Experiment No. :..... Date :.....

8.2 Abstract Models Of Storage

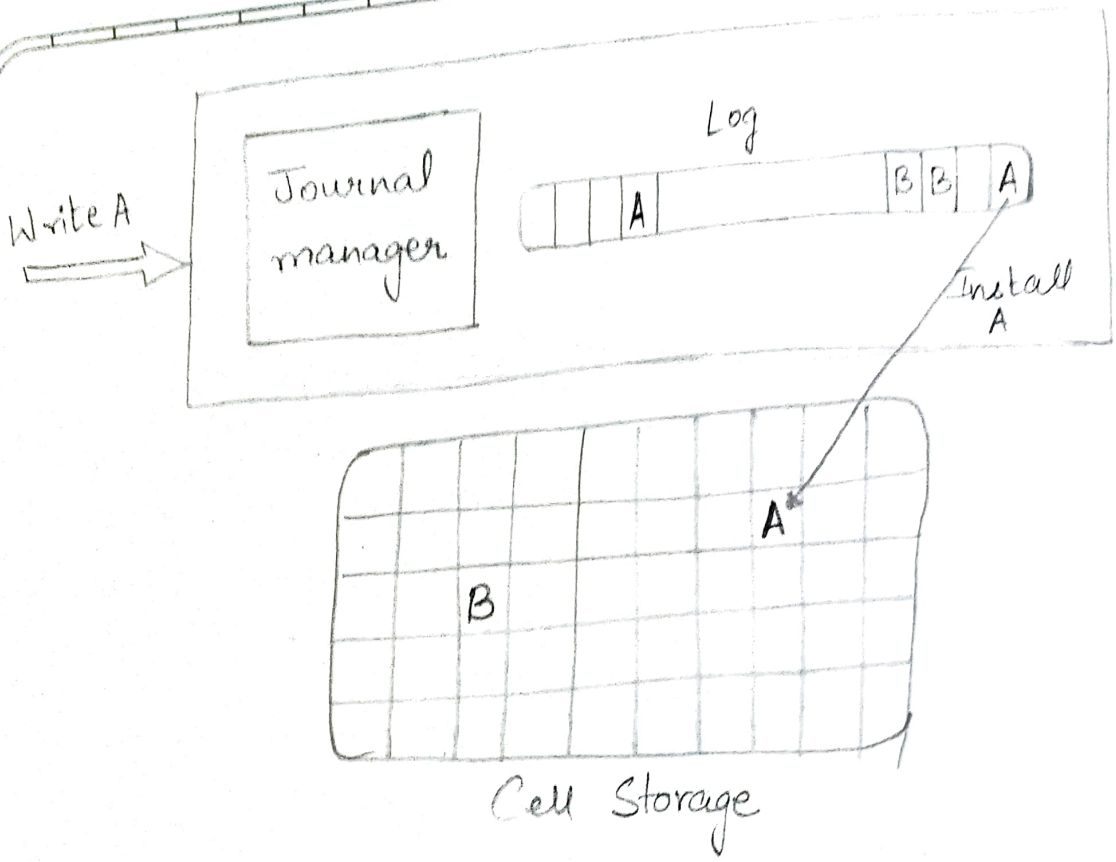
Storage model describes the layout of a data structure in a physical storage - a local disk, a removable media, or storage accessible via network.

(1) CELL STORAGE -

* assume that the storage consists of cells of the same size and that each object fits exactly in once cell.
* This model reflects the physical organization of several storage media.
* Primary memory of a computer is organized as an array of memory cells and a secondary storage device.
* Eg: a disk is organized in sectors or blocks read and written as a unit.

(2) JOURNAL STORAGE -

* system that keeps track of the changes that will be made in journal before committing them to the main file system.
* Journal is usually a circular log in a dedicated area of the file system.
* In the event of a system crash/power failure, such systems are quicker to bring back online and less likely to become corrupted.



Q. FILE SYSTEMS.

File system is a collection of directories. Each directory provides information about a set of files.

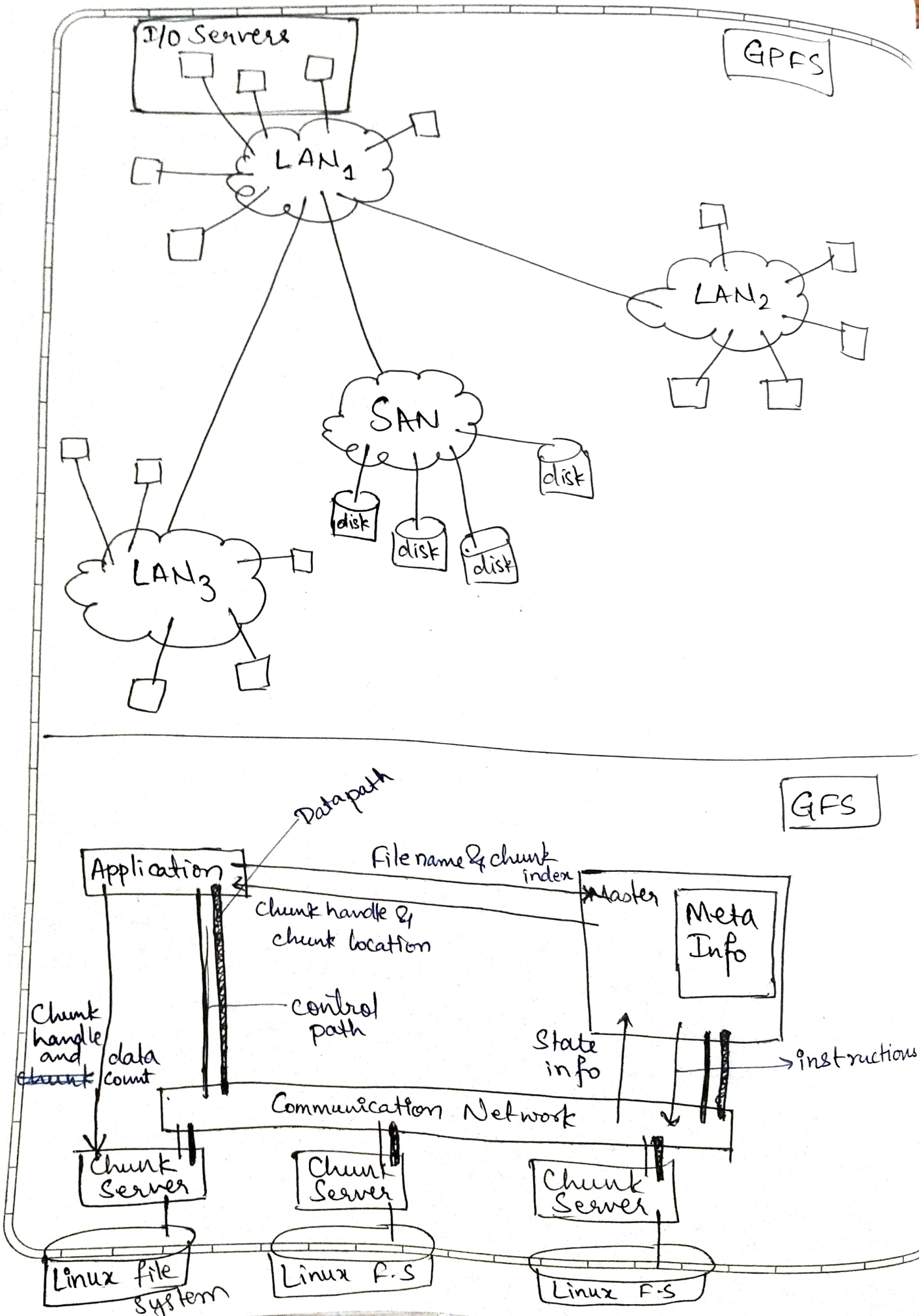
1. Traditional → Unix File System [UFS].
2. Distributed → Network File System. [NFS]
   → have been used for some time, but do not scale well and have reliability problems,
   → NFS server could be a single point of failure.
3. Storage Area Network [SAN].
   → allow cloud servers to deal with non-disruptive changes in the storage configuration.
   → Storage is pooled and then allocated based on the needs of the server.
4. Parallel File System [PFS].
   * scalable file system
   * capable of distributing files across a large no. of nodes, with a global naming space.

* Interconnection network of a PFS could be SAN.

Experiment No. :..... Date :.....

Q. General Parallel File System [GPFS].

* Parallel I/O implies concurrent execution of multiple input/output operations.
* Support for parallel I/O is essential for the performance of many applications.
* Concurrency control is a critical issue for parallel file system. Several semantics for handling the shared access are possible.
* GPFS:
  - developed by IBM as a successor of TigerShark multimedia file system.
  - designed for optimal performance of large clusters.
  - maximum file size $(2^{63}-1)$ bytes.
  - file consists of blocks of equal size, ranging from 16KB to 1MB, stripped across several disks.
* To recover from system failures, GPFS records all metadata updates in a write-ahead log file.
* The log files are maintained by each I/O node for each file system it mounts.
* The system uses RAID device with the stripes equal to the block size and dual-attached RAID controller.
* To further improve fault tolerance, GPFS data files as well as metadata are replicated on two different physical disks.


GPFS
I/O Servers
LAN1
LAN2
LAN3
SAN
disk
disk
disk
disk
GFS
Application
Datapath
File name & chunk index
Master
Meta Info
Chunk handle & chunk location
control path
Chunk handle and data count
State info
instructions
Communication Network
Chunk Server
Chunk Server
Chunk Server
Linux file system
Linux F.S
Linux F.S

Experiment No. :..... Date :.....

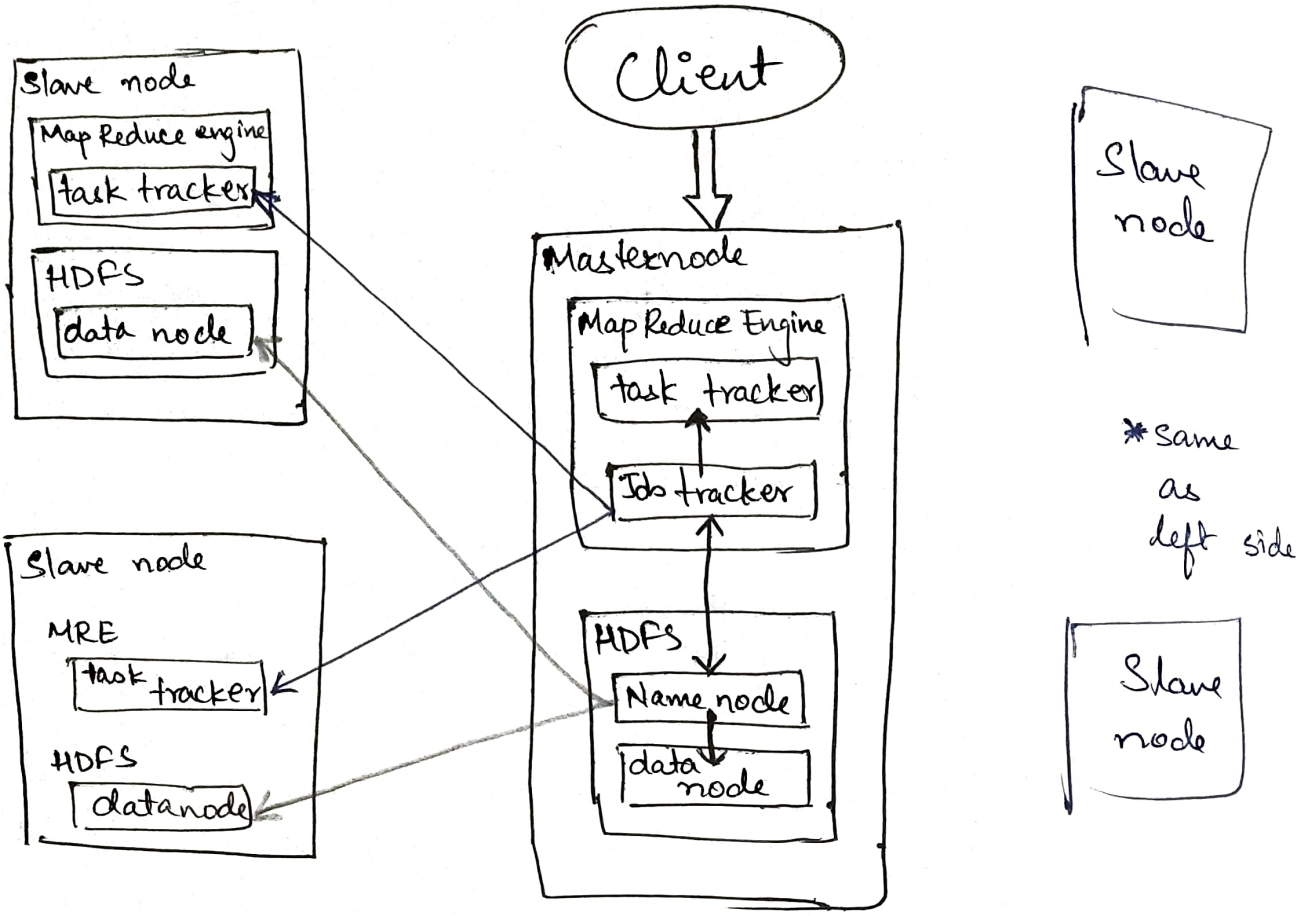
Q. Google File System

* GFS uses thousands of storage systems built from inexpensive commodity components to provide petabytes of storage to a large user community with diverse needs.

* Architecture –
- the master maintains state information about all system components. It controls a number of chunk servers.
- A chunk server runs under Linux. It uses metadata provided by the master to communicate directly with the application.
- The data and control paths are separate.
- Arrows show the flow of control between the application, the master and the chunk servers.

* Design Considerations –
- scalability & reliability
- The vast majority of files range in size from few GB to hundereds of TB.
- The most frequent / common operation is to append to an existing file.
- Sequential read operations are the norm.
- users process the data in bulk & are less concerned with the response time.
- consistency model should be relaxed to simplify the system implementation without overhead on the application developer.

Q. HADOOP CLUSTER USING HDFS.



* the cluster includes a master and four slave nodes.
* each node runs a MapReduce engine and a database engine.
* The job tracker of the master's engine communicates with task trackers on all the nodes and with the name node of HDFS.
* The name node of HDFS shares info about the data placement with the job tracker to minimize communication between the nodes where data is located and the ones where it is needed.
* The task tracker supervises the execution of the work allocated to the node.
* The job tracker attempts to dispatch the tasks to available slaves closest to the node.