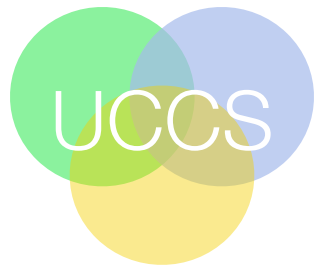


Oak Ridge National Laboratory

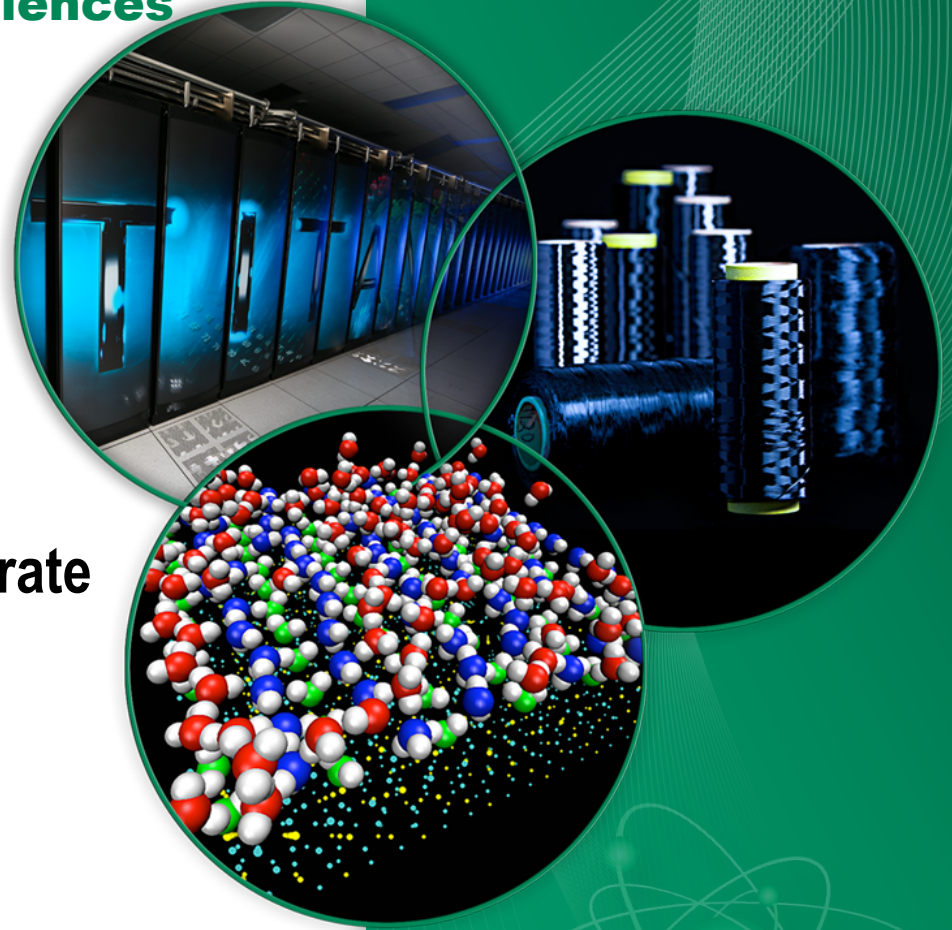
Computing and Computational Sciences



UCCS

Universal Common
Communication Substrate

Presented by:
Pavel Shamis (Pasha)



The Team

- **ORNL**

- Pavel Shamis / Pasha
- Manjunath Gorentla Venkata / Manju
- Oscar Hernandez
- Stephen Poole
- Tommy Janjusic
- Swen Boehm
- Douglas Fuller

- **UH**

- Tony Curtis
- Donald Aaron Welch
- Swaroop Pophale
- Siddharta Jana

- **UTK**

- George Bosilca
- Thomas Herault
- Aurélien Bouteiller

- **LANL**

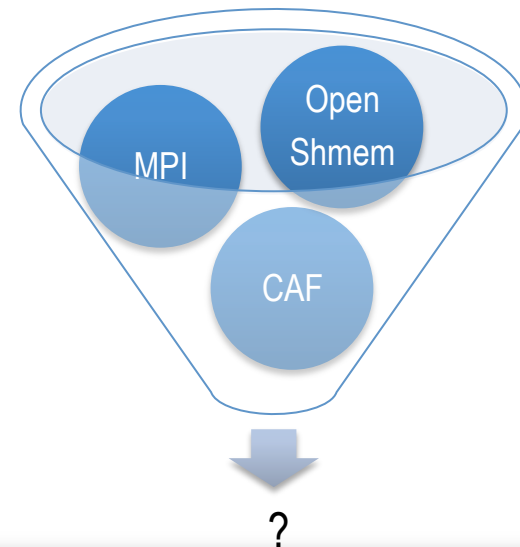
- Ginger Young

- **DoD**

- Nick P.
- Kevin B

History

- “*Déjà vu*” of OpenSHMEM implementer
 - We have seen this network code somewhere ?
 - A lot of similarity in initialization and communication flow ?
 - Critical-path flow are similar but not identical
- ULPs can have a high degree of overlap in the requirements they place on the lower level network layers
 - Communication interface can have a high degree of overlap in communication semantics
 - Send/Recv, AM, RDMA, AMO, Collectives, etc.

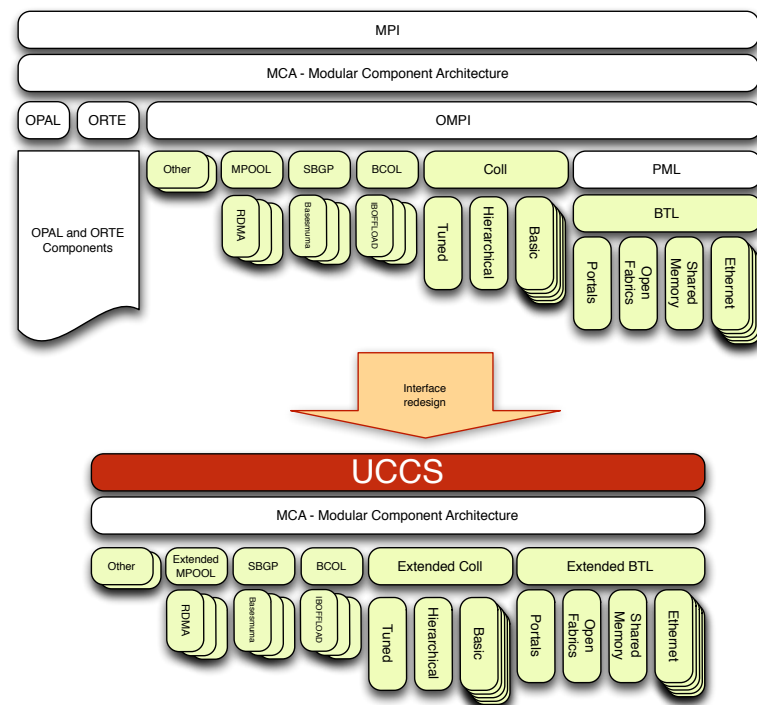


History - continued

- Idea of re-using high performance communication codes has been around for while:
 - ONET (~2009) – Rich Graham & Steve Pool
 - OpenSHMEM / “Yoda” (~2010) - Mellanox & ORNL collaboration

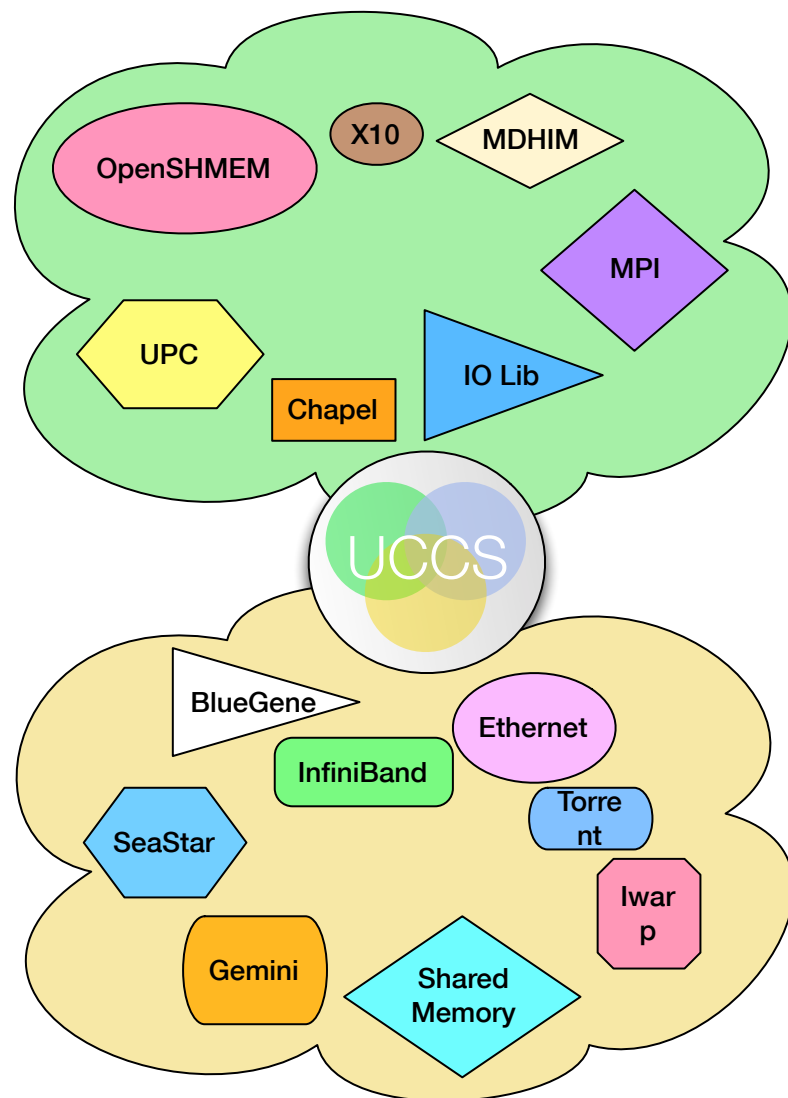
History - continued

- Universal Common Communication Substrate (UCCS) – Beginning...
 - Let's re-use internal MPI network codes and expertise to design a standalone communication middleware that serves broader HPC community with an initial focus OpenSHMEM/PGAS (but not only...)
- In addition to high-performance implementation we want to standardize the API



Goals

- Provide a common low-level scalable, robust, portable, simple and performance driven communication API for multiple parallel programming models over modern network interfaces
- Increasing code reusability and reducing development effort
- Include performance/power measurement capabilities in a central location

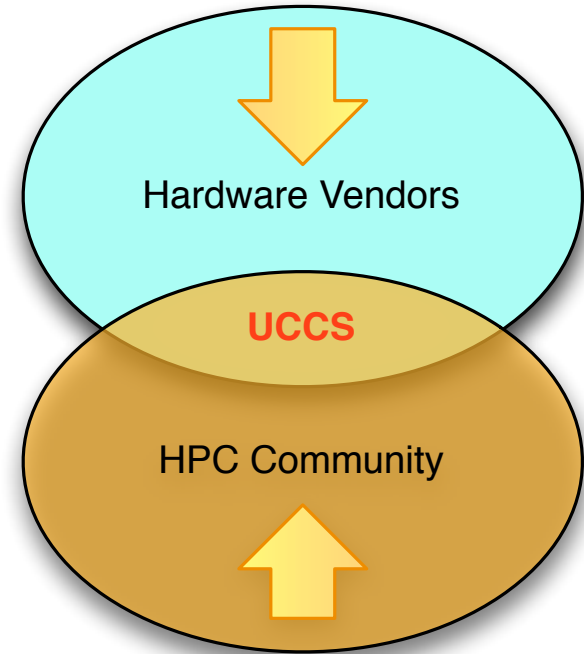


Goals - Continued

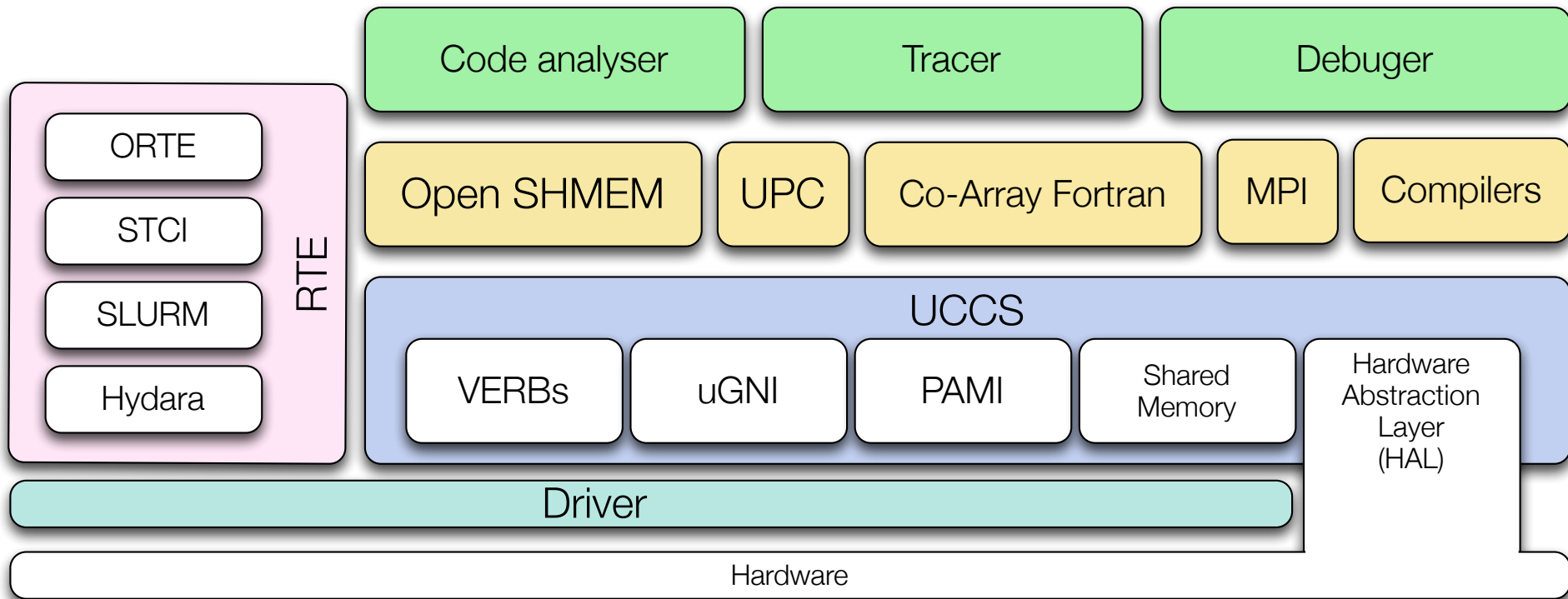
- Support hybrid programming environments efficiently
- Provide flexible API to accommodate requirements of I/O systems, Big Data applications, and Languages
- Runtime support for multiple network technologies (when possible)
- Provide and an interface for code translation (CAF, UPC, etc)
- Performance
- Define specification describing the communication middleware

Long Term Goals

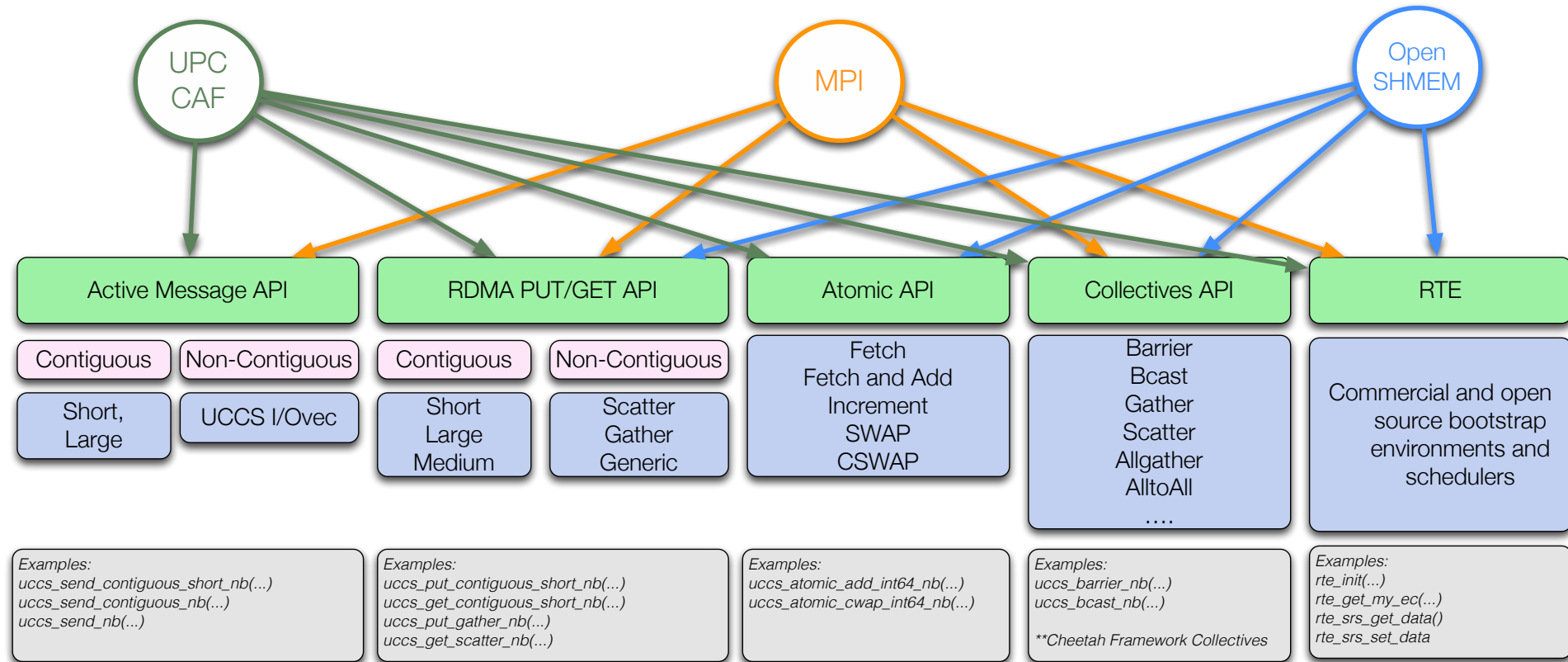
- Direct network hardware support
- Co-design
 - Hardware
 - Compilers
- Community support



Overview



UCCS API



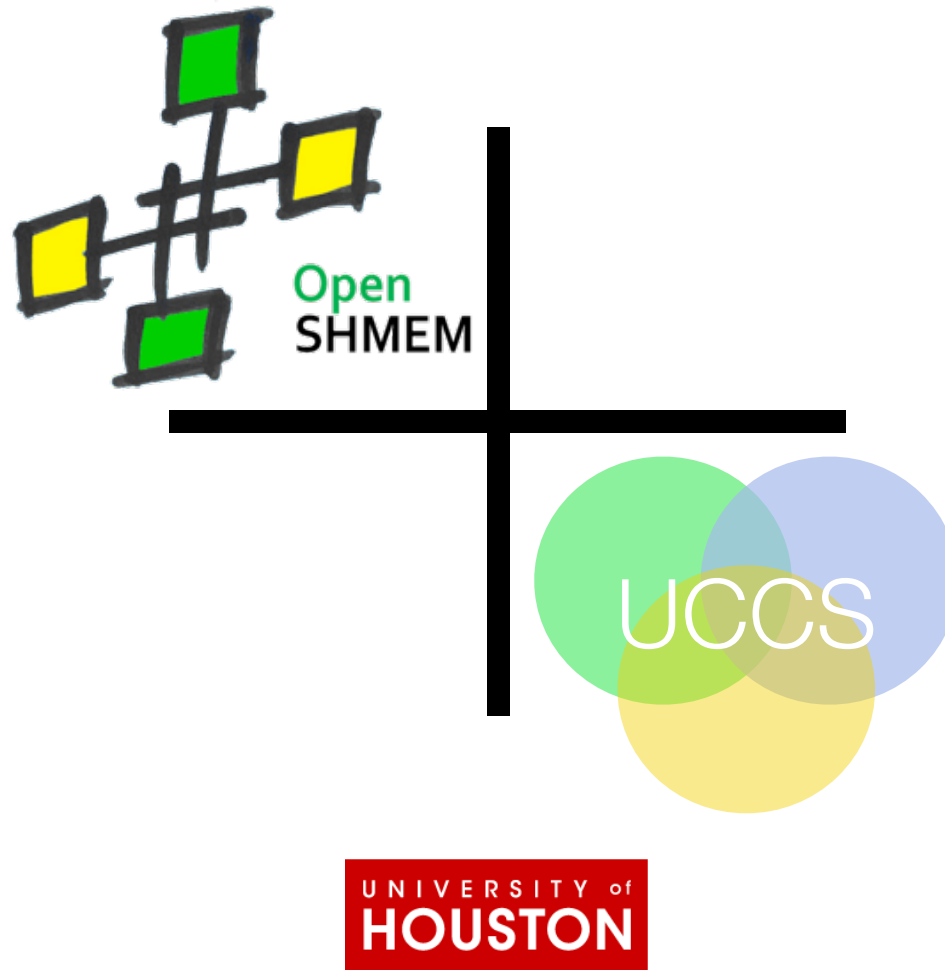
Implementation Details

- Initial code was based on Byte Transfer Layer (BTL)....
 - But we had to rewrite most of the critical path
- Modular Component Architecture (MCA)
 - Based on Open MPI MCA, which is essentially dynamically loaded libraries/components
 - Available as a standalone library: <http://uccs.github.io/libocoms/>
 - OCOMS – Open Component Module Service

Implementation detail - continued

- Runtime Environment Abstraction – libRTE
 - A standalone Abstraction for Runtime environments
 - STCI, ORTE, ALPS, SLURM
 - Will be available soon: <http://uccs.github.io/librte/>
- Collectives - <http://www.csm.ornl.gov/cheetah/>
 - Work in progress
- We don't like “bundles”
- UCCS Specifications v0.1, v0.2, and v0.3 work in progress
- Supported networks (pre-production): Infiniband , Cray.

OpenSHMEM and UCCS

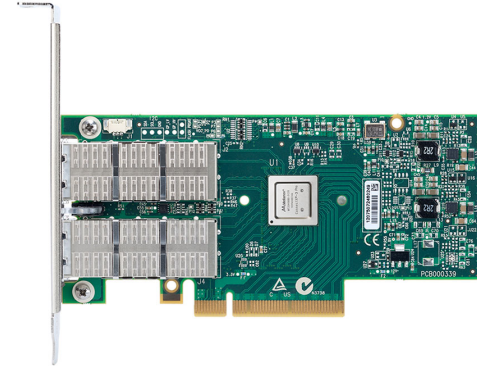


Update

- OpenSHMEM Reference Implementation
 - Internal interfaces were extended to support UCCS and libRTE and continue to support GASnet
 - UCCS is used as development platform for future OpenSHMEM extensions and research
 - Non-blocking communication
 - Extended network operations
 - Collectives, etc.
- OpenSHMEM-UCCS pre-production version is used internally for extensions evaluation and application development

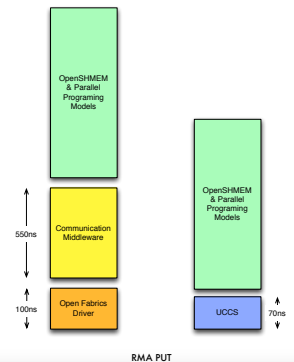
InfiniBand

- Mellanox Connect-X and Connect-IB HCAs provide technology enabling efficient and high-performance implementation of OpenSHMEM and UCCS
 - Low software overhead
 - Hardware Offload



Mellanox InfiniBand	UCCS	OpenSHMEM
RDMA	V	V
AMO	V	V
Collectives/CORE-Direct	V	V

- UCCS provides experimental user-level VERBS bypass mode for Mellanox Connect-X devices



UTK slides starts here

We are open for colaboration

<http://uccs.github.io/uccs>
uccs-info@ornl.gov



Acknowledgements



This work was supported by the United States Department of Defense & used resources of the Extreme Scale Systems Center at Oak Ridge National Laboratory.

Questions ?

