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

Scott B. Baden (PI) and Paul Hargrove (co-PI) 2.3.1.14

GASNet-EX at Lawrence Berkeley National Lab (http://gasnet.lbl.gov)

- GASNet-EX: communications middleware to support exascale clients
  - One-sided communication − Remote Memory Access (RMA)
  - Active Messages - remote procedure call
  - Implemented over the native APIs for all networks of interest to ECP
- GASNet-EX is an evolution of GASNet-1 for exascale
  - Retains GASNet-1's wide portability (laptops to production supercomputers)
  - Provides backwards compatibility with GASNet-1 clients
  - Focus remains on one-sided RMA and Active Messages
  - Reduces CPU and memory overheads
  - Improves many-core support
- GASNet-1 clients include:
  - Multiple UPC and CAF/Fortran08 compilers
  - Stanford's Legion Programming System
  - Cray Chapel Language
  - OpenSHMEM Reference Implementation
  - Omni XcalableMP Compiler
- GASNet-EX clients include:
  - ECP ST: UPC++ and Legion; and PaRSEC exploring
  - ECP AD: ExaBiome exploring
  - non-ECP: Cray Chapel exploring

- GASNet-EX augments and enhances GASNet-1
  - Enhancements address needs of modern asynchronous PGAS models
  - Interfaces adjusted for improved scalability
  - Features critical to UPC++ are being co-designed
  - Using input from Legion and Cray Chapel, who plan to adopt the new APIs

- Features delivered in FY17 and so far in FY18 include:
  - "Immediate mode" injection to avoid stalls due to back-pressure
  - Explicit handling of local-completion (source buffer lifetime)
  - New AM interfaces, for instance to reduce buffer copies between layers
  - Vector-Index-Strided for non-contiguous point-to-point RMA
  - Remote Atomics, implemented with NIC offload where available
- Features to deliver in remainder of FY18 include:
  - Teams and non-blocking collectives
  - Dependent operations to control ordering of in-flight operations
- Features for FY19 and beyond may include:
  - Offset-based addressing
  - Multiple endpoints/segments, for instance to enhance multithreading support
  - Support for "out-of-segment" remote addresses

Highlights from Current Work

Example of EX interface updates: RMA Put

- GASNet-1:

```
gasnet_handle_t
gasnet_put_nb(gasnet_node_t node, void *dest_addr,
              void *src_addr, size_t nbytes);
```

- GASNet-EX:

```
gex_Event_t
gex_RMA_PutNB(gex_TM_t tm, gex_Rank_t rank, gex_Addr_t dest_addr,
              void *src_addr, size_t nbytes,
              gex_Event_t *lc_opt, gex_Flags_t flags);
```

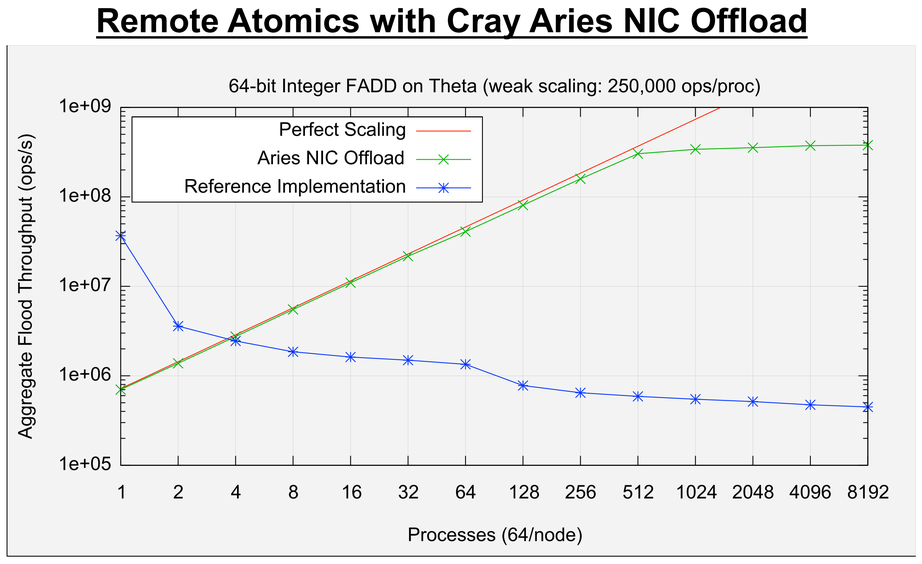
- **`gex_Event_t`** return type introduces events to generalize GASNet handles.
- **`tm`** argument adds team (ordered sets of ranks), into which **`rank`** indexes.
- **`gex_Addr_t`** type will enable offset-based addressing via same interface.
- **`lc_opt`** argument introduces explicit control over local completion, generalizing the bulk/non-bulk interfaces of GASNet-1.
- **`flags`** argument provides extensibility. For instance:
  - To select new optional behaviors (e.g., immediate mode and offset-based addressing)
  - To provide assertions regarding the arguments (e.g., to streamline the operation)

Vector-Indexed-Strided (VIS) Interfaces for Non-Contiguous RMA

SYSTEM	NETWORK	INDEXED		STRIDED		VECTOR	
		GET	PUT	GET	PUT	GET	PUT
Cori-I	Cray Aries	11.68 ×	10.06 ×	12.55 ×	12.63 ×	8.83 ×	7.69 ×
Theta	Cray Aries	10.03 ×	7.70 ×	11.10 ×	9.94 ×	7.13 ×	5.89 ×
Titan	Cray Gemini	7.33 ×	7.21 ×	8.09 ×	8.61 ×	5.33 ×	5.51 ×
SummitDev	Mellanox InfiniBand	5.45 ×	5.17 ×	5.67 ×	5.63 ×	4.29 ×	4.29 ×
Cetus	IBM BG/Q	2.66 ×	3.49 ×	4.01 ×	4.34 ×	2.10 ×	2.82 ×

- Formalizes and generalizes an unofficial extension to GASNet-1
- Three metadata formats for different scenarios
  - Vector: fully general array of iovec-like (address, length) pairs
  - Indexed: array of addresses and a single length
  - Strided: arbitrary rectangular sections of dense multi-dimensional arrays
    - GASNet-EX adds transposition and reflection capabilities
- The table above shows the speed-up resulting from recent work that enables use of aggressive pack/unpack optimizations. Details of the benchmark are given in the report for ECP Milestone STPM17-5.

Remote Atomics with Cray Aries NIC Offload



- Implements the Atomic Domains concept (first introduced by UPC 1.3)
  - Domains permit use of NIC offload even when not coherent with CPU
  - Domains are created collectively outside the critical path
  - A Domain has an associated data type and set of allowed operations
    - Domains select the best implementation for the data type and ops
    - e.g. use offload if and only if NIC implements **all** the requested ops
- Example: non-blocking atomic fetch-and-add (FADD) on unsigned 64-bit integer

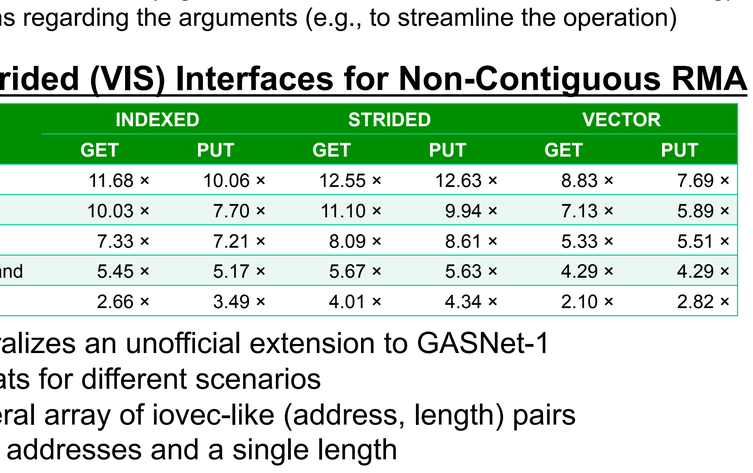
```
gex_Event_t ev = // *result = ATOMICALLY( *target += addend )
gex_AD_OpNB_U64(domain, &result, target_rank, target_address,
                GEX_OP_FADD, addend, 0 /*unused op2*/, flags);
```

- **`flags`** includes optional behaviors and assertions, such as memory fences
- GASNet-EX provides a network-independent "reference implementation"
  - Uses Active Messages to perform operations using the target CPU
  - Uses GASNet-Tools for atomicity (inline assembly for numerous CPUs)
- Specialization for Cray Aries improves performance vs. reference implementation
  - Reduces latency of inter-node FADD from 4.9us to 2.8us
  - Greatly increases throughput under contention
- The figure above shows throughput of 1 to 8192 processes (64 per node) performing pipelined FADD of a central counter (measured on ALCF's Theta).

GASNet-EX Performance on Cray Aries

ALCF Theta

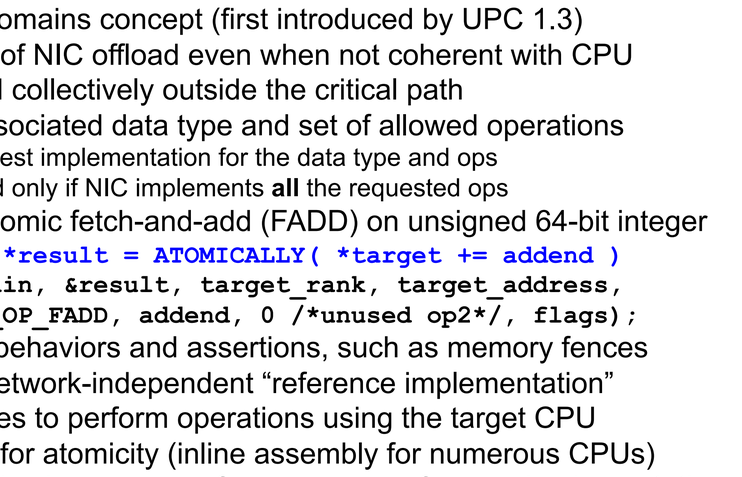
- Cray XC-40
- Cray Aries network
- GASNet-EX aries-conduit
- Cray MPICH 7.7.0
- Node configuration
  - 64-core 1.3GHz Intel Xeon Phi 7230
  - 192 GB of DDR
  - Quad/cache mode
- Intel C Compiler, v18.0.0.128
- System software
  - Cray PrgEnv-intel/6.0.4
  - Cray PE/2.5.13



GASNet-EX Performance on InfiniBand

OLCF SummitDev (single-rail only)

- IBM S822LC
- Mellanox InfiniBand EDR
- GASNet-EX ibv-conduit
- IBM Spectrum MPI 10.1.0.4
- Node configuration
  - 2x 10-core 3.5GHz IBM POWER8
  - 4x NVIDIA Tesla P100 GPU
  - 256 GB DDR4
- IBM XL C/C++ for Linux, V13.1.5 (5725-C73, 5765-J08)
- System software
  - Linux 3.10.0-514.21.2.el7.ppc64le
  - libibverbs 1.2.1mlnx1
  - IB f/w 12.17.1016



Flood Bandwidth Graphs report tests of achievable one-way bandwidth for point-to-point data transfer between two nodes.
All bandwidths have been converted for uniform reporting in units of Gibibytes/sec (GiB/sec), where GiB = 2^{30} bytes.

- GASNet-EX: uni-directional non-blocking put flood bandwidth — `testlarge -in 10000 [MAXSZ]`
- NetPIPE v3.7.2: uni-directional stream Send/Recv test — `Npmpi -s -n 10000 -u [MAXSZ]`
- OSU Benchmarks v5.3: test osu_bw, uni-directional Isend/Irecv flood bandwidth — `osu_bw -i 10000 -m [MAXSZ]`