SC24 Atlanta, GA | hpc creates.

PDSW '24

Silent Errors to Scientific Applications: Impacts of PFS Metadata Corruptions

Saisha Kamat[1], Dong Dai[2], Mai Zheng[3], Bo Fang[4]

[1] University of North Carolina at Charlotte, [2] University of Delaware,

[3] Iowa State University, [4] Pacific Northwest National Laboratory

Background : HPC Applications and PFS Failures

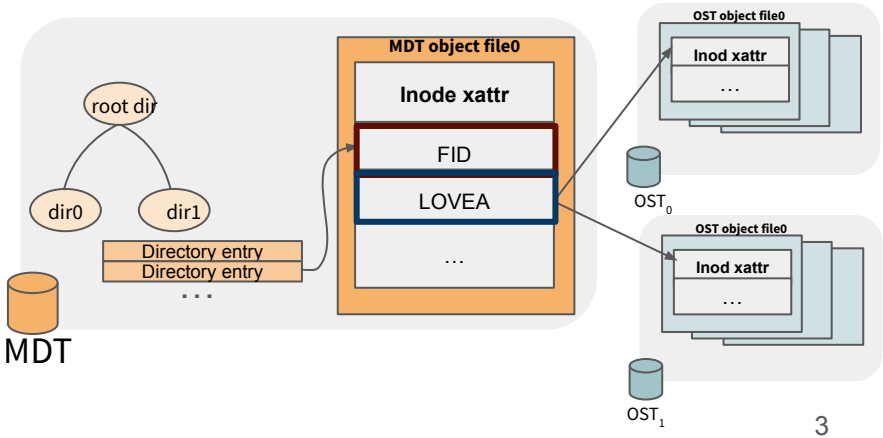
- Parallel File Systems (PFS) are essential blocks of HPC.
- HPC Applications depend on PFS to ensure efficient and reliable data access.
- PFS failures can occur and potentially impact applications, leading to errors.

- Previous studies on how corruptions impact applications focus on
  - Data corruptions (bit flip, shorn writes, dropped writes)

- What about Metadata corruptions?
  - Updated frequently, hence easy to be corrupted or inconsistent.
  - May lead to more complicated consequences to applications.

Impact of PFS Metadata Corruption: Fault Model

- Lustre Architecture Overview
  - Metadata Target (MDT): Stores metadata ; Object Storage Target (OST): Stores file data objects and related metadata
- Key PFS Metadata (Essential for File Access)
  - FID: Unique 128-bit file identifier
  - LOVEA: Contains stripe information for locating file data on OSTs
- Fault Model
  - Represents common disk corruption scenarios

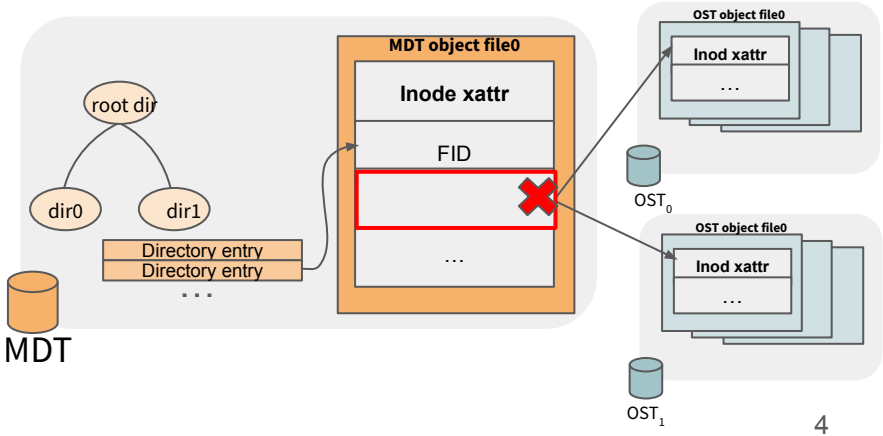


Corruption type	Description
Missing	Metadata is deleted
Random	Metadata replaced with random value
Swap_similar	Metadata swapped with a similar file's metadata
Swap_random	Metadata swapped with a random file's metadata

Table 1: Fault Model

Impact of PFS Metadata Corruption: Fault Injection

- Select application input file
  - **Nyx** : Adaptive mesh refinement (AMR)-based cosmological simulation;
    - Single input file : “NVB..z5.hdf5”
  - **Montage** : Astronomical image mosaic creation
    - One file randomly select from ten: “2mass-atlas..21.fit”
- Inject Fault
  - Corrupt a metadata field (eg. LOVEA Missing)
  - Run the application



Corruption type	Description
Missing	Metadata is deleted
Random	Metadata replaced with random value
Swap_similar	Metadata swapped with a similar file's metadata
Swap_random	Metadata swapped with a random file's metadata

Table 1: Fault Model

Impact of PFS Metadata Corruption: App Behaviour

- Applications respond in different ways
  - Normal Output: Unaffected by fault
  - Failures: Some cases stop ; **many cases stalls or show silent errors (e.g. incorrect/partial processed outputs)**
- Challenges with fault handling
  - Applications fail to detect faults, continue with silent errors
  - Running LFSCK-Lustre's dedicated file system checker is unable to fix issues (not shown in this WiP)

Behaviour Type	Description
Normal	Process all files- No fault impact; Correct output
Stops	Stops abruptly due to fault; No output
Stalling	Stalls indefinitely ; No output
Wrong output	Processes all files; Fault cause Wrong output
Partial Output	Skips corrupt file; Fault cause partial processing

Table 2: Application Behaviour Description

	Applications	Missing	Random	Swap_similar	Swap_random
FID	**Nyx**	Normal	Stalling	Stalling	Stalling
	Montage	Normal	Stalling	Stalling	Stalling
LOVEA	**Nyx**	Stops	Stops	Wrong output	Stops
	Montage	Partial output	Partial output	Wrong output	Partial output

Table 3: Application Behaviour due to Metadata Faults

Proposed Solution : Checksum tool

- Checksum Prototype overview
  - Detect metadata faults at runtime
  - Intercepts file system using FUSE for metadata checksum verification
  - Stops application upon detecting corruption **(prevent silent errors)**
- Evaluations
  - Functionality: Successfully detects metadata corruption and stops application
  - Performance: Significant overhead due to FUSE-based implementation

	Applications	All four fault cases
FID	**Nyx**	Stops
	Montage	Stops
LOVEA	**Nyx**	Stops
	Montage	Stops

Table 4: Functionality Evaluation

Conclusion and Future Work

- Metadata corruptions causes silent errors; both PFS and application do not handle effectively
- Proposed checksum tool can prevent silent errors
- In future, we plan to optimize checksum tool performance

Thank you

Application Behaviour w/ & w/o LFSCK Run

	Applications	Missing	Random	Swap_similar	Swap_random
FID	**Nyx**	Normal	Stalling	Stalling	Stalling
	Montage	Normal	Stalling	Stalling	Stalling
LOVEA	**Nyx**	Stops	Stops	Wrong output	Stops
	Montage	Partial output	Partial output	Wrong output	Partial output

Table: Behaviour Without LFSCK Run

	Applications	Missing	Random	Swap_similar	Swap_random
FID	**Nyx**	Stop	Normal	Wrong output	Stops
	Montage	Partial output	Normal	Wrong output	Partial output
LOVEA	**Nyx**	Stops	Stops	Stops	Stops
	Montage	Partial output	Partial output	Partial output	Partial output

Table: Behaviour With LFSCK Run

Application Behaviour w/ & w/o LFSCK Run

	Applications	Missing	Random	Swap_similar	Swap_random
FID	**Nyx**	Normal	Stalling	Stalling	Stalling
	Montage	Normal	Stalling	Stalling	Stalling
LOVEA	**Nyx**	Stops	Stops	Wrong output	Stops
	Montage	Partial output	Partial output	Wrong output	Partial output

Table: Behaviour Without LFSCK Run

	Applications	Missing	Random	Swap_similar	Swap_random
FID	**Nyx**	Stop	Normal	Wrong output	Stops
	Montage	Partial output	Normal	Wrong output	Partial output
LOVEA	**Nyx**	Stops	Stops	Stops	Stops
	Montage	Partial output	Partial output	Partial output	Partial output

Repair creates new MDT: empty file

Repair restores Correct FID

Unable to detect: No repair

Repair mimics content of similar/random file

Repair creates new OSTs: empty file

Table: Behaviour With LFSCK Run