UCX: An Open Source Framework for HPC Network APIs and Beyond

Pavel Shamis (Pasha)
Principal Research Engineer
Co-Design Collaboration

The Next Generation

HPC Communication Framework

Collaborative Effort

Industry, National Laboratories and Academia
Challenges

- **Performance Portability** (across various interconnects)
  - Collaboration between industry and research institutions
    - …but mostly industry (because they built the hardware)

- **Maintenance**
  - Maintaining a network stack is time consuming and expensive
  - Industry have resources and strategic interest for this

- **Extendibility**
  - MPI+X+Y ?
  - Exascale programming environment is an ongoing debate
UCX – Unified Communication X Framework

- Unified
  - Network API for multiple network architectures that target HPC programming models and libraries
- Communication
  - How to move data from location in memory A to location in memory B considering multiple types of memories
- Framework
  - A collection of libraries and utilities for HPC network programmers
### History

**MXM**
- Developed by Mellanox Technologies
- HPC communication library for InfiniBand devices and shared memory
- Primary focus: MPI, PGAS

**PAMI**
- Developed by IBM on BG/Q, PERCS, IB VERBS
- Network devices and shared memory
- MPI, OpenSHMEM, PGAS, CHARM++, X10
- C++ components
- Aggressive multi-threading with contexts
- Active Messages
- Non-blocking collectives with hardware acceleration support

**UCCS**
- Developed by ORNL, UH, UTK
- Originally based on Open MPI BTL and OPAL layers
- HPC communication library for InfiniBand, Cray Gemini/Aries, and shared memory
- Primary focus: OpenSHMEM, PGAS
- Also supports: MPI

---

**Decades of community and industry experience in development of HPC software**
What we are doing differently…

- UCX **consolidates** multiple industry and academic efforts
  - Mellanox MXM, IBM PAMI, ORNL/UTK/UH UCCS, etc.
- Supported and maintained by industry
  - IBM, Mellanox, NVIDIA, Pathscale, ARM
What we are doing differently...

- Co-design effort between national laboratories, academia, and industry

Applications: LAMMPS, NWChem, etc.

Programming models: MPI, PGAS/Gasnet, etc.

Middleware:

Driver and Hardware
A Collaboration Efforts

- Mellanox co-designs network API and contributes MXM technology
  - Infrastructure, transport, shared memory, protocols, integration with OpenMPI/SHMEM, MPICH
- ORNL & LANL co-designs network API and contributes UCCS project
  - InfiniBand optimizations, Cray devices, shared memory
- ARM co-designs the network API and contributes optimizations for ARM eco-system
- NVIDIA co-designs high-quality support for GPU devices
  - GPUDirect, GDR copy, etc.
- IBM co-designs network API and contributes ideas and concepts from PAMI
- UH/UTK focus on integration with their research platforms
Licensing

- Open Source
  - BSD 3 Clause license
  - Contributor License Agreement – BSD 3 based
UCX Framework Mission

- Collaboration between industry, laboratories, and academia
- Create open-source production grade communication framework for HPC applications
- Enable the highest performance through co-design of software-hardware interfaces
- Unify industry - national laboratories - academia efforts

### API
Exposes broad semantics that target data centric and HPC programming models and applications

### Performance oriented
Optimization for low-software overheads in communication path allows near native-level performance

### Production quality
Developed, maintained, tested, and used by industry and researcher community

### Community driven
Collaboration between industry, laboratories, and academia

### Research
The framework concepts and ideas are driven by research in academia, laboratories, and industry

### Cross platform
Support for Infiniband, Cray, various shared memory (x86-64 and Power), GPUs

Co-design of Exascale Network APIs
Architecture
UCX Framework

**UC-P for Protocols**
High-level API uses UCT framework to construct protocols commonly found in applications

**Functionality:**
Multi-rail, device selection, pending queue, rendezvous, tag-matching, software-atomics, etc.

**UC-T for Transport**
Low-level API that expose basic network operations supported by underlying hardware. Reliable, out-of-order delivery.

**Functionality:**
Setup and instantiation of communication operations.

**UC-S for Services**
This framework provides basic infrastructure for component based programming, memory management, and useful system utilities

**Functionality:**
Platform abstractions, data structures, debug facilities.
A High-level Overview

UC-T (Hardware Transports) - Low Level API
- Low Level API
  - RMA, Atomic, Tag-matching, Send/Recv, Active Message
  - Transport for InfiniBand VERBs
  - Driver
  - RC
  - UD
  - XRC
  - DCT
  - Transport for Gemini/Aries drivers
  - GNI
  - SYSV
  - POSIX
  - KNEM
  - CMA
  - XPMEM
  - Transport for Accelerator Memory communication
  - GPU

UC-P (Protocols) - High Level API
- High Level API
  - Transport selection, cross-transport multi-rail, fragmentation, operations not supported by hardware
  - Message Passing API Domain: tag matching, rendezvous
  - PGAS API Domain: RMAs, Atomics
  - Task Based API Domain: Active Messages
  - I/O API Domain: Stream

UC-S (Services)
- Services
  - Common utilities
  - Utilities
  - Data structures
  - Memory Management

Applications
- Applications
  - MPICH, Open-MPI, etc.
  - OpenSHMEM, UPC, CAF, X10, Chapel, etc.
  - Parsec, OCR, Legions, etc.
  - Burst buffer, ADIOS, etc.

Hardware
- Hardware
  - OFA Verbs Driver
  - Cray Driver
  - OS Kernel
  - CUDA

Utilities
- Utilities
  - Burst buffer
  - ADIOS

Data structures
- Data structures
  - ACES

Memory Management
- Memory Management
  - Memory Management

Hardware
- Hardware
  - ACES

Applications
- Applications
  - MPICH, Open-MPI, etc.
  - OpenSHMEM, UPC, CAF, X10, Chapel, etc.
  - Parsec, OCR, Legions, etc.
  - Burst buffer, ADIOS, etc.

UCX
- UCX
  - Message Passing API Domain: tag matching, rendezvous
  - PGAS API Domain: RMAs, Atomics
  - Task Based API Domain: Active Messages
  - I/O API Domain: Stream

Transport for InfiniBand VERBs driver
- Transport for InfiniBand VERBs driver
  - RC
  - UD
  - XRC
  - DCT
  - Transport for Gemini/Aries drivers
  - GNI
  - SYSV
  - POSIX
  - KNEM
  - CMA
  - XPMEM
  - Transport for Accelerator Memory communication
  - GPU
UCP API (DRAFT) Snippet
(https://github.com/openucx/ucx/blob/master/src/ucp/api/ucp.h)

- `ucs_status_t ucp_put(ucp_ep_h ep, const void *buffer, size_t length, uint64_t remote_addr, ucp_rkey_h rkey)`
  Blocking remote memory put operation.

- `ucs_status_t ucp_put_nbi (ucp_ep_h ep, const void *buffer, size_t length, uint64_t remote_addr, ucp_rkey_h rkey)`
  Non-blocking implicit remote memory put operation.

- `ucs_status_t ucp_get (ucp_ep_h ep, void *buffer, size_t length, uint64_t remote_addr, ucp_rkey_h rkey)`
  Blocking remote memory get operation.

- `ucs_status_t ucp_get_nbi (ucp_ep_h ep, void *buffer, size_t length, uint64_t remote_addr, ucp_rkey_h rkey)`
  Non-blocking implicit remote memory get operation.

- `ucs_status_t ucp_atomic_add32 (ucp_ep_h ep, uint32_t add, uint64_t remote_addr, ucp_rkey_h rkey)`
  Blocking atomic add operation for 32 bit integers.

- `ucs_status_t ucp_atomic_add64 (ucp_ep_h ep, uint64_t add, uint64_t remote_addr, ucp_rkey_h rkey)`
  Blocking atomic add operation for 64 bit integers.

- `ucs_status_t ucp_atomic_fadd32 (ucp_ep_h ep, uint32_t add, uint64_t remote_addr, ucp_rkey_h rkey, uint32_t *result)`
  Blocking atomic fetch and add operation for 32 bit integers.

- `ucs_status_t ucp_atomic_fadd64 (ucp_ep_h ep, uint64_t add, uint64_t remote_addr, ucp_rkey_h rkey, uint64_t *result)`
  Blocking atomic fetch and add operation for 64 bit integers.

- `ucs_status_ptr_t ucp_tag_send_nb (ucp_ep_h ep, const void *buffer, size_t count, ucp_datatype_t datatype, ucp_tag_t tag, ucp_send_callback_t cb)`
  Non-blocking tagged-send operations.

- `ucs_status_ptr_t ucp_tag_recv_nb (ucp_worker_h worker, void *buffer, size_t count, ucp_datatype_t datatype, ucp_tag_t tag, ucp_tag_t tag_mask, ucp_tag_recv_callback_t cb)`
  Non-blocking tagged-receive operation.
Preliminary Evaluation (UCT)


- Two HP ProLiant DL380p Gen8 servers
- Mellanox SX6036 switch, Single-port Mellanox Connect-IB FDR (10.10.5056)
- Mellanox OFED 2.4-1.0.4. (VERBS)
- Prototype implementation of Accelerated VERBS (AVERBS)
OpenSHMEM and OSHMEM (OpenMPI) Put Latency (shared memory)

Lower is better

Slide courtesy of ORNL UCX Team
OpenSHMEM and OSHMEM (OpenMPI)

Put Injection Rate

Higher is better

Connect-IB

Message Rate (put operations / second)

Message Size

Slide courtesy of ORNL UCX Team
OpenSHMEM and OSHMEM (OpenMPI) GUPs Benchmark

### Key Points
- **Higher is better**
- **Connect-IB**

#### Graph Details
- **Y-axis**: GUPS (billion updates per second)
- **X-axis**: Number of PEs (two nodes)
- **Lines**:
  - UCX (mlx5)
  - OSHMEM (mlx5)

**Slide courtesy of ORNL UCX Team**
MPICH - Message rate
Preliminary Results

Connect-IB

“non-blocking tag-send”

MPICH/UCX  MPICH/MXM

Slide courtesy of Pavan Balaji, ANL - sent to the ucx mailing list
Where is UCX being used?

- Upcoming release of Open MPI 2.0 (MPI and OpenSHMEM APIs)
- Upcoming release of MPICH
- OpenSHMEM reference implementation by UH and ORNL
- PARSEC – runtime used on Scientific Linear Libraries
What Next?

- UCX Consortium!
  - http://www.csm.ornl.gov/newsite/

- UCX Specification
  - Early draft is available online:
    http://www.openucx.org/early-draft-of-ucx-specification-is-here/

- Production releases
  - MPICH, Open MPI, Open SHMEM(s), Gasnet, and more…

- Support for more networks and applications and libraries

- UCX Hackathon 2016!
  - Will be announced on the mailing list and website
https://github.com/orgs/openucx

WEB: www.openucx.org
Contact: info@openucx.org

Mailing List:
https://elist.ornl.gov/mailman/listinfo/ucx-group
ucx-group@elist.ornl.gov
Questions?

Unified Communication - X Framework

WEB: www.openucx.org
Contact: info@openucx.org
WEB: https://github.com/orgs/openucx

Mailing List:
https://elist.ornl.gov/mailman/listinfo/ucx-group
ucx-group@elist.ornl.gov