OPENFABRICS ALLIANCE

UCX

Unified Communication - X Framework

Background

MXM

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

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

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 hw accleration support

Introduction

UCX - Collaboration between industry, laboratories, and academia to create open-source production grade communication framework for data centric and HPC applications

Goals

Performance oriented

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

Community driven

Collaboration between industry, laboratories, and academia

Production quality

Developed, maintained, tested, and used by industry and researcher community

API

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

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

Collaboration

- Mellanox co-designs network interface and contributes MXM technology
  - Infrastructure, UD, RC, DCT, shared memory, protocols, integration with OpenMPI/SHMEM, MPICH
- ORNL co-designs network interface and contributes UCCS project
  - IB optimizations, Crays devices, shared memory
- NVIDIA co-designs high-quality support for GPU devices
  - GPU-Direct, GDR copy, etc.
- IBM co-designs network interface and contributes ideas and concepts from PAMI
- UH/UTK focus on integration with their research platforms

The Framework

UC-S for Services

This framework provides basic infrastructure for component based programming, memory management, and useful system utilities

Functionality:
Platform abstractions and data structures

UC-T for Transport

Low-level API that expose basic network operations supported by underlying hardware

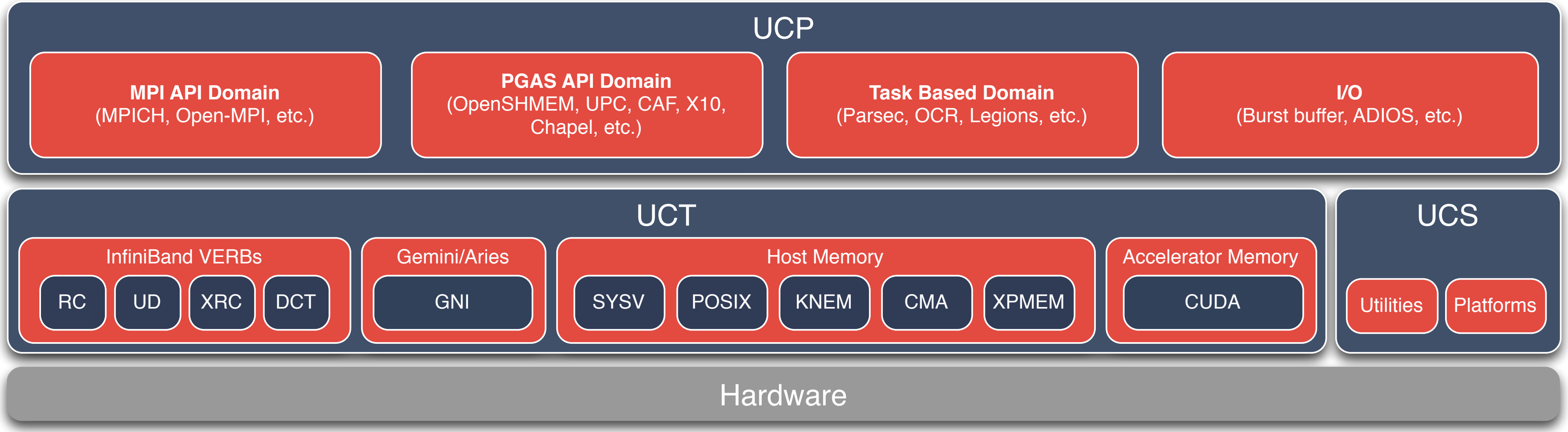
Functionality:
work request setup and instantiation of operations

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.

High-level Overview



Clarifications

- UCX is NOT a driver
- Responsibility of hardware driver

  Close-to-hardware API layer (defined by hardware specification) providing an access to hardware’s capabilities

- UCX relies on drivers supplied by vendors

  InfiniBand Verbs, Accelerated Verbs

  Libfabrics

  Cray GNI, etc.

Priorities

- Performance, performance, performance...
- Production grade software
- Enabling programming models and languages beyond Message Passing
    - PGAS libraries and languages, task-based programming models (OCR, Legions, Parsec, etc)
    - Data analytics and processing (ADIOS)
    - I/O

Project Management

- Hosted on GitHUB
- One/Two maintainers per organization
- Googletest testing environment
- Changes are accepted only through Pull Requests (PR) - **NO EXCEPTIONS**
  - All PRs are tested
  - Jenkins (Mellanox), Buildbot (ORNL) hooked up with GitHUB

Project Management

- API definitions and changes are discussed within developers (mail-list, github)
- PRs with API changes have to be approved by ALL maintainers
- PR within maintainer “domain” has to be reviewed by the maintainer or team member (Example: Mellanox reviews all IB changes)

Integration

- UCX will be integrated with major MPI distributions, OpenSHMEM, PGAS languages, etc.
- UCX as a research vehicle for upcoming runtimes, programming models, and I/O libraries

Licensing

- BSD 3 Clause license
- Contributor License Agreement – BSD 3 based