Exploring Resource Migration Using the CephFS Metadata Cluster

Michael Sevilla*†, Scott Brandt*, Carlos Maltzahn*, Ike Nassi*, Sam Fineberg†

*UC Santa Cruz †HP Storage

February 18, 2014

Migrating Resources Improves Performance

Migrating resources is an important part of load balancing
- Depends on (1) system parameters and (2) migration overheads

Metadata management → explore new migration heuristics
- Popularity, not size, drives metadata distribution

CephFS → prototyping platform for heuristics
- Built for locality; migration tools are implemented

Metadata is different than data

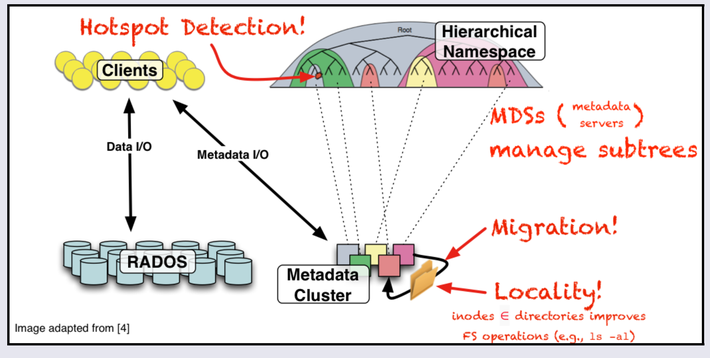
Poor throughput scalability:
- ✗ highly accessed; synchronous; small writes [3, 4]
- ✗ proven techniques are insufficient [1, 2]

This problem, once reserved for HPC, is now in large data centers.

Why Use CephFS?

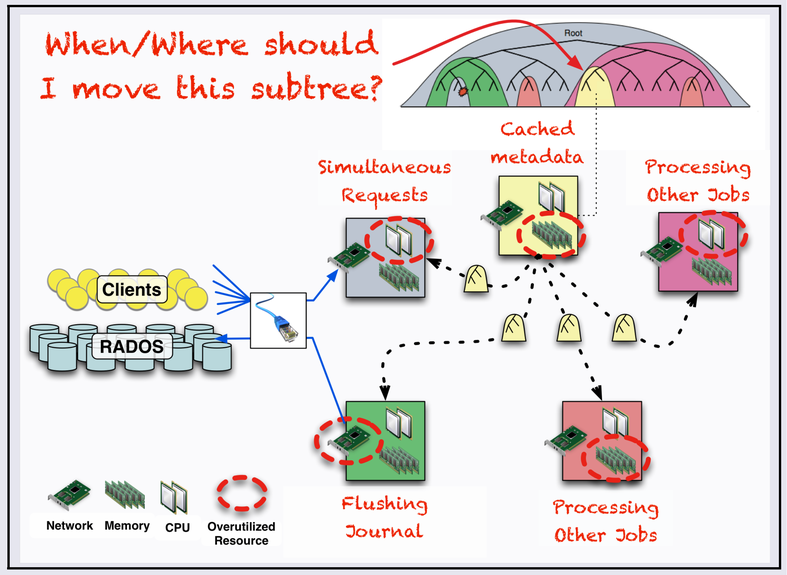
Ceph provides distributed storage
- data striped across a reliable object store (RADOS)
- data located with hash-based algorithm

CephFS: POSIX file system that uses RADOS
- dynamic subtree partitioning [5]



- fragments write-intensive directories; replicates read-intensive content

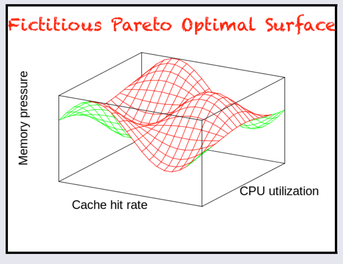
MDS Cluster State Should Affect Migration

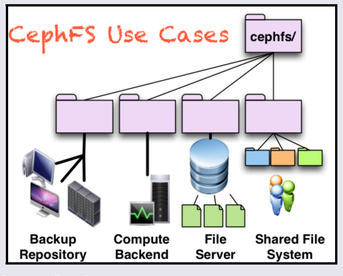


Eviction/Balancing must account for each MDS's resource utilization

Smarter Eviction & Balancing Heuristics

Construct Pareto Optimal Surfaces
- performance counters & timing analysis





Resolve trade-offs with migration heuristics

Identify viable heuristics with machine learning
- auto-correlation → periodicity; decision trees → predict performance

Implement heuristics as a distributed service

Balancing Resources ↛ Performance

Preliminary Results (5 Servers each with 8GB RAM, 4 cores)
- Workload: 50,000 file create requests (`mdtest`)
- Migration depends on CPU utilization & request rate

CephFS Balances Load

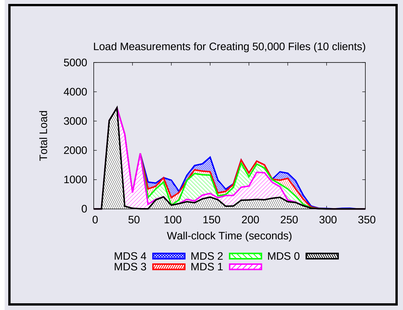


Figure: Popular directories are fragmented across multiple MDS servers.

Load Distr. ↛ Performance

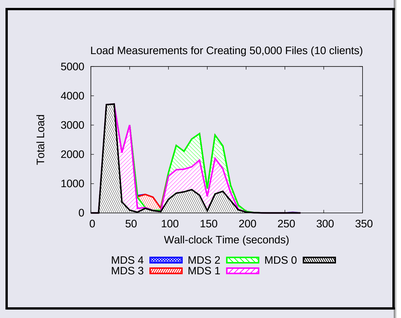


Figure: Changing the MDS order (MDS0↔MDS4) increases per-MDS load.

Hotspots can improve caching, if the CPUs can handle them

Conclusion

Load-balanced system ↛ optimal system behavior

Identify which parameters to optimize

Optimizing for latency, throughput, resource utilization, wear-leveling, power, balanced heat dissipation, network traffic, client load...

... will produce different workload distributions!

References

[1] S. R. Alam, H. N. El-Harake, K. Howard, N. Stringfellow, and F. Verzelloni.
Parallel I/O and the Metadata Wall.
In *Proceedings of the 6th Workshop on Parallel Data Storage*, PDSW'11, 2011.

[2] K. McKusick and S. Quinlan.
GFS: Evolution on Fast-forward.
Commununications ACM, 53(3):42–49, Mar. 2010.

[3] D. Roselli, J. R. Lorch, and T. E. Anderson.
A Comparison of File System Workloads.
In *Proceedings of the Annual Conference on USENIX Annual Technical Conference*, ATEC '00, pages 4–4, 2000.

[4] S. A. Weil, S. A. Brandt, E. L. Miller, D. D. E. Long, and C. Maltzahn.
Ceph: A Scalable, High-Performance Distributed File System.
In *Proceedings of the 7th USENIX Symposium on Operating Systems Design & Implementation*, OSDI'06, 2006.

[5] S. A. Weil, K. T. Pollack, S. A. Brandt, and E. L. Miller.
Dynamic Metadata Management for Petabyte-Scale File Systems.
In *Proceedings of the 17th ACM/IEEE Conference on Supercomputing*, SC'04, 2004.