Distributed File Systems

Tran Giang Son, tran-giang.son@usth.edu.vn

ICT Department, USTH

File Systems

File Systems

File Systems

- System that permanently stores data
- Layered on top of a lower-level physical storage medium
- Divided into logical units called “files”
  - Files $\in$ directories
  - Directories $\in$ volume
  - Directories $\in$ directories
  - Addressable with “Path”

Metadata

- Volume
  - Available space
  - Formatting info
  - Character set
  - ...

Metadata

- Volume
  - Available space
  - Formatting info
  - Character set
  - ...
- File
  - Name
  - Owner / group
  - Date
  - Last access date
  - ...

Virtual File System

- /
  - Nested directories
  - Symlinks
  - Mount points

Low Level Organization

- File data and metadata stored separately
- File descriptors + meta-data stored in inodes
  - Large tree or tables
  - File content lookups
- Replicable

Low Level Organization

- Disks: `/dev/sdX` (Linux) or `/dev/diskX` (macOS)
  - Sequential array of blocks
  - 1KB chunks
  - Tree structure is flattened into blocks
  - Fragmentation

Fragmentation

A	B	C	(free space)

A	B	C	A	(free space)

A	(free space)	C	A	(free space)

A	D	C	A	D	(free)

Local File System

- **Very** numerous
  - Unix-like systems:

Local File System

- **Very** numerous
  - Unix-like systems: ext2/3/4

Local File System

- **Very** numerous
  - Unix-like systems: ext2/3/4 btrfs

Local File System

- **Very** numerous
  - Unix-like systems: ext2/3/4 btrfs zfs

Local File System

- **Very** numerous
  - Unix-like systems: ext2/3/4 btrfs zfs ReiserFS Reiser4

Local File System

- **Very** numerous
  - Unix-like systems: ext2/3/4 btrfs zfs ReiserFS Reiser4 JFS

Local File System

- **Very** numerous
  - Unix-like systems: ext2/3/4 btrfs zfs ReiserFS Reiser4 JFS MinixFS

Local File System

- **Very** numerous
  - Unix-like systems: ext2/3/4 btrfs zfs ReiserFS Reiser4 JFS MinixFS UFS UFS2

Local File System

- **Very** numerous
  - Unix-like systems: ext2/3/4 btrfs zfs ReiserFS Reiser4 JFS MinixFS UFS UFS2 XFS

Local File System

- **Very** numerous
  - Unix-like systems: ext2/3/4 btrfs zfs ReiserFS Reiser4 JFS MinixFS UFS UFS2 XFS ZFS

Local File System

- **Very** numerous
  - Unix-like systems: ext2/3/4 btrfs zfs ReiserFS Reiser4 JFS MinixFS UFS UFS2 XFS ZFS SquashFS

Local File System

- **Very** numerous
  - Unix-like systems: ext2/3/4 btrfs zfs ReiserFS Reiser4 JFS MinixFS UFS UFS2 XFS ZFS SquashFS
  - macOS:

Local File System

- **Very** numerous
  - Unix-like systems: ext2/3/4 btrfs zfs ReiserFS Reiser4 JFS MinixFS UFS UFS2 XFS ZFS SquashFS
  - macOS: HFS

Local File System

- **Very** numerous
  - Unix-like systems: ext2/3/4 btrfs zfs ReiserFS Reiser4 JFS MinixFS UFS UFS2 XFS ZFS SquashFS
  - macOS: HFS HFS+

Local File System

- **Very** numerous
  - Unix-like systems: ext2/3/4 btrfs zfs ReiserFS Reiser4 JFS MinixFS UFS UFS2 XFS ZFS SquashFS
  - macOS: HFS HFS+ APFS

Local File System

- **Very** numerous
  - Unix-like systems: ext2/3/4 btrfs zfs ReiserFS Reiser4 JFS MinixFS UFS UFS2 XFS ZFS SquashFS
  - macOS: HFS HFS+ APFS
  - DOS/Windows:

Local File System

- **Very** numerous
  - Unix-like systems: ext2/3/4 btrfs zfs ReiserFS Reiser4 JFS MinixFS UFS UFS2 XFS ZFS SquashFS
  - macOS: HFS HFS+ APFS
  - DOS/Windows: FAT FAT16 FAT32

Local File System

- **Very** numerous
  - Unix-like systems: ext2/3/4 btrfs zfs ReiserFS Reiser4 JFS MinixFS UFS UFS2 XFS ZFS SquashFS
  - macOS: HFS HFS+ APFS
  - DOS/Windows: FAT FAT16 FAT32 NTFS

Local File System

- **Very** numerous
  - Unix-like systems: ext2/3/4 btrfs zfs ReiserFS Reiser4 JFS MinixFS UFS UFS2 XFS ZFS SquashFS
  - macOS: HFS HFS+ APFS
  - DOS/Windows: FAT FAT16 FAT32 NTFS
  - Flash devices:

Local File System

- **Very** numerous
  - Unix-like systems: ext2/3/4 btrfs zfs ReiserFS Reiser4 JFS MinixFS UFS UFS2 XFS ZFS SquashFS
  - macOS: HFS HFS+ APFS
  - DOS/Windows: FAT FAT16 FAT32 NTFS
  - Flash devices: JFFS

Local File System

- **Very** numerous
  - Unix-like systems: ext2/3/4 btrfs zfs ReiserFS Reiser4 JFS MinixFS UFS UFS2 XFS ZFS SquashFS
  - macOS: HFS HFS+ APFS
  - DOS/Windows: FAT FAT16 FAT32 NTFS
  - Flash devices: JFFS JFFS2

Local File System

- **Very** numerous
  - Unix-like systems: ext2/3/4 btrfs zfs ReiserFS Reiser4 JFS MinixFS UFS UFS2 XFS ZFS SquashFS
  - macOS: HFS HFS+ APFS
  - DOS/Windows: FAT FAT16 FAT32 NTFS
  - Flash devices: JFFS JFFS2 exFAT
- Case (in)sensitive
- Some has encryption supports: APFS HFS+ btrfs ZFS ext4

Local File System on Unix-like Systems

- VFS
- UID/GID
- File mode for access (RWX)
- Superblock, journaling, snapshots

Journaling

- Log changes to a "journal" before it is commited
  - For "multiple writes" in an atomic action, e.g. append to file

Journaling

- Log changes to a "journal" before it is commited
  - For "multiple writes" in an atomic action, e.g. append to file
    - Update metadata
    - Allocate space
    - Write data
  - Can "playback" journal to recover data quickly

Journaling

- Log changes to a "journal" before it is commited
  - For "multiple writes" in an atomic action, e.g. append to file
    - Update metadata
    - Allocate space
    - Write data
  - Can "playback" journal to recover data quickly
- Log what?
  - Changes to file content: slow. twice.
  - Changes to metadata: fast, prone to corruption

Journaling

- Log changes to a "journal" before it is commited
  - For "multiple writes" in an atomic action, e.g. append to file
    - Update metadata
    - Allocate space
    - Write data
  - Can "playback" journal to recover data quickly
- Log what?
  - Changes to file content: slow. twice.
  - Changes to metadata: fast, prone to corruption
- Where?

Journaling

- Log changes to a "journal" before it is commited
  - For "multiple writes" in an atomic action, e.g. append to file
    - Update metadata
    - Allocate space
    - Write data
  - Can "playback" journal to recover data quickly
- Log what?
  - Changes to file content: slow. twice.
  - Changes to metadata: fast, prone to corruption
- Where? ext2/3/4

Journaling

- Log changes to a "journal" before it is commited
  - For "multiple writes" in an atomic action, e.g. append to file
    - Update metadata
    - Allocate space
    - Write data
  - Can "playback" journal to recover data quickly
- Log what?
  - Changes to file content: slow. twice.
  - Changes to metadata: fast, prone to corruption
- Where? ext2/3/4 xfs3

Journaling

- Log changes to a "journal" before it is commited
  - For "multiple writes" in an atomic action, e.g. append to file
    - Update metadata
    - Allocate space
    - Write data
  - Can "playback" journal to recover data quickly
- Log what?
  - Changes to file content: slow. twice.
  - Changes to metadata: fast, prone to corruption
- Where? ext2/3/4 xfs3 ReiserFS

Journaling

- Log changes to a "journal" before it is commited
  - For "multiple writes" in an atomic action, e.g. append to file
    - Update metadata
    - Allocate space
    - Write data
  - Can "playback" journal to recover data quickly
- Log what?
  - Changes to file content: slow. twice.
  - Changes to metadata: fast, prone to corruption
- Where? ext2/3/4 xfs3 ReiserFS JFS

Journaling

- Log changes to a "journal" before it is commited
  - For "multiple writes" in an atomic action, e.g. append to file
    - Update metadata
    - Allocate space
    - Write data
  - Can "playback" journal to recover data quickly
- Log what?
  - Changes to file content: slow. twice.
  - Changes to metadata: fast, prone to corruption
- Where? ext2/3/4 xfs3 ReiserFS JFS XFS

Journaling

- Log changes to a "journal" before it is commited
  - For "multiple writes" in an atomic action, e.g. append to file
    - Update metadata
    - Allocate space
    - Write data
  - Can "playback" journal to recover data quickly
- Log what?
  - Changes to file content: slow. twice.
  - Changes to metadata: fast, prone to corruption
- Where? ext2/3/4 xfs3 ReiserFS JFS XFS btrfs

RAID

- Redundant Array of Inexpensive/Independent Disks
- Levels
  - RAID0: Striped. Performance.
  - RAID1: Mirror. Reliability.
  - RAID5: Striped with distributed parity (no mirror)
  - RAID6: Striped with dual distributed parity
  - RAID10: Striped mirrors
- Where? btrfs ZFS

Snapshot

- What?

Snapshot

- What?
  - Copy of a set of files and directories
  - A certain point in time

Snapshot

- What?
  - Copy of a set of files and directories
  - A certain point in time
- Types?
  - Read-only
  - Read-write: copy-on-write
  - Supported by filesystem, or LVMs
  - Backup from read-only snapshots

Snapshot

- What?
  - Copy of a set of files and directories
  - A certain point in time
- Types?
  - Read-only
  - Read-write: copy-on-write
  - Supported by filesystem, or LVMs
  - Backup from read-only snapshots
- Where? UFS ZFS

Problems

- Size
- Speed
  - Large file? small files?
  - Reads? writes?
- Block size
  - Small : less waste, more overhead
  - Large : more waste, less overhead

Distributed File Systems

What?

- Access to files on remote servers
- Concurrency
  - Consistency
  - Locking
- Support for local caching and replication

What?

- Security
- Reliability
- Consistency
- Parallel

Why?

Why?

- File sharing
- Backups
- Centralized storage system
- Size

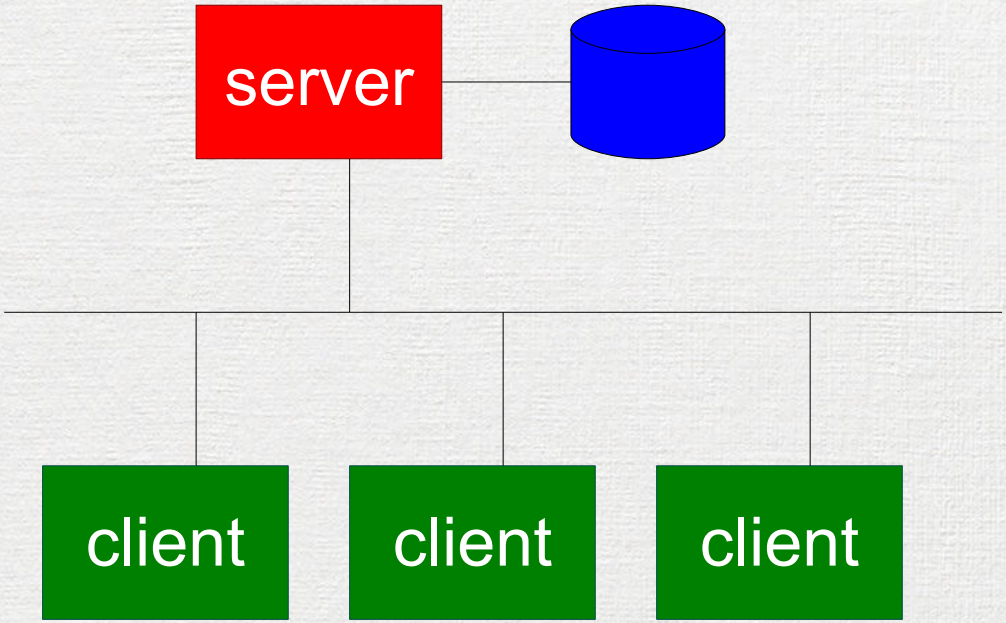
Examples

- Network File System (NFS)
- Andrew File System (AFS)
- GlusterFS

NFS

- 1980s by Sun
- Most widely known distributed filesystem
- Presented with standard POSIX FS interface
- Network drives are mounted into local directory hierarchy

NFS



NFS Protocol

- Originally stateless (similar to what?)
  - UDP, not TCP
  - File locking
  - All requests have enough info
    - E.g. write to what file? at what offset? what content?
- TCP ftw.

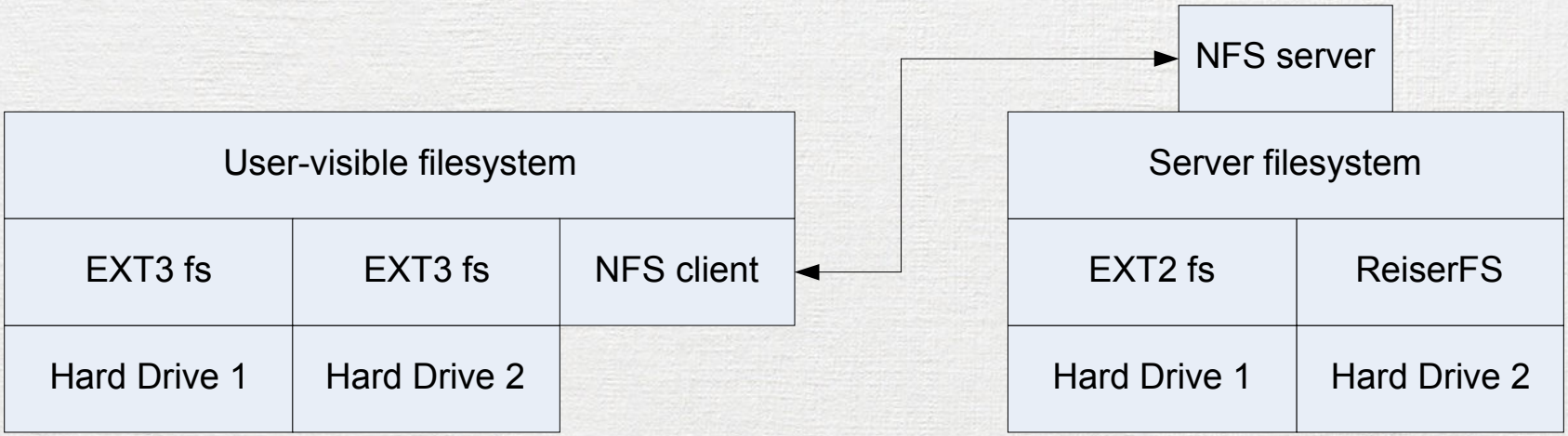
NFS Overview

- RPC
- Client
  - On top of VFS
  - vnode to NFS RPCs
- Server
  - Stateless
  - Write to storage before return

NFS Server

- Use existing file system on server
- No local disk layout
- Export local disk to clients
  - `/etc/exports`
  - `/storage 192.168.0.0/24(rw,sync)`

NFS Server


NFS server
User-visible filesystem
Server filesystem
EXT3 fs
EXT3 fs
NFS client
EXT2 fs
ReiserFS
Hard Drive 1
Hard Drive 2
Hard Drive 1
Hard Drive 2

NFS Locking

- Stateful locking
  - Client informs servers to lock
  - Server notifies clients of lock requests
  - Lease-based: must renew locks
  - Disconnect $\rightarrow$ release lock

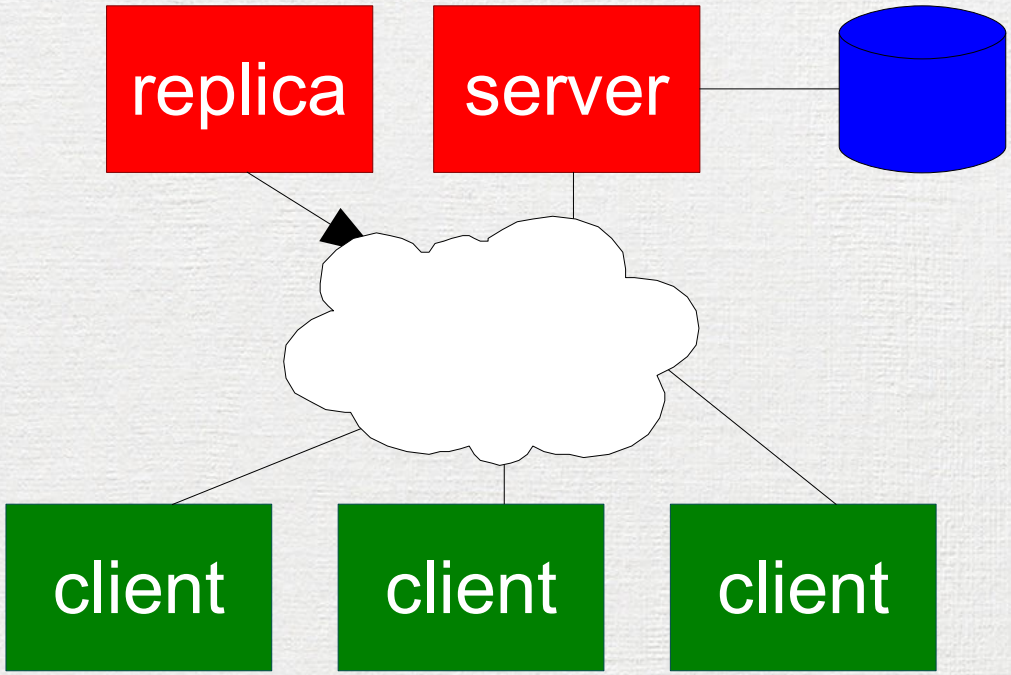
NFS Summary

- Very popular
- Full POSIX interface
- Single server
  - Scalability problem
  - Simplifies protocol, consistency
- Problems
  - Fault tolerance
  - Scable performance
  - Consistency

AFS

- The Andrew File System
- Carnegie Mellon University, 1983
  - OpenAFS
- Kerberos authentication
- Scalabe
- Client side caching
- Read only replication

AFS



AFS vs NFS[1]

- Client/server ratio
  - NFS 25:1
  - AFS 100s:1
- Reliability
  - NFS server dies → everyone dies
  - AFS server dies → everyone reads only from replica
- Security
  - Keberos vs non-keberos

[1] Source: **Morgan Stanley**

AFS Local Caching

- File reads/writes on local cached copy
- Modified copy is synchronized back on close
- Open local copies are notified
- Problem: consistency

AFS Replication

- Read-only of file system volumes
- Guaranteed to be atomic checkpoints
- Changes do not propagate to existing read-only copies

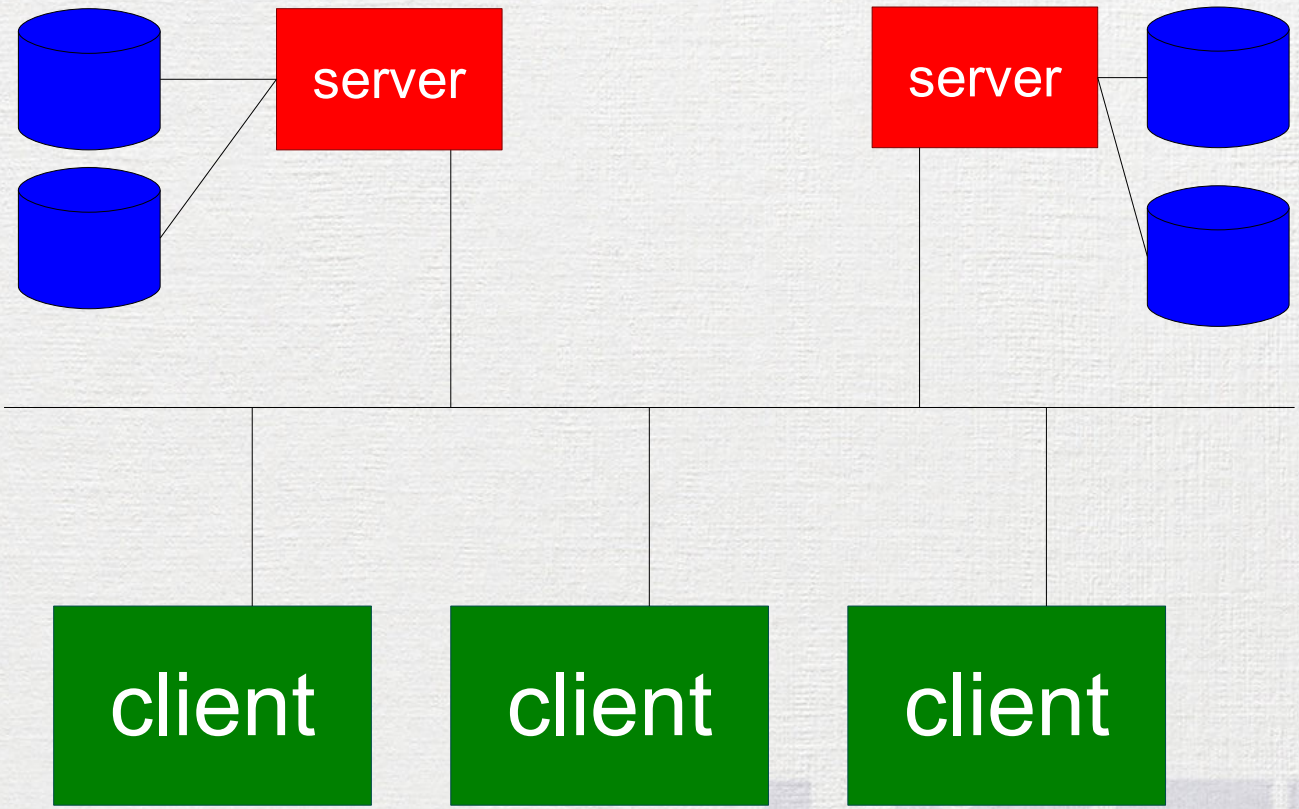
AFS Summary

- Not POSIX
  - Security/permissions
  - No write-through (no immediate sync)
- High availability with replicas and local caching
- Scalability for read loads

GlusterFS

- Network-attached storage file system
- Scalablity
- High availibility
- Ethernet / Infiniband
- FUSE or NFS translator

GlusterFS



GlusterFS

- Brick: basic unit of storage
- $Volume = bricks$
  - Presents to user
- Peer/node
  - Server hosting the bricks
  - Running Gluster daemons
- Trusted storage pool
  - Group of peers
- Client: the machine which mounts the volume

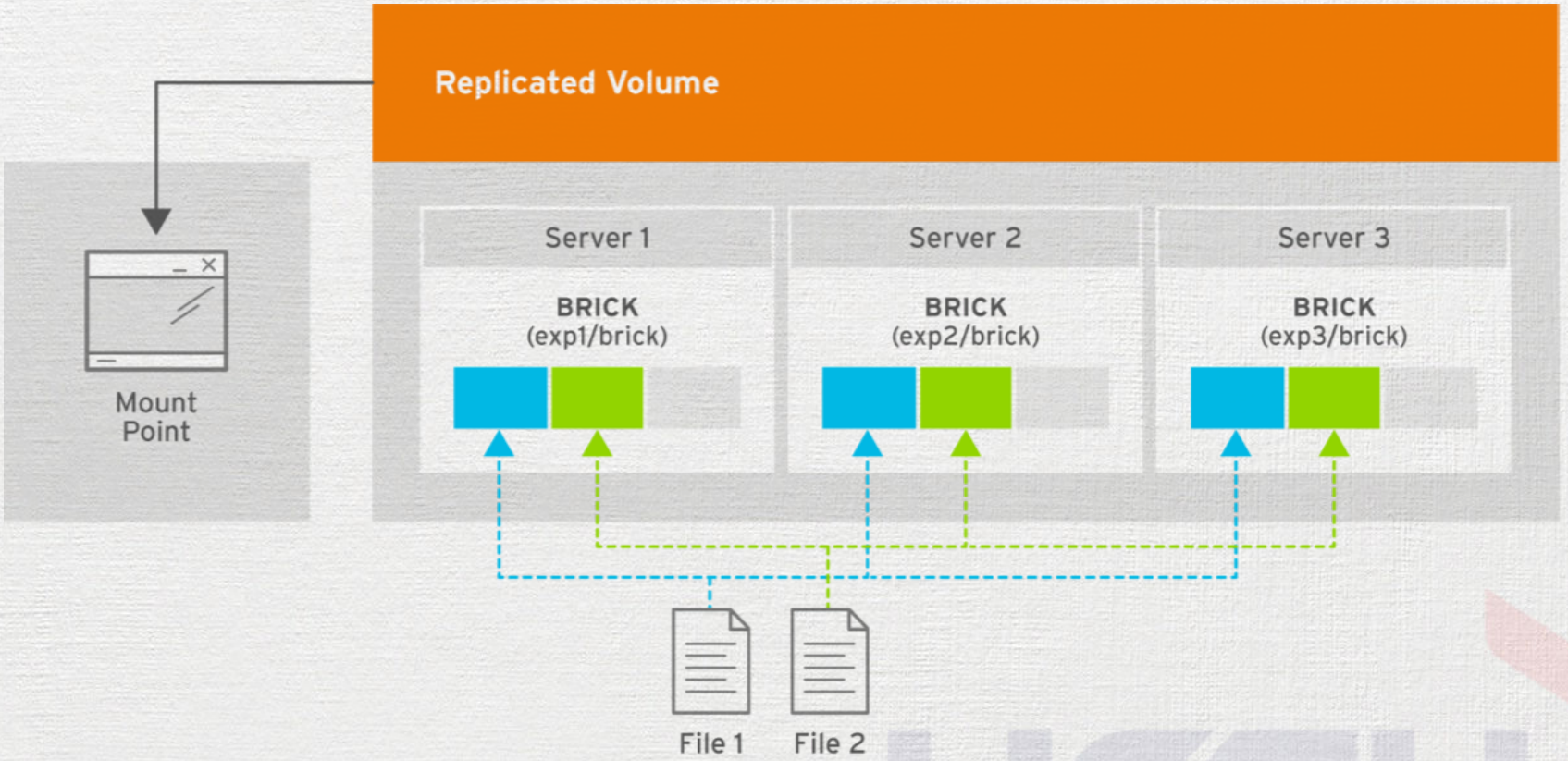
GlusterFS

- Use common hardware
- Use existing file system for bricks
  - A file system mount point
- Scalability: similar to RAID0
  - Striped among bricks
  - Performance
  - Capacity
  - Aggregated resources
  - **No metadata server**

GlusterFS

- High availability: similar tor RAID1
  - Mirrored among bricks
- Can be both striped and mirrored

GlusterFS


Replicated Volume
Server 1
Server 2
Server 3
BRICK
(exp1/brick)
BRICK
(exp2/brick)
BRICK
(exp3/brick)
Mount
Point
File 1
File 2

Practical work 6: GlusterFS

- Install GlusterFS on your laptops, make a trusted pool
- Create a distributed replicated volume
- Write a short report in LaTeX:
  - Name it « 06.glusterfs.tex »
  - Write the commmands for above steps
  - Perform benchmarks
    - Small files: number of accesses/s vs number of servers
    - Large files: read speed (MB/s) vs number of servers
  - Who does what.
- Work in your group, in parallel
- Push your report to corresponding forked Github repository