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

Paul H. Hargrove (PI)
With team members: Dan Bonachea, Max Grossman, Amir Kamil, Colin A. MacLean, Daniel Waters

UPC++ at Lawrence Berkeley National Lab (upcxx.lbl.gov)

- UPC++ is a C++11 PGAS library
- Lightweight, asynchronous, one-sided communication (RMA)
- Asynchronous remote procedure call (RPC)
- Data transfers may be non-contiguous
- Futures manage asynchrony, enable communication overlap
- Collectives, teams, remote atomic updates
- Provides building blocks to construct irregular data structures

Latest software release: March 2021
- Easy on-ramp and integration
- Latest software release: March 2021
- Interoperable with MPI
- Enables incremental development
- Runs on systems from laptops to supercomputers
- Data transfers may be non-contiguous

UPC++ enhances overlap in performance of communication patterns

Integration efforts with EvxGraph (WBS2.2.6.07)

• With PNNL team, have developed two UPC++ versions of a graph matching problem from their IPDPS’19 paper
• RMA version uses Puts to communicate among processes
• RPC version uses asynchronous remote procedure calls to execute logic on remote parts of the graph
• Initial results on NERSC Cori Haswell (3.848-Gh Friendster):
  - Both UPC++ versions competitive with (or better than) best MPI versions up to at least 4,096 processes

Integration efforts with NWCHEMEX (WBS2.2.1.02)

• Ported TACC code from Global Arrays to UPC++
• TAMM offers distributed in-memory data store and compute for NWCHEMEX
• Achieved comparable performance to non-distributed code
• Continuously updating UPC++ enabled optimization opportunities

Case 1: Easy Distributed Hash-Table via Function Shipping and Futures

• Distributed hash-table design is based on function shipping
  - RPC inserts the key metadata at the target
  - Once the RPC completes, an attached callback issues a one-sided RMA Put (rput) to store the value data

```
// the global variable corresponds to each logical node
std::unordered_map<int, global_ptr> local_map;
insert in local map, return future of rput
future<> rput_future = rput(key, value); // attach callback to complete rput
```

• Benefits:
  - Use of RPC simplifies distributed data-structure design
  - Argument passing, remote queue management and progress engine are factored out of the application code
  - Argument passing occurs at compile time

Efficient weak scaling to 512 nodes (34k cores) on Cori Xeon Phi

Case 2: Asynchronous Sparse Matrix Solvers

• A time consuming operation in multifrontal sparse solvers:
  - Extend-addr: update a distributed sparse matrix, scattering the packed data source

- Challenge:
  - This operation has low computational intensity and exhibits irregular communication patterns

- Solution:
  - UPC++ function shipping via RPC enables efficient communication and asynchrony, increasing overlap and improving performance of Extend-add

- Impact:
  - UPC++ enhances overlap in Extend-add, yielding up to a 1.6x speedup over MPI collective and 3.1x over MPI message-passing implementations. The green line in the figure corresponds to the fastest of these two variants.

Support for GPUDirect RDMA (GDR) – UPC++ and Legion/Realm Benchmarks

• GASNet-EX supports GPUDirect RDMA (GDR) since 2020.11.0
  - Removes host CPU and memory bottlenecks from one-sided transfers from GPU memory (see diagram)
  - Currently supports NVIDIA GPUs + Mellanox InfiniBand
  - Other accelerators and networks are subject of future work

• Preliminary comparison of UPC++-MPI to MPI-3 RMA in GDR-enabled IBM MPI show UPC++ saturating more quickly to the peak (top-right plot)

• Realm is the low-level runtime for the Legion Programming System (WBS 2.3.1.08)
  - Communications services originally implemented over GASNet-1
  - GASNet-1 backend still works using legacy API support in current GASNet-EX

• Realm introduced a new GASNet-EX communications backend (Dec 2020)
  - Enables capabilities specific to GASNet-EX
  - Leverages Immediate, NPAck, and local completion events for AM
  - Most notable new capability is GDR support

• Some performance benefits of using GASNet-EX’s GDR support in Realm:
  - Large GPU memory xfers: same bandwidth as host memory (bottom-right plot)
  - Small GPU memory xfers: 2.2x to 3.0x latency improvement