Big Data
Hadoop

Introduction

Contents

- Big Data – Definitions, Significance
- Hadoop – Architecture
- HDFS – Overview

What is Big Data

- A catch-phrase used to describe a massive volume of both structured and unstructured data that is so large that it's difficult to process using traditional database and software techniques.
- "*Big data is high volume, high velocity, and high variety information assets that require new forms of processing to enable enhanced decision making, insight, discovery, and process optimization.*"

- Amazon was averaging about 26.5 million transactions per day, over a million transactions per hour that translates into roughly 2.5 peta bytes an hour.
- Flipkart was averaging 45,000 transactions a day.
- Google processes 20 PB's of information per day.

Why – Big Data

- Businesses have thrived on their ability to derive insights from information or data.
- It has helped them make better, smarter, real time, and fact-based decisions.
- This demand for intensity of knowledge is what has fueled the growth of “big data.”
- Data has grown exponentially with the advent of social media and mobile technology.
- There is hardly any field that is not likely to be benefited by “big data” initiatives.
- *Big data* is definitely expected to lay the foundation for advancements in health care, medicine, data analysis, scientific research, education, energy, transportation, financial services, retail, and the telecommunications industry.

Why – Big Data

- **Competitive advantage:**
  - Amazon has better recommendation system because of user data.
  - For LinkedIn, Facebook – analysis of networking data is the competitive advantage.
- **Better User Experience:**
  - Telecom services improve network by analysis call data
  - Facebook/Whatsapp improves product by analyzing logs
  - Companies prefer implicit feedback rather than explicit feedback
  - Less churn – Predicting customer loyalty
- **Necessity:**
  - Flipkart sold 1 lakh moto-e sets in a few minutes
  - Companies need to be prepared from day-1
  - A new application can go viral overnight, users increase from a few dozens to thousands or millions

Why – Big Data

- **RDBMS**
  - Highly structured data
  - Organized into flat tables of rows and columns
  - Normalized and relationships defined across tables
- **Drawbacks**
  - Scaling
  - Supporting large number of concurrent users
  - Performance slows down and response times deteriorate
  - Vertical scaling is possible and not designed for horizontal scaling
  - Variety aspect of data – Cannot handled semi-structured and unstructured data

Hadoop – History

- In 2002-04 Doug Cutting with Mike Cafarella started working on Nutch – an open source project
- In 2003-04 Google published papers on Google File System and MapReduce which are implemented in Nutch
- In 2006 they formed Hadoop project; around the same time Doug Cutting joined Yahoo! which provided him a team for full-fledged development of Hadoop
- In January 2008, Hadoop was made its own top-level project at Apache

What is Hadoop?

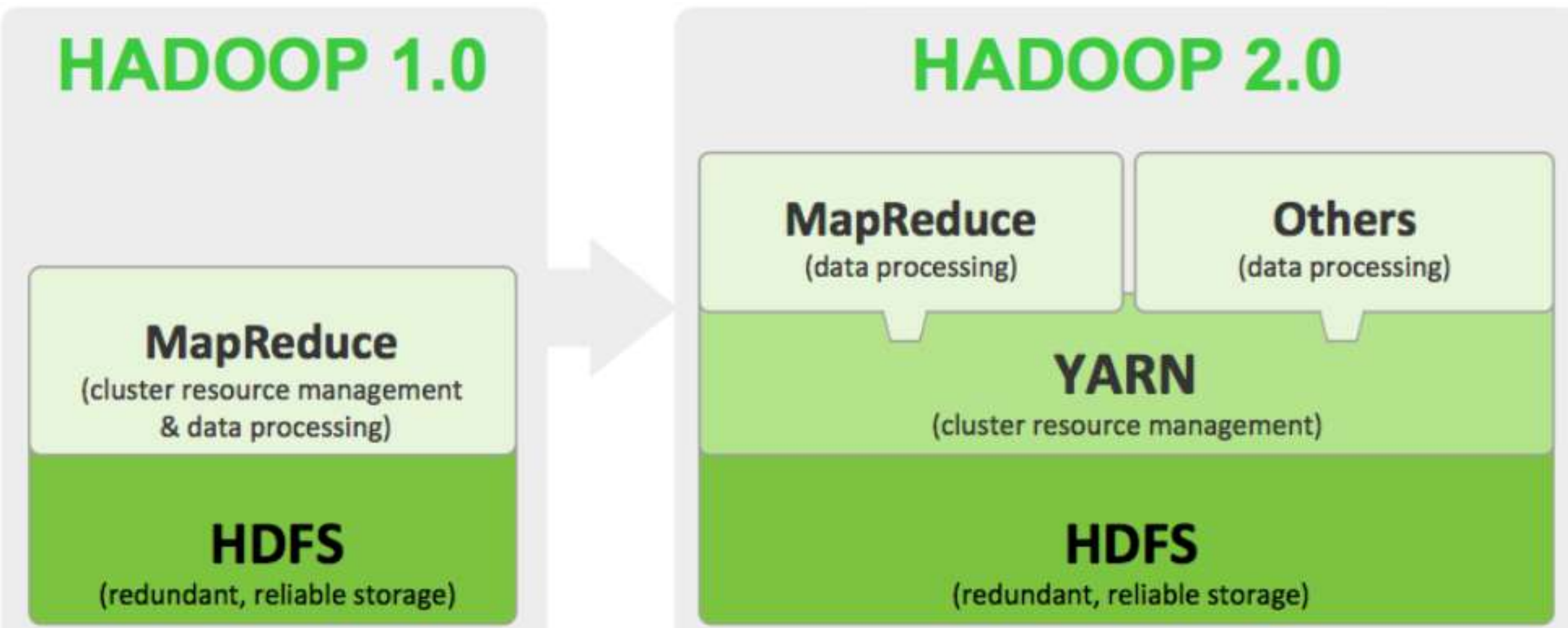
- Apache Hadoop is a software framework that allows for the distributed processing of large data sets across clusters of computers using simple programming models.
- Designed to scale up from single servers to thousands of machines, each offering local computation and storage.
- Designed to detect and handle failures at the application layer, so delivering a highly-available service on top of a cluster of computers, each of which may be prone to failures.
- Hadoop provides – a reliable, scalable platform for storage and analysis of large amounts of data on clusters of commodity machines.
- These machines typically run Linux operating system (OS).
- Built using the Java language; any cluster of machines that supports Java can run Hadoop.

What is Hadoop?

- Components of Hadoop
  - Hadoop 1.0
    - HDFS – Distributed storage
    - MapReduce – Distributed processing
  - Hadoop 2.0
    - HDFS - Distributed storage
    - Distributed Processing
      - YARN - Resource management
      - MapReduce - Distributed computation

What is Hadoop?

- Hadoop 1.0 and Hadoop 2.0



Hadoop Distributed File System (HDFS)

- HDFS is a highly fault-tolerant file system designed to run on clusters of commodity hardware.
- Provides high throughput access to application data and is suitable for applications that have large data sets.
- HDFS is designed more for batch processing rather than interactive use by users.
- HDFS is built for write-once, read-many-times pattern of data processing.
- Files in HDFS are divided into *blocks* and written across the cluster nodes by a single writer. Writes are always made at the end of the file, in append-only fashion.
- Each block is 128 MB by default; block size is configurable.
- Every block is replicated 3 times by default on different nodes of the cluster. Replication factor is configurable.
- Replication provides fault-tolerance to the system.

HDFS

- HDFS has a master/slave architecture.
- On master node runs the background process **NameNode**. On slave nodes the background called **DataNode** runs.
- The NameNode and DataNode are pieces of software designed to run on commodity machines. These machines typically run a GNU/Linux operating system (OS).
- Master server or *NameNode* manages the file system namespace and regulates access to files by clients.
- **DataNode** which are the workhorses of the cluster stores the actual data which is divided into blocks.
- HDFS supports a traditional hierarchical file organization. A user or an application can create directories and store files inside these directories.
- The file system namespace hierarchy is similar to other existing file systems; one can create and remove files, move a file from one directory to another, or rename a file.
- HDFS supports access permissions and user quotas.

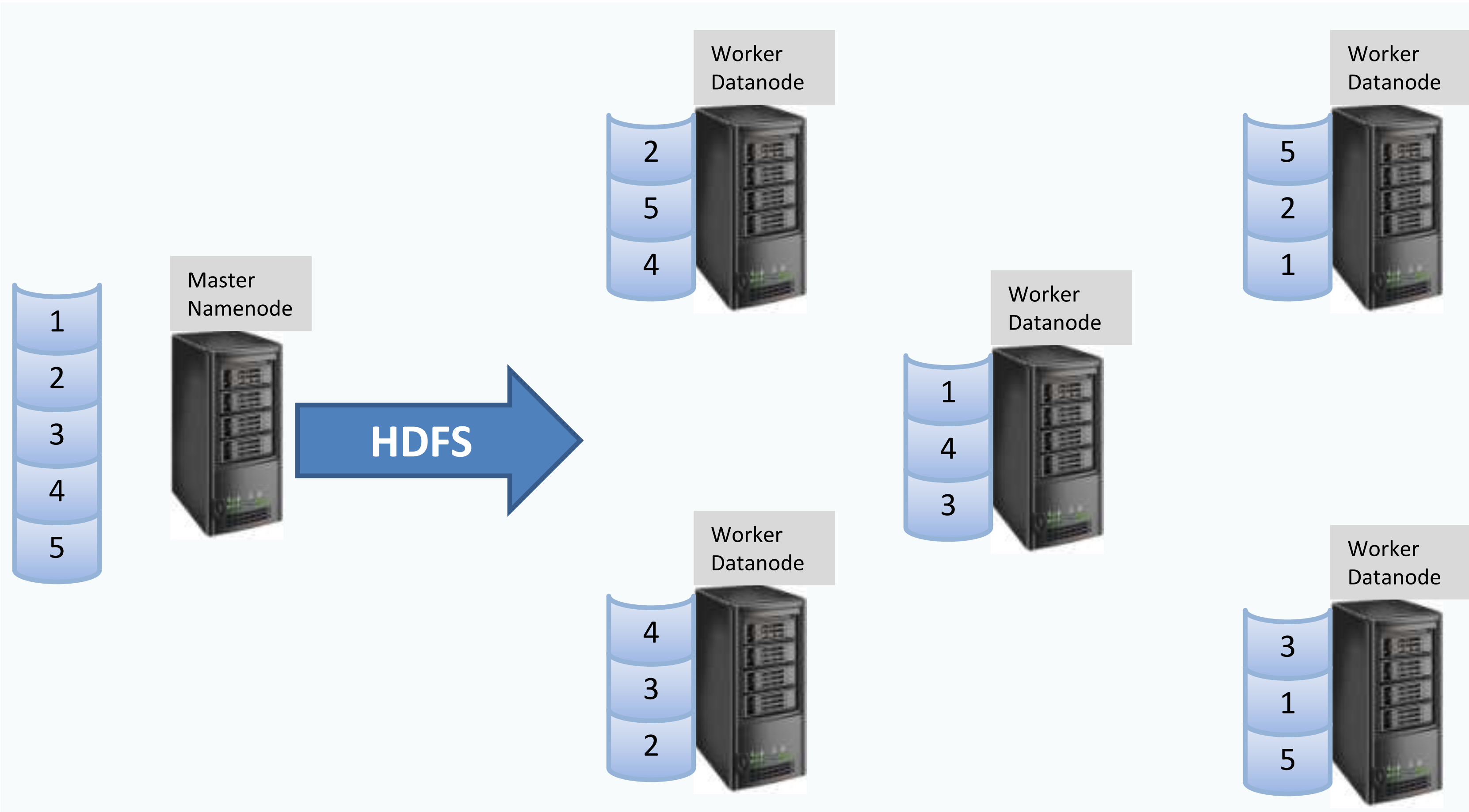
HDFS

- NameNode maintains the file system tree and the metadata for all the files and directories in the tree.
- This information is stored persistently on the local disk in the form of two files: the namespace image and the edit log.
- The namenode also knows the datanodes on which all the blocks for a given file are located; however, it does not store block locations persistently, because this information is reconstructed from datanodes when the system starts.
- Executes file system namespace operations like opening, closing, and renaming files and directories.
- It determines mapping of blocks to DataNodes.

HDFS

- **Datanodes** are the work-horses of the filesystem and store the actual data.
- Responsible for serving read and write requests from the file system's clients.
- Also perform block creation, deletion, and replication upon instruction from the NameNode.
- Periodically send a Heartbeat and a Blockreport to NameNode.
- Heartbeat indicates that the DataNode is functioning properly. A Blockreport contains a list of all blocks on a DataNode.
- A client accesses the filesystem on behalf of the user by communicating with the namenode and datanodes.

Hadoop Distributed File System (HDFS)



HDFS

- A *secondary namenode,* is run whose main role is to periodically merge the namespace image with the edit log to prevent the edit log from becoming too large.
- The secondary namenode usually runs on a separate physical machine because it requires plenty of CPU and as much memory as the namenode to perform the merge. It keeps a copy of the merged namespace image, which can be used in the event of the namenode failing.
- The namenode is still a *single point of failure* (SPOF) in Hadoop 1.0
- Each cluster had a single NameNode, and if that machine or process became unavailable, the cluster as a whole would be unavailable until the NameNode was either restarted or brought up on a separate machine.
- To recover from a failed namenode in this situation, an administrator starts a new primary namenode with one of the filesystem metadata replicas and configures datanodes to use this new namenode.

HDFS

- Prior to Hadoop 2.0 NameNode was a Single Point Of Failure. This has been remedied in Hadoop 2.0 and above by adding support for HDFS High Availability.
- HDFS High Availability feature provides the option of running two (or more, as of Hadoop 3.0.0) redundant NameNodes in the same cluster in a standby configuration.
- Datanodes send block reports to both namenodes because the block mappings are stored in a namenode's memory, and not on disk.
- The namenodes use highly available shared storage to share the edit log.
- There are two choices for the highly available shared storage: an NFS filer, or a *quorum journal manager (QJM).*
- If the active namenode fails, the standby can take over very quickly (in a few tens of seconds) because it has the latest state available in memory: both the latest edit log entries and an up-to-date block mapping.
- Automatic failover is enabled by two additional components – Zookeeper quorum and ZooKeeper Failover Controller which monitor the health of NamaNodes and elect another Namenode as failover alternative if the active one goes down.

HDFS – Commands

Web UI:

```
https://<ip address>:50070
```

```
$ hadoop version
$ hadoop fs -ls /
$ hadoop fs -ls
$ hadoop fs -mkdir <directory name>
$ hadoop fs -copyFromLocal <filename>
$ hadoop fs -put <directory>/<file name>
$ hadoop fs –copyToLocal <file name>
$ hadoop fs –get <directory>/<file name>
$ hadoop dfs -cat test2.txt
$ hadoop dfs -tail test3.txt ➔ Displays last KB of file
```

HDFS – Commands

```
$ hadoop fs -rm <file name>
$ hadoop fs -rm -r <directory>

$ hadoop fs -mv <dir1/filename1> <dir2/filename2>
$ hadoop fs -cp <dir1/filename1> <dir2/filename2>

$ hadoop fs -expunge
```

`$ hadoop fs -chmod a+r test3.txt` → Give read permissions to all for the file test3.txt

`$ hadoop fs -chmod go-w test3.txt` → Remove write permissions to group and others for the file test3.txt

`$ hdfs fsck <directory> / -files` → Checks and prints out files being checked

`$ hdfs fsck <directory> / -blocks` → Checks and prints out blocks being checked

HDFS – Commands

$ hadoop fs -du <directory/file name> → Check disk usage i.e. how much space this directory occupies in HDFS

$ hadoop fs -count hdfs:/ → Count the number of directories, files and bytes under the specified path

$ hadoop fs -df hdfs:/ → Report the amount of space used and available on currently mounted filesystem

$ hdfs fsck /user/hduser → Runs a HDFS filesystem checking utility

$ sudo -uhdfs hdfs dfsadmin -report → Displays the list of DataNodes

$ sudo -uhdfs hdfs dfsadmin -safemode enter → Moves the cluster into safemode which is a read-only mode. `Leave` option will move it back to normal mode.

$ hadoop fs -Ddfs.block.size=134217728 -put <filename1> <filename2> → Copy to HDFS with block size of 128MB

$ hadoop fs -Ddfs.block.size=67108864 -put <filename1> <filename2> → Copy to HDFS with block size of 64MB

$ hadoop fs -setrep 2 <filename> → Copy to HDFS with replication factor as 2. Using option –w will request the command to wait for replication to complete. Can take a long time.

$ hadoop fs -stat "%n %o %r" <filename> → Displays the name, block size and replication factor of the file.