An Introduction to GASNet-EX for Chapel Users

Paul H. Hargrove

gasnet-staff@lbl.gov gasnet.lbl.gov

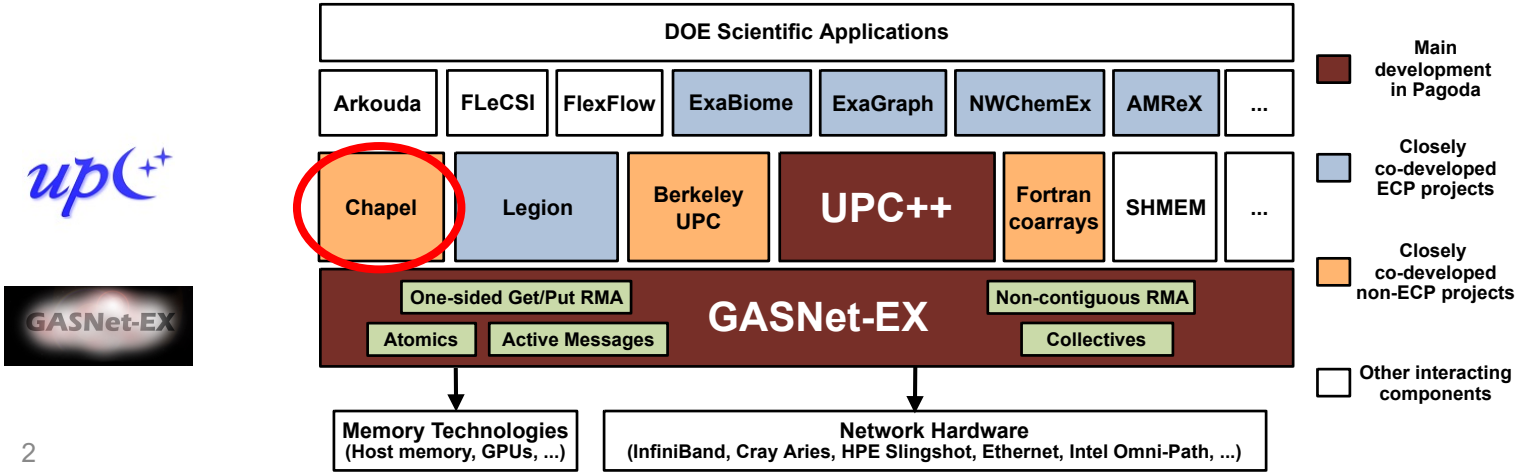
Joint work with Dan Bonachea and the LBNL Pagoda Project (CRD/CLaSS)

The Pagoda Project

https://go.lbl.gov/pagoda

Support for lightweight communication for exascale applications, frameworks and runtimes

- **GASNet-EX** middleware layer providing a network-independent interface suitable for Partitioned Global Address Space (PGAS) runtime developers
- **UPC++** C++ PGAS library for application, framework and library developers, a productivity layer over GASNet-EX



BACKGROUND AND HISTORY

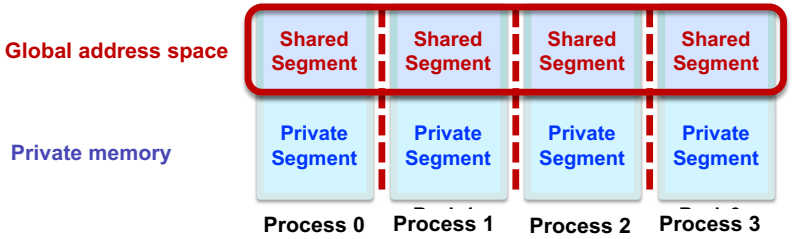
GASNet…

- is "**G**lobal **A**ddress **S**pace **Net**working"
- is an AM and RMA API for implementing PGAS models
- is designed for compilers and authors of low-level code
- is MPI-interoperable on most platforms
- performs comparably to (and often better than) MPI
- has influenced design of RMA in MPI-3.0 and later
- **is recommended for Chapel multi-locale communication on non-Cray systems (like Summit)**
  - `third-party/gasnet`

The PGAS model

Partitioned Global Address Space

- Support global memory
  - leveraging the network's RDMA capability
- Distinguish private and shared memory
- Separate synchronization from data movement



Languages that provide PGAS:

Chapel, UPC, Fortran coarrays (Fortran 2008+), X10, Titanium…

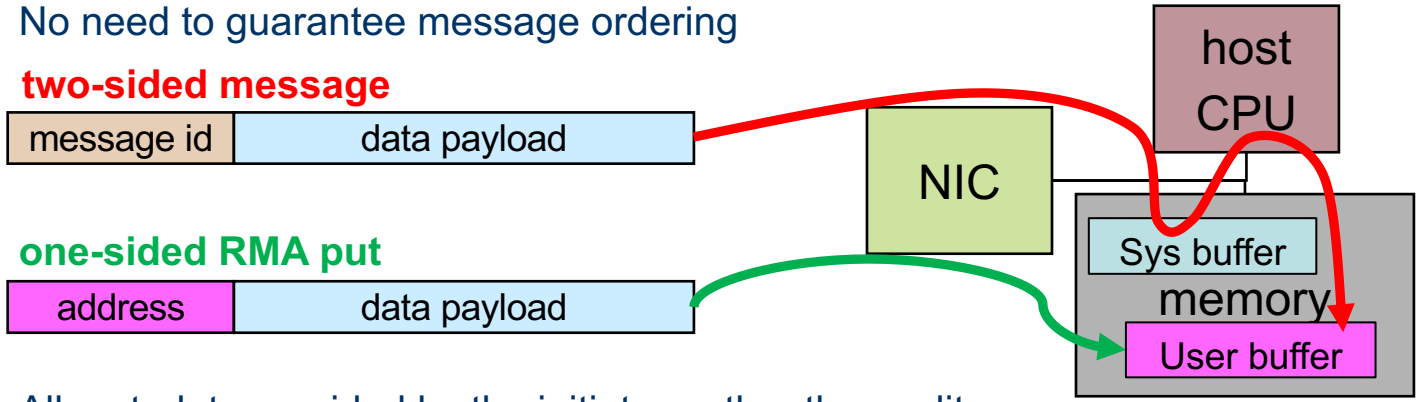
Libraries that provide PGAS:

UPC++, OpenSHMEM, Co-Array C++, Global Arrays, DASH…

A key semantic property is support for one-sided RMA

Reducing communication overhead using one-sided RMA

- Idea: Let each process directly access another's memory via a global pointer
- Communication is **one-sided :** there is no "receive" operation
  - No need to match sends to receives
  - No unexpected messages
  - No need to guarantee message ordering



  - All metadata provided by the initiator, rather than split between sender and receiver
  - Supported in hardware through RDMA (Remote Direct Memory Access)
- Looks like shared memory: shared data structures with asynchronous access

GASNet-1: Historical Overview

- Started in 2002 to provide a portable network communication runtime for three PGAS languages:
  - UPC, Titanium and CAF

- Primary features:
  - Non-blocking RMA (one-sided Put and Get)
  - Active Messages (simplification of Berkeley AM-2)
- Chosen over then-current alternatives: MPI-2, ARMCI

GASNet: Adoption and Portability

Client runtimes

LBNL UPC++	Legion (Stanford/NVIDIA/…)	Titanium
Berkeley UPC	Caffeine (Fortran 2008+)	OpenSHMEM reference impl.
GCC/UPC	Rice Co-Array Fortran	Omni XcalableMP
Clang UPC	OpenUH Co-Array Fortran	PARADISE++ Devastator
Chapel (Cray/HPE)	OpenCoarrays in GCC Fortran	At least 6 others known to us

Network conduits

OpenFabrics Verbs (**InfiniBand**)	IBM DCMF (BG/P)	Myricom GM (Myrinet)
Mellanox MXM and VAPI (InfiniBand)	IBM LAPI (Colony and Federation)	Dolphin SISCI
Cray uGNI (Gemini and **Aries**)	Cray Portals3 (Seastar)	Sandia Portals4
Intel PSM2 (OmniPath)	SHMEM (Cray X1 and SGI Altix)	
IBM PAMI (BG/Q and others)	Quadric elan3/4 (QsNet I/II)	
UDP (any TCP/IP network)	**OFI (Slingshot, Omni-Path…)**	**Shared memory (no network)**
MPI 1.1 or newer	**UCX (many)**	

Supported platforms

- Over 10 compiler families, 15 operating systems and dozens of architectures

* These lists and counts include both current and past support

https://gasnet.lbl.gov

GASNet-EX: Overview

- GASNet-EX is the next generation of GASNet
  - Motivated by the needs of newer programming models such as UPC++, Legion and Chapel
  - Incorporates 20 years of lessons learned and focuses on the challenges of emerging exascale systems
  - Provides backward compatibility for GASNet-1 clients
- Motivating goals include
  - Support more client asynchrony
  - Enable more client adaptation
  - Improve memory footprint
  - Improve threading support
  - Support offload to network h/w
  - Support for device memory

https://gasnet.lbl.gov

API OVERVIEW

GASNet-EX API Highlights

- Active Messages (AM)
  - Restricted form of remote procedure call
  - Used to implement language/library features
- Remote Memory Access (RMA)
  - Put, Get and Atomic RMW operations
  - Leverages network RDMA hardware when available
- Memory kinds
  - Extends RMA to non-host memory such as GPUs
- Non-contiguous RMA
  - E.g. move entire multi-dimensional array sections in a single operation
- Teams and Collectives

GASNET-EX AND CHAPEL

Some Chapel History

- December 2006
  - Chapel 0.4 released with only single-locale support
- July 2007
  - GASNet 1.8.0 source added to `third-party` directory
- November 2007
  - GASNet 1.10.0 source added to `third-party` directory
- March 2008
  - Chapel 0.7 released with multi-locale support based on GASNet 1.10.0

Why Chapel chose GASNet

- “Portability was probably the biggest factor for going with GASNet. ”
- “GASNet seemed to be on an upward swing, gaining momentum. Other general languages such as CAF, UPC, and Titanium had implementations on GASNet.”
- “GASNet’s Active Message interface.”
- “We observed better performance for contiguous data.”
- “Non-blocking models provided by GASNet are pretty rich. Enables possible compiler communication optimizations.”
- “Chapel did have ports to ARMCI, MPI, and PVM at some point. They definitely were not as natural a fit.”
- “The GASNet team proved to be interested in creating stable, solid, useful, well-engineered software, so it made it an increasingly obvious and ‘safe’ bet over time.”

Slide credit: Brad Chamberlain, June 3, 2022;
Drawing in part from his notes and communications from March 2006 and January 2007.

How Chapel uses GASNet today

Currently, the Chapel runtime uses only features which were available in the GASNet-1 releases:

- Active Messages
  - Primary use is for remote task launch
- RMA Put and Get
  - Data movement
- Non-contiguous RMA Put and Get
  - Used for strided transfers (array sections)

GASNet-EX/Chapel Highlight from April 2021

This figure shows improvements in scalability of the Arkouda Argsort benchmark made in April 2021.

While some of the improvement comes from aggregation improvements on the Chapel side, the majority is due to improvements in ibv-conduit handling of dynamic memory registration, especially on large-memory systems.

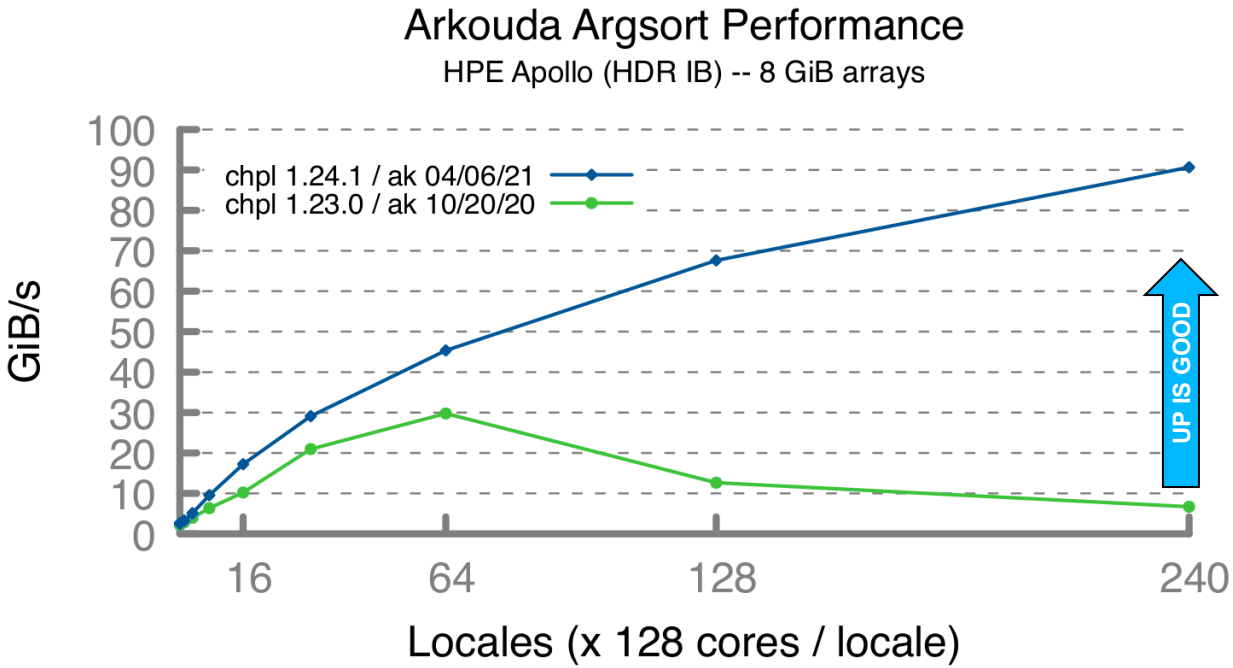


Figure provided by the Chapel team at HPE

https://gasnet.lbl.gov

BERKELEY LAB

GASNet-EX/Chapel Highlight from June 2021

This figure shows a roughly 20% improvement in Arkouda Argsort performance, achieved in June 2021.

This is the result of joint work with the Chapel team at HPE to address false sharing in a Mellanox-provided library which impacted performance of GASNet-EX (and thus Chapel) as one increases the number of CPU cores used per node.

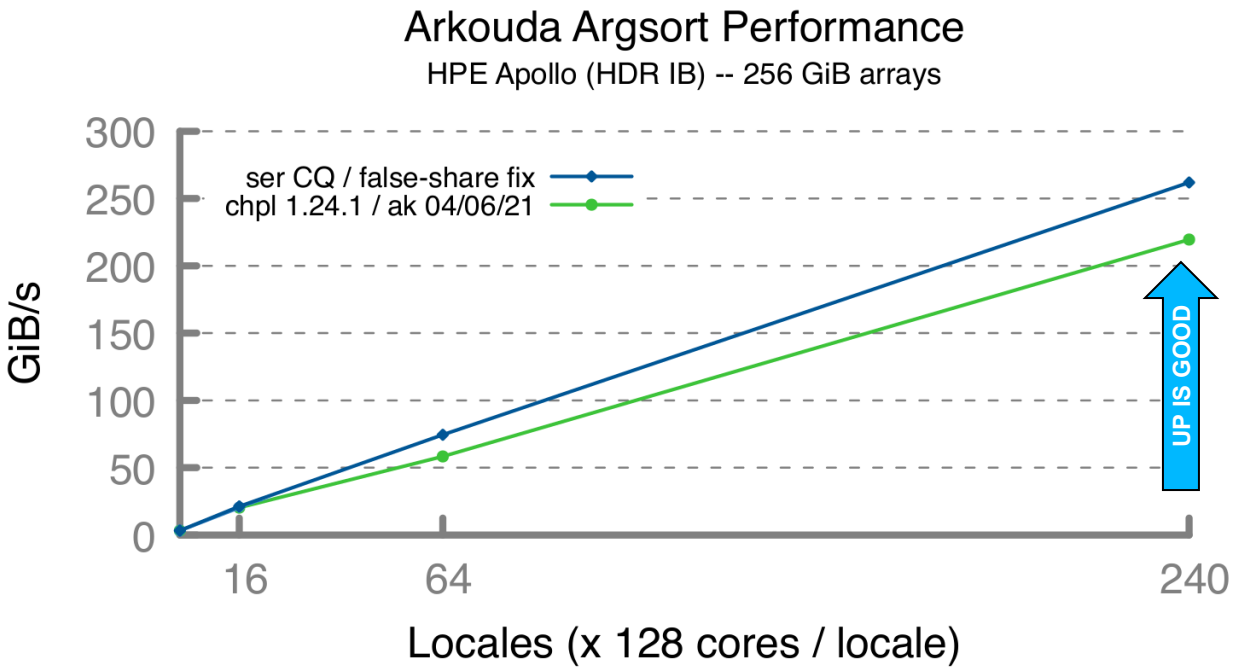


Figure provided by the Chapel team at HPE

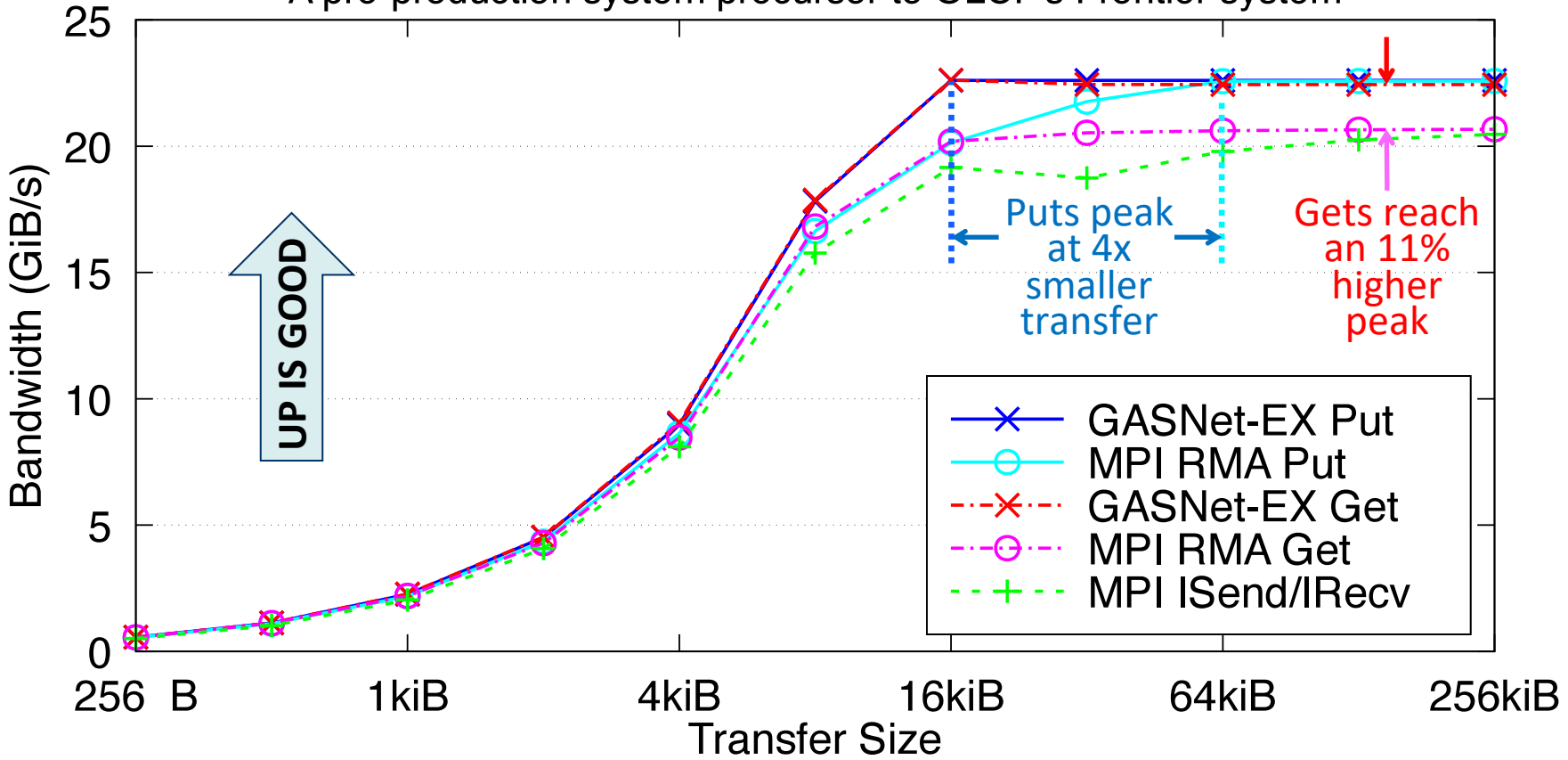
https://gasnet.lbl.gov

BERKELEY LAB

PERFORMANCE

Crusher: HPE Cray EX / Slingshot-11, HPE Cray MPI

A pre-production system precursor to OLCF's Frontier system

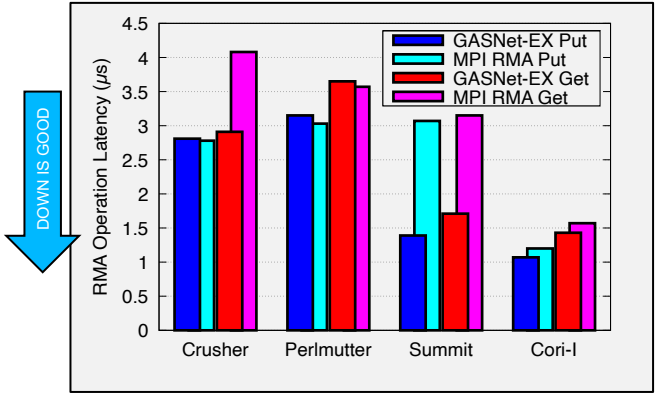


A comparison of uni-directional point-to-point host-memory flood bandwidth benchmarks, run March 2022 on OLCF's Crusher system.
Shows the performance of RMA (Put and Get) operations using GASNet-EX and both RMA and message-passing (Isend/Irecv) using HPE Cray MPI.
Results were obtained using current GASNet tests and Intel MPI Benchmarks, respectively.

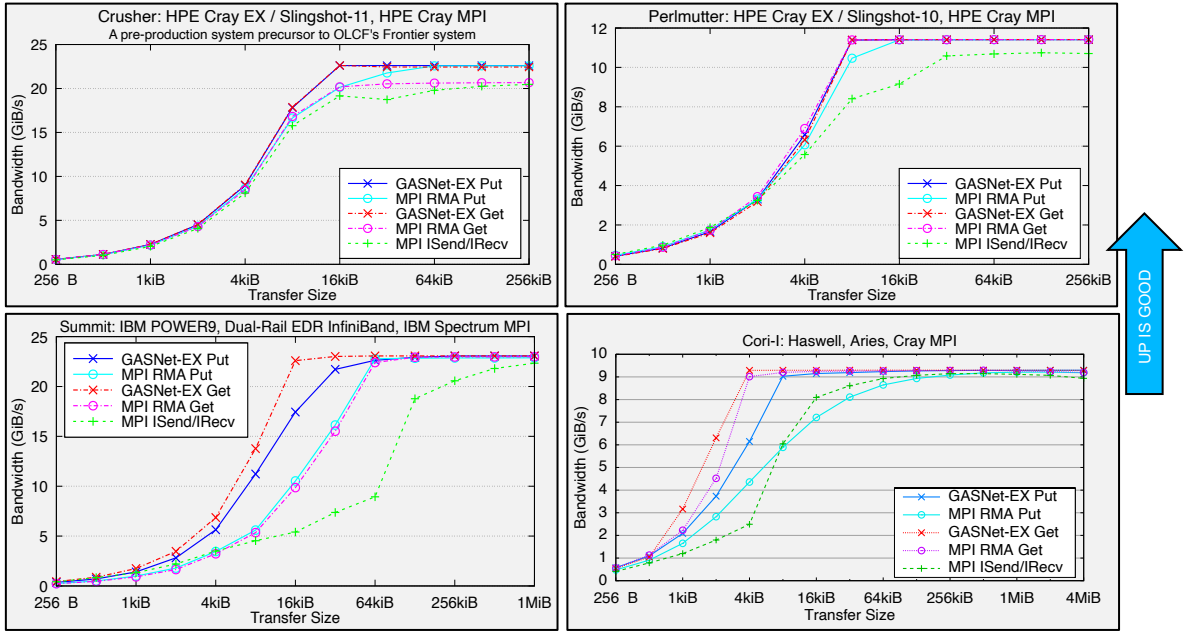
GASNet-EX RMA Performance versus MPI RMA and Isend/Irecv

- Four distinct network hardware types
- The performance of GASNet-EX matches or exceeds that of MPI RMA and message-passing:
  - 8-byte Put latency up to 55% better
  - 8-byte Get latency up to 45% better
  - Better flood bandwidth efficiency: often reaching same or better peak at ½ or ¼ the transfer size

8-Byte RMA Operation Latency (one-at-a-time)



Uni-directional Flood Bandwidth (many-at-a-time)



Cori results collected September 2018; all others collected March 2022.
GASNet-EX tests were run using then-current GASNet-EX library and its tests.
MPI tests were run using then-current center default MPI version and Intel MPI Benchmarks.
For experimental details see Languages and Compilers for Parallel Computing (LCPC'18).
doi.org/10.25344/S4QP4W

GASNet-EX

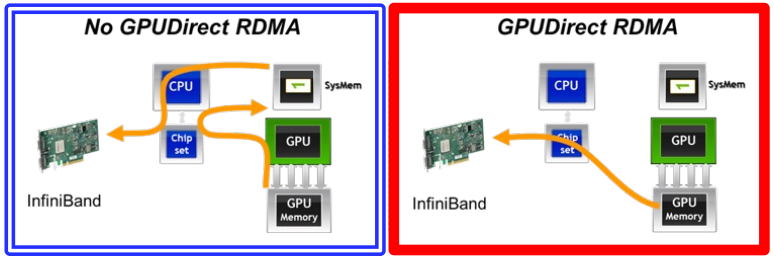
Hewlett Packard Enterprise

CRAY

IBM

BERKELEY LAB

RMA to/from GPU Memory



Measurements of flood bandwidth of **`upcxx::copy()`** on Summit

Difference between the two most recent releases shows benefit of GASNet-EX's support for GPUDirect RDMA (GDR)

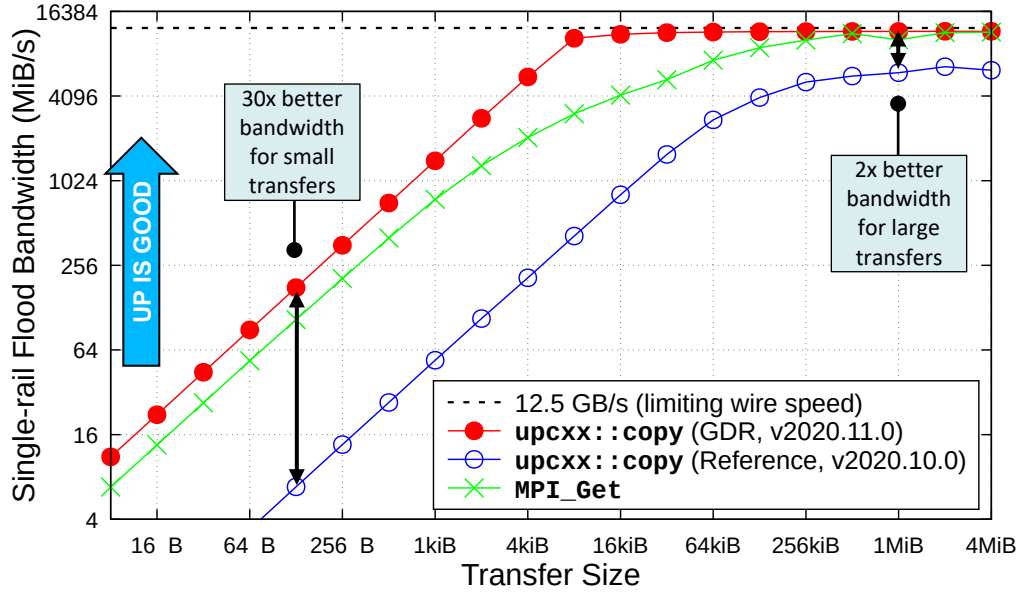
- No longer staging through host memory
- Large xfers: 2x better bandwidth
- Small xfers: up to 30x better bandwidth

Get operations to/from GPU memory now perform comparably to host memory

Preliminary comparisons to MPI-3 RMA in GDR-enabled IBM MPI show UPC++ saturating more quickly to the peak

RMA Get Bandwidth (remote GPU to local host memory)

UPC++ 2020.11.0 vs. IBM Spectrum MPI 10.3.1.2 on OLCF Summit



UPC++ results were collecting using the version of the **`cuda_benchmark`** test that appears in the 2020.11.0 release.
MPI results are from **`osu_get_bw`** test in a CUDA-enabled build of OSU Micro-Benchmarks 5.6.3.
All tests were run between two nodes of OLCF Summit, over its EDR InfiniBand network.

Switching things up:
RMA between AMD GPUs on Slingshot-10

Highlights:

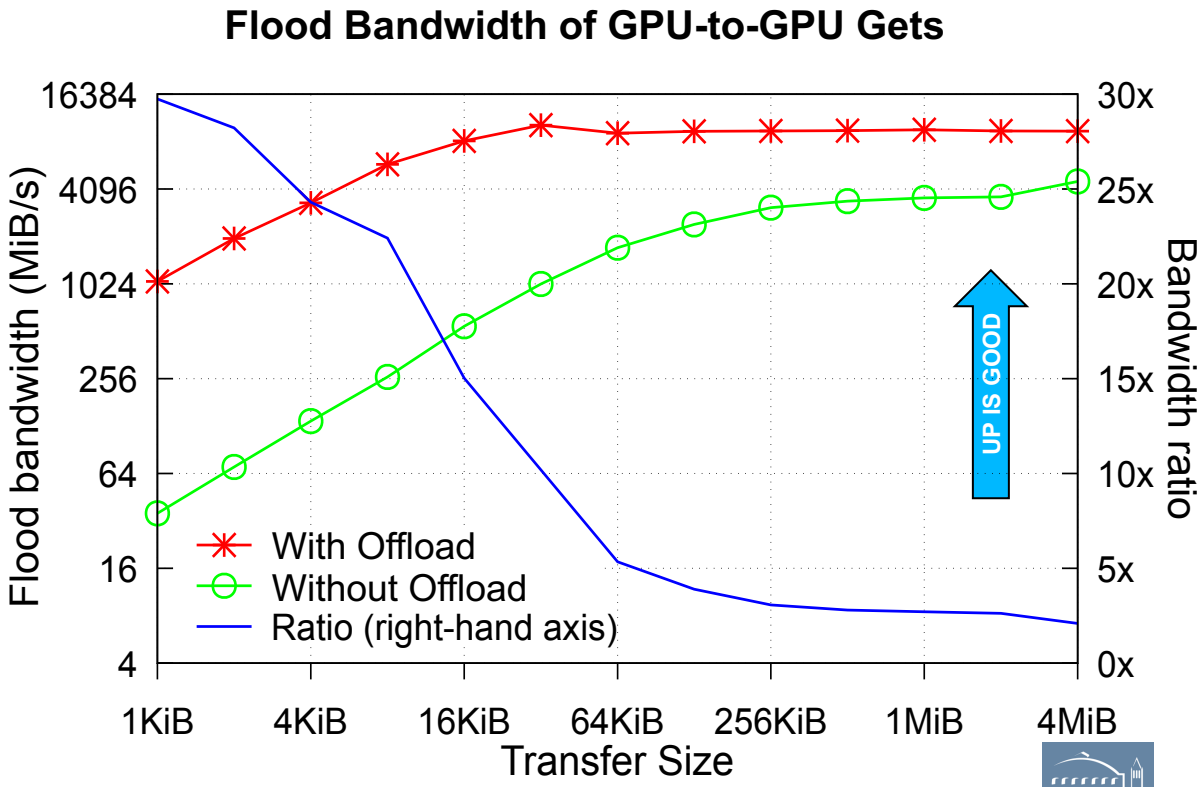
- Different network stack
- Different GPU vendor
- 2x improvement for large xfers
- 30x improvement for small xfers

Expanded GPU support, Sep 2021:

- AMD GPUs (ROCmRDMA)
- ucx-conduit

System details:

- AMD CPUs and GPUs
- HPE Slingshot-10
  - HPE Rosetta switch
  - Mellanox ConnectX-6 NIC



BERKELEY LAB

FUTURE WORK

Top Priorities for the Future

- Extend memory kinds to ofi-conduit
  - Current support includes ibv-conduit and ucx-conduit
  - However, libfabric (ofi-conduit) is the API for HPE Slingshot-11
    - Slingshot-11 is the network for all three of DOE's announced exascale systems (Frontier, Aurora and El Capitan), among others
- Extend memory kinds to Intel GPUs
  - Current support Nvidia and AMD GPUs
  - However, Aurora will feature Intel GPUs
- Assist the Chapel team in efforts to use more of GASNet-EX

Acknowledgements

This research was funded in part 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.

THANK YOU!

gasnet-staff@lbl.gov

gasnet.lbl.gov

LCPC'18: Bonachea, Hargrove.
"GASNet-EX: A High-Performance, Portable Communication Library for Exascale",
https://doi.org/10.25344/S4QP4W