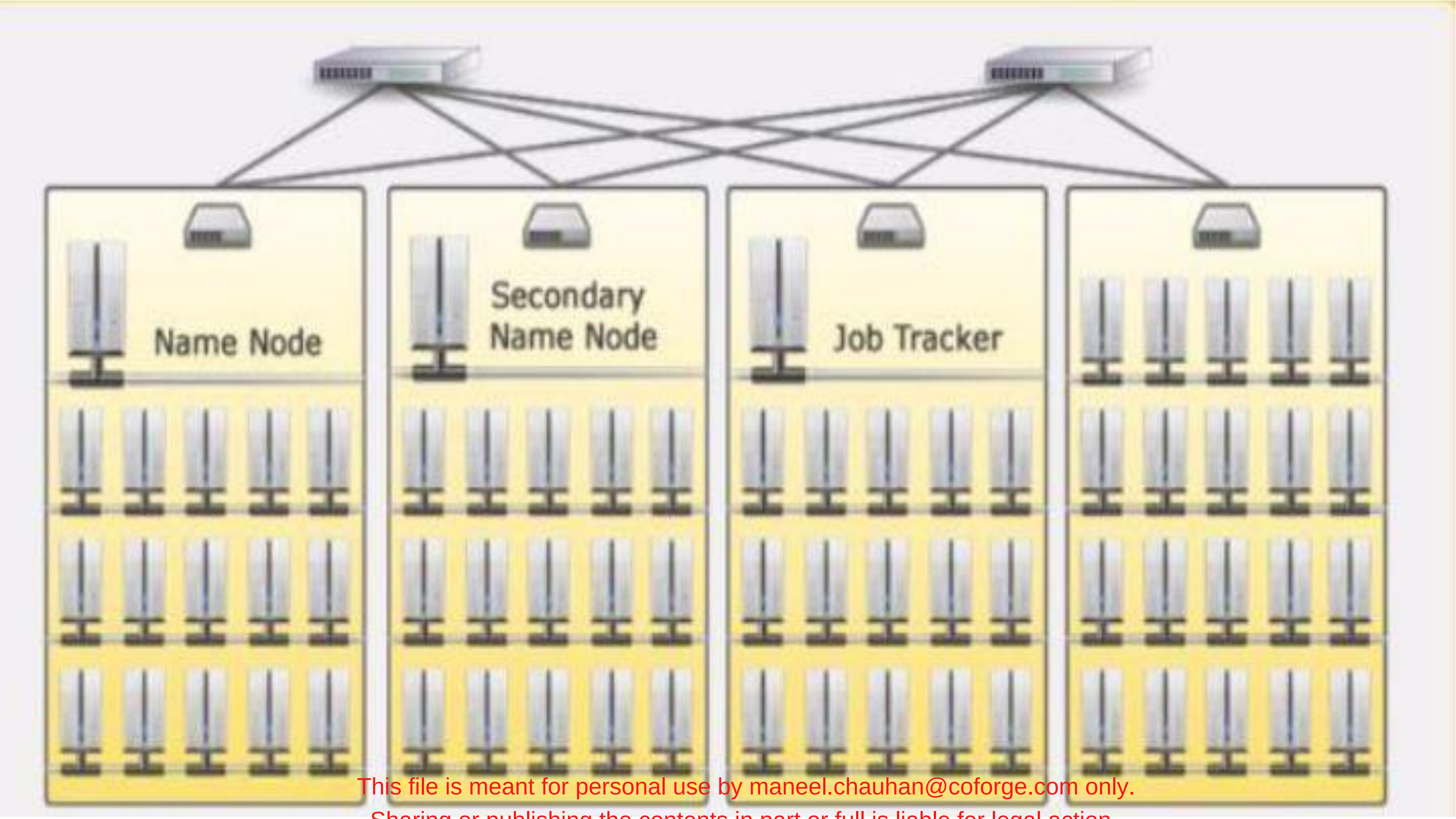
The decision of which block resides on which data node is not done RANDOMLY !



Client machine directly writes the files to the data nodes once the name node provides the details about data nodes.

Hadoop cluster deployed in a production environment into multiple racks

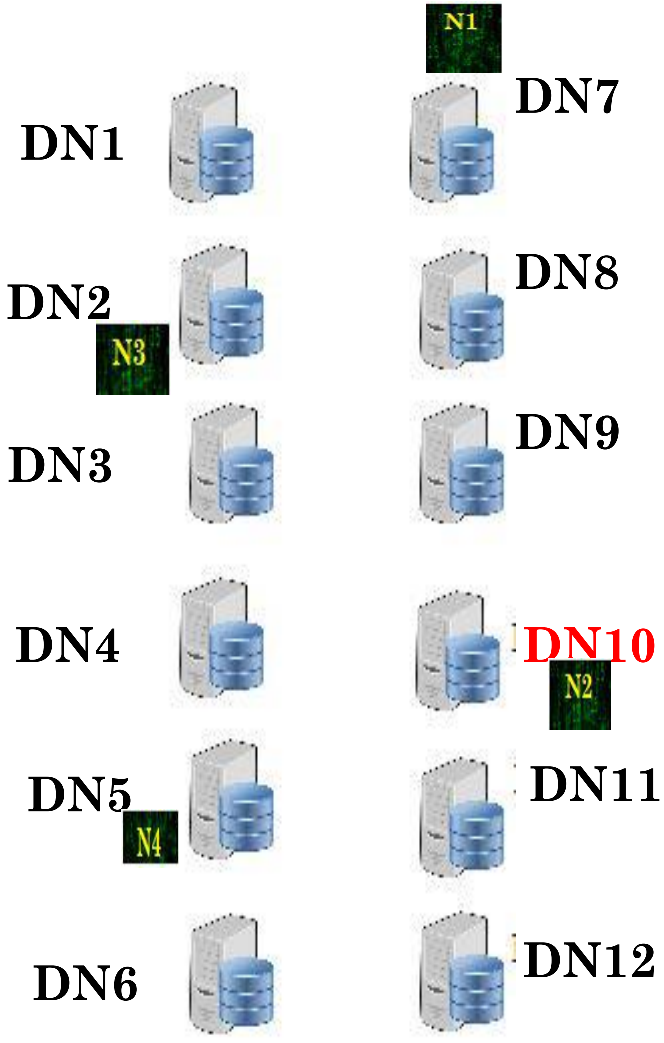
Multilayer switches/routers interconnect the switches on each rack



Failure of a data node

Q)What happens in the event of a data node Failure ? (eg : DN 10 fails)
A)Data saved on that node will be lost

To avoid loss of data, copies of the Data blocks on data node is stored on multiple data nodes. This is called data replication.



Client machine

Name node

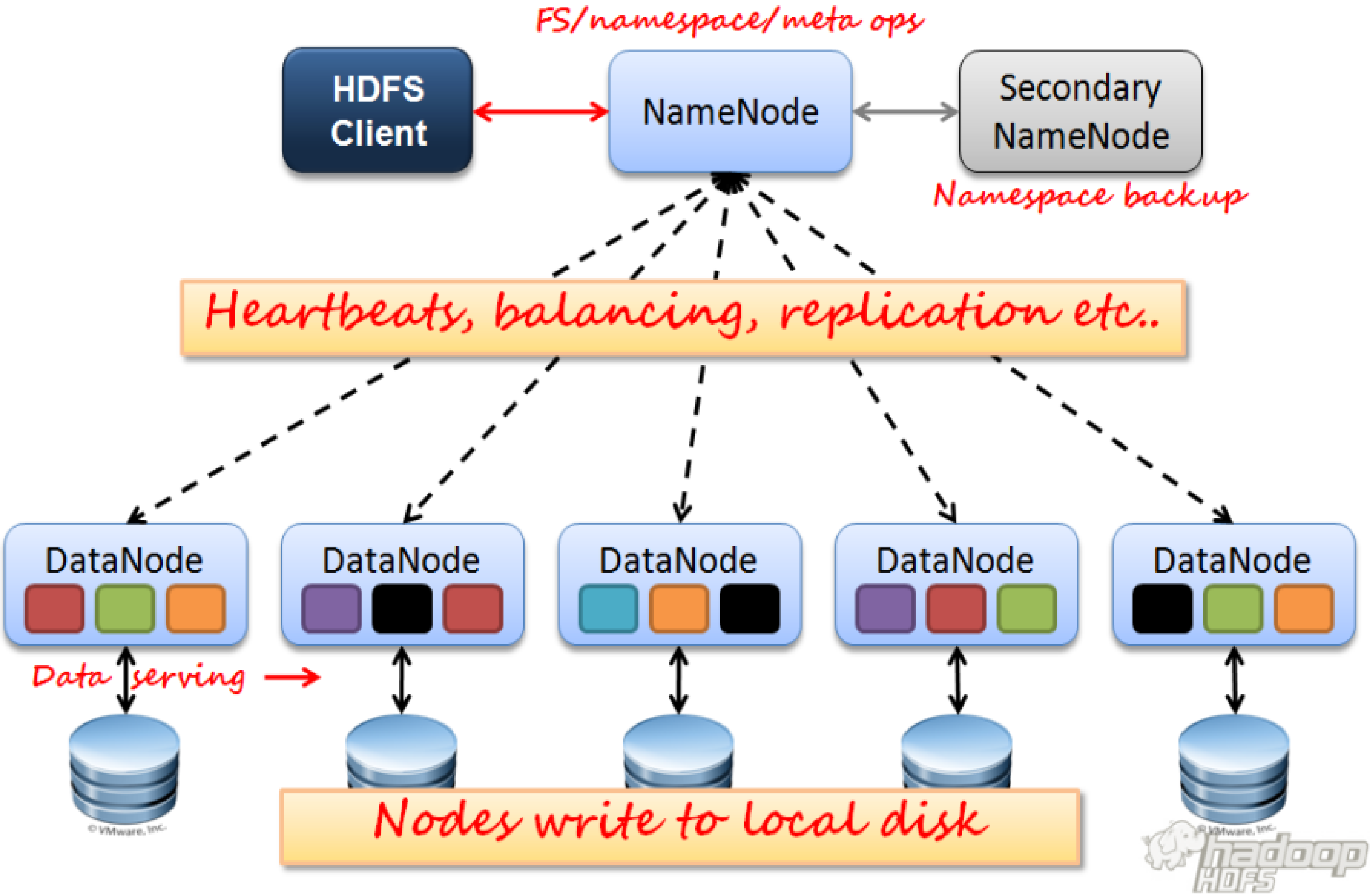
Replication of data blocks

- **How many copies of each block to save?**

  Its decided by REPLICATION FACTOR (by default its **3**, i.e. every block of data on each data node is saved on 2 more machines so that there is total 3 copies of the same data block on different machines)

  This replication factor can be set on per file basis while the file is being written to HDFS for the first time.

Design and Architecture Overview

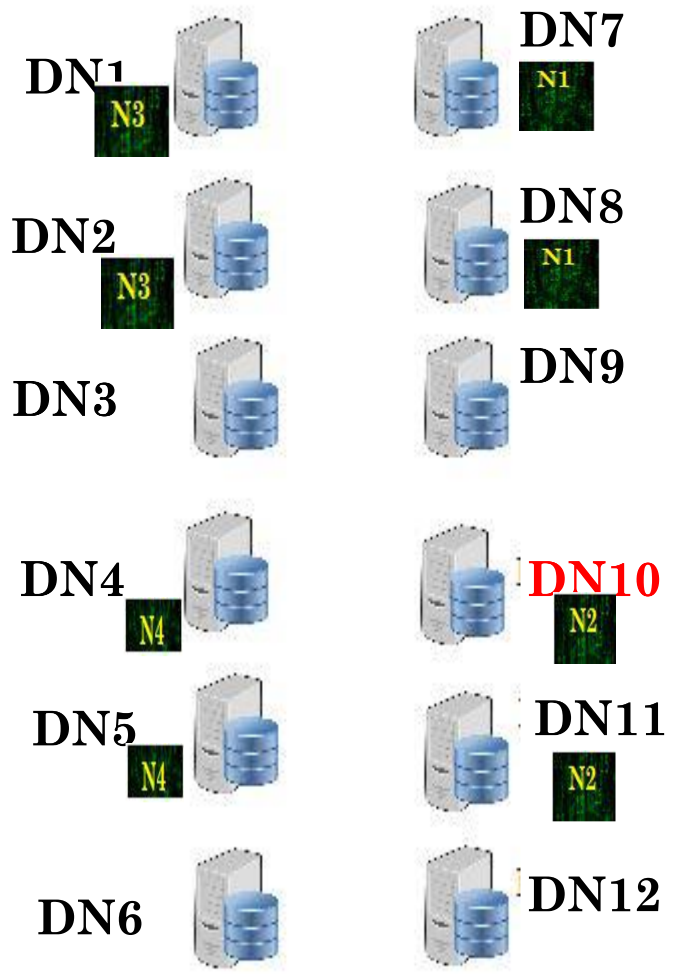


Data replication on failure and failed node recovers

Replication factor =2

DN10 fails

Replication factor for block N2 is now 1 !

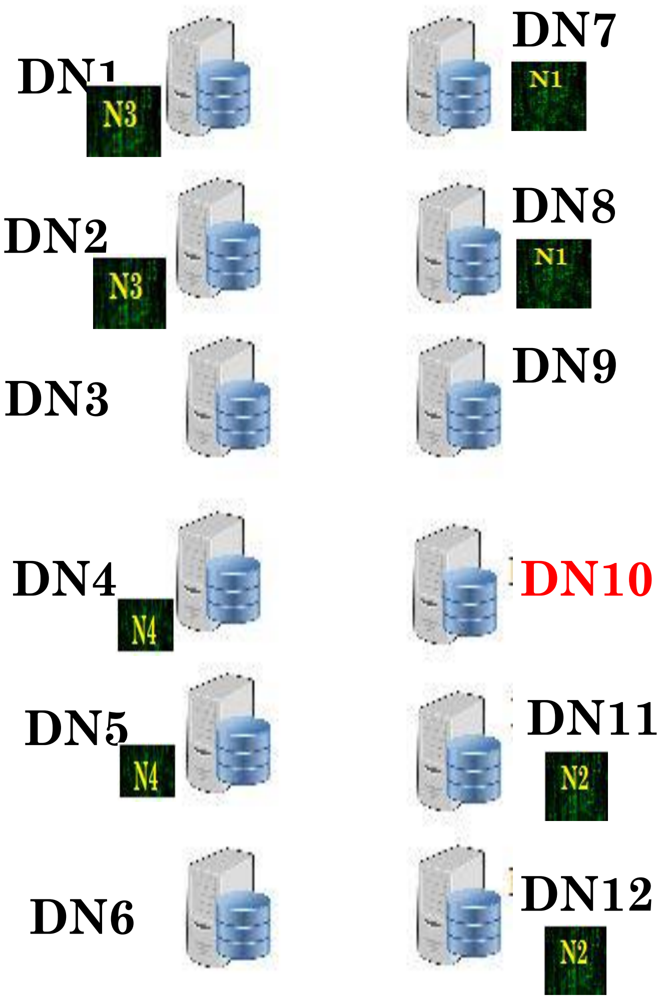


Data replication on failure and failed node recovers Pt2

Replication factor =2

DN10 fails

The data on the failed node would be copied to some other node in the cluster automatically. In this case its copied to DN12 from its nearest neighbour DN11



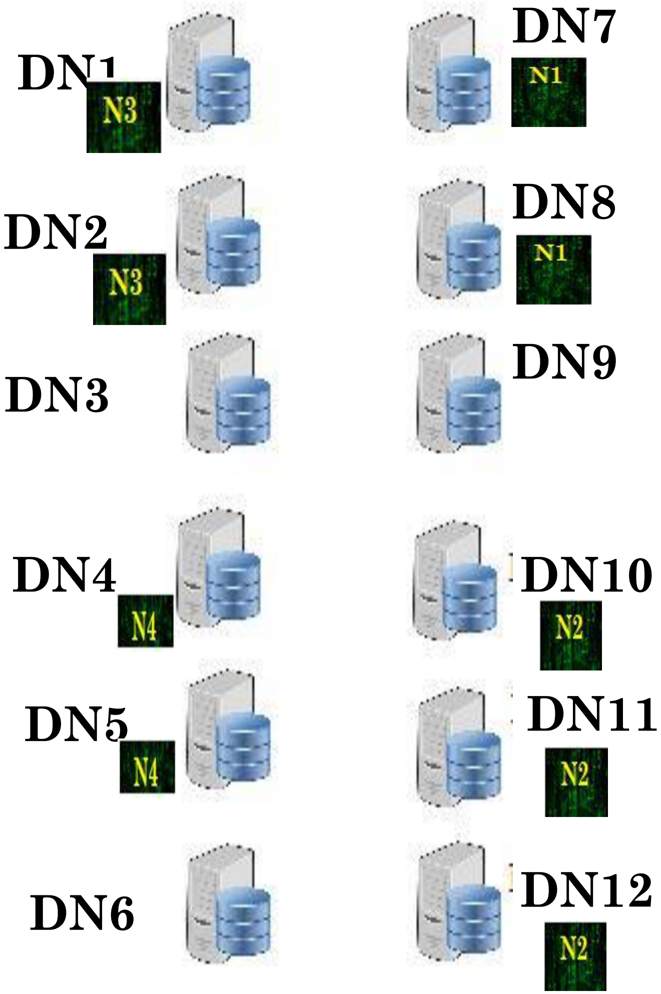
Client machine
Name node

DN10 is up again after some time …

Replication factor = 3 for block N2

HDFS will delete one extra copy of N2 from any of the 3 nodes (DN10, DN11 or DN12)

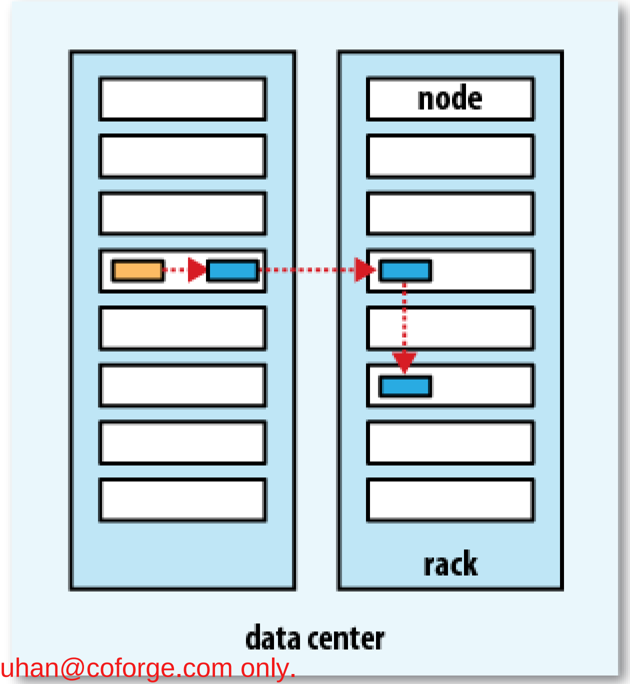
This ensures that the replication count is maintained all the time



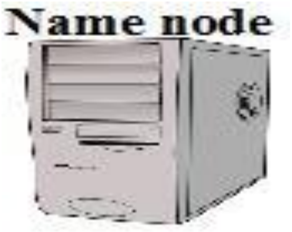
Replica Placement Strategy

Q. How does namenode choose which datanodes to store replicas on?

- Replica Placements are rack aware. Namenode uses the network location when determining where to place block replicas.
- Tradeoff: Reliability v/s read/write bandwidth e.g.
  - If all replica is on single node - lowest write bandwidth but no redundancy if nodes fails
  - If replica is off-rack - real redundancy but high read bandwidth (more time)
  - If replica is off datacenter – best redundancy at the cost of huge bandwidth
- Hadoop’s default strategy:
  - 1st replica on same node as client
  - 2nd replica on off rack any random node
  - 3rd replica is same rack as 2nd but other node
- Clients always read from the nearest no
- Once the replica locations is chosen a pipeline is built taking network topology into account



Name node & secondary name node in Hadoop 1.0



I know where the file blocks are..

I shall back up the data of name node

BEWARE !
I **do not** work in **HOT STANDBY** mode in the event of name node failure…..

In Hadoop 1.0, there is no active standby secondary name node.
(HA : Highly available is another term used for HOT/ACTIVE STANDBY)

If the name node fails, the entire cluster goes down ! We need to manually restart

The name node and the contents of the secondary name node has to be copied to it.

When to/not to use HDFS?

- HDFS is Good for…
- **Storing large files**
- Terabytes, Petabytes, etc.
- millions rather billions of files (less number of large files)
- Each file typically 100MB or more
- **Streaming data**
- WORM - write once read many times patterns
- Optimized for batch/streaming reads rather than random reads
- Append operation added to Hadoop 0.21
- Cheap commodity hardware
- **HDFS is not so Good for…**
- Large amount of small files
- Better for less no of large files instead of more small files
- Low latency reads
- Many writes: write once, no random writes, append mode write at end of file

Summary

Introduction to HDFS

Replication factor

Rack awareness

When not to use HDFS