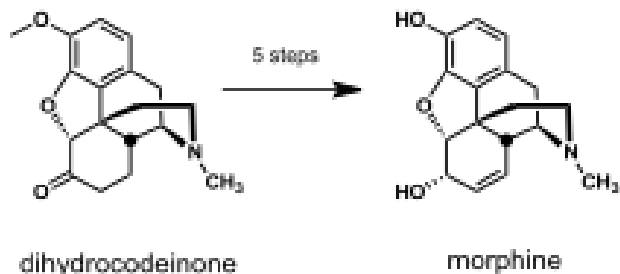
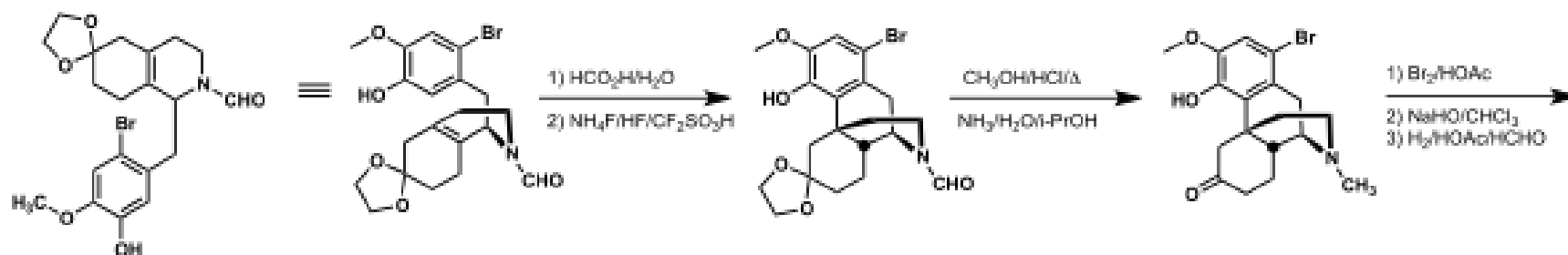
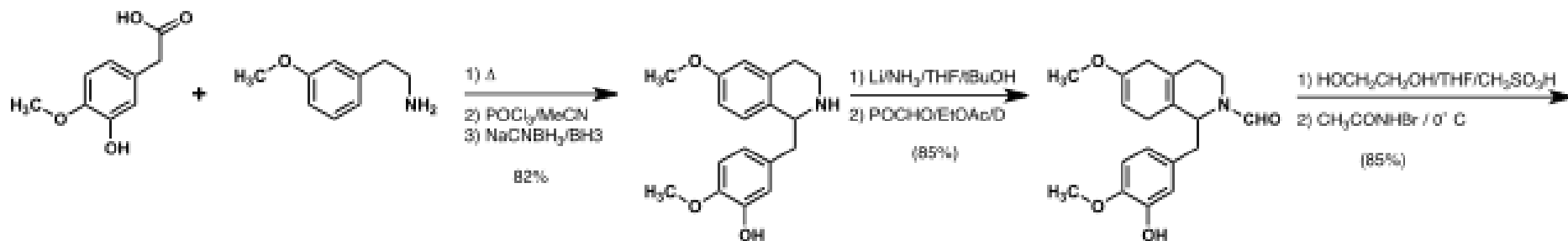


MANAGEMENT AND PROCESSING OF BIG DATA LEVEL-I

SESSION-1



Organic chemistry level-I

MANAGEMENT AND PROCESSING OF BIG DATA

LEVEL-I

SESSION-1





Know your classmates

Data vs Information

- **Data :**

- Simply fact or figure

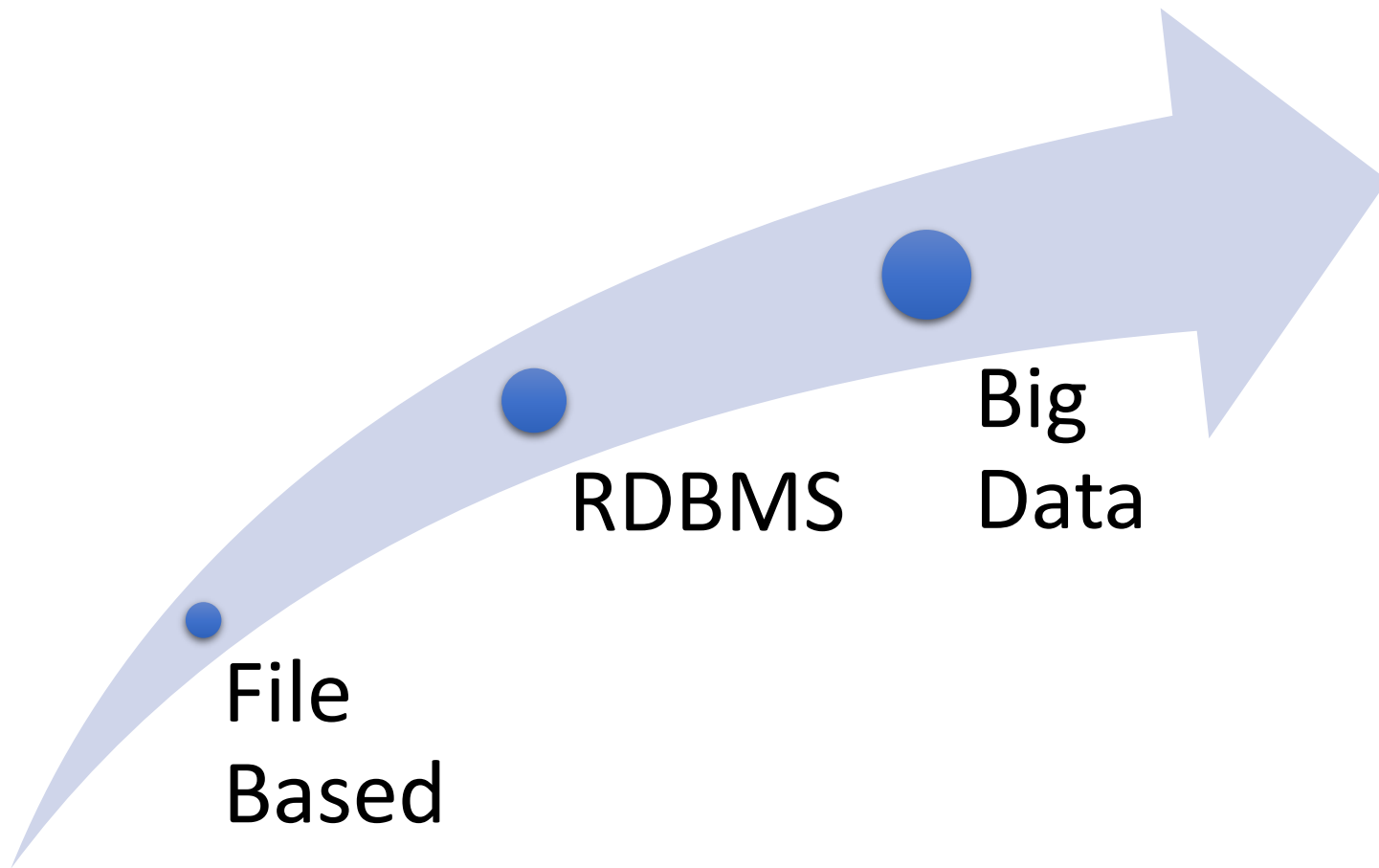
For example: a number 15

- **Information:**

- Context + data

For example: 15 degree centigrade is the temperature of Montreal on 07th Sept 2019 at 09:35 AM.

Evolution in Data management



What's Big Data?

- International Data Corporation (IDC) has measured data footprint in 2013: 4.4 zettabytes
- 1 zettabyte = 1 billion terabytes
- Forecast is to have 44 zettabytes by 2020
- Where does this data come from?

Ref: Hadoop definitive guide 4th edition, O'Reilly publications

Characteristics of Big Data

- Volume
- Velocity
- Variety
- Value

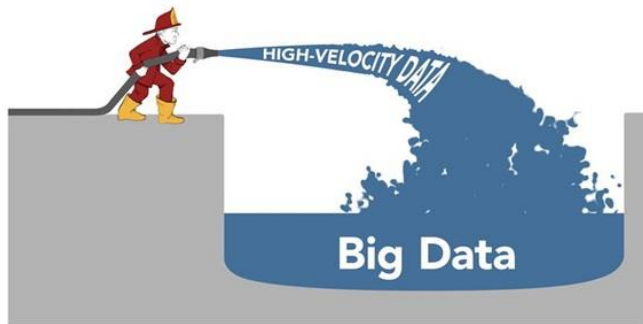


Volume

- Any guess how much amount of data we are producing within this room?
- Connected smart cars will generate 25GB data per hour

Ref: <https://qz.com/344466/connected-cars-will-send-25-gigabytes-of-data-to-the-cloud-every-hour/>

Velocity



- What happens in an internet second
 - 54,907 Google searches
 - 7,252 tweets
 - 125,406 YouTube videos
 - 2,501,018 emails sent

Ref: <http://www.dailymail.co.uk/sciencetech/article-3662925/What-happens-internet-second-54-907-Google-searches-7-252-tweets-125-406-YouTube-video-views-2-501-018-emails-sent.html#ixzz4sNJmz06e>

Variety

- Structured
- Semi structured
- Unstructured
- XML/JSON
- Web logs
- Sensor data



Value



Applications

- Finance
- Pharma
- Retail
- Manufacturing
- Insurance
- Travel industry

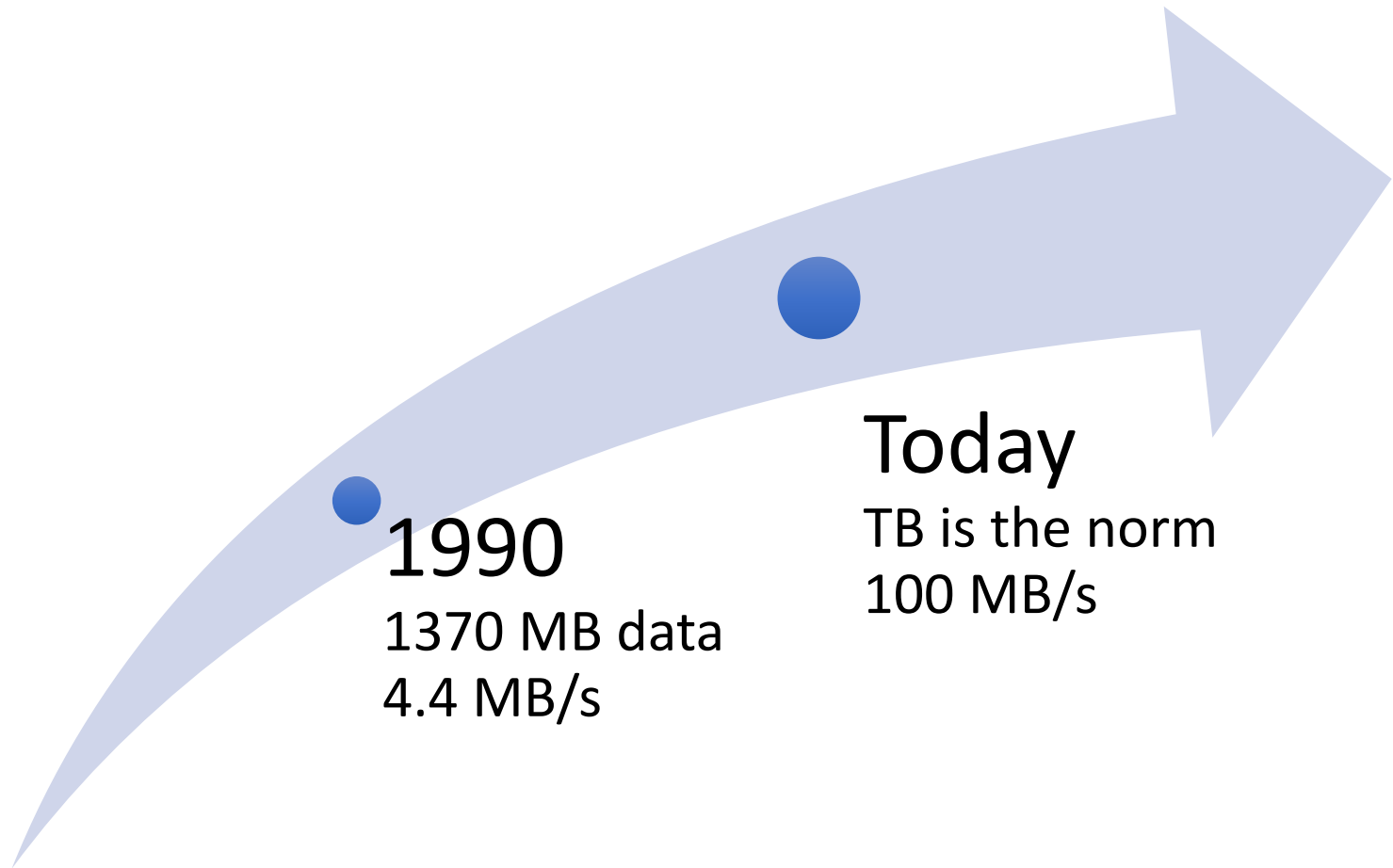
What is next?

- The good news is “We have big data to analyze”
- But the challenge is “How to store and process it”

What's the solution?

- Build a bigger system with increased computing power
- “In pioneer days they used oxen for heavy pulling, and when one ox couldn't budge a log, they didn't try to grow a larger ox. We shouldn't be trying for bigger computers, but for more systems of computers” – Grace Hopper

Storage Technology



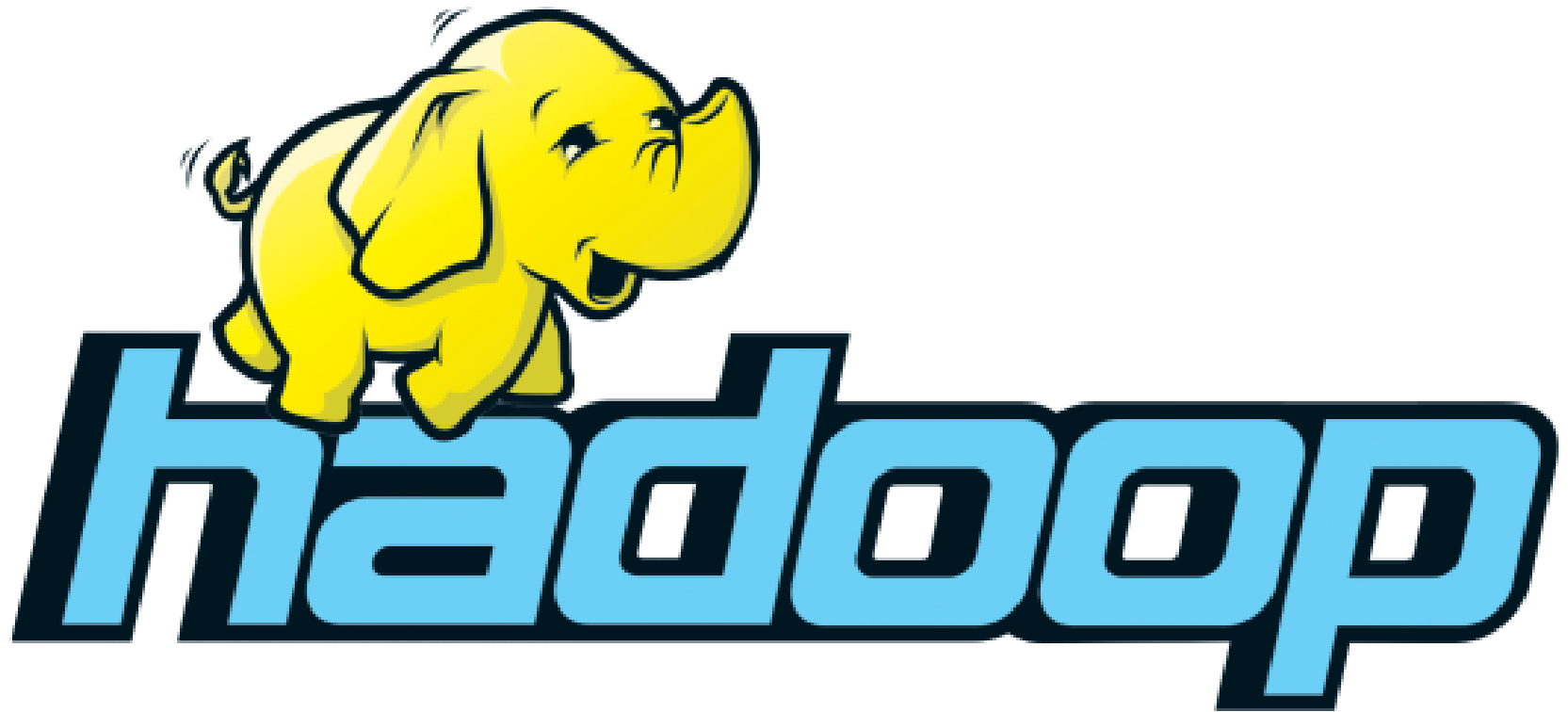
Volunteer computing

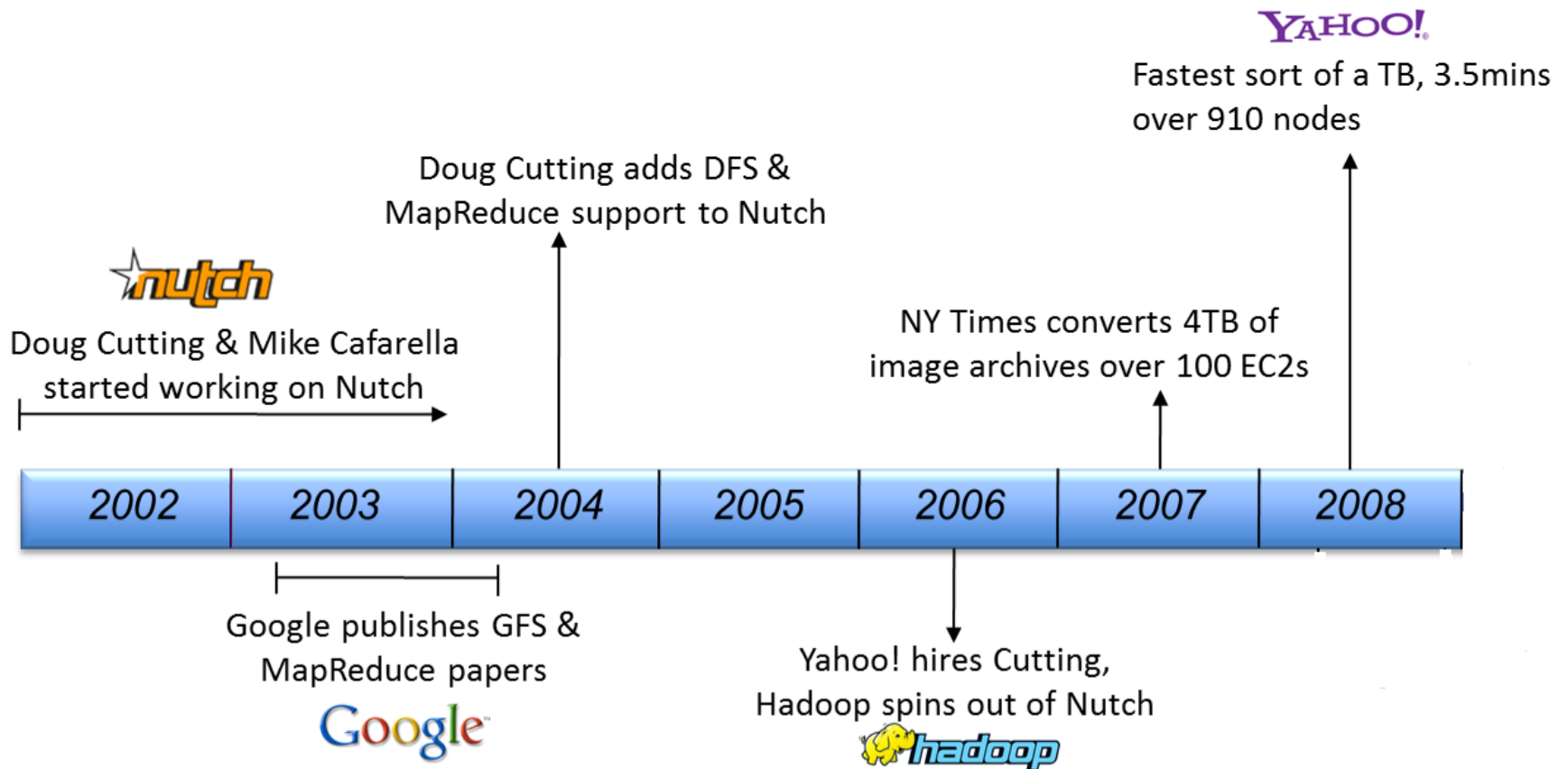
- System is highly compute intensive
- Small amount of data on remote machine
- Low bandwidth
- Based on Internet

Grid computing

- Based on Message Passing Interface (MPI)
- Uses shared filesystem
- Programmer has to think at task level as opposed to data level
- Missing abstraction of fault tolerance

Distributed Computing



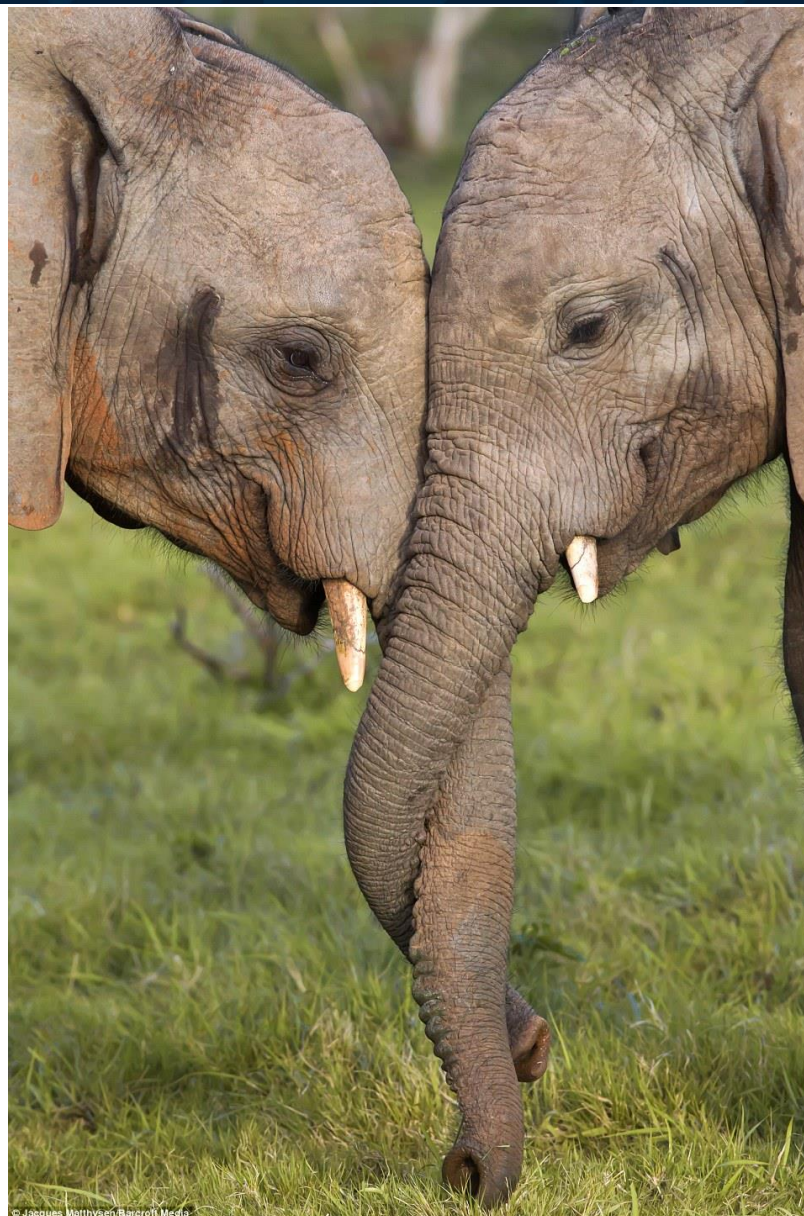


Ref: Couldera Inc

Major Vendors

- Cloudera
- Hortonworks
- MapR
- Amazon cloud
- Databricks
- Microsoft cloud

- HDFS
- Map Reduce
- YARN
- Pig
- Hive
- Impala
- Sqoop
- Spark



- Shell scripting
- SQL
- Git
- IntelliJ Idea
- Java
- Scala
- Python

Resources

- Content repository

https://github.com/shyam-kantesariya/big_data_course

- Email:

kantesariyashyam@gmail.com

- LinkedIn:

<https://www.linkedin.com/in/kantesariyashyam/>

Environment Setup

- Git Bash
- IntelliJ IDEA
- Cloudera VM

Exercise 1: Unix shell

- Please refer to Exercise 1 document

Brain Teaser

- $A[n]$ is an array of n integers. Function *sum_array* does summation of all elements. But for a bigger array it runs out of heap memory and program crashes.
- Help me rewrite the same recursive function so that it runs for any array size

```
def sum_array(n):  
    if n==0:  
        return A[n]  
    else:  
        return A[n] + sum_array(n-1)
```



Recap

- Data processing trend
- Big Data and its characteristics
- Applications of big data
- Various computing technologies
- History of Hadoop
- Unix commands

Hadoop Components

HDFS: Storage

Namenode: Master node

Datanode: Worker node

Job Tracker: Master Coordinator Process

Task Tracker: Worker Coordinator Process

Exercise 2: HDFS Commands

- Please refer to Exercise 2 document

HDFS

- HDFS is a file system designed for storing very large files with streaming data access patterns, running on clusters of commodity hardware

Large Files

Streaming data
access

Commodity
hardware

HDFS Blocks

- Single unit of storage
- Default block size is 128MB
- Size of block will drive the ratio of time to read a block to the seek for a block

Exercise

- What should be the block size to make seek time 1% of read time for given hardware configuration
 - Seek time: 10 ms
 - Data read rate: 100MB/s
- Solution:
 - Let say x MB is the block size then read time = $x/100$ seconds
 - To fulfill the given condition 1% of $x/100 = 10$ ms
 - Hence $x=100$ MB

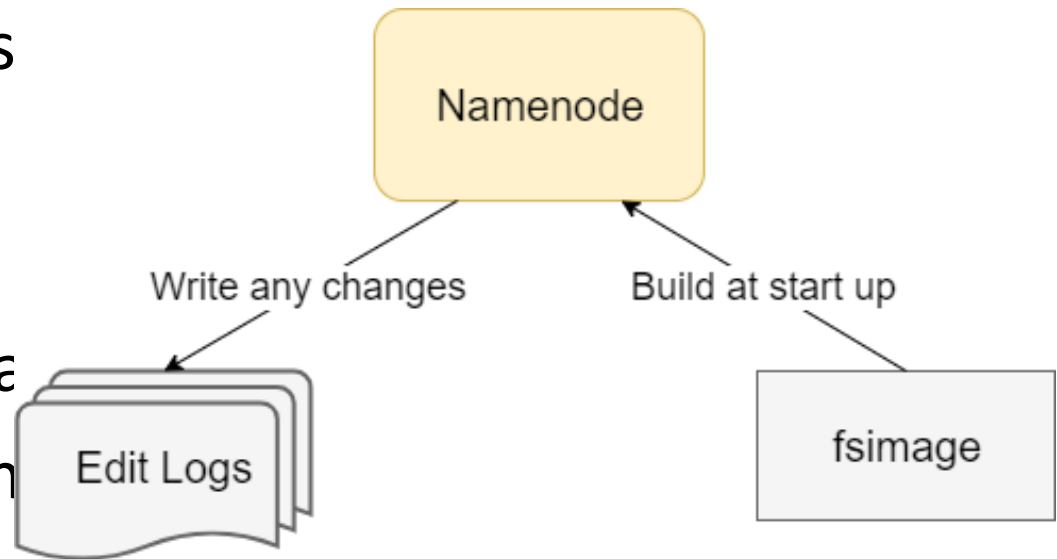
Benefits of blocks

- Files can be larger than a single disk
- Simplicity at storage level as data node doesn't store any metadata
- Fault tolerance by replicating blocks

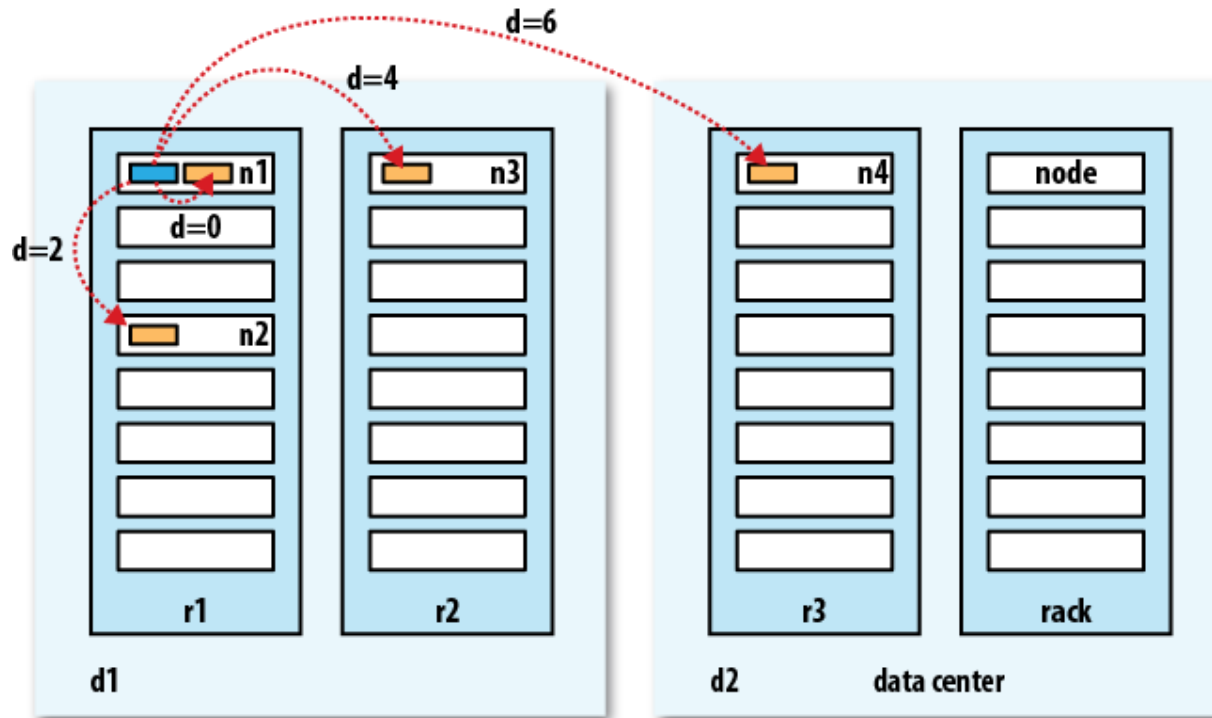
Filesystem metadata

- Namenode stores the metadata

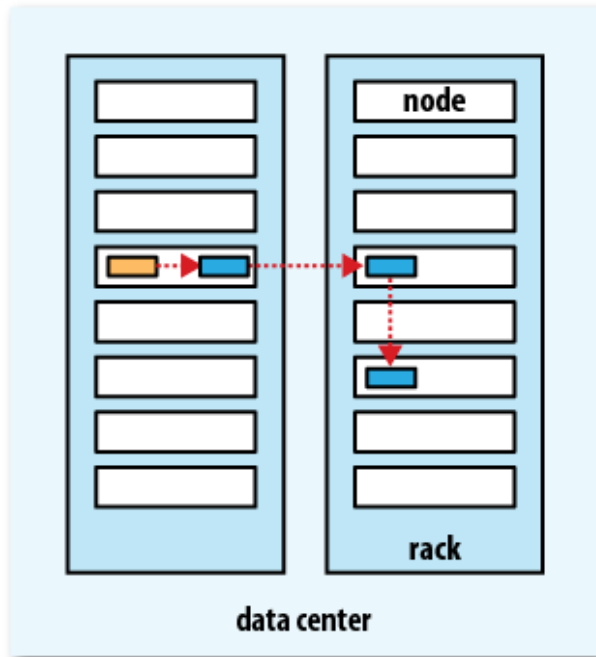
- Backup of metadata secondary namenode



Network Topology



Rack awareness



HDFS CLI Read Commands

- Copy a file from local file system to HDFS

```
hadoop fs -copyFromLocal <source> <target>
```

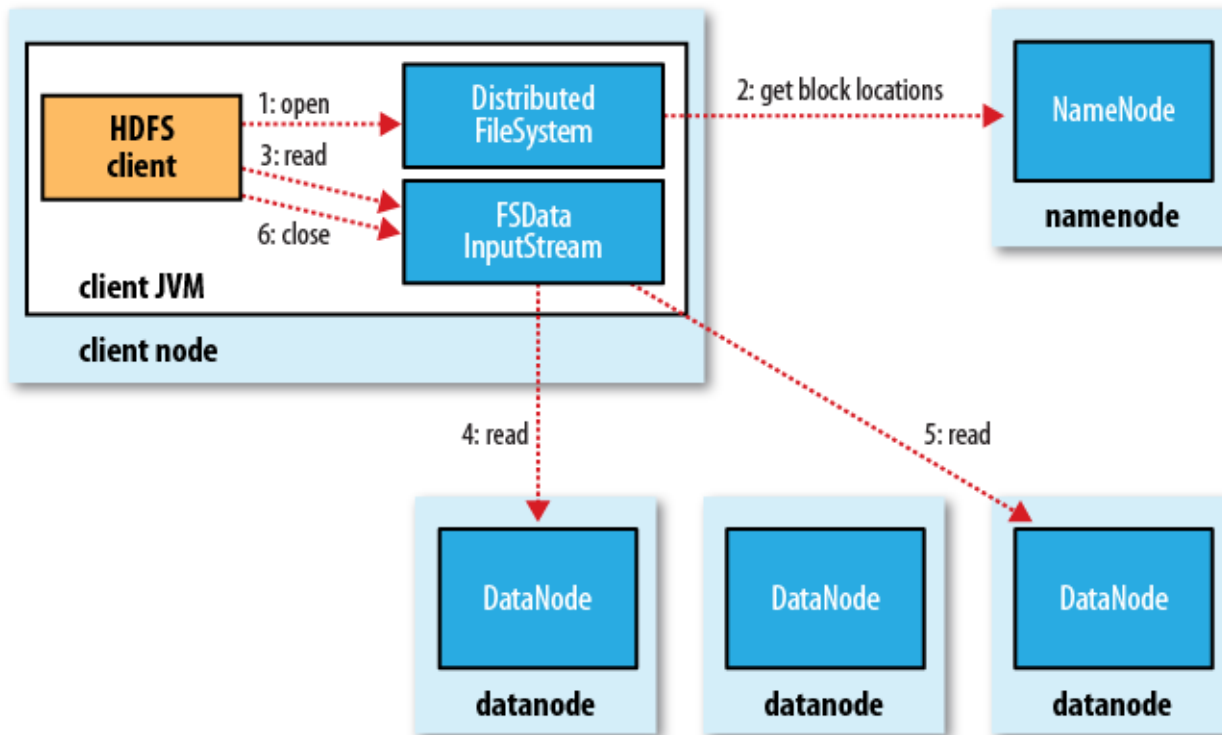
- Get merged file from HDFS to local file system

```
hadoop fs -getmerge <source> <target>
```

- Cat a file from HDFS

```
hadoop fs -cat <filename>
```

HDFS Read operation



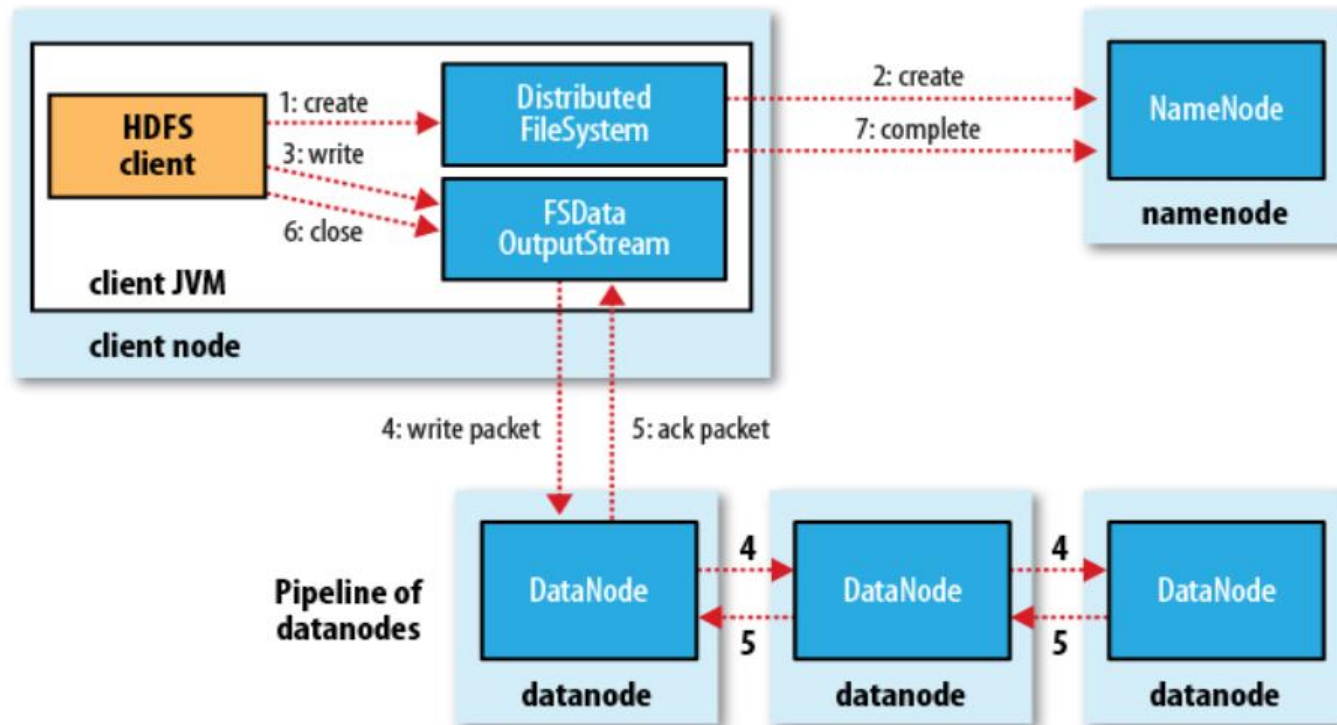
HDFS CLI Write Commands

- Write a file from local file system to HDFS

```
hadoop fs -copyFromLocal <source> <target>
```

```
hadoop fs -put <source> <target>
```

HDFS Write operation



HDFS not made for

- Low-latency data access
- Lots of small files
 - Each file|block|directory stores around 150 bytes of metadata. Hence 1 million files each of one block will consume 300 MB of storage on Namenode
- Multiple writers, arbitrary file modifications

Exercise

- Calculate the memory(RAM) requirement on Namenode for given cluster configurations
 - Cluster size: 200 nodes
 - Storage capacity of each node: 24 TB
 - Block size: 128MB
 - Replication factor: 3
 - Metadata storage size for each block: 150 bytes
- Solution
 - $(200 * 24 * 10^{12} * 150) / (128 * 10^6 * 3)$

RDBMS vs Hadoop

Attribute	RDBMS	Hadoop
Data Size	Gigabytes	Petabytes
Access	Interactive & Batch	Batch
Updates	Multiple Read/Write	Write once, Read multiple times
Transaction	ACID	None
Structure	Schema-on-write	Schema-on-read
Integrity	High	Low
Scaling	Nonlinear	Linear

Reference

- Apache Hadoop

<https://hadoop.apache.org/>

- Reference book: Hadoop definitive guide by Tom White

<https://www.oreilly.com/library/view/hadoop-the-definitive/9781491901687/index.html>

- Cloudera VM

https://www.cloudera.com/downloads/quickstart_vms/5-13.html

- IntelliJ Idea

<https://www.jetbrains.com/idea/download/#section=windows>

- Git bash

<https://git-scm.com/downloads>

- Unix

<http://www.ee.surrey.ac.uk/Teaching/Unix/>