Latency and Bandwidth Microbenchmarks of US Department of Energy Systems in the June 2023 Top500 List

Christopher M. Siefert
Sandia National Laboratories
Albuquerque, NM, USA
csiefer@sandia.gov

Carl Pearson
Sandia National Laboratories
Albuquerque, NM, USA
cwpears@sandia.gov

Stephen L. Olivier
Sandia National Laboratories
Albuquerque, NM, USA
slolivi@sandia.gov

Andrey Prokopenko
Oak Ridge National Laboratory
Oak Ridge, TN, USA
prokopenkoav@ornl.gov

Jonathan J. Hu
Sandia National Laboratories
Albuquerque, NM, USA
jhu@sandia.gov

Timothy J. Fuller
Sandia National Laboratories
Albuquerque, NM, USA
tjfulle@sandia.gov

ABSTRACT

As a rule, Top 500 class supercomputers are extensively benchmarked as part of their acceptance testing process. However, barring publicly posted LINPACK / HPCG results, most benchmark results are often inaccessible outside the hosting institution. Moreover, these higher level benchmarks do not provide easy answers to common questions such as "What is the realizable memory bandwidth?" or "What is the launch latency on the accelerator?" To partially address these issues, we executed selected single-node micro-benchmarks — focused on latencies and memory bandwidth — on every US Department of Energy system above rank 150 of the June 2023 Top 500 list, excepting NERSC's Cori and ORNL's Frontier TDS (now decommissioned or repurposed). We hope to provide an easy "first stop" reference for users of current Top 500 systems and inspire users and administrators of other Top 500 systems to similarly compile and make available benchmark results for their systems.

CCS CONCEPTS

• **Computer systems organization → Multicore architectures**; • **Hardware → Testing with distributed and parallel systems**.

SC-W 2023, November 12–17, 2023, Denver, CO, USA

ACM ISBN 979-8-4007-0785-8/23/11…$15.00
https://doi.org/10.1145/3624062.3624203

KEYWORDS

high performance computing, micro-benchmarking, top500, supercomputing

ACM Reference Format:
Christopher M. Siefert, Carl Pearson, Stephen L. Olivier, Andrey Prokopenko, Jonathan J. Hu, and Timothy J. Fuller. 2023. Latency and Bandwidth Microbenchmarks of US Department of Energy Systems in the June 2023 Top500 List. In *Workshops of The International Conference on High Performance Computing, Network, Storage, and Analysis (SC-W 2023), November 12–17, 2023, Denver, CO, USA.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3624062.3624203

1 INTRODUCTION

As part of acceptance testing, a Top 500 [17] class computer typically undergoes extensive benchmarking. As an example, an early (Knights Corner) precursor of Los Alamos National Laboratory's (LANL) Trinity machine was extensively documented with two micro-benchmarks as well as eight application benchmarks by Rajan et al., [33] over a year before Trinity's expected full deployment. A similar set of benchmarks would be run for the Knights Landing (KNL) nodes later in the same acceptance testing process [35]. As is somewhat typical, these reports either consider the new system in isolation (the former report) or compare the new system against a single older system (in this case LANL's prior Cielo supercomputer). While they often give an excellent snapshot of the machine's performance at a single point in time (or over the period of acceptance testing as the software toolchain is refined), these results tend to be myopic in focus – after all, they are designed to test a *single* machine.

Developers of portable application codes are interested in not *one* machine, but many. They often want to know how machine characteristics compare between platforms. Some information (e.g., GPU and CPU model information)

are stored at the Top 500 site, as well as LINPACK [2] and HPCG [7] benchmark information. But the Top 500 represents the apex of collected data for supercomputing systems. To get other benchmarking data on these systems, one must find the relevant technical reports or proceedings papers.

The goal of this paper is to provide a collection of benchmarking results on all extant US Department of Energy (DOE) systems of rank 150 or above in the June 2023 Top 500. These represent systems of active use by DOE application developers and provide a "first stop" for developers looking for answers to performance questions which can be addressed by microbenchmarks. In particular, we focus on node level performance. Inter-node performance, while of interest to application developers, is highly dependent on network topology and loading [20] and there is a distinct lack of vendor portable methods to obtain information about where in the machine a particular job happened to be scheduled [24]. But it is not just for the reason of simplicity that node level performance is the focus of this paper. For modern accelerator-based systems (at this point in time consisting NVIDIA or AMD GPUs) the behavior of the accelerators can vary substantially, with latencies and bandwidths changing noticeably between accelerators. While accelerator vendors tend to highlight ideal bandwidths in their promotional materials, latencies tend to be mentioned only in passing in said material, if at all.

We present a summary of our chosen microbenchmarks in section 3, distinguishing between benchmarks run on accelerator and non-accelerator platforms. We then present computational results on US DOE platforms in section 4. Finally, we present conclusions and suggestions for future work in section 5.

2 RELATED WORK

Intra-node microbenchmarks have a long history of interest. Our work is primarily distinguished not by the development of novel microbenchmarks, but by leveraging a key set of well-understood existing microbenchmarks to summarize the performance of a representative set of high performance systems. Benchmark suites like NAS Parallel Benchmarks [18] and miniapplication suites like Mantevo [22] can be helpful for understanding system performance on classes of applications, but their increased complexity compared to microbenchmarks can make it difficult to isolate particular system characteristics like bandwidth and latency.

Perhaps the most well-known HPC microbenchmark is STREAM, designed to capture sustainable memory bandwidth [31]. That work popularized the key observation that CPU performance was improving much faster than memory bandwidth. Hence, this paper does not seek to measure sustained FLOPs, but rather, various intra-node data transfer rates. McCalpin went on to evaluate STREAM benchmark results on more than a dozen systems [30], a model our work seeks to emulate. On the inter-node side, Liu et al. [29] offer a microbenchmark comparison of Myrinet, Quadrics, and InfiniBand interconnects. They focus on latency, bandwidth, CPU time, and message overheads.

The widespread adoption of GPU-accelerated systems has led to corresponding interest in GPU microbenchmarks. Bureddy et al. [21] introduce a wide variety of GPU+MPI microbenchmarks, which they evaluate on a single two-node computer. This implicitly acknowledges the challenge of such an evaluation on a realistic system, which we expect to address in future work. Ji et al. [25] examine the relationship between latency and transfer size for host-host, host-GPU, and GPU-GPU communications, in the context of considering how to create a GPU-aware MPI implementation.

BabelStream is a version of STREAM ported to a variety of parallel programming models. Deakin et al. [23] introduce and evaluate BabelStream on 14 different CPUs and GPUs for McCalpin's STREAM plus the implementations created for six additional programming models. We also use BabelStream to compare achievable bandwidths on the systems, but our system selection is motivated by a cross-section of the Top500, rather than covering all practical GPU architectures and programming models. Comm|Scope [32] and Li et al. [27, 28] both developed and evaluated intra-node communication microbenchmarks focused on high-bandwidth interconnects (and collectives in Li et al.). Our work uses version 0.12.0 of Comm|Scope, which added support for AMD GPUs through the HIP programming model.

Khorassani, Chi, Subramoni, and Panda [26] provide a detailed performance evaluation of SpectrumMPI, OpenMPI+UCX, and MVAPICH2-GDR on Summit and Sierra. They do not report numerical values for GPU-to-GPU MPI latency (as we do in Table 5), but our results appear consistent with theirs. They observe substantial latency differences in some MPI implementations on the same system. Our evaluation hews to the default configuration of each platform, but testing multiple implementations when available may be considered as future work.

There have been a variety of independent efforts to develop benchmarks for MPI (and pre-MPI) networks [5, 6, 14, 19]. All provide point-to-point latency benchmarks among many others. Of particular interest is [19], which includes a performance evaluation of networks from 1990 to 2002, providing a useful reference at the time. Our work uses the OSU implementation due to its familiarity to the community.

3 MICROBENCHMARKS OF INTEREST

Our selection of microbenchmarks reflects our focus on node-level performance. We consider two different families of microbenchmarks, depending on whether the system contains

OMP_NUM_THREADS	OMP_PROC_BIND	OMP_PLACES
1	not set	not set
1	“true”	not set
#cores	not set	not set
#cores	“true”	not set
#cores	“spread”	“cores”
#threads	not set	not set
#threads	“true”	not set
#threads	“close”	“threads”

Table 1: Combinations of OpenMP environment variables used for testing the maximum achievable host memory bandwidth, both in the single thread and “all threads” cases.

an accelerator (e.g. NVIDIA or AMD GPU), or not (e.g., a CPU-only system or a self-hosted Intel Xeon Phi).

3.1 Non-Accelerator Architectures

For non-accelerator architectures, we consider host memory bandwidth as measured by the OpenMP backend of BabelStream 4.0 [23]. We estimate realizable single-thread memory bandwidth as well as the bandwidth attainable when using all available threads. As the maximum number of cores and the maximum number of SMT threads are not always the same, we test several combinations of OpenMP options and report the highest realized memory bandwidth in section 4. These combinations are listed on Table 1. BabelStream 4.0 does not account for any write-allocate traffic; the bandwidth numerator is twice the allocation size for copy, mul, and dot, and three times the allocation size for Add and Triad.

In addition, we measure point-to-point MPI latency using the OSU Micro-Benchmarks 7.1.1 [14]. For CPU systems we estimate two different single-node latencies, latency between two MPI processes assigned to the same processor (“on-socket”) and latency between MPI processes assigned to two different processors (“on-node”). DOE applications commonly use one MPI rank core for CPU runs, or per accelerator for GPU system (as opposed to one MPI rank per node + OpenMP within a node), so intra-node MPI communication performance is a relevant measurement. For Xeon Phi systems, they are run in “quad” mode with a single NUMA domain, but we still consider both a “close” core pair — cores 0 and 1 — which we record under “on-socket,” and a “far” core pair — cores 0 and $N-1$, where N is the number of cores on the Xeon Phi — which is recorded under “on-node.”

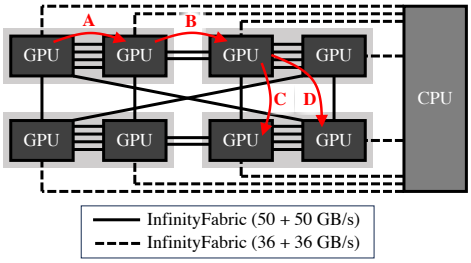

Figure 1: Frontier node diagram, based on machine documentation [11]. Arrows indicate different connections measured in latency experiments reported in Table 5 and Table 6. RZVernal and Tioga share a similar node topology.

3.2 Accelerator Architectures

As with the non-accelerator architectures (subsection 3.1), we begin with the consideration of on-device memory latency measured by the CUDA or ROCm (as appropriate) backend of BabelStream 4.0 [23]. MPI latency is measured in two different ways — host-to-host and device-to-device, again via OSU Micro-Benchmarks 7.1.1 [14]. In addition, we consider device kernel launch and empty queue wait costs, as well as host-to-device, device-to-host and device-to-device memory latency and bandwidth. These are computed using Comm|Scope v0.12.0 [32]. Similar to the case of non-accelerator machines, not all GPUs on the system are equidistant from each other. The GPU topologies vary from system to system, with relevant details presented in Figs.1, 2, and 3.

4 MICROBENCHMARKING RESULTS

In this study we consider every active US Department of Energy (DOE) system above rank 150 in the June 2023 Top 500 list [17]. We divide the list into non-accelerator and accelerator based systems, which are shown in Table 2 and Table 3 accordingly. Note that even in some cases where the Table 3 shows identical system summaries, there still can be differences as not every detail of the systems is represented. The diagrams in Figure 1, Figure 2, and Figure 3 show the node topologies of the different classes of GPU systems, providing context for the results presented later in this section.

Binaries for each of the three tests, OSU Micro-Benchmarks, BabelStream and Comm|Scope are executed 100 times. The mean and standard deviation are calculated across those 100 tests. Within the binary tests are repeated multiple times. We used the default settings for repeats within the binaries. For the OSU Micro-Benchmarks, this setting involves 1,000 repeats for small messages and 100 messages for large ones. For BabelStream and Comm|Scope it is 100 repeats.

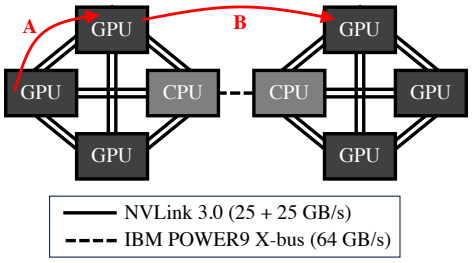

Figure 2: Summit node diagram, based on machine documentation [16]. Arrows indicate different connections measured in latency experiments reported in Table 5 and Table 6. Sierra and Lassen share a similar node topology, except that they have four GPUs per node rather than six.

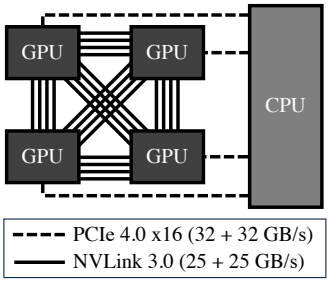

Figure 3: Perlmutter node diagram, based on machine documentation [15]. Polaris shares a similar node topology.

For BabelStream on non-accelerator systems, we choose the highest single and multicore memory bandwidth from the OpenMP configuration options described in Table 1 chosen over all the possible BabelStream operations (i.e., Copy, Mul, Add, Triad and Dot) for the largest vector size we ran (which is at least 128 MB in every case). For BabelStream on accelerator systems, we do not use OpenMP and thus pick the best over all of the BabelStream operations for the largest vector size we ran (1 GB for all accelerator systems). We note that on the MI250X platforms, BabelStream only uses one of the two Graphics Compute Dies (GCDs), which is why the reported memory bandwidth is less than half of the notional 3276.8 GB/s advertised by AMD [9]. We declined to report CPU memory bandwidth results on the accelerator systems since in many applications, the CPU is used primarily for coordinating device kernels and initiating MPI communication and is not usually used in way in which the memory system is heavily taxed.

For the OSU Micro-Benchmarks, on non-accelerator systems, we consider both on-socket and on-node communication as described in subsection 3.1. On accelerator systems,

Rank/Name	Location	CPU
29. Trinity	LANL	Intel Xeon Phi 7250
94. Theta	ANL	Intel Xeon Phi 7230
109. Sawtooth	INL	Intel Xeon Platinum 8268
127. Eagle	NREL	Intel Xeon Gold 6154
141. Manzano	SNL	Intel Xeon Platinum 8268

Table 2: US DOE non-accelerator based supercomputers in the top 150 of the June 2023 Top500. All data on this table is taken from the Top 500 [17].

we consider host-to-host and device-to-device transfers, the latter being broken down into several categories based on the device-to-device interconnects, which are described in more detail in Appendix A. On most of the accelerator systems, there is some amount of interconnect heterogeneity between accelerators. Perlmutter and Polaris have the same interconnect between all four GPUs. Sierra, Summit and Lassen have two different classes of GPU interconnection — one via NVLink and one via PCIe. These are separated as classes "A" and "B" in the following chart. Frontier, RZVernal and Tioga have different GPUs that are connected via four, two or one infinity fabric links (denoted "A," "B," and "C") in the following charts, as well as GPUs that do not have a direct connection (denoted "D").

For Comm|Scope on accelerator systems, we estimate kernel launch latency, empty queue wait latency, and data copy costs. Kernel launch latency is measured by recording the wall time that it takes to launch (not complete) empty, zero-argument kernels. Empty queue wait latency measures the wall time taken to complete a device synchronize call with an empty work queue. Data copy cost measurements invoke and complete an asynchronous memcopy between the source and target devices. If the source is the host, the source buffer is pinned. For data copies, we average the device-to-host and host-do-device latencies and bandwidths and report those together. Latency is measured using 128B transfers. Bandwidth is measured using 1GB transfers. For device-to-device latencies, we differentiate based on the type of interconnect between the devices, as described in more detail in Appendix A. Comm|Scope is built on the `benchmark` [10] support library, which is responsible for determining how many operations to average for each test. Like the other benchmarks, 100 such tests are aggregated to produce the reported mean and standard deviation.

We begin with the non-accelerator based platforms, which consist of five different Intel Xeon and Intel Xeon Phi based machines. Results on BabelStream and the OSU microbenchmarks can be found in Table 4. The three traditional Xeon CPU systems (Sawtooth, Eagle and Manzano) all have somewhat similar memory bandwidth for both a single core (13-16

Rank/Name	Location	CPU	Accelerator
1. Frontier	ORNL	AMD EPYC	AMD MI250X
5. Summit	ORNL	IBM Power9	NVIDIA GV100
6. Sierra	LLNL	IBM Power9	NVIDIA GV100
8. Perlmutter[1]	NERSC	AMD EPYC 7763	NVIDIA A100
19. Polaris	ANL	AMD EPYC 7532	NVIDIA A100
36. Lassen	LLNL	IBM Power9	NVIDIA V100
116. RZVernal	LLNL	AMD EPYC	AMD MI250X
132. Tioga	LLNL	AMD EPYC	AMD MI250X

Table 3: US DOE accelerator based supercomputers in the top 150 of the June 2023 Top500, excepting Cori and Frontier TDS. All data on this table is taken from the Top 500 [17]. [1] A100s with 40GB HBM used.

GB/s) and all cores (200-250 GB/s) as well as sub-microsecond MPI latencies both on-socket and on-node. For the Xeon Phi systems, we see a substantial performance disparity between Trinity and Theta, especially in the realm of MPI latency. At the suggestion of Argonne staff, we tried the ALCF MPI Benchmarks [8], as an alternative to the OSU microbenchmarks, and they reported a slightly lower MPI latency (sub-5 μs), but nowhere near as small as Trinity.

To the best of our knowledge, no precise theoretical memory bandwidth numbers for Knights Landing's MCDRAM have been published, though Intel claims > 450 GB/s [34]; the "Peak" bandwidth numbers for Trinity and Theta reflect this. Trinity's and Theta's Knights Landing CPUs were configured in "quad cache" mode, where the MCDRAM acts as a system-managed cache for the DDR4 main memory. Overheads of managing the cache may contribute to lower "all" bandwidth on Trinity, but do not adequately explain the suspiciously low measurement on Theta, which underperforms the rest of the platforms substantially.

Now we consider the accelerator-based platforms. Here we report BabelStream and OSU microbenchmark results in Table 5 and Comm|Scope results in Table 6. We note that the three NVIDIA V100 machines (Summit, Sierra and Lassen) have a substantially lower device memory bandwidth than the NVIDIA A100 machines (Perlmutter and Polaris) and the AMD MI250X machines (Frontier, RZVernal and Tioga). The latter two categories report fairly similar achieved memory bandwidth (about 1.3 TB/s). 1536 Perlmutter nodes have A100s with 40GB HBM memory, and 256 nodes have A100s with 80GB - in this work, we only measure the 40 GB A100s as those make up the majority of nodes in the machine.

Again, recall that BabelStream only uses one of the two GCDs on the AMD MI250X GPU, so the overall bandwidth of the GPU would be roughly double what is reported if another GPU stream were copying data at the same time. Host MPI latencies are sub-microsecond on all accelerator machines, which is consistent with results on the non-accelerator architectures. Device MPI latencies show a substantial difference between the NVIDIA V100 machines (roughly 18-19 μs), the NVIDIA A100 machines (10-14 μs) and the sub-microsecond latencies we see on the AMD MI250X machines. We also note that all GPUs appear to be roughly equidistant on the MI250X machines, while the NVIDIA V100 platforms add roughly $1\mu s$ for the non-NVLink connections.

Kernel launch latencies exhibit a clear hierarchy of 4-5 μs for the V100 machines and 1.5-2.15 μs for the A100 and MI250X machines, with the MI250X machines falling on the high (RZVernal/Tioga) and low (Frontier) ends of that range. Kernel wait latencies are 5-6 μs for the V100 machines, roughly 1 μs for the A100 machines and $.1 - .2\mu s$ for the MI250X machines. Host-to-device and device-to-host latencies show a different trend, with the MI250X machines measured at 12-13 μs, the V100 machines next at 7-8 μs, and the A100 machines fastest at 4-6 μs. For host-to-device and device-to-host bandwidth, the V100 machines perform best, reaching 40-60 GB/s due to NVLink interconnects between the CPU and accelerators, while all other machines reach roughly 25 GB/s over PCIe interconnects. Device to device transfer latency is roughly 25 μs via the NVLink connections on the V100 and about 2 μs slower on the non-NVLink connections. The two A100 machines (Perlmutter and Polaris) show a substantial difference (14 μs vs. 32 μs) in their device-to-device latency performance, with a small variation based on which GPU pair is tested. These systems have the same GPU SKU, the same number of GPUs per node, and the same GPU-GPU interconnects, so it is possible that the difference is explained by system software differences (e.g., CUDA driver version). The MI250X platforms exhibit a 10-12 μs latency, with the quad infinity connections on RZVernal and Tioga running a full 4 μs faster than the similar pairs on Frontier. Inter-device latency in Comm|Scope is *substantially* slower than the inter-device latency shown by the OSU microbenchmarks. This is likely due to the former's use of hipMemcpyAsync as a means of copying data, while the MPI implementation likely uses remote memory access (RMA).

For accelerator platforms, we can summarize the results of Table 5 and Table 6 by providing ranges for all of the mean values reported in the tables. These results can be found in Table 7. This table provides an easier means of digesting the above results when one is primarily interested in comparing how different accelerators perform, rather the comparing different systems.

5 CONCLUSIONS AND FUTURE WORK

Despite extensive benchmarking of Top-500 class systems, results are difficult to access, leading the HPC community to have a fragmented and patchwork understanding of key

Rank/Name	Memory Bandwidth (GB/s)			MPI Latency (μs)	
	Single	All	Peak	On-Socket	On-Node
29. Trinity	12.36 ± 0.16	347.28 ± 5.76	> 450 [34]	0.67 ± 0.01	0.99 ± 0.01
94. Theta	18.76 ± 0.58	119.72 ± 0.54	> 450 [34]	5.95 ± 0.01	6.25 ± 0.05
109. Sawtooth	13.06 ± 0.35	238.70 ± 8.39	281.50 [13]	0.48 ± 0.01	0.48 ± 0.01
127. Eagle	13.45 ± 0.03	208.24 ± 0.92	255.97 [12]	0.17 ± 0.00	0.38 ± 0.01
141. Manzano	15.27 ± 0.05	234.86 ± 0.12	281.50 [13]	0.32 ± 0.00	0.56 ± 0.01

Table 4: Mean and standard deviation of observed memory bandwidth (GB/s) and MPI latency (μs) for US DOE non-accelerator supercomputers taken over 100 runs. Peak bandwidth refers to the theoretical maximum achievable. On Xeon Phi systems, socket represents transfers between the first and second codes with node representing transfers between the first and last cores.

Rank/Name	Memory Bandwidth (GB/s)		MPI Latency (μs)	MPI Latency (μs) Device-to-Device			
	Device	Peak	Host-to-Host	A	B	C	D
1. Frontier	1336.35 ± 1.11	1600 [4]	0.45 ± 0.01	0.44 ± 0.00	0.44 ± 0.00	0.44 ± 0.00	0.44 ± 0.00
5. Summit	786.43 ± 0.11	900 [1]	0.34 ± 0.07	18.10 ± 0.22	19.30 ± 0.15		
6. Sierra	861.40 ± 0.65	900 [1]	0.38 ± 0.01	18.72 ± 0.12	19.76 ± 0.37		
8. Perlmutter	1363.74 ± 0.23	1555.2 [3]	0.46 ± 0.06	13.50 ± 0.13			
19. Polaris	1362.75 ± 0.17	1555.2 [3]	0.21 ± 0.00	10.42 ± 0.03			
36. Lassen	861.03 ± 0.53	900 [1]	0.37 ± 0.00	18.68 ± 0.20	19.72 ± 0.13		
116. RZVernal	1291.38 ± 0.77	1600 [4]	0.49 ± 0.00	0.50 ± 0.01	0.50 ± 0.01	0.50 ± 0.00	0.49 ± 0.01
132. Tioga	1336.81 ± 0.97	1600 [4]	0.49 ± 0.00	0.50 ± 0.00	0.50 ± 0.00	0.50 ± 0.00	0.49 ± 0.01

Table 5: Mean and standard deviation of observed memory bandwidth (GB/s) and MPI latency (μs) for US DOE accelerator supercomputers taken over 100 runs. Peak refers to the theoretical maximum bandwidth. For Summit, Sierra, and Lassen, A refers to GPUs directly connected by NVLinks, and B otherwise. For Frontier, RZVernal, and Tioga, A, B, and C refer to quad-, dual-, and single infinity fabric links, while D refers to a GPU without a direct connection.

Rank/Name	Kernel		(H→D + D→H)/2		D→D Latency			
	Launch	Wait	Latency	Bandwidth	A	B	C	D
1. Frontier	1.51 ± 0.00	0.14 ± 0.00	12.91 ± 0.02	24.87 ± 0.01	12.02 ± 0.05	12.56 ± 0.03	12.68 ± 0.02	12.02 ± 0.10
5. Summit	4.84 ± 0.01	4.31 ± 0.01	7.82 ± 0.07	44.88 ± 0.00	24.97 ± 0.16	27.44 ± 0.14		
6. Sierra	4.13 ± 0.01	5.59 ± 0.02	7.27 ± 0.23	63.40 ± 0.01	23.91 ± 0.16	27.70 ± 0.12		
8. Perlmutter	1.77 ± 0.01	0.98 ± 0.00	4.24 ± 0.01	24.74 ± 0.00	14.74 ± 0.41			
19. Polaris	1.83 ± 0.00	1.32 ± 0.01	5.33 ± 0.02	23.71 ± 0.00	32.84 ± 0.30			
36. Lassen	4.56 ± 0.00	5.52 ± 0.01	7.76 ± 0.32	63.34 ± 0.02	24.56 ± 0.28	27.69 ± 0.10		
116. RZVernal	2.16 ± 0.01	0.12 ± 0.00	12.20 ± 0.07	24.88 ± 0.00	9.85 ± 0.01	12.58 ± 0.00	12.45 ± 0.02	10.21 ± 0.01
132. Tioga	2.15 ± 0.01	0.12 ± 0.00	12.19 ± 0.04	24.88 ± 0.00	9.85 ± 0.02	12.59 ± 0.01	12.46 ± 0.01	10.12 ± 0.02

Table 6: Mean and standard deviation of observed kernel launch / wait latencies (μs) as well has memory transfer latencies (μs) and bandwidths (GB/s) for US DOE accelerator supercomputers taken over 100 runs. For Summit, Sierra, and Lassen, A refers to GPUs directly connected by NVLinks, and B otherwise. For Frontier, RZVernal, and Tioga, A, B, and C refer to quad-, dual-, and single infinity fabric links, while D refers to a GPU without a direct connection.

Accelerator	Memory BW	MPI Lat.	Kernel Launch	Kernel Wait	H2D/D2H Lat.	H2D/D2H BW	D2D Lat.
V100	786.43–861.40	18.10–18.72	4.13–4.84	4.31–5.59	7.27–7.82	44.88–63.40	23.91–24.97
A100	1362.75–1363.74	10.42–13.50	1.77–1.83	0.98–1.32	4.24–5.33	23.71–24.74	14.74–32.84
MI250X	1291.38–1336.81	0.44–0.50	1.51–2.16	0.12–0.14	12.19–12.91	24.87–24.88	9.85–12.02

Table 7: Maximum and minimum of device bandwidth (GB/s) device MPI latency (μs) kernel launch / wait latencies (μs) as well has memory transfer latencies (μs) and bandwidths (GB/s) across US DOE accelerator supercomputers.

performance parameters across a variety of systems. In an effort to correct this issue, this paper presents intra-node latency and bandwidth measurements for all fourteen active U.S. Department of Energy systems above rank 150 in the June 2023 Top 500 list. This paper is intended to be a first reference for developers of performance-portable application codes when they need information measured by these benchmarks.

Four future areas of investigation are planned. First, we plan to extend this work to include inter-node measurements. The challenge is to develop a practical set of benchmarks that provide actionable information regarding network contention, node-vs-network capability (e.g. injection bandwidth), network topology, MPI implementation, collective communication, and GPU-network integration without becoming unwieldy. Second, the results in this paper will quickly go out-of-date as new systems are built and old ones are retired. We hope to refine our methodology and publish updated benchmarks approximately once per year. Third, the US DOE has a specific set of mission criteria that drive its HPC procurements. Consequently, its systems may not represent other interesting design points in the Top 500 list. For instance, we did not report results from any AMD or Arm CPU systems, because the US DOE does not have any within the Top 150. Comparing results between Intel, AMD and Arm CPU systems would be of interest in the future. make We encourage anyone with an interest and access to a substantially different system to contact us for collaboration in a future publication of this nature. Fourth, prior work has identified substantial latency differences on the same systems between MPI implementations [26]. On systems where users are empowered to change MPI implementations, it may be worth measuring under a variety of configurations.

ACKNOWLEDGMENTS

SAND2023-09836C. Sandia National Laboratories is a multimission laboratory managed and operated by National Technology and Engineering Solutions of Sandia, LLC., a wholly owned subsidiary of Honeywell International, Inc., for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA-0003525. This written work is authored by an employee of NTESS. The employee, not NTESS, owns the right, title and interest in and to the written work and is responsible for its contents. Any subjective views or opinions that might be expressed in the written work do not necessarily represent the views of the U.S. Government. This manuscript has been authored by UT-Battelle, LLC, under contract DE-AC05-00OR22725 with the U.S. Department of Energy. The publisher acknowledges that the U.S. Government retains a non-exclusive, paid-up, irrevocable, world-wide license to publish or reproduce the published form of this written work or allow others to do so, for U.S. Government purposes. The DOE will provide public access to results of federally sponsored research in accordance with the DOE Public Access Plan.

This research used resources of the Oak Ridge Leadership Computing Facility at the Oak Ridge National Laboratory, which is supported by the Office of Science of the U.S. Department of Energy under Contract No. DE-AC05-00OR22725. This research used resources of the National Energy Research Scientific Computing Center, which is supported by the Office of Science of the U.S. Department of Energy under Contract No. DE-AC02-05CH11231. This research made use of Idaho National Laboratory computing resources which are supported by the Office of Nuclear Energy of the U.S. Department of Energy and the Nuclear Science User Facilities under Contract No. DE-AC07-05ID14517. This research used resources of the Argonne Leadership Computing Facility, a DOE Office of Science User Facility supported under Contract No. DE-AC02-06CH11357. This research used resources of the Los Alamos National Laboratory, supported by the US Department of Energy under contract No 89233218CNA000001. The authors would also like to thank Christopher Knight of Argonne National Laboratory and James Elliott of Sandia National Laboratories for consulting on configuration and run options on various machines.

REFERENCES

[1] 2017. *NVIDIA Tesla V100 GPU Architecture.* https://images.nvidia.com/content/volta-architecture/pdf/volta-architecture-whitepaper.pdf

[2] 2018. *HPL - A Portable Implementation of the High-Performance Linpack Benchmark for Distributed-Memory Computers.* https://www.netlib.org/benchmark/hpl

[3] 2020. *NVIDIA A100 Tensor Core GPU Architecture.* https://images.nvidia.com/aem-dam/en-zz/Solutions/data-center/nvidia-

ampere-architecture-whitepaper.pdf
[4] 2021. *Introducing AMD CDNA 2 Architecture.* https://www.amd.com/system/files/documents/amd-cdna2-white-paper.pdf
[5] 2021. *mpi-benchmarks.* https://github.com/intel/mpi-benchmarks
[6] 2022. *BenchPress.* https://github.com/bienz2/BenchPress
[7] 2022. *HPCG Benchmark.* https://www.hpcg-benchmark.org/
[8] 2023. *alcf-mpi-benchmarks.* https://github.com/argonne-lcf/alcf-mpi-benchmarks
[9] 2023. *AMD Instinct MI250X Accelerator.* https://www.amd.com/en/products/server-accelerators/instinct-mi250x
[10] 2023. benchmark. http://github.com/google/benchmark
[11] 2023. *Frontier User Guide.* https://docs.olcf.ornl.gov/systems/frontier_user_guide.html
[12] 2023. *Intel Xeon Gold 6154 Processor.* https://ark.intel.com/content/www/us/en/ark/products/120495/intel-xeon-gold-6154-processor-24-75m-cache-3-00-ghz.html
[13] 2023. *Intel Xeon Platinum 8268 Processor.* https://ark.intel.com/content/www/us/en/ark/products/192481/intel-xeon-platinum-8268-processor-35-75m-cache-2-90-ghz.html
[14] 2023. *OSU Micro-benchmarks.* http://mvapich.cse.ohio-state.edu/benchmarks/
[15] 2023. *Perlmutter Architecture.* https://docs.nersc.gov/systems/perlmutter/architecture/
[16] 2023. *Summit User Guide.* https://docs.olcf.ornl.gov/systems/summit_user_guide.html
[17] 2023. *TOP500 June 2023.* https://www.top500.org/lists/top500/2023/06/
[18] David H Bailey, Eric Barszcz, John T Barton, David S Browning, Robert L Carter, Leonardo Dagum, Rod A Fatoohi, Paul O Frederickson, Thomas A Lasinski, Rob S Schreiber, et al. 1991. The NAS Parallel Benchmarks—Summary and Preliminary Results. In *Proceedings of the 1991 ACM/IEEE Conference on Supercomputing*. 158–165.
[19] Christian Bell, Dan Bonachea, Yannick Cote, Jason Duell, Paul Hargrove, Parry Husbands, Costin Iancu, Michael Welcome, and Katherine Yelick. 2003. An Evaluation of Current High-Performance Networks. In *Proceedings International Parallel and Distributed Processing Symposium*. IEEE.
[20] Abhinav Bhatele, Kathryn Mohror, Steven H. Langer, and Katherine E. Isaacs. 2013. There goes the Neighborhood: Performance Degradation due to Nearby Jobs. In *SC '13: Proceedings of the International Conference on High Performance Computing, Networking, Storage and Analysis*. 1–12. https://doi.org/10.1145/2503210.2503247
[21] Devendar Bureddy, Hao Wang, Akshay Venkatesh, Sreeram Potluri, and Dhabaleswar K Panda. 2012. OMB-GPU: A Micro-Benchmark Suite for Evaluating MPI Libraries on GPU Clusters. In *Recent Advances in the Message Passing Interface: 19th European MPI Users' Group Meeting, EuroMPI 2012, Vienna, Austria, September 23-26, 2012. Proceedings 19*. Springer, 110–120.
[22] Paul Stewart Crozier, Heidi K Thornquist, Robert W Numrich, Alan B Williams, Harold Carter Edwards, Eric Richard Keiter, Mahesh Rajan, James M Willenbring, Douglas W Doerfler, and Michael Allen Heroux. 2009. *Improving Performance via Mini-Applications.* Technical Report SAND2009-5574. Sandia National Laboratories. https://www.osti.gov/biblo/993908.
[23] Tom Deakin, James Price, Matt Martineau, and Simon McIntosh-Smith. 2018. Evaluating Attainable Memory Bandwidth of Parallel Programming Models via BabelStream. *International Journal of Computational Science and Engineering* 17, 3 (2018), 247–262. https://doi.org/10.1504/IJCSE.2018.095847
[24] Brice Goglin, Emmanuel Jeannot, Farouk Mansouri, and Guillaume Mercier. 2018. Hardware Topology Management in MPI Applications through Hierarchical Communicators. *Parallel Comput.* 76 (2018), 70–90. https://doi.org/10.1016/j.parco.2018.05.006
[25] Feng Ji, Ashwin M Aji, James Dinan, Darius Buntinas, Pavan Balaji, Wu-chun Feng, and Xiaosong Ma. 2012. Efficient Intranode Communication in GPU-accelerated Systems. In *2012 IEEE 26th International Parallel and Distributed Processing Symposium Workshops*. IEEE, 1838–1847.
[26] Kawthar Shafie Khorassani, Ching-Hsiang Chu, Hari Subramoni, and Dhabaleswar K Panda. 2019. Performance Evaluation of MPI Libraries on GPU-enabled OpenPOWER Architectures: Early Experiences. In *High Performance Computing: ISC High Performance 2019 International Workshops, Frankfurt, Germany, June 16-20, 2019, Revised Selected Papers 34*. Springer, 361–378.
[27] Ang Li, Shuaiwen Leon Song, Jieyang Chen, Jiajia Li, Xu Liu, Nathan R Tallent, and Kevin J Barker. 2019. Evaluating Modern GPU Interconnect: PCIe, NVLink, NV-SLI, NVSwitch and GPUDirect. *IEEE Transactions on Parallel and Distributed Systems* 31, 1 (2019), 94–110.
[28] Ang Li, Shuaiwen Leon Song, Jieyang Chen, Xu Liu, Nathan Tallent, and Kevin Barker. 2018. Tartan: Evaluating Modern GPU Interconnect via a Multi-GPU Benchmark Suite. In *2018 IEEE International Symposium on Workload Characterization (IISWC)*. IEEE, 191–202.
[29] Jiuxing Liu, Balasubramanian Chandrasekaran, Weikuan Yu, Jiesheng Wu, Darius Buntinas, Sushmitha Kini, Dhabaleswar K Panda, and Pete Wyckoff. 2004. Microbenchmark Performance Comparison of High-Speed Cluster Interconnects. *IEEE Micro* 24, 1 (2004), 42–51.
[30] John D McCalpin. 1995. Sustainable Memory Bandwidth in Current High Performance Computers. (1995). https://www.cs.virginia.edu/~mccalpin/papers/bandwidth/bandwidth.html
[31] John D McCalpin et al. 1995. Memory Bandwidth and Machine Balance in Current High Performance Computers. *IEEE Computer Society Technical Committee on Computer Architecture (TCCA) Newsletter* 2, 19-25 (1995).
[32] Carl Pearson, Abdul Dakkak, Sarah Hashash, Cheng Li, I-Hsin Chung, Jinjun Xiong, and Wen-Mei Hwu. 2019. Evaluating characteristics of CUDA communication primitives on high-bandwidth interconnects. In *Proceedings of the 2019 ACM/SPEC International Conference on Performance Engineering*. 209–218.
[33] Mahesh Rajan, Doug Doerfler, and Simon Hammond. 2015. *Trinity Benchmarks on Intel Xeon Phi (Knights Corner).* Technical Report SAND2015-0454C. Sandia National Laboratories. https://www.osti.gov/biblo/1504115.
[34] Avinash Sodani, Roger Gramunt, Jesus Corbal, Ho-Seop Kim, Krishna Vinod, Sundaram Chinthamani, Steven Hutsell, Rajat Agarwal, and Yen-Chen Liu. 2016. Knights landing: Second-generation intel xeon phi product. *Ieee micro* 36, 2 (2016), 34–46.
[35] N Wichmann, C Nuss, P Carrier, R Olson, S Anderson, M Davis, R Baker, E Draeger, S Domino, A Agelastos, and M Rajan. 2015. *Performance on Trinity (a Cray XC40) with Acceptance-Applications and Benchmarks.* Technical Report SAND2016-3635C. Sandia National Laboratories. https://www.osti.gov/biblio/1365199.

A MACHINE INFORMATION

We include compiler, device library and MPI versions used on the various platforms in Table 8 and Table 9. We attempted to use the default environment on all systems, if there was one available (some systems do not load compilers by default) and if it worked for all three benchmark codes.

Special consideration was needed on certain systems, which is detailed below:

Rank/Name	Compiler	MPI
29. Trinity	intel/2022.0.2	cray-mpich/7.7.20
94. Theta	intel/19.1.0.166	cray-mpich/7.7.14
109. Sawtooth	intel/19.0.5	intel-mpi/2019.0.117
127. Eagle	gcc/8.4.0	openmpi/4.1.0
141. Manzano	intel/16.0	openmpi/1.10

Table 8: Compilers and MPI versions on non-accelerator US DOE machines.

- **Sierra**: A patch to BabelStream was needed, which removed the -march=native and -forward-unknown-to-host-compiler compiler options which are not supported by the compilers on Sierra.
- **Lassen**: A patch to BabelStream was needed, which removed the -march=native and -forward-unknown-to-host-compiler compiler options which are not supported by the compilers on Lassen.
- **Theta**: We could not get the libnuma support in Comm|Scope to compile, so we built Comm|Scope without it.

B ARTIFACT DESCRIPTION

B.1 Artifact Identification

The contributions of the paper are results of latency and bandwidth benchmarks for DOE systems in the upper tier of the Top 500 List as of June 2023. The three benchmark software packages and their provenance are given below:

(1) BabelStream (memory bandwidth) by University of Bristol, obtained from https://github.com/UoB-HPC/BabelStream
(2) OSU Microbenchmarks (MPI latency) by Ohio State University, obtained from http://mvapich.cse.ohio-state.edu/benchmarks/ (Not made available by OSU as a public git code repository / Downloaded as tarball)
(3) Comm|Scope (GPU kernel and data transfer latency) by IBM-Illinois Center for Cognitive Computing Systems Research (C3SR), obtained from https://github.com/c3sr/comm_scope

Because these benchmarks are publicly available, we expect that the results of the paper can be reasonably reproduced for similar systems with similar software environment configurations.

B.2 Reproducibility of Experiments

To build BabelStream, we first `cmake` then `make`. We execute the benchmark suite, sweeping the input size space from 16k to somewhere between 16M and 128M double precision values, stepping by powers of two. On systems with GPUs, we measure the GPU memory bandwidth. For 100 trials of each test configuration, about one hour is required to complete such GPU bandwidth measurements, depending on the machine. On CPU only systems, we measure the CPU memory bandwidth, varying the OpenMP parameters as specified in Table 1. For 100 trials of each test configuration, several hours are required to complete such CPU bandwidth measurements, depending on the machine. BabelStream results for the highest performing benchmark within the suite in each instance are reported under "Memory Bandwidth" in Tables 4, 5, and 7.

To build the OSU Microbenchmarks, we first `configure` then `make`. We execute the point-to-point MPI latency test. On systems with GPUs, we measure latency between pairs of GPUs and between pairs of CPUs. On systems with more than one CPU socket, we conduct one set of experiments between two processes on the same socket and one set between two processes on different sockets. On systems with a single KNL, we conduct experiments between two processes on the first two cores and also between the first and last cores. For 100 trials of each test configuration, 1-2 hours are required, depending on the machine. OSU point-to-point latency results are reported as "MPI Latency" in Tables 4, 5, and 7.

To build Comm|Scope: we first `cmake` then `make`. On systems with NVIDIA GPUs, we execute the Comm_cudaMemcpyAsync_GPUToGPU, Comm_cudaMemcpyAsync_PinnedToGPU, Comm_cudaMemcpyAsync_GPUToPinned, Comm_cudaDeviceSynchronize, and Comm_cudart_kernel tests. On systems with AMD GPUs, we execute the Comm_hipMemcpyAsync_GPUToGPU, Comm_hipMemcpyAsync_PinnedToGPU, Comm_hipMemcpyAsync_GPUToPinned, Comm_hipDeviceSynchronize, and Comm_hip_kernel tests. On CPU only systems, Comm|Scope is not used. For 100 trials of each test configuration, 1-2 hours are required, depending on the machine. Comm|Scope results are reported in Table 6 and the last five columns of Table 7.

Received 09 August 2023

Rank/Name	Compiler	Device Library	MPI
1. Frontier	amd-mixed/5.3.0	amd-mixed/5.3.0	cray-mpich/8.1.23
5. Summit	xl/16.1.1-10	cuda/11.0.3	spectrum-mpi/10.4.0.3-20210112
6. Sierra	gcc/8.3.1	cuda/10.1.243	spectrum-mpi/rolling-release
8. Perlmutter	gcc/11.2.0	cuda/11.7	cray-mpich/8.1.25
19. Polaris	nvhpc/21.9	cuda/11.4	cray-mpich/8.1.16
36. Lassen	gcc/7.3.1	cuda/10.1.243	spectrum-mpi/rolling-release
116. RZvernal	amd/5.6.0	amd/5.6.0	cray-mpich/8.1.26
132. Tioga	amd/5.6.0	amd/5.6.0	cray-mpich/8.1.26

Table 9: Compilers, device libraries and MPI versions on accelerator US DOE machines.