Benchmarking Effects of Erasure Scheme and MPI Configuration on MarFS Throughput

Janya Budaraju (Johns Hopkins University), Paul Karhnak (University of Virginia), Zachary Snyder (Montana State University)

Mentors: Garrett Ransom, David Bonnie

Background

MarFS Overview

- Open-source LANL campaign storage implementation used in production clusters.
- Provides higher resiliency and capacity than scratch, higher throughput than tape.
- Uses "multi-layer erasure coding" at multiple points in the data transfer process to limit throughput slowdowns.

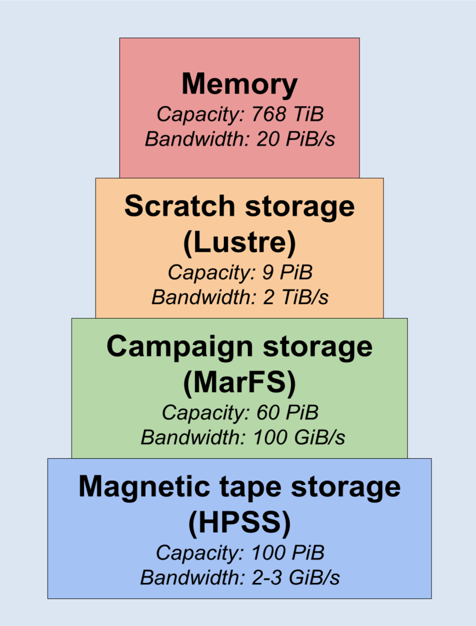

Figure 1. LANL storage stack.

Erasure Coding Overview

- Data is split into *N* data blocks, which are then encoded into *E* parity blocks using a Reed-Solomon error correction algorithm from the Intel ISA-L library.
- Each block consists of the same number of bytes (partsize, or *PSZ*).
- Parity blocks are used to reconstruct data blocks upon data loss; one "stripe" can lose up to *E* data blocks.

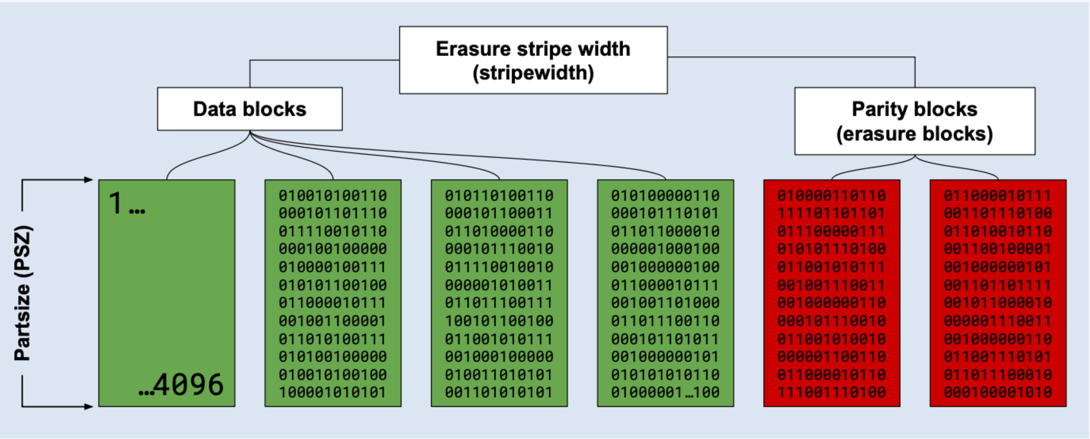

Figure 2. A 4+2 erasure coding scheme with partsize = 4096 bytes

Ranks per Node Overview

- MarFS utilizes a parallel file copy operation (*pfcp*, LANL utility under pftools) to write to/read from a MarFS mount.
- To parallelize, OpenMPI distributes work across compute nodes, with a minimum number of MPI processes (units of work) per node specified in the *pftool* configuration.

Project Overview

Motivations

- Storage performance and filesystem throughput at a cluster level can be highly dependent on hardware factors, such as cache layout or processor architecture.
- Finding optimal MarFS erasure scheme and MPI configurations currently requires extensive expertise & assumptions about hardware performance.
- Erasure scheme configuration is currently dependent upon assumptions about the ISA-L erasure coding library and how its performance scales across large workloads.

Deliverables

- Create a software suite of benchmarking tools that can be run on any cluster with a functional MarFS mount to evaluate, visualize, and optimize *N, E, PSZ,* and minimum ranks per node while abstracting away hardware specifics.
- Identify MarFS performance patterns and better understand ISA-L erasure coding in HPC workloads.

Partsize Benchmarking

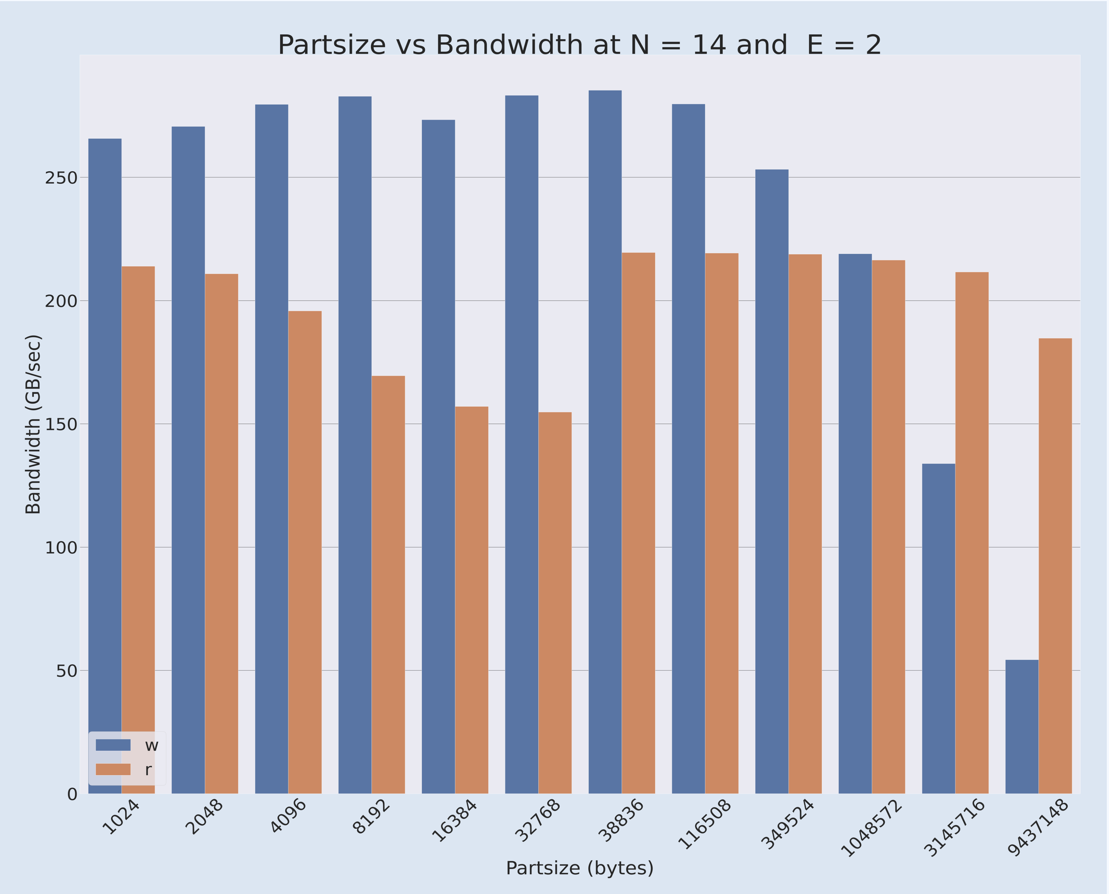

Figure 3. Benchmarking throughput at fixed data blocks and parity block values to assess effects of partsize on performance.

- Read and write both have peak throughput at *PSZ*=38836 bytes → *PSZ* was fixed at 38836 bytes for tests across other parameters.
- Write throughput drops off at high *PSZs*; we believe reduction is due to growing matrix multiplication operations to perform erasure coding.
- We believe lower read performance for *PSZ* < 38836 is due to MarFS buffer handling. Aligned to MarFS buffers at *PSZ* ≥ 38836, performance increased and stabilized.

Rank Benchmarking

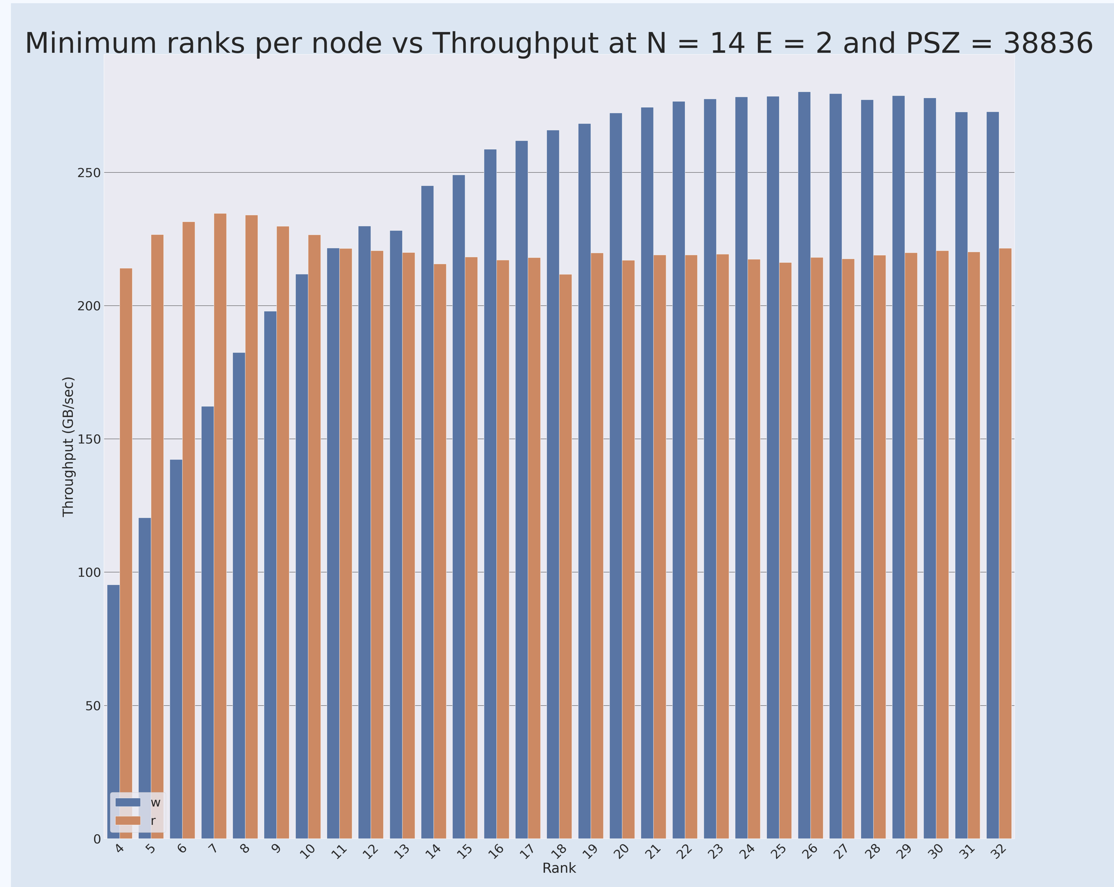

Figure 4. Benchmarking throughput at a fixed erasure scheme to assess effects of ranks per node on performance.

- Write bandwidth traces a logarithmic shape, with a sharp initial increase before stabilizing as ranks reach 20. Likely from increasing parallelism without using up memory bandwidth or causing cache thrashing.
- Read reaches peak throughput at 7 ranks per node, then levels off and stabilizes. Potentially due to a peak in read-specific cache coherency at ~7 ranks per node.
- From these results, we fixed ranks per node to 23 for heatmap generation.

N + E Benchmarking

Reads

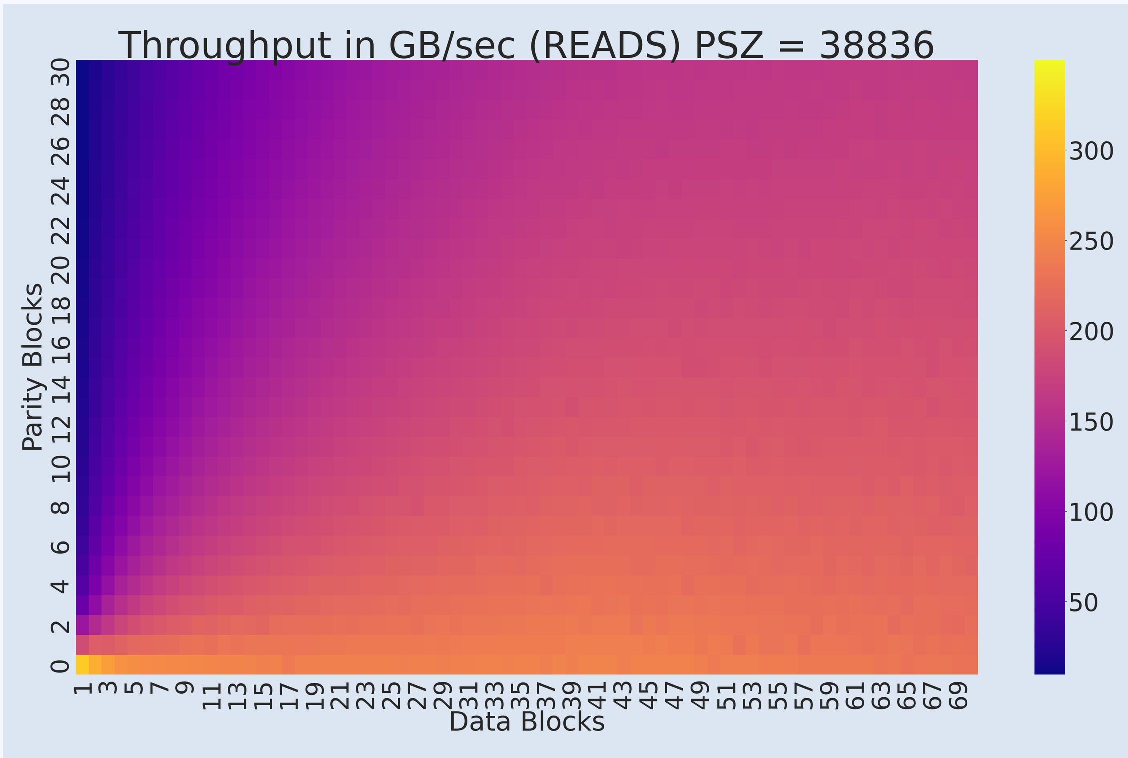

Figure 5. Benchmarking read throughput at fixed *PSZ* and ranks per node to assess performance at varying *N* and *E* values.

- As expected, throughput across read operations for varying *N+E* values creates a clean gradient.
- As no erasure coding is performed on read, there is no computational scaling that occurs at higher *E* values, which is consistent with results.
- Read operation is slowest at low *N* and high *E* values as throughput recorded is "logical" throughput, which only records data block movements; high *E* values result in more data being moved, consuming time but not contributing to logical bandwidth, slowing performance.

Writes

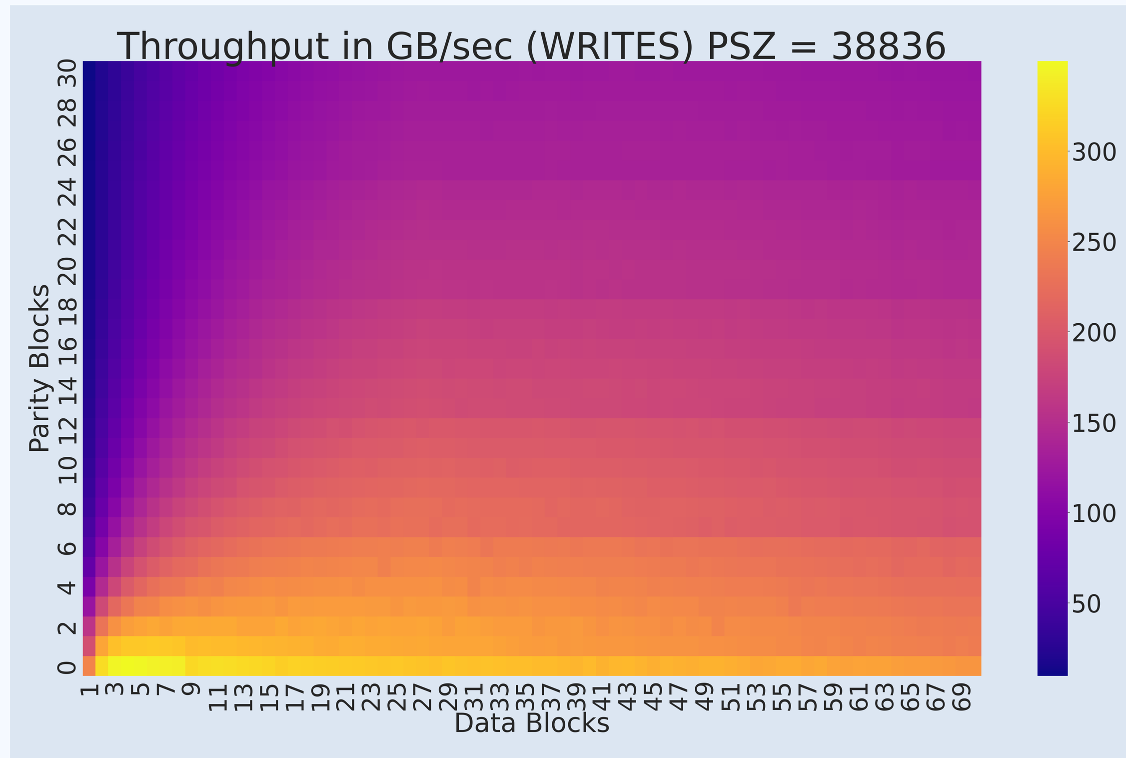

Figure 6. Benchmarking write throughput at fixed *PSZ* and ranks per node to assess performance at varying *N* and *E* values.

- The write heatmap reveals clear horizontal bands or chunks stratifying throughput.
- We believe the ISA-L library's Reed-Solomon implementation creates banded write patterns; resource usage exceeding specific *E* values may push the implementation's data structures a discrete step across cache layers and memory.
- Write operations slow significantly at higher *E* values, reflecting increasing computational demands of erasure coding.

Methods

- Build MarFS dependencies (including pftools and ISA-L) and create shared FUSE mount across nodes.
- Utilize a "no-op DAL", or *no-operation data abstraction layer*, to "fake out" rest of MarFS stack and perform all operations on truncated files.
- Automate testing by running pfcp operations with varied *N, E, PSZ,* and minimum rank per node configurations, then parsing for throughput data, via Python and Bash scripts
- Visualize data through graphing script which takes CSV-formatted benchmarking data.
- Run benchmarks on 10 compute + 1 head node team-built cluster (GIGABYTE R262-Z32-00 nodes, AMD EPYC 7502 32- core CPUs, 128 GB DDR4 RAM running Rocky Linux 8.8).

Challenges

CRC Optimization

- Initially, the no-op DAL performed significantly below expectations.
- Analysis with callgrind revealed that CRC generation limited performance; using an optimized ISA-L library function instead boosted throughput by up to 5x.

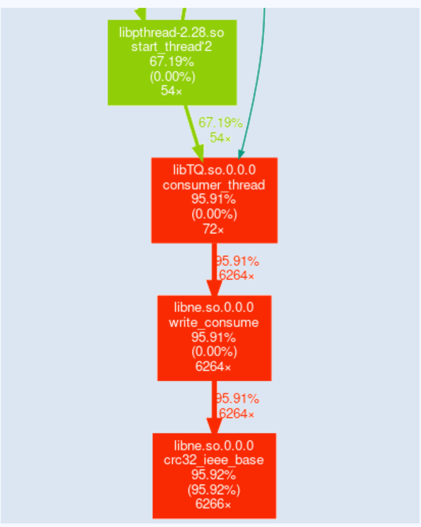

Figure 7. Portion of callgrind graph pre-CRC patch.

Read Performance

- Reads were not as performant as expected—as erasure coding occurs only on writes, reads were expected to be much faster, but were only slightly quicker at high *E* counts.
- Troubleshooting ruled out CPU memory controllers, file size effects, and PSZ effects; can likely be attributed to unexpected read overhead in MarFS implementation.

Discussion

- Software suite effectively benchmarked MarFS based on erasure scheme and MPI parameters independently of hardware details, providing objective performance metrics and clarifying throughput relationship to parameters.
- Observed performance offered insight into ISA-L behavior, particularly smooth read gradient, lower throughput at higher *E* values, and banded write pattern.
- Selecting optimal erasure scheme requires balancing throughput with resiliency to meet user needs; however, these tools provide a key starting point to accommodating growing storage system performance needs at LANL.

Future Work

- Investigating possible MarFS stack overhead underlying read inefficiency.
- Comparing performance after migrating MarFS to NFS 4.2.
- Benchmarking a future hybrid MarFS and magnetic tape storage system ("Marchive" storage).

Los Alamos NATIONAL LABORATORY

NNSA National Nuclear Security Administration

Managed by Triad National Security, LLC, for the U.S. Department of Energy's NNSA.

LA-UR-23-28684

Johns Hopkins University · University of Virginia · Montana State University