Today

Distributed Subsystems (Lesson 7)

✓ * Global memory system

✓ * Distributed shared memory

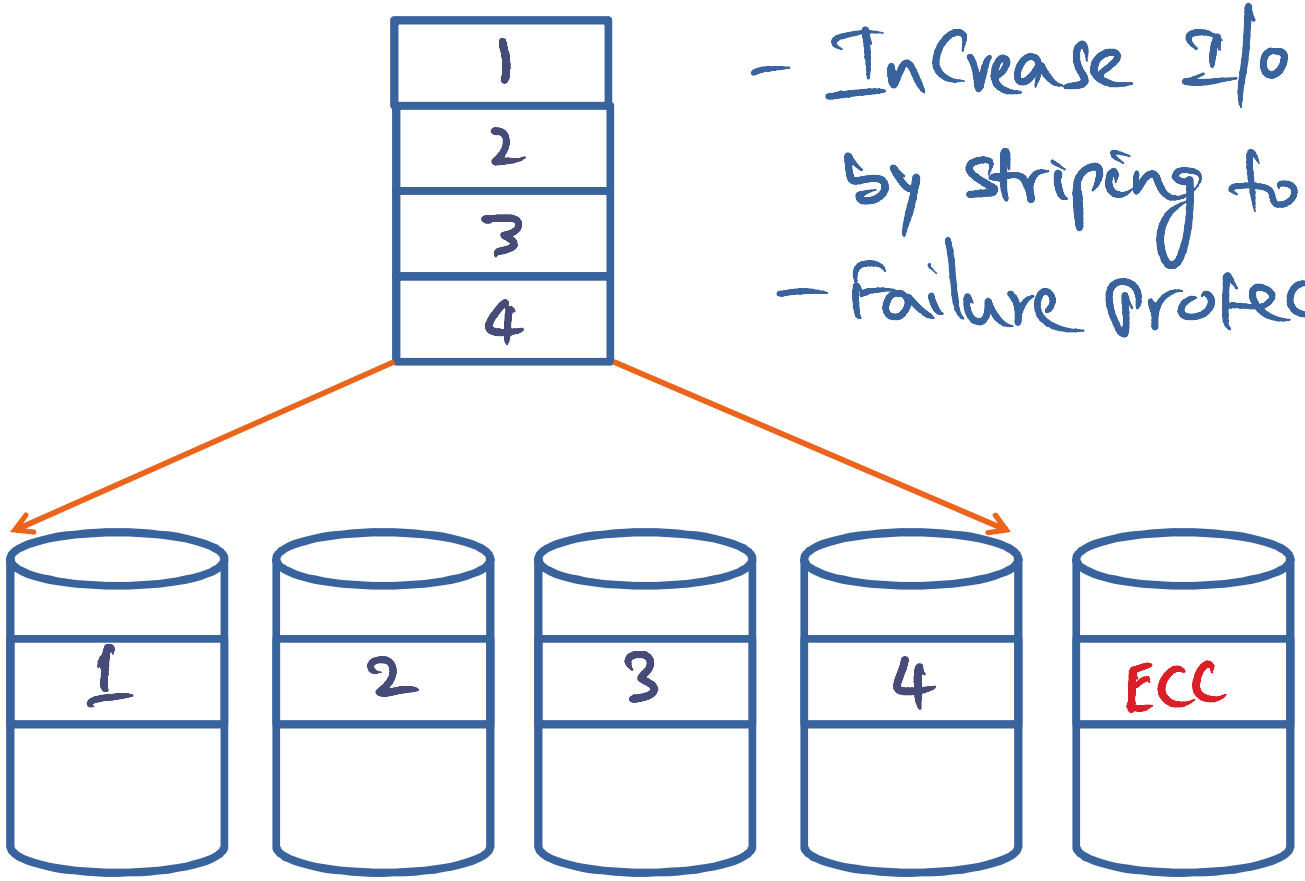
⇒ * Distributed file systems

Wednesday

Failures + recovery (Lesson 8)

* please watch LRVM lecture *

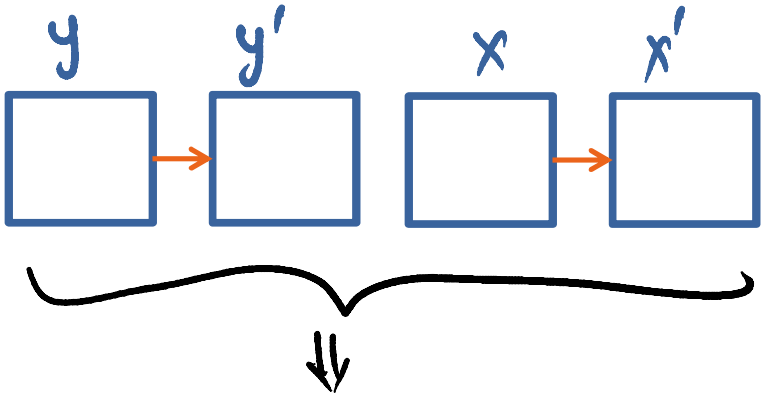
Preliminaries: Striping a file to multiple Disks

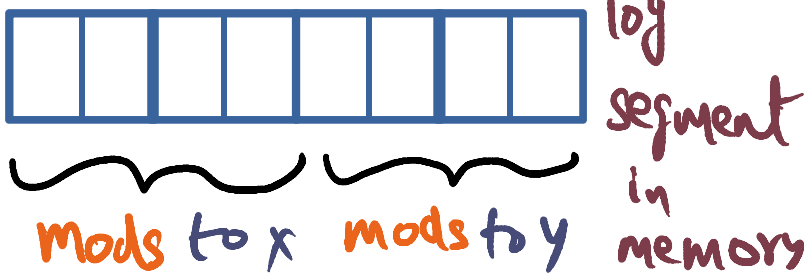


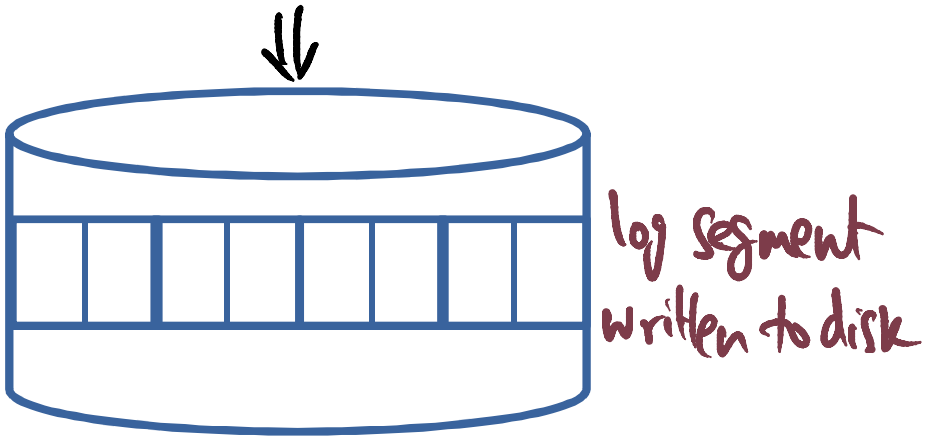
- Increase I/O Bandwidth by striping to parallel disks
- Failure protection by ECC

Drawbacks
- Cost
- Small write Problem

Preliminaries: Log structured File System







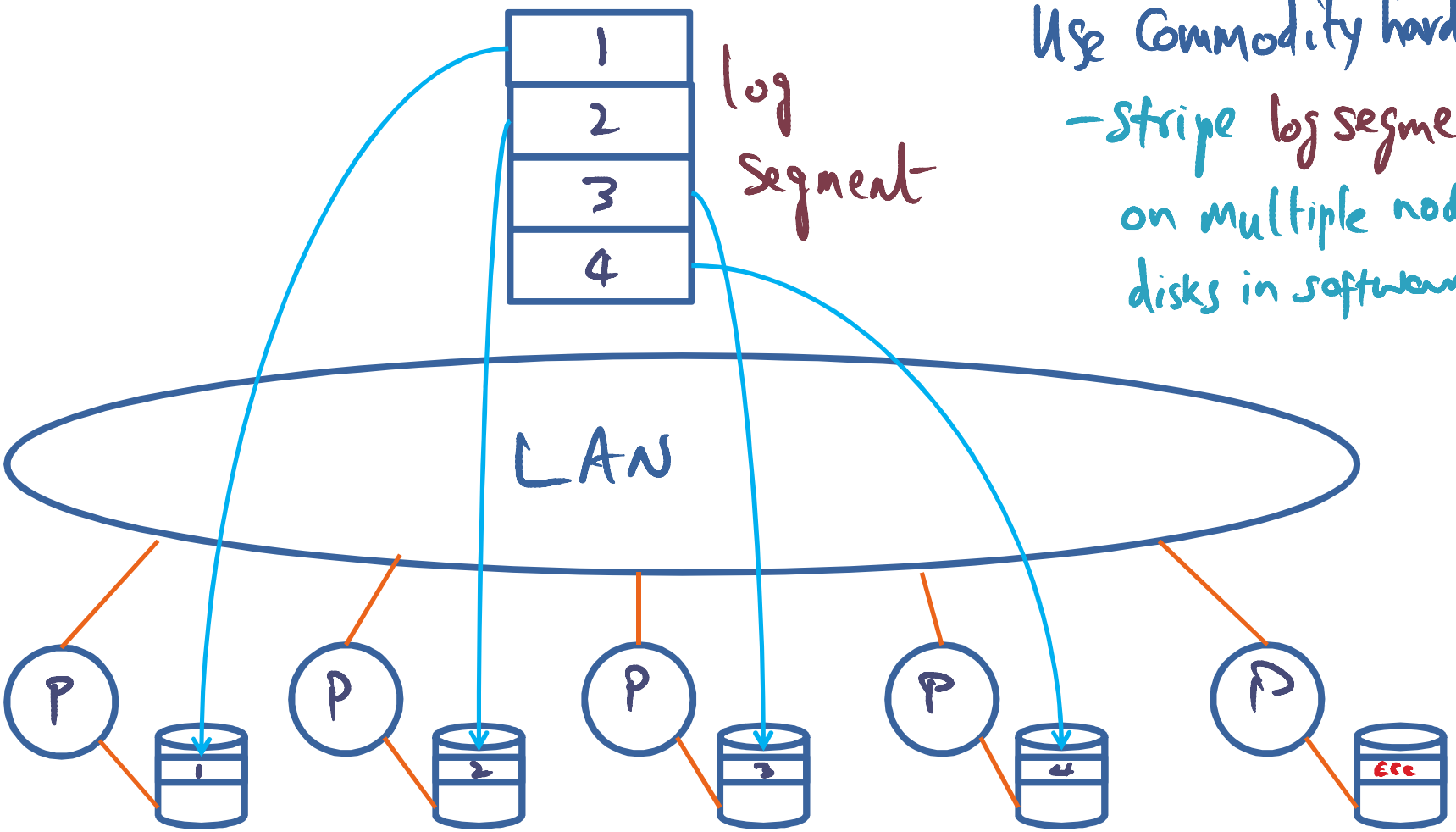
- Buffer changes to multiple files in one contiguous log segment data structure
- Flush log segment to disk once it fills up or periodically

solves small write problem

Preliminaries: Software RAID

Zebra File System (UC-Berkeley)

- Combines LFS + RAID



Use Commodity hardware

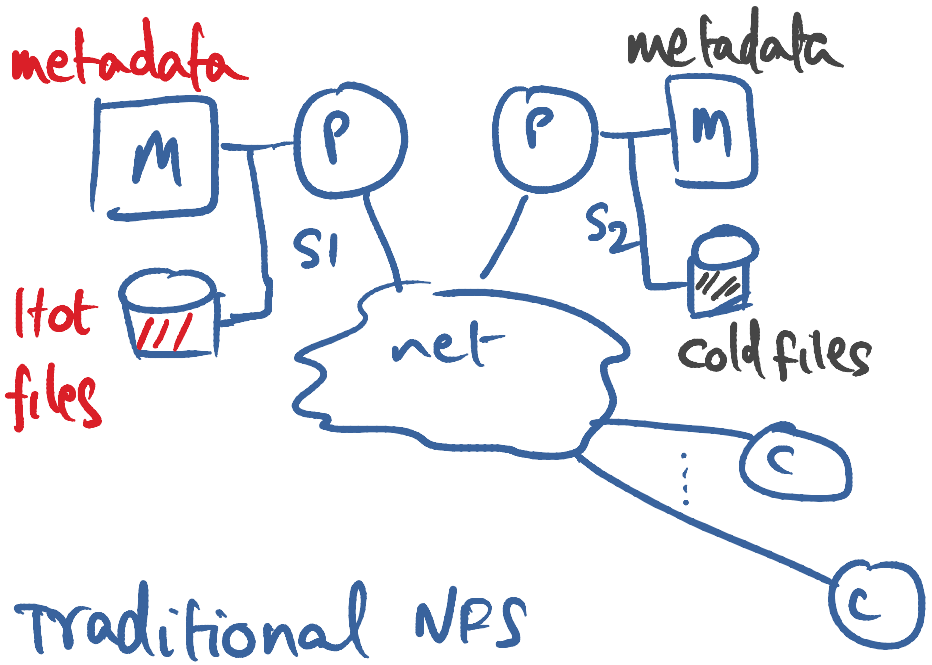
- stripe log segment on multiple nodes' disks in software

Putting them all together Plus more

xFS – a DFS

- log based striping (from Zebra)
- Cooperative caching (from prior UCB work)
- Dynamic Management of data + metadata
- Subsetting Storage Servers
- Distributed log cleaning

Dynamic management

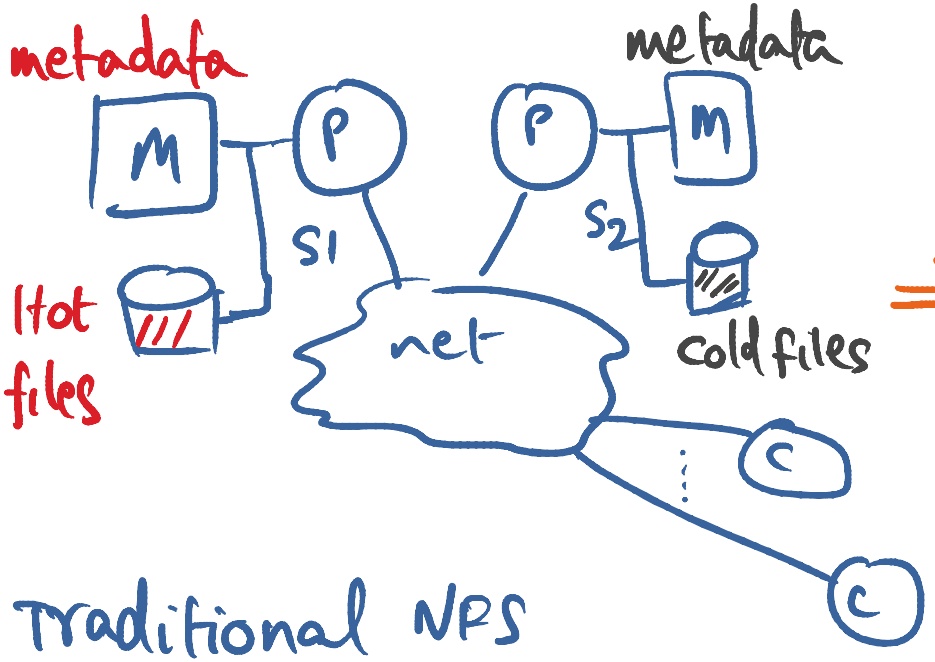


Traditional NFS

With centralized server(s)

- Memory contents
  - * metadata, file cache, client caching directory

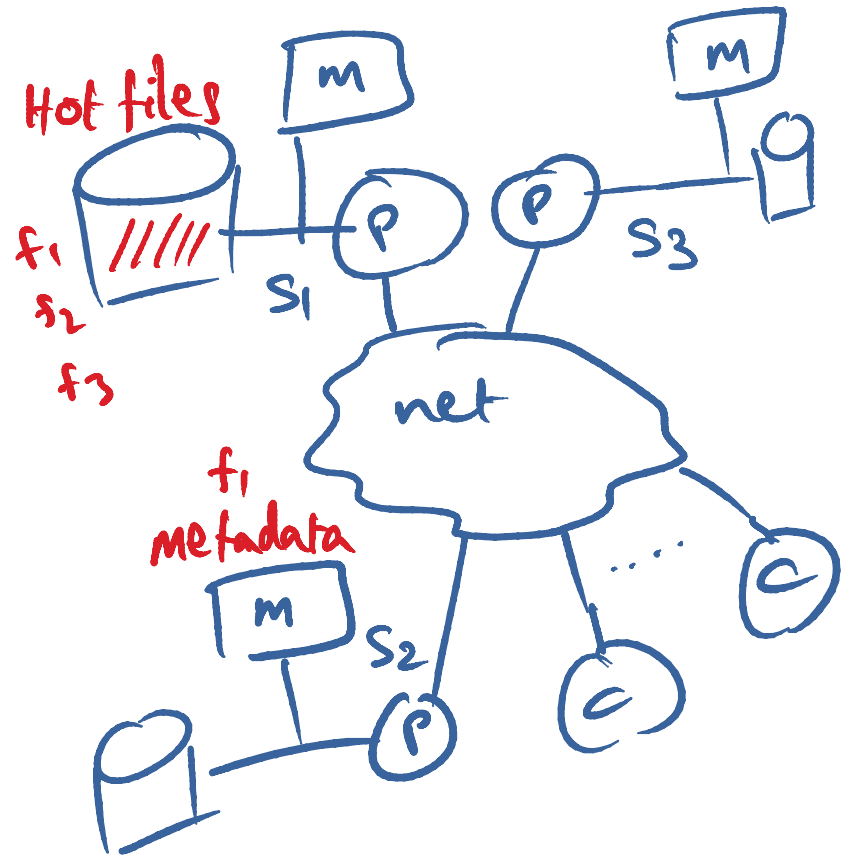
Dynamic management



Traditional NFS

With centralized server(s)

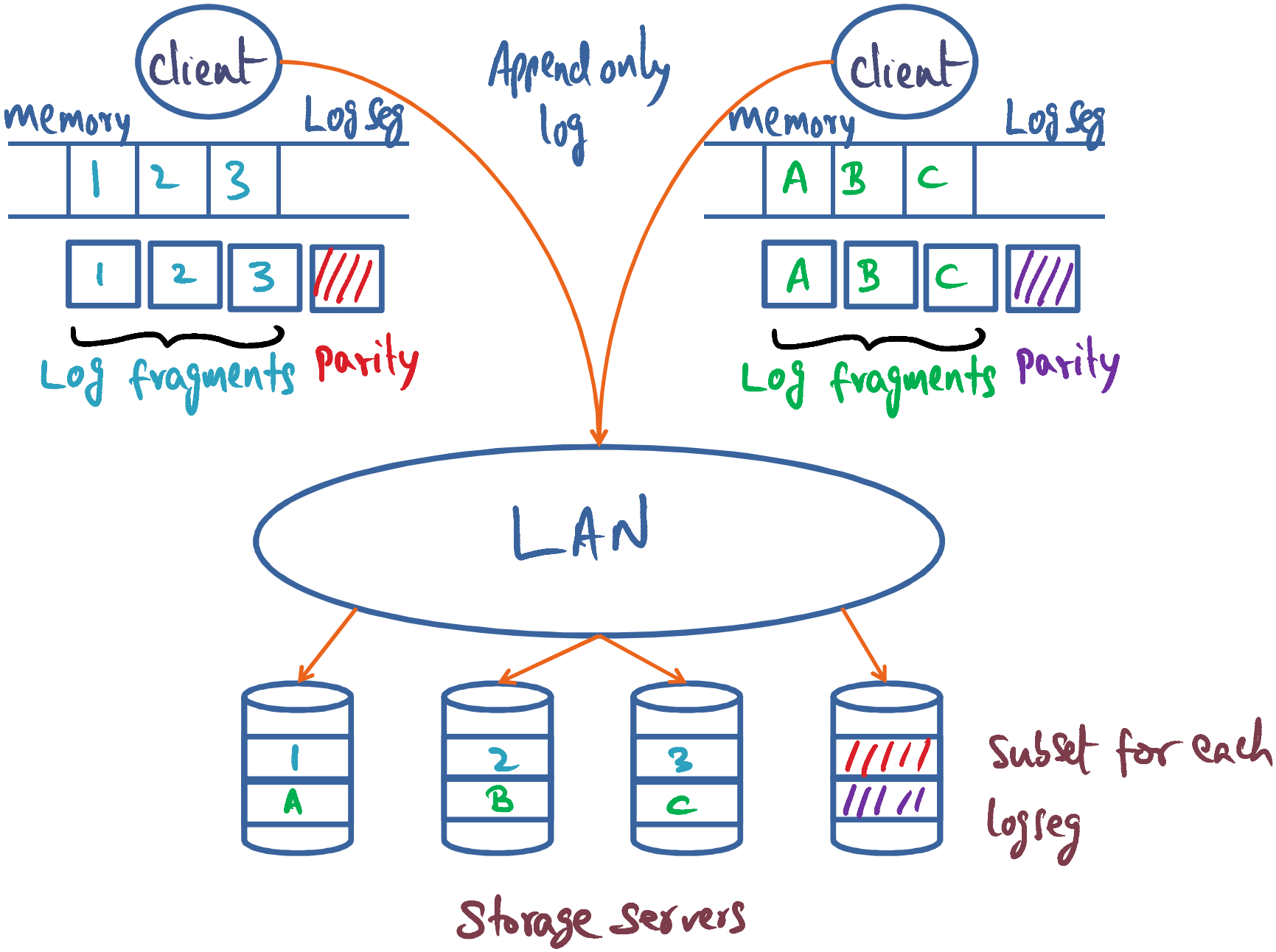
- Memory Contents
  * metadata, file cache, client caching directory



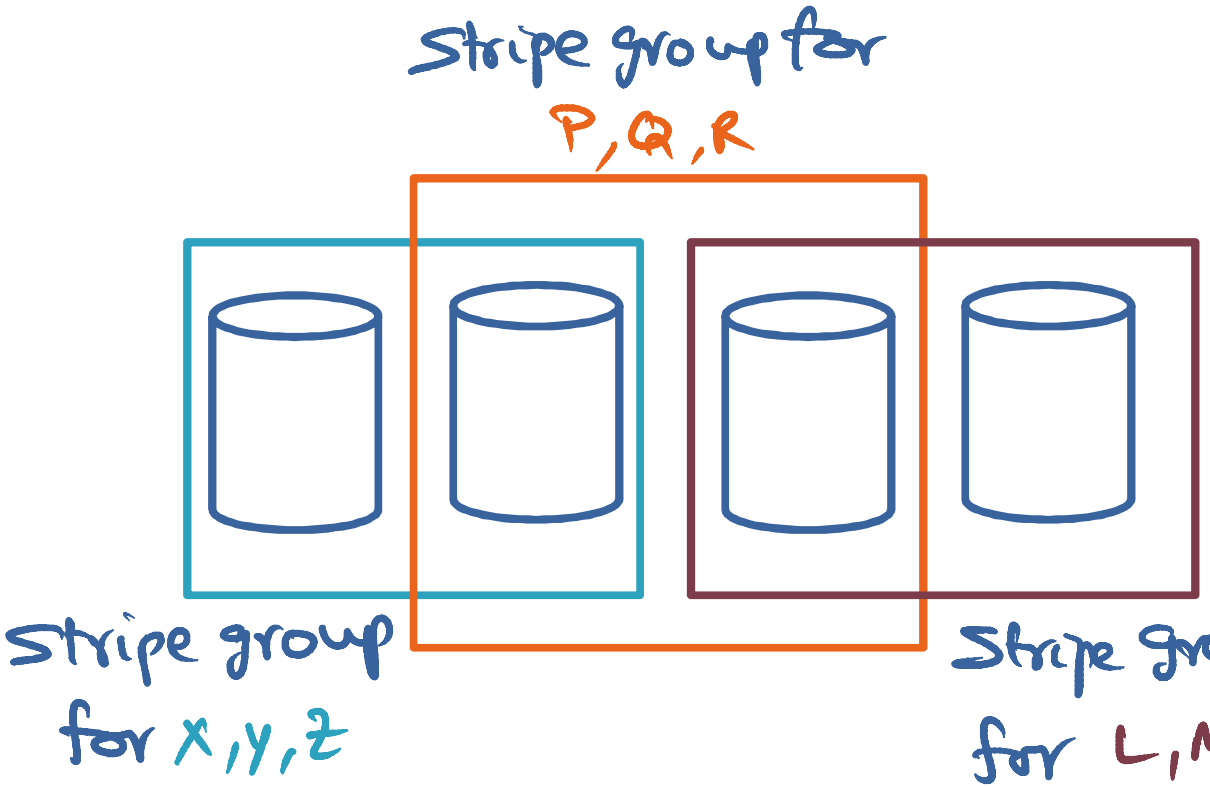
XFS

- metadata management dynamically distributed
- Cooperative client file caching

Log based Striping and Stripe groups



Stripe Group

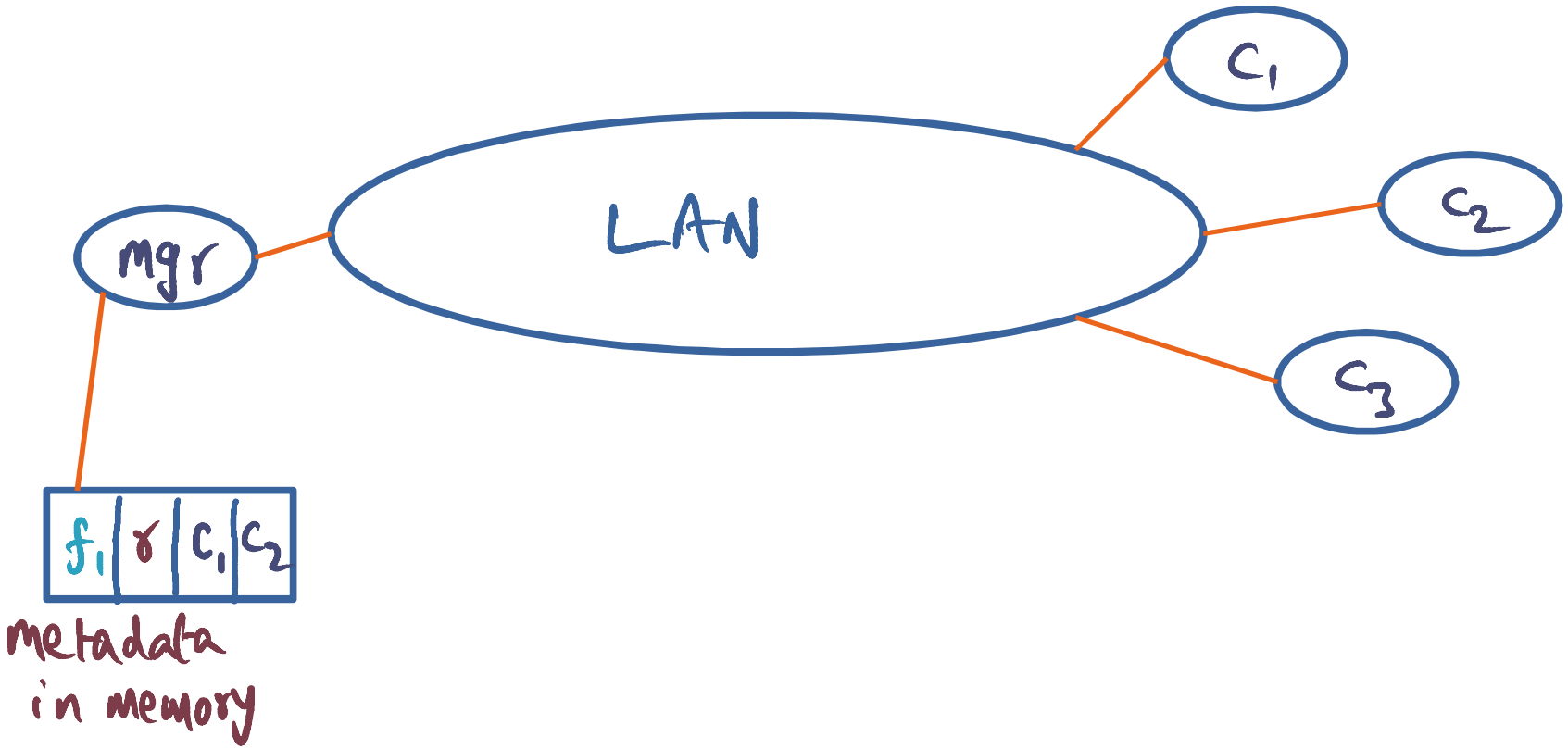


- Subset servers into stripe groups
- Parallel client activities
- increased availability
- efficient log cleaning

Cooperative Caching

Cache Coherence

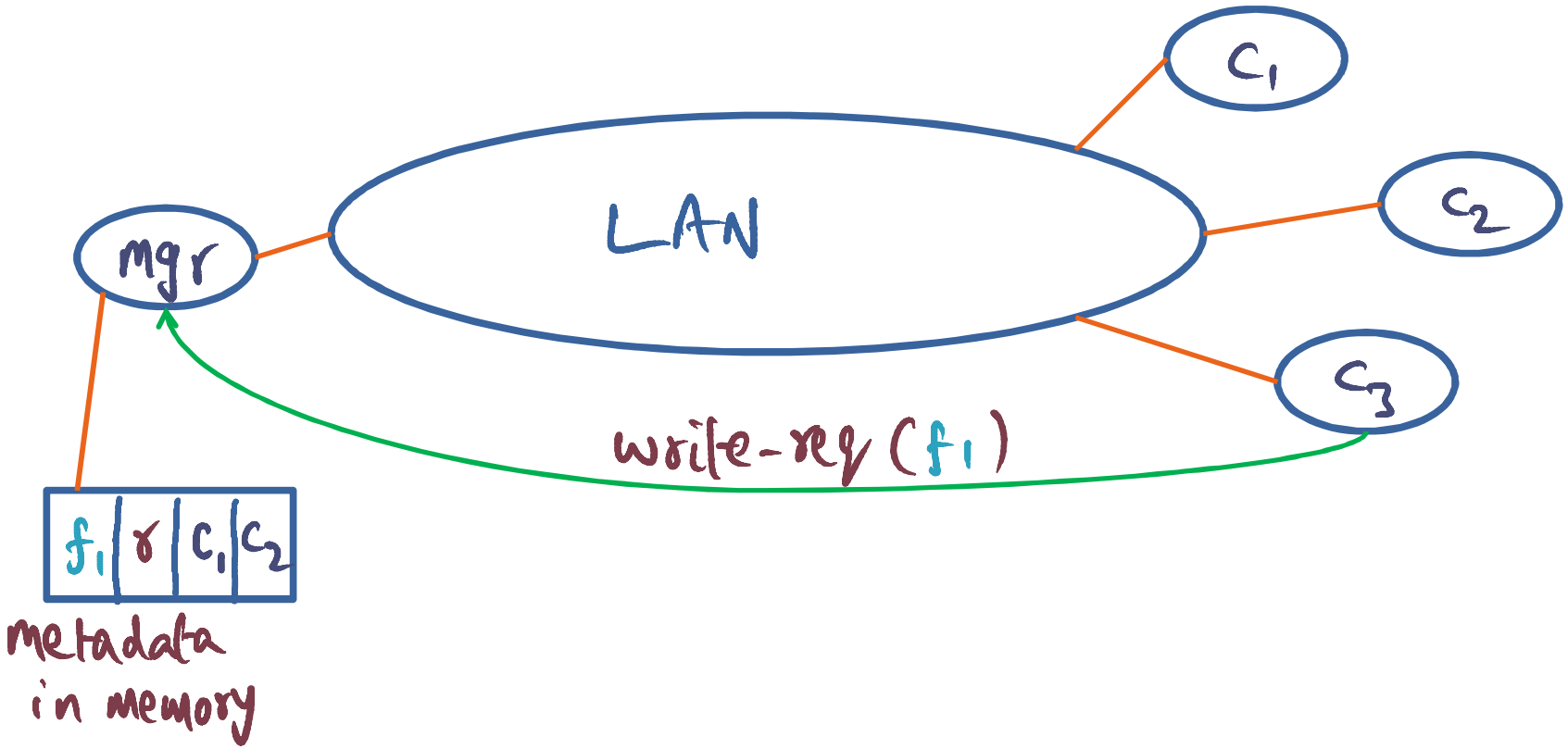
- Single writer, multiple readers
- file/lock unit of coherence



Cooperative Caching

Cache Coherence

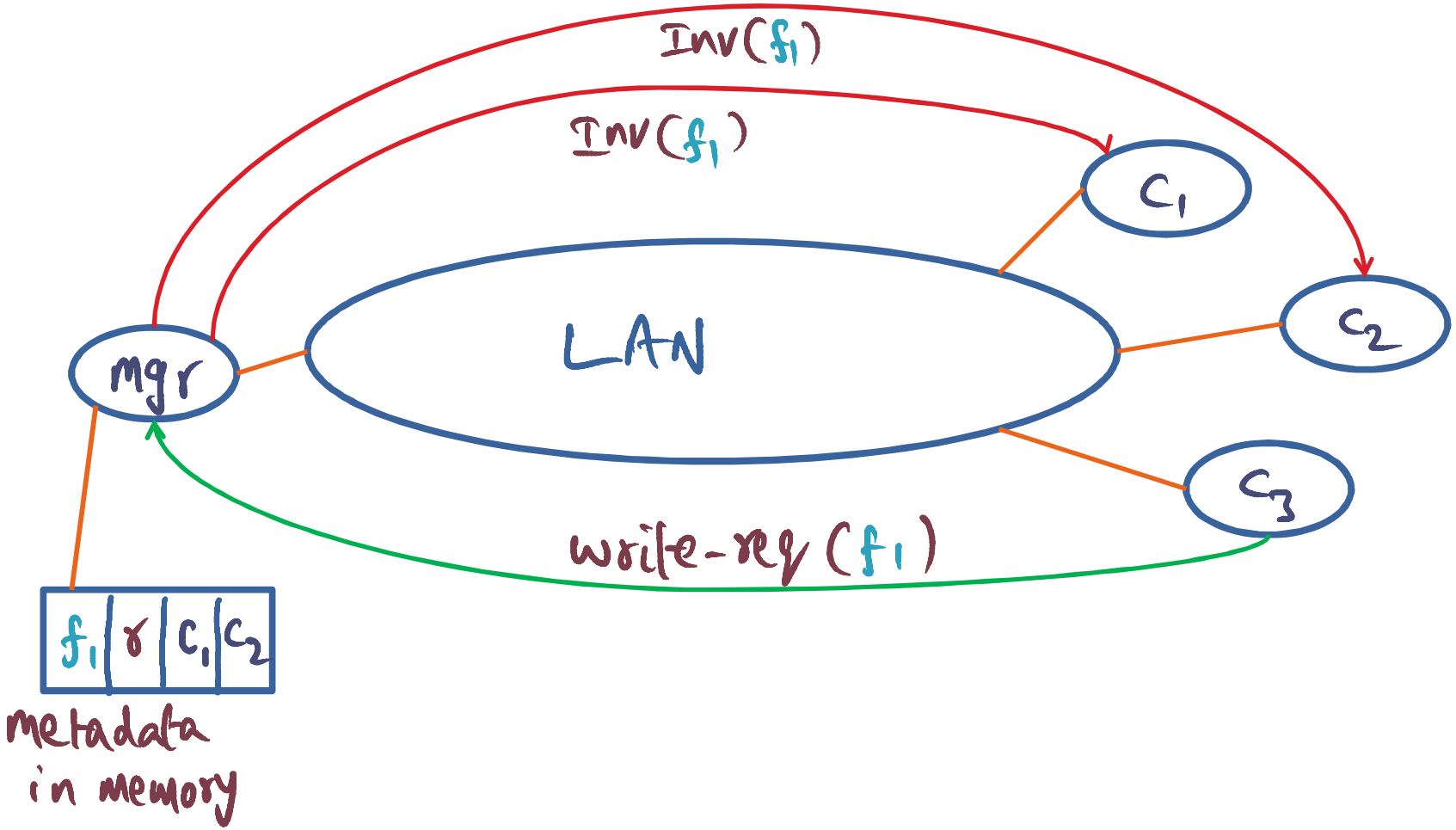
- Single writer, multiple readers
- file/block unit of coherence



Cooperative Caching

Cache Coherence

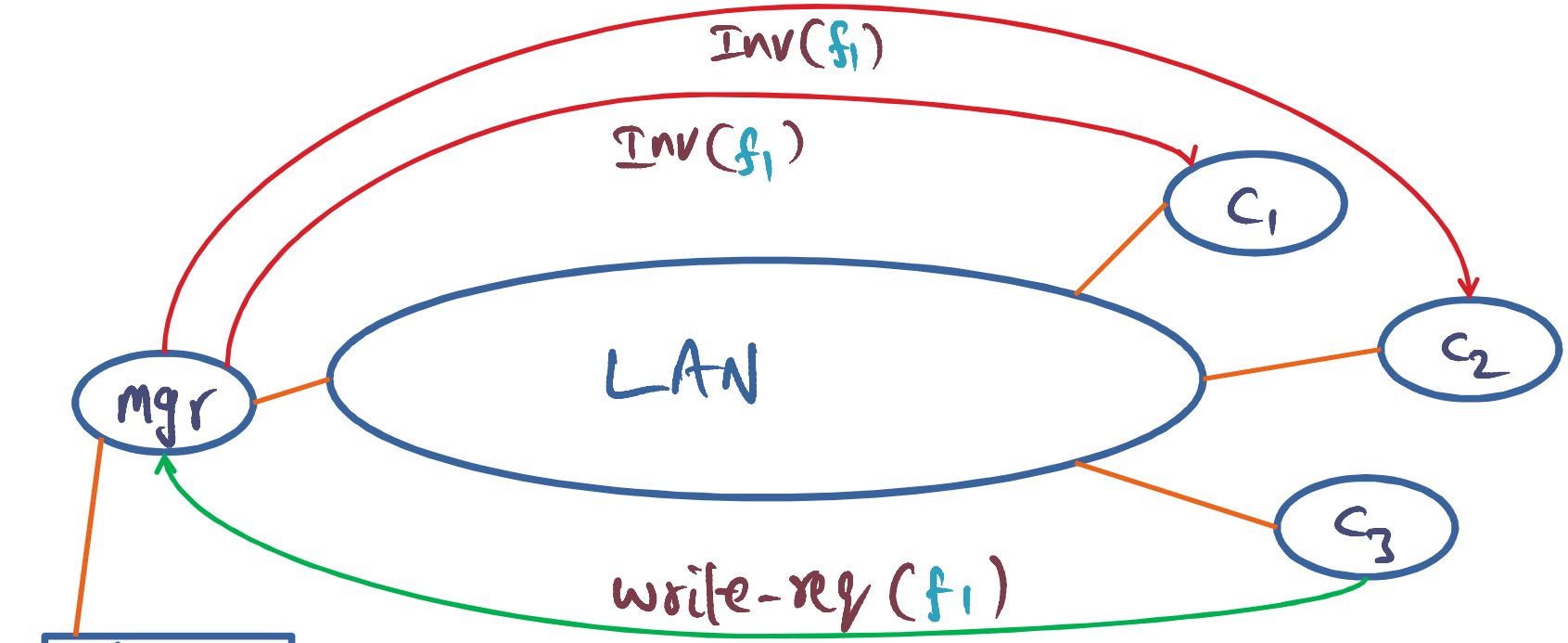
- Single writer, multiple readers
- file/block unit of coherence



Cooperative Caching

Cache Coherence

- Single writer, multiple readers
- file/lock unit of coherence



write-request

- receive token
- manager revokes on reads

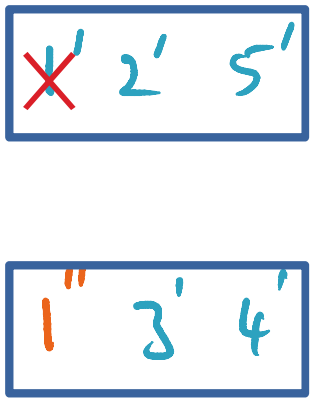
Log cleaning

Log Segment evolution

writes to file blocks 1, 2, 5
— may belong to different files

Log cleaning

Log Segment evolution



Seg1

Seg2

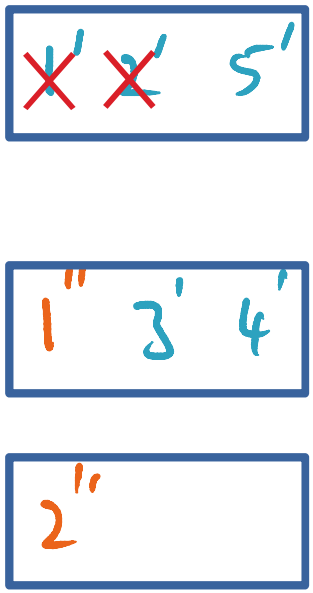
writes to file blocks 1, 2, 5
— may belong to different files

block 1 overwritten ⇒ Kill old block

writes to blocks 3 + 4

Log cleaning

Log Segment evolution



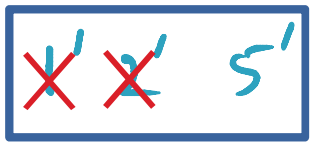
Writes to file blocks 1, 2, 5
— may belong to different files

block 1 overwritten ⇒ Kill old block
writes to blocks 3 + 4

block 2 overwritten ⇒ Kill old block

Log cleaning

Log Segment evolution



Seg 1

Writes to file blocks 1, 2, 5
— may belong to different files

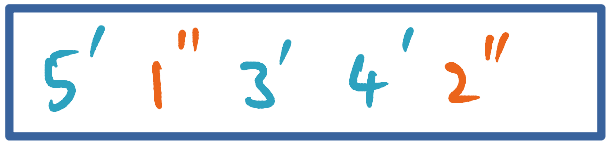
Seg 2

block 1 overwritten ⇒ Kill old block
writes to blocks 3 + 4

Seg 3

block 2 overwritten ⇒ Kill old block

⇓ Coalesce

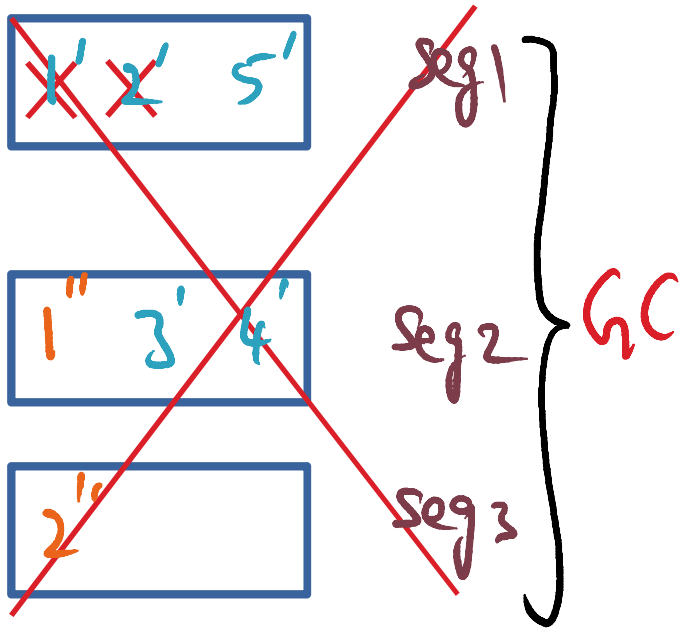


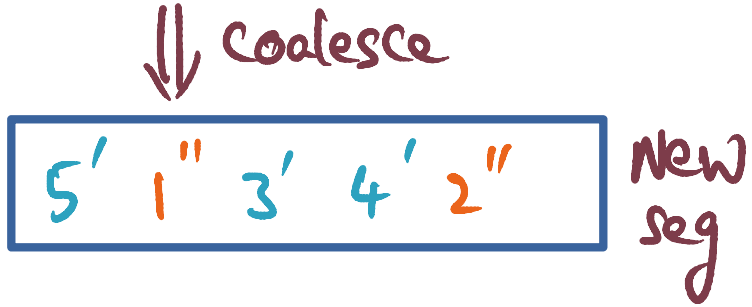
New seg

Aggregate all "live" blocks into new segment

Log cleaning

Log Segment evolution





Writes to file blocks 1, 2, 5
— may belong to different files

block 1 overwritten ⇒ Kill old block
writes to blocks 3 + 4

block 2 overwritten ⇒ Kill old block

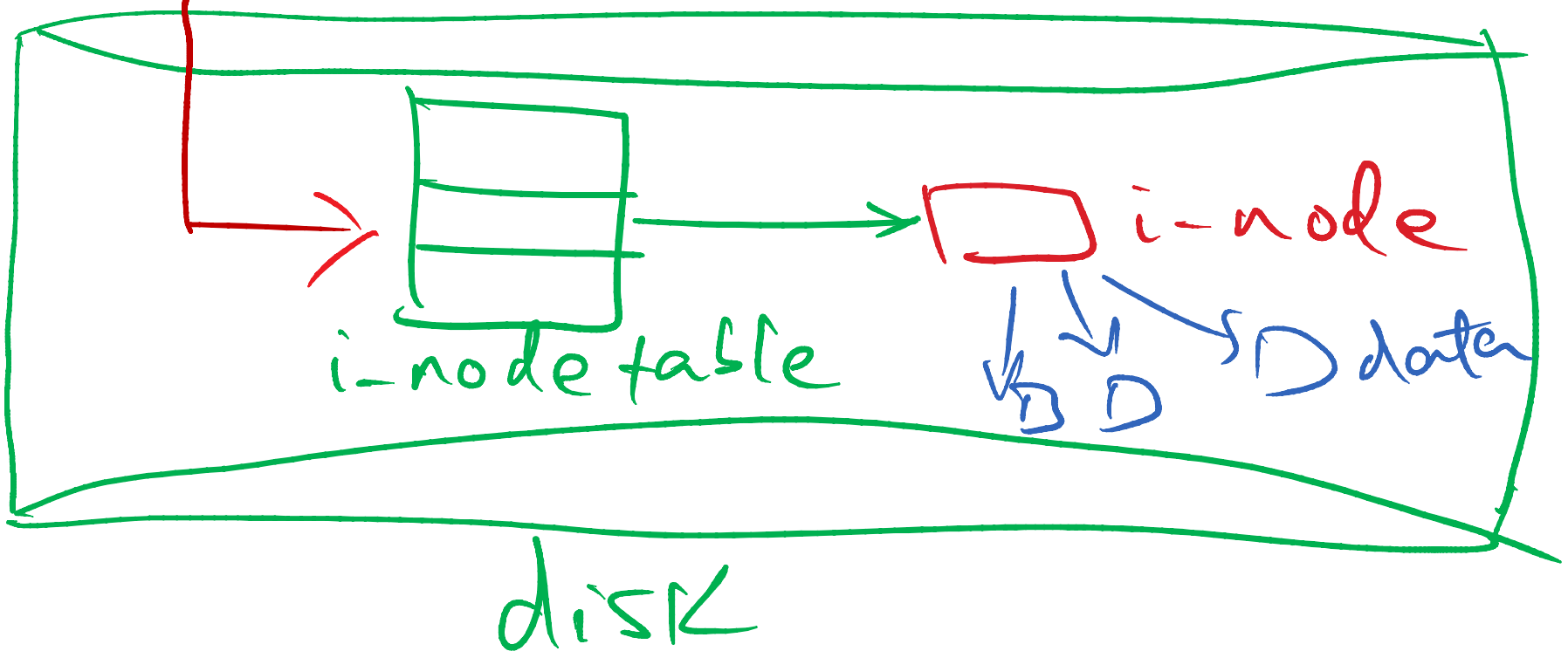
Aggregate all "live" blocks into new segment

In Unix

on file creation

— allocate i-number

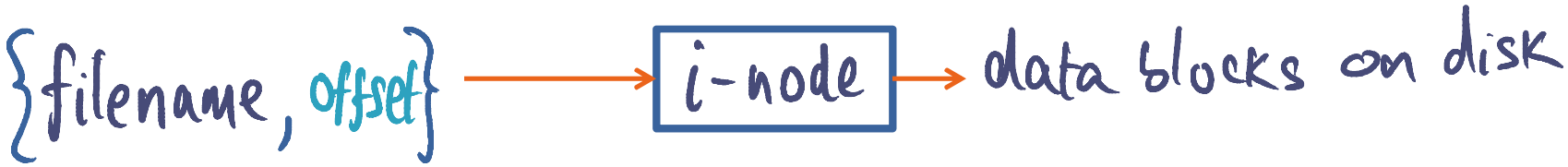
file table:
name → i-number



File System

- data structures
  persistent
- But bring them into
  memory for manipulation
- Applies to data +
  metadata (file table / i-node)

Unix File System



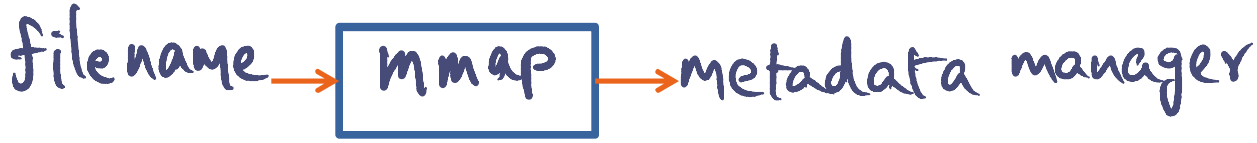
Group activity

- XFS data structures (where are they? Replicated?)
  - File dir
  - Mmap
  - Imap
  - i-node
  - Stripe group
  - logseg
  - Data blocks

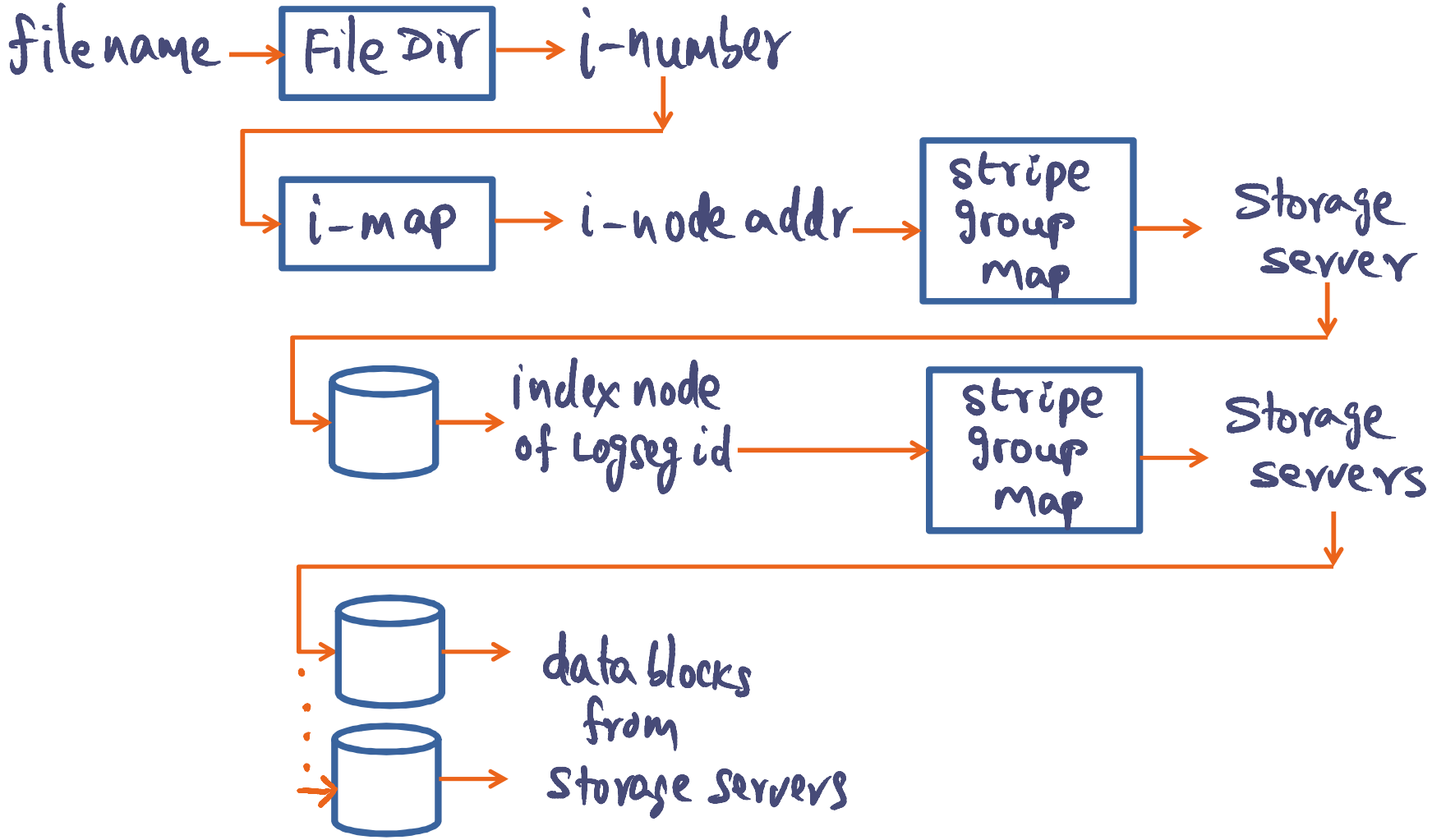
- File Dir (on the server that hosts the file) (i.e., on the physical machine where a logical volume resides)
  - Human readable name to i-number mapping
  - File name to index number; xFS stores directories in regular files, allowing a client to learn an index number by reading a directory. In xFS, the index number listed in a directory determines a file's manager. When a file is created, its index number is chosen such that the file's manager is on the same machine as the client that creates the file.
- Manager map (replicated globally)
  - Index number to manager mapping
- Imap (partitioned among managers)
  - Used by manager for file to on-disk log location mapping (split among managers)
- Stripe group map (replicated globally)
  - Log segment id to set of servers mapping

xFS data Structures

Client node action

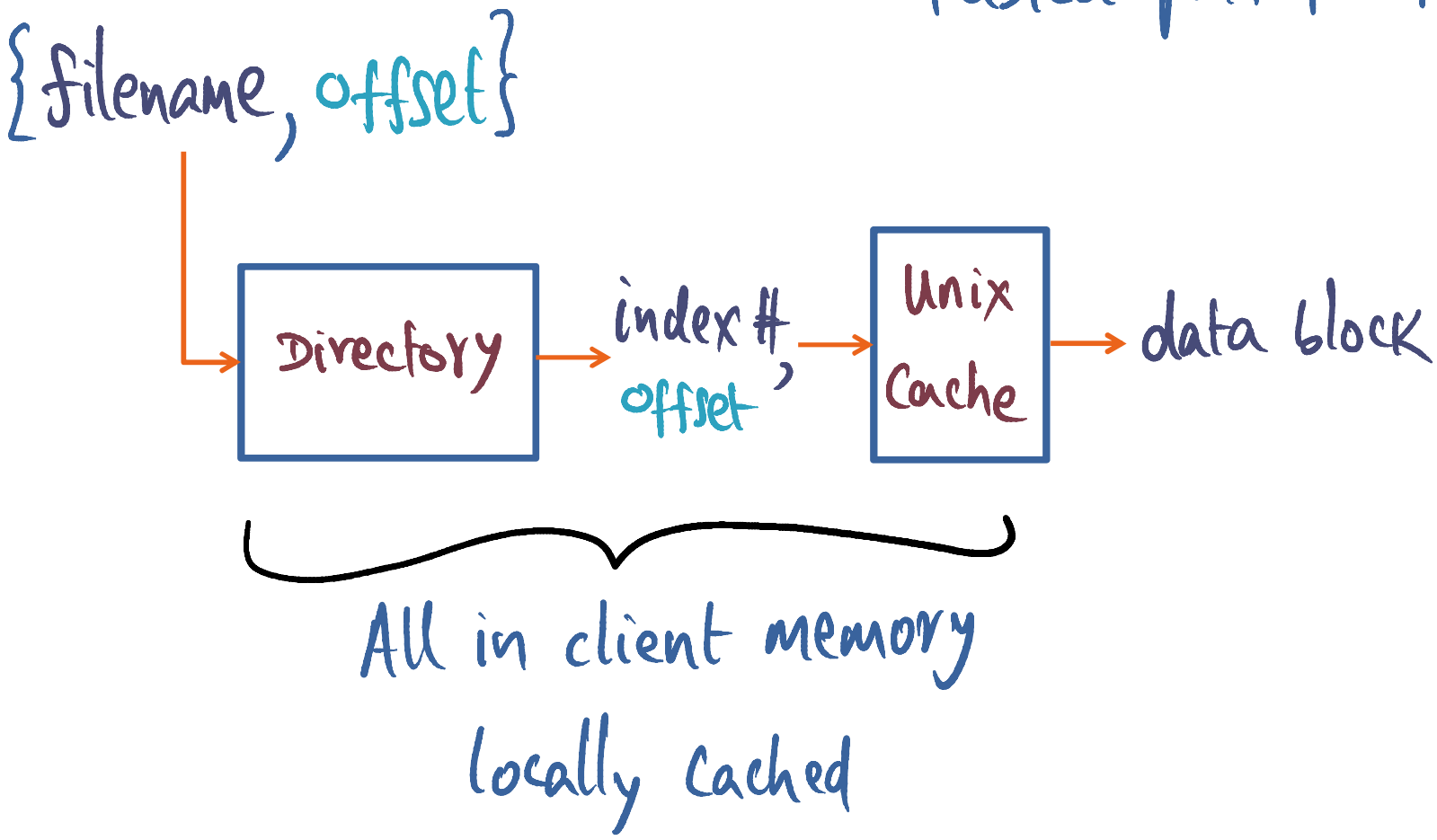


Manager node actions

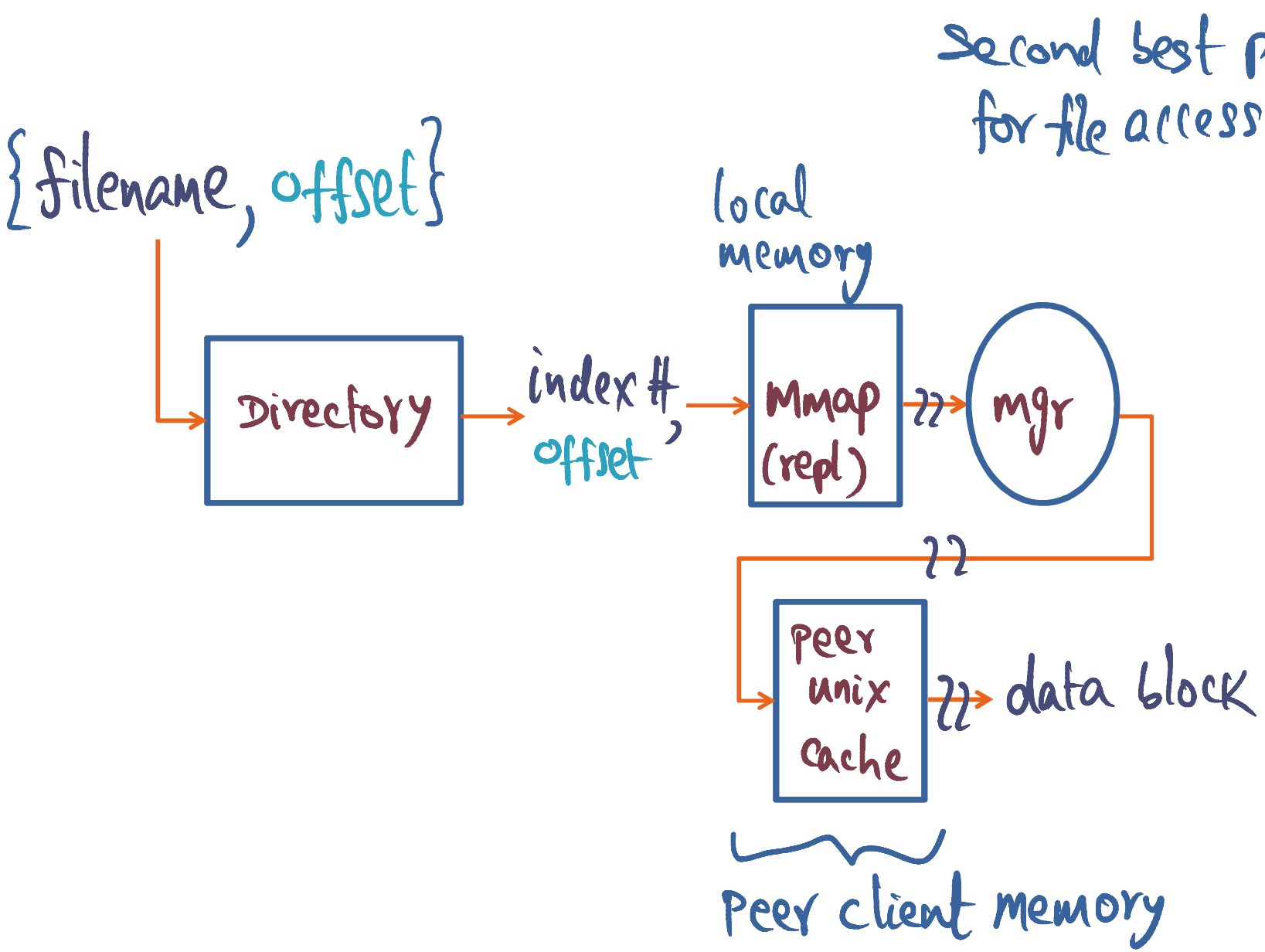


Client Reading a file – Own Cache

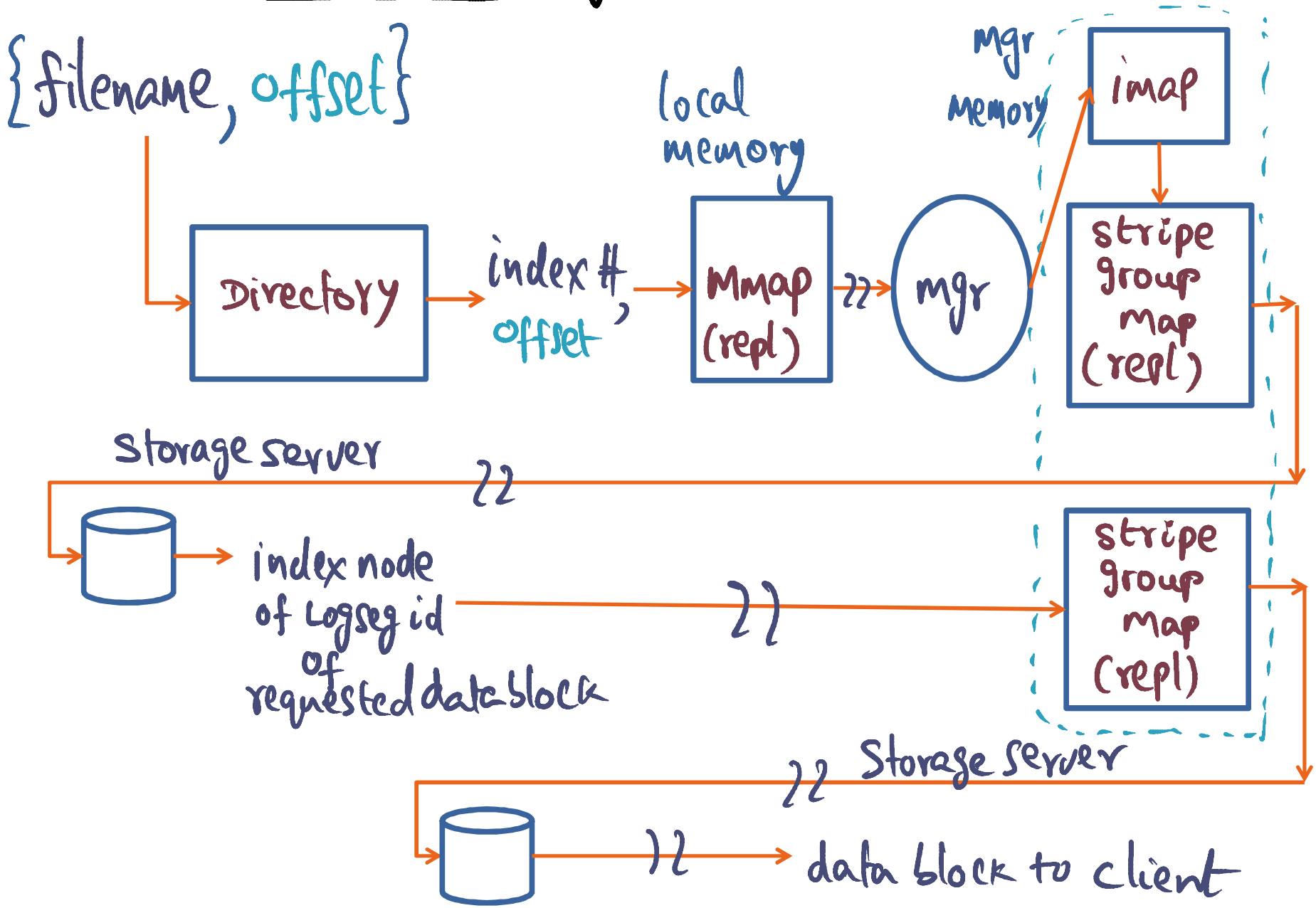
Fastest path for file access



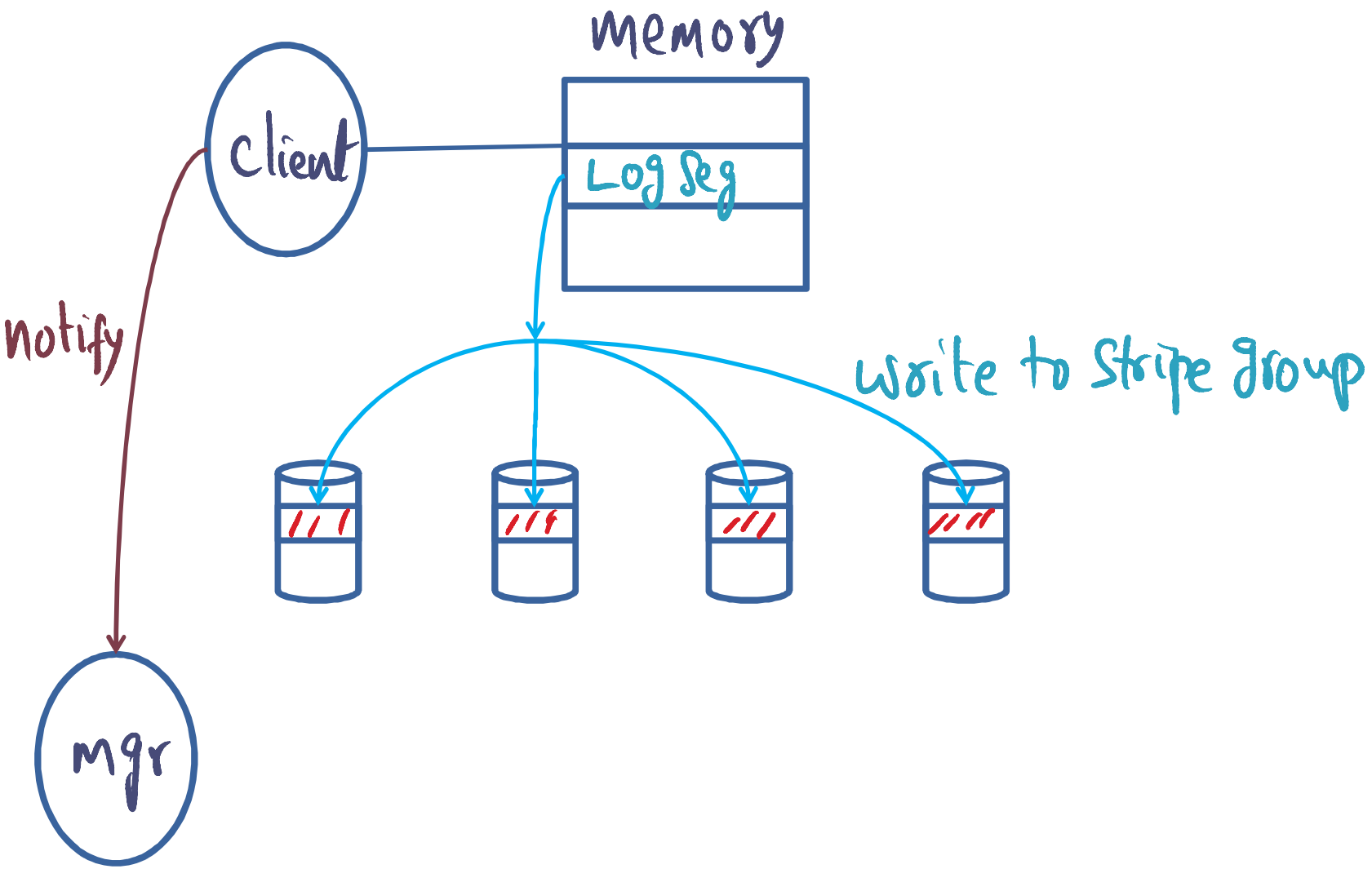
Client Reading a file – Get from peer cache



Client Reading a file – the real long way!



Client writing a file



Key Takeaways

- Network file systems are important components of any computing environment – corporate or university. There are companies that have sprung up (such as NetApp) solely to peddle scalable NFS products.
- Design and implementation of distributed file systems, in particular how to make the implementation scalable by removing centralization and utilizing the memory available in the nodes of a LAN intelligently.
- Such techniques for identifying and removing bottlenecks are the re-usable nuggets that can be taken and applied to the design and implementation of other distributed subsystems in addition to file systems themselves.

Overall, creative ways to fully utilize idle memory available in the nodes of a LAN.