An Overview of the Classes and Uses of Distributed Filesystems

Calvin Winkowski

Linux and Unix Users Group at Virginia Tech

April 24, 2014

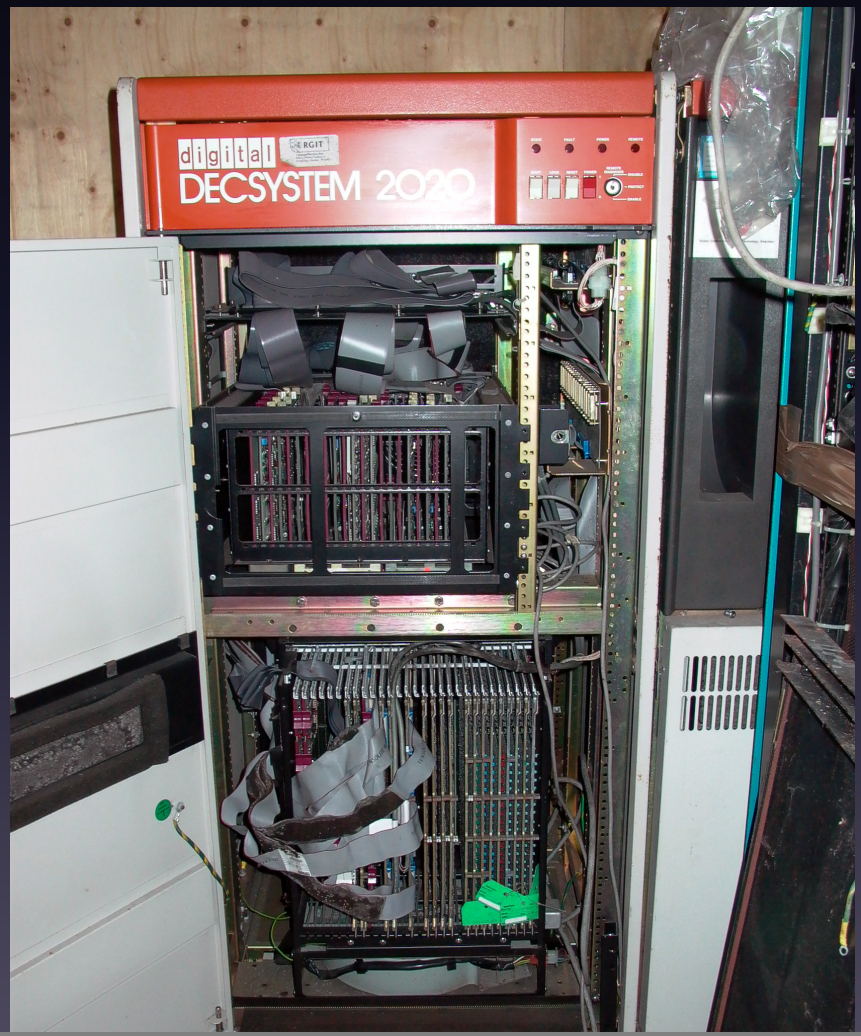
Origins

ITS

- MIT AI Lab - Incompatible Timesharing System
- Virtual software devices
- Inter-machine FS access

Locus

- Atomic operations
- Transparent data location
- Replication
- Caching



AFS

Goals

- Develop the future environment of computing
- Reduction in central timesharing devices
- Networking will be pivotal

Design

- File locking
- Location independent
- Hierarchy of file systems
- Storage independent of consumers

Modern World

Data

- Large data stores, very fast access
- High availability

Compute

- Variable requirements
- High speed & small storage vs Low speed & large storage

Gains of DFSs

Capability

- Speed (Write vs Read)
- Replication
- Locking
- De-duplication

Utility

- Unified Namespace
- Access Control
- Enumeration
- Failover vs Fault Tolerance
- Load Balancing

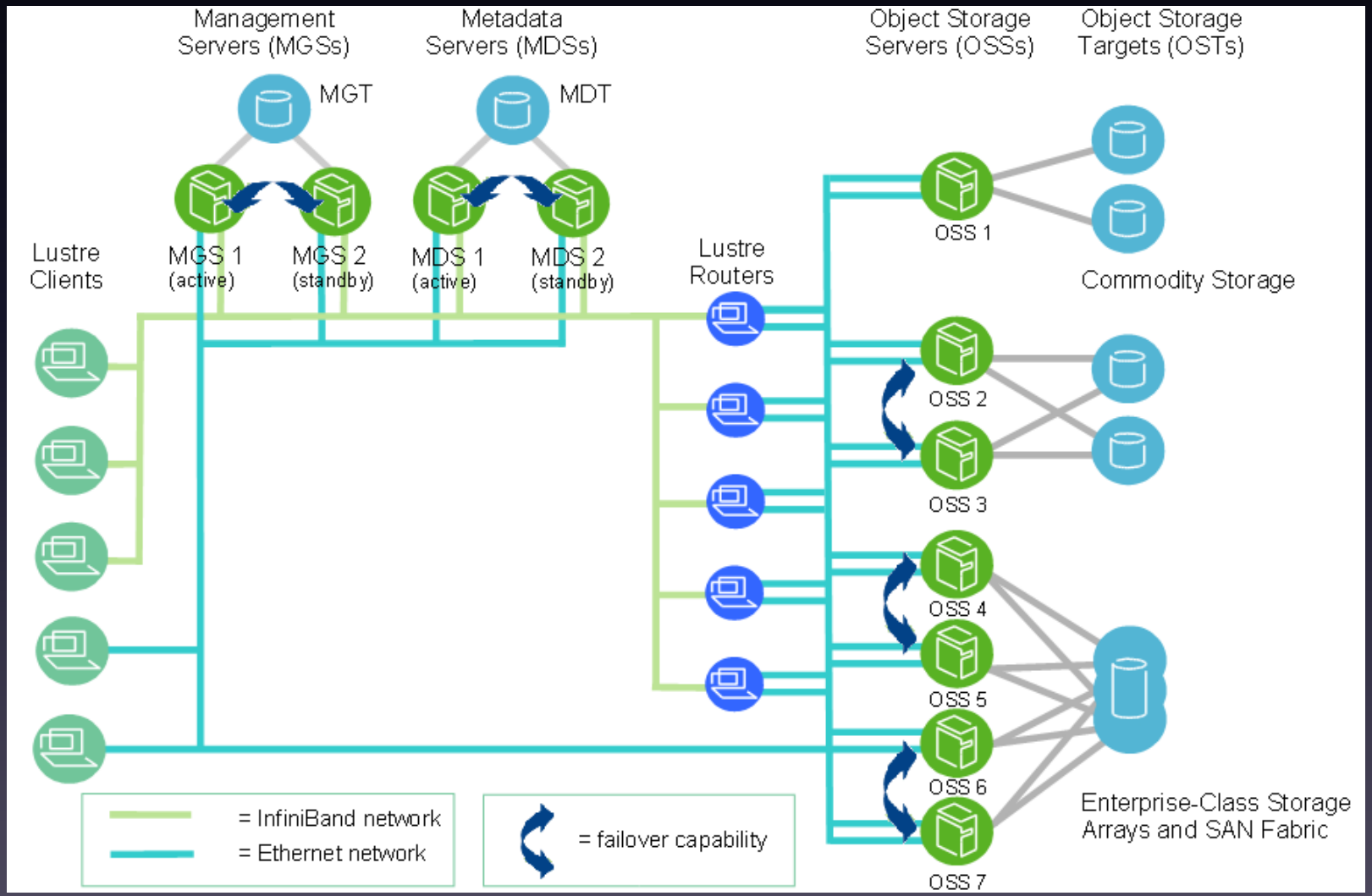
Classes of DFSs

- Centralized
- Decentralized
- Mapreduce
- Grid

Basics of Centralized DFSs

- Central Metadata server
- Distributed object storage
- Parallel Access is simplified
- Fast throughput, higher latency
- Loss of high-availability
- Favours big compute

Lustre


Management Servers (MGSs)
Metadata Servers (MDSs)
Object Storage Servers (OSSs)
Object Storage Targets (OSTs)
MGT
MDT
Lustre Clients
MGS 1 (active)
MGS 2 (standby)
MDS 1 (active)
MDS 2 (standby)
Lustre Routers
OSS 1
OSS 2
OSS 3
OSS 4
OSS 5
OSS 6
OSS 7
Commodity Storage
Enterprise-Class Storage Arrays and SAN Fabric
= InfiniBand network
= Ethernet network
= failover capability

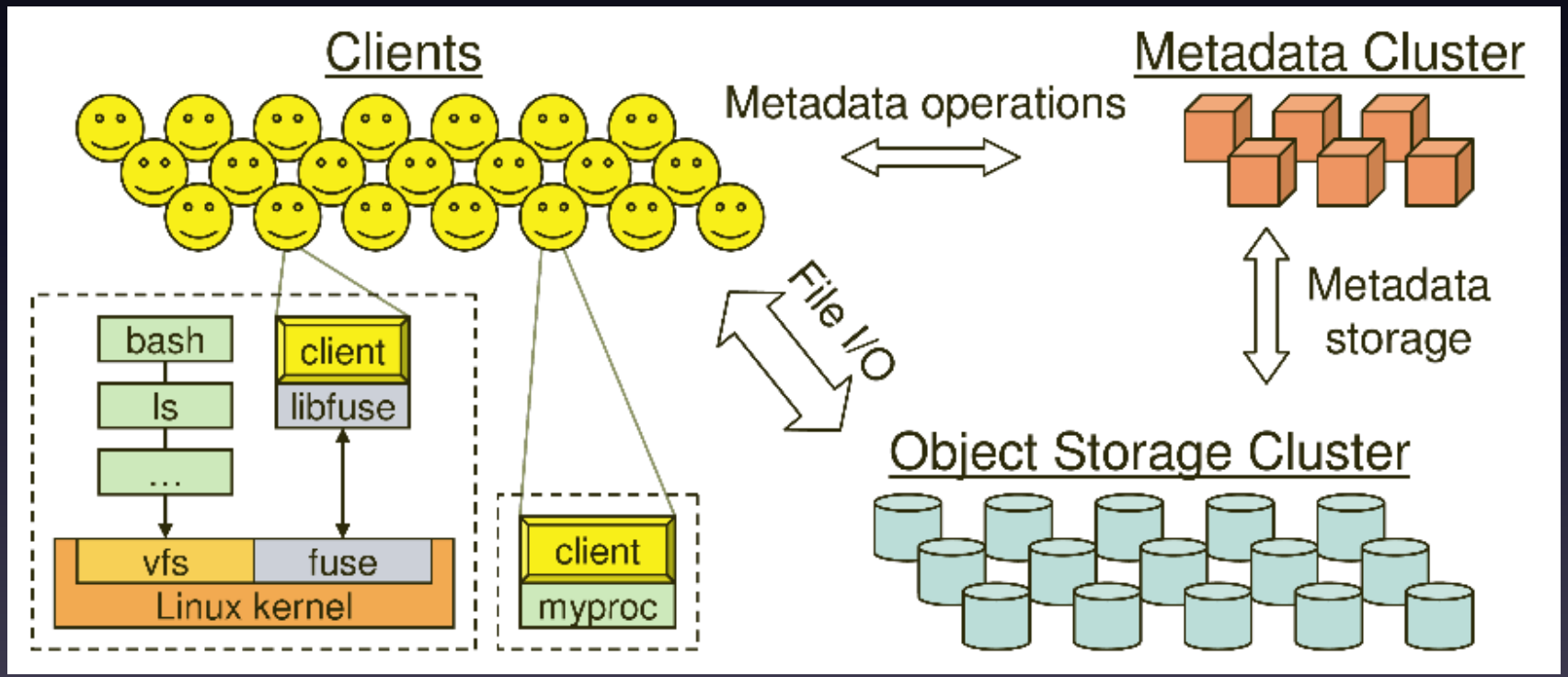
Examples of Centralized DFSs

- Lust*re* ← Note the spelling
- XtreemFS
- pNFS
- AFS

Basics of Decentralized DFSs

- All data is clustered
- Increases locking complexity
- HA friendly
- Loss of throughput
- May increase latency

CEPH


Clients
Metadata Cluster
Metadata operations
File I/O
Metadata
storage
Object Storage Cluster
bash
ls
...
client
libfuse
vfs
fuse
Linux kernel
client
myproc

Examples of Decentralized DFSs

- CEPH
- GFS
- GlusterFS
- FhGFS
- Tahoe-LAFS

MapReduction

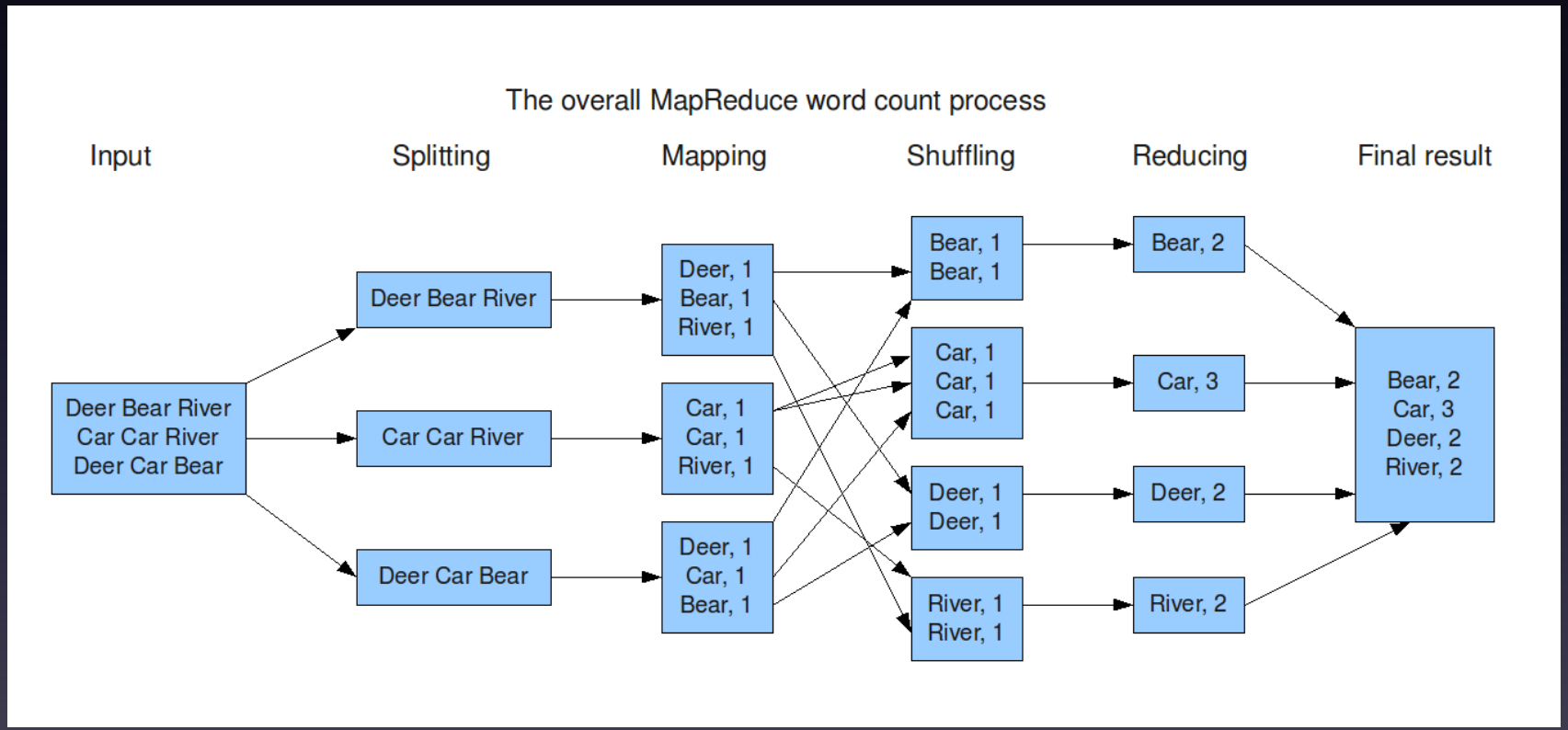
Map

- Mathematical definition
- Organizing the data for consumption

Reduce

- Produce a series of values
- Generating results

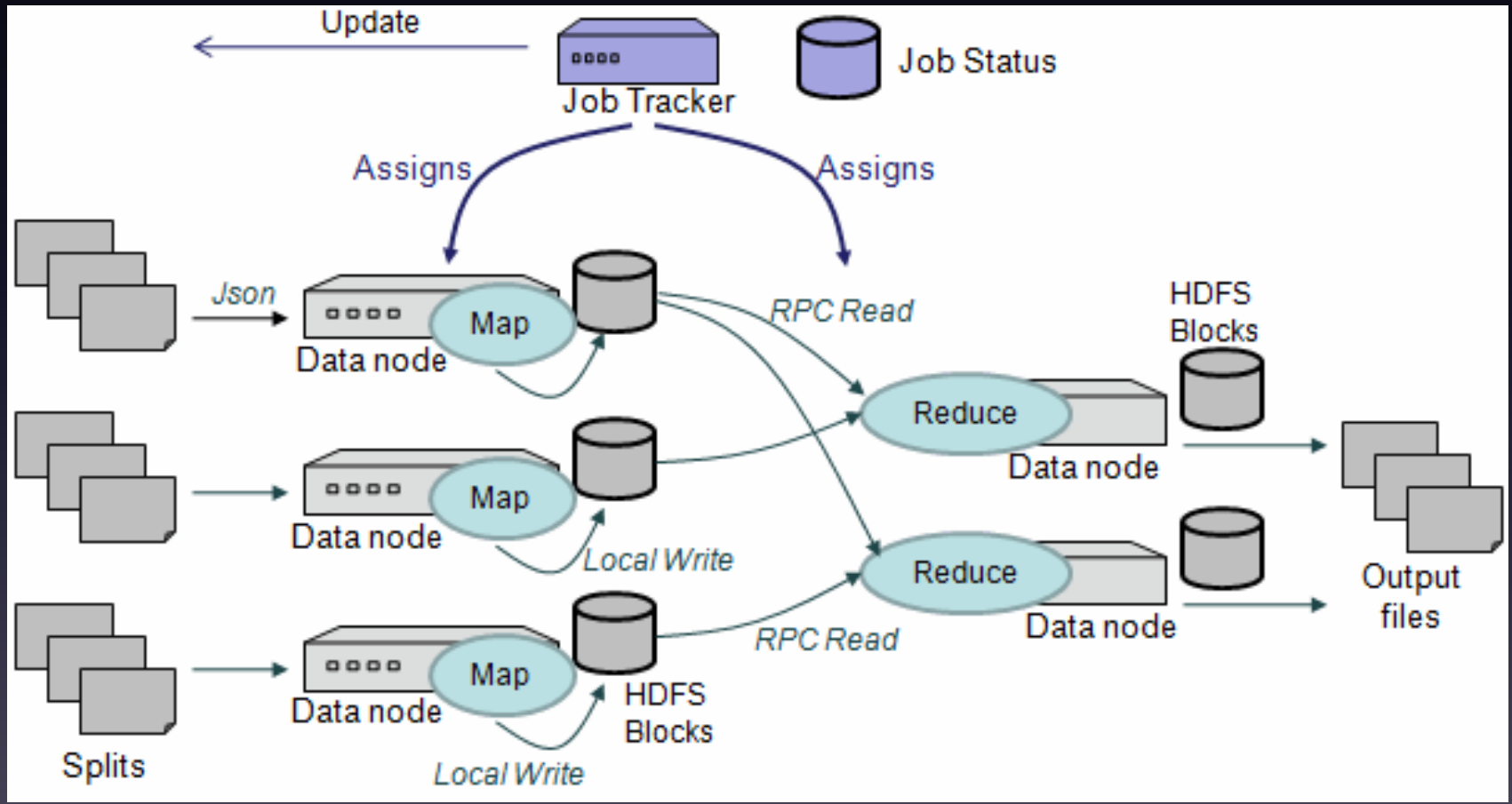
MapReduction Example



Basics of MapReduce DFSs

- Favours big data
- Some centralized entry point
- Master maps to storage servers
- Integrate with MapReduce frameworks
- Usually provides an API

Hadoop


Update
Job Tracker
Job Status
Assigns
Assigns
Json
Map
Data node
RPC Read
HDFS
Blocks
Reduce
Data node
Map
Data node
Local Write
Reduce
Data node
Output
files
Map
Data node
HDFS
Blocks
RPC Read
Splits
Local Write

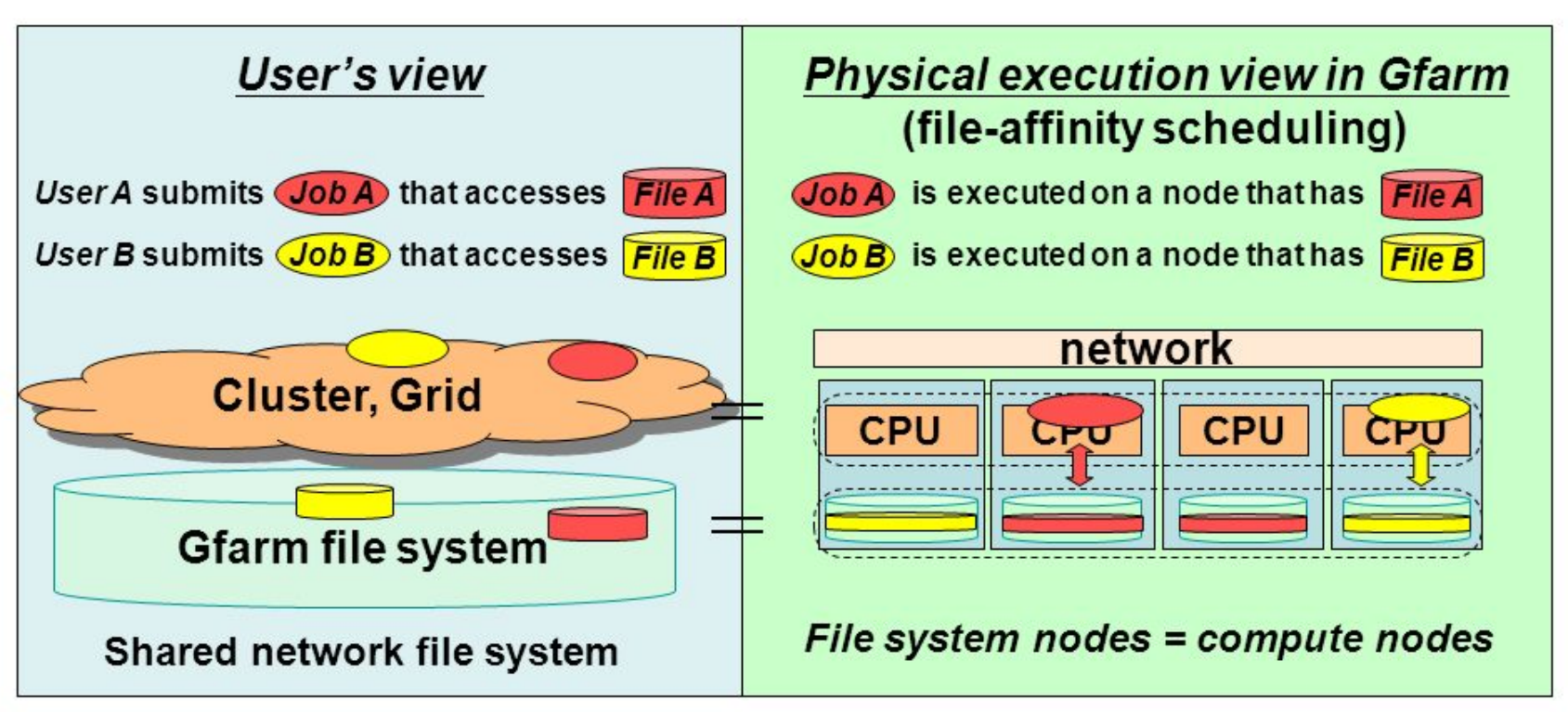
Examples of MapReduce DFSs

- Hadoop
- GFS (Not to be confused with GFS, GFS2, GlusterFS, GPFS)
- GloudStore
- QFS

Grid File Systems

- Take advantage of many small nodes
- Often run on workstations
- "Crowd source" storage
- Gfarm File System
- Scalable I/O
- Grid computing

Gfarm


User's view
User A submits Job A that accesses File A
User B submits Job B that accesses File B
Cluster, Grid
Gfarm file system
Shared network file system
Physical execution view in Gfarm
(file-affinity scheduling)
Job A is executed on a node that has File A
Job B is executed on a node that has File B
network
CPU
CPU
CPU
CPU
File system nodes = compute nodes

Questions