**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** 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 GAS-EX, delivering high performance and portability from laptops to exascale supercomputers. HPC application software using UPC++ includes: MetahipMer2 metagenome assembler, SIMoV viral propagation simulation, NWChem EX 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 atomic
- Serialization APIs to simplify communication of rich C++ objects
- Expanded support for subset teams and collects replace experimental features in v0.1
- Design and implementation choices to enable execution at extreme scale

---

**Four Notable Project Accomplishments Under ECP Funding**

- **GASNet “then”**
  - First GASNet spec in 2002 doi:10.25344/S4MW28
  - HPCWire article on 20th anniversary: doi:10.25344/S4B4P4
  - 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 atoms
  - Direct transfers between NIC and device (e.g GPUs)
- Implementation improvements to scalability in time and memory enable runs at larger scales

---

**GASNet Then and Now (gasnet.lbl.gov)**

- **First GASNet spec in 2002**
  - doi:10.25344/S4MW28
- **HPCWire article on 20th anniversary**
  - doi:10.25344/S4B4P4
- **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 atoms
  - Direct transfers between NIC and device (e.g GPUs)
- Implementation improvements to scalability in time and memory enable runs at larger scales

---

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’s Office of Science and the National Nuclear Security Administration.

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.

© 2023, Lawrence Berkeley National Laboratory