Introduction to Big Data
&
Hadoop Ecosystem Training

Nagabhushan

Agenda – Day 1

- Introduction to Big Data & Hadoop
- Hadoop - Use Cases & History
- Commercial Distributions of Hadoop
- Hadoop’s Storage Architecture - HDFS
- Hadoop Setup

What is Big Data?

- *3 Vs of Big Data*
    - *Volume ➔ Size*
    - *Velocity ➔ Speed*
    - *Variety ➔ Different Forms*

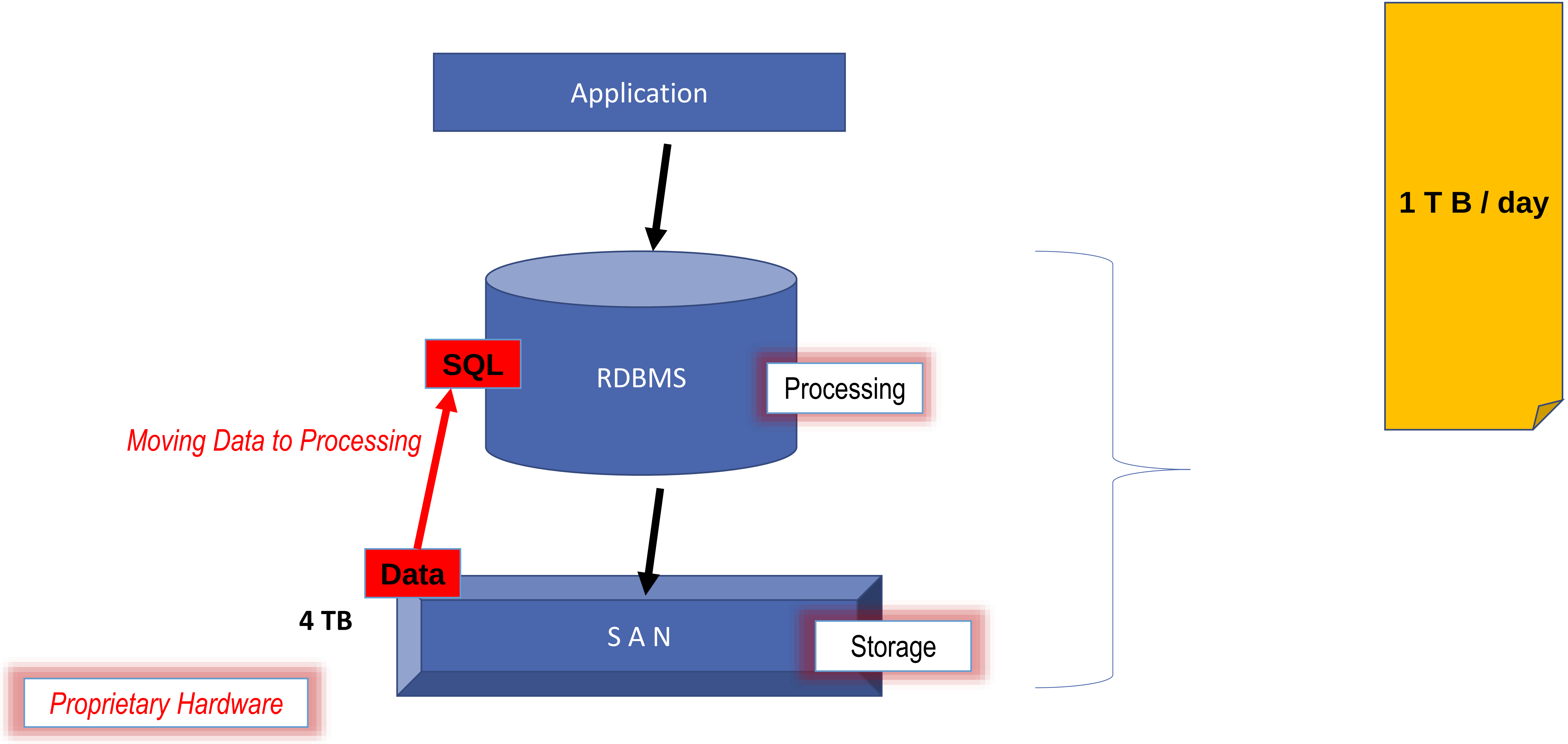
- ***Hadoop's V ➔ VALUE***

- *How to store Big Data? ➔ HDFS*
- *How to process Big Data? ➔ MapReduce (Hadoop 1.x) / YARN (Hadoop 2.x)*

Data Measurement Scale

1 Kilobyte	*KB*	*1000*
1 Megabyte	*MB*	*1000000*
1 Gigabyte	*GB*	*1000000000*
1 Terabyte	*TB*	*1000000000000*
1 Petabyte	*PB*	*1000000000000000*
1 Exabyte	*EB*	*1000000000000000000*
1 Zettabyte	***ZB***	***1000000000000000000000 X 5***
1 Yotabyte	*YB*	*1000000000000000000000000*

Problems with the traditional system



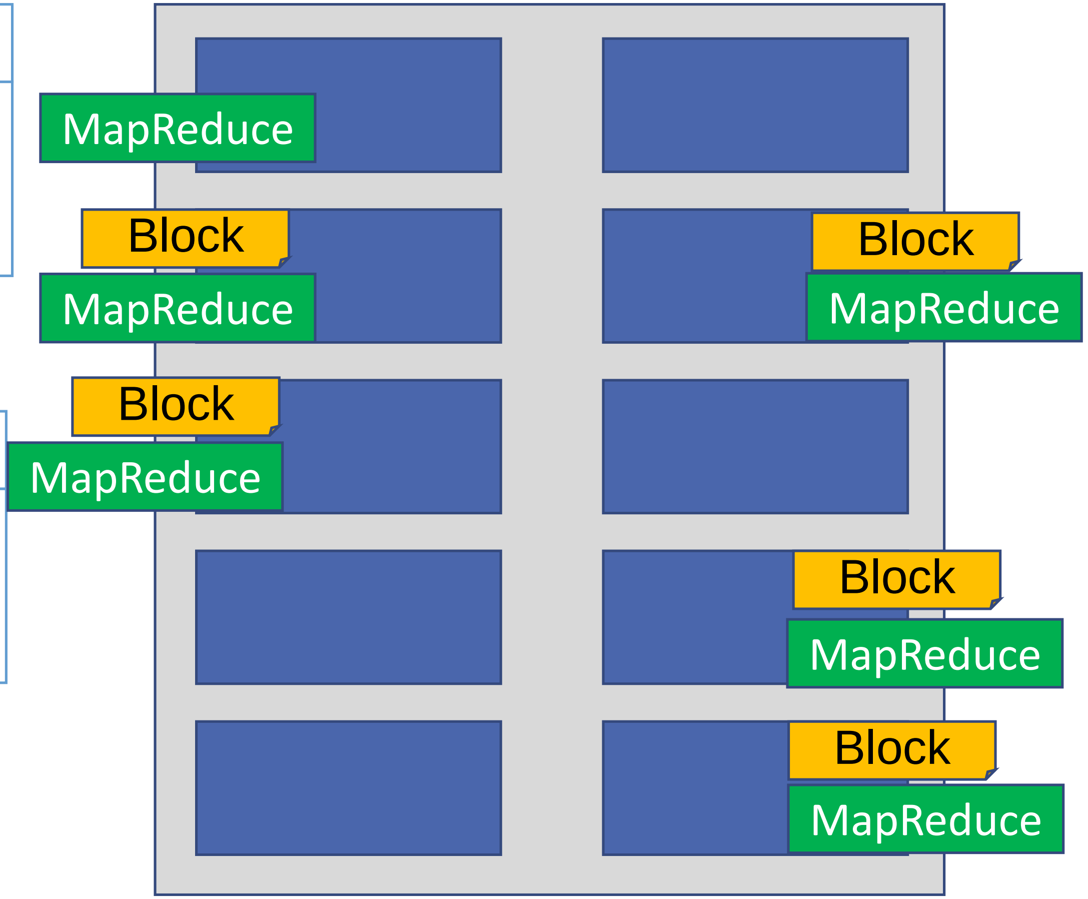
Big Data Systems to the Rescue ➔ *Hadoop*

Node

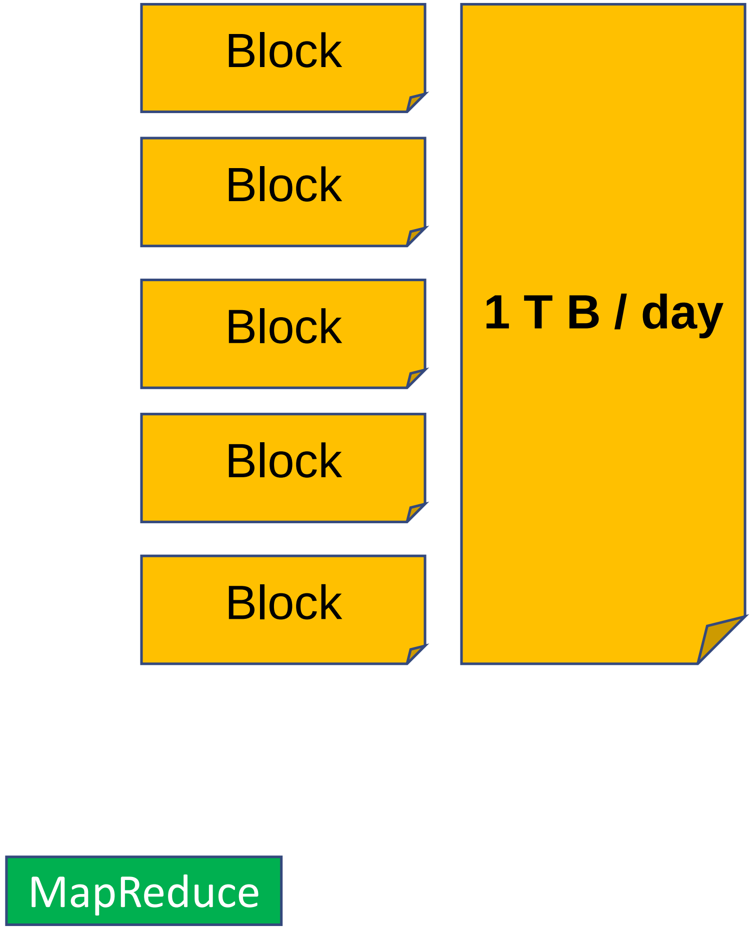
8 Cores Xeon Processor
64 GB RAM
24 TB Storage

Cluster

80 Cores Xeon Processor
640 GB RAM
240 / 4 TB Storage



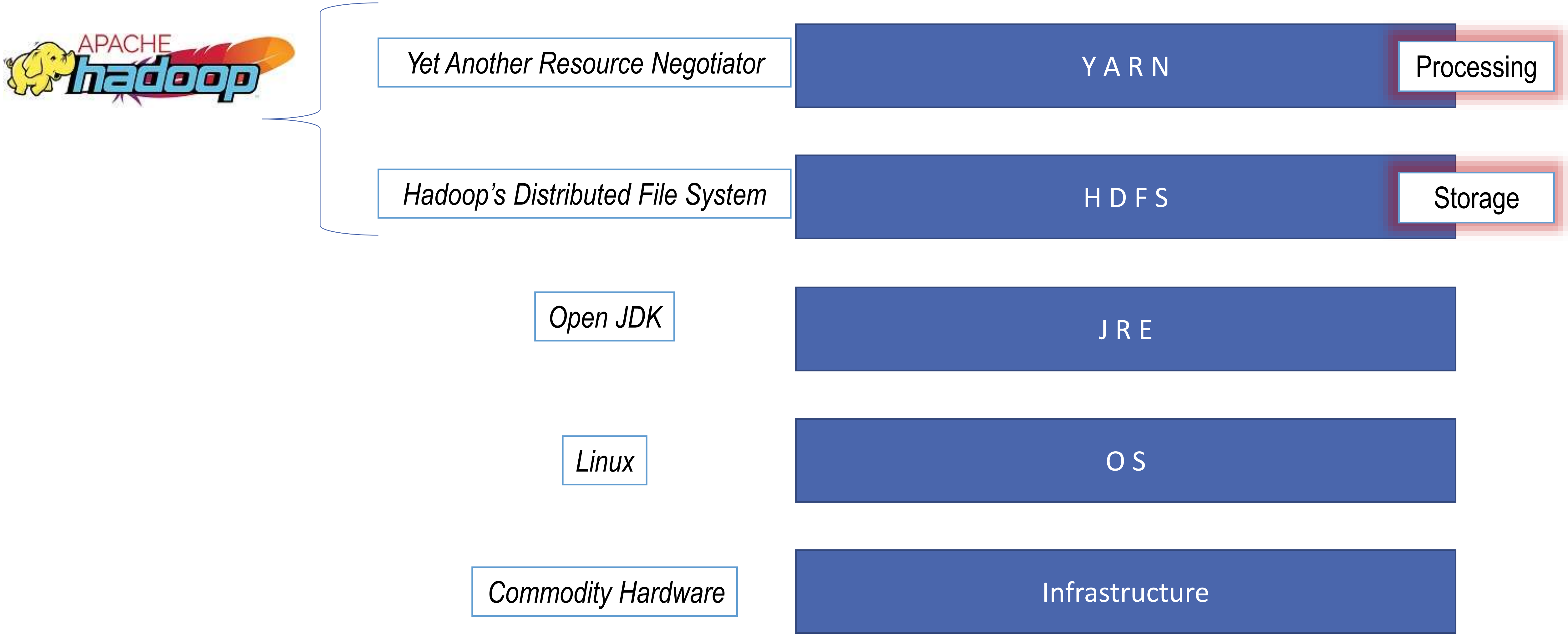
10 Node Hadoop Cluster



Moving Processing to Data

Commodity Hardware

Hadoop Layout / Node



Yet Another Resource Negotiator

YARN

Processing

Hadoop's Distributed File System

HDFS

Storage

Open JDK

JRE

Linux

OS

Commodity Hardware

Infrastructure

Features of Hadoop

- *Commodity Hardware*
- *Open Source http://hadoop.apache.org/*
- *Distributed Storage → Parallel Processing*
- *Scale Out Architecture (Horizontal Scaling)*
- *Fault Tolerance*
- *Data Locality → A new paradigm of moving <u>processing to data</u>*
- *Java software library*
- *WORM → Write Once Read Many*

Limitations of Hadoop

- *Batch Processing (MR approach)*
- *No updates (yet)* ***Alternative → MapR FS***
- *No Random Reads / Writes* ***Alternative → HBase***
- *Too many small data blocks / files*

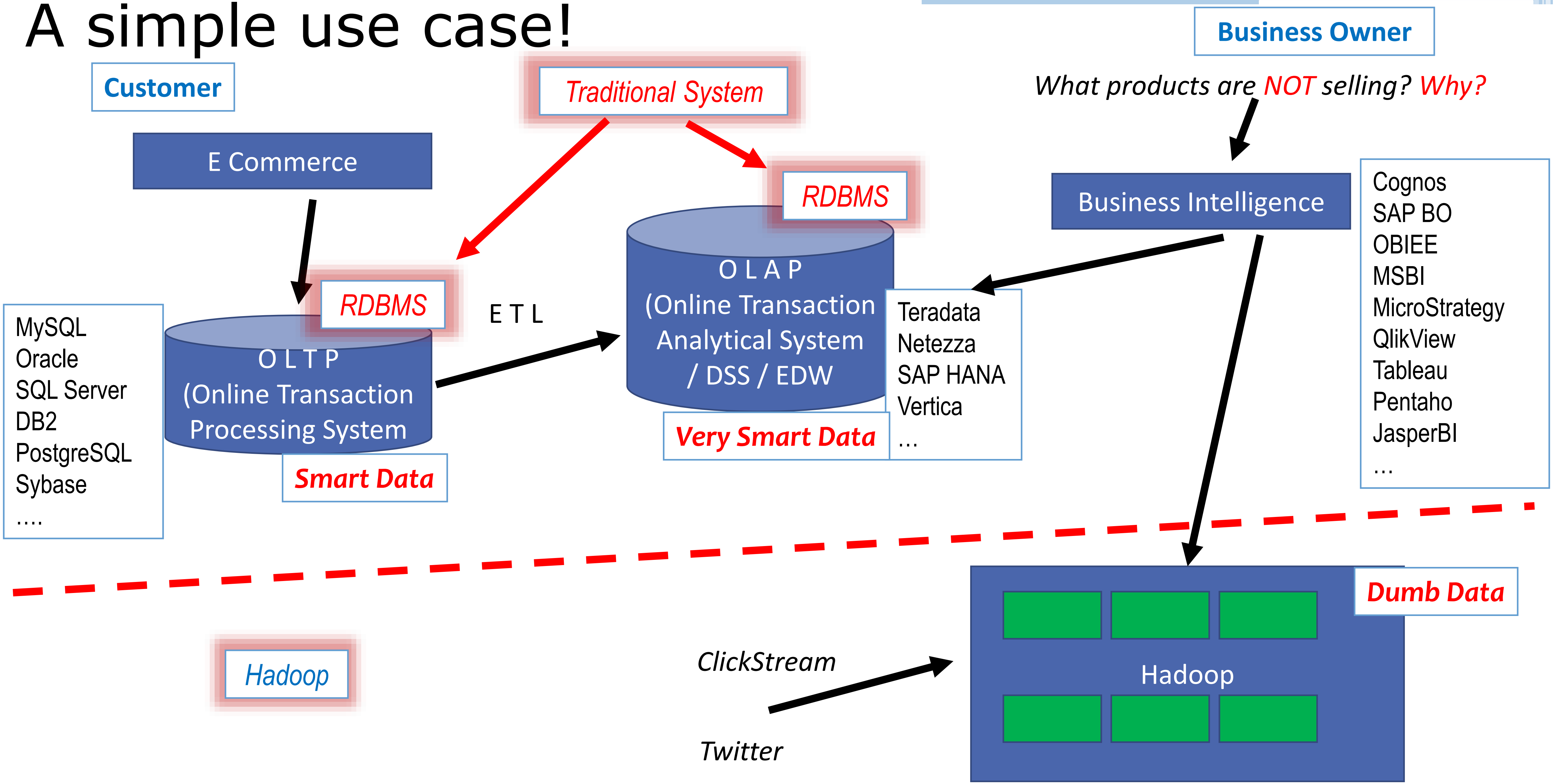
NoSQL Vs HDFS

- *HDFS ➔ Distributed File System* **Dumb Data**

- *NoSQL ➔ Distributed Database* **Smart Data**

Handle Big Data

A simple use case!



Customer

Traditional System

Business Owner

What products are NOT selling? Why?

E Commerce

RDBMS

O L T P
(Online Transaction
Processing System

Smart Data

MySQL
Oracle
SQL Server
DB2
PostgreSQL
Sybase
....

E T L

RDBMS

O L A P
(Online Transaction
Analytical System
/ DSS / EDW

Very Smart Data

Teradata
Netezza
SAP HANA
Vertica
…

Business Intelligence

Cognos
SAP BO
OBIEE
MSBI
MicroStrategy
QlikView
Tableau
Pentaho
JasperBI
…

Hadoop

ClickStream

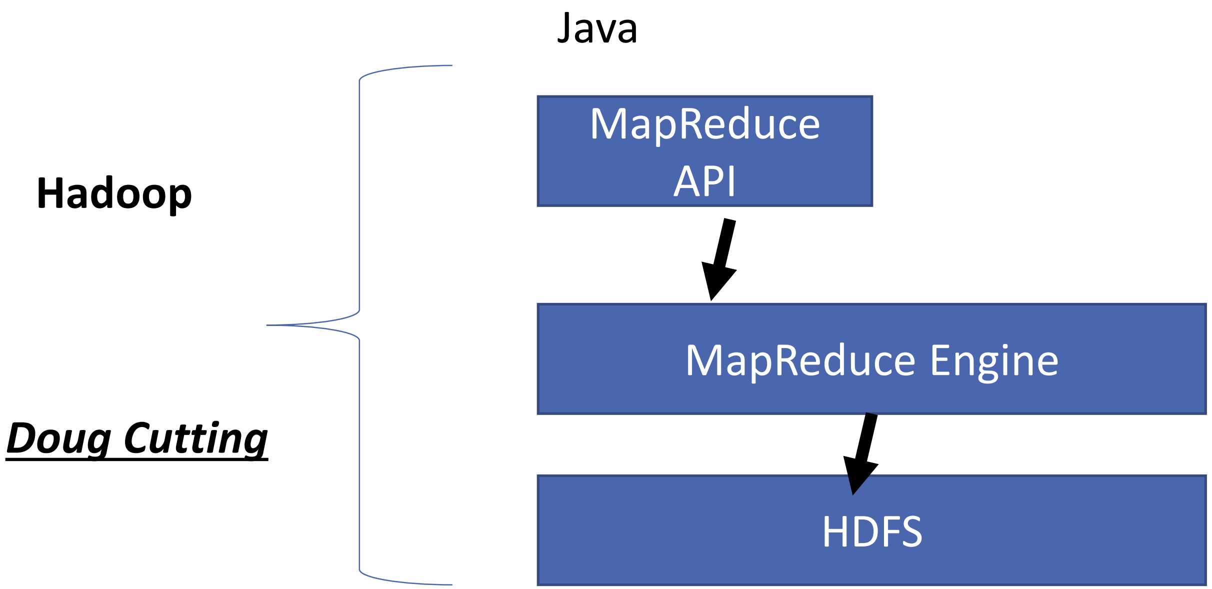
Twitter

Hadoop

Dumb Data

History of Hadoop

- *Google published whitepapers on “GFS” and “MapReduce” in Dec 2004*
- *Yahoo hired “Doug Cutting” to work on the whitepapers and Hadoop was the result*
- *Yahoo handed over the project to “Apache Software Foundation” in 2006*



Java

Hadoop

Doug Cutting

MapReduce API

MapReduce Engine

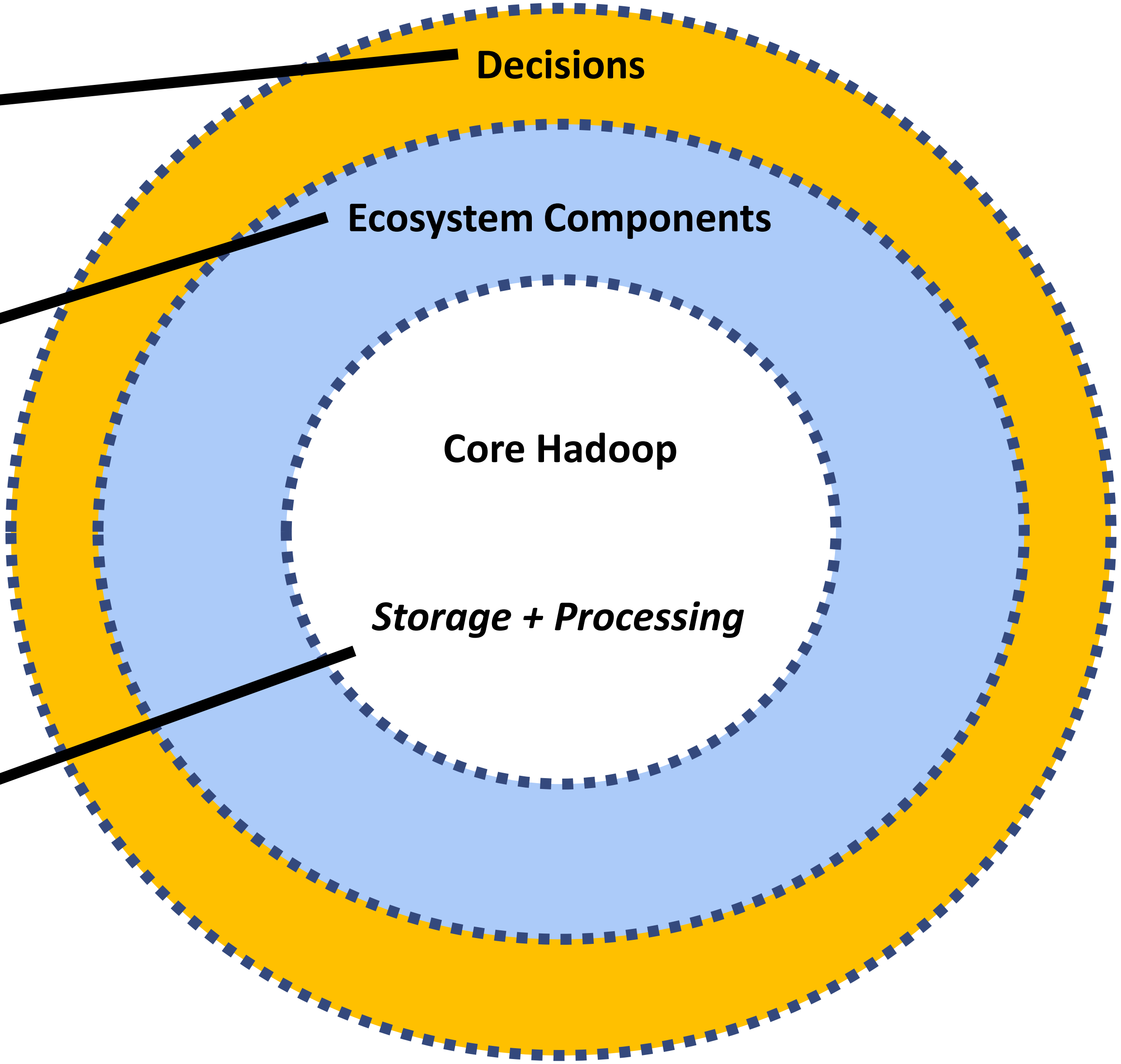
HDFS

Hadoop Ecosystem

Visualization, Metrics, Statistics

Analytics → ***MR Programs, Hive, Pig, HBase,***
Impala, Drill, Spark
Ingestion → ***Sqoop, Flume****, Kafka*
Administration → *Ambari, CM, MCS*
Synchronization → *ZooKeeper*
Workflow → *Oozie*
WebUI → *HUE*
Security → *Kerberos, Sentry..*

HDFS *+* ***YARN***

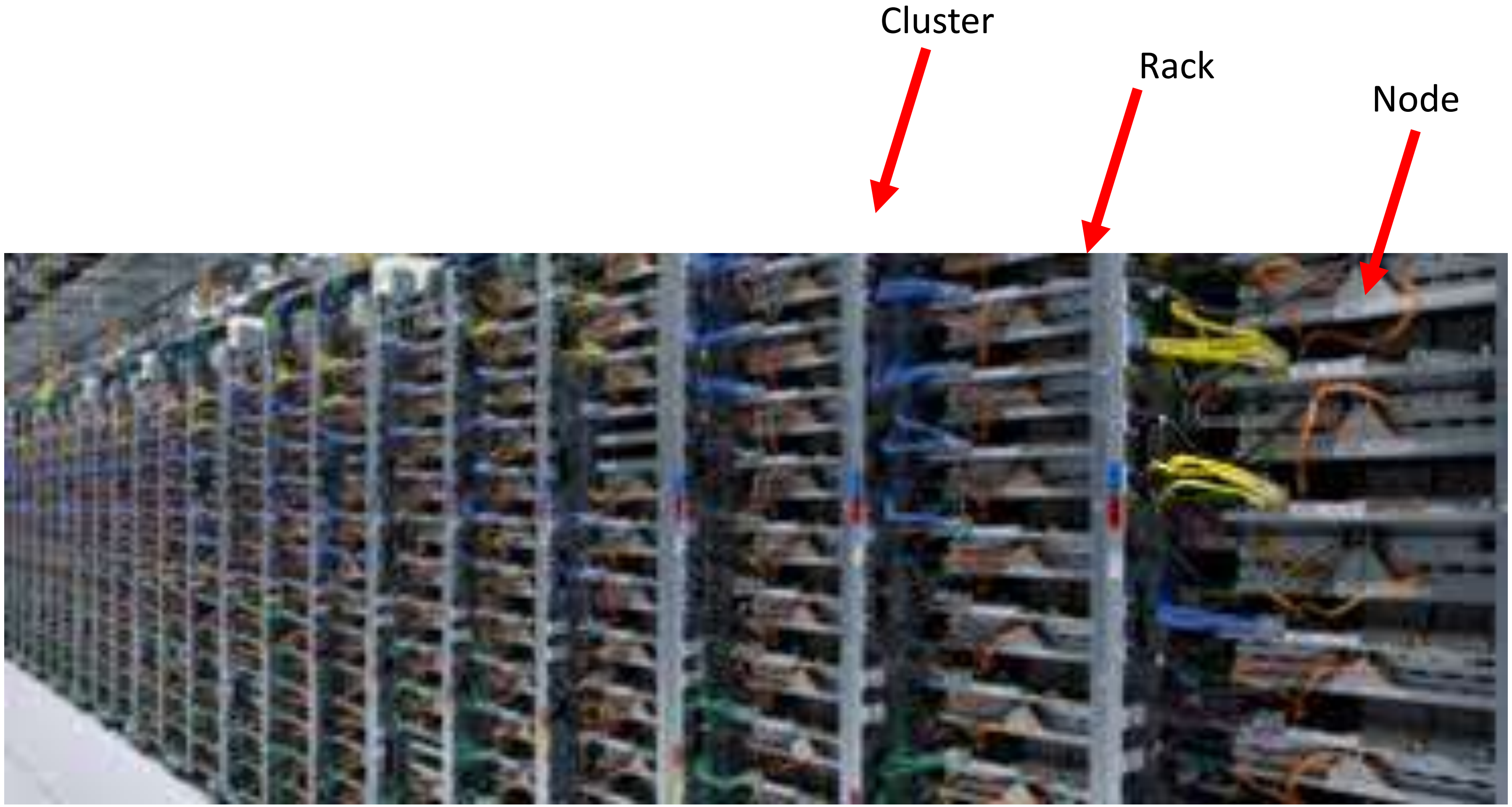


Commercial Distributions of Hadoop

- *Cloudera*
- *Hortonworks*
- *MAPR*
- *Big Insights (IBM)*

Hadoop’s Storage Architecture - HDFS

Hadoop Terminologies



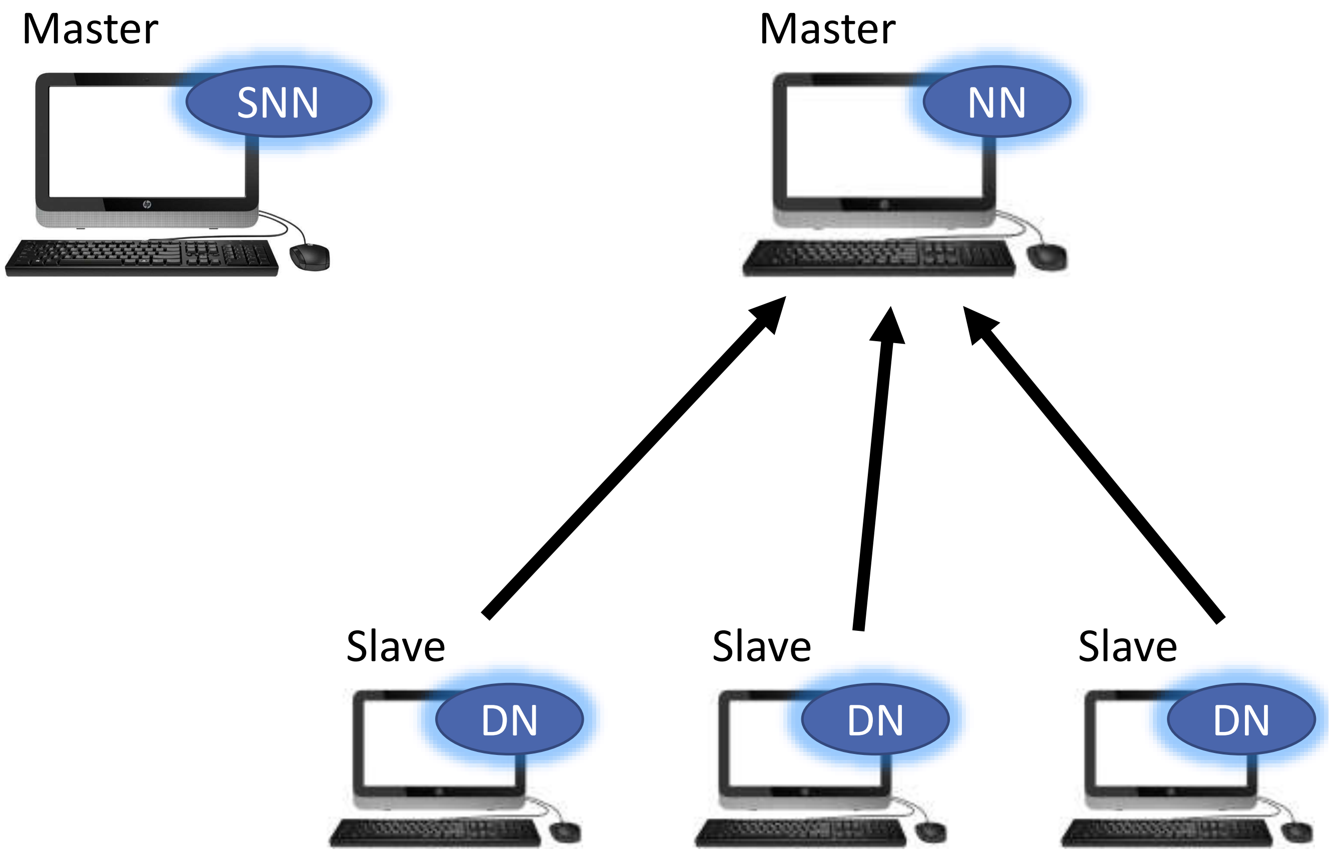
Cluster

Rack

Node

HDFS Daemons

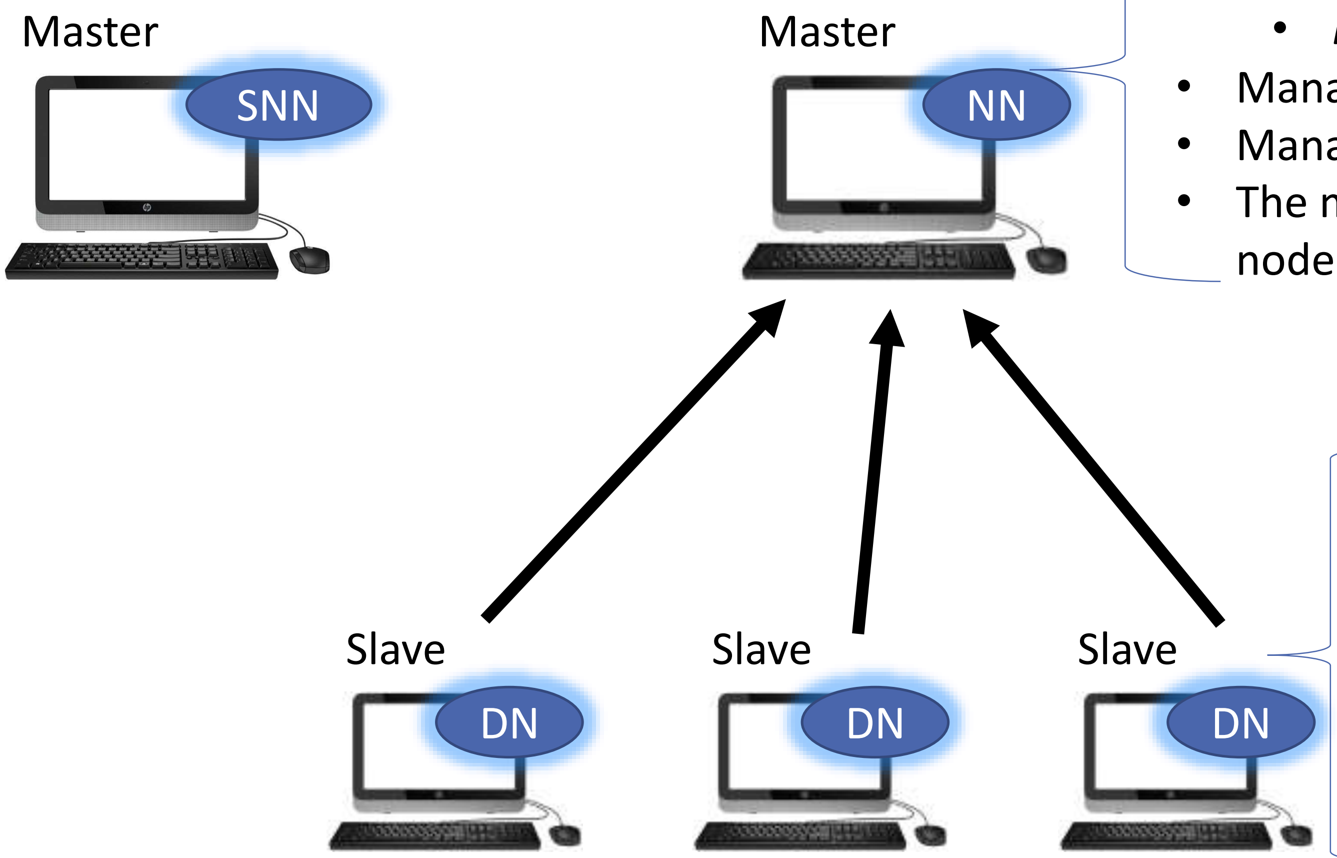
Master – Slave Architecture



NN	NameNode
DN	DataNode
SNN	Secondary NameNode

HDFS Daemons - Responsibilities

Not a hot backup



Master

SNN

Master

NN

Slave

DN

Slave

DN

Slave

DN

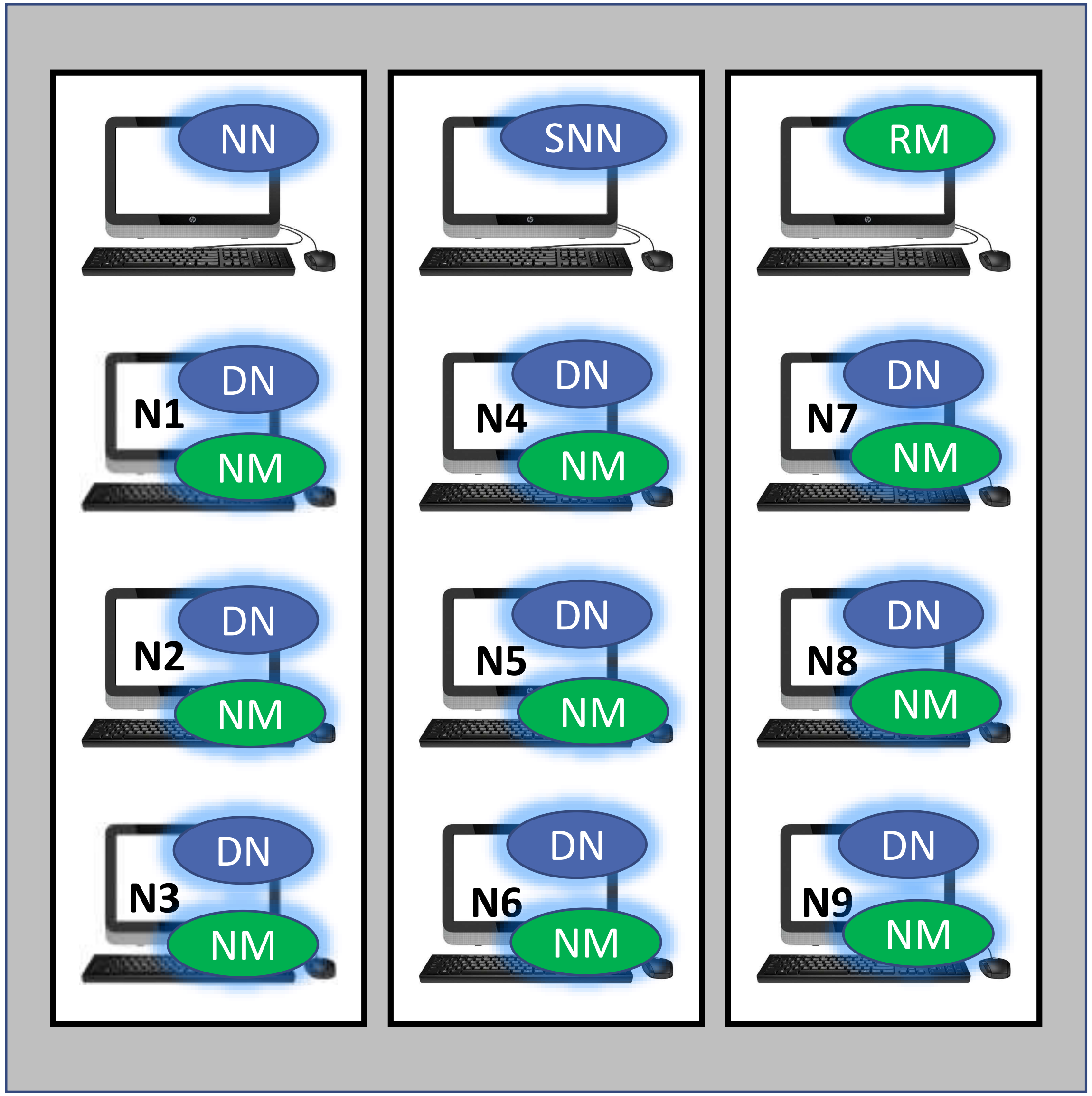
- Stores the metadata of the File System
  - *File Vs Block Mapping*
  - *Block Vs Node Mapping*
- Manages distribution ➔ allocation of DataNodes
- Manages block replication ➔ Fault Tolerance
- The metadata files are stored on the local file system of the node running NN

- Stores the “DataBlocks” of a file
- Data Integrity
- Sends heartbeats to NN
- Sends regular block reports to NN
- DataBlocks are stored as files on the local file system of the node running DN

Hadoop Daemons distributed over a cluster

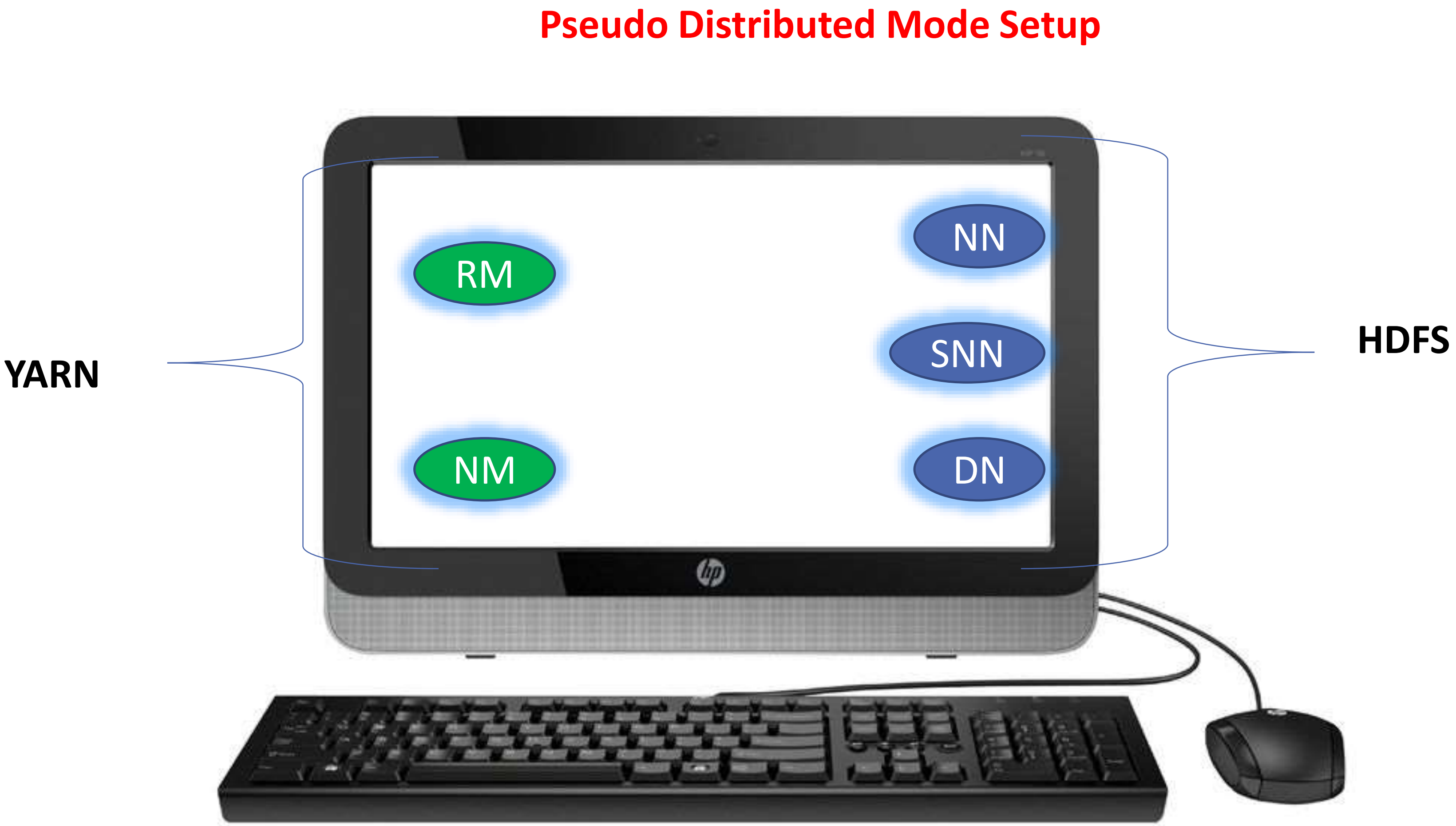
Hadoop Client

Gateway



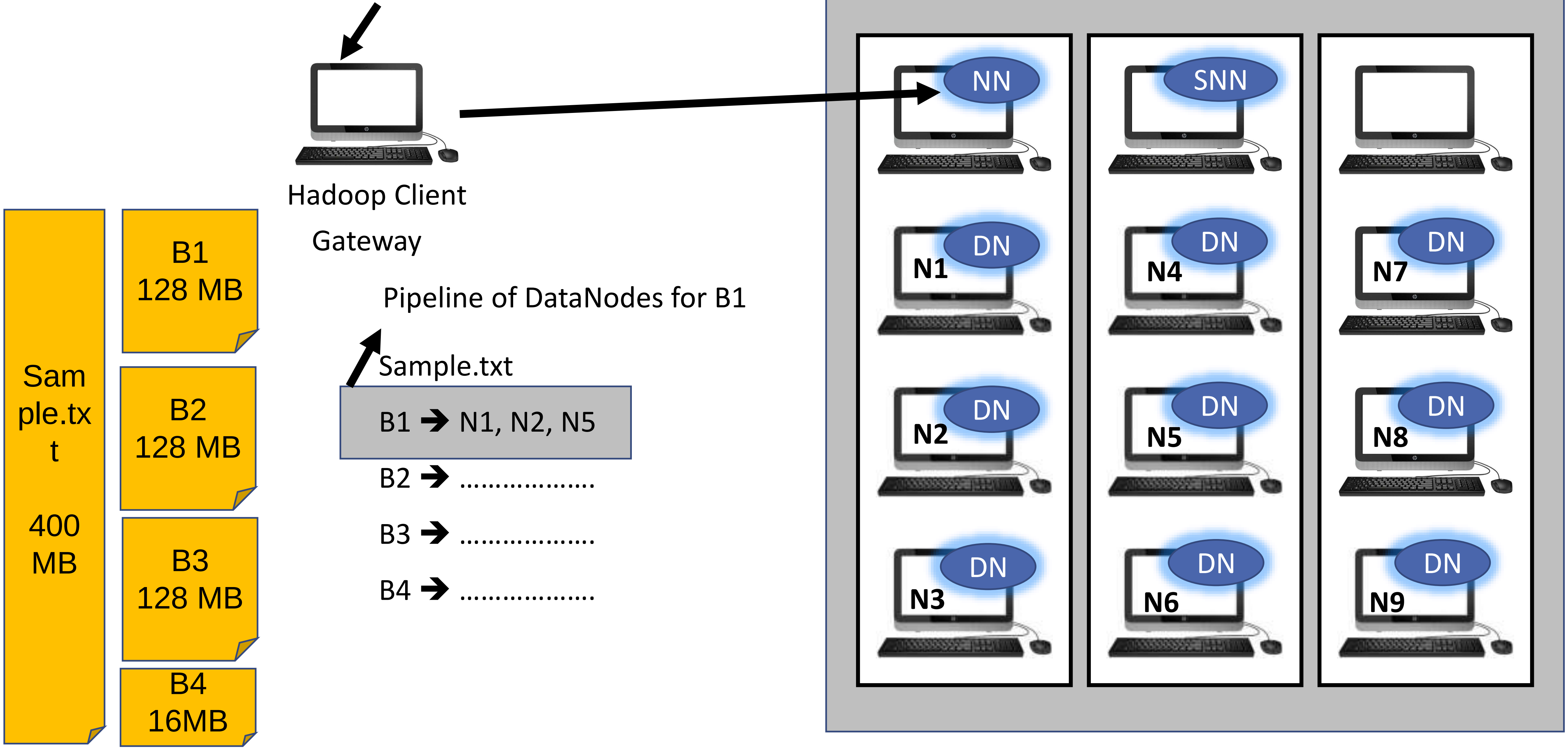
DataNode & NodeManager co-exist

Hadoop Daemons distributed over a single node cluster



Anatomy of a File Write - HDFS

```
$ hadoop fs –put <Source> <Destination>
```



Sample.txt

400 MB

B1
128 MB

B2
128 MB

B3
128 MB

B4
16MB

Hadoop Client

Gateway

Pipeline of DataNodes for B1

Sample.txt

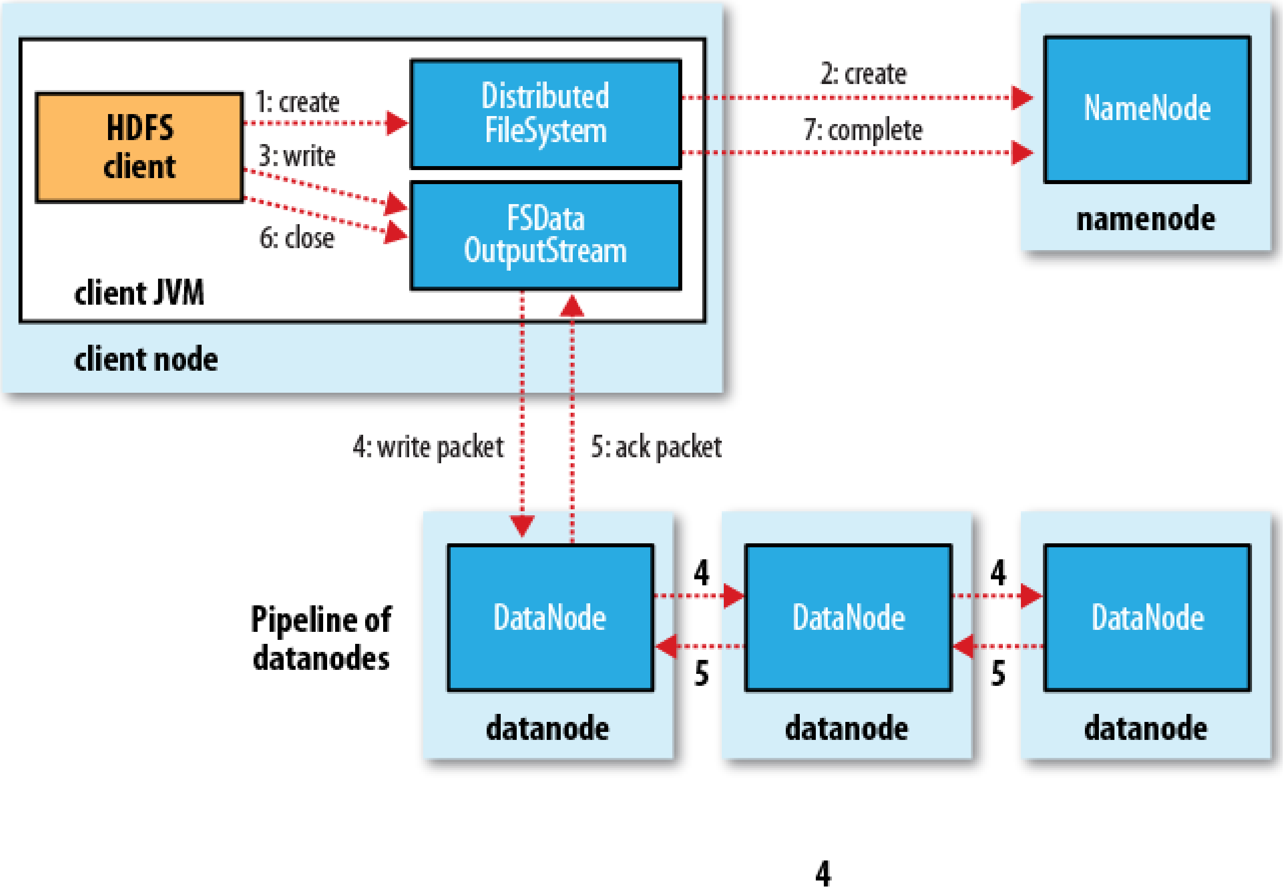
B1 ➔ N1, N2, N5

B2 ➔ ……………….

B3 ➔ ……………….

B4 ➔ ……………….

Anatomy of a File Write - HDFS



Page 70: Hadoop – The Definitive Guide

Rack Awareness

- *With a standard replication factor = 3, HDFS block placement policy is to put*
  - *1st replica on a node within a local rack*
  - *2nd replica on a different node in the local rack*
  - *3rd replica on a different node in a remote rack*

Hadoop Configuration Files

Default Hadoop Configuration

core-default.xml

hdfs-default.xml

mapred-default.xml

yarn-default.xml

dfs.replication = 3
dfs.blocksize = 134217728 = 128 MB
dfs.heartbeat.interval = 3
dfs.namenode.stale.datanode.interval = 30000

dfs.namenode.checkpoint.period = 3600
dfs.namenode.checkpoint.txns = 1000000

Customized Hadoop Configuration

core-site.xml

hdfs-site.xml

mapred-site.xml

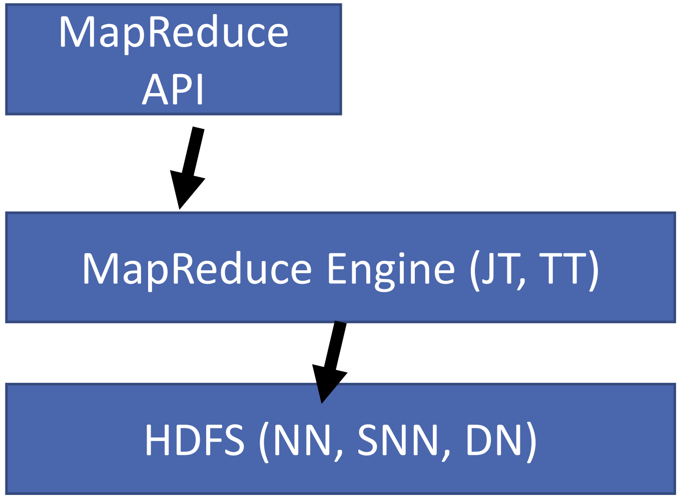
yarn-site.xml

dfs.replication = 1

$HADOOP_HOME/etc/hadoop ➔ Hadoop’s conf dir

Comparison

Hadoop 1.x

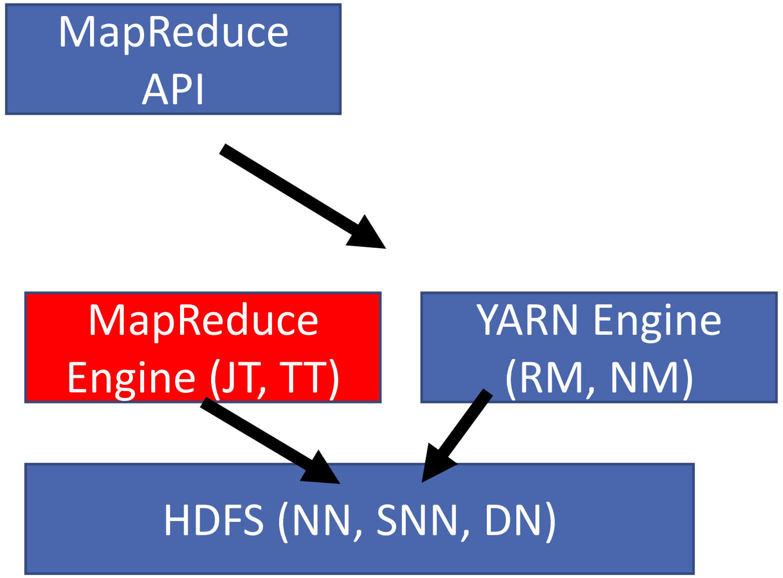


MapReduce API

MapReduce Engine (JT, TT)

HDFS (NN, SNN, DN)

Hadoop 2.x



MapReduce API

MapReduce Engine (JT, TT)

YARN Engine (RM, NM)

HDFS (NN, SNN, DN)

Hadoop Setup

Infrastructure

- *In premise*
- *SAN*
- *Cloud – AWS / GCP / Azure*
- ***Virtualization***

O S

- *RHEL*
- *CentOS*
- ***Ubuntu***
- *Fedora*
- *SUSE*
- *…*

JDK

- ***Open JDK***
- *Oracle JDK*
- *IBM JDK*
- *….*

Hadoop

- *Cloudera*
- ***Apache***
- *Hortonworks*
- *MAPR*
- *Big Insights*

Hadoop Setup Modes

- *Standalone Mode*
- ***Pseudo Distributed Mode***
- *Fully Distributed Mode*

Hadoop Setup Modes

- *Standalone Mode*
  - *Single Node*
  - *Non Distributed*
  - *Hadoop runs as a single Java process*

- ***Pseudo Distributed Mode***
  - ***Single Node & Pseudo Distributed***
    - ***HDFS ➔ 1 NN, 1 DN, 1 SNN***
    - ***YARN ➔ 1 RM, 1 NM***
  - ***Each Hadoop daemon runs in a separate JVM***

- *Fully Distributed Mode*
  - *Multi Node Setup*
  - *Production Setup*

Hadoop Setup Steps

- *Pre-Requisites*
  - *Linux*
  - *Java*
  - *ssh (passphraseless)*
- *Download and unpack Hadoop packages*
- *Customize Hadoop*
  - *core-site.xml*
  - *hdfs-site.xml*
  - *mapred-site.xml*
  - *yarn-site.xml*
  - *hadoop-env.sh*
- *Format the NameNode*
- *Start Hadoop Services*