



UCF

Unified
Communication
Framework



UCX State of the Union

Pavel (Pasha) Shamis

2022



Open-source framework for high-performance networks

- UCP extensible datapath API – “nbx”
 - Mandatory parameters – passed as regular arguments
 - Optional/extended parameters – passed in a struct “ucp_request_param_t”
 - Allows future extensions without ABI/API breakage, and still enjoy fastcall
- UCP active messages – generic high-level communication primitive
 - Added rendezvous protocol, which uses zero-copy RDMA for bulk transfer
 - Support GPU memory for all operation types

```
ucs_status_ptr_t ucp_am_send_nbx(ucp_ep_h ep, unsigned id,  
..... const void *header, size_t header_length,  
..... const void *buffer, size_t count,  
..... const ucp_request_param_t *param);
```

- New client-server connection protocol
 - Quick one-sided disconnect with remote notification (like TCP)
 - Multi-device and multi-path
 - Revamp RDMA_CM and TCP connection managers for better stability

- GPU support improvements
 - Select NIC according to GPU locality on the PCIe bus
 - Support statically-linked Cuda applications
 - Global cache for Cuda IPC remote memory handles

- Error handling improvements
 - Keepalive on UCP layer to detect stale peers
 - Auto-revoke all queued requests when connection is closed

- Global configuration file to set UCX parameters
- Shared memory to support asynchronous wakeup
- UD performance optimizations
- Java bindings – full support for UCP API
- Support RDMA atomics on GPU memory target
- FUSE virtual filesystem for monitor and statistics of UCX runtime
- Support setting UCT/UCS parameters through UCP API

- Release schedule:
 - 1.12.0 - November 2021
 - 1.13.0 - March 2022
 - 1.14.0 - July 2022
 - 1.15.0 - November 2022
- Wire protocol compatibility
- EFA SRD support for AWS systems
- Static library support
- GPU protocols improvements and linux DMAbuf support
- Rendezvous protocol with scatter-gather lists
- Golang bindings
- UCP active message improvements
 - Set receive buffer alignment
 - Fragmented receive protocol
- One-sided improvements:
 - Support GPU atomic operations (both on source and target)
 - Multi rail and out-of-order with PUT/FENCE

