FFUSE: I/O Fault Injection and Resilience Scoring for HPC and AI Workloads

Anila Ghazanfar · ***Julian Kunkel (advisor)***

Georg-August-Universität Göttingen (GWDG)

Same faults, different outcomes: checkpoint format was associated with how workloads fail under I/O-layer faults.

LAMMPS / HDF5: 29.6% crash, FIORE 0.333 vs. ResNet-50 / PyTorch pickle: 51.9% crash, FIORE 0.222 — 52 fault-injection tests (26 per workload) across 7 phases — neither workload reported silent bit-level corruption (0%)

Research Problem & Motivation

- Scientific simulations (LAMMPS) and AI training (ResNet-50) increasingly share HPC storage.
- I/O faults cause crashes, silent data corruption, and incorrect results, with very different consequences for each domain.
- **Goal**:
  - Compare HPC and AI resilience under identical, controlled I/O fault conditions.
  - Develop a unified metric for cross-domain resilience assessment.

Related Work

- **FFIS [1]:** Metadata integrity; limited fault types.
- **PFault [2]:** Black-box node-level fault emulation for parallel filesystems.
- **TRANSOM [3]:** LLM resilience; focuses on process and hardware faults.
- **AI fault-tolerance studies ([4], [5]):** SDC via checkpoint modification or accelerator faults.

The FFUSE Framework

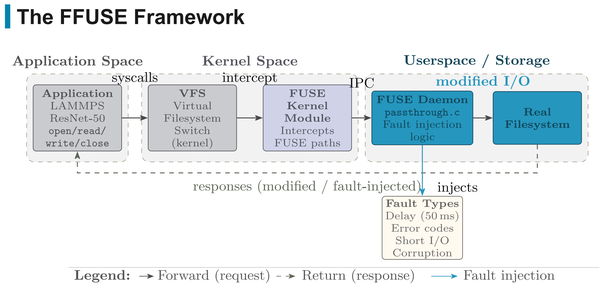


Figure 1. FUSE architecture with fault injection at the userspace daemon.

Evaluation Metrics

$$\text{Crash Rate} = \frac{N_{\text{crash}}}{N_{\text{total}}} \times 100$$

$$\text{Failure Rate} = \frac{N_{\text{failure}}}{N_{\text{total}}} \times 100 = \frac{N_{\text{crash}} + N_{\text{silent}}}{N_{\text{total}}} \times 100$$

$$\text{Silent Failure Rate} = \frac{N_{\text{silent}}}{N_{\text{total}}} \times 100$$

$$\text{Error Detection Rate} = \frac{N_{\text{reported}}}{N_{\text{total}}} \times 100$$

$$\text{Data Integrity Rate} = \frac{N_{\text{hash_ok}} + N_{\text{op_ok}}}{N_{\text{total}}} \times 100$$

where

- N_{total} = total runs.
- N_{crash} = total crashed runs.
- $N_{reported}$ = number of failed test runs with error report.
- $N_{failure}$ = number of failed test runs, due to crash or silent corruption.
- N_{silent} = runs with silent corruption.
- N_{hash_ok} = SHA-256 match to baseline.
- N_{op_ok} = integrity flag passed (incomplete outputs).

Seven-Phase Fault Injection Workflow

The orchestrator iterates over seven fault phases per workload, from a clean baseline through cascading and silent-corruption faults, then merges results and computes FIORE.

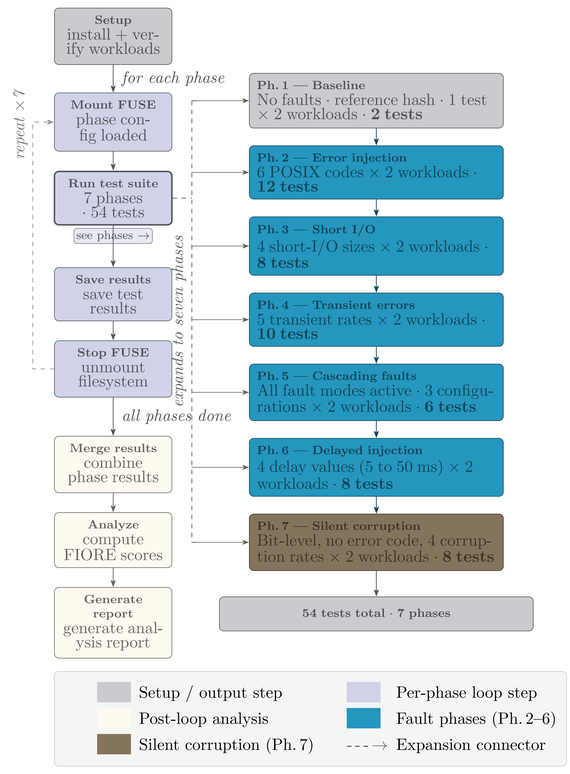


Figure 2. FFUSE execution workflow, 7 phases, 54 tests total including 52 fault injection tests and 2 baseline tests.

The FIORE (Fault Injection Observed Resilience Evaluation)

Measures reporting quality among failures, not overall resilience.

$$\text{FIORE} = \frac{N_{\text{unreported}}}{N_{\text{failure}}}$$

$$\text{Reporting Coverage} = \frac{N_{\text{reported}}}{N_{\text{failure}}} = 1 - \text{FIORE}$$

where

- $N_{unreported}$ = number of failed test runs with no recorded error report.
- $N_{failure}$ = number of failed test runs, due to crash or silent corruption.
- FIORE = 0 means every failure was reported;
  = 1 means no failures were reported.
  = N/A if there were no failures.
- Reporting Coverage = 1 means good reporting coverage.
  = 0 means poor reporting coverage.

Experimental Setup

Two representative workloads:

- **LAMMPS (HPC):** molecular dynamics simulator, checkpoints via HDF5.
- **ResNet-50 (AI):** PyTorch image classifier, checkpoints via direct pickle serialization (.pt).

Checkpoint format at a glance:

	LAMMPS	ResNet-50
Format	HDF5	Pickle (.pt)
Buffering	Yes	No
Metadata check	Yes	Minimal
Priority	Robustness	Speed

Key Findings

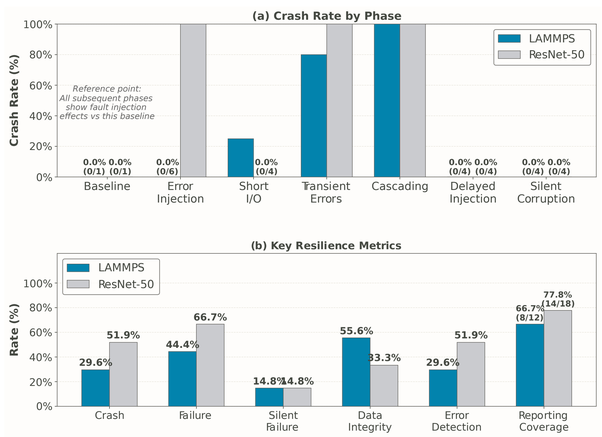


Figure 3. Crash rate by phase, and key resilience metrics.

Shared Observed vulnerabilities (both workloads):

100% crash on cascading faults

0% crash on silent-corruption

Silent corruption was only caught via post-execution SHA-256 hash comparison — not by either application.

Domain-specific behavior:

LAMMPS — Resilient

- Tolerated partial I/O and latency faults better than ResNet-50; failures were typically explicit.
- 4/12 unreported failures.

ResNet-50 — Fragile

- Short I/O and transient errors caused checkpoint corruption, indicating silent state divergence.
- 4/18 unreported failures.

FIORE Scores

LAMMPS (HPC): 0.333 (4/12), 95% Wilson CI [0.138, 0.609]

ResNet-50 (AI): 0.222 (4/18), 95% Wilson CI [0.090, 0.452]

Checkpoint Format Ablation (Pilot)

Design: One ResNet-50 CIFAR-10 training/resume path, same FUSE injector, state, and seed; three serializers (.pt, HDF5, SafeTensors); 1 trial per format and scenario (15 cases).

Format	Baseline	Silent corruption (10% per write)	Short I/O (512 bytes)	Transient errors (10%)	POSIX error (EIO)
PyTorch .pt	Exact recovery	Resumed; final state differs (41 injected events)	Not decoded	Write failed	Write failed
HDF5	Exact recovery	Not reopened (89 injected events)	Not decoded	Write failed	Write failed
SafeTensors	Exact recovery	Resumed; final state differs (31 injected events)	Not decoded †	Write failed	Write failed

Figure 4. Outcome by format and fault scenario (n=1 per cell). All formats recovered exactly at baseline; no injected-fault case did. Silent-corruption cells give injected events (41 / 89 / 31), which differed by serializer. † Reader terminated with SIGBUS.

Key Takeaway

- Silent corruption: unreported in both workloads, caught only by hash verification.
- Outcomes differed by workload and checkpoint format; the pilot suggests format can change recovery.
- Format as the main determinant: untested, needs exposure-matched, multi-seed runs.

Limitations & Future Work

- Filesystem-layer injection only (no hardware or firmware faults).
- CPU-only, two workloads, wide 95% CIs.
- Format pilot: n=1, shortened, unequal exposure.
- **Future Work**: exposure-matched, multi-seed format ablations; Lustre/GPFS; GPU memory faults; CI/CD resilience regression testing.

Open source code

Acknowledgments

Supported by an NHR Graduate School Fellowship (Göttingen GWDG), funded by the BMFTR and the state governments participating in the NHR.

With funding from the: Federal Ministry of Research, Technology and Space and the state governments participating in the NHR

References

[1] B. Fang et al. 2021. Characterizing impacts of storage faults on HPC applications. IEEE CLUSTER, 409–420. doi:10.1109/CLUSTER48925.2021.00048

[2] R. Han et al. 2022. A study of failure recovery and logging of high-performance parallel file systems. ACM Trans. Storage 18, 1–44. doi:10.1145/3483447

[3] B. Wu et al. 2023. TRANSOM: An efficient fault-tolerant system for training LLMs. arXiv:2310.10046.

[4] Y. He et al. 2023. Understanding and mitigating hardware failures in deep learning training systems. ISCA '23. doi:10.1145/3579371.3589105

[5] E. Rojas, D. Pérez, E. Meneses. 2022. Exploring the effects of silent data corruption in distributed deep learning training. SBAC-PAD, 21–30. doi:10.1109/SBAC-PAD55451.2022.00013

Anila Ghazanfar
anila.ghazanfar@stud.uni-goettingen.de

Julian Kunkel (advisor)
julian.kunkel@gwdg.de