Big Data Analytics

Mr. Abhaya Kumar Sahoo
School of Computer Engineering
KIIT, Deemed to be University
Bhubaneswar

Contents

- Big Data Introduction
- Hadoop Introduction
- Different types of Components in Hadoop HDFS
- MapReduce
- PIG
- Hive

Big Data Introduction

Have you hard " Flood of Data" ?

Where is the source?

What is next ?

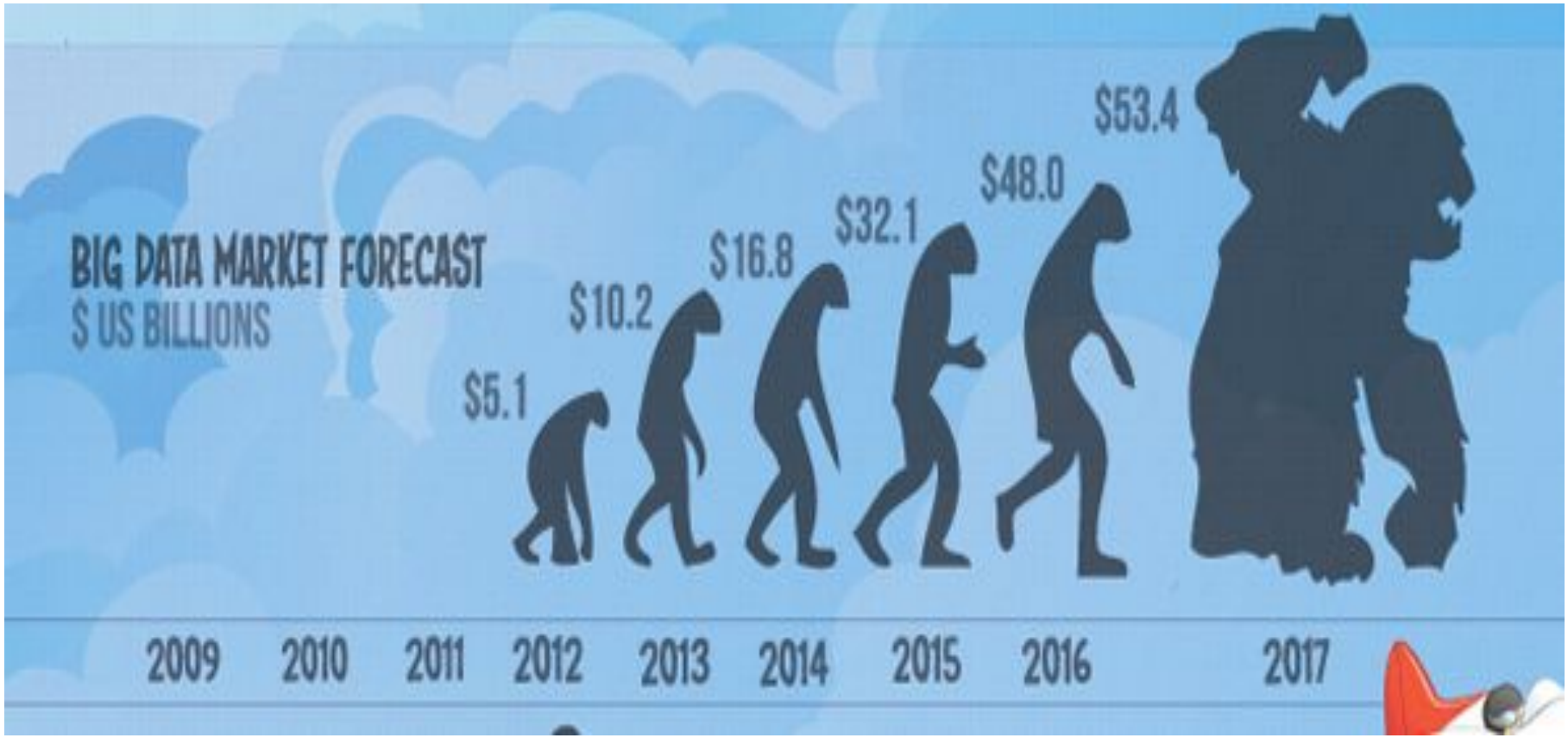
Survey

- The New York Stock Exchange generates about one terabyte of new trade data per day.
- Facebook hosts approximately 10 billion photos, taking up one petabyte of storage.
- Ancestry.com, the genealogy site, stores around 2.5 petabytes of data.
- The Internet Archive stores around 2 petabytes of data, and is growing at a rate of 20 terabytes per month.
- 6,000 pictures uploaded to flickr.
- 1.500 blogs.
- 600 new youTube video.

Survey

- The Large Hadron Collider near Geneva, Switzerland, will produce about 15 petabytes of data per year.
- Google process 20Petabytes in a day.
- Wayback machine has 3petabytes + 100TB per month
- eBay has 6.5 petabytes of user data + 50TB per day.
- Google receives 20 lakhs search quires.
- Facebook receives 34,722 likes in a day.
- Apple receives 47,000 App downloads.
- 37,000 minutes of call on Skype.
- 98,000 posts on tweets.

Now in 2017


BIG DATA MARKET FORECAST
$ US BILLIONS
$5.1
$10.2
$16.8
$32.1
$48.0
$53.4
2009
2010
2011
2012
2013
2014
2015
2016
2017

Big Data Definition

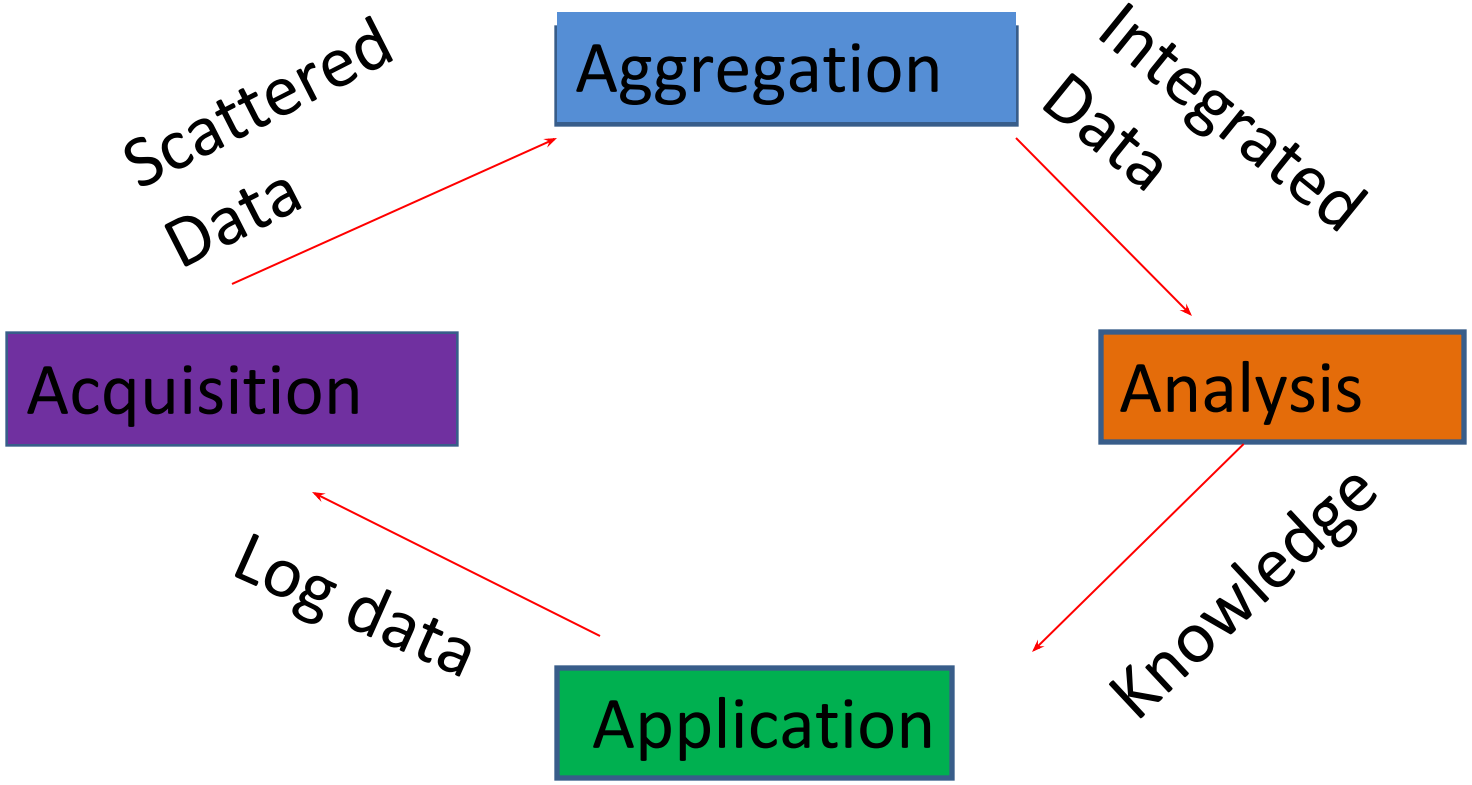
No single standard definition…

"*Big Data*" is data whose scale, diversity, and complexity require new architecture, techniques, algorithms, and analytics to manage it and extract value and hidden knowledge from it…

"Big Data are high-volume, high-velocity, and/or high-variety information assets that require new forms of processing to enable enhanced decision making, insight discovery and process optimization" (Gartner 2012)

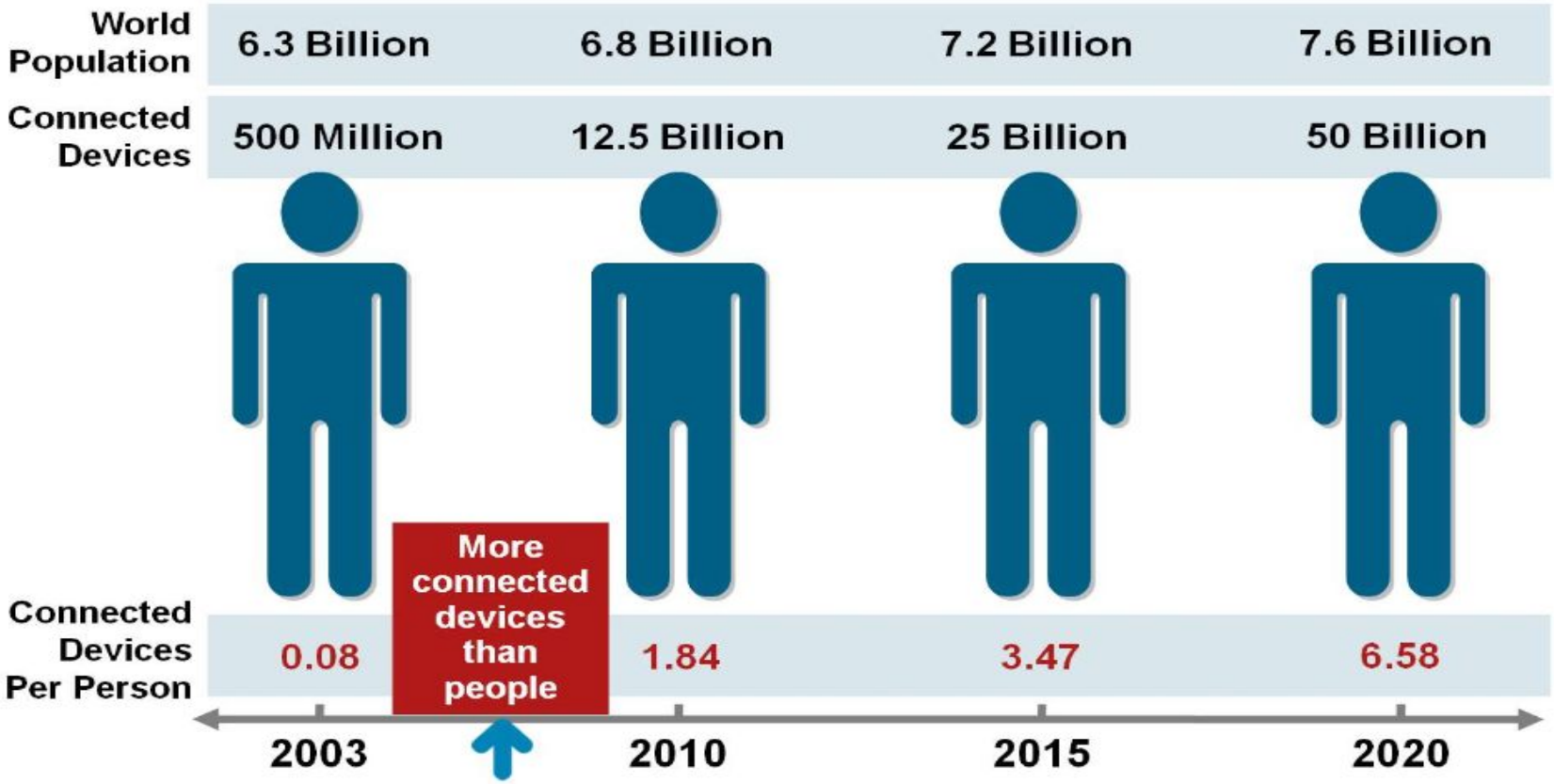
- Complicated (intelligent) analysis of data may make a small data "appear" to be "big"
- Any data that exceeds our current capability of processing can be regarded as "big"

Lifecycle of Data: 4 “A”s



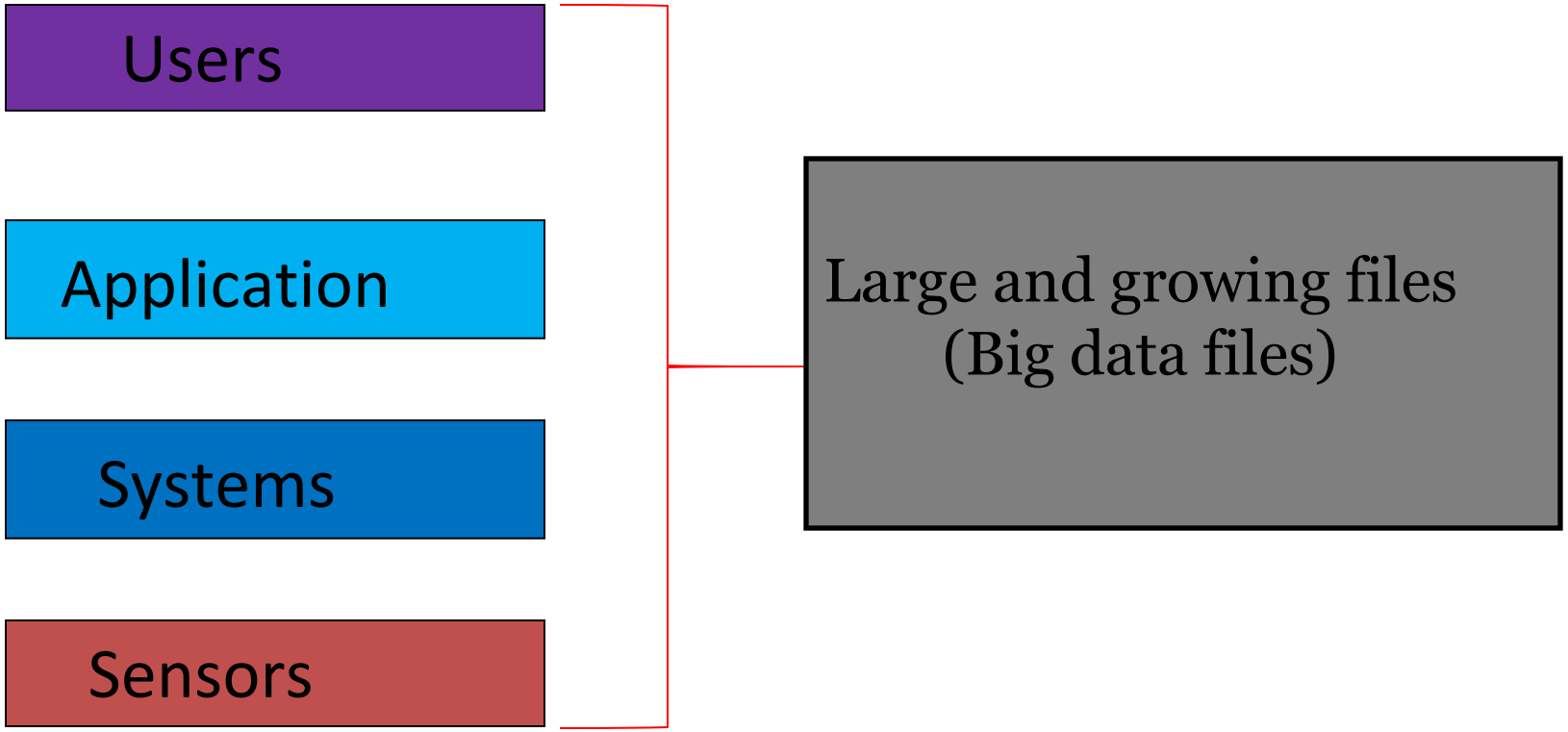
Why Big Data ?

Figure 1. The Internet of Things Was "Born" Between 2008 and 2009



Source: Cisco IBSG, April 2011

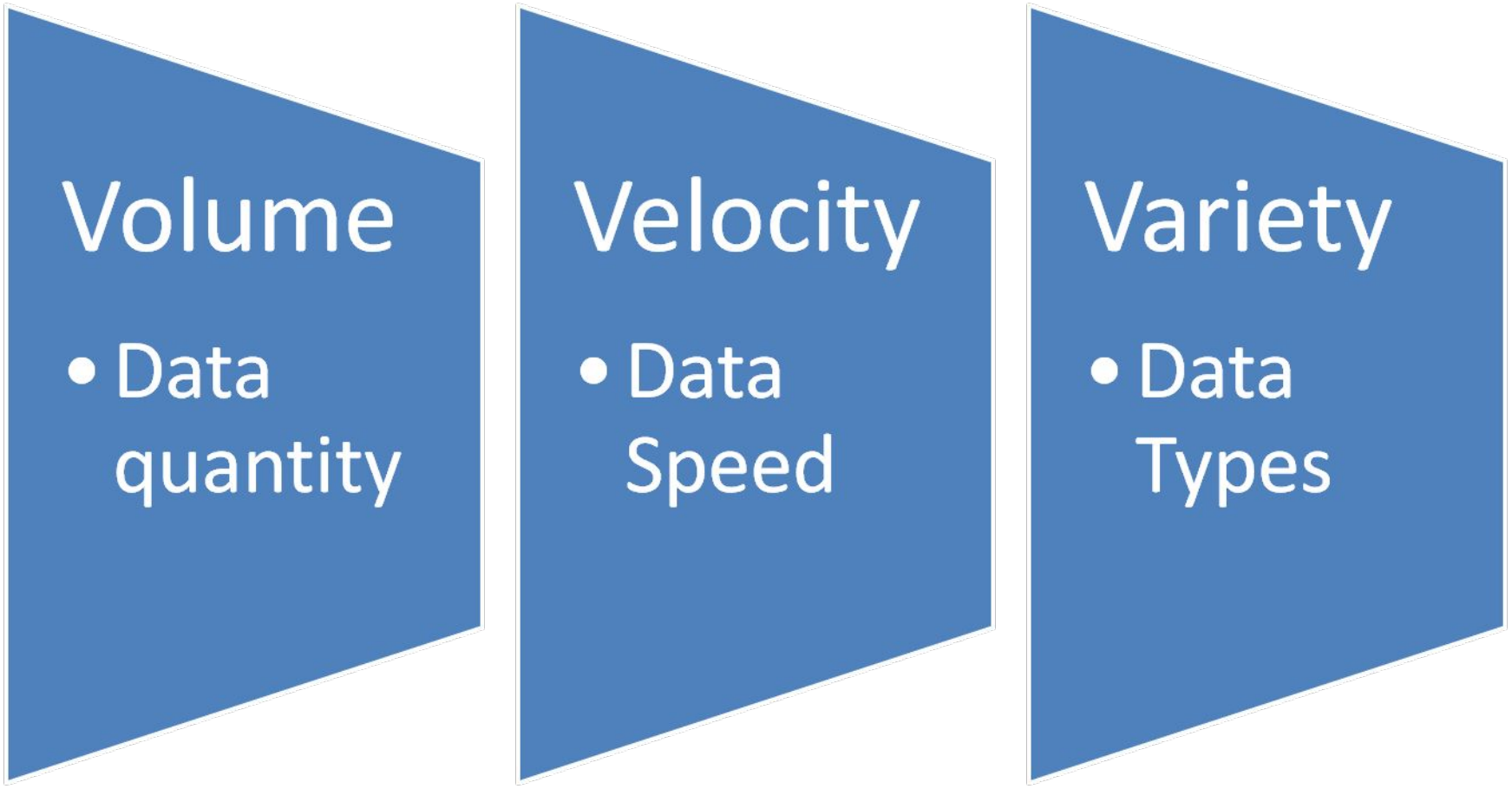
Why Big Data ?


Users
Application
Systems
Sensors
Large and growing files
(Big data files)

Why Big Data ?

- Growth of Big Data is needed
  - Increase of storage capacities
  - Increase of processing power
  - Availability of data(different data types)
  - Every day we create 2.5 quintillion bytes of data; 90% of the data in the world today has been created in the last two years alone

Three Characteristics of Big Data V3s



Volume

A zettabyte is 10^{21} bytes, or equivalently one thousand exabytes, one million petabytes, or one billion terabytes. (one Petabyte = 1024 Terabytes)

- From 0.8 zettabytes to 35zb within 2020.
- Data volume is increasing exponentially

Volume

- A typical PC might have had 10 gigabytes of storage in 2000.
- Today, Facebook ingests 500 terabytes of new data every day.
- Boeing 737 will generate 240 terabytes of flight data during a single flight across the US.
- The smart phones, the data they create and consume; sensors embedded into everyday objects will soon result in billions of new, constantly-updated data feeds containing environmental, location, and other information, including video.

Velocity

- Clickstreams and ad impressions capture user behavior at millions of events per second
- high-frequency stock trading algorithms reflect market changes within microseconds
- machine to machine processes exchange data between billions of devices
- infrastructure and sensors generate massive log data in real-time
- on-line gaming systems support millions of concurrent users, each producing multiple inputs per second.

Velocity

- Data is begin generated fast and need to be processed fast
- Online Data Analytics
- Late decisions missing opportunities

Examples

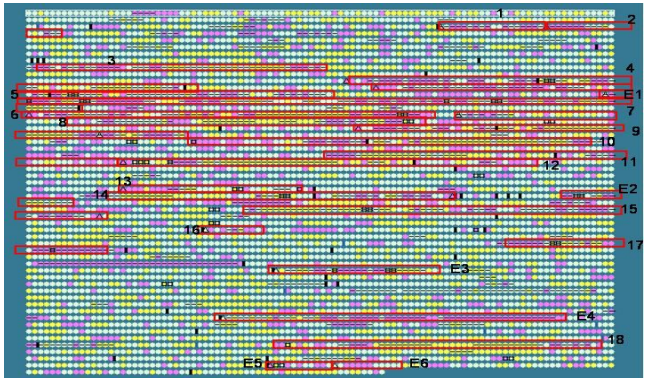
- **E-Promotions:** Based on your current location, your purchase history, what you like send promotions right now for store next to you

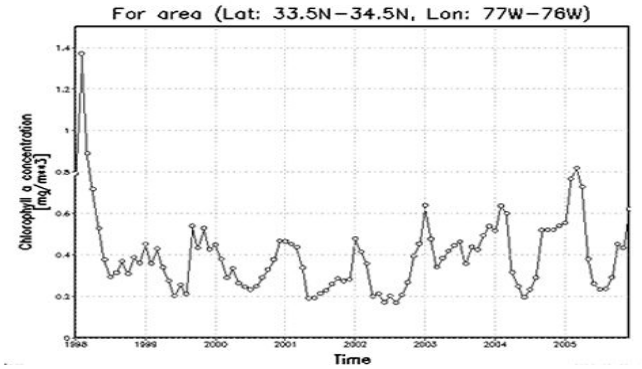
- **Healthcare monitoring:** sensors monitoring your activities and body any abnormal measurements require immediate reaction

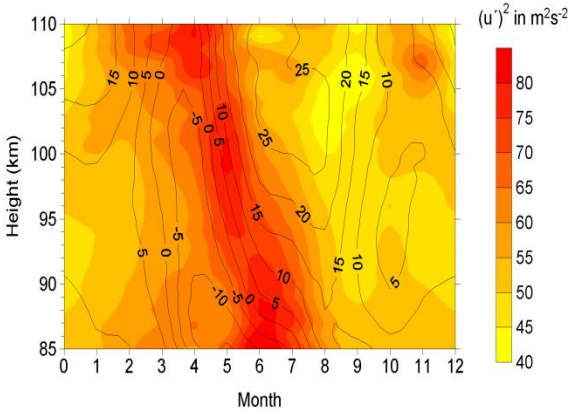
Variety

- Various formats, types, and structures
- Text, numerical, images, audio, video, sequences, time series, social media data, multi-dim arrays, etc…
- Static data vs. streaming data
- A single application can be generating/collecting many types of data

Variety

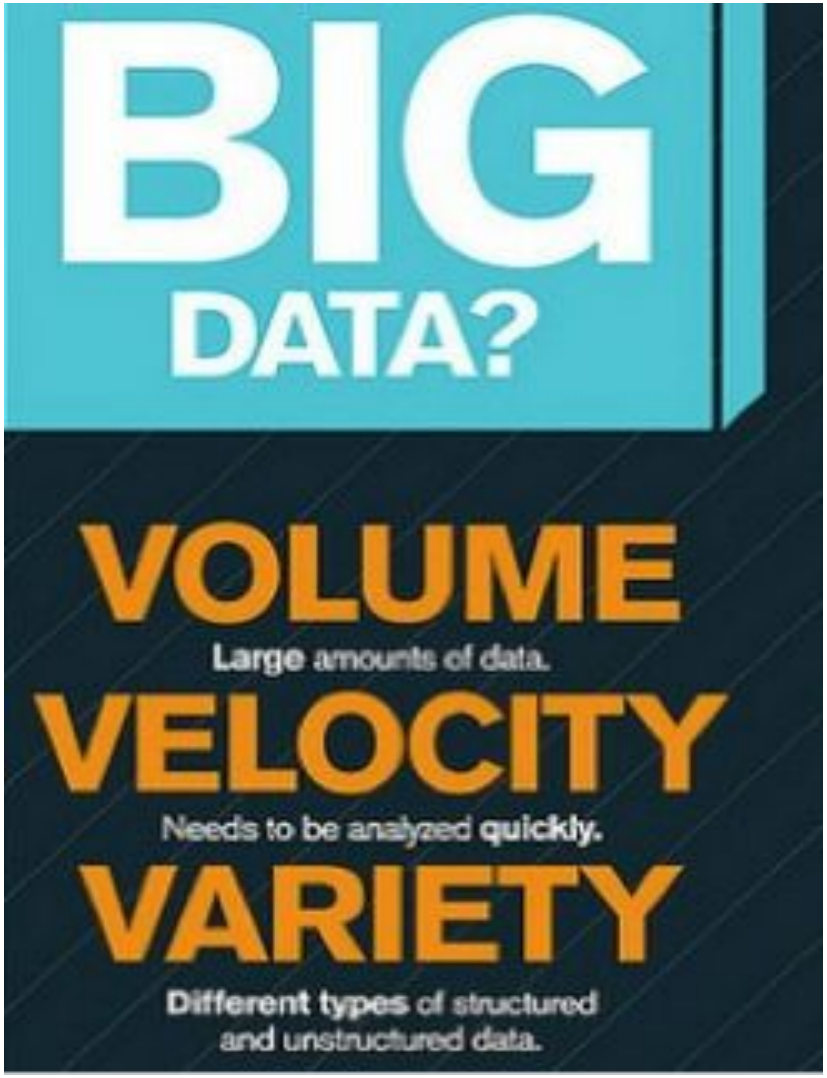


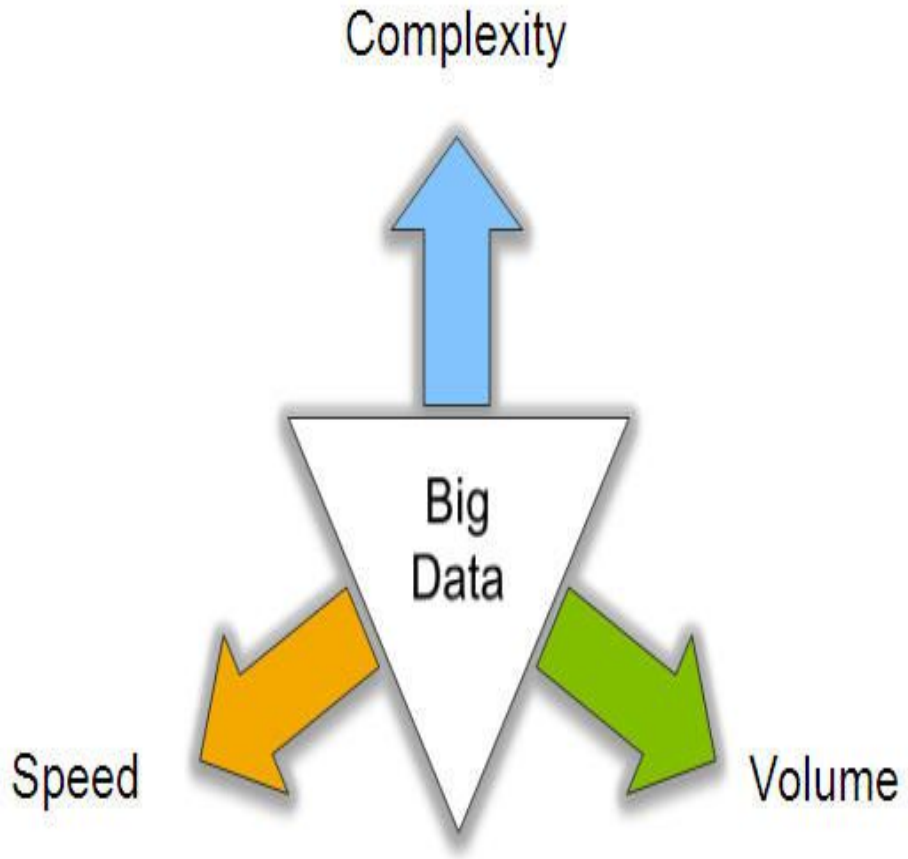




To extract knowledge all these types of data need to linked together

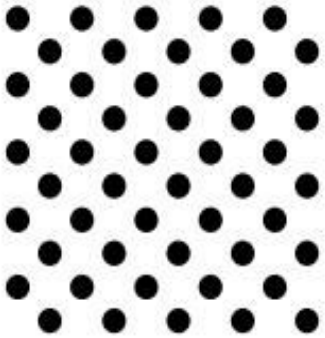
Big Data:3Vs


BIG
DATA?
VOLUME
Large amounts of data.
VELOCITY
Needs to be analyzed quickly.
VARIETY
Different types of structured
and unstructured data.


Complexity
Big
Data
Speed
Volume

4Vs

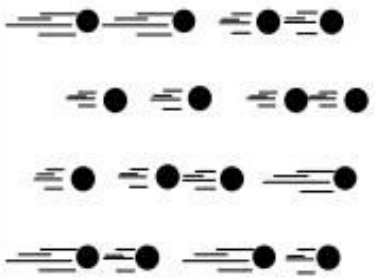
Volume



Data at Rest

Terabytes to exabytes of existing data to process

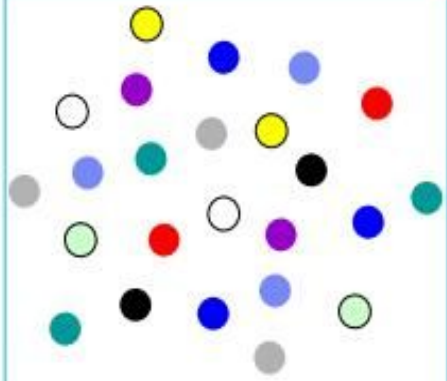
Velocity



Data in Motion

Streaming data, milliseconds to seconds to respond

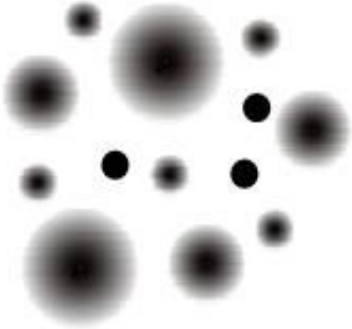
Variety



Data in Many Forms

Structured, unstructured, text, multimedia

Veracity*



Data in Doubt

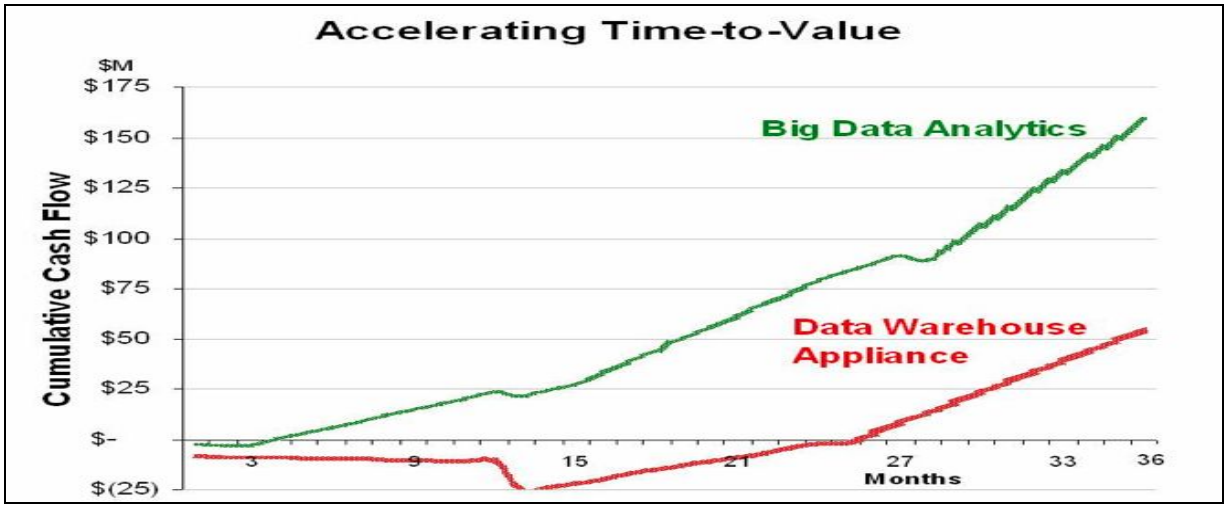
Uncertainty due to data inconsistency & incompleteness, ambiguities, latency, deception, model approximations

Big Data Analytics

- Examining large amount of data
- Appropriate information
- Identification of hidden patterns, unknown correlations
- Competitive advantage
- Better business decisions: strategic and operational
- Effective marketing, customer satisfaction, increased revenue

Big Data Analytics

- Big data is more real-time in nature than traditional DW applications
- Traditional DW architectures (e.g. Exadata, Teradata) are not well-suited for big data apps
- Shared nothing, massively parallel processing, scale out architectures are well-suited for big data apps



Application Of Big Data analytics

Smarter Healthcare

Multi-channel sales

Homeland Security

Telecom

Traffic Control

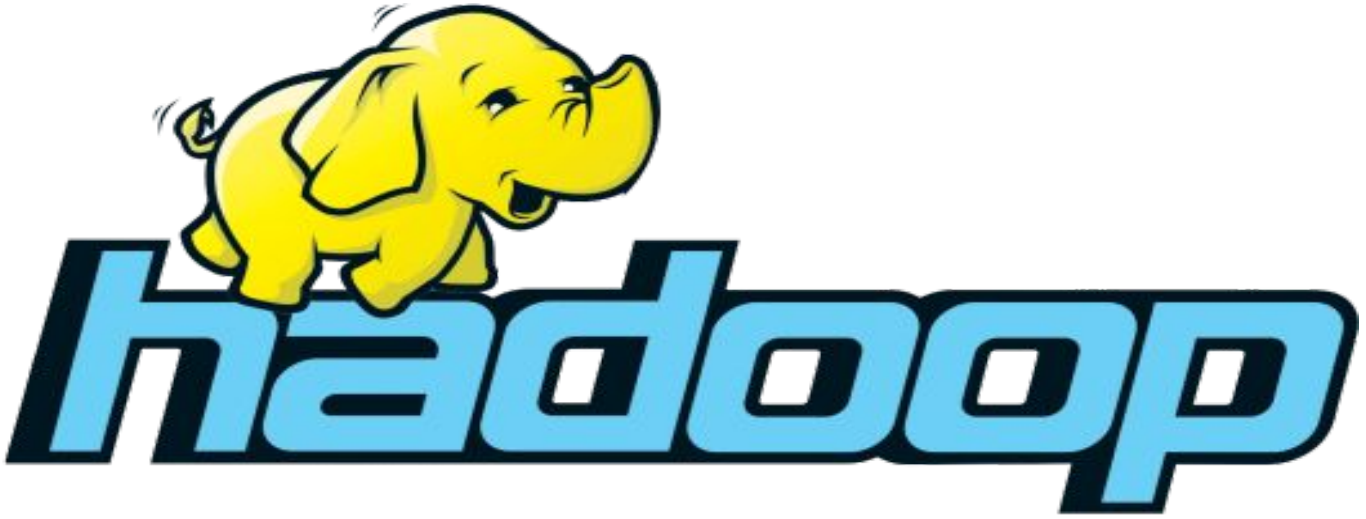
Trading Analytics

Manufacturing

Search Quality


THANKS

Hadoop



Hadoop Introduction

- Hadoop is analysis for Bigdata.
- It is an open-source framework for creating distributed applications that process huge amount of data.
- Huge means:
  - 25,000 machines
  - More than 10 clusters
  - 3 PB of data
  - 700+ users
  - 10,000+ jobs / week

Other Definition

- Hadoop is an open-source distributed, batch-processing and fault-tolerance system which is capable of sorting huge amount of data (TB, PB, ZettaByte) along with processing on the same amount of data.
- Easy to use parallel programming model.
- Consists two main core components
  - HDFS
  - MapReduce

History

- Hadoop was created by Doug Cutting and Mike Cafarella.
- Doug Cutting was working at Yahoo.
- He named this project as per his son's toy elephant.
- It was originally developed to support distribution for the Nutch search engine project.

Brief History

- 2004—implemented by Doug Cutting and Mike Cafarella.
  - December 2005—Nutch ported to the new framework. Hadoop runs reliably on 20 nodes.
  - January 2006—Doug Cutting joins Yahoo!.
  - February 2006—Apache Hadoop project officially started to support the standalone development of MapReduce and HDFS.
  - February 2006—Adoption of Hadoop by Yahoo! Grid team.
  - April 2006—Sort benchmark (10 GB/node) run on 188 nodes in 47.9 hours.

Brief History

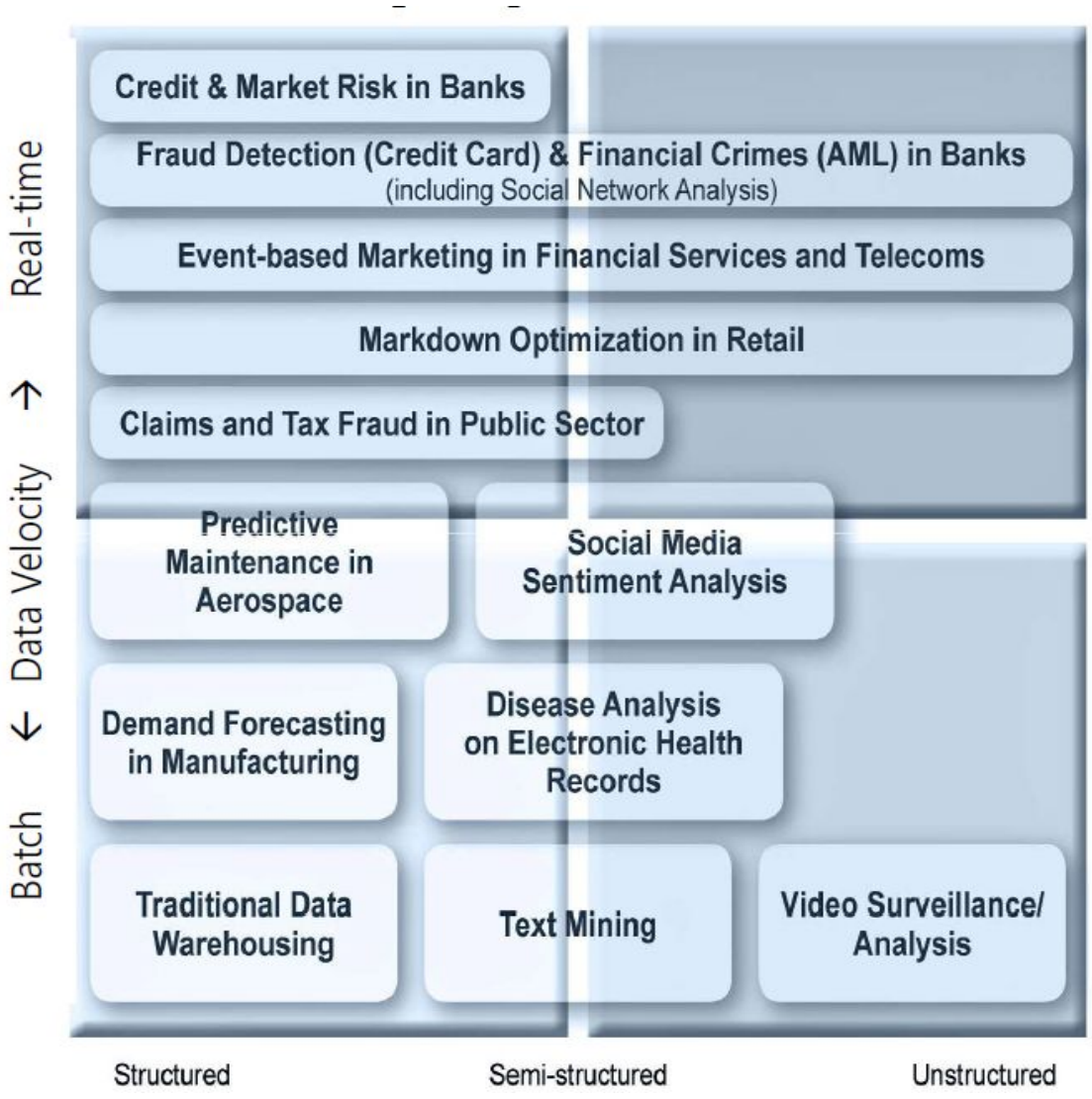
- May 2006—Yahoo! set up a Hadoop research cluster—300 nodes.
- May 2006—Sort benchmark run on 500 nodes in 42 hours (better hardware than April benchmark).
- October 2006—Research cluster reaches 600 nodes.
- December 2006—Sort benchmark run on 20 nodes in 1.8 hours, 100 nodes in 3.3 hours, 500 nodes in 5.2 hours, 900 nodes in 7.8 hours.
- January 2007—Research cluster reaches 900 nodes.
- April 2007—Research clusters—2 clusters of 1000 nodes.

Brief History

- April 2008—Won the 1 terabyte sort benchmark in 209 seconds on 900 nodes.
- October 2008—Loading 10 terabytes of data per day on to research clusters.
- March 2009—17 clusters with a total of 24,000 nodes.
- April 2009—Won the minute sort by sorting 500 GB in 59 seconds (on 1,400 nodes) and the 100 terabyte sort in 173 minutes (on 3,400 nodes).

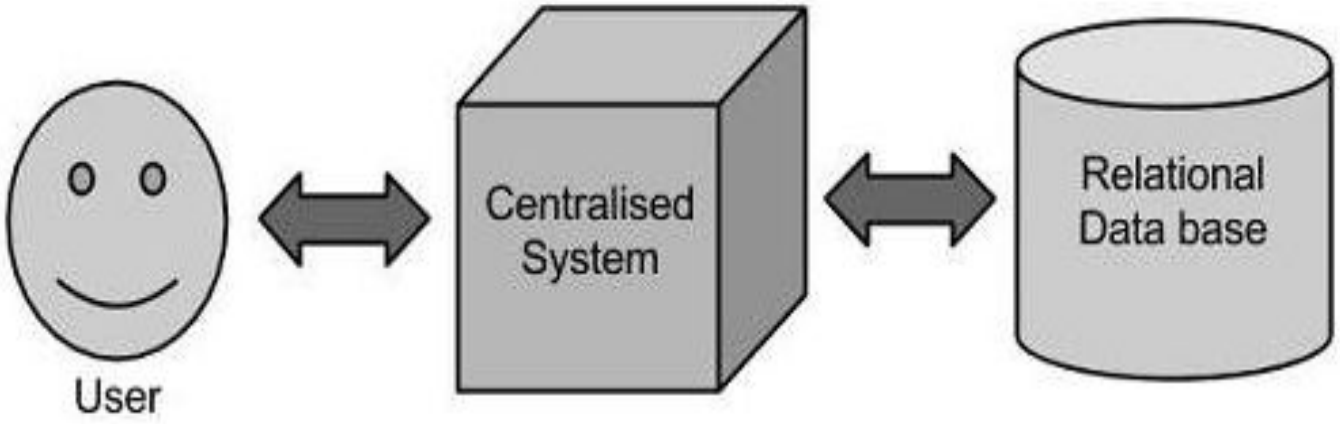
Data in Nature

Structured data :
Relational data.
Semi Structured data :
XML data.
Unstructured data :
Word, PDF, Text,
Media Logs.



How Hadoop is Different?

Traditional Approach

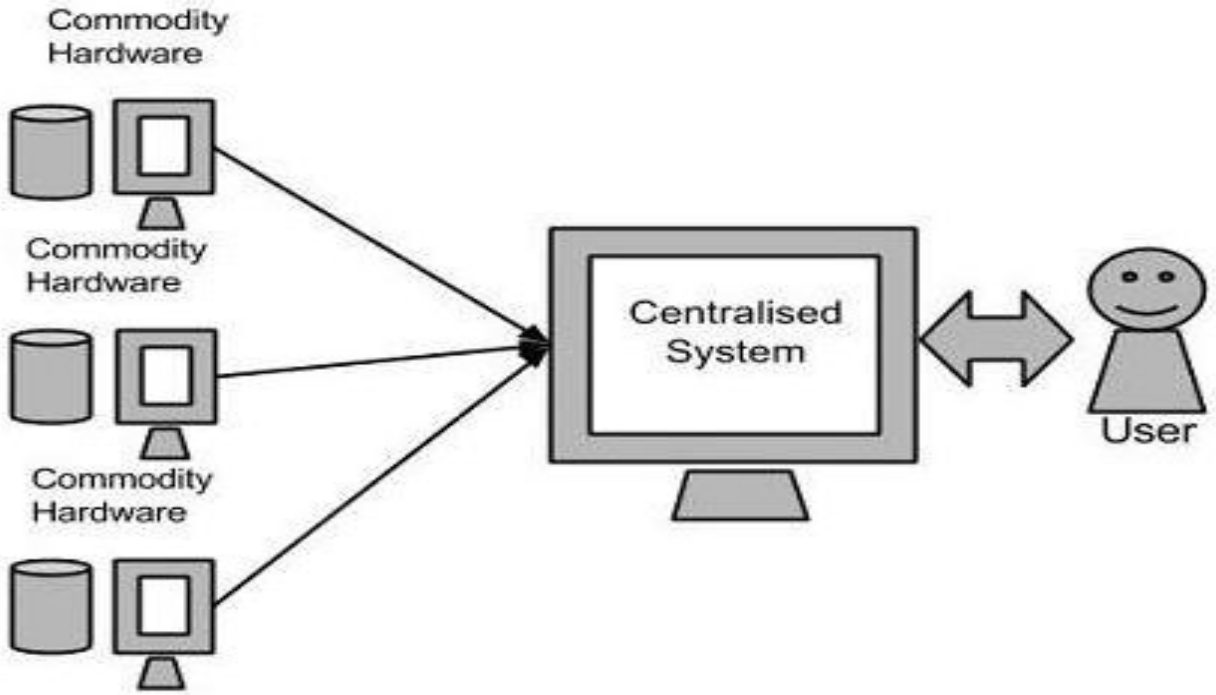


Here data will be stored in an RDBMS like Oracle Database, MS SQL Server or DB2 and sophisticated soft-wares can be written to interact with the database, process the required data and present it to the users for analysis purpose.

LIMITATIONS: This approach works well where we have less volume of data that can be accommodated by standard database servers, or up to the limit of the processor which is processing the data.

How Hadoop is Different?

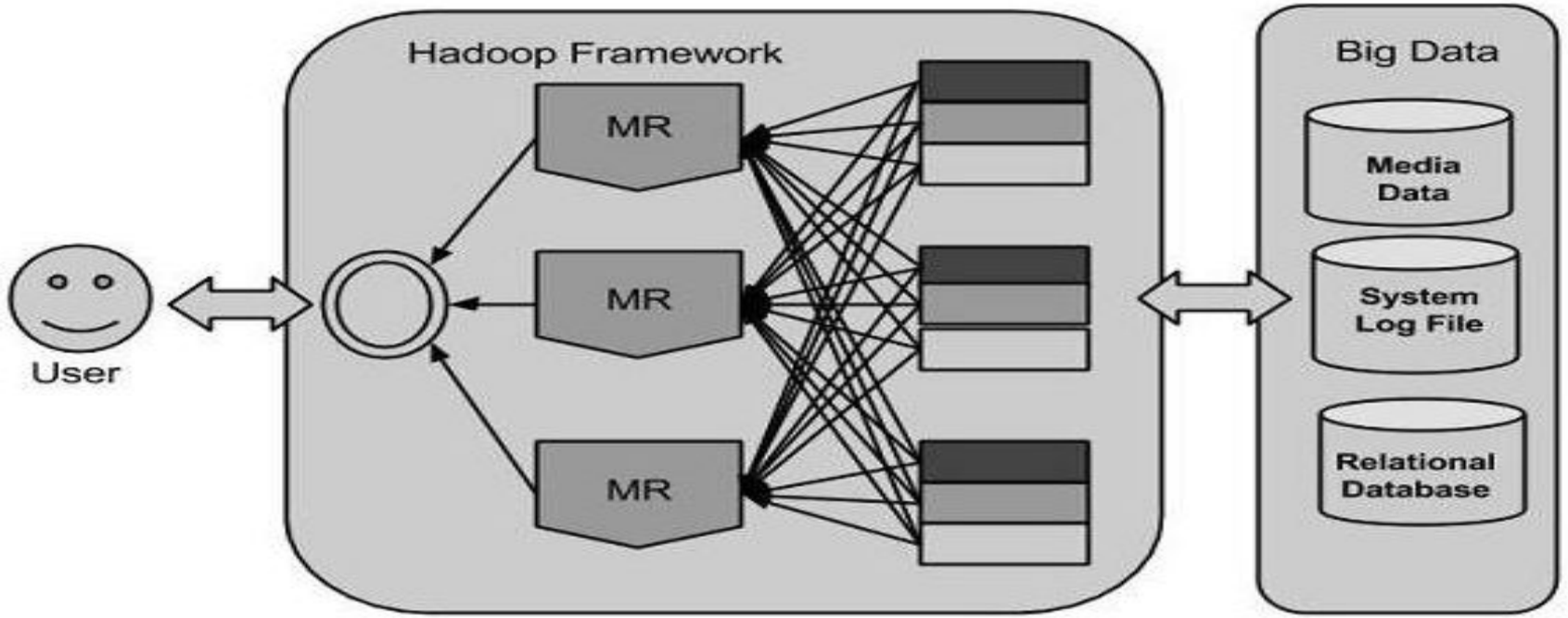
Google's Solution



Google solved this problem using an algorithm called MapReduce. This algorithm divides the task into small parts and assigns those parts to many computers connected over the network, and collects the results to form the final result dataset.

How Hadoop is Different?

Hadoop



Hadoop runs applications using the MapReduce algorithm, where the data is processed in parallel on different CPU nodes. In short, Hadoop framework is capable enough to develop applications capable of running on clusters of computers and they could perform complete statistical analysis for a huge amounts of data.

Advantages of Hadoop

Fast: In HDFS the data distributed over the cluster and are mapped which helps in faster retrieval. Even the tools to process the data are often on the same servers, thus reducing the processing time. It is able to process terabytes of data in minutes and Peta bytes in hours.

Scalable: Hadoop cluster can be extended by just adding nodes in the cluster.

Cost Effective: Hadoop is open source and uses commodity hardware to store data so it really cost effective as compared to traditional relational database management system.

Resilient to failure: HDFS has the property with which it can replicate data over the network, so if one node is down or some other network failure happens, then Hadoop takes the other copy of data and use it. Normally, data are replicated thrice but the replication factor is configurable.

Advantages of Hadoop

Flexible: Hadoop enables businesses to easily access new data sources and tap into different types of data (both structured and unstructured) to generate value from that data. This means businesses can use Hadoop to derive valuable business insights from data sources such as social media, email conversations or clickstream data, log processing, recommendation systems, data warehousing, market campaign analysis and fraud detection.

Components of Hadoop

- **Hadoop Common-** Pre-defined set of utilities and libraries that can be used by other modules within the Hadoop ecosystem.
- **Hadoop Distributed File System (HDFS) -** The default big data storage layer for Apache Hadoop is HDFS.
  - It is the "Secret Sauce" of Apache Hadoop components as users can dump huge datasets into HDFS and the data will sit there nicely until the user wants to leverage it for analysis.
- **MapReduce- Distributed Data Processing Framework-** MapReduce is a Java-based system created by Google where the actual data from the HDFS store gets processed efficiently.
  - MapReduce breaks down a big data processing job into smaller tasks.
  - MapReduce is responsible for the analysing large datasets in parallel before reducing it to find the results.

Components of Hadoop

- **YARN-** forms an integral part of Hadoop 2.0.
  - YARN is great enabler for dynamic resource utilization on Hadoop framework as users can run various Hadoop applications without having to bother about increasing workloads.
- **Data Access Components of Hadoop Ecosystem-**
  - **Pig-** is a convenient tools developed by Yahoo for analysing huge data sets efficiently and easily.
  - It provides a high level data flow language that is optimized, extensible and easy to use.
  - The most outstanding feature of Pig programs is that their structure is open to considerable parallelization making it easy for handling large data sets.

Components of Hadoop

- **Hive-** developed by Facebook is a data warehouse built on top of Hadoop and provides a simple language known as HiveQL similar to SQL for querying, data summarization and analysis.
- Hive makes querying faster through indexing.

❖ **Data Integration Components of Hadoop Ecosystem-**

- **Sqoop-** Sqoop component is used for importing data from external sources into related Hadoop components like HDFS, HBase or Hive.
- It can also be used for exporting data from Hadoop, other external structured data stores.
- Sqoop parallelized data transfer, mitigates excessive loads, allows data imports, efficient data analysis and copies data quickly.
- **Flume-** is used to gather and aggregate large amounts of data. Apache Flume is used for collecting data from its origin and sending it back to the resting location (HDFS).

Components of Hadoop

❖ **Data Storage Component of Hadoop Ecosystem-**

- **HBase –** HBase is a column-oriented database that uses HDFS for underlying storage of data.
- HBase supports random reads and also batch computations using MapReduce.

❖ **Monitoring and Management Components of Hadoop Ecosystem-**

- **Oozie-**is a workflow scheduler where the workflows are expressed as Directed Acyclic Graphs.
- Oozie runs in a Java servlet container Tomcat and makes use of a database to store all the running workflow instances, their states ad variables along with the workflow definitions to manage Hadoop jobs (MapReduce, Sqoop, Pig and Hive).

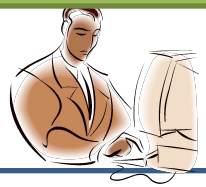
Components of Hadoop

- **Zookeeper-** Zookeeper is the king of coordination and provides simple, fast, reliable and ordered operational services for a Hadoop cluster.
- Zookeeper is responsible for synchronization service, distributed configuration service and for providing a naming registry for distributed systems.
- **AMBARI-** for RESTful API, mainly web user interface.
- **MAHOUT-** for machine learning.

Advantages of Hadoop

- Hadoop framework allows the user to quickly write and test distributed systems. It is efficient, and it automatic distributes the data and work across the machines and in turn, utilizes the underlying parallelism of the CPU cores.
- Hadoop does not rely on hardware to provide fault-tolerance and high availability (FTHA), rather Hadoop library itself has been designed to detect and handle failures at the application layer.
- Servers can be added or removed from the cluster dynamically and Hadoop continues to operate without interruption.
- Another big advantage of Hadoop is that apart from being open source, it is compatible on all the platforms since it is Java based.

Computational view of Bigdata



Data Visualization

Data Access

Data Analysis

Data Understanding

Data Integration

Formatting, Cleaning

Storage

Data

Topics related to Bigdata

Human-Computer Interaction

Data Visualization

Databases

Information Retrieval

Machine Learning

Data Access

Data Analysis

Data Mining

Computer Vision

Speech Recognition

Data Understanding

Data Integration

Natural Language Processing

Data Warehousing

Formatting, Cleaning

Signal Processing

Many Applications!

Storage

Data

Information Theory

HDFS

- Hadoop comes with a distributed file system called HDFS.
- In HDFS data is distributed over several machines and replicated to ensure their durability to failure and high availability to parallel application.

HDFS Concepts

- ✔**Blocks**
- ✔ **Name Node**
- ✔ **Data Node**

Block: A Block is the minimum amount of data that it can read or write.

- HDFS blocks are 128 MB by default and this is configurable.
- Files n HDFS are broken into block-sized chunks, which are stored as independent units.

HDFS Concepts

Name Node: HDFS works in master-worker pattern where the name node acts as master.

- Name Node is controller and manager of HDFS as it knows the status and the metadata of all the files in HDFS; the metadata information being file permission, names and location of each block.
- The metadata are small, so it is stored in the memory of name node,allowing faster access to data.
- Moreover the HDFS cluster is accessed by multiple clients concurrently,so all this information is handled by a single machine.
- The file system operations like opening, closing, renaming etc. are executed by it.

HDFS Concepts

Data Node: They store and retrieve blocks when they are told to; by client or name node.

They report back to name node periodically, with list of blocks that they are storing.

The data node being a commodity hardware also does the work of block creation, deletion and replication as stated by the name node.

Hadoop Models of Installation

- ✔**Standalone / local model-**After downloading Hadoop in your system, by default, it is configured in a standalone mode and can be run as a single java process.
- ✔ **Pseudo distributed model [Labwork]-**It is a distributed simulation on single machine. Each Hadoop such as hdfs, yarn, MapReduce etc., will run as a separate java process. This mode is useful for development.
- ✔ **Fully Distributed mode [Realtime]-**This mode is fully distributed with minimum two or more machines as a cluster.

Standalone / local model-

- Single machine
- No Daemons are running
- Everything runs in single JVM
- Standard OS storage
- Good for development and test with small data but will not catch all error.

Pseudo distributed model

- Single machine but cluster is simulated
- Daemons in separate JVM
- Separate JVMs in single node
- Good for development and Debugging

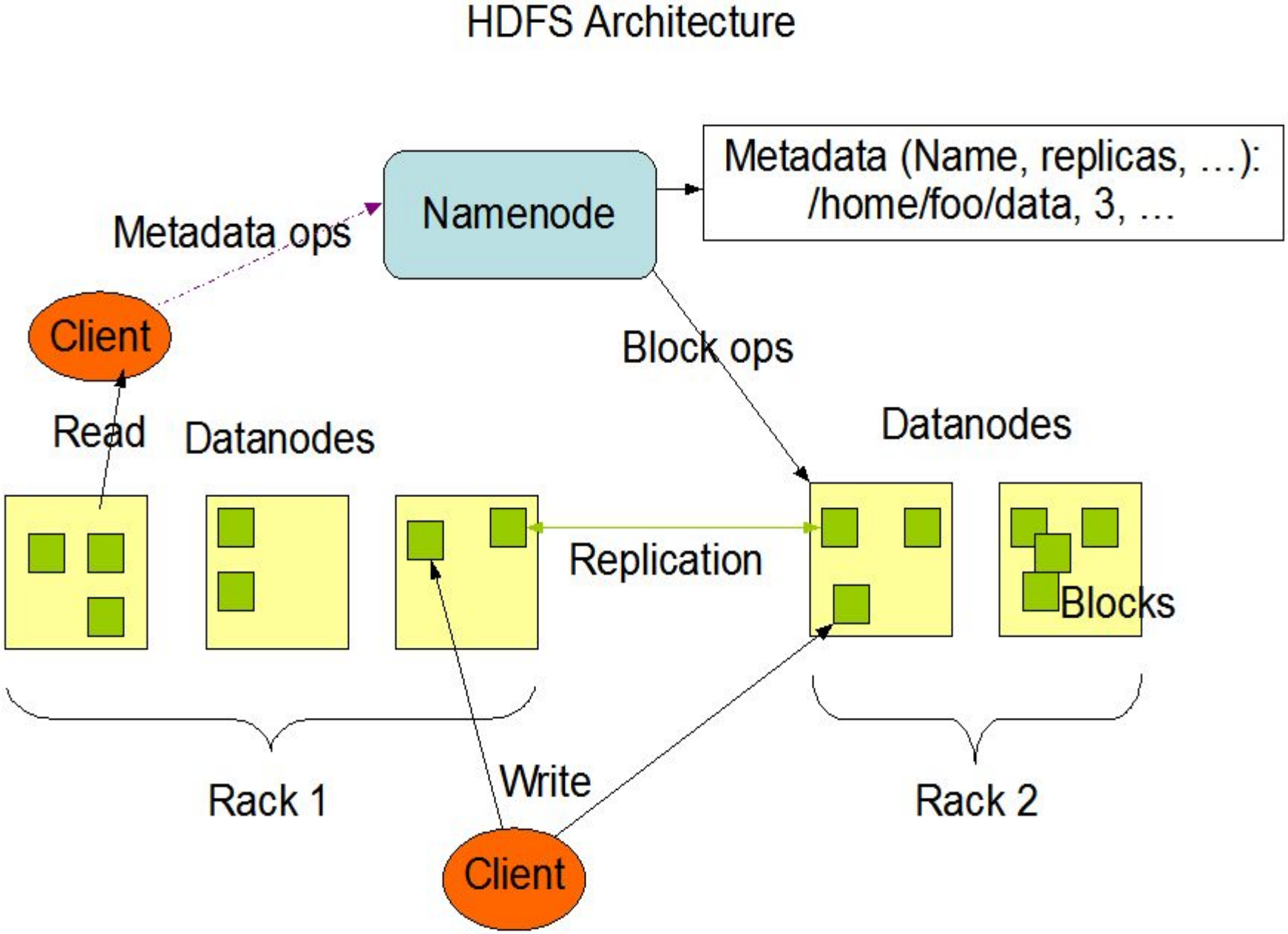
Full Distributed Mode

- Run Hadoop on cluster of machines
- Daemons run
- Production environment
- Good for staging and production

Hadoop Architeture

5 Daemons

- ✔NameNode
- ✔ DataNode
- ✔ JobTracker
- ✔ TaskTarcker
- ✔ Secondary NameNode


HDFS Architecture
Namenode
Metadata (Name, replicas, …):
/home/foo/data, 3, …
Metadata ops
Client
Block ops
Read
Datanodes
Datanodes
Replication
Blocks
Rack 1
Write
Rack 2
Client

Functionality of Daemons

Namenode

- ✔NameNode is the centerpiece of HDFS.
- ✔NameNode is also known as the Master
- ✔NameNode only stores the metadata of HDFS – the directory tree of all files in the file system, and tracks the files across the cluster.
- ✔NameNode does not store the actual data or the dataset. The data itself is actually stored in the DataNodes.
- ✔NameNode knows the list of the blocks and its location for any given file in HDFS. With this information NameNode knows how to construct the file from blocks.
- ✔NameNode is so critical to HDFS and when the NameNode is down, HDFS/Hadoop cluster is inaccessible and considered down.
- ✔NameNode is a single point of failure in Hadoop cluster.
- ✔NameNode is usually configured with a lot of memory (RAM). Because the block locations are help in main memory.

Functionality of Daemons

DataNode

- ✔DataNode is responsible for storing the actual data in HDFS.
- ✔DataNode is also known as the Slave
- ✔NameNode and DataNode are in constant communication.
- ✔When a DataNode starts up it announce itself to the NameNode along with the list of blocks it is responsible for.
- ✔When a DataNode is down, it does not affect the availability of data or the cluster. NameNode will arrange for replication for the blocks managed by the DataNode that is not available.
- ✔DataNode is usually configured with a lot of hard disk space. Because the actual data is stored in the DataNode.

Functionality of Daemons

Job Tracker –

- ✔JobTracker process runs on a separate node and not usually on a DataNode.
- ✔JobTracker is an essential Daemon for MapReduce execution in MRv1. It is replaced by ResourceManager/ApplicationMaster in MRv2. (MRv1=uses the JobTracker to create and assign tasks to data nodes, which can become a resource bottleneck when the cluster scales out far enough (usually around 4,000 nodes).)
- ✔JobTracker receives the requests for MapReduce execution from the client.
- ✔JobTracker talks to the NameNode to determine the location of the data.
- ✔JobTracker finds the best TaskTracker nodes to execute tasks based on the data locality (proximity of the data) and the available slots to execute a task on a given node.

Functionality of Daemons

Job Tracker –

- ✔JobTracker monitors the individual TaskTrackers and the submits back the overall status of the job back to the client.
- ✔JobTracker process is critical to the Hadoop cluster in terms of MapReduce execution.
- ✔When the JobTracker is down, HDFS will still be functional but the MapReduce execution can not be started and the existing MapReduce jobs will be halted.
- ✔ The JobTracker locates TaskTracker nodes with available slots at or near the data.
- ✔The JobTracker submits the work to the chosen TaskTracker nodes.
- ✔The TaskTrackerThe TaskTracker nodes are monitored. If they do not submit heartbeat signals often enough, they are deemed to have failed and the work is scheduled on a different TaskTracker.

Heart Beat Mechanism

- To process the data, Job Tracker assigns certain tasks to the Task Tracker.
- Let us think that, while the processing is going on one DataNode in the cluster is down.
- Now, NameNode should know that the certain DataNode is down , otherwise it cannot continue processing by using replicas.
- To make NameNode aware of the status(active / inactive) of DataNodes, each DataNode sends a ***"Heart Beat Signal"*** for every 10 minutes(Default).
- This mechanism is called as ***HEART BEAT MECHANISM.***

Functionality of Daemons

Job Tracker –

- ✔ Responsible for scheduling and rescheduling the tasks in the form of mapreduce job.
- ✔ Jobtracker will recide on top of namenode.

Functionality of Daemons

TaskTracker –

- ✔TaskTracker runs on DataNode. Mostly on all DataNodes.
- ✔TaskTracker is replaced by Node Manager in MRv2.
- ✔Mapper and Reducer tasks are executed on DataNodes administered by TaskTrackers.
- ✔TaskTrackers will be assigned Mapper and Reducer tasks to execute by JobTracker.
- ✔TaskTracker will be in constant communication with the JobTracker signalling the progress of the task in execution.
- ✔TaskTracker failure is not considered fatal. When a TaskTracker becomes unresponsive, JobTracker will assign the task executed by the TaskTracker to another node.

Functionality of Daemons

TaskTracker –

- ✔Responsible for instanting and maintaining indivisual map and reduce work.
- ✔Knowns as S/w daemon for Hadoop architeture.
- ✔The tasks executed by task tracker is called MRjobs.

Functionality of Daemons

Secondary NameNode:

- ✔Secondary NameNode in hadoop is a specially dedicated node in HDFS cluster whose main function is to take checkpoints of the file system metadata present on namenode. It is not a backup namenode.
- ✔It just checkpoints namenode's file system namespace.
- ✔The Secondary NameNode is a helper to the primary NameNode but not replace for primary namenode.
- ✔As the NameNode is the single point of failure in HDFS, if NameNode fails entire HDFS file system is lost. So in order to overcome this, Hadoop implemented Secondary NameNode whose main function is to store a copy of **FsImage** file and **edits** log file.
- ✔FsImage is a snapshot of the HDFS file system metadata at a certain point of time and EditLog is a transaction log which contains records for every change that occurs to file.

Functionality of Daemons

Secondary NameNode:

- ✔It is replaced by checkpoint node.
- ✔It acts as a separate physical file.
- ✔When namenode downs the secondary node comes to picture.
- ✔It never takes all the functionalities of namenode.
- ✔It only reads fsImage space and edit log.

What is Checkpoint Mechanism ?

It is the mechanism used in hadoop cluster to make sure all the metadata information of hadoop file system will get updated in the two persistence files.

- ✔ fsnamespace Image
- ✔ edit log

Checkpoint mechanism can be configured either hourly, daily and weekly.

What is Speculative execution of Hadoop?

If a datanode fails to response within 10mins, the namenode will mark that node either it is dead, running slow or functionally slow.

Immediately the jobtracker will assign the same task to some other ideal nodes within hadoop cluster. This is know as Speculative execution of Hadoop.

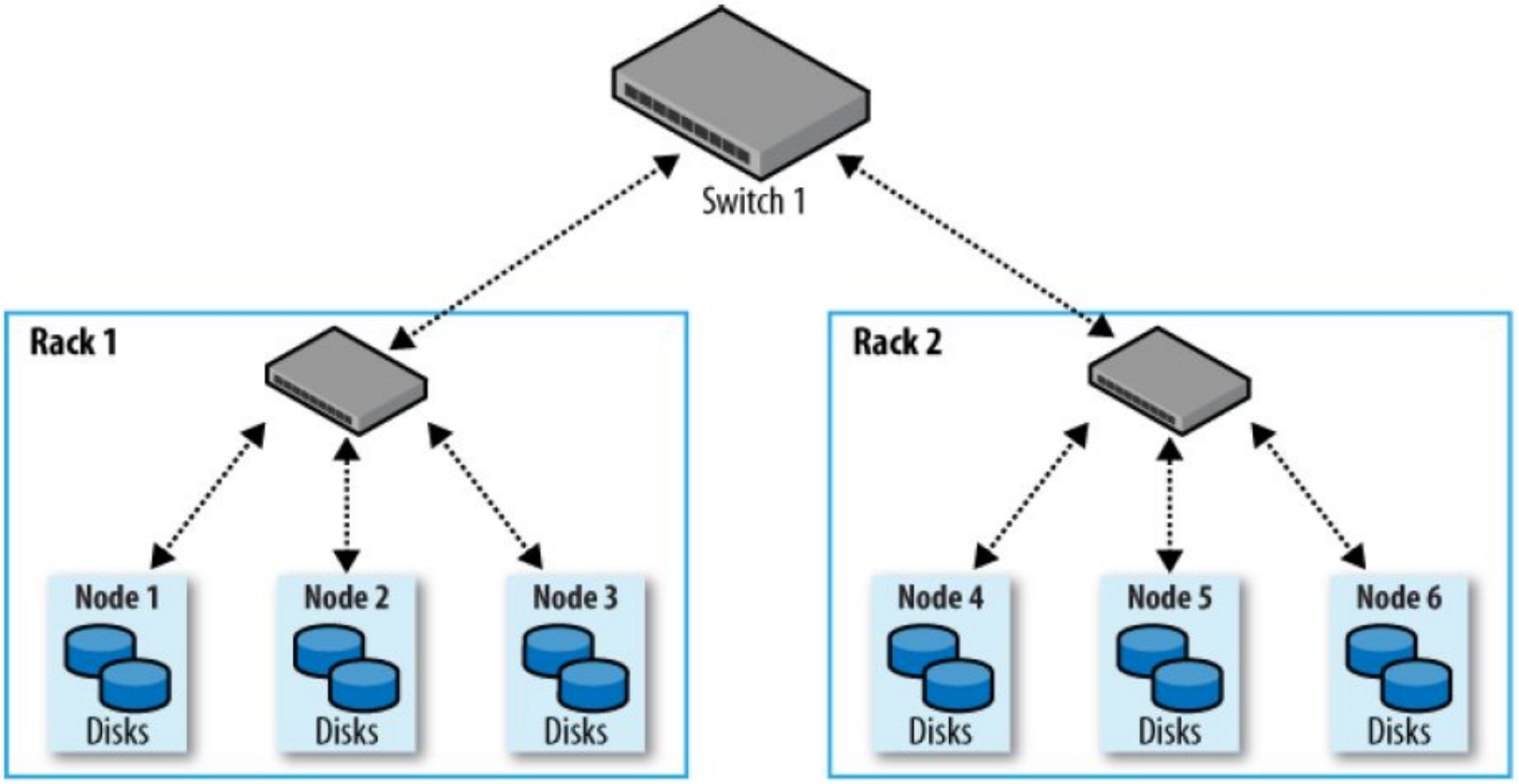
It does not allow to feel any deal for the end user.


THANKS

Network Topology

- Hadoop cluster architecture consists of a two-level network topology.
- Typically there are 30 to 40 servers per rack, with a 1 GB switch for the rack.
- An uplink to a core switch or router (which is normally 1 GB or better).

Two-level network topology



Rack awareness

- To get maximum performance out of Hadoop, it is important to configure Hadoop so that it knows the topology of your network.
- If your cluster runs on a single rack, then there is nothing more to do, since this is the default.
- For multirack clusters, you need to map nodes to racks.
- By doing this, Hadoop will prefer within-rack transfers (where there is more bandwidth available) to off-rack transfers when placing MapReduce tasks on nodes.
- Network locations such as nodes and racks are represented in a tree, which reflects the network "distance" between locations.

The namenode uses the network location when determining where to place block replicas, the MapReduce scheduler uses network location to determine where the closest replica is as input to a map task.

Rack awareness

- In network the rack topology is described by two network locations, we can present */switch1/rack1 and /switch1/rack2.*
- For the figure we can present the locations as*/rack1 and /rack2.*
- The Hadoop configuration must specify a map between node addresses and network locations. The map is described by a Java interface, DNSToSwitchMapping.
- The topology.node.switch.mapping.impl configuration property defines an implementation of the DNSToSwitchMapping interface that the namenode and the jobtracker use to resolve worker node network locations.
- The *names* parameter is a list of IP addresses, and the return value is a list of corresponding network location strings.

Parallel Copying with distcp

- So far we learned HDFS access is on single-threaded access.
- For parallel processing of files you would have to write a program yourself.
- Hadoop comes with a useful program called *distcp for copying large amounts of data* to and from Hadoop filesystems in parallel.

% **hadoop distcp hdfs://namenode1/foo hdfs://namenode2/bar**

This will copy the */foo directory (and its contents) from the first cluster to the /bar* directory on the second cluster, so the second cluster ends up with the directory structure */bar/foo.*

If */bar doesn't exist, it will be created first. You can specify multiple source* paths, and all will be copied to the destination.

Parallel Copying with distcp

- There are more options to control the behavior of *distcp, including ones to preserve file* attributes, ignore failures, and limit the number of files or total data copied.
- *distcp is implemented as a MapReduce job where the work of copying is done by the* maps that run in parallel across the cluster.
- There are no reducers. Each file is copied by a single map, and *distcp tries to give each map approximately the same amount of* data, by bucketing files into roughly equal allocations.
- If you try to use *distcp between two HDFS clusters that are running different versions,* the copy will fail if you use the *hdfs protocol, since the RPC systems are incompatible.*
- To overcome this you can use the read-only HTTP-based HFTP filesystem to read from the source.

THANK YOU