GASNet-EX RMA Communication Performance on Recent Supercomputing Systems

Paul H. Hargrove, Dan Bonachea
Computer Languages and Systems Software Group
Lawrence Berkeley National Laboratory
{PHHargrove, DOBonachea}@lbl.gov

Abstract—Partitioned Global Address Space (PGAS) programming models, typified by systems such as Unified Parallel C (UPC) and Fortran coarrays, expose one-sided Remote Memory Access (RMA) communication as a key building block for High Performance Computing (HPC) applications. Architectural trends in supercomputing make such programming models increasingly attractive, and newer, more sophisticated models such as UPC++, Legion and Chapel that rely upon similar communication paradigms are gaining popularity.

GASNet-EX is a portable, open-source, high-performance communication library designed to efficiently support the networking requirements of PGAS runtime systems and other alternative models in emerging exascale machines. The library is an evolution of the popular GASNet communication system, building upon 20 years of lessons learned. We present microbenchmark results which demonstrate the RMA performance of GASNet-EX is competitive with MPI implementations on four recent, high-impact, production HPC systems. These results are an update relative to previously published results on older systems. The networks measured here are representative of hardware currently used in six of the top ten fastest supercomputers in the world, and all of the exascale systems on the U.S. DOE road map.

Index Terms—HPC, PGAS, RMA, Active Messages, Exascale Computing, Middleware

I. BACKGROUND

GASNet-EX is a language-independent, networking middleware layer that provides network-independent, high-performance communication primitives for High-Performance Computing (HPC). Unlike the dominant MPI communication standard, the GASNet-EX interface and implementation are designed specifically to meet the needs of alternative programming models on emerging exascale systems. GASNet-EX is implemented directly over the native/proprietary APIs of many networks, including all of those in use at the HPC centers of the U. S. Department of Energy’s Office of Science [1]. GASNet-EX’s interface is primarily intended as a compilation target and for use by runtime library writers (as opposed to domain scientists), and the primary goals are high performance, interface portability, and expressiveness. GASNet-EX provides communication services for many projects, including both programming models and other parallel libraries and frameworks. Examples of alternative HPC programming models using GASNet-EX include: UPC++ [2–4], the Legion programming system [5], HPE’s Chapel language [6], the Omni Xscalable Compiler [7], and many UPC [8–10] and CAF/Fortran [11–14] compiler runtimes. GASNet-EX has also been adopted for communication services by a number of parallel libraries and frameworks, including [15, 16]. See [17] for full details on current client software, and Fig. 1 for an overview of the GASNet-EX software ecosystem.

GASNet-EX offers a variety of communication services to runtime clients, notably including: Remote Memory Access (RMA), Active Messages (AM), remote atomic memory operations and non-blocking collectives; see [18] for further details on GASNet-EX features. This paper focuses on the RMA performance of GASNet-EX, which is critical to the efficiency of many client applications. Specifically, we repeat the performance evaluation presented in our earlier paper [18] on more recent production HPC platforms that feature correspondingly newer network hardware and software stacks. As such, some descriptive portions of that paper are reproduced here with permission.

II. EXPERIMENTAL SYSTEMS

For our benchmarking efforts, we selected four high-impact production HPC systems at U.S. Department of Energy (DOE) computing centers. The networks in these systems are representative of hardware currently used in six of the top ten fastest supercomputers in the world, according to the June 2022 Top500 [19] list (current at the time of writing). Notably, the number one spot in that list is held by Frontier [20] with the same Slingshot-11 network that we evaluate on Perlmutter (see below), which will also appear in DOE’s other two announced exascale systems: Aurora [21] and El Capitan [22].

Our measurements attempt to reproduce the experience of a non-expert end-user. Therefore, all systems were used as configured by the respective HPC centers, using default

![Fig. 1. GASNet-EX software ecosystem](image-url)
versions of installed environment modules for all software except for GASNet-EX and the microbenchmark codes.

**Summit**: The “Summit” [23] system at OLCF [24] consists of IBM AC922 nodes, each with two 22-core POWER9 CPUs and connected to a 100Gb/s EDR InfiniBand network by two Mellanox “ConnectX-5” HCAs, each with affinity to a single socket.


**Perlmutter SS-10 / SS-11**: The last two systems evaluated are partitions of the HPE Cray EX system at NERSC [25] known as “Perlmutter” [29]. Nodes in both partitions contain a single 64-core AMD EPYC 7763 “Milan” CPU and are connected to a 200Gb/s HPE Slingshot network, but they differ in the NICs which attach them to the network. The Slingshot-10 (“SS-10”) nodes each hold two Mellanox “ConnectX-5” NICs operating at 100Gb/s, while the Slingshot-11 (“SS-11”) nodes each hold four HPE-proprietary “Cassini” NICs operating at 200Gb/s.

As with all HPC-relevant network hardware, the systems evaluated provide Remote Direct Memory Access (RDMA) communication support; this allows GASNet-EX to offload RMA communication work to the network hardware on both sides, while minimizing software overheads associated with payload copying or CPU-driven communication. GASNet-EX’s lightweight RMA operations are deliberately designed to streamline this mapping, avoiding semantics encumbrances that could incur impedance mismatches or other unwanted overheads.

When building software (including GASNet-EX and all microbenchmarks) we followed the instructions without the application of any expert knowledge. No configuration settings, environment variables, or similar means were used to tune the benchmark performance of GASNet-EX or MPI. Each of the HPC centers provide multiple compiler family options in their production software environment. We selected each system’s default version of the GNU compilers, even where a different compiler family was the default, because GNU is universally available and enjoys widespread compatibility with library clients.

We benchmarked GASNet-EX version 2022.3.0 using two tests selected from those provided with the source code distribution. For MPI benchmarking we measured the vendor-supplied default MPI implementation on each system, using the publicly available Intel MPI Benchmarks (IMB) [30] version v2021.3 (the latest official benchmark release at the time of writing). For detailed experimental methodology, see the Artifact Description (AD) Appendix.

---

1Benchmarks on Perlmutter utilize only a single NIC per process, since neither GASNet-EX nor HPE Cray MPI currently support more.

2with one exception: On Summit we set environment variables to restrict both GASNet-EX and the MPI implementation to a single rail per process on the dual-rail network, to ensure a meaningful comparison. We recommend this configuration because it can yield significant latency improvements.

---

### III. RMA Flood Bandwidth Benchmark Results

A “flood bandwidth” benchmark measures achievable bandwidth at a given transfer size by initiating a large number of non-blocking transfers and waiting for them all to fully complete. The reported metric is the total volume of data transferred, divided by the total elapsed time. We report unidirectional (one initiator to one target) flood bandwidths, where the passive target waits in an appropriate synchronization operation.

For GASNet-EX we used the testlarge microbenchmark to measure performance of the gex_RMA_PutNBI and gex_RMA_GetNBI functions, synchronized with a final gex_NBI_Wait. We measured flood bandwidth of the MPI_Put and MPI_Get functions using the “Aggregate” timings from, respectively, the Unidir_put and Unidir_get tests from the IMB-RMA suite (these tests measure the time to issue RMA and synchronize using MPI_Win_flush, within a passive-target access epoch established by a MPI_Win_lock(SHARED) call outside the timed region – see [30] for further details). The testlarge benchmark reports bandwidths in units of “MiB/s” ($10^{20}$ bytes per second), whereas the IMB tests use “MB/s” ($10^{6}$ bytes per second). Both have been converted to “GiB/s” ($10^{9}$ bytes per second) for the plots which follow. Although RMA and message passing are semantically different, for comparison purposes the plots also report uni-directional bandwidth of MPI_Isend/MPI_Irecv, from the “Aggregate” timings of the Uniband test from the IMB-MPII suite.

All tests ran between two compute nodes, using a single process and single NIC on each. Data was collected from 16 distinct batch jobs, each running one instance of each GASNet-EX and MPI test back-to-back. Each data point plotted reports the maximum achieved bandwidth for that benchmark and transfer size. For RMA and message passing tests we used 10,000 and 500 iterations, respectively.

In Fig. 2, “×” markers denote GASNet-EX RMA, “○” markers denote MPI RMA, and “+” markers denote MPI message passing. RMA Put results are distinguished by the use of solid lines (in shades of blue), while RMA Get results use dot-dashed lines (in shades of red). Dashed lines (in green) are message-passing results. The horizontal axis (transfer size) is logarithmic, while the vertical axis (bandwidth) is linear.

We find the uni-directional flood bandwidth of GASNet-EX RMA operations is uniformly comparable to or better than the corresponding MPI RMA operations. GASNet-EX results are seen to rise more rapidly to the maximum bandwidth, exceeding 90% of saturation bandwidth at transfer sizes as small as 4KiB to 8KiB, for both Put and Get across all four systems. While both GASNet-EX and MPI RMA operations eventually reach each system’s asymptotic saturation bandwidth, their approach to the maximum differs. On Summit and Cori Haswell, GASNet-EX reaches the maximum for significantly smaller transfers than for MPI; in other words, one can achieve the maximum bandwidth with a wider range of transfer sizes using GASNet-EX than using MPI. On
Fig. 2. Uni-directional flood bandwidth versus transfer size
Perlmutter SS-10, the RMA performance of the two libraries is nearly indistinguishable. In the case of Perlmutter’s higher-bandwidth SS-11 NICs, GASNet-EX Puts reach the maximum at about half the transfer size required for MPI. For RMA Gets on SS-11, MPI’s rise to the maximum bandwidth is anomalously slow relative to all other bandwidth data.

On all systems GASNet-EX RMA flood bandwidth performance is at least comparable to MPI message passing, with a significant advantage across a wide range of transfers on three of the four systems.

### V. Conclusion and Future Work

We presented updated microbenchmark results demonstrating the RMA performance of GASNet-EX is competitive with several vendor MPI implementations on modern production HPC systems whose networks are representative of emerging exascale systems. We found that GASNet-EX RMA bandwidth outperformed the equivalent MPI RMA operations by up to 2.7x for Puts and up to 3.1x for Gets at certain transfer sizes, reaching saturation bandwidth at up to 8x smaller transfer sizes. For small-transfer latency, GASNet-EX RMA outperformed MPI RMA by up to 2.38x.

The Slingshot results on Perlmutter utilize GASNet-EX’s most recent ofi-conduit backend, which is currently labeled as “experimental” because it still remains largely untuned. On HPE Slingshot both ofi-conduit and Cray MPI communicate with the NIC via the underlying OFI libfabric layer, which also continues to evolve in both stability and performance. As such, we expect results from both libraries on Slingshot (especially for the SS-11 Cassini NIC) will continue to improve as both software stacks continue to mature. Future work includes tuning the performance of ofi-conduit and specializing its implementation to more effectively expose hardware capabilities.

GASNet-EX has recently extended its RMA interfaces to enable RMA for memory located on accelerator devices, such as the GPUs made by NVIDIA and AMD, which have become very popular in HPC systems and will provide the majority of computational power on exascale platforms. Current and future work includes expanding this feature to support additional varieties of accelerators and leverage hardware offload capabilities on additional network fabrics.

### Acknowledgments

This research was supported by the Exascale Computing Project (17-SC-20-SC), a collaborative effort of the U.S. Department of Energy Office of Science and the National Nuclear Security Administration.

This research used resources of the National Energy Research Scientific Computing Center, a DOE Office of Science User Facility supported by the Office of Science of the U.S. Department of Energy under Contract No. DE-AC02-05CH11231.

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.

---

3Diagnosing performance anomalies in HPE’s implementation of MPI RMA is outside the scope of this paper, but the authors acknowledge the possibility that expert tuning of the sort we’ve expressly avoided might address this behavior.

4The anomalously poor outlier behavior of MPI RMA Get on Perlmutter SS-11 has been excluded from this summary.
REFERENCES


This appendix describes the methodology used for all experiments whose results are presented in this paper. This is in accordance with the SC22 Reproducibility Initiative [32].

The intent is to measure the peak communication performance of the GASNet-EX and MPI middleware libraries running natively on the network hardware of production supercomputing systems. As such, all experiments were run natively on compute-node hardware as standard user-mode processes, without any containerization or virtual-machine technology.

A. Software used to perform benchmarking

- GASNet v2022.3.0 source code:
  https://gasnet.lbl.gov/EX/GASNet-2022.3.0.tar.gz
  - No external data required to initialize
  - Benchmarks are in the `tests` directory, written in C with direct calls to GASNet-EX
- Intel MPI Benchmarks v2021.3 source code:
  - No external data required to initialize
  - Benchmarks are in the `src_cpp` directory, written in C++ with direct calls to MPI

B. Benchmark commands

The following command lines were used to launch the various benchmarks, where `[RUN]` is a placeholder for “`jsrun -p 2 -r 1`” on Summit, and for “`srun --cpu_bind=cores -n2 -N2`” on the other systems. In all cases, these commands ran within an exclusive batch allocation of two compute nodes. All tests ran between two compute nodes, using a single process and single NIC on each.

- Bandwidth tests:
  - [RUN] `testlarge -m -in 10000 4194304 B`
  - [RUN] `IMB-RMA -time 600 -iter_policy off -iter 10000 -msglog 4:22 Unidir_put`
  - [RUN] `IMB-RMA -time 600 -iter_policy off -iter 10000 -msglog 4:22 Unidir_get`
  - [RUN] `IMB-MPI1 -time 600 -iter_policy off -iter 500 -msglog 4:22 Uniband`

- Latency tests:
  - [RUN] `testsmall -m -in 1000000 4096 A`
  - [RUN] `IMB-RMA -time 600 -iter_policy off -iter 1000000 -msglog 2:12 Unidir_put`
  - [RUN] `IMB-RMA -time 600 -iter_policy off -iter 1000000 -msglog 2:12 Unidir_get`
  - [RUN] `IMB-MPI1 -time 600 -iter_policy off -iter 1000000 -msglog 2:12 PingPong`

Data was collected from 16 distinct batch jobs, each running one instance of each GASNet-EX and MPI test back-to-back. Each data point plotted reports the maximum average bandwidth (or minimum average latency) for that benchmark and transfer size.

The `testlarge` benchmark reports bandwidths in units of “MiB/s” ($2^{20}$ bytes per second), whereas the IMB tests use “MB/s” ($10^{6}$ bytes per second). Both have been converted to “GiB/s” ($2^{30}$ bytes per second) for the bandwidth plots.

C. Systems and their software environments

1) OLCF Summit:

- Hardware in each IBM Power System AC922 node:
  - Dual-socket 22-core 3.07GHz IBM POWER9 CPUs
  - Dual-rail Mellanox EDR InfiniBand with “ConnectX-5 Ex” HCAs
  - 512 GB DDR4-2666 system memory
  - 6x NVIDIA Volta V100 GPUs (not used)
- Software environment used:
  - Red Hat Enterprise Linux release 8.2
  - Compute node kernel `4.18.0-193.46.1.el8_2.ppc64le`
  - GASNet-EX v2022.3.0, ibv-conduit
  - Relevant environment modules: (provided by the HPC center)
    * gcc/9.1.0
    * spectrum-mpi/10.4.0.3-20210112
2) **Cori Haswell**:

- Hardware in each Cray XC40 node:
  - Dual-socket 16-core 2.3GHz Intel Xeon E5-2698v3 “Haswell” CPUs
  - Cray Aries network with Dragonfly topology [27]
  - 128 GB DDR4-2133 memory
- Software environment used:
  - SUSE Linux Enterprise Server 15 SP2
  - Compute node kernel 5.3.18-24.46_6.0.29-cray_ari_c
  - GASNet-EX v2022.3.0.aries-conduit
  - Relevant environment modules: (provided by the HPC center)
    * PrgEnv-gnu/6.0.10
    * gcc/11.2.0
    * craype/2.7.10
    * cray-mpich/7.7.19
    * gni-headers/5.0.12.0-7.0.3.1_3.12__gd0d73fe.ari
    * craype-network-aries
    * craype-haswell

3) **Perlmutter SS-10**:

- Hardware in each HPE Cray EX node:
  - Single-socket 64-core 2.45GHz AMD EPYC 7763 “Milan” CPU
  - 2x Mellanox “ConnectX-5” HCAs (100Gb/s each) connected via HPE Slingshot network
  - 256 GB DDR4-3200 system memory
  - 4x NVIDIA Ampere A100 GPUs (not used)
- Software environment used:
  - SUSE Linux Enterprise Server 15 SP3
  - Compute node kernel 5.3.18-150300.59.43_11.0.51-cray_shasta_c
  - GASNet-EX v2022.3.0.ofi-conduit, with verbs;ofi_rxm (InfiniBand) libfabric provider
  - Relevant environment modules: (provided by the HPC center)
    * PrgEnv-gnu/8.3.3
    * gcc/11.2.0
    * craype/2.7.16
    * cray-mpich/8.1.17
    * cray-pmi/6.1.3
    * libfabric/1.11.0.4.124
    * craype-network-ofi
    * craype-x86-milan

4) **Perlmutter SS-11**:

- Hardware in each HPE Cray EX node:
  - Single-socket 64-core 2.45GHz AMD EPYC 7763 “Milan” CPU
  - 4x HPE “Cassini” NICs (200Gb/s each) connected via HPE Slingshot network
  - 256 GB DDR4-3200 system memory
  - 4x NVIDIA Ampere A100 GPUs (not used)
- Software environment used:
  - SUSE Linux Enterprise Server 15 SP3
  - Compute node kernel 5.3.18-150300.59.43_11.0.51-cray_shasta_c
  - GASNet-EX v2022.3.0.ofi-conduit, with cxix (HPE Cassini) libfabric provider
  - Relevant environment modules: (provided by the HPC center)
    * PrgEnv-gnu/8.3.3
    * gcc/11.2.0
    * craype/2.7.16
    * cray-mpich/8.1.17
    * cray-pmi/6.1.3
    * libfabric/1.15.0.0
    * craype-network-ofi
    * craype-x86-milan