Omni-Path and the Open Fabrics Interfaces

Brian Smith
Director of Technology
www.cornelisnetworks.com

Notices and Disclaimers

INFORMATION IN THIS DOCUMENT IS PROVIDED IN CONNECTION WITH CORNELIS NETWORKS PRODUCTS. NO LICENSE, EXPRESS OR IMPLIED, BY ESTOPPEL OR OTHERWISE, TO ANY INTELLECTUAL PROPERTY RIGHTS IS GRANTED BY THIS DOCUMENT. EXCEPT AS PROVIDED IN CORNELIS NETWORKS'S TERMS AND CONDITIONS OF SALE FOR SUCH PRODUCTS, CORNELIS NETWORKS ASSUMES NO LIABILITY WHATSOEVER, AND CORNELIS NETWORKS DISCLAIMS ANY EXPRESS OR IMPLIED WARRANTY, RELATING TO SALE AND/OR USE OF CORNELIS NETWORKS PRODUCTS INCLUDING LIABILITY OR WARRANTIES RELATING TO FITNESS FOR A PARTICULAR PURPOSE, MERCHANTABILITY, OR INFRINGEMENT OF ANY PATENT, COPYRIGHT OR OTHER INTELLECTUAL PROPERTY RIGHT. CORNELIS NETWORKS PRODUCTS ARE NOT INTENDED FOR USE IN MEDICAL, LIFE SAVING, OR LIFE SUSTAINING APPLICATIONS.

Cornelis Networks may make changes to specifications and product descriptions at any time, without notice. Designers must not rely on the absence or characteristics of any features or instructions marked "reserved" or "undefined". Cornelis Networks reserves these for future definition and shall have no responsibility whatsoever for conflicts or incompatibilities arising from future changes to them. The information here is subject to change without notice. Do not finalize a design with this information.

All products, dates, and figures specified are preliminary based on current expectations, and are subject to change without notice. Roadmap not reflective of exact launch granularity and timing. The products described in this document may contain design defects or errors known as errata which may cause the product to deviate from published specifications. Current characterized errata are available on request.

Any code names featured are used internally within Cornelis Networks to identify products that are in development and not yet publicly announced for release. Customers, licensees and other third parties are not authorized by Cornelis Networks to use code names in advertising, promotion or marketing of any product or services and any such use of Cornelis Networks' internal code names is at the sole risk of the user.

All products, computer systems, dates and figures specified are preliminary based on current expectations and are subject to change without notice. Material in this presentation is intended as product positioning and not approved end user messaging.

Performance tests are measured using specific computer systems, components, software, operations, and functions. Any change to any of those factors may cause the results to vary. You should consult other information and performance tests to assist you in fully evaluating your contemplated purchases, including the performance of that product when combined with other products.

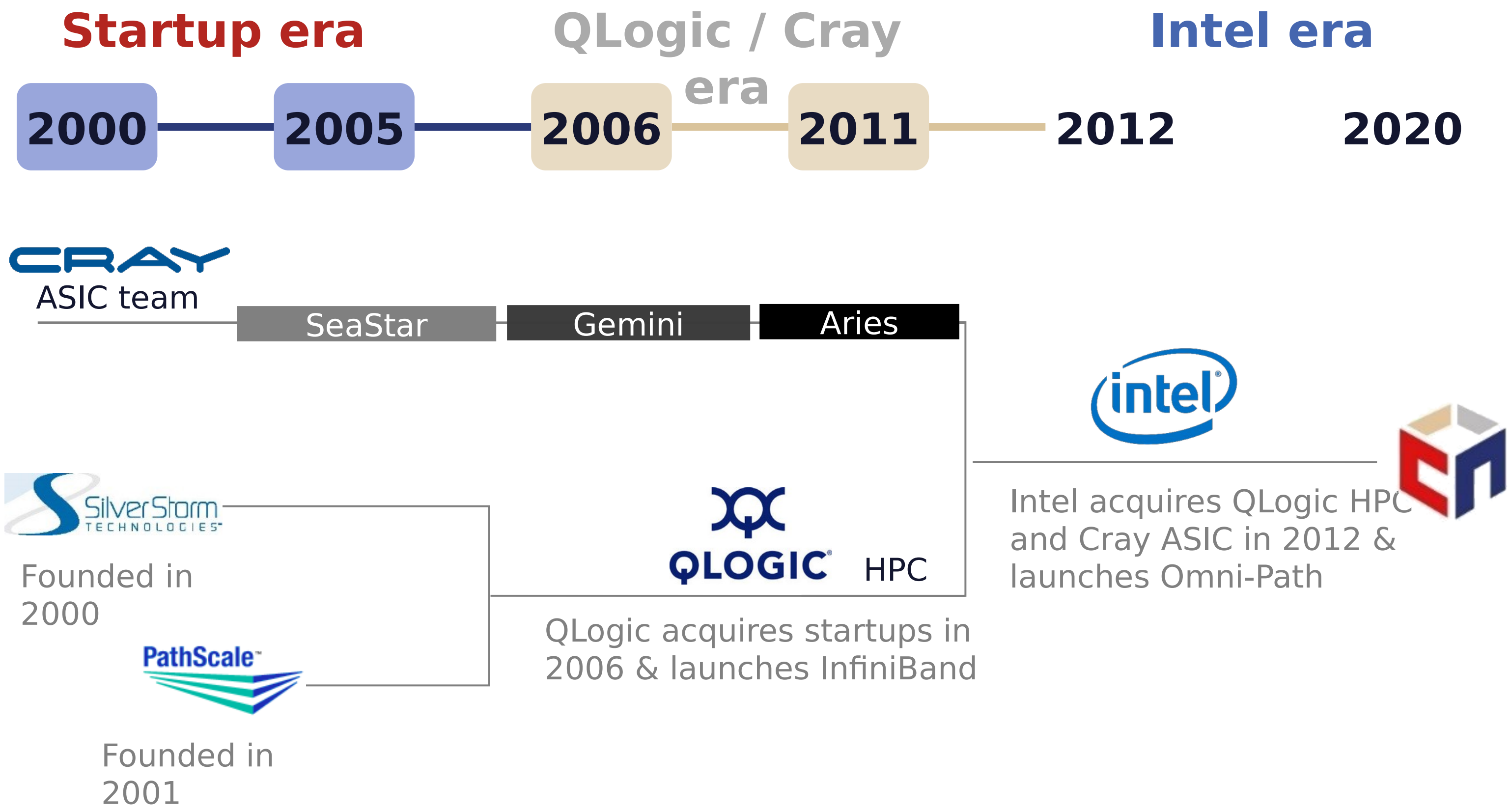
Cornelis Networks technologies' features and benefits depend on system configuration and may require enabled hardware, software, or service activation. Performance varies depending on system configuration.

Cornelis, Omni-Path, Omni-Path Express, and the Cornelis Networks logo belong to Cornelis Networks, Inc. Other names and brands may be claimed as the property of others.

Agenda

- Introduction
- OpenFabrics Alliance (OFA) and libfabric
  - History
  - Libfabric architecture
  - High-level comparison to Verbs and UCX
- Omni-Path and OpenFabrics
  - History – PSM2
  - Current - OPX
- Looking Ahead

Who we are



Startup era

QLogic / Cray era

Intel era

2000 2005 2006 2011 2012 2020

ASIC team

SeaStar

Gemini

Aries

Founded in 2000

Founded in 2001

HPC

QLogic acquires startups in 2006 & launches InfiniBand

Intel acquires QLogic HPC and Cray ASIC in 2012 & launches Omni-Path

Technology built on ~$1B investment over 20 years

Cornelis Networks

- ✓ Acquired Intel Omni-Path business
- ✓ Delivering complete networking solut
- ✓ Supporting 500+ global deployments
- ✓ Developing strong ecosystem suppor
- ✓ Serving Government, Academic, Scientific, & Commercial segments
- ✓ Enhancing Omni-Path solutions with next generation development

End-to-end Interconnect Solutions

Host and Management Software

OPENFABRICS ALLIANCE

Adapters

Edge Switches

Director Class Switches

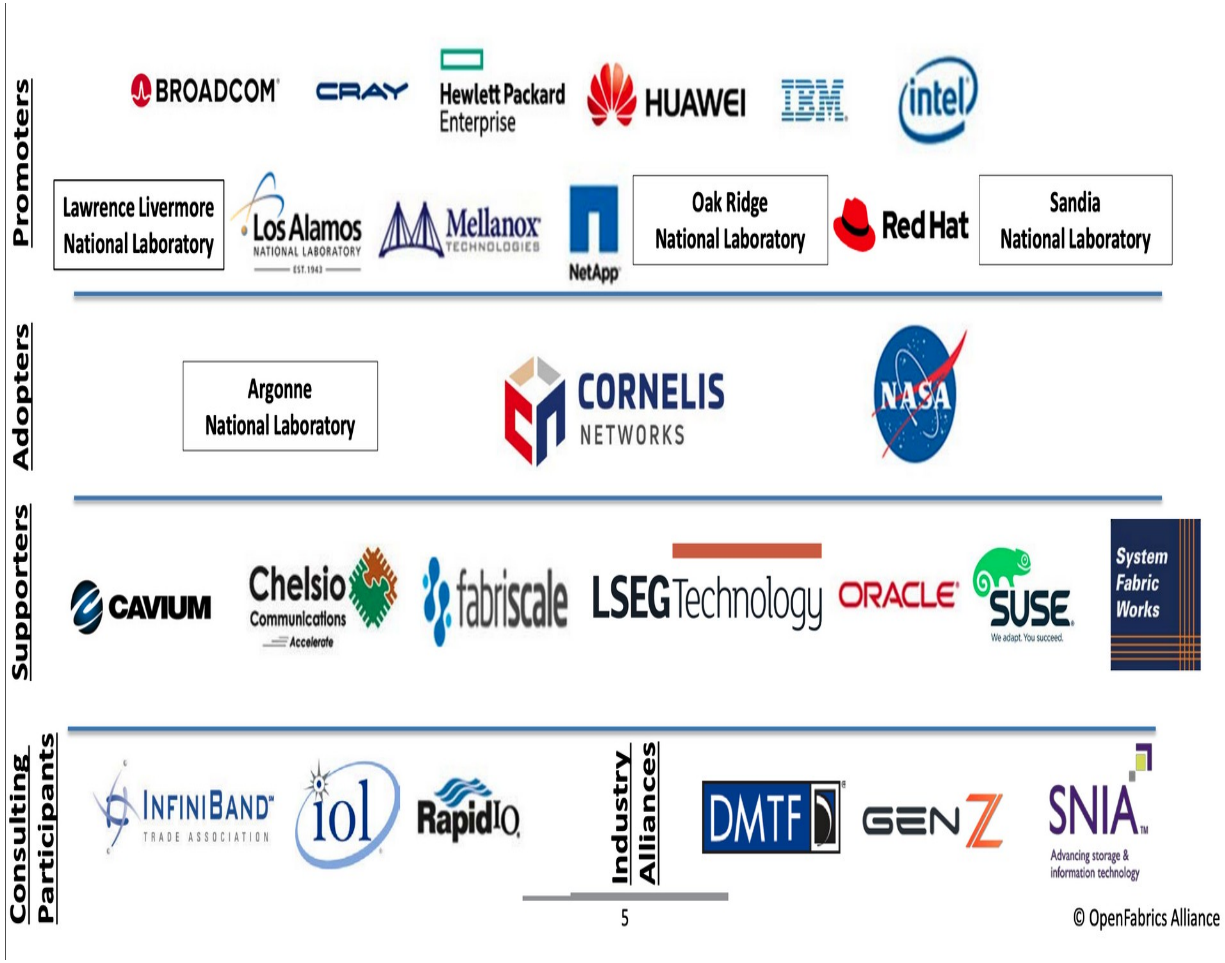
Copper and Optical Cables

InfiniBand & Ethernet Gateways

Fully Integrated, Open, and Interoperable

OpenFabrics Alliance (OFA) - Background

- Started in 2004
- Advances the development of open-source software for networking
  - Support and maintain existing fabric technologies
  - Develop for new emerging technologies and applications
- Works closely with other open-source communities to ensure adoption
  - Linux kernel, SNIA, DMTF
- Large body of promoters, adopters, and supporters


Promoters
BROADCOM
CRAY
Hewlett Packard Enterprise
HUAWEI
IBM
intel
Lawrence Livermore National Laboratory
Los Alamos NATIONAL LABORATORY EST.1943
Mellanox TECHNOLOGIES
NetApp
Oak Ridge National Laboratory
Red Hat
Sandia National Laboratory
Adopters
Argonne National Laboratory
CORNELIS NETWORKS
NASA
Supporters
CAVIUM
Chelsio Communications Accelerate
fabriscale
LSEG Technology
ORACLE
SUSE We adapt. You succeed.
System Fabric Works
Consulting Participants
INFINIBAND TRADE ASSOCIATION
iol
RapidIO
Industry Alliances
DMTF
GENZ
SNIA Advancing storage & information technology
5
© OpenFabrics Alliance

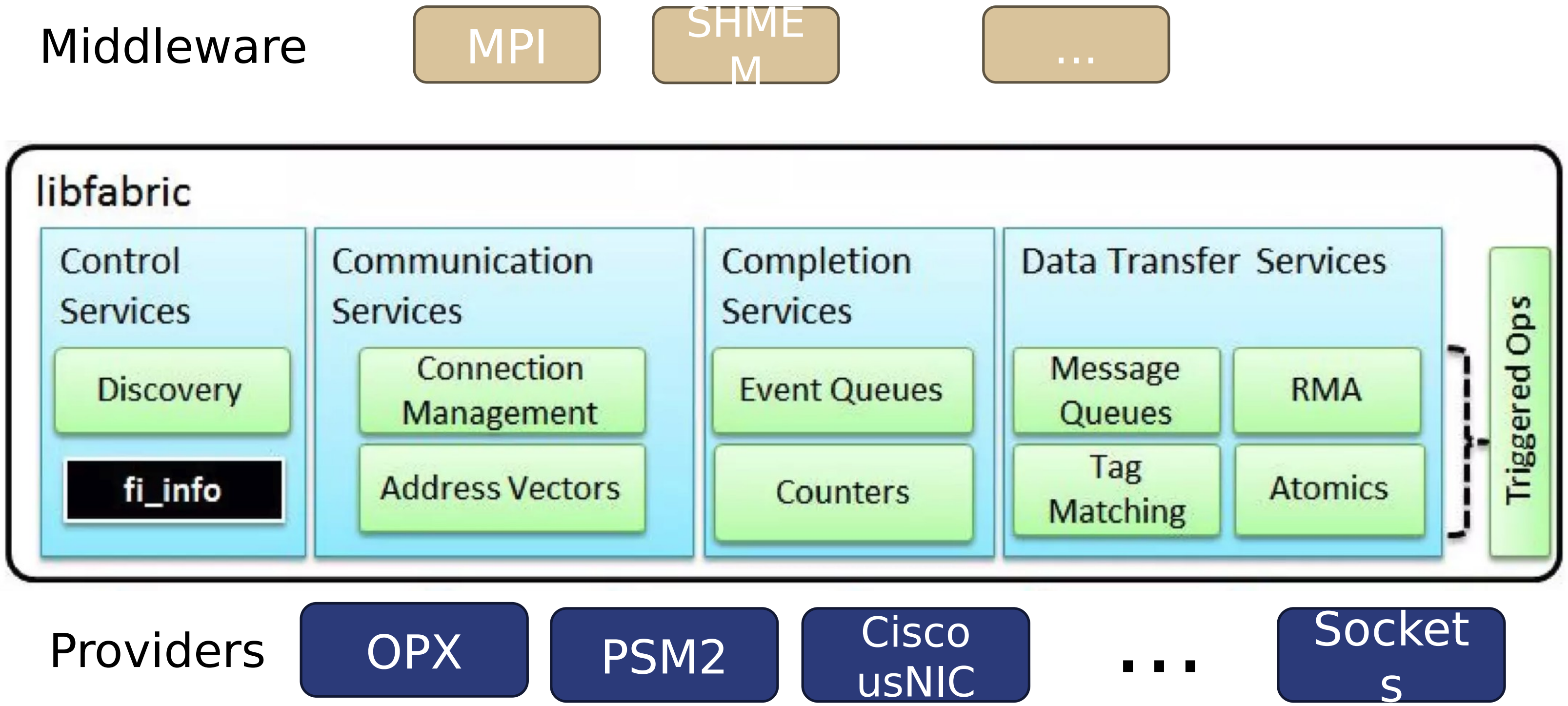
OFA Background (cont.)

- Consists of multiple working groups
  - OpenFabrics Management Framework Working Group (OFMFWG) develops management framework and interfaces
  - Marketing Working Group (MWG)
  - Fabric Software Development Platform Working Group (FSDPWG) – Works to ensure OFA is supported in the community
  - OpenFabrics Interfaces Working Group (OFIWG) develops high performance fabric interfaces (libfabric)
- OFA Annual Workshop
  - Typically in April; was in Columbus in ‘23
- Industry Alliance Program

OFIWG -- libfabric

- OFIWG Charter:
  - Develop an extensible open source framework and interfaces aligned with upper layer protocols and application needs for high-performance fabric services
- Maximize impedance match between ULPs and network APIs
  - Detailed analysis of MPI, SHMEM, and other programming paradigms to ensure APIs are well matched
  - Additional work with storage, AI/ML/DL, databases, etc
- Designed from the ground up to be scalable and high-performance
  - Scalable address resolution and storage via address vectors
  - Optimized software path to hardware
  - Agile development, frequent code releases
  - Application-centric
- Networks/hardware exposed via “providers”

Libfabric Architecture



Middleware

MPI

SHMEM

...

Providers

OPX

PSM2

Cisco usNIC

...

Sockets

Control Services

- Interrogate the fabric for capabilities it can provide or the application needs
- fi_getinfo()
  - Similar to getaddrinfo()
  - Capabilities and mode structures
- Capability bits are desired features and services requested by the application
  - FI_RMA, FI_TAGGED, FI_ATOMIC, FI_FENCE, etc
- Mode bits are used to convey requirements that an application must adhere to when using the given fabric
  - Usually related to hardware limitations
  - Size of context structure, buffered receives, keep header space available etc
  - Application may see improved performance if it implements a feature

Communication Services

- Setup communication between processes
- Multiple endpoint types
  - Connection-less, unreliable datagram
  - Reliable, connected
  - Reliable, unconnected
  - Scalable Endpoints – Consolidate hw resources in single sw resource
    - Improved threading performance and reduce memory
- Utilizes address vectors
  - Maps higher level addresses (e.g. MPI rank) to fabric specific addresses
  - FI_AV_MAP – 64b address type. Direct map to hardware typically
  - FI_AV_TABLE – Uses an index so minimal memory footprint, but requires lookup per message operation

Completion Services

- Asynchronous completion support
  - Counters
    - Lightweight completion mechanism for data transfers
  - Event queues
    - Report completion of asynchronous operations
  - Completion queues
    - High performance queues for data transfer completions
    - Optimized to report successful completions
  - Poll set
    - Providers using the host processor to progress data transfers via application thread

Data transfer Services

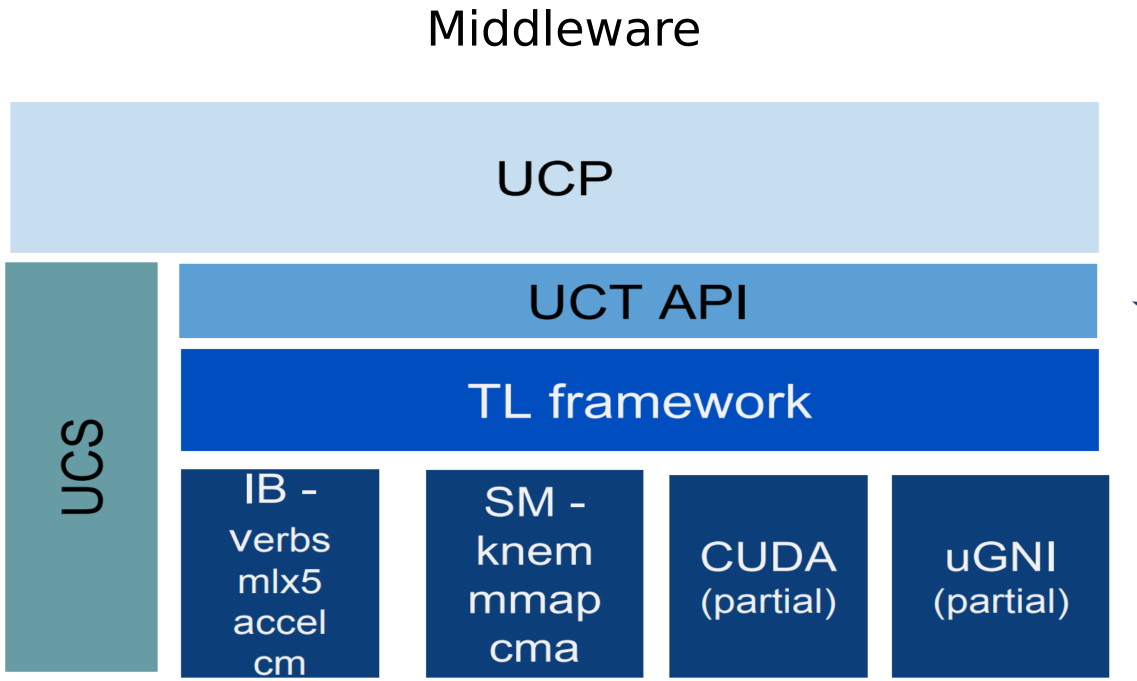
- Supports multiple communication paradigms
  - Message queues
  - RMA
  - Atomics
  - Tag matching

Libfabric vs Verbs

- Verbs API was not designed around HPC messaging
- Requires significant setup and memory for basic data movement
- Setup data structures, then use a generic send.
  - Loops over work requests, with multiple branches
  - Then, nested loop over SGEs, with multiple branches
- Hundreds of lines of code, most of it not directly related to hardware
- Libfabric has much cleaner interfaces – no loops and more predictable branches
  - Fewer lines of code, most of which are optimized paths to hardware

Libfabric vs UCX

- Unified Communication Framework (UCF) has very similar goals to OFA
  - Industry/partner support
  - Multiple working groups/projects – UCX, UCC, OpenSNAPI
  - Primarily used on IB and RoCE
- Unified Communication X (UCX) – similar to libfabric
  - Both provide functionality needed for HPC application spaces
  - UCX is point to point (UCC for collectives)



Libfabric vs UCX (Philosophical/Opinion)

- UCX is very efficient for constructing HPC primitives used by ULPs (e.g. MPI/SHMEM)
    - MPI tag matching, RDMA operations
    - Simpler API
    - Similar to internal mechanisms in OpenMPI/MPICH for interfacing with networks
    - Uses callbacks
- Libfabric is more application centric
    - Provides fabric communication services
    - More end point options – enables more use cases outside of traditional HPC spaces
    - Sockets provider allows running almost anywhere
    - Uses CQs or poll sets
    - Threadsafe by default early on
- Both provide very similar performance benefits

Omni-Path and OpenFabrics

- Omni-Path hardware originally utilized Performance Scale Messaging (PSM2) for messaging
  - Supported Intel MPI, OpenMPI, and NCCL
  - Shim wrapper to make PSM2 a provider in libfabric
- New efforts around OPX – Omni-Path eXpress
  - Originally based on BG/Q libfabric provider
  - Minimal instruction counts, highly memory efficient provider
  - Completely standalone provider
  - Utilizes existing hfi1 driver
  - 100% switchable in user-space

OPX vs PSM2 - Processing a Packet

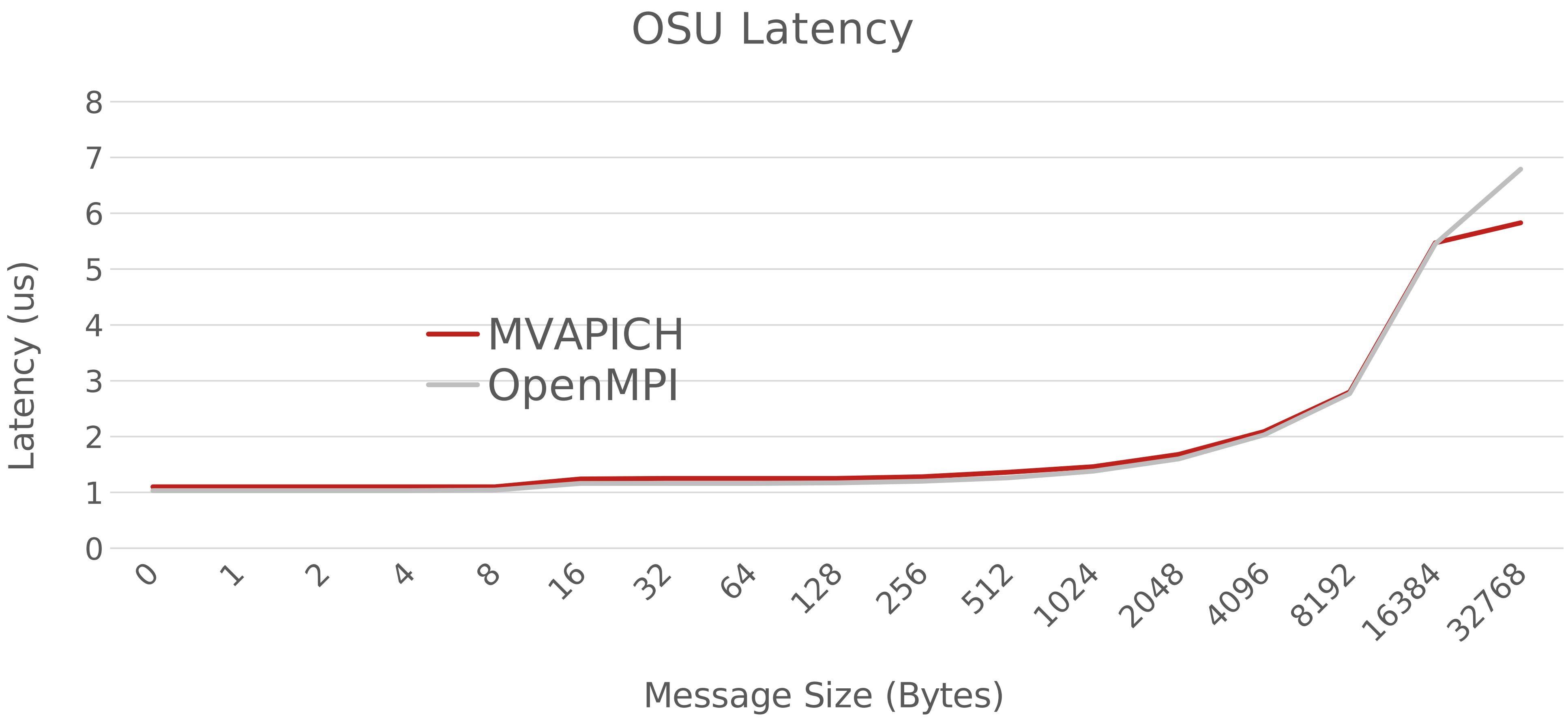
- Optimized incoming packet processing (Do a single MPI_Recv(...))
  - Intel SDE testing shows tremendous improvement in instruction count
  - Significant improvements in cache line footprint

	PSM2	OPX	Improvement
Instruction count	3064	1170	62%
Cache lines for code	205	124	40%
Cache line loads	93	55	41%
New cache line access	354	209	41%

OPX Provider

- Upstream-first, entirely open-source
- Optimal protocol and HW paths are selected at runtime
  - Each protocol exploits its own sw/hw path
  - Eager
  - Multipacket Eager
  - Rendezvous
- Ensures support for ULPs
  - Intel, OpenMPI, MPICH, and MVAPICH
  - Sandia OpenSHMEM, GASNet
- Software stack for CN5000 and beyond

Performance Results



OMPI 4.1.5, MVAPICH 3,0b – CN internal Icelake Cluster. Default mpirun options. Libfabric 5ad7ca12a

Omni-Path Future – CN5000

- 400G foundation adaptor with 48 port edge switches and 576 port Director Class Switches
- Support for copper cables in racks and optical cables between racks
- New topologies – MegaFly and DragonFly
  - Up to 330k total endpoints
- OpenBMC support on all switches
  - RedFish API support
- Fine-grain adaptive routing support for advanced congestion control and avoidance
- Same software stack as OPA100 today using libfabric and OPX provider