WBS 2.3.1.14

UPC++ and GASNet: PGAS Support for Exascale Apps and Runtimes

Paul H. Hargrove (PI)

with the contributions of current staff: Dan Bonachea, Johnny Corbino, Amir Kamil, Colin A. MacLean, Damian Rouson, Daniel Waters

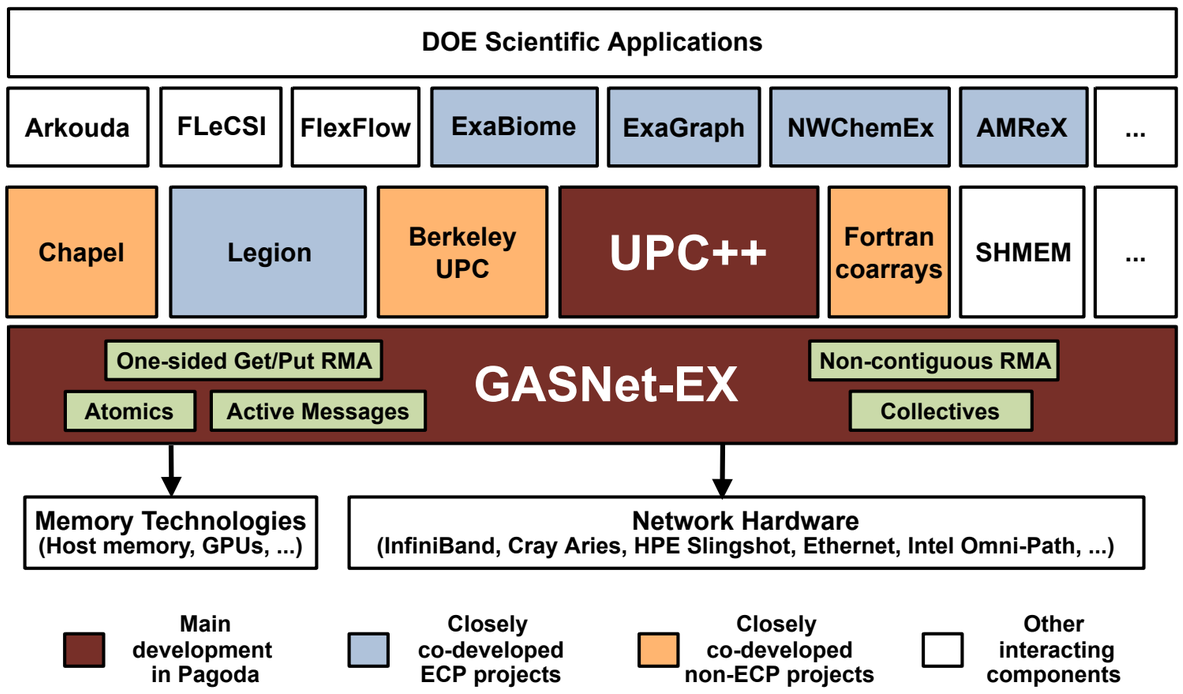
and former team members: John Bachan, Scott B. Baden, Steven Hofmeyr, Max Grossman, Mathias Jacquelin, Brian van Straalen

The Pagoda Project at Lawrence Berkeley National Lab (go.lbl.gov/pagoda)

The Pagoda project is developing a programming system to support HPC application development using the Partitioned Global Address Space (PGAS) model. The first component is GASNet-EX, a portable, high-performance, global-address-space communication library. The second component is UPC++, a C++ template library. Together, these libraries enable agile, lightweight communication such as arises in irregular applications, libraries and frameworks running on exascale systems.

GASNet-EX is a portable, high-performance communications middleware library which leverages hardware support to implement Remote Memory Access (RMA) and Active Message communication primitives. GASNet-EX supports a broad ecosystem of alternative HPC programming models, including UPC++, Legion, Chapel and multiple implementations of UPC and Fortran Coarrays. GASNet-EX is implemented directly over the native APIs for networks of interest in HPC. The tight semantic match of GASNet-EX APIs to the client requirements and hardware capabilities often yields better performance than competing libraries.

UPC++ provides high-level productivity abstractions appropriate for Partitioned Global Address Space (PGAS) programming such as: remote memory access (RMA), remote procedure call (RPC), support for accelerators (e.g. GPUs), and mechanisms for aggressive asynchrony to hide communication costs. UPC++ implements communication using GASNet-EX, delivering high performance and portability from laptops to exascale supercomputers. HPC application software using UPC++ includes: MetaHipMer2 metagenome assembler, SIMCoV viral propagation simulation, NWChemEx TAMM, and graph computation kernels from ExaGraph.



UPC++ Then and Now (upcxx.lbl.gov)

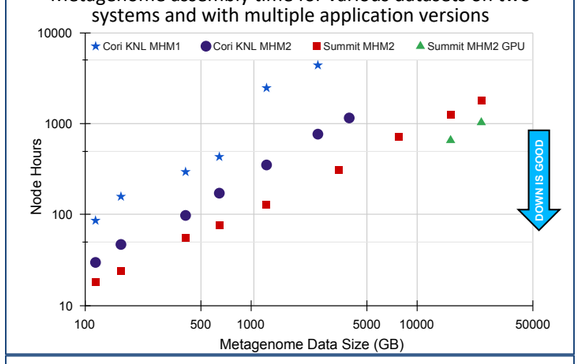
- UPC++ “then”
  - v0.1 began in 2014 doi:10.1109/IPDPS.2014.115
- UPC++ “now”
  - Began with ECP funding and is known as “v1.0”
  - See doi:10.25344/S4V88H for an introduction to UPC++ v1.0
  - Major changes to the API incorporating lessons learned
  - Entirely new library design and implementation
    - GASNet-EX replaces GASNet-1 as the network backend
- Notable differences include:
  - New asynchrony model
  - Better support for multi-threading and hierarchical programming
  - RMA extended to expose modern hardware resources and capabilities, including GPU memory and remote atomics
  - Serialization APIs to simplify communication of rich C++ objects
  - Expanded support for subset teams and collectives replace experimental features in v0.1
  - Design and implementation choices to enable execution at extreme scale

GASNet Then and Now (gasnet.lbl.gov)

- GASNet “then”
  - First GASNet spec in 2002 doi:10.25344/S4MW28
  - HPCWire article on 20th anniversary: doi:10.25344/S4BP4G
  - Now referred to as “GASNet-1”
- GASNet “now”
  - Began with ECP funding and is known as “GASNet-EX”
  - See doi:10.25344/S4QP4W for an introduction to GASNet-EX
  - Many *additions* to the API incorporating lessons learned
    - GASNet-EX retains compatibility for GASNet-1 clients
- Notable differences include:
  - More expressive APIs enable new client behaviors, including:
    - Increased opportunities for client asynchrony and overlap
    - Client adaptation to transient resource constraints
  - Extended RMA APIs expose modern hardware capabilities:
    - Network-accelerated atomics
    - Direct transfers between NIC and device (e.g GPUs)
  - Implementation improvements to scalability in time and memory enable runs at larger scales

Four Notable Project Accomplishments Under ECP Funding

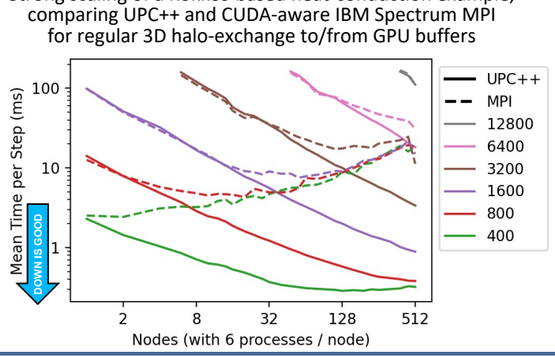
Metagenome assembly time for various datasets on two systems and with multiple application versions



Using their UPC++-based MetaHipMer2 (MHM2) application, the ExaBiome AD project completed a world-record breaking assembly of a 25TB metagenome on OLCF Summit in August 2021. They broke this record with assembly of a 30TB metagenome a year later (not shown above) on 1500 nodes of Summit (63k cores and 9k GPUs).

Figure courtesy of Rob Egan and Steven Hofmeyr of the ExaBiome team.

Strong scaling of a Kokkos-based heat-conduction example, comparing UPC++ and CUDA-aware IBM Spectrum MPI for regular 3D halo-exchange to/from GPU buffers

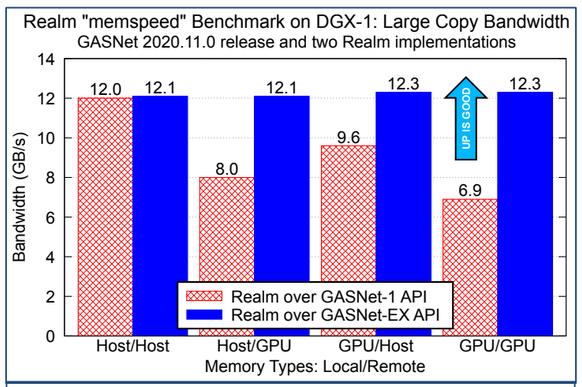


This figure shows strong scaling on OLCF Summit
- Using six processes/node and one GPU/process
- Six problem sizes and node counts from 1 to 512
- Runs with UPC++ and CUDA-aware IBM Spectrum MPI

UPC++ consistently meets or exceeds the performance of MPI.
For details see PAW-ATM’21: doi:10.25344/S4630V

Colors indicate problem size (side length of the computed cube).
Solid and dashed line styles indicate the UPC++ and MPI versions respectively.

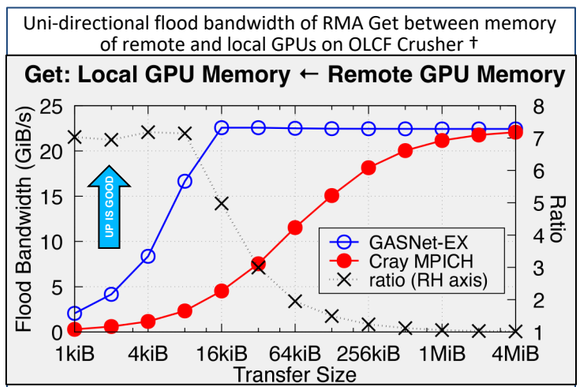
Realm "memspeed" Benchmark on DGX-1: Large Copy Bandwidth
GASNet 2020.11.0 release and two Realm implementations



This figure shows performance of Realm (the low-level runtime of the Legion ST project) with their GASNet-1 (“then”) and GASNet-EX (“now”) backends. The latter uses many features new to GASNet-EX. These results show one benefit of using GASNet-EX features: performance of RMA involving GPU memory matches that of host memory RMA.
See also SC21 poster: doi:10.25344/S4P306

Data courtesy of Sean Treichler of the Legion team.

Uni-directional flood bandwidth of RMA Get between memory of remote and local GPUs on OLCF Crusher †



Compared to GPU-aware Cray MPICH (filled red circles), GASNet-EX (open blue circles) reaches peak performance using smaller transfers. Both implement RMA transfers involving GPU memory using ROCmRDMA (direct PCI peer-to-peer transfers) between the GPU and NIC. GASNet-EX has up to a 7.2x RMA flood bandwidth advantage (the ratio is plotted with ×s against the right-hand axis).

† Crusher is a preproduction system precursor to OLCF’s Frontier system.

This research was supported 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 Argonne Leadership Computing Facility, which is a DOE Office of Science User Facility supported under Contract DE-AC02-06CH11357.

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.