E2000 Performance From Microbenchmarks to Applications

Bill Loewe*
william.loewe@hpe.com
Hewlett Packard Enterprise
Spring, TX, USA

Michael Moore*
michael.moore@hpe.com
Hewlett Packard Enterprise
Spring, TX, USA

Sakib Samar*
sakib.samar@hpe.com
Hewlett Packard Enterprise
Spring, TX, USA

Chris Walker*
christopher.walker@hpe.com
Hewlett Packard Enterprise
Spring, TX, USA

Abstract

With the advance of the Exascale Age and its continued gains in FLOPS performance, the associated I/O demands increase commensurately. To address this, the HPE Cray Supercomputing Storage Systems E2000 is the next generation of the HPE Cray Supercomputing Storage product line with a focus on performance. This paper discusses the architecture changes in the E2000 and provides node and file system microbenchmarks measuring bandwidth, IOPS, and metadata performance. The improved PCIe and NVMe drive speeds in addition to the higher density enclosure in the E2000-F allow for more than twice the throughput and IOPS performance compared to the previous generation with nearly all of the performance achievable by optimal application workloads. System configuration choices, such as number of storage targets and BIOS settings, which influence system level performance will be compared with an aim to optimize the gains and determine ideal client/server tunings. Finally, performance of application-relevant workloads including random access, shared file, and AI/ML storage workloads will be presented along with discussion of application and job changes to utilize the E2000 performance improvements.

CCS Concepts

• **Information systems** → **Distributed storage**; *RAID*; • **Hardware** → *Testing with distributed and parallel systems.*

Keywords

Lustre, Performance,

ACM Reference Format:
Bill Loewe, Michael Moore, Sakib Samar, and Chris Walker. 2025. E2000 Performance From Microbenchmarks to Applications. In *CUG 2025: Computing Horizons, May 05–08, 2025, Jersey City, NJ*. ACM, New York, NY, USA, 13 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

*All authors contributed equally to this research.

CUG '25, Newport, NJ

ACM ISBN 978-x-xxxx-xxxx-x/YYYY/MM
https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 Introduction

The new HPE Cray Supercomputing Storage System E2000 is the next generation of the HPE Cray Supercomputing Storage Systems E1000. In addition to iterative advances in CPU, memory, NVMe drive, and PCIe speeds, the E2000 features storage enclosure design changes that allow 32, instead of 24, NVMe drives per enclosure. The hardware architecture characteristics are described to inform subsequent configuration and microbenchmark discussions. The performance characteristics of the storage server hardware are compared between HPE Cray Supercomputing Storage Systems E1000 and E2000 with local storage microbenchmarks as the basis for understanding the application performance improvements provided by the E2000. Storage node microbenchmarks show the all-flash E2000-F is able to achieve drive-limited write performance and nearly 80% of available read performance representing a nearly 300% increase in locally available disk bandwidth compared to the E1000-F.

Several I/O microbenchmarks are used to evaluate the storage characteristics of the HPE Cray Supercomputing Storage Systems E2000 as compared to the E1000. The results are discussed in relation to the previously described architectural changes and compared with optimal scaling of per-drive performance to RAID device and object storage target (OST) performance. Lustre network selftest (LST) is discussed and client-driven benchmarks including IOR and mdtest are used to demonstrate full-stack performance including sequential, random, and metadata workloads as models for user applications. The importance of evaluating BIOS settings such as NUMA nodes per socket (NPS) and Lustre configuration parameters such as the libcfs CPU Partition Table (CPT) during platform bring-up are discussed and their impact shown using client microbenchmarks. Optimal results for IOR benchmarks using GridRAID and ldiskfs on the E2000-F show a greater than 150% sequential write performance improvement and greater than 125% sequential read performance improvement over the E1000-F using GridRAID. The throughput performance is NVMe-drive limited for writes and network limited for reads. IOPS for the same configuration show a greater than 325% increase in 4k write IOPS and a greater than 100% increase in 4k read IOPS over the E1000-F for an ldiskfs on GridRAID (distributed parity) configuration.

The MLPerf Storage Benchmark suite will augment traditional microbenchmarks to evaluate the E1000 and E2000 across a set of representative AI/ML storage workloads including 3D U-Net, ResNet-50, and CosmoFlow.

Finally, a traditional HPC application, WRF, is used to to evaluate application-level I/O time for a shared file collective MPI-IO write workload. Further, application I/O resource allocation and Lustre striping choices when migrating applications and workloads to the E2000 is discussed.

2 Platform Description

To begin, the E1000 and E2000 hardware platforms are described, including a discussion of select BIOS and Lustre settings. The primary focus of the comparison between platforms will be on the NVMe-based metadata storage targets (MDTs) and object storage targets (OSTs). In the initial E2000 release, the hard drive enclosure drive quantity and the connectivity of the enclosures to the controllers is the same between the two platforms. Lustre client microbenchmarks are provided for hard drive OSTs for completeness but are not substantively discussed.

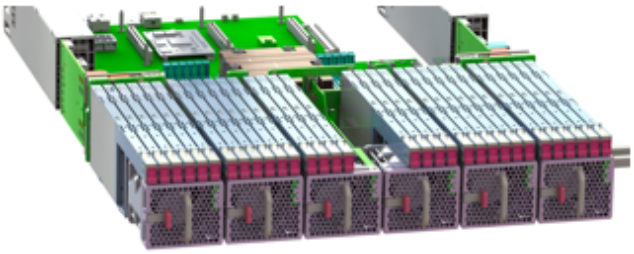

Figure 1: E2000 Scalable Unit, dual controller enclosure

2.1 Hardware Platform

The E2000 hardware platform brings two primary improvements over the E1000 to backend NVMe IO performance: an increase in the per-drive speed from the adoption of the PCIe Gen 5 interface and Gen-5 enabled drives, and a boost in aggregate performance from the expansion in the number of NVMe drives from 24 to 32. As shown in Table 1, the per-drive sequential read and write performance increased by roughly 90% and 50%, respectively, and the per-drive read and write IOPS increased by roughly 60% and 120%, respectively. The change in drive count from 24 to 32 brings a theoretical linear increase in the aggregate chassis performance of 33%.

In addition to backend NVMe performance, the E2000 platform also brings performance gains via CPU improvements, additional faster memory, and the adoption of higher-performing network interconnects. As shown in Table 2, the E2000 adopts the AMD 9354 "Genoa" processor, which gives a 30% clock speed boost over the E1000 CPU, as well as a doubling of L2 and L3 caches and memory bandwidth per socket. Each node in the E2000 comes with 384 GB of DDR5 RAM which can facilitate the use of additional server-side caching although it is disabled by default. Finally, the E2000 supports the use of next-generation high-speed interconnects, for example, the Nvida ConnectX-7 Infiniband HBA. In LNet Self Test benchmarks, a dual-injection ConnectX-7 configuration delivered nearly 100 GB/s of aggregate bandwidth per controller, for a total of 200 GB/s per enclosure.

Drive	Samsung 1733a	Samsung 1743
PCIe	Gen 4	Gen 5
Seq. Read (MB/s)	7,500	14,000
Seq. Write (MB/s)	4,100	6,000
Ran. Read (KIOPS)	1,550	2,500
Ran. Write (KIOPS)	135	300

Table 1: Samsung 1733a and 1743 NVMe Drive Comparison

Component	E1000	E2000
CPU Model	AMD 7502P	AMD 9354P
CPU Cores	32	32
Controllers / enclosure	2	2
CPU Base Clock	2.5 GHz	3.25 GHz
Memory Speed	DDR4 3200MT/s	DDR5 4800 MT/s
Memory Quantity	256 GB, 8 DIMMs	384 GB, 12 DIMMs
PCIe	PCIe gen4	PCIe gen 5
PCIe slots	4	4
NVMe slots	24	32
NVMe form factor	U.2	ES.3
SAS Connectivity	12 Gbps SAS	12 Gbps SAS
Primary NVMe drive	Samsung 1733a	Samsung 1743
Network Connectivity	200 Gbps [1]	400 Gbps [1]

Table 2: E1000 and E2000 Hardware Platform Comparison

2.2 Software Considerations

Realizing the hardware improvements of the E2000 platform requires evaluating a wide range of settings across the BIOS, kernel, and Lustre. While many of the tunings are well understood, several were investigated during platform bring-up. In the microbenchmark section, two specific settings will be discussed to highlight balancing performance across workloads and achieving end-user accessible performance as close to hardware limits as possible. The first setting, NUMA nodes per socket (NPS), defines how CPU cores from multiple Cache/Core Dies (CCDs) are grouped into different NUMA nodes, each with associated physical memory[1]. Memory and PCIe lane locality are the two main hardware resources influenced by changes to this setting. Additionally, an NPS setting of greater than one requires the software to be NUMA aware in order to make effective use of the additional NUMA nodes. Second, the number of CPU partitions defined in the CPU partition table (CPT) of Lustre's libcfs [6] are investigated. The CPT configuration determines the mapping of CPU cores into logically divided partitions and the performance of Lustre components using those partitions, such as the Lustre networking stack (LNet), are impacted by the number of partitions and how cores are placed in those partitions.

3 Microbenchmarks

Microbenchmarks provide the basis for comparison between the two storage platforms. The microbenchmarks will begin at the lowest layer of stack and progress to microbenchmarks emulating

[1] Single or Double Injection per node

OST count	Drives per controller	Write	Write per Drive	Read	Read per Drive
2	16	148855	5815	222759	6971
4	16	150128	5864	224137	7004
4	32	149471	5839	328355	10261

Table 3: E2000-F GridRAID ldiskfs obdfilter survey performance reported in MiB/s

application workloads. Lustre traditionally uses obdfilter-survey [3] to evaluate local, object storage server (OSS) and object storage target (OST) performance using large sequential writes and reads with obdfilter-survey. Moving beyond the storage servers, IOR and mdtest [2] will be used to compare key synthetic workloads between platforms. MLPerf Storage benchmarks will be used to both expand synthetic workloads to include storage microbenchmarks of key ML applications.

3.1 Storage Node Microbenchmarks

Beginning with benchmarks on the storage node allows an assessment of the storage server's efficiency in aggregating individual devices into storage targets. As benchmarks move to higher levels of abstraction or increasing distance from the storage devices, the maximum potential performance is determined by the preceding layer.

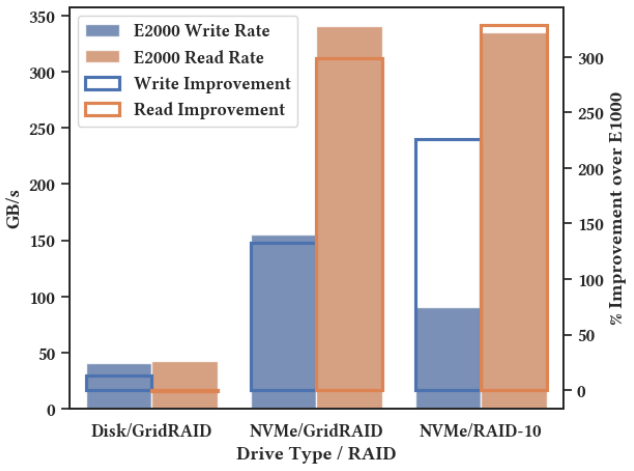

Figure 2: obdfilter-survey Scalable Unit Performance and Relative Improvement from E1000 to E2000 [1]

Improvements in per-drive and per-enclosure performance move the limiting NVMe performance element to the RAID layout of the Lustre targets and its interaction with the physical layout of the E2000 hardware. As shown in Table 2, the E2000 NVMe enclosure, like the E1000, has two controllers per enclosure. Each controller has 2 PCIe lanes to all 32 enclosure slots which gives a maximum observed bandwidth of approximately 7 GiB/s per disk slot for PCIe Gen 5. This PCIe bandwidth limit does not constrain sequential write performance, since per-drive write speed is 6000 MB/s (Table 1). Read performance, however, is constrained by PCIe bandwidth, since maximum sequential read performance from a Samsung 1743 drive is 14000 MB/s, far above the available PCIe bandwidth to a single controller. It is therefore imperative for streaming read performance that drives be accessed from both controllers.

Read performance is optimized in the E2000 by partitioning each disk effectively in half and serving one OST from each partition, so that there are four OSTs per enclosure, each with 16 disks. By default, two OSTs are served from each controller so that all 32 drives are served from both controllers.

The advantage of reading from both controllers is shown clearly in Table 3. The top line shows a 2 OST configuration, where each controller serves 16 drives in a single GridRAID array. In this case, the write rate per drive, at 5815 MiB/s, is close to the spec for this drive model, but the read rate per drive at 6971 MiB/s is limited by the x2 PCIe connection. Increasing the RAID count to 2 per node, but where each node still serves 16 drives, does not significantly increase performance for either writes or reads, indicating that the number of RAIDs does not significantly affect performance. The bottom line shows a configuration with 4 RAID arrays but where each node serves the full 32 drives. Like the two other cases, per-disk write bandwidth is capped by the drives at around 5800 MiB/s, but read performance increases by nearly 50%.

Figure 2 shows the absolute rates and relative improvement of the E2000 from the E1000 using scalable unit level measurements. The compounding improvements of density, accessible per-drive bandwidth, CPU, and memory yield a 130% improvement in write throughput and nearly 300% improvement in read throughput of an E2000-F GridRAID (distributed parity) scalable unit providing a potential peak OST rate of greater than 150 GB/s write and 340 GB/s read [2].

To further investigate the improvements, fio [5] was used directly on the object storage servers (OSS) measuring performance to all NVMe drives simultaneously and to RAID devices simultaneously using an ldiskfs with GridRAID configuration . Figure 3 shows the efficiency of obdfilter and fio, the measured rates relative to drive specification, for E1000 and E2000 NVMe-based scalable units. The `fio to Drives` measurement is relative to all individual drives in the enclosure while `obdfilter` and `fio to RAID` account for parity overhead visible in write performance. The fio measurements are lower in the stack than obdfilter and provide an estimate of the highwater mark for potential Lustre performance. The results show that a larger percentage of available per-drive performance can be achieved on the E2000 platform than on the E1000. Write performance is effectively drive-bound for both fio and obdfilter-survey on the E2000 representing an additional 15% improvement over drive quantity and performance changes. While fio performance to all drives individually is similar between platforms, above 75%, E2000 largely maintains that performance through the stack to Lustre providing an additional 29% improvement in efficiency relative to E1000. Given the similar software stack used between E1000 and E2000 the improved efficiency observed is most likely attributable to memory and CPU changes although the precise contributions were not individually analyzed.

[1] Disk refers to a 2 disk enclosure scalable unit e.g. E2000-D2

[2] Peak OST rates do not take OSS network connectivity into account

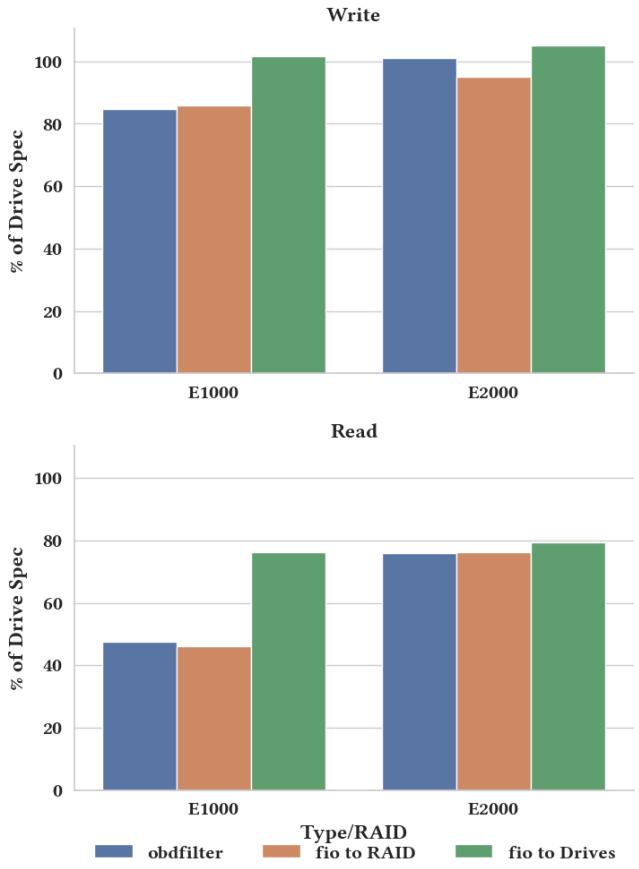

Figure 3: NVMe Scalable Unit GridRAID efficiency relative to drive specification

Through microbenchmarks locally on the OSSes the expected peak performance for scalable units was determined. The results confirm the significant improvements for NVMe OSTs from per-drive performance and drive quantity. Additionally, other hardware changes to CPU and memory yielded improved efficiency allowing the E2000 NVMe targets to be drive limited for throughput-oriented writes and achieve 75% of potential disk read bandwidth. While there may be room for improvement the current plan of record provides two 400 Gbps interfaces per OSS which would mask any additional improvements in read effeciency since read throughput locally exceeds the available network bandwidth to clients. Next, microbenchmarks using the full Lustre client I/O path are evaluated.

3.2 Traditional Storage Microbenchmarks

The high performance computing (HPC) environment has traditionally favored MPI applications for storage microbenchmarks. In this section we use the canonical benchmarks for measuring data, IOR, and metadata, mdtest, performance. While evaluating the communication performance between Lustre clients and servers is an important component in overall performance we do not present results here due to the limited scale of test systems beyond the enclosure-level summary results in Table 4 in section 3.2.1. Although E1000 results are not provided, the 200 Gbps network interfaces on that platform provide at least 22.5 GB/s and 45 GB/s of bi-directional bandwidth for single and dual injection storage servers while the E2000 doubles the rate per injection port and yields at least 45 GB/s and 90 GB/s of bi-directional bandwidth for single and dual injection storage servers. The provided E2000 single storage server measurements are very close to nominal rates for bandwidth.

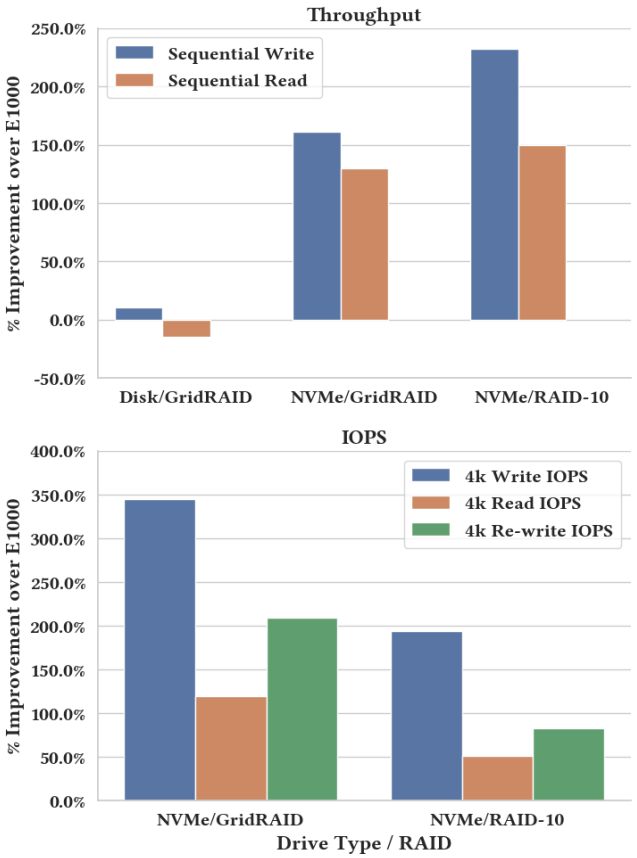

Figure 4: E2000 IOR improvement relative to E1000

3.2.1 IOR. IOR [2] is used to generate a synthetic, homogeneous workload across a set of MPI tasks. The main workload focus for these tests is each MPI rank accessing a single file known as file-per-process. This provides an optimal workload without contention from distributed locks. This workload allows measuring the optimal throughput using 1 MB or larger requests sequentially accessing the file and optimal IOPS using 4 KiB requests to randomly accessed locations in the rank's file. These are not the only relevant workload but provide a key starting point in understanding performance characteristics of a new platform. The results discussed below are buffered, fixed time IOR tests, example test invocations are provided in Appendix A. For a single scalable unit the results are comparable between tests accessing a fixed amount of data or running for a fixed time. Buffered access was chosen since it's more prevalent and with new Hybrid IO [4] functionality in the Lustre

client the distinction between buffered and direct is less critical in performance discussions.

First, we compare top-of-line performance between E1000 and E2000 for sequential and random access workloads. Comparing the local storage improvement in Figure 2 with the client achievable performance in Figure 4 for NVMe based RAID throughput shows the write performance has comparable improvements. Lustre clients are able to utilize the full write bandwidth available to each storage target. However, the read performance shows a relatively large but more modest improvement. The OSS locally achievable performance is limited by the network connectivity of the OSSes, each node has two 400 Gbps interfaces, or an aggregate of 200 GB/s per scalable NVMe unit. As previously described, the local disk bandwidth exceeds the available network bandwidth which limits the relative improvement of the E2000 reads. However, the E2000-F reads are network limited while the E1000-F reads were not. Although local storage tests measuring IOPS were not performed a comparable scale of improvement is achieved for 4 KiB random accesses. Due to the E2000-D using the same enclosure and SAS backend, disk performance is expected to be largely comparable between E1000 and E2000. While there is a slight improvement in write performance an unresolved defect impacting read performance is present and visible in relative performance improvement. The expectation is for read performance to also show a slight improvement relative to the E1000-D1 and E1000-D2 when the defect is resolved.

As discussed in section 2.2, the NUMA nodes per socket (NPS) and CPU partition table (CPT) can have a significant impact on performance. As part of platform bring up performance of microbenchmarks are evaluated across NPS settings and a subset of possible CPT configuration. NPS values are 1, 2, or 4. Although a CPU partition table can be arbitrarily defined the number of partitions was varied to include 4, 8, and 16 using sequential assignment of cores including their corresponding SMT cores. A generally recommended CPT configuration places cores that share an L3 cache in the same partition. since this provides several partitions and optimizes for locality. For the 32-core AMD processors used in the E1000 and E2000 that corresponds to 8 CPTs with 4 physical cores per CPT. Figure 5 shows that while throughput workloads seem to be primarily impacted by NUMA nodes per socket, IOPS workloads are primarily impacted by CPU partitions. The influence of CPT is largely expected, LNet is able to process a higher message count with more partitions regardless of NPS with a doubling between `NPS=1` and `NPS=2`. There may also be some improvement in backend disk performance although that was further investigated in this work. Table 4 shows the throughput and small message rate of an OSS as measured by LNet Selftest (LST) for `NPS=1`. The LST results used a concurrency of 16, which was not optimal for 4k messages, but a small impact of CPT on message rate is visible. An optimal small message rate for LST does demonstrate a rate greater than the reported IOR 4k read IOPS.

The NPS findings on E2000 were unexpected. On E1000, `NPS=2` achieved better performance on ldiskfs for throughput workloads and `NPS=4` achieved better performance on ldiskfs for IOPS workloads. On E2000, `NPS=1` is the higher performing option for the two workloads, with read IOPS being the exception showing slightly improved rates with `NPS=4, CPT=16`. For ZFS, the lack of NUMA-awareness necessitated using `NPS=1`.

Configuration		1 MB RPC, MB/s		4 KB RPC, RPC/s	
NPS	CPT	Write	Read	Write	Read
1	4	99,034	94,444	405,668	414,891
1	8	99,002	99,033	427,851	428,617
1	16	98,953	99,031	454,760	457,183

Table 4: LNet Selftest E2000 OSS Performance

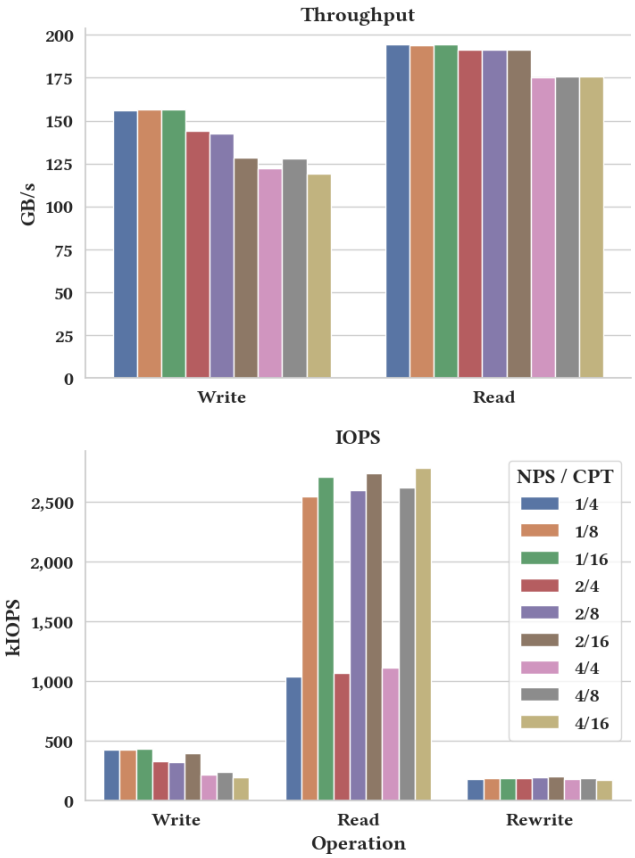

Figure 5: E2000 IOR throughput and IOPS varying OSS NPS and CPT

Enabling simultaneous multithreading (SMT) was also evaluated, although results are not presented. On E1000 storage servers SMT is disabled excluding RAID-10 storage servers. However, on E2000, enabling SMT yielded similar or improved performance across tested workloads and is enabled by default.

All preceding data points were collected using an ldiskfs backend. Beginning with the E1000, ZFS using dRAID2, a distributed parity RAID, was offered as an alternative to ldiskfs on disk and NVMe based storage targets. A full discussion of implementation and feature differences is outside the scope of this paper. We do provide

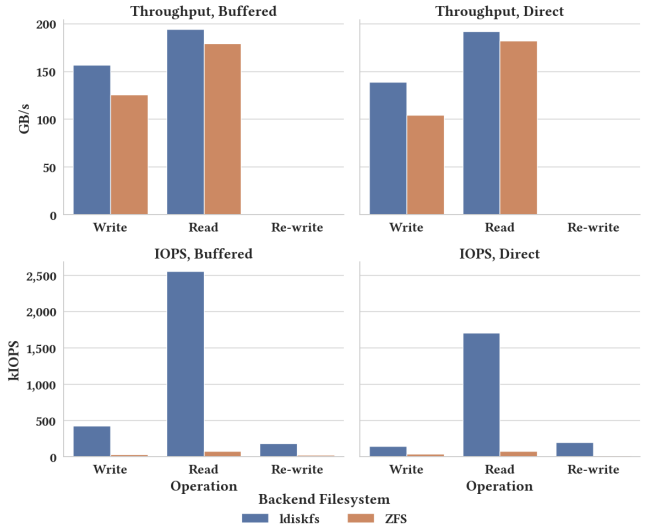

Figure 6: E2000 IOR Performance of ldiskfs and ZFS backends for throughput and IOPS workloads

comparative results for the same workloads on the same system configured with both ldiskfs and ZFS. Both configurations used the same NPS and CPT setting of `NPS=1` and `CPT=8` which are the planned defaults for distributed parity based OSSes. Figure 6 shows a moderate difference between ldiskfs and ZFS in peak IOR throughput. Ldiskfs+ is currently the recommended storage backend filesystem for IOPS workloads.

IOR performance on E2000 shows significant improvements in client accessible performance for both sequential and random access workloads relative to E1000 performance. Throughput workloads are able to achieve near NVMe drive write performance and is network limited for read workloads.

3.2.2 mdtest. To complement IOR's performance measurements of bandwidth and IOPS, mdtest [2] is used for metadata performance generating a synthetic, homogeneous metadata workload across MPI tasks. This benchmark measures rates of creates, stats, and deletes for directories or files with different metadata models of unique or shared directory access. For file operations, the benchmark can provide writing and read of small files. The following results use a single, scalable metadata unit (MDU) comprised of two servers and either 2 or 4 metadata targets (MDTs). All tests include 1-million objects per MDT and report an average of three iterations.

In the first set of tests, we compare peak E2000 metadata performance for unique and shared directory access using 32 compute nodes. As expected, the additional locking required for multiple MPI tasks accessing a shared directory impacts the performance relative to the task-unique directory access shown in Figure 7. For updates needed with creates and removes, this is particularly apparent at more than twice the performance for the unique directories. This is also seen to a lesser degree with the stats and reads, but as they are query operations, less impacted.

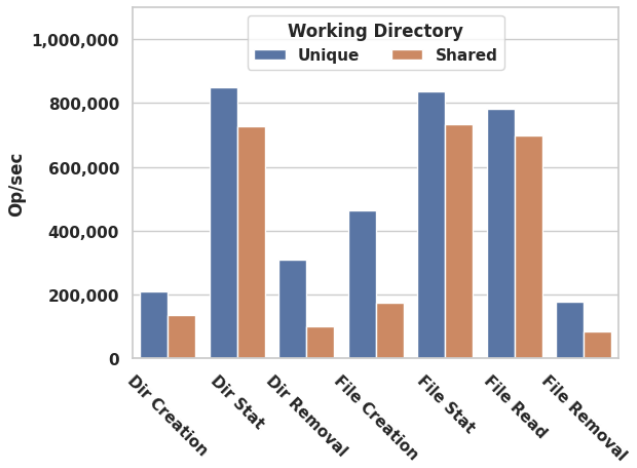

Figure 7: E2000, 4 MDT, ldiskfs metadata performance for unique and shared working directory

To understand these gains over the previous generation, the E2000 was compared against the E1000. In addition to the hardware changes noted earlier, the E2000 system uses NDR Infiniband while the earlier E1000 system is using HDR Infiniband. As well, the E2000 provides 4 MDTs per MDU, whereas the E1000 has 2 MDTs per MDU. For the E1000 configuration, 20 compute nodes were available. Each system used this count for a fair comparison with similar load.

The results provided in Figure 8 show there are gains with the E2000 for the various metadata operations, but of particular note are the file operations of create, stat, and read as these are commonly required of high-performing applications with large file counts.

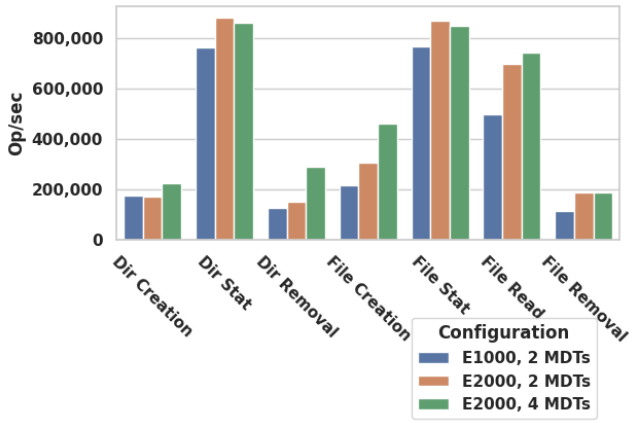

Figure 8: Comparative Performance of single E1000 and E2000 MDU across unique working metadata operation

Though the E1000 uses 2 MDTs for its peak performance, in the process of tuning the E2000 MDU configuration for the product, there was an opportunity to determine any advantages to different MDT counts. With the E2000 the higher performing CPU and

memory bandwidth can accommodate more than 1 MDT, such that the comparison of 1 and 2 MDTs per server in the 2-server MDU was conducted in Figure 9. This showed that there was a noticeable gain for file create performance with moving to 4 MDTs from 2 MDTs in the configuration. Other operations were less improved with the additional MDTs, but the general advantage of high file create rates promoted the 4-MDT design.

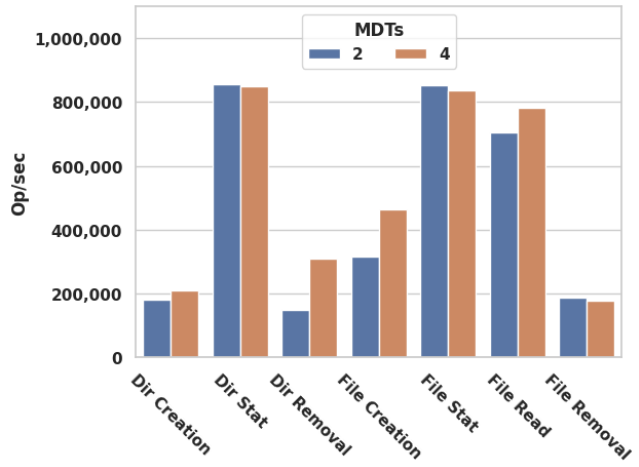

Figure 9: E2000 single MDU performance with ldiskfs using unique working directories using 2 and 4 MDTS

As well, small-file tests were included with 4 KiB- and 32 KiB-files as compared with empty 0 KiB-files. In Figure 10 this shows that there is little difference in small I/O to files for creates, stats, and removes as this can be done asynchronously and with buffering. This is in contrast to the read operation which must be retrieved from the storage target, but is small enough to not take advantage of any readahead.

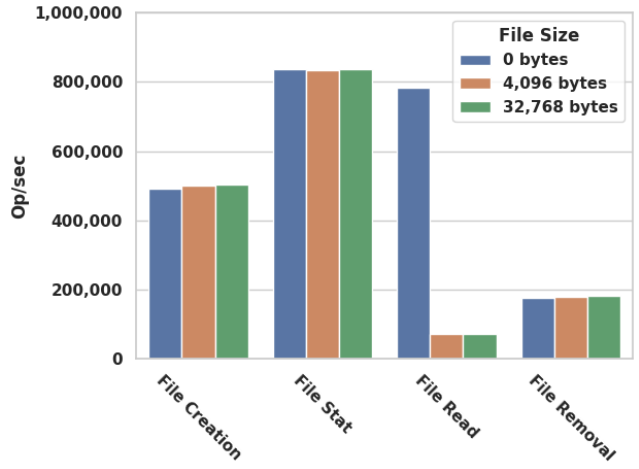

Figure 10: E2000 single MDU performance for file operations varying file size

Similar to the ldiskfs and ZFS comparison for bandwidth and IOPS, results were collected comparing the metadata performance as well. Like the OSTs, the NPS and CPT settings of `NPS=1` and `CPT=8` are used.

In the testing in Figure 11, generally ldiskfs shows a gain over ZFS, with the exception of the file stat performance. This may be on account of the `NPS=1` setting used for ZFS but which has been seen to hamper ldiskfs metadata performance generally, though not covered here.

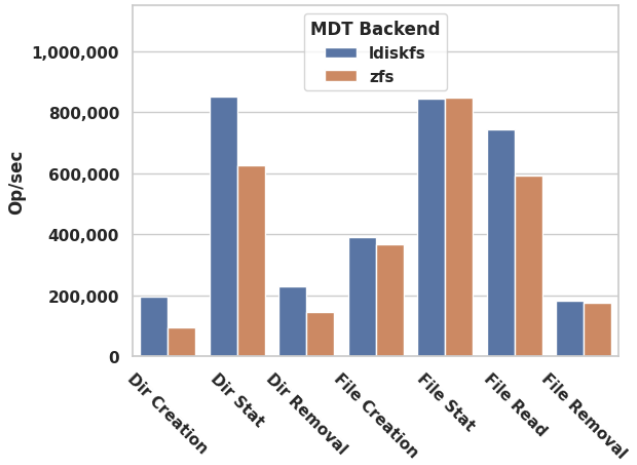

Figure 11: Comparative Performance of single E2000 MDU between ldiskfs and ZFS unique directory metadata operations

For E2000 metadata performance, mdtest demonstrates significant gains across the directory and files operations over its predecessor. This is due to a combination of hardware improvements and changes to the MDT count with some expected gains from general Lustre software improvements.

3.3 MLPerf Storage Benchmarks

The MLPerf™ Storage Benchmark [8] evaluates the performance of storage systems supporting compute clusters executing AI/ML training workloads. It measures how efficiently a storage system can supply data to training processes and achieve high throughput and utilization.

The benchmark has two primary goals:

- Comparability: Ensure fair, standardized evaluation across different submissions using the *CLOSED* submission class.
- Flexibility: Allow experimentation and tuning of system configurations in the *OPEN* submission class to highlight unique features.

MLPerf Storage defines two submission divisions: CLOSED and OPEN. The *CLOSED* class disallows changes to benchmark and storage configurations, ensuring a level playing field and easy comparability. The *OPEN* class permits tuning and customization, with all modifications disclosed in the results.

For each workload, the key performance metric is samples per second, provided that a minimum Accelerator Utilization (AU)

threshold is met. To qualify as a valid run, AU must be at least 90%.

Performance is proportional to the number of GPUs used. Higher values for samples per second and GPU counts indicate better performance.

The MLPerf Storage benchmark suite includes the following workloads, further details on each of the workloads can be found in the benchmark Submission Guidelines [9].

Model	Dataset Seed
3D U-Net	KiTS 19 (140 MB/sample)
ResNet-50	ImageNet (150 KB/sample)
CosmoFlow	N-body simulation (2 MB/sample)

Table 5: MLPerf Storage Benchmark Workloads

The benchmark also supports emulation of NVIDIA H100 and A100 GPUs to evaluate storage system performance across a range of AI hardware.

A complete overview of HPE's MLPerf Storage v1.0 submission using the Cray ClusterStor E1000 is available at the MLCommons website [8]. This submission was made September of 2024.

The sections below compares those submitted MLPerf Storage v1.0 results, obtained using the E1000, with the E2000.

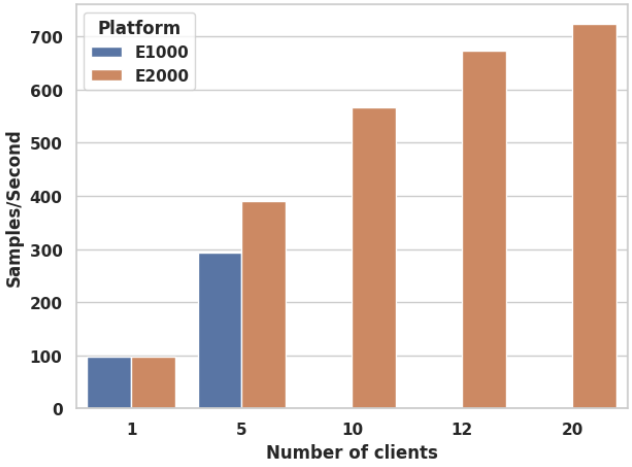

Figure 12: Performance of Unet3D using the E1000 and E2000

3.3.1 UNet3D. Figure 12 illustrates the performance of the UNet3D workload. The bars on the left of each number of clients represent the E1000 results, while the right bars correspond to the E2000. Since the Unet3D submission with the E1000 was made with 5 compute nodes at the time, the figure only includes a direct comparison with 1 and 5 clients. When comparing the performance using 5 compute nodes, the E2000 supported an additional five H100 GPUs compared to the E1000. This results in an increased throughput and samples/second. The E2000 approximately has twice the samples per second than the E1000 which is in line with previously observed microbenchmark results.

Additionally, the E2000 can scale further to:

- 10 clients, supporting 30 H100 GPUs and obtaining approximately 567 samples/second.
- 20 clients, supporting 40 H100 GPUs and obtaining approximately 725 samples/second.

Figure 13: Performance of Resnet50 using the E1000 and E2000

3.3.2 ResNet-50. Similar to UNet3D, the ResNet-50 workload also exhibited significant gains on the E2000.

A single client supported up to 62 H100 GPUs, representing an increase of 16 GPUs compared to the E1000. This resulted in a measurable increase in samples/second, where the E1000 obtained approximately 56,000 samples and the E2000 achieved around 100,000 samples each second.

4 Applications and Transitioning to E2000

Microbenchmarks provide a convenient way to assess performance of specific workloads and compare performance between systems. The microbenchmark assessment of E2000 indicates a dramatic improvement in scalable unit performance over the E1000. However, compute and storage systems are used for applications with heterogenous I/O workloads that are more complex than microbenchmarks can capture. Nex, performance of a specific application and general guidance for sizing client I/O resources migrating from E1000 to E2000 is discussed.

4.1 WRF

WRF is a widely used application for weather forecasting and climate research [11]. There are several methods for writing history and restart files but a single method, using parallel netCDF, is evaluated in the tests to represent a realistic workload applicable to other applications. The previous E1000 microbenchmark results were made on a slightly smaller, 20 nodes, Infiniband system while the E2000 system had 32 nodes. Due to application requirements a different, larger system with dual socket Milan CPU, single injection Slingshot-200 compute nodes were used for the E1000 tests;

32 nodes were used in all reported results. The compute node differences preclude application walltime comparison, and to a lesser extent I/O performance comparisons, but the I/O times and reported statistics provide a reasonable comparison between the two storage platforms.

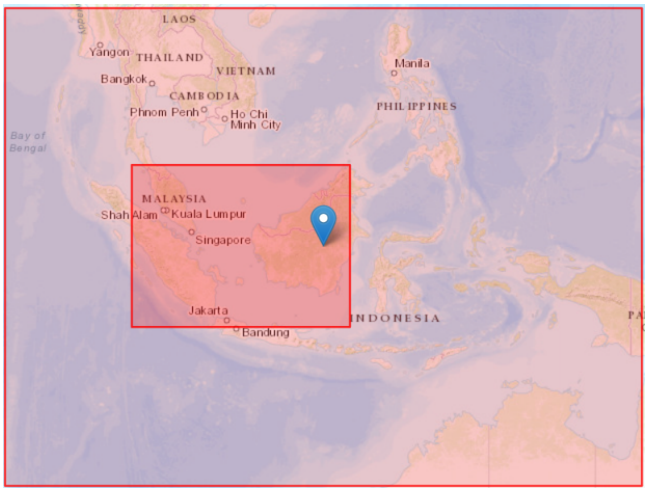

Figure 14: WRF nested domain use case

The WRF simulation contains two nested domains with 3 km (domain 1) and 1 km (domain 2) resolution. A 70-minute simulation is ran using a 9-second timestep with 51 vertical levels. History files are written by domain 1 every 3 minutes and domain two every 1 minute. Restart files are written by both domains every 15 minutes. The total input size is 29 GB and output size is 1.7 TB. The individual history and restart files are relatively small but provide a meaningful amount of writes for the small scale jobs. Jobs are ran using a total of 32 compute nodes on each system. Parallel NetCDF uses collective MPI-IO to write files which allows for use of collective buffering for more optimal shared file write performance. WRF reports the time spent writing each individual history and restart file. The cumulative total of the individual restart and history write times are the basis for evaluating the performance of each platform and Lustre striping and MPI-IO hints.

The application I/O workload was profiled using Darshan [7] and I/O operation counts from the profile are show by API in Figure 15. The I/O operation counts are reported for a test that used collective buffering with 32 aggregators for history and restart file writes with Lustre Lockahead [10] and disabled collective buffering for reads. The operation counts match the expectation that collective writes and reads are the predominate calls. The POSIX write and read counts match the use of collective buffering for writes and not for reads – the count of POSIX writes is lower than MPI-IO and higher for POSIX reads. Disabling collective buffering for writes saw significantly higher I/O times and is not further evaluated. Enabling and Disabling collective buffering for input file reads did not impact overall run time and was disabled in all reported results.

Two main collective buffering parameters were evaluated for testing collective MPI-IO write performance. First, as shown in the

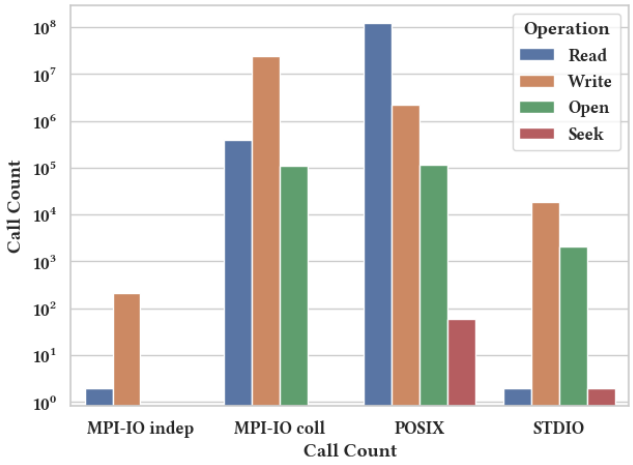

Figure 15: WRF I/O Operations by API

upper plot of Figure 16, the number of collective buffering aggregator ranks per OST is evaluated with Lustre Lockahead. Although dependent on the specific workload and other factors, in this case the benefit of additional aggregator ranks on I/O time is clear. Due to the relatively small files and small collective writes, roughly 75% are 10 KB to 100 KB, the performance of the shared file writes achieve a fraction of the previous discussed peak write rates of the E2000-F. Cray MPICH MPI-IO timers report 10 - 12 GB/s of net write bandwidth for each restart or output file. The second parameter evaluated is the combination of Lustre locking and Lustre striping. The lower plot in Figure 16 shows I/O write time using Lustre default locking and Lustre Lockahead with 4 or 32 total stripes on 4 OSTs with all tests using 32 collective buffering aggregator rank, 1 per compute node. In the case of 32 Lustre stripes on 4 OSTs, each aggregator is writing to it's own stripe and there is no lock contention. Comparing the `32,Default` and `32,Lockahead` timing provides a measurement of the overhead associated with Lustre Lockahead when there is no benefit on file writes. There is overhead on file open to begin using Lustre Lockahead and in this workload that overhead is incurred for 103 files and cumulatively account for an additional 4% - 5% of I/O write time compared to the default Lustre locking case. While small or medium scale jobs may be able to use the strategy of one Lustre stripe per compute node, either with or without overstriping, it isn't an effective use of resources without overstriping (1 rank per OST is 1-2 GB/s of writes) and is not plausible at large scale even with overstriping. The use of 4 OSTs with 1 stripe per OST shows the clear advantage of Lustre Lockahead, `4,Lockahead` relative to default Lustre locking, `4,Default`, when multiple aggregator ranks per OST are used. To maximize performance for both disk and NVMe OSTs, multiple aggregators per OST are recommended. The key insights from Figure 16 is that while Lustre Lockahead does have a small overhead it provides a scalable solution for shared file writes that is comparable to the one stripe per aggregator option that is not scalable. Also note, even with 32 Lustre stripes, the sub-stripe sized writes of this workload mean collective buffering provides an order of magnitude reduction in write times compared to disabling collective buffering.

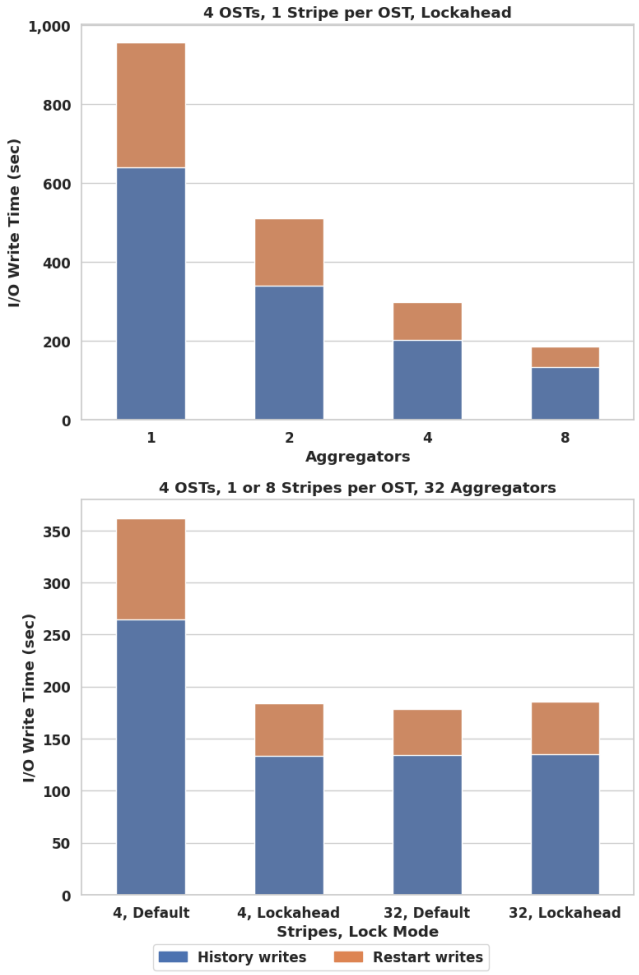

Figure 16: WRF write time comparison between Lustre Striping and MPI-IO Hints

Figure 17 includes a comparison of two optimal collective buffering write configurations, all using 32 collective buffering aggregator ranks, with default Lustre locking and one stripe per aggregator (32 stripes) and using Lustre Lockahead with 1 stripe per OST (4 stripes) on the E1000 and E2000. Although there are differences in the compute nodes and fabric of the systems used, looking at only I/O write times the E2000 shows a 40 - 50% reduction in I/O write time for a single E2000-F compared to a single E1000-F. This improvement corresponds to the difference in Cray MPICH reported collective MPI-IO write net bandwidth rates of history and restart file writes of between 6 - 8 GB/s for the E1000 configuration and 10 - 12 GB/s for the E2000. For this specific use case the improvement in I/O time, around 190 seconds, kept the percentage of the overall job time spent on I/O relatively constant, 13% of total walltime on E1000 and 12.4% of total walltime on E2000. The walltime for the WRF use case on the previous generation compute and storage was around 2900 seconds and 1485 seconds for the current generation. This is an important data point, keeping I/O time constant across CPU and storage generations is a critical part of ensuring efficient overall system utilization.

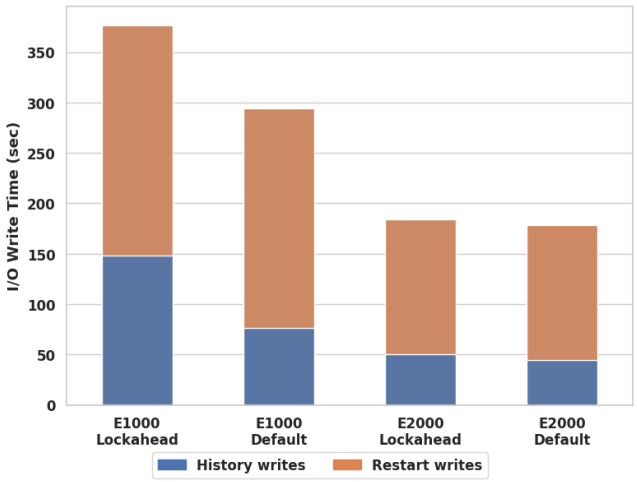

Figure 17: WRF write time comparison between 1 E1000-F and 1 E2000-F for optimal default and Lustre Lockahead MPI-IO hints

4.2 Application changes to utilize E2000

There are two key items to consider when moving an application or workflow from a previous generation platform, such as E1000, to fully utilize the E2000 – the single storage target performance and the storage target count per scalable unit. In summary, the E2000-D will have the same target counts as it's corresponding E1000 configuration with slightly higher performance. The E2000 MDU will have double the MDTs, 4, with similar or better per-MDT performance depending on the workload. The E2000-F will also have double the OSTs, 4, with slightly better to significantly better performance for each storage target depending on the workload – meaning at least 2 times the performance per scalable unit as discussed in section 3.2.1.

4.2.1 Metadata. For applications making use of a single E1000 MDT without a meaningful metadata load no changes should be required moving to E2000 and there may be a decrease in metadata time depending on the workload. For applications currently using multiple MDTs through Lustre DNE on E1000 there may also be an improvement in metadata time with no change. However, doubling the number of MDTs used, or increasing the count for very large MDT counts, should see meaningful reductions in metadata time if the application benefits from multiple MDTs. Although not included in results it does not seem additional compute nodes will be necessary to achieve peak metadata rates based on the limited results available at this time.

4.2.2 E2000-F. Applications using E2000-F OSTs will need to consider changes based on how their current compute I/O resource sizing is calculated and how well the applications utilizes existing targets. As described earlier, the per OST performance between E1000 and E2000 is comparable so for cases where I/O rank or node

count, MPI-IO collective buffering aggregator count, or other means to size resources are calculated on Lustre stripe count changes may not be required. Figure 18 shows peak throughput and write IOPS performance can be achieved with 16 nodes, for a fixed process-per node-value of 8, while read IOPS continue seeing improved rates with increasing node counts up to 32. While the curves may look slightly different for other process-per-node counts, for buffered workloads it tend to be node limited by 8 PPN. While the per OST performance remains comparable the scalable unit performance is at least double depending on the workload. In the case of comparing I/O resources per scalable unit, approximately double the compute I/O resources would be needed to drive the peak E2000 performance.

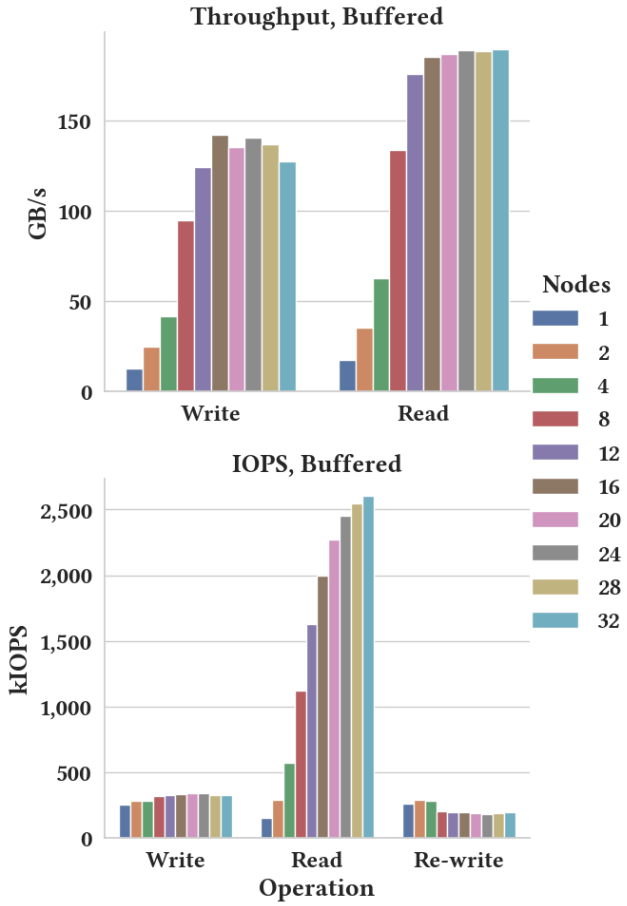

Figure 18: Throughput and IOPS when scaling compute node count

5 Conclusion

The HPE Cray Supercomputing Storage Systems E2000 is the next generation storage, providing significant improvements in both throughput and IOPS workloads over the preceding generation. Storage node microbenchmarks demonstrate not only increased per-drive performance but improved efficiency that exposes at or near drive performance limits up through the Lustre Object Storage Target. Lustre client microbenchmarks confirm that the E2000-F can deliver near NVMe-limited write performance of 150 GB/s and network-limited read performance approaching 200 GB/s which represents a greater than 150% and 100% increase respectively over the E1000 for the same hardware footprint for ldiskfs on GridRAID targets. These gains represent more than just generational hardware refreshes or platform changes – the solution, including the software stack, is well-balanced and able to expose the full hardware capabiltities of the platform to application usable performance.

Application workloads introduce additional complexity not captured in traditional microbenchmarks. For the specific WRF use case evaluated the application was able to achieve similar improvements to the microbenchmarks in I/O write time even though the application used a non-optimal, but real-world, I/O workload. The roughly 100% improvement in application visible I/O performance with the E2000 meant that the reduced application walltime from compute node changes, had the same percentage of application time spent on I/O, roughly 12% in this case. I/O is a necessary component in most, if not all, workflows and keeping a constant percentage of application time in I/O prevents eroding generational compute nodes with I/O requirements. Finally, for users migrating between platforms applications will need a similar number of compute resources per storage target, such as collective buffering aggregators or I/O nodes, making the transition easy for many applications while still allowing aplpications to capture the large performance gains for each E2000-F scalable unit.

Acknowledgments

Thank you to Hyei-Sun Park for her assistance building WRF and developing a use case for the specific test environment.
Thank you to Ayad Jassim and Mausmi Kotecha for their assistance in building and running OpenFOAM.

References

[1] AMD. 2021. *Socket SP3 Platform NUMA Topology for AMD Family 19h Models 00h–0Fh.* AMD. Retrieved March 25, 2025 from https://www.amd.com/content/dam/amd/en/documents/processor-tech-docs/design-guides/56795_1_00-PUB.pdf
[2] IOR Developers. 2024. IOR and mdtest. https://github.com/hpc/ior Version 4.0.0.
[3] Lustre Devlopers. 2007. README.obdfilter-survey. Retrieved March 25, 2025 from https://github.com/hpc/lustre/blob/master/lustre-iokit/obdfilter-survey/README.obdfilter-survey
[4] Patrick Farrell. 2024. LUG 2024: Hybrid IO Update. https://wiki.lustre.org/images/a/a0/LUG2024-Hybrid_IO_Path_Update-Farrell.pdf Accessed: 2025-04-11.
[5] FIO. 2024. FIO - Flexible I/O Tester. https://github.com/fio/fio Accessed: 2025-04-02.
[6] Anjus George, Rick Mohr, James Simmons, and Sarp Oral. 2021. *Understanding Lustre Internals - Second Edition.* Technical Report ORNL/TM-2021/2212. Oak Ridge National Laboratory. https://info.ornl.gov/sites/publications/Files/Pub166872.pdf
[7] Jakob Luettgau, Shane Snyder, Tyler Reddy, Nikolaus Awtrey, Kevin Harms, Jean Luca Bez, Rui Wang, Rob Latham, and Philip Carns. 2023. Enabling Agile Analysis of I/O Performance Data with PyDarshan. In *Proceedings of the SC '23 Workshops of the International Conference on High Performance Computing, Network, Storage, and Analysis* (Denver, CO, USA) *(SC-W '23)*. Association for Computing Machinery, New York, NY, USA, 1380–1391. doi:10.1145/3624062.3624207
[8] MLCommons. 2024. MLPerf Storage v1.0 results. https://github.com/mlcommons/storage_results_v1.0
[9] MLCommons. 2024. MLPerf Storage v1.0 Submission Guidelines. https://github.com/mlcommons/storage/blob/main/Submission_guidelines.md
[10] Michael Moore, Patrick Farrell, and Bob Cernohous. 2017. Lustre Lockahead: Early Experience and Performance Using Optimized Locking. In *Proceedings of*

the Cray User Group (CUG) 2017. Cray User Group, Redmond, WA, USA. https://cug.org/proceedings/cug2017_proceedings/includes/files/pap141s2-file1.pdf

[11] W. C. Skamarock and Coauthors. 2008. A description of the Advanced Research WRF version 3. *NCAR Tech. Note NCAR/TN-475+STR* (2008), 113. doi:10.5065/D68S4MVH

A Microbenchmark Test Methodology

fio, all NVMe devices

```
[global]
rw=read
bs=4M
direct=1
ioengine=posixaio
iodepth=64
numjobs=16
group_reporting=1
time_based=1
runtime=60
ramp_time=5
# offset for accessing the same device on both controllers
offset=[0|2]T
# one job per NVMe device
[jobX]
filename=/dev/nvmeYn1
```

fio, RAID devices

```
[global]
rw=read
bs=4M
direct=1
iodepth=64
numjobs=384
group_reporting=1
time_based=1
runtime=60
ramp_time=5
offset=[0|2]T

# one job per RAID device
[jobX]
filename=/dev/mdY
```

A.1 obdfilter-survey

Fstrim was ran prior to performing tests on NVMe devices

```
tests_str='write read' \
    nobjlo=1 nobjhi=1 \
    thrlo=1024 thrhi=1024 \
    rszlo=4096 rszhi=4096 \
    size=1048576 \
    obdfilter-survey
```

A.2 IOR testing

The general test methodology used for each IOR test

- Manually creating files ensuring each node was writing the same number of files per OST
- Dropping kernel caches on compute nodes after each individual write and read test
- Running fstrim before each test
- Throughput tests used 20 compute nodes with 8 PPN for E1000 and 32 compute nodes with 8 PPN for E2000 tests.
- IOPS tests used 20 compute nodes with 24 PPN for E1000 and 32 compute nodes with 24 PPN for E2000 tests.

Fixed data and fixed time tests were ran for throughput tests, only fixed time tests were ran for IOPS tests. Using larger files minimizes adjacent random accesses but also causes extremely long test times.

IOR, throughput, fixed time, direct IO

```
IOR -C -e -E -F -v -k -o TESTDIR/IOR  -t 64m \
    --posix.odirect -b 512g -w -D 500
IOR -C -e -E -F -v -k -o TESTDIR/IOR -t 64m \
    --posix.odirect -b 512g -r -D 60
```

IOR, throughput, fixed data, buffered

```
IOR -C -e -E -F -v -k -o TESTDIR/IOR -t 1m \
    -b 256g -w
IOR -C -e -E -F -v -k -o TESTDIR/IOR -t 1m \
    -b 256g -r
```

IOR, IOPS, fixed data, buffered

```
IOR -E -F -v -k -b 8g -o TESTDIR/IOR -w \
    -D 180 -t 4k -z
IOR -E -F -v -k -b 8g -o TESTDIR/IOR -w \
    -D 180 -t 4k -z
IOR -E -F -v -k -b 8g -o TESTDIR/IOR \
    --posix.odirect -w -t 64m
IOR -E -F -v -k -b 8g -o TESTDIR/IOR -r \
    -D 180 -t 4k -z
```

A.3 MDTEST testing

The general test methodology used for each MDTEST test

- Dropping kernel caches on compute nodes before and after each individual mdtest run.
- E2000-only tests used 32 compute nodes with 16 PPN for E2000 tests.
- E1000 and E2000 comparison tests used 20 compute nodes with 16 PPN.
- lfs mkdir -c MDT_COUNT -i -X 2 BASEDIR used for setting default layout.
- 1,048,576 objects per MDT used for all tests.
- File-only tests included verbatim-Fverbatim only

MDTEST, task-unique directories

```
MDTEST -n (( MDT_COUNT * OBJECTS_PER_MDT / PROCS )) \
    -d BASEDIR -u -w FILE_SIZE -e FILE_SIZE \
    -C -E -T -r -i 3 -p 30 -N 1 -v -v
```

MDTEST, single-shared directory

```
MDTEST -n (( MDT_COUNT * OBJECTS_PER_MDT / PROCS )) \
```

```
    -d BASEDIR -w FILE_SIZE -e FILE_SIZE \
    -C -E -T -r -i 3 -p 30 -N 1 -v -v
```

A.4 MLPerf Storage

The MLPerf Storage v1.0 guidelines were followed when performing the tests for the Unet3d and the Resnet50 workloads. These include,

- Generating datasets for the workload that satisfy the minimum dataset size requirement from MLPerf Storage.
- Dropping kernel caches in between each benchmark runs.
- Reporting results with a passing Accelerator Utilization (AU) only. Currently, a 90 AU is required to pass the benchmarks.

Dataset generation for Unet3D

```
./benchmark.sh datagen --hosts hostname.txt \
--workload unet3d --accelerator-type h100 \
--num-parallel 16 \
--param dataset.num_files_train=140000 \
--param dataset.data_folder=unet3d_data
```

Dataset generation for Resnet50

```
./benchmark.sh datagen --hosts hostname.txt \
--workload resnet50 --accelerator-type h100 \
--num-parallel 34 --param dataset.num_files_train=455000 \
--param dataset.data_folder=resnet50_data
```

Unet3d benchmark run

```
./benchmark.sh run --hosts hostname.txt \
--workload unet3d --accelerator-type h100 \
--num-accelerators 40 --results-dir unet3d_h100 \
--param dataset.num_files_train=140000 \
--param dataset.data_folder=unet3d_data
```

Resnet50 benchmark run

```
./benchmark.sh run --hosts hostname.txt \
--workload resnet50 --accelerator-type h100 \
--num-accelerators 62 --results-dir resnet_h100 \
--param dataset.num_files_train=102740 \
--param dataset.data_folder=resnet50_data
```

A.5 WRF

Input files were striped across 2 and 4 OSTs with a stripe size of 1 MB. Output files were also striped across 2 and 4 OSTs with a stripe-size of 1 MB. As noted in results Lustre overstriping was used across either 2 or 4 OSTs. The use of Quilt I/O servers was evaluated but due to the small use case they were not beneficial in reducing run time and were not included in the analysis. The following MPI-IO hints were used to enable Lustre lockahead with collective buffering for writes and disabling collective buffering and data sieving for reads. The number of aggregators per OST (`cray_cb_nodes_multiplier`) are set as indicated in results. Formatted for clarity.

```
MPICH_MPIIO_HINTS="
   wrfout*:romio_cb_write=enable:romio_cb_read=enable:\
      cray_cb_nodes_multiplier=8:\
      cray_cb_write_lock_mode=2:\
      romio_ds_write=disable:romio_ds_read=disable:\
   wrfrst*:romio_cb_write=enable:romio_cb_read=enable:\
      cray_cb_nodes_multiplier=8:\
      cray_cb_write_lock_mode=2:\
      romio_ds_write=disable:romio_ds_read=disable\
   wrfinput*:romio_cb_write=disable:romio_cb_read=disable:\
      romio_ds_write=disable:romio_ds_read=disable"
```