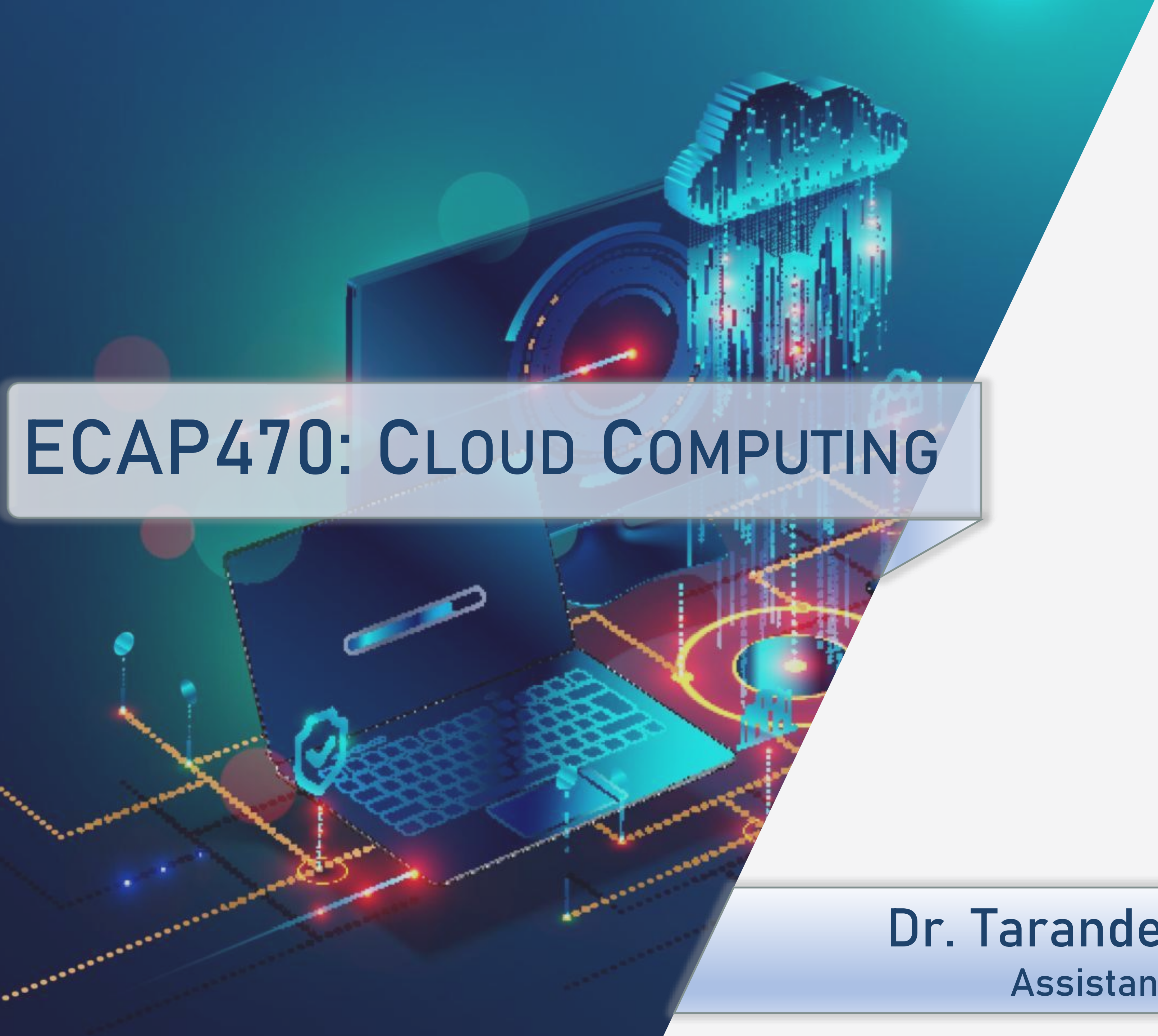


ECAP470: Cloud Computing

Dr. Tarandeep Kaur
Assistant Professor

Learning Outcomes

After this lecture, you will be able to,

✓ learn about the hadoop distributed file system (hdfs).

✓ know the architecture and operations of hdfs.

✓ explore the comparison of gfs and hdfs.

Distributed File System (DFS)

- A set of client and server services that allow an organization using Microsoft Windows servers to organize many distributed SMB file shares into a distributed file system.
- Two components to its service: Location transparency (via the namespace component) and redundancy (via the file replication component).

Distributed File System (DFS)

Microsoft's DFS is referred to interchangeably as 'DFS' and 'Dfs' by Microsoft and is unrelated to the DCE Distributed File System, which held the 'DFS' trademark but was discontinued in 2005.

Distributed File System (DFS)

- DFS root can only exist on a server version of Windows (from Windows NT 4.0 and up) and OpenSolaris (in kernel space) or a computer running Samba (in user space.)
- Enterprise and Datacenter Editions of Windows Server can host multiple DFS roots on the same server.

Distributed File System (DFS)

Two ways of implementing DFS on a server:

Standalone DFS Namespace.

Domain-based DFS Namespace.

Hadoop Distributed File System (HDFS)

- HDFS is a distributed file system that handles large data sets running on commodity hardware.
- Used to scale a single Apache Hadoop cluster to hundreds (and even thousands) of nodes.

Hadoop Distributed File System (HDFS)

- One of the major components of Apache Hadoop, the others being MapReduce and YARN.
- HDFS should not be confused with or replaced by Apache HBase, which is a column-oriented non-relational database management system that sits on top of HDFS and can better support real-time data needs with its in-memory processing engine.

Features of HDFS

- Suitable for the distributed storage and processing.
- Hadoop provides a command interface to interact with HDFS.
- The built-in servers of namenode and datanode help users to easily check the status of cluster.
- Streaming access to file system data.
- Provides file permissions and authentication.

HDFS Goals

Goals of HDFS:

- Fast Recovery from Hardware Failures.
- Access to Streaming Data.
- Accommodation of Large Data Sets.
- Portability.

HDFS ASSUMPTIONS

- Simple Coherency Model.
- Moving Computation is Cheaper than Moving Data.

HDFS Key Features

HDFS Key Features	Description
Storing bulks of data	HDFS is capable of storing terabytes and petabytes of data.
Minimum intervention	It manages thousands of nodes without operators’ intervention.
Computing	HDFS provides the benefits of distributed and parallel computing at once.
Scaling out	It works on scaling out, rather than on scaling up, without a single downtime.
Rollback	HDFS allows returning to its previous version post an upgrade.
Data integrity	It deals with corrupted data by replicating it several times.

An Example of HDFS

Example of HDFS

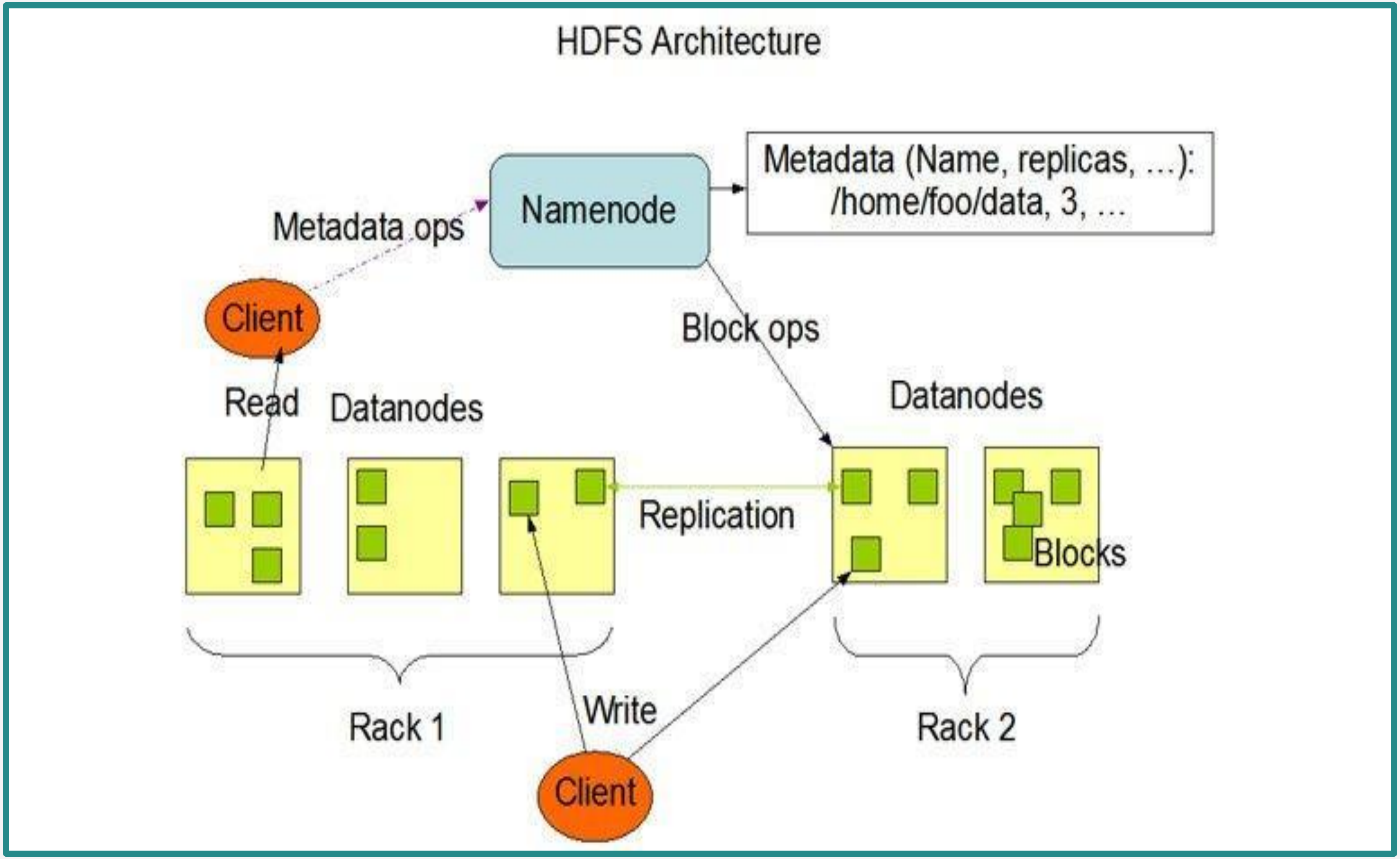
Hadoop Distributed File System (HDFS)

IBM and Cloudera

HDFS Architecture

- HDFS follows the master-slave data architecture.
- Each cluster comprises a single Namenode.
- Another component in the HDFS cluster is- Datanode, usually one per node in the HDFS cluster.

HDFS Architecture



HDFS Architecture - Namenode

- Namenode is the commodity hardware that contains the GNU/Linux operating system and the namenode software. It is a software that can run on commodity hardware.

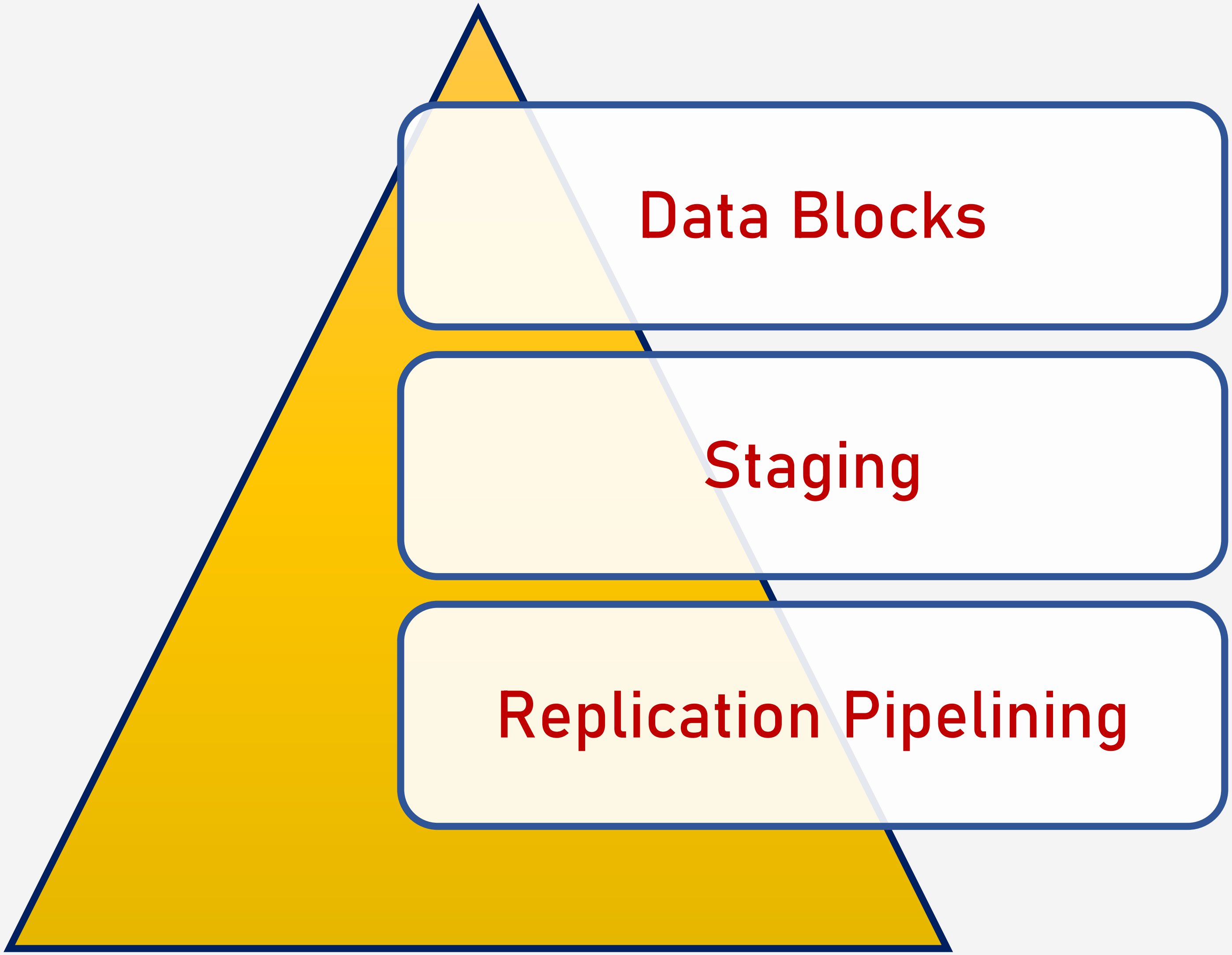
HDFS Architecture - Datanode

- Datanode is a commodity hardware having the GNU/Linux OS and datanode software.

HDFS Architecture- Blocks

- HDFS stores a file in a sequence of blocks. It is easy to configure the block size and the replication factor.

Data Organization in HDFS



Data Blocks

Staging

Replication Pipelining

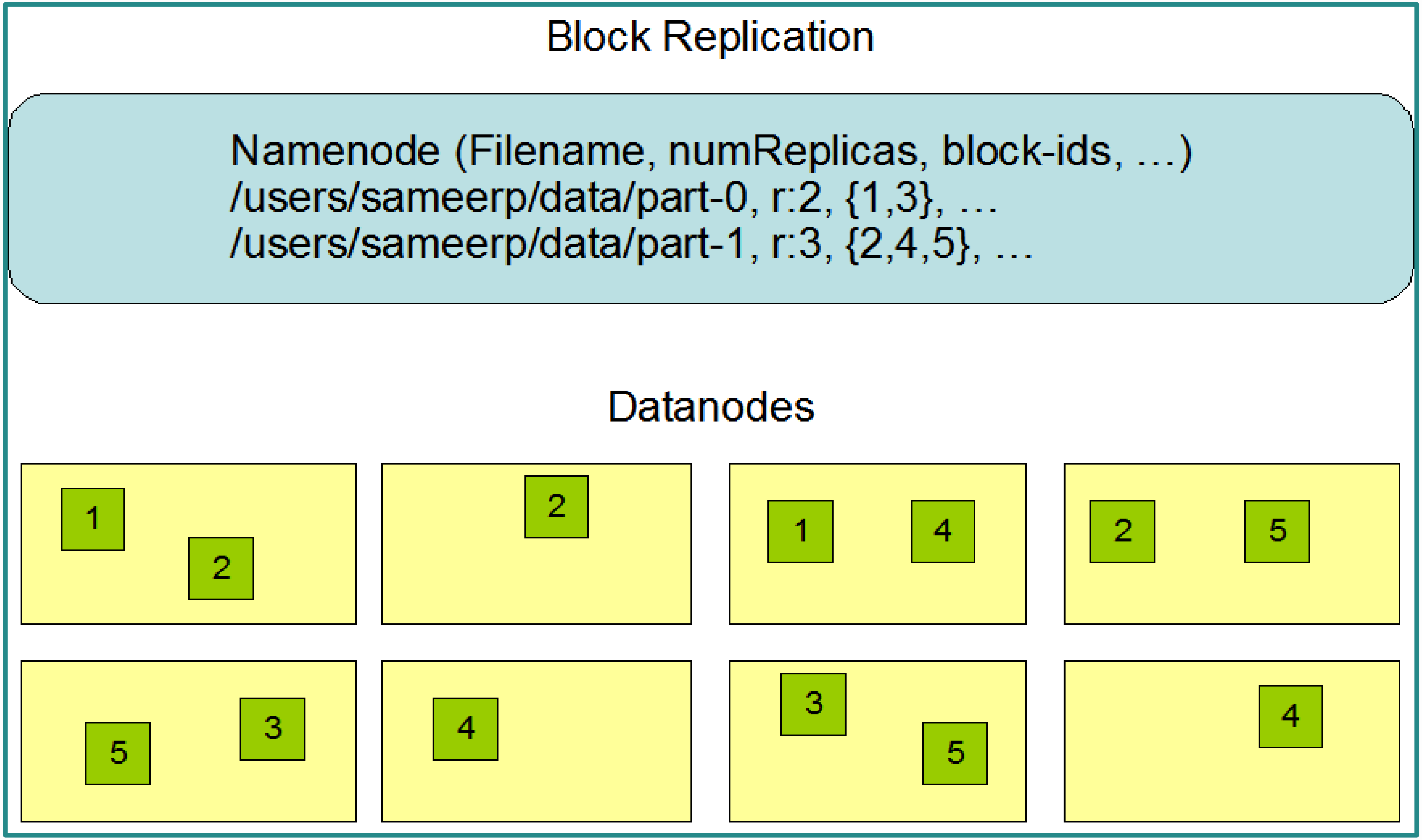
The File System Namespace

- HDFS data platform format follows a strictly hierarchical file system.
- An application or a user first creates a directory, and there will be files within this directory. The file system hierarchy is identical to other file systems.

The File System Namespace

- Data Replication
- Replica Placement
- Replica Selection
- Safemode

The File System Namespace



Block Replication

Namenode (Filename, numReplicas, block-ids, …)
/users/sameerp/data/part-0, r:2, {1,3}, …
/users/sameerp/data/part-1, r:3, {2,4,5}, …

Datanodes

HDFS Operations

Starting HDFS

Initially you have to format the configured HDFS file system, open namenode (HDFS server), and execute the following command:

```
$ hadoop namenode -format
```

HDFS Operations

- After formatting the HDFS, start the distributed file system.

```
$ start-dfs.sh
```

HDFS Operations

Listing Files in HDFS-

```
$ $HADOOP_HOME/bin/hadoop fs -ls <args>
```

HDFS Operations

Inserting Data into HDFS

Step 1- You have to create an input directory.

```
$ $HADOOP_HOME/bin/hadoop fs -mkdir /user/input
```

Step 2- Transfer and store a data file from local systems to the Hadoop file system using the put command.

```
$ $HADOOP_HOME/bin/hadoop fs -put /home/file.txt
/user/input
```

Step 3- You can verify the file using ls command.

```
$ $HADOOP_HOME/bin/hadoop fs -ls /user/input
```

HDFS Operations

Retrieving Data from HDFS

Step 1- Initially, view the data from HDFS using cat command.

```
$ $HADOOP_HOME/bin/hadoop fs -cat /user/output/outfile
```

Step 2- Get the file from HDFS to the local file system using get command.

```
$ $HADOOP_HOME/bin/hadoop fs -get /user/output/ /home/hadoop_tp/
```

HDFS Operations

Shutting down HDFS

```
$ stop-dfs.sh
```

Communication Protocols in HDFS

- HDFS communication protocols are layered on top of the TCP/IP protocol.
- Client establishes a connection to a configurable TCP port on the NameNode machine. It talks about the ClientProtocol with the NameNode.

Robustness in HDFS

- Primary objective of HDFS is to store data reliably even in the presence of failures.
- Three common types of failures are-
  - ✓ NameNode failures
  - ✓ DataNode failures and
  - ✓ Network partitions

Advantages of Using HDFS

HDFS is by far the most resilient and fault-tolerant technology that is available as an open-source platform, which can be scaled up or scaled down depending on the needs, making it really hard for finding an HDFS replacement for Big Data Hadoop storage needs.

Advantages of Using HDFS

- Distributed across hundreds or even thousands of servers.
- Works quite well for data loads that come in a streaming format.
- Works exclusively well for large datasets.
- Works on the assumption that moving of computation is much easier, faster, and cheaper than moving of data of humongous size.

Advantages of Using HDFS

- Accessibility.
- Highly Profitable.

Comparison – GFS vs HDFS

HDFS is a simplified version of GFS.

Similarities-

- Master and Slaves.
- Data blocks and replication.
- Tree structure.

Comparison- GFS vs HDFS

- File Appends.
- Master Failure.
- Garbage Collection. (GC)

That’s all for now…