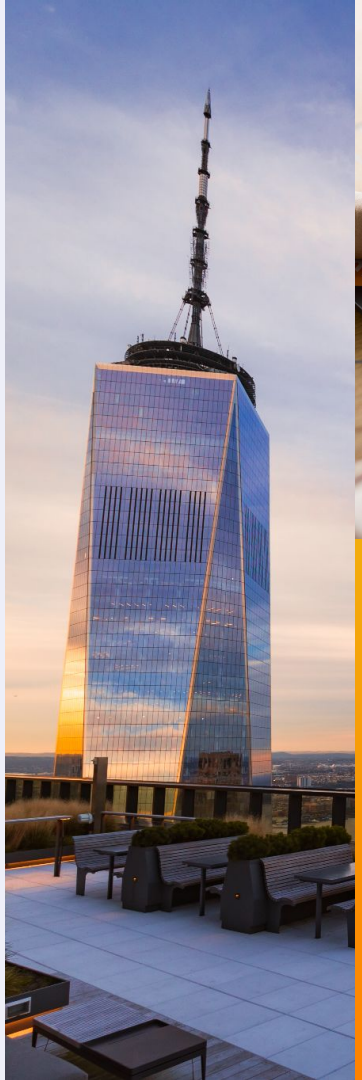
Lustre DNE3

Busting the Small Files Myth

Steve Crusan and Brock Johnson

LUG 2025 Lustre User Group

HRT hudson river trading



Agenda

Topics

- Who is HRT?
- Mythbusting Lustre small file / metadata performance
- Design decisions for better metadata performance/scaling
- Lustre Wishlist
- Summary

About HRT

Hudson River Trading (HRT) is a **multi-asset class quantitative trading firm** that provides liquidity on global markets and directly to our clients.

Not Traditional HPC
We go beyond classic checkpoint/restart workloads.

Flexible Storage Needs
Supports loosely coupled, parallel workloads.

User-Centric Approach
No rigid pipelines; users can experiment freely.

NFS Experts
Deep understanding of consistency and coherence models.

High Reliability
Downtime and built-in performance issues are unacceptable.

How We Use Lustre

Generalized Scratch Space

Supports a wide range of workloads and allows users extreme levels of experimentation and productivity.

Not All Storage is Equal

Some systems perform better for specific tasks (more on this later).

Scalable and Resistant

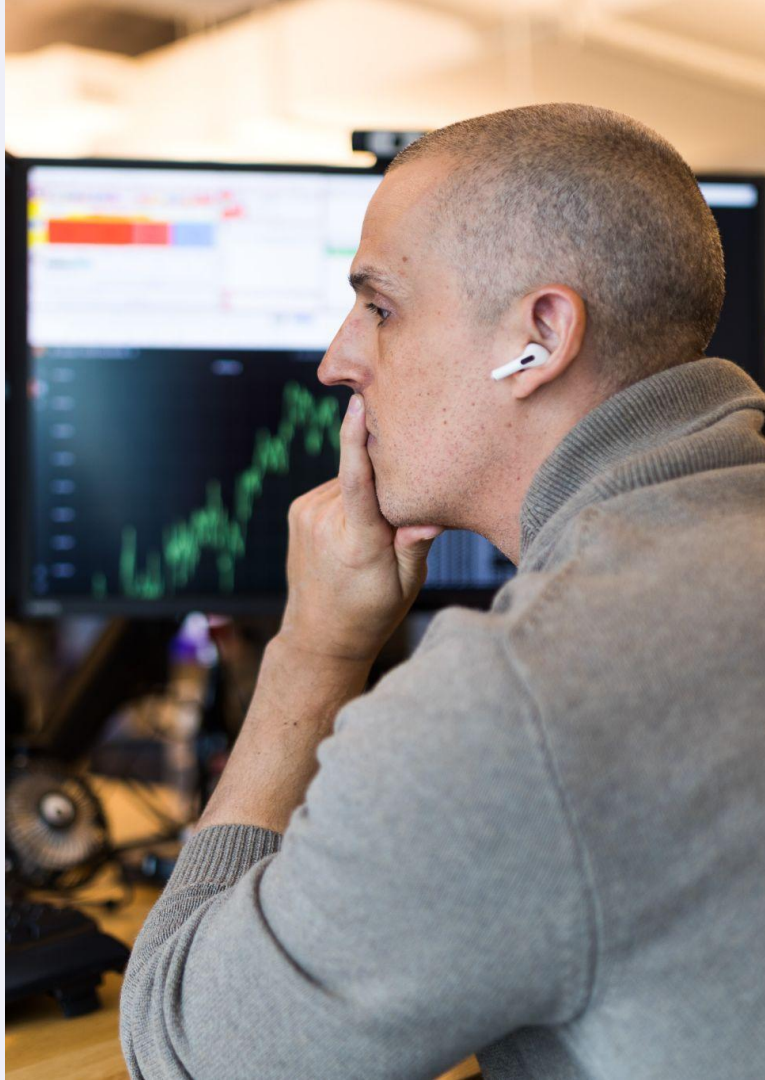
Lustre (especially with DNE3) excels at handling large, complex workloads

Performance Matters

We measure inefficiencies in throughput, IOPS, and metadata, and most importantly, overall job runtime and CPU vs I/O efficiency.

Lustre as a Tactical Asset

- Debugging Power
- Isolation for Analysis
- Deep Visibility
- Robust Troubleshooting
- Leveraging Community Insights



What We've Learned

Heavy DNE3 User

Running at scale since late 2021.

Real-World Challenges

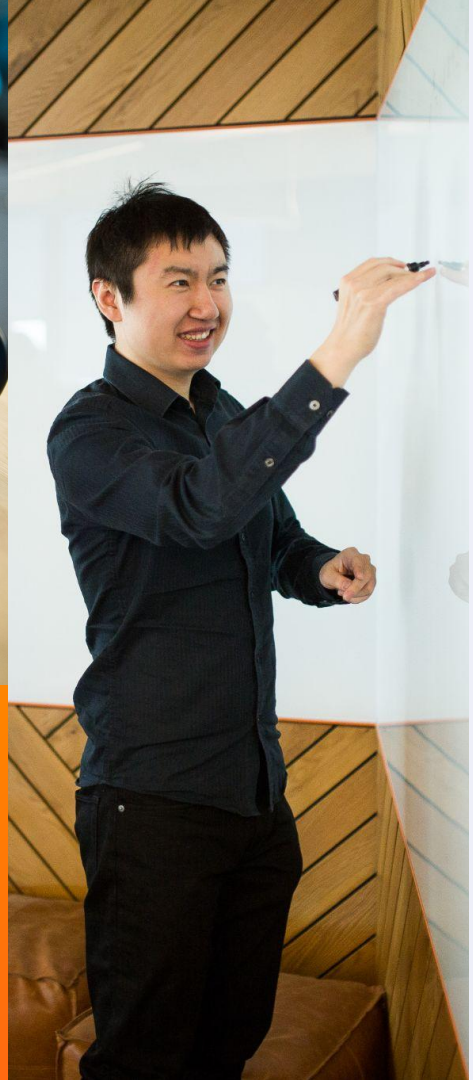
Issues emerged only under sustained, large-scale use.

Lessons Learned

Early growing pains, but strong long-term performance.

Whamcloud Partnership

Exceptional support throughout our journey.



HRT

DNE3 Small File Mythbusting

Where did this idea come from?

- Lustre was late to **active/active metadata servers**, but discussions predate v1.0.
- **HPC "Common Knowledge"** – Small files on Lustre were a no-go.
- **MDT0 Overload** – Accidental metadata-heavy workloads caused system-wide issues.

And Were the Haters Right?

Many institutions worked around this by **optimizing code** to reduce metadata load.

DNE1 Arrives (Lustre 2.4/2.5 LTS) – Introduced directory "pinning" to specific MDTs.

Limited Adoption – Most sites stuck to single active/passive MDS setups.

Laid the Groundwork for DNE3 – Over a decade in the making.

DNE3: Solving Small File Scalability Challenges

The Evolution of DNE

DNE2 Introduced – Intended to address **highly concurrent directories**, but adoption was limited.

Manual Effort Required – Admins had to **handcraft** DNE1 directory trees to balance load.

Performance Potential – Helped in specific cases (e.g., **io500 mdtest-hard**) with DNE2, manual “load balancing” with DNE1.

The Problem Before DNE3

Artificial Hotspots – Even well-balanced DNE1 setups led to per-user metadata bottlenecks.

Scalability Issues – Without DNE3, Lustre required **active manual intervention** for efficiency.

Common Pitfall – Over time, MDT imbalance emerged (e.g., **MDT0 fills up while others sit empty**).

So... *Were* the Haters Right?

Yes (Historically)

Small file performance wasn't a priority in early Lustre use cases.

Limited real-world testing and public benchmarking reinforced the perception.

Without careful **manual DNE1 planning**, metadata performance could suffer.

No (Today)

PFL + DoM (since Lustre 2.10/2.13) significantly improved small file handling.

DNE3 Automates Scaling – No more manual metadata allocation/performance headaches.

Major Metadata Improvements – Ongoing optimizations over the last 6–7 years.

io500 Benchmarking – A key driver in detecting and improving performance issues.

DNE3 small file mythbusting

Examples

Myth: Lustre Can’t Handle Many Files

✅ Reality: Lustre scales. We’ve managed 25B+ files/dirs on a single filesystem.

THE NUMBERS

80/20 Rule – A few users drive most inode usage.

<3% Inode Imbalance – Evenly distributed metadata across MDTs.

Expected Scale – **5–10B files per filesystem** (minimum), always planning for much more.

Minimal Admin Overhead – No major performance issues in most cases.

Myth: Lustre Performs Poorly for Small Files

✅ Reality: Lustre handles massive, dynamic small-file workloads efficiently.

REAL-WORLD EXAMPLE: 25B+ FILES/DIRS IN ACTION

High-Intensity Workload – Always running, always pushing storage limits.

Crashes Non-Lustre Systems – Moved after multiple failures elsewhere.

Massive Directory Trees – 100K+ items per directory, constant growth.

Complex File Operations – Frequent **small file creation**, symlink resolution, and multi-user queries.

Evenly Distributed Load – No common **MDT bottlenecks** or performance degradation.

Myth: Lustre Performs Poorly for Small Files

✅ Reality: Lustre handled a misconfigured, high-intensity small-file workload that broke other systems.

REAL-WORLD EXAMPLE: MISCONFIGURED APP CHAOS

Extreme Open/Close Cycles – App constantly **stat'ed, truncated, wrote, and closed files**.

Other POSIX Systems Failed – The workload overwhelmed every other storage systems.

Lustre Took the Hit – Moved the workload to Lustre, which handled it successfully.

Next Slide: Live Metadata Counters – Data pulled from **Lustre server-side counters via REST API**.

Open Call Bug – Open() calls weren't logged, so **add ~1M additional opens.**

NAME	TIOPS	CLOSE	CROSSDIR RENAME	GETATTR	GETXATTR	LINK	MKDIR	MKNOD	OPEN	RENAME	RMDIR	SAMEDIR RENAME	SETATTR	SETXATTR	STATFS	SYNC	UNLINK
-MDT0000	76660.74	25487.13	0.00	25627.85	12.70	0.00	7.82	18.57	26.38	8.79	0.00	8.79	25441.20	0.00	12.70	8.79	0.00
-MDT0001	71287.62	23733.94	0.00	23805.12	10.73	0.00	5.85	10.73	12.68	1.95	0.00	1.95	23695.91	0.00	6.83	1.95	0.00
-MDT0002	74258.40	24673.19	0.00	24731.68	22.42	0.00	10.72	30.22	48.74	17.55	0.00	17.55	24672.21	0.00	15.60	18.52	0.00
-MDT0003	74478.02	24777.81	0.00	24829.59	19.54	0.00	6.84	34.19	42.01	6.84	0.00	6.84	24737.76	0.00	9.77	6.84	0.00
-MDT0004	77535.91	25781.35	0.00	25847.58	20.45	0.00	8.77	29.22	37.98	17.53	0.00	17.53	25747.26	0.00	10.71	8.77	8.77
-MDT0005	74684.64	24836.00	0.00	24896.51	18.54	0.00	6.83	30.25	41.96	20.49	0.00	20.49	24784.28	0.00	8.78	11.71	8.78
-MDT0006	73455.46	24362.78	0.00	24595.16	18.55	0.00	10.74	25.39	42.96	17.58	0.00	17.58	24329.58	0.00	17.58	17.58	0.00
-MDT0007	76352.94	25400.85	0.00	25466.28	10.74	0.00	8.79	16.60	27.35	9.77	0.00	9.77	25378.38	0.00	14.65	9.77	0.00
-MDT0008	77894.27	25911.98	0.00	25927.62	20.53	0.00	8.80	25.41	41.05	16.62	0.00	16.62	25894.38	0.00	14.66	16.62	0.00
-MDT0009	75878.19	25219.83	0.00	25286.22	20.50	0.00	13.67	33.20	48.82	13.67	0.00	13.67	25196.40	0.00	17.57	14.65	0.00
-MDT000a	77902.10	25903.95	0.00	25965.42	19.51	0.00	9.76	29.27	46.83	15.61	0.00	15.61	25867.85	0.00	12.68	15.61	0.00
-MDT000b	74988.19	24952.43	0.00	25001.28	10.75	0.00	6.84	14.65	23.45	7.82	0.00	7.82	24941.68	0.00	12.70	8.79	0.00
-MDT000c	78444.77	26104.21	0.00	26160.00	9.79	0.00	2.94	15.66	19.57	11.74	0.00	11.74	26088.55	0.00	6.85	4.89	8.81
-MDT000d	77269.98	25704.62	0.00	25753.41	17.56	0.00	6.83	24.39	35.13	10.73	0.00	10.73	25682.18	0.00	12.68	11.71	0.00
-MDT000e	79788.26	26552.10	0.00	26624.43	11.73	0.00	5.87	11.73	18.57	5.87	0.00	5.87	26535.48	0.00	10.75	5.87	0.00
-MDT000f	76058.24	25283.08	0.00	25338.75	20.51	0.00	6.84	31.25	45.91	12.70	0.00	12.70	25279.17	0.00	12.70	14.65	0.00
-MDT0010	74459.96	24750.88	0.00	24833.03	17.60	0.00	9.78	27.38	39.12	10.76	0.00	10.76	24735.23	0.00	14.67	9.78	0.98
-MDT0011	74530.97	24767.70	0.00	24841.05	20.54	0.00	9.78	26.40	42.05	14.67	0.00	14.67	24760.86	0.00	16.63	16.63	0.00
-MDT0012	76629.26	25484.18	0.00	25529.09	23.43	0.00	8.79	27.34	44.91	17.57	0.00	17.57	25446.11	0.00	12.69	17.57	0.00
-MDT0013	73894.21	24563.80	0.00	24624.25	19.50	0.00	9.75	26.32	43.87	17.55	0.00	17.55	24542.35	0.00	11.70	17.55	0.00
-MDT0014	74607.68	24806.68	0.00	24880.95	11.73	0.00	5.86	21.50	28.34	14.66	0.00	14.66	24791.04	0.00	17.59	5.86	8.80
-MDT0015	75901.04	25228.20	0.00	25318.87	14.62	0.00	11.70	22.42	38.02	11.70	0.00	11.70	25211.63	0.00	19.50	12.67	0.00
-MDT0016	75945.25	25139.76	0.00	25531.06	16.59	0.00	9.76	24.40	38.06	10.73	0.00	10.73	25137.81	0.00	13.66	12.69	0.00
-MDT0017	77894.67	25893.62	0.00	25966.84	22.46	0.00	4.88	24.41	35.15	14.65	0.00	14.65	25895.57	0.00	9.76	12.69	0.00
-MDT0018	39921.45	13245.64	0.00	13296.41	20.50	0.00	7.81	27.34	39.06	17.57	0.00	17.57	13230.02	0.00	8.79	10.74	0.00
-MDT0019	40492.96	13432.23	0.00	13508.40	19.53	0.00	9.76	28.32	41.01	14.65	0.00	14.65	13399.03	0.00	13.67	11.72	0.00
-MDT001a	40765.31	13545.82	0.00	13615.12	15.62	0.00	10.74	17.57	27.33	9.76	0.00	9.76	13486.28	0.00	14.64	12.69	0.00
-MDT001b	39661.72	13156.52	0.00	13195.54	19.51	0.00	10.73	29.26	47.79	16.58	0.00	16.58	13132.14	0.00	20.48	16.58	0.00
-MDT001c	76312.82	25386.96	0.00	25432.74	18.50	0.00	5.84	19.48	30.19	9.74	0.00	9.74	25374.30	0.00	11.69	13.63	0.00
-MDT001d	75874.31	25232.09	0.00	25307.00	16.54	0.00	7.78	21.40	29.18	7.78	0.00	7.78	25216.53	0.00	20.43	7.78	0.00
-MDT001e	76526.94	25412.19	0.00	25469.68	27.28	0.00	15.59	34.10	56.51	21.44	0.00	21.44	25428.76	0.00	18.51	21.44	0.00
-MDT001f	77333.11	25700.41	0.00	25770.85	18.59	0.00	9.78	23.48	37.18	16.63	0.00	16.63	25711.17	0.00	14.68	13.70	0.00
-MDT0020	82619.51	27476.95	0.00	27534.33	17.50	0.00	6.81	31.12	42.79	21.39	0.00	21.39	27438.05	0.00	10.70	9.72	8.75
-MDT0021	77058.83	25638.99	0.00	25673.23	15.66	0.00	6.85	26.42	37.18	9.78	0.00	9.78	25620.39	0.00	10.76	9.78	0.00
-MDT0022	81501.46	26835.69	0.00	28094.53	16.59	0.00	5.86	22.44	32.20	10.73	0.00	10.73	26456.09	0.00	6.83	9.76	0.00
-MDT0023	77180.36	25684.18	0.00	25719.30	11.71	0.00	6.83	16.59	27.32	9.76	0.00	9.76	25678.32	0.00	6.83	9.76	0.00
-MDT0024	80967.17	26901.23	0.00	26990.03	23.42	0.00	9.76	35.13	50.75	13.66	0.00	13.66	26895.37	0.00	18.54	15.61	0.00
-MDT0025	68731.71	22851.95	0.00	22900.80	21.49	0.00	7.82	28.33	40.06	12.70	0.00	12.70	22827.52	0.00	16.61	11.72	0.00
-MDT0026	77461.99	25777.04	0.00	25815.13	11.72	0.00	6.84	12.70	21.49	9.77	0.00	9.77	25777.04	0.00	10.74	9.77	0.00
-MDT0027	79786.91	26521.17	0.00	26598.24	17.56	0.00	7.80	24.39	37.07	12.68	0.00	12.68	26526.05	0.00	13.66	15.61	0.00
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
total	2912997.34	968119.11	0.00	972303.35	702.75	0.00	334.78	978.99	1466.03	522.17	0.00	522.17	966989.94	0.00	530.98	482.17	44.88

Myth: Lustre Performs Poorly for Small Files

✅ Reality: Lustre efficiently handles deeply nested, high-concurrency metadata workloads.

REAL-WORLD EXAMPLE: GUID-STYLE DIRECTORY TREES

User Workaround – Instead of single directories with many files, users built **B-tree style hashed paths** (2-3+ levels deep).

Parallel Execution – Massive **multi-threaded compute grid** writes and reads files simultaneously.

Heavy LDLM – Each thread **stat()s and creates (overlapping) parent directories**, adding lock contention.

Some Systems Struggled – Performance bottlenecks on other storage systems led to **repeatable benchmarking**.

Measurable Performance Data – Created explicit **timing benchmarks** to track scaling behavior.

Myth: Lustre Performs Poorly for Small Files

✅ Reality: Lustre ranks #2 in a repeatable benchmark, proving DNE3's effectiveness.

BENCHMARK INSIGHTS

A Close Second – Lustre **closely trails the top-performing NFS-based system**.

Similar Architectures – The #1 system **shares DNE3's directory ownership/sharding model**, validating its approach.

Consistent Results – Thousands of tests show **stable rankings over years, across versions, kernels, and system sizes**.

The Competition? Not Even Close – Other storage systems are **orders of magnitude slower** under high concurrency.

Impact Beyond Benchmarks – Many non-Lustre systems see **latency spikes and degraded performance for unrelated workloads** while this GUID-tree populator is running.

Myth: Lustre Performs Poorly for Small Files

✅ Reality: Lustre Consistently Scores High on io500 Benchmarks.

BENCHMARK INSIGHTS

Lustre Excels in io500 – Check the **production** and **10-node production** rankings.

Transparent & Reproducible – Click **“Reproducible”** on any Lustre/EXAScaler system to see exact configurations.

Example: io500 Entry #723 – **DNE3 was already enabled, no exotic tuning needed.**

Proof in the Numbers – High performance isn’t just theoretical—**it’s been tested at scale.**

HRT’s Next Steps? – Greenlight to submit io500 for one of our systems, just haven’t done it yet. Some other “big claim” vendors have zero entries, or we’d need an electronic microscope to read the carbon data.

Lustre, Like Everything Else, Has Varied Small File Performance

Performance is Context-Dependent

Radar Chart Perspective – Across our internal metadata workloads and io500, Lustre would be a close second overall.

The #1 Filer? – Excels in metadata operations, but limited in scaling and throughput.

Performance Variability – Lustre can be:

- Best-in-class for certain workloads
- Average for others
- Up to 20% slower in some cases

Addressing the Critics

"I Found Something That Runs Poorly on Lustre!" – Believable. But what's the context?

Legacy Code? – Running 20-year-old Fortran might not be optimized.

No Storage System is Magic – Every system has trade-offs.

The Real Takeaway

Lustre is competitive and always improving.

Don't dismiss Lustre for small files—evaluate it based on real-world needs.

Lustre Metadata Design Considerations

Do

Over-Provision Metadata Storage

- **Prioritize metadata capacity/performance over data capacity/performance**
- Extra 50T of metadata > Extra 50T of data storage if budget constrained
- Extra MDS pairs/sets > Extra OSS pairs/sets (in most cases).

Optimize DNE3 Round-Robin Depth

- Default **DNE3 depth = 3**, then switches to space balancing.
- We use **17 at the root** of project/user trees, delaying space balancing.
- Space balancing is largely handled by **randomization/round robin**—deeper allocations improve performance!

Don’t

Mixing OSS & MDS on the Same Server

- **Heavy OST throughput** blocks metadata IOPs due to **CPU/NIC contention**.
- No amount of **QoS, thread prioritization, or clever tuning** will fully solve this.
- **Rule of thumb:** Keep **data (OSS) and metadata (MDS) separate**.

Oversizing DoM (Data-on-Metadata)

- **Max DoM size = 1GB**—setting it too high turns your **MDS into a throughput bottleneck**.
- Instead, use **PFL layouts**:
- **DoM (optional) → Flash OST → HDD OST** (See **ORNL LUG 2022/2023** for examples).

Lustre Wishlist

Lustre general wishlist

Faster Failover & Recovery

Reduce 3-5m hangs → < 1m to support more workloads.

Improved Metadata & Data Management

Automate balancing; manual expansion is impractical.

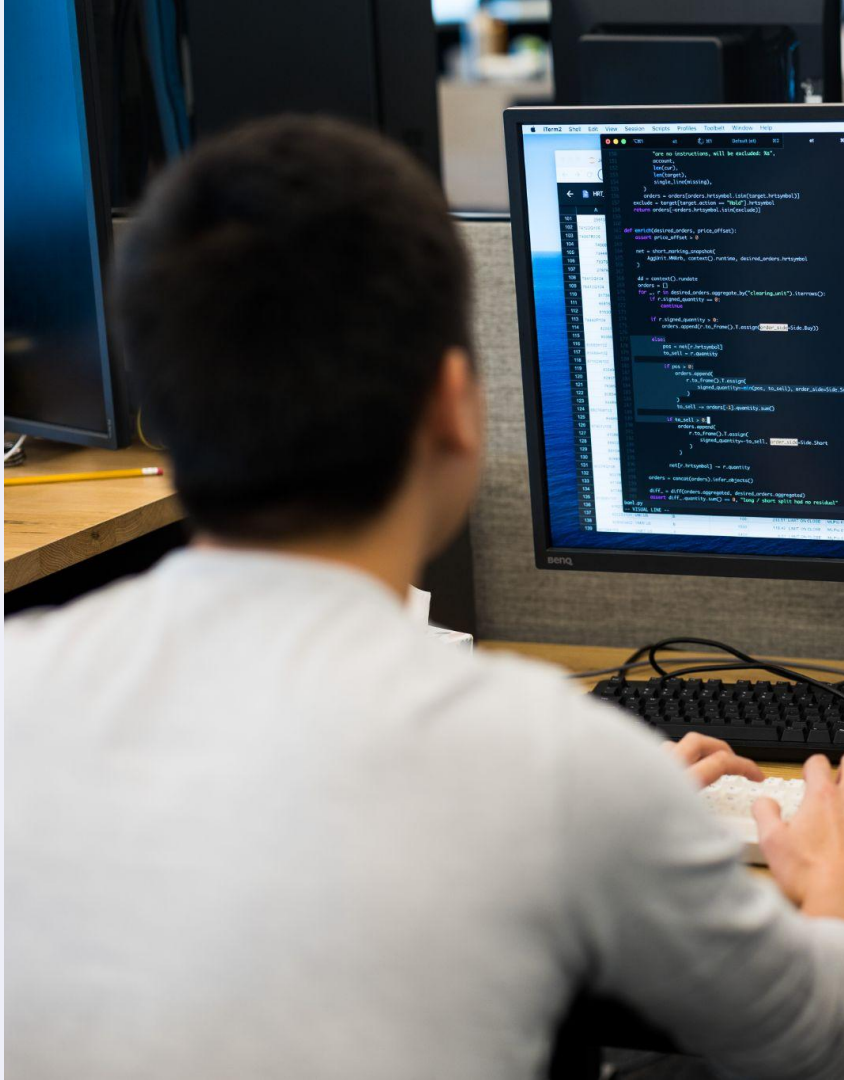
Flexible Storage with FLR Erasure Coding

Move beyond RAID & ZFS dependencies; let Lustre handle redundancy.

Native High Availability (HA) Stack

Replace corosync/pacemaker with Lustre-native failover logic.

Summary



In closing...

Fix Your Code

"A supercomputer is a device for turning compute-bound problems into I/O-bound problems."

- Ken Batcher

Open Source Wins Again

Lustre's Legacy: Dominating storage for TOP10/100/500 systems for decades.

Long-Term Success: Strong initial design, continuous development, and community contributions.

Beyond HPC: Now excelling in modern, complex non-HPC workloads.

Competition Comes and Goes

Common critiques:

- "Outdated architecture"
- "Not built for flash"
- "Too brittle"

Lustre Endures – 25+ years of real-world testing, feedback, and expertise.

Most competitors haven't been battle-tested at scale.

And finally…

Lustre is quite good at small files / metadata performance now, it’s time to stop saying otherwise.

Questions?

I charge 1 beer per question, which will be collected at happy hour.

We’re Hiring!

Whatever your experience lies in, we’re always looking for talented engineers.

www.hudsonrivertrading.com/careers/

Thanks!

Special thanks (in no particular order):

- <3 HRT
- <3 Whamcloud/DDN
- <3 Lustre Community
- <3 Our Users