DE LA RECHERCHE À L'INDUSTRIE

www.cea.fr

Lecture on Parallel Filesystems

Resiliency

Jacques-Charles Lafoucriere

ENSIIE| 2018

Basic Rule of Resiliency

ANY COMPONENT WHICH CAN FAIL WILL FAIL

Why Failure is an Issue?

To get performances we need

- Caches for local access
- Delegation for independency
- Asynchronous events to hide latencies

Distributed components have to be coherent

- Servers can reply before update on permanent storage
- Client have states in memory newer than what the server can recover from disk after crash

Main Failure Sources

- Failures
  - Disk
  - Network
  - Node
- Component failure
  - Client
  - MDS failure (and failover)
  - OST failure (and failover)
  - Transient network partition

Origin of Failures

Hardware Failure

- Total
- Partial

Loss of Communications

- Network reboot
- Load
  - Network, client, server …

Data Corruption

Bugs

Lustre Client Failure

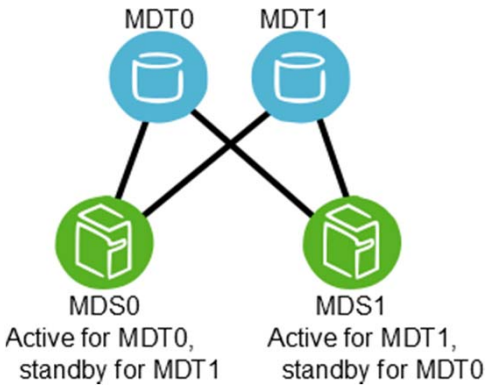
Recovery is based on lock revocation

- If a client does not respond
  - To call back from DLM
  - To ping from servers
- It is evicted
  - Need to reconnect
  - Need to reacquire all locks

Server Failure

MDS recovery is based on

- Shared storage
- STONITH
  - Shoot The Other Node In The Head
- Support Active/Active model



OSS recovery is based on

- Shared storage
- STONITH
  - Shoot The Other Node In The Head
- Support Active/Active model
- OST unavailability is detected by MDS which stop use it for new files
- OST can be set RO

STONITH Challenges

How to detect a pair node is dead?

- Multiple communication links

How to be sure the node pair is killed?

- Remote management of power distribution
- Avoid cross detection issue
  - A detects B is dead
  - B detects A is dead
  - A and B kills B and A …

Automatic HA is difficult to setup

- Manual HA is safer but require a human action

Server Failure (Cont.)

Server updates are asynchronous

- In case of server crash it may have acked a request and lost it after
  - Client uses Replay

Server can crash before ack

- Client uses Resend
- Client uses XID numbers and transaction numbers to keep track of
  - Network requests
  - Committed transactions

Server Failure (Cont.)

Another Solution

- Data duplications
- Used by Hadoop FS or Google FS
  - Each file has 3 copies
  - 2 in the same rack
  - 1 in a distant rack
- Consistent replication for mutable data is hard
  - Writes have to reach every replica
  - Must be done in certain ways (order matters), done in the same order on all replicas, or if the writes are commutative …
- Suppose disks are very cheap
- Better for load where Reads dominate

Quorum

Quorum-based voting can be used as

- A replica control method
  - When the same information has to be updated in multiple places
  - 2 votes: Read and Write
- A commit method to ensure transaction atomicity in the presence of network partitioning
  - A transaction is executed if the majority of sites vote to execute it
  - 2 votes: Abort and Commits

CAP Theorem

- It is impossible for a distributed data store to simultaneously provide more than two out of the following three guarantee
  - Consistency
    - Every read receives the most recent write or an error
  - Availability
    - Every request receives a (non-error) response
  - Partition Tolerance
    - The system continues to operate despite an arbitrary number of messages being dropped (or delayed)
- In the presence of a network partition, one has to choose between consistency and availability

File System Coherency

What is FS Coherency

A PFS is made of many independent components

- Namespace
- Metadata
  - File Layout
- Data objects

All these components need to be coherent

- File must have a valid layout
- Layout must point to valid objects
- Layout must point to the right objects
- All objects must belong to a file

How to Guaranty the Coherency?

Offline checking

- During some maintenance time
- Run a FS check tool
  - Could take days

Online checking

- Run while the file system is mounted
- Does no impact file system use

Lustre Example: lfsck

lfsck features

- Check and repair the Object Index table
  - Map Lustre FID to MDT ldiskfs inodes
- Check and repair the directory FID-in-dirent and LinkEA consistency
- Check and repair MDT-OST consistency
  - Unreferenced OST objects
  - Mismatched references
  - Multiple references
- Check and repair multiple MDT incoherencies

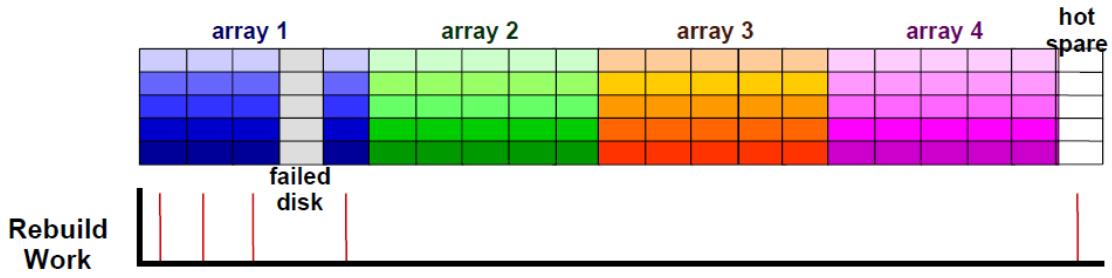
- Checked targets/rules can be controlled

RAID

Conventional RAID

Conventional RAID Limits RAID Rebuild Performance

- RAID rebuild operations span a limited number of disks because a RAID group maps to a small subset of pdisks. This provides fewer disks for RAID rebuild operations.
- Example: 21 disks @ 4 x 3+P+Q RAID-6 arrays with 1 hot spare

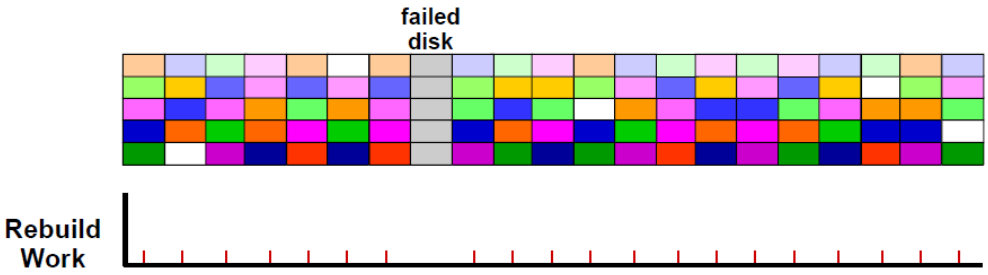


Only 5 disks participate in the rebuild operation - the 4 good disks and the hot spare.

Declustered RAID explained

Declustered Arrays Improve RAID Rebuild Performance

- RAID rebuild operations span all disks in the DA because the RAID groups span all disks in the DA. This provides more disks for RAID rebuild operations.
- Example: 21 disks configured with 1 virtual spare disk
  - Data from any virtual 3+P+Q vdisk is distributed across all pdisks in DA



All 20 surviving disks participate in the rebuild.

Thank you for your attention

ENSIIE | 2018

Commissariat à l'énergie atomique et aux énergies alternatives
Centre DAM-Ile de France | 91297 Bruyères-le-Châtel Cedex
T. +33 (0)1 69 26 40 00 | F. +33 (0)1 69 26 70 86
Direction des applications militaires
Département sciences de la simulation et de l'information
Service informatique scientifique et réseaux