Intel® Omni-Path Tutorial

Ravindra Babu Ganapathi

Product Owner/ Technical Lead Omni Path Libraries

Sayantan Sur

Senior Software Engineer

Legal Notices and Disclaimers


What’s new with Intel® Omni-Path Architecture

Intel® OPA Hitting Its Stride in 100G HPC Fabrics

Top500 listings continue to grow

- 4 Top 15 systems, 13 Top 100 systems
- 36% more systems from Nov 2016 list
- First Skylake systems
- And almost 10% of Top500 Rmax performance

New deployments all over the globe

- All geos, and new deployments in HPC Cloud and AI

Expanding capabilities

- NVMe over Fabric, Multi-modal Data Acceleration (MDA), and robust support for heterogeneous clusters

Source: Top500.org

Intel® OPA in the HPC Cloud Today

Rescale

https://www.hpcwire.com/2017/04/18/knights-landing-processor-omni-path-makes-cloud-debut/

Penguin

http://insidehpc.com/2017/03/penguin-computing-adds-omni-path-lustre-hpc-cloud/

Cloud providers deploying Intel® Omni-Path in single tenant, non virtualized HPC cloud environments

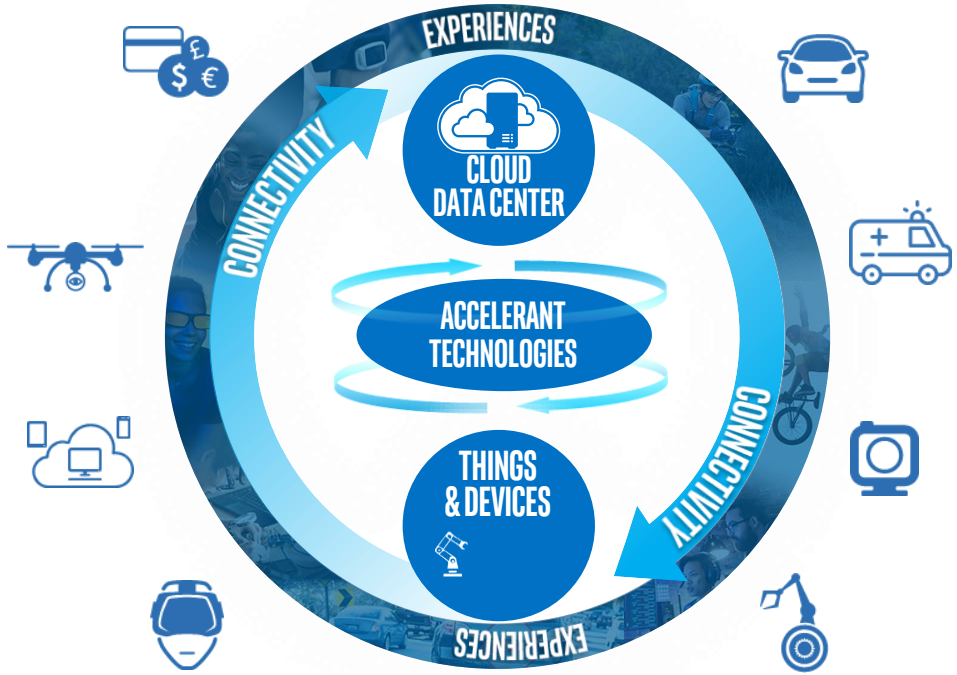
Next Up for Intel® OPA: Artificial Intelligence

Intel offers a complete AI Portfolio

- From CPUs to software to computer vision to libraries and tools

Intel® OPA advantages: Breakthrough performance on scale-out apps

- Low latency
- High bandwidth
- High message rate
- GPU Direct RDMA support
- Xeon Phi Integration

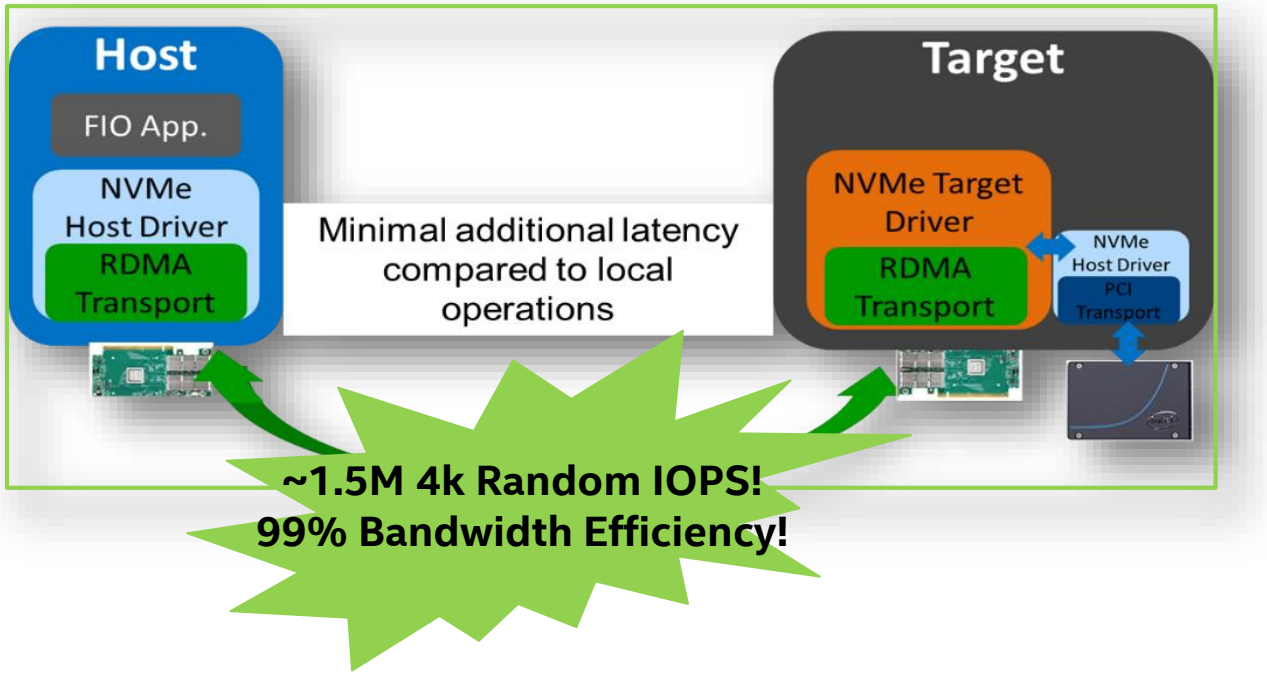


World-class interconnect solution for shorter time to train

NVMe* over OPA: Ready for Prime Time

OPA + Intel® Optane™ Technology

- High Endurance
- Low latency
- High Efficiency
- Total NVMe over Fabric Solution!



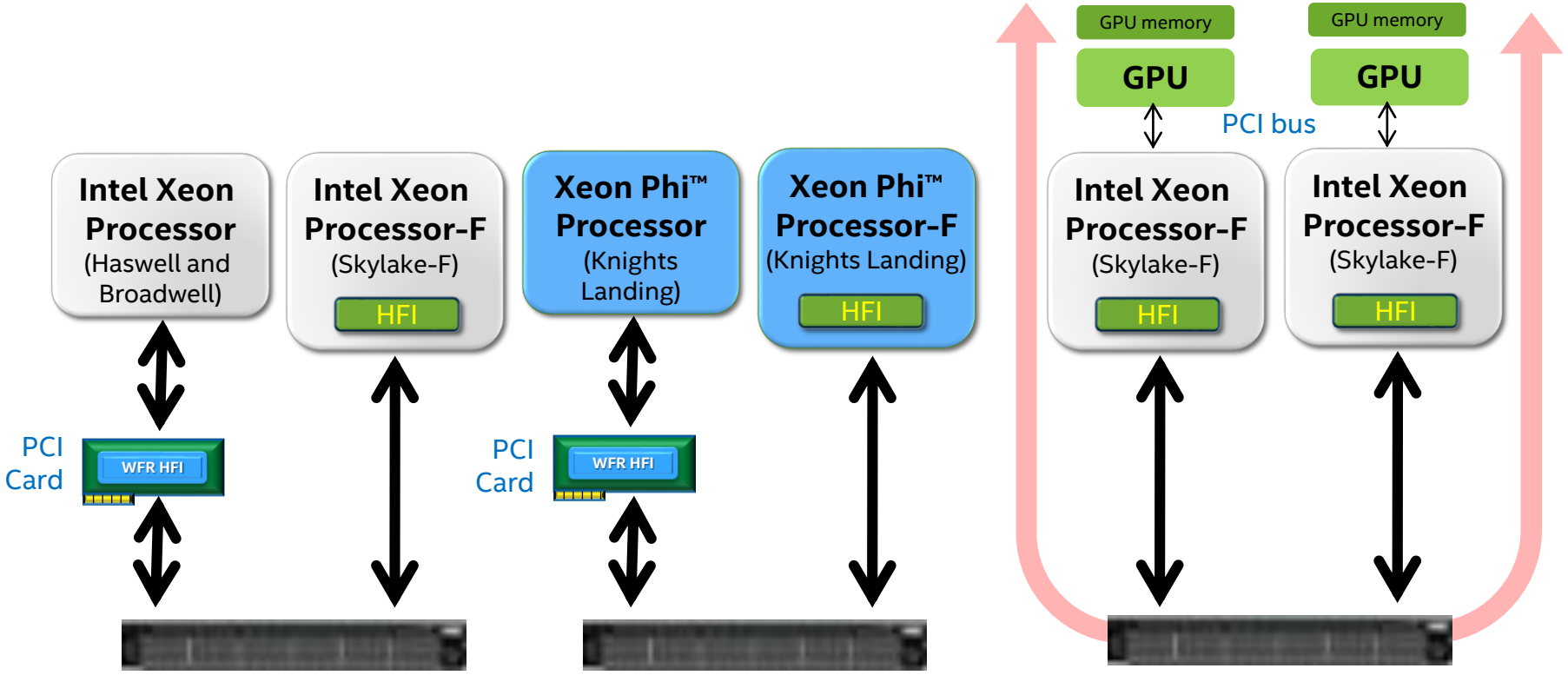
NVMe-over-OPA status

- Supported in 10.4.1 release
- Compliant with NVMeF specs 1.0

Only Intel is delivering a total NVMe over Fabric solution!

Target and Host system configuration: 2 x Intel® Xeon® CPU E5-2699 v3 @ 2.30Ghz, Intel® Server Board S2600WT, 128GB DDR4, CentOS 7.3.1611, kernel 4.10.12, IFS 10.4.1, NULL-BLK, FIO 2.19 options hfi1 krcvqs=8 sge_copy_mode=2 wss_threshold=70

Intel® Omni-Path Maximizing Support for Heterogeneous Clusters



Greater flexibility for creating compute islands depending on user requirements

Multi-Modal Data Acceleration (MDA): Optimizing Data Movement through the Fabric

VERBS

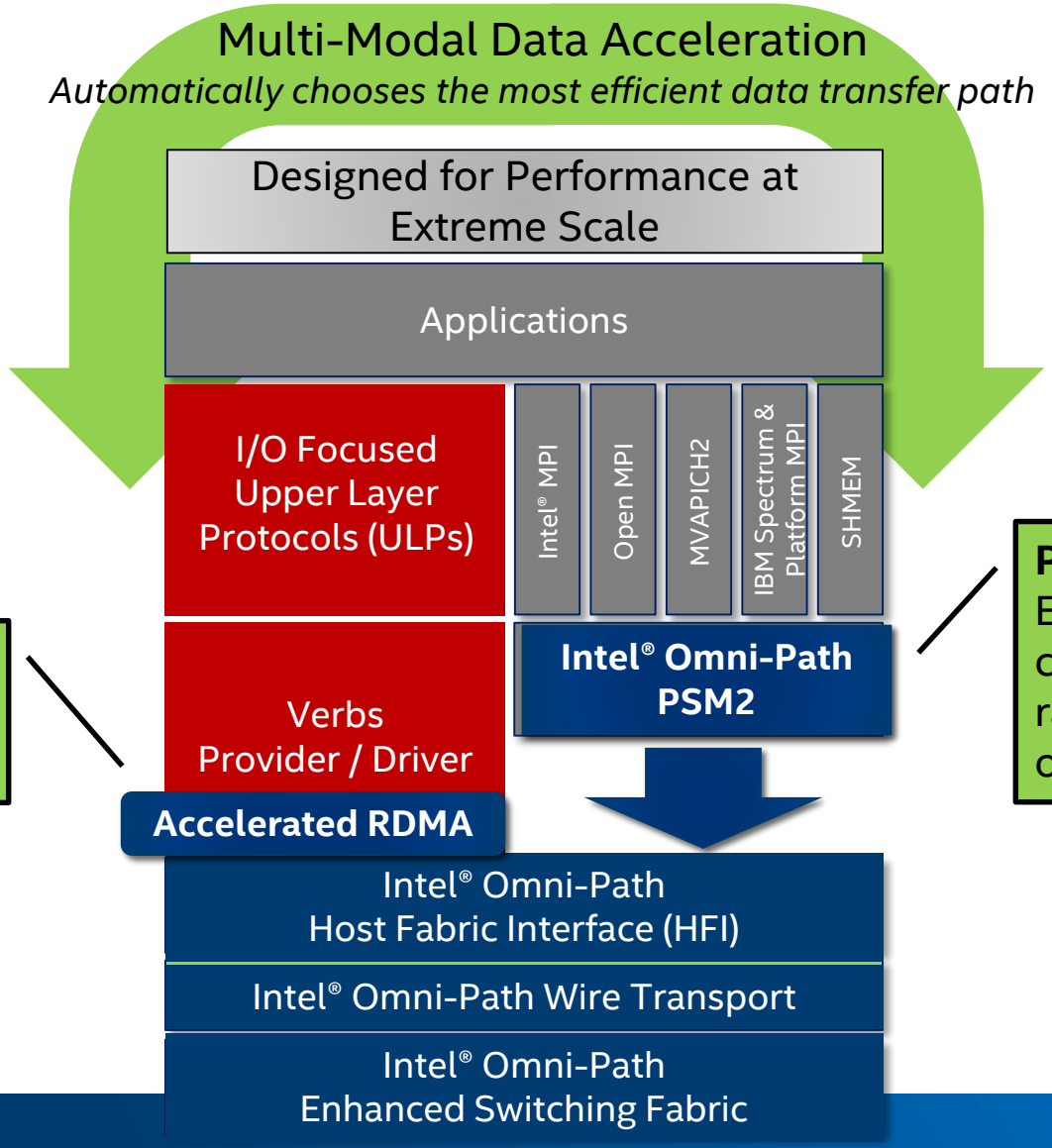
Storage traffic

(Large data packets, bandwidth sensitive)

MPI

HPC app traffic

(Small / Med data packets, latency, bandwidth sensitive)

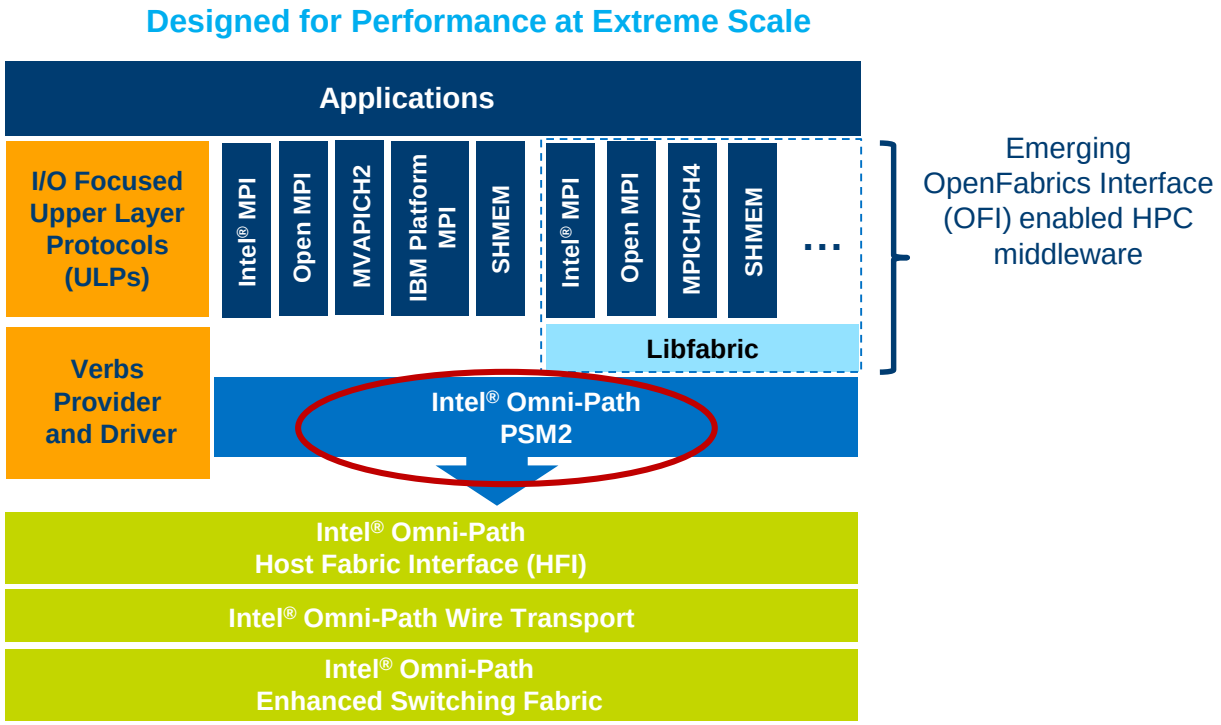


Accelerated RDMA: Performance enhancements for large message read or writes

Perf Scaled Messaging 2 (PSM2): Efficient support for MPI, 1/10 the code path, high b/w and message rate, consistent latency independent of cluster scale


PSM2 Enhancements

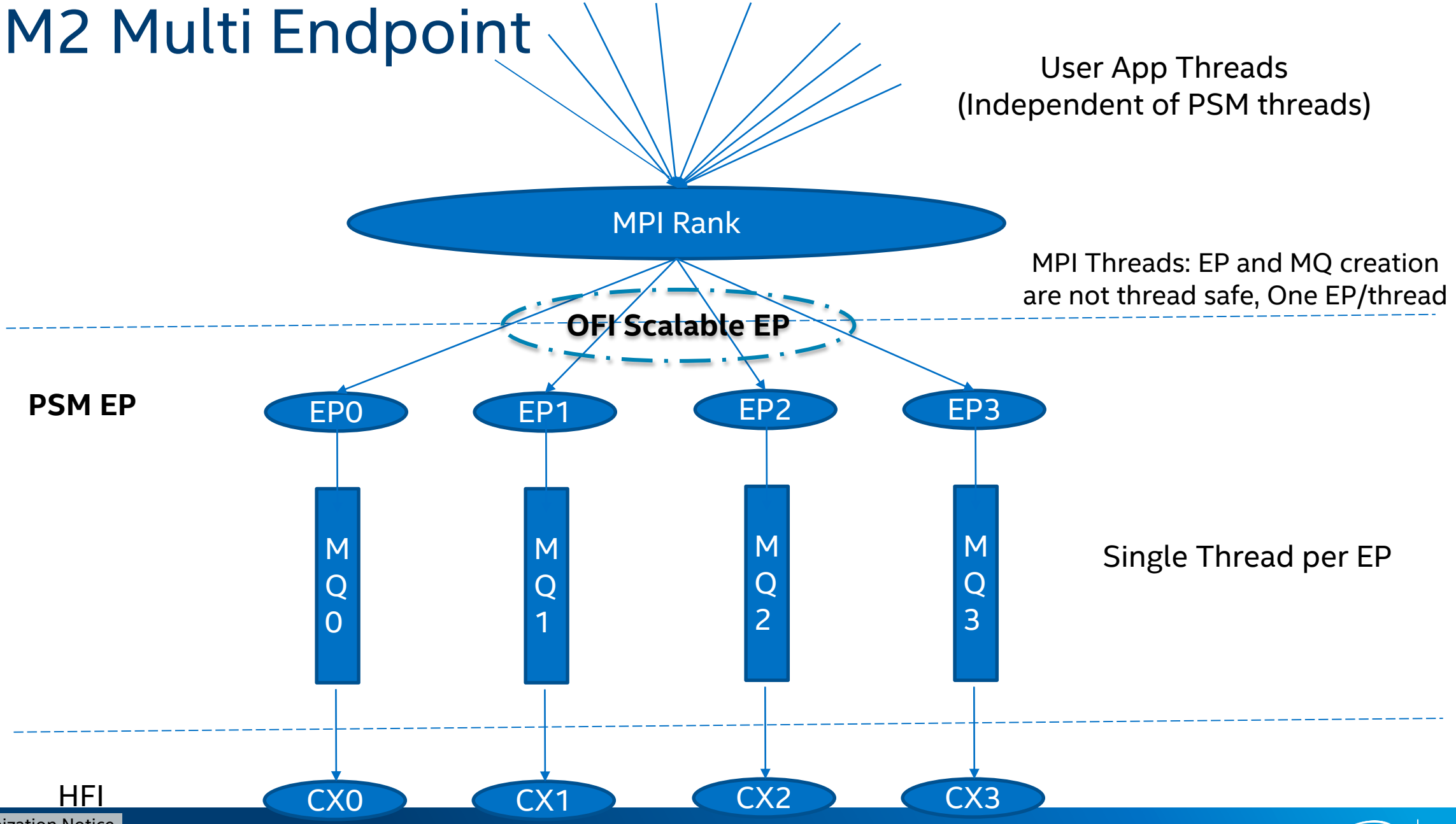
Intel® Omni-Path Architecture **HPC**
Design Focus Architected for Your MPI Application



PSM2 – Performance Scaled Messaging 2

- PSM2 API is a low level high performing communication interface
- Semantics matches with that of compute middleware such as MPI and/or OFI
  - PSM2 EP maps to HW context, Each EP associated with matched queue
- API's are optimized for both latency and bandwidth
  - PIO/Eager for small message latency
  - DMA/Expected for optimal Bandwidth with large message size
- Intel MPI, Open MPI and MVAPICH2 use PSM2 transport for Omni-Path Fabric
- PSM2 Programmer's Guide available @ https://www.intel.com/content/dam/support/us/en/documents/network-and-i-o/fabric-products/Intel_PSM2_PG_H76473_v6_0.pdf

PSM2 Multi Endpoint



PSM2 Multiple Endpoint

- OFI PSM2 provider now supports Scalable Endpoints
- Single process opening and concurrently using PSM2 Endpoints
- Endpoints are directly mapped to hardware context giving full parallelism
- Enables hybrid programming models such as MPI + OpenMP
  - Performance is scaling similar to multiple MPI processes
  - One MPI process with multiple threads can saturate multiple HFI
- Multiple threads using a single endpoint not supported
  - Higher level software expected to use locks

Inter and Intra-Node GPUDirect

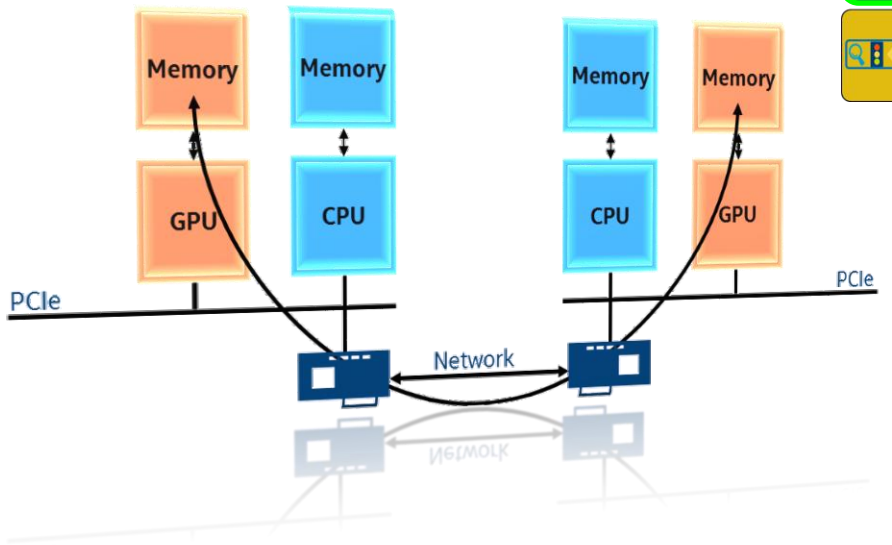
CPU/Fabric Integration

Optimized Host Implementation

Enhanced Fabric Architecture

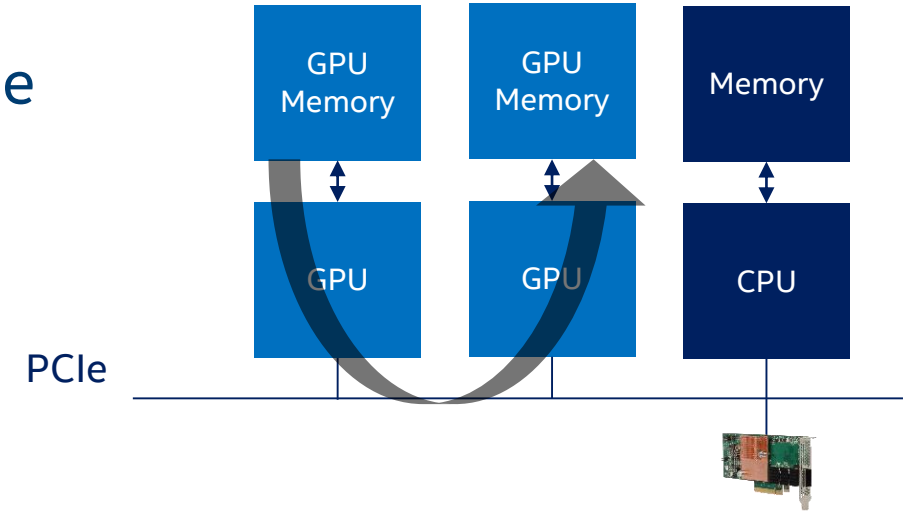
GPUDirect RDMA v3 Support

- GPUDirect RDMA support
- Optimized Transfer through Host Buffer



GPUDirect P2P Intra-Node Support

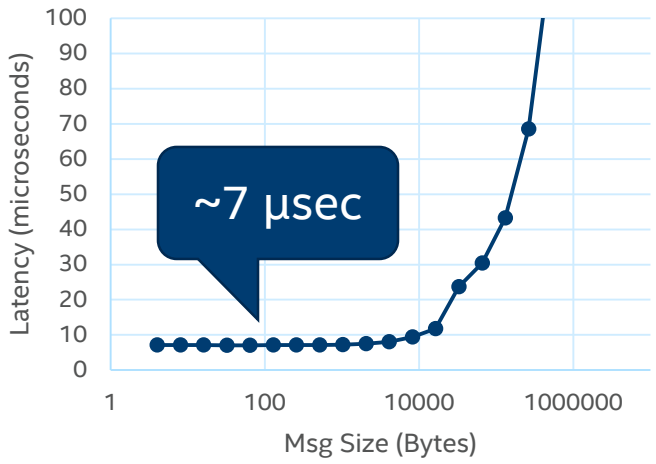
- Ranks/processes within the same node
- Both single- and multi-GPU support
- Direct GPU to GPU acceleration



GPU Buffer Transfer Latency & Bandwidth - Ohio State MicroBenchmarks v5.3.2
Intel® Omni-Path Architecture - Intel® Xeon® CPU E5-2697 v3
NVIDIA* Corporation GK110BGL [Tesla K40c]*

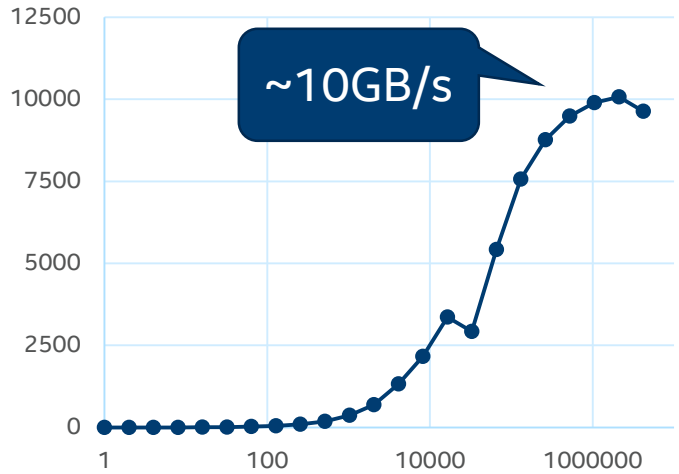
Latency

osu_latency -d cuda D D



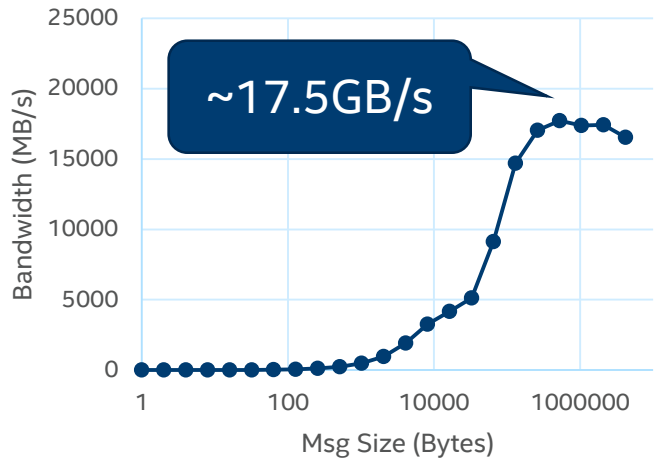
Uni-dir Bandwidth

osu_bw -d cuda D D



Bi-dir Bandwidth

osu_bibw -d cuda D D



PSM2_GPUDIRECT_RECV_THRESH
Allows you to specify a threshold value (in bytes). If the threshold is exceeded, the GPUDirect* RDMA feature will not be used on the receive side of a connection.
Default: PSM2_GPUDIRECT_RECV_THRESH=0 ***(specified in bytes)***

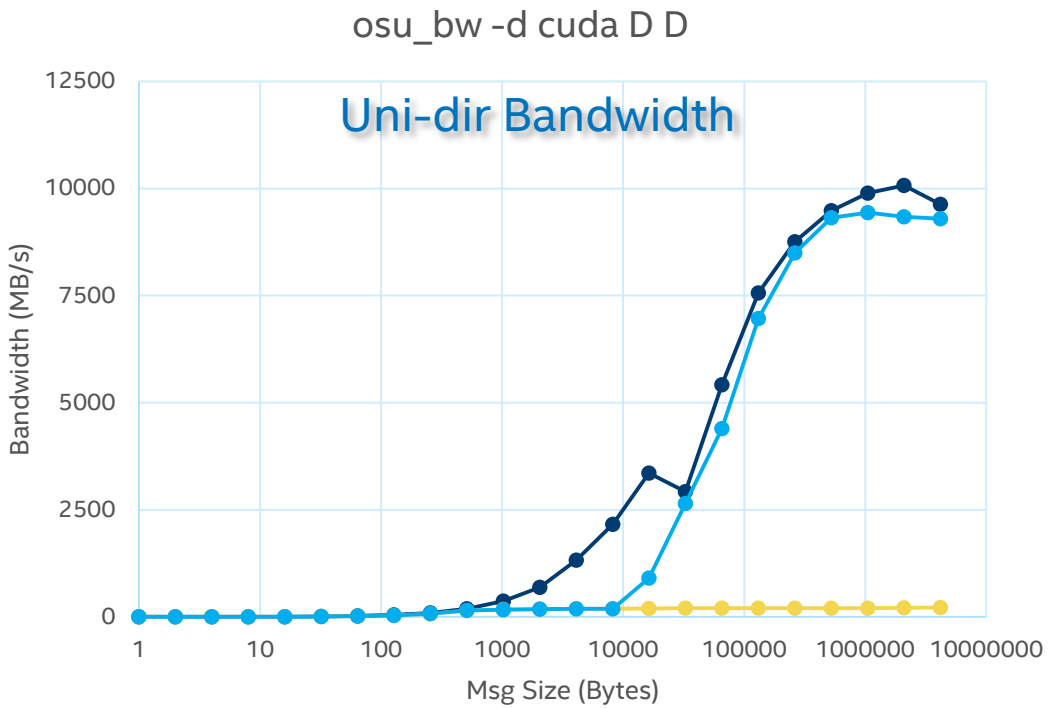
PSM2_GPUDIRECT_SEND_THRESH
Allows you to specify a threshold value (in bytes). If the threshold is exceeded, the GPUDirect* RDMA feature will not be used on the send side of a connection.
Default: PSM2_GPUDIRECT_SEND_THRESH=30000 ***(specified in bytes)***

Example run command: mpirun -np 2 --map-by ppr:1:node -host host1,host2 -x PSM2_CUDA=1 -x PSM2_GPUDIRECT=1 -x HFI_UNIT=1 --allow-run-as-root taskset -c 14 ./osu_latency -d cuda D D

Last updated: June 29, 2017

Intel® Xeon® processor E5-2697 v3, Red Hat Enterprise Linux Server release 7.2 (Maipo), 3.10.0-327.el7.x86_64. 3D controller: NVIDIA Corporation GK110BGL [Tesla K40c] OSU Microbenchmarks version 5.3.2 Open MPI 1.10.4-cuda-hfi as packaged with IFS 10.5.0.0.119. GPU device connected to second socket. Benchmark processes pinned to second socket. Dual socket servers connected back to back with no switch hop. Intel® Turbo Boost Technology enabled, Intel® Hyper-Threading Technology enabled.

GPU Buffer Transfer Latency & Bandwidth - Ohio State MicroBenchmarks v5.3.2
Intel® Omni-Path Architecture - Intel® Xeon® CPU E5-2697 v3
NVIDIA* Corporation GK110BGL [Tesla K40c]*



GPUDirect* works best when both the GPU and HFI are in the same PCIe* root complex. Two environment variables are designed to improve performance:

PSM2_GPUDIRECT_RECV_THRESH
Allows you to specify a threshold value (in bytes). If the threshold is exceeded, the GPUDirect* RDMA feature will not be used on the receive side of a connection.
Default: PSM2_GPUDIRECT_RECV_THRESH=0 ***(specified in bytes)***

PSM2_GPUDIRECT_SEND_THRESH
Allows you to specify a threshold value (in bytes). If the threshold is exceeded, the GPUDirect* RDMA feature will not be used on the send side of a connection.
Default: PSM2_GPUDIRECT_SEND_THRESH=30000 ***(specified in bytes)***

GPU and HFI on same CPU socket

GPU and HFI on different CPU sockets

GPU and HFI on different CPU sockets with PSM2_GPUDIRECT_RECV_THRESH=8192 (bytes)

Example run command: mpirun -np 2 --map-by ppr:1:node -host host1,host2 -x PSM2_CUDA=1 -x PSM2_GPUDIRECT=1 -x HFI_UNIT=0 -x PSM2_GPUDIRECT_RECV_THRESH=8192 --allow-run-as-root taskset -c 14 ./bw -d cuda D D

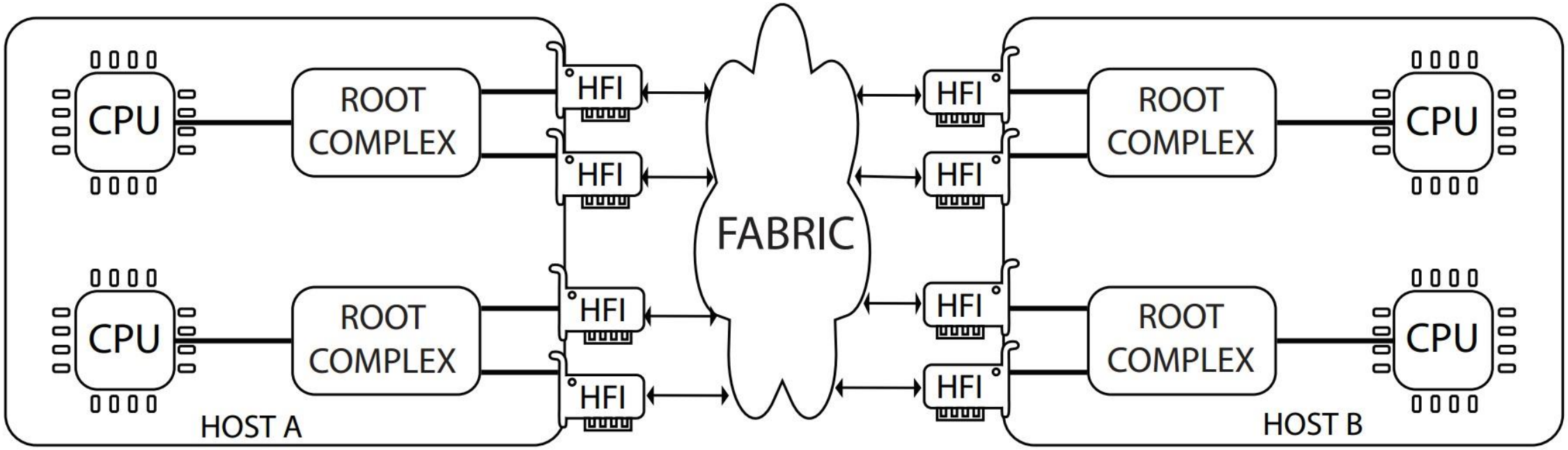
Last updated: June 29, 2017

Intel® Xeon® processor E5-2697 v3, Red Hat Enterprise Linux Server release 7.2 (Maipo), 3.10.0-327.el7.x86_64. 3D controller: NVIDIA Corporation GK110BGL [Tesla K40c] OSU Microbenchmarks version 5.3.2 Open MPI 1.10.4-cuda-hfi as packaged with IFS 10.5.0.0.119. GPU device connected to second socket. Benchmark processes pinned to second socket. Dual socket servers connected back to back with no switch hop. Intel® Turbo Boost Technology enabled, Intel® Hyper-Threading Technology enabled.

PSM2 Multi-Rail and Device Selection Enhancements

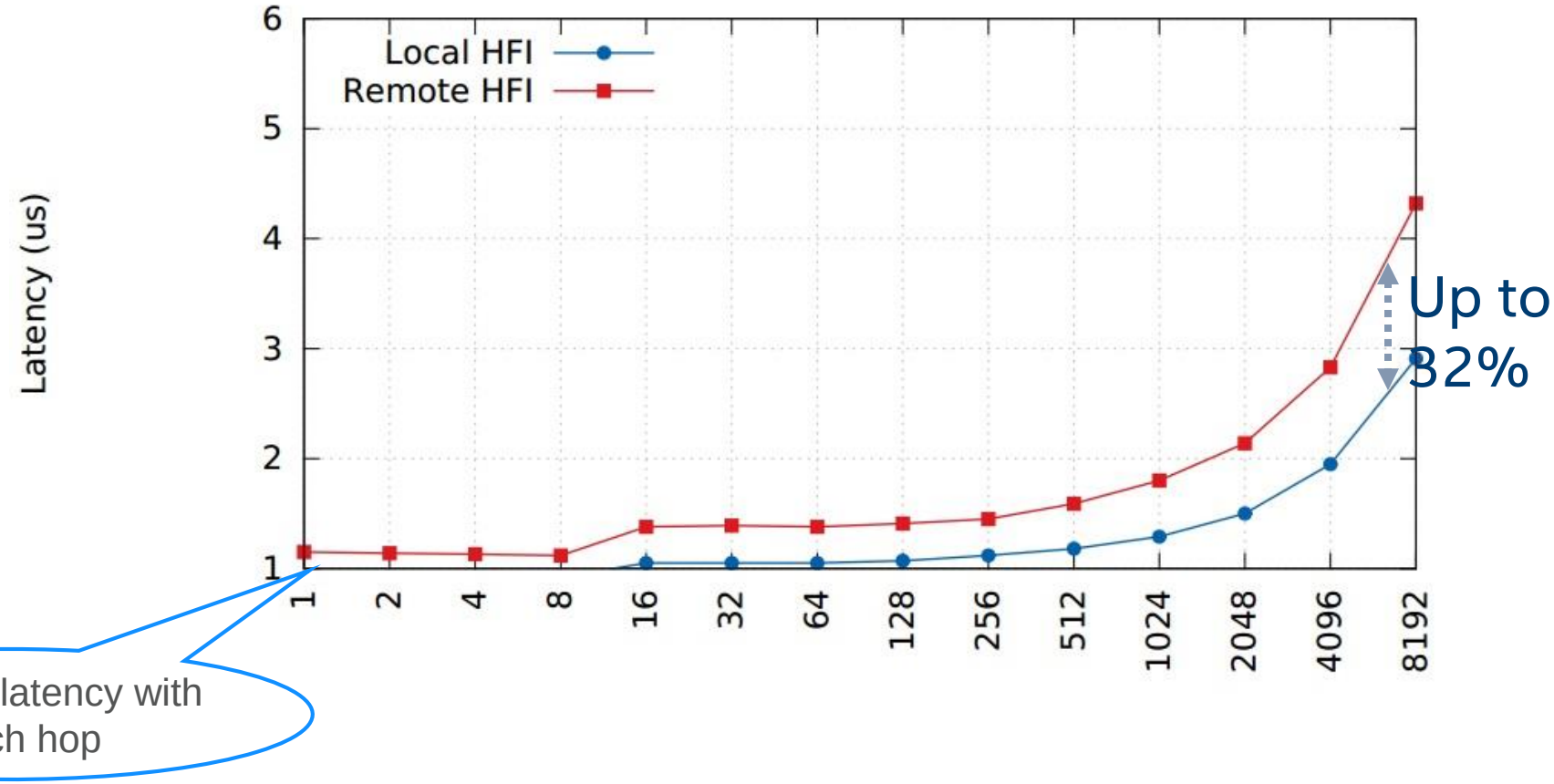
- PSM2 env variables for optimal app performance
  - PSM2 MULTIRAIL: Takes the value 0 to 2
    - Value 0: multi-rail turned off
    - Value 1: multi-rail across all the HFI within the node
    - Value 2: multi-rail within a NUMA domain (root complex)
  - PSM2 MULTIRAIL MAP
    - unit:port, unit:port, unit:port…
  - HFI SELECTION ALG
    - “Round Robin”
    - “Packed”

NUMA CONFIGURATION: FOUR DEVICES PER NODE



Dual-socket Intel® Xeon CPU E5-2699 v4 (commonly labeled Broadwell cores) , frequency of 2.2 GHz for all the cores. Intel® Hyper-threading technology turned on. Each system has 4 Intel® OPA in each of the 4 x16 PCIe slots and Intel® QuickPath Interconnect (QPI) link connection between the sockets. Two such servers are connected over a switch fabric using Omni-Path Edge switch.

PLATFORM IMPACT: OSU LATENCY, LOCAL VS REMOTE HFI

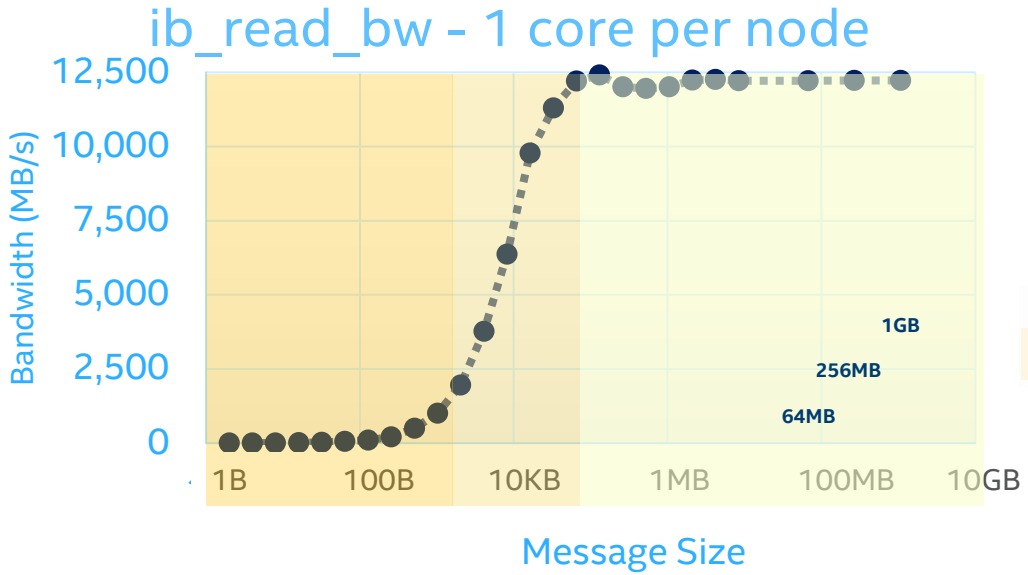


Dual-socket Intel® Xeon CPU E5-2699 v4 (commonly labeled Broadwell cores) , frequency of 2.2 GHz for all the cores. Intel® Hyper-threading technology turned on. Each system has 4 Intel® OPA in each of the 4 x16 PCIe slots and Intel® QuickPath Interconnect (QPI) link connection between the sockets. Two such servers are connected over a switch fabric using Omni-Path Edge switch.


intel
Accelerated RDMA

INTEL® OMNI-PATH ARCHITECTURE (INTEL® OPA)

RDMA BANDWIDTH PERFORMANCE - ACCELERATED RDMA

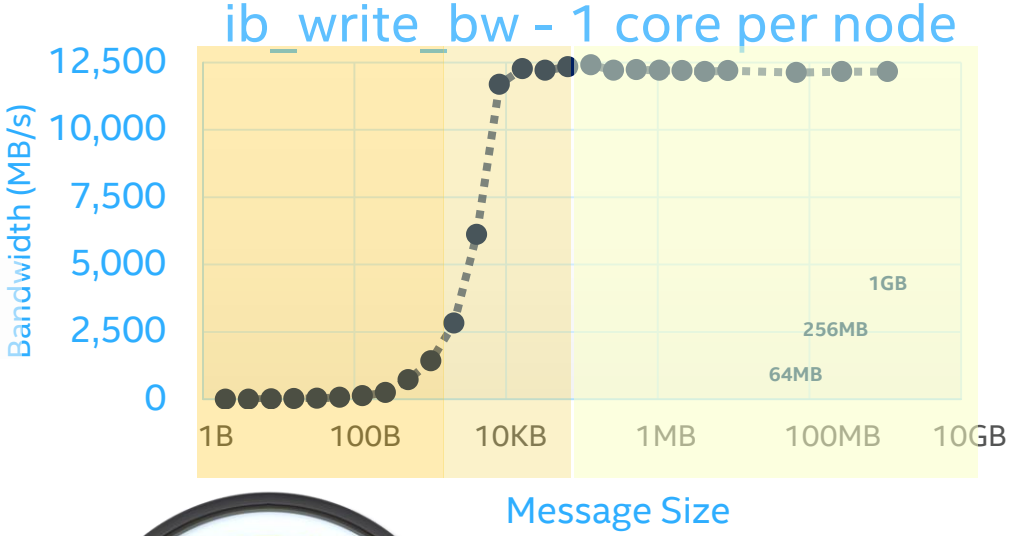


HIGHER is Better

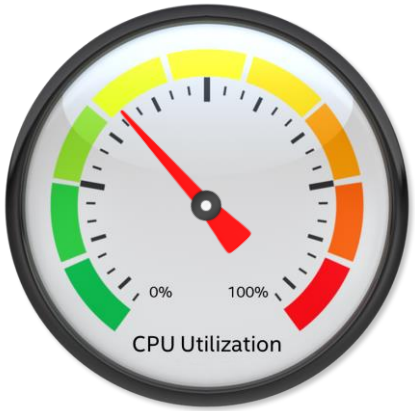
Key:
Acc RDMA enabled

Small | Medium | Large

Multiple Data Transmission methods



Up to **40% lower CPU utilization** with Accelerated RDMA enabled[1]
Supporting high bandwidth across message sizes



Intel® Xeon® processor E5-2697a v4, Intel® Turbo Boost Technology enabled, Intel® Hyper-Threading Technology disabled. Memory: 2133 MHz DDR4 per node . 3.10.0-327.36.3.el7.x86_64, RHEL 7.2. Intel® OPA: ib_write_bw version 5.33. Average bandwidth is reported. Server side command: ib_write_bw -a (or -s Size) -F -R. Client side command: ib_write_bw -a (or -s Size) -F -R <server IPoIB address>. Intel® OPA HFI - Intel Corporation Device 24f0 – Series 100 HFI ASIC. OPA Switch: Series 100 Edge Switch – 48 port. IFS 10.3.1.0.22. Contents of /etc/modprobe.d/hfi1.conf: options hfi1 cap_mask=0x4c09a01cbba eager_buffer_size=8388608 krcvqs=4 pcie_caps=0x51 max_mtu=10240. 1. CPU statistics measured from /proc/stat immediately before and after the benchmark. Total CPU % is calculated as Δ(user + system)/ Δ(user + system + idle) for all cores. At 1 GB message size, ib_read_bw client : 6.24% CPU Acc. RDMA disabled, 3.70% CPU Acc. RDMA enabled. ib_write_bw server: 6.23% CPU Acc RDMA disabled, 3.70% CPU Acc. RDMA enabled

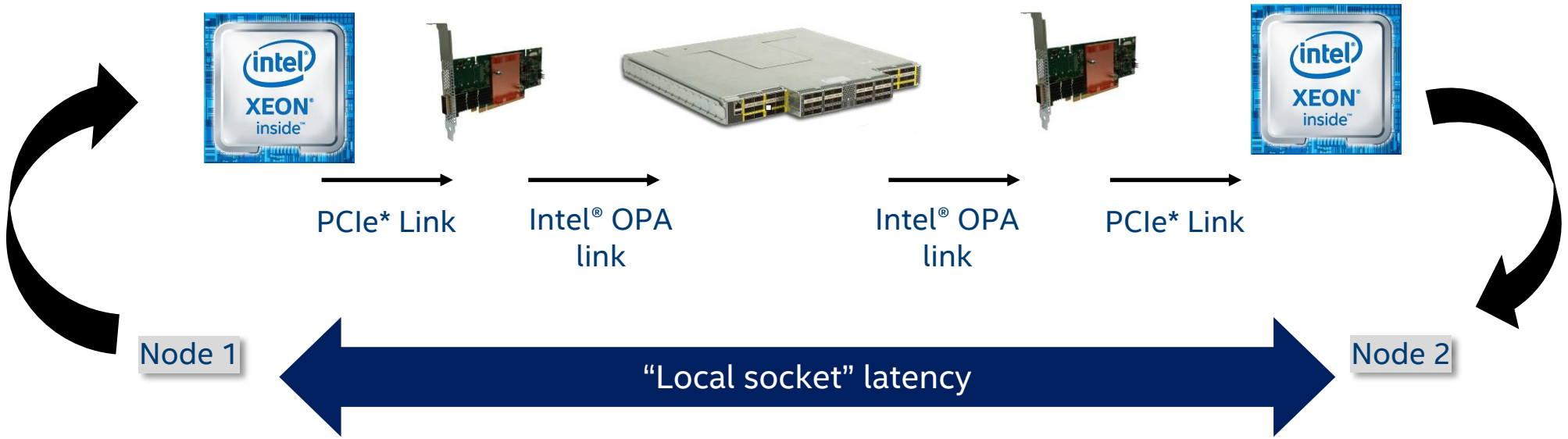
Last Update: March 10 2017


intel
SKL Performance

INTEL® QUICKPATH INTERCONNECT VS ULTRAPATH INTERCONNECT

- Previous generation Intel® Xeon® processors were interconnected with the Intel® QuickPath Interconnect (QPI)
- Intel® Xeon® Scalable Family processors (codename Skylake) are now connected with the Intel® UltraPath Interconnect

Example: Local MPI processes on dual socket servers (no QPI or UPI involvement) :

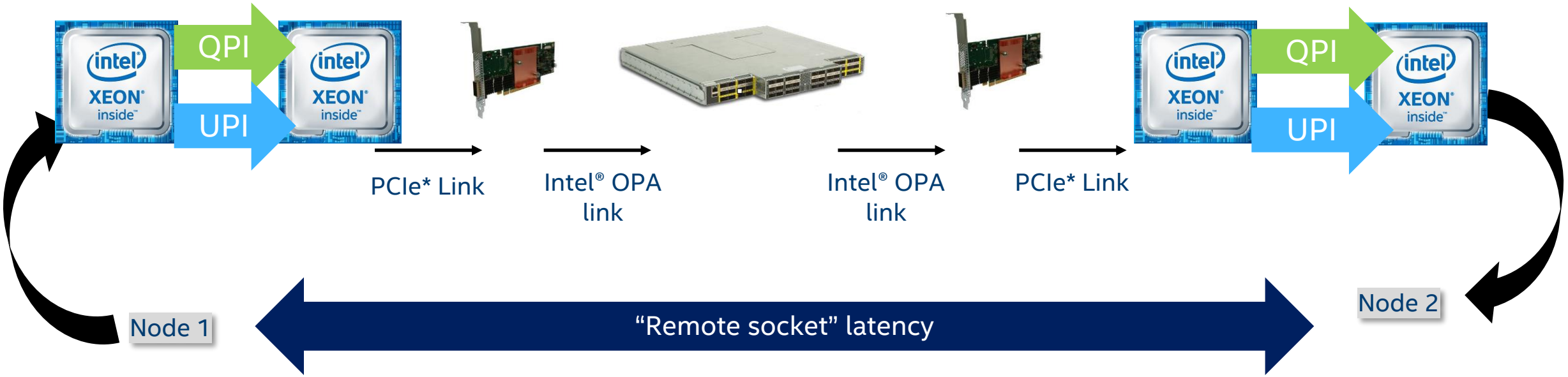


For more detail on Intel® UPI see: https://software.intel.com/en-us/articles/intel-xeon-processor-scalable-family-technical-overview

INTEL® QUICKPATH INTERCONNECT VS ULTRAPATH INTERCONNECT

For systems with one Intel® OPA adapter installed on a single socket, any data communication involving cores on the other socket need to traverse the QPI/UPI

Example: Remote MPI processes on dual socket servers:



For more detail on Intel® UPI see: https://software.intel.com/en-us/articles/intel-xeon-processor-scalable-family-technical-overview

INTEL® QUICKPATH INTERCONNECT VS ULTRAPATH INTERCONNECT

Benefits of the UPI vs QPI for processes on the remote socket:

	Intel® Xeon® E5-2697A v4 CPU[1] QPI	Intel® Xeon® Platinum 8170 CPU[2] UPI
MPI Latency using local socket	**1.13 µsec**	**1.10 µsec**
MPI Latency using remote socket	**1.38 µsec**	**1.26 µsec**
Difference is the intra-socket latency (x2 servers)	**0.25 µsec**	**0.16 µsec**

up to 36% faster

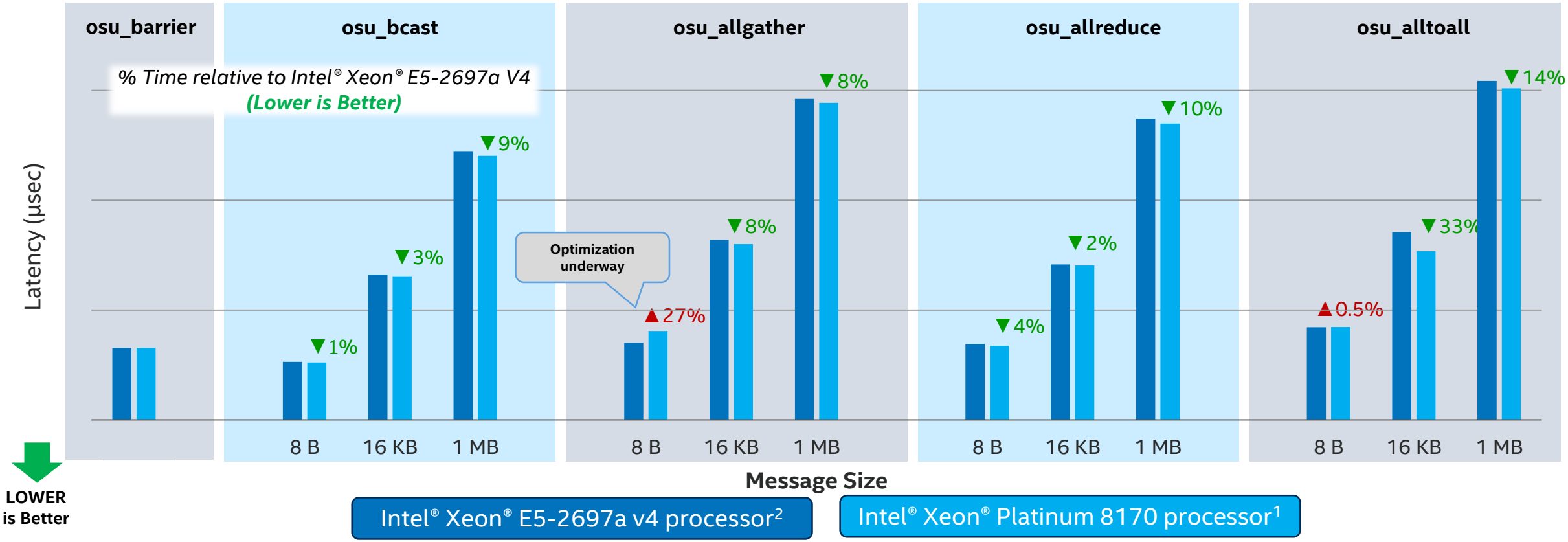
See configuration item #2

***CPU frequency fixed to 2.1 GHz for both processors**

For more detail on Intel® UPI see: https://software.intel.com/en-us/articles/intel-xeon-processor-scalable-family-technical-overview

MPI COLLECTIVE PERFORMANCE - OHIO STATE MICROBENCHMARKS

MPI PROCESS LOCAL TO HFI

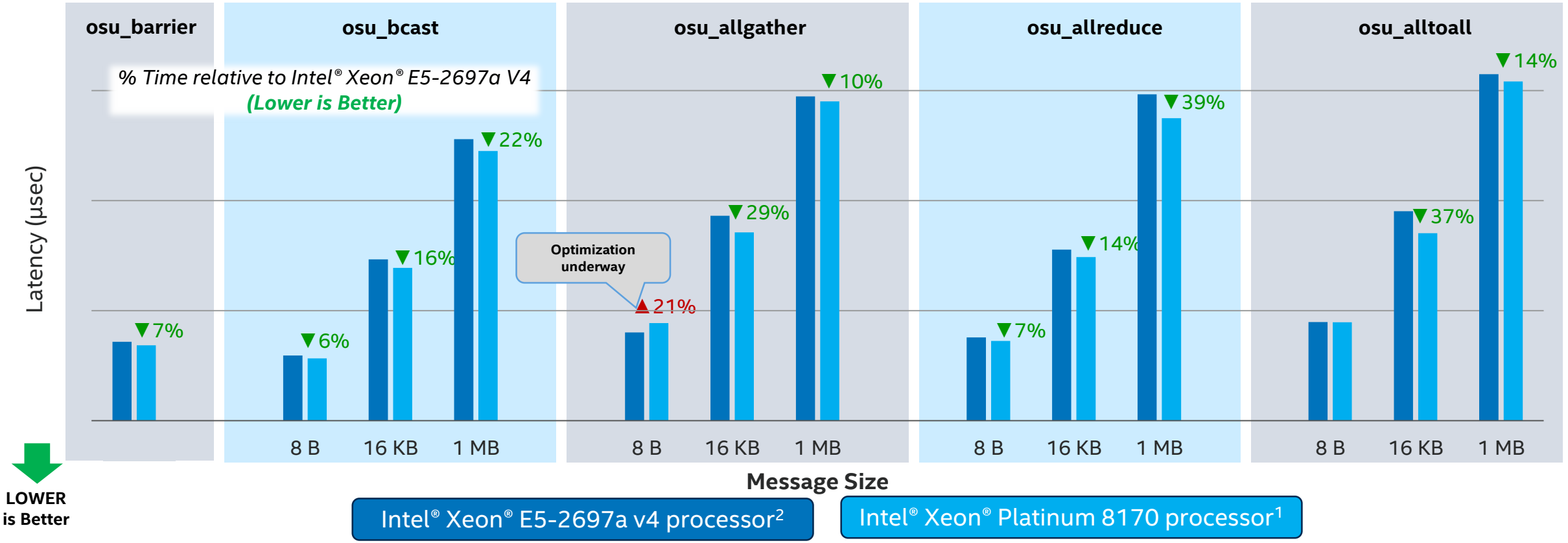


See configuration item #3

For more detail on Intel® UPI see: https://software.intel.com/en-us/articles/intel-xeon-processor-scalable-family-technical-overview

MPI COLLECTIVE PERFORMANCE - OHIO STATE MICROBENCHMARKS

MPI PROCESS REMOTE TO HFI

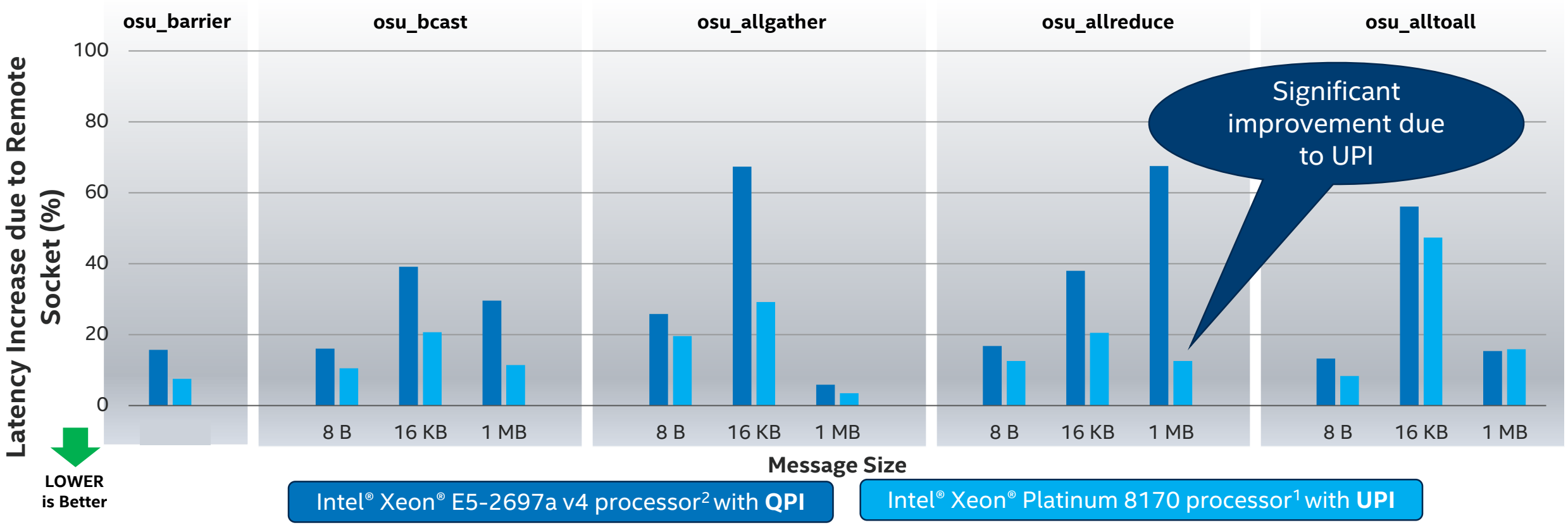


See configuration item #3

For more detail on Intel® UPI see: https://software.intel.com/en-us/articles/intel-xeon-processor-scalable-family-technical-overview

MPI COLLECTIVE PERFORMANCE - OHIO STATE MICROBENCHMARKS

INTEL® QUICKPATH VS ULTRAPATH INTERCONNECT



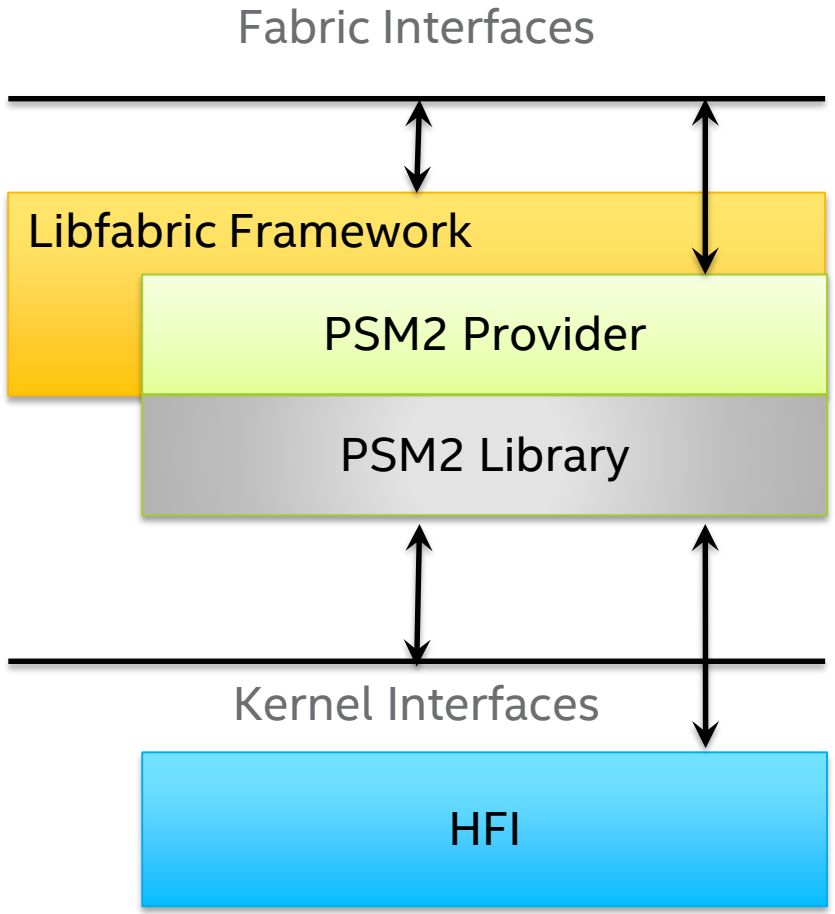
See configuration item #3

For more detail on Intel® UPI see: https://software.intel.com/en-us/articles/intel-xeon-processor-scalable-family-technical-overview

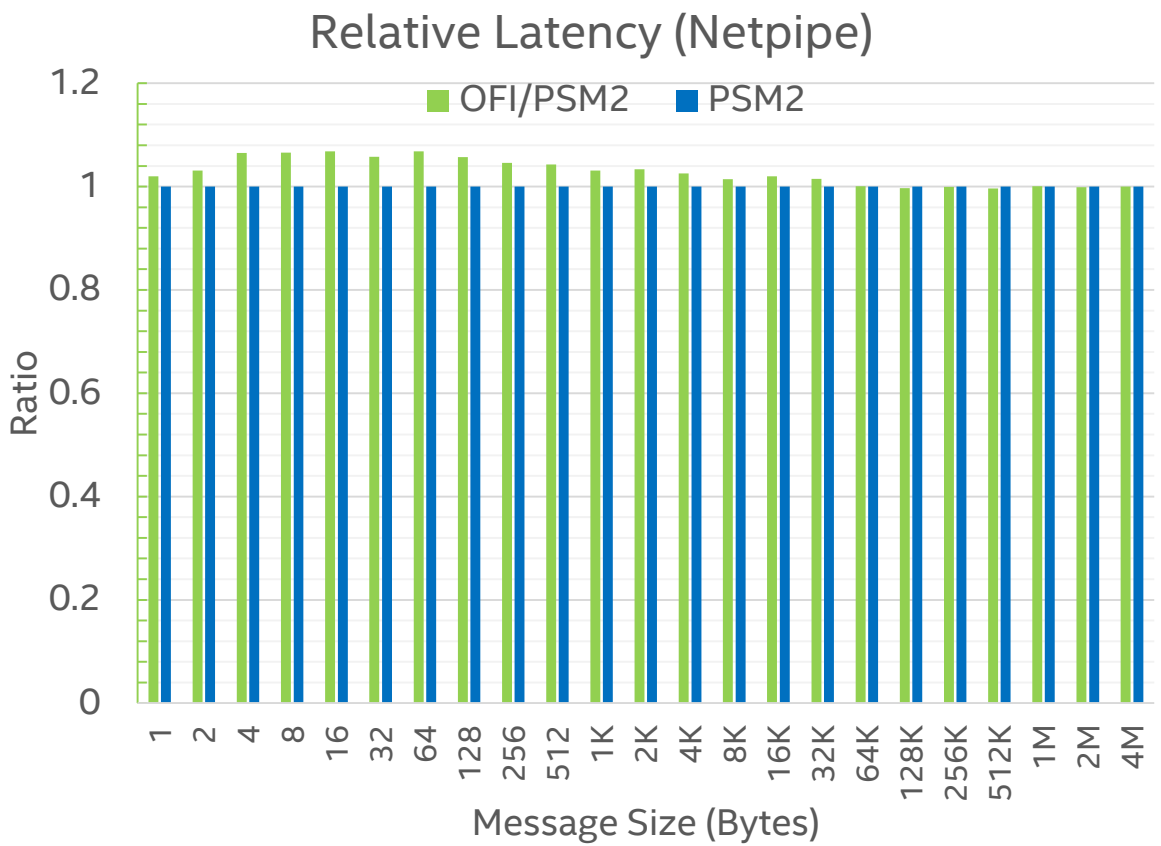

intel
OFI over Omni-Path

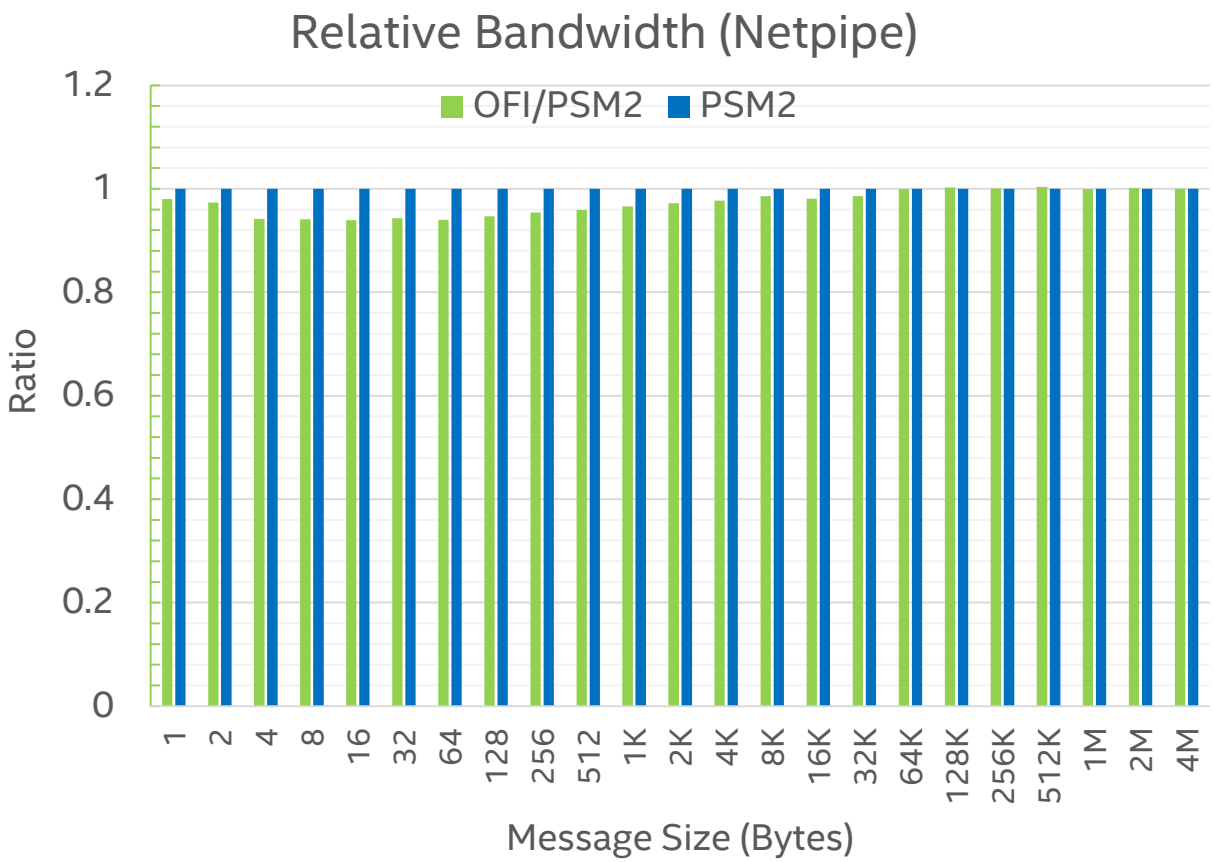
Libfabric on Intel® OPA

- Libfabric PSM2 provider uses the public PSM2 API
- Semantics matches with that of compute middleware such as MPI
  - PSM2 EP maps to HW context, Each EP associated with matched queue
- Regular Endpoints share a single HW context
- Scalable Endpoints provide independent HW contexts
- PSM2 library uses PIO/Eager for small message and DMA/Expected for Bandwidth for large messages
- Current focus is full OFI functionality
- Performance optimizations are possible to reduce call overheads, and provide better semantic mapping of OFI directly to HFI



Intel® OPA/PSM Provider Performance





Performance is comparable for most messages

- 6.7% overhead for 64B latency, and 6.1% overhead for 16B bandwidth
- Optimizations are ongoing

Intel Xeon E5-2697 (Ivy Bridge) 2.6GHz; Intel Omni-Path; RHEL 7.3; libfabric 1.5.0 alpha; IFS 10.5.0.0.115

MPI/OFI Design Principles

Directly map MPI constructs to OFI features as much as possible

Focus on optimizing communication calls

- Avoid branch, function calls, cache misses, memory allocation

MPI	OFI
MPI_Send, MPI_Recv, etc.	fi_tsend, fi_trecv
Window	Memory region (fi_mr(3))
Comm. calls (MPI_Put, MPI_Get, etc…)	fi_write, fi_read, … (fi_rma(3))
Atomic comm. calls (MPI_Accumulate, etc…)	fi_atomic, … (fi_atomic(3))

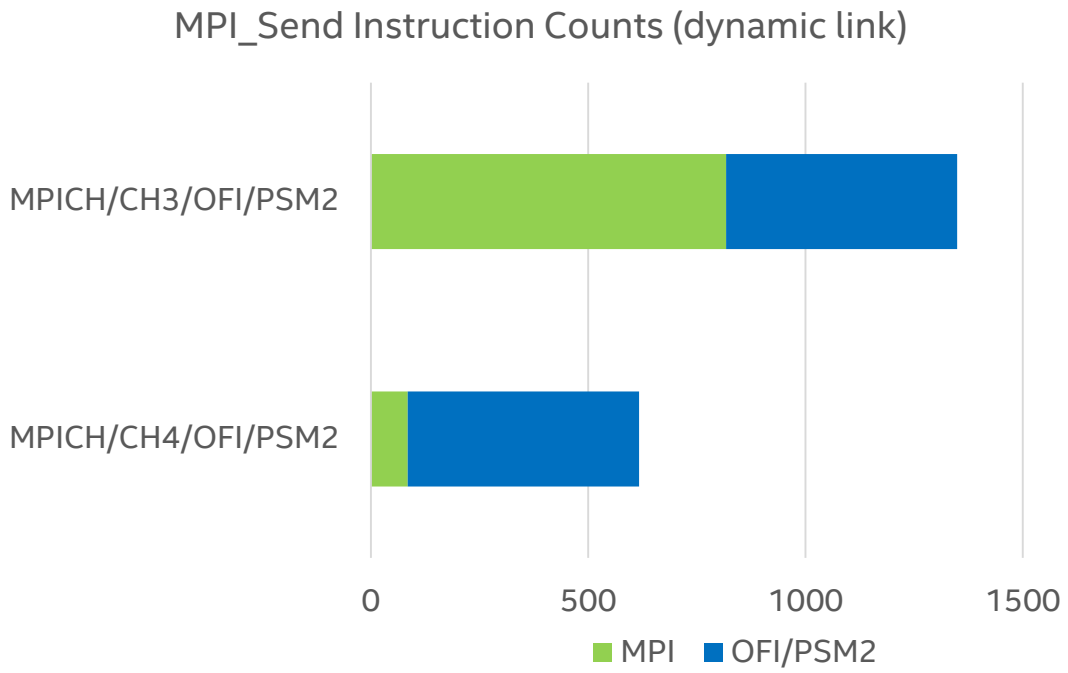
MPICH-OFI

Open-source implementation based on MPICH

- Uses the new CH4 infrastructure
  - Co-designed with MPICH community
- Targets existing and new fabrics via Open Fabrics Interface (OFI)
  - Ethernet/sockets, Intel® Omni-Path, Cray Aries*, IBM BG/Q*, InfiniBand*
- OFI is intended as the default communication interface for future versions MPICH

Improving on MPICH/CH3 Overheads

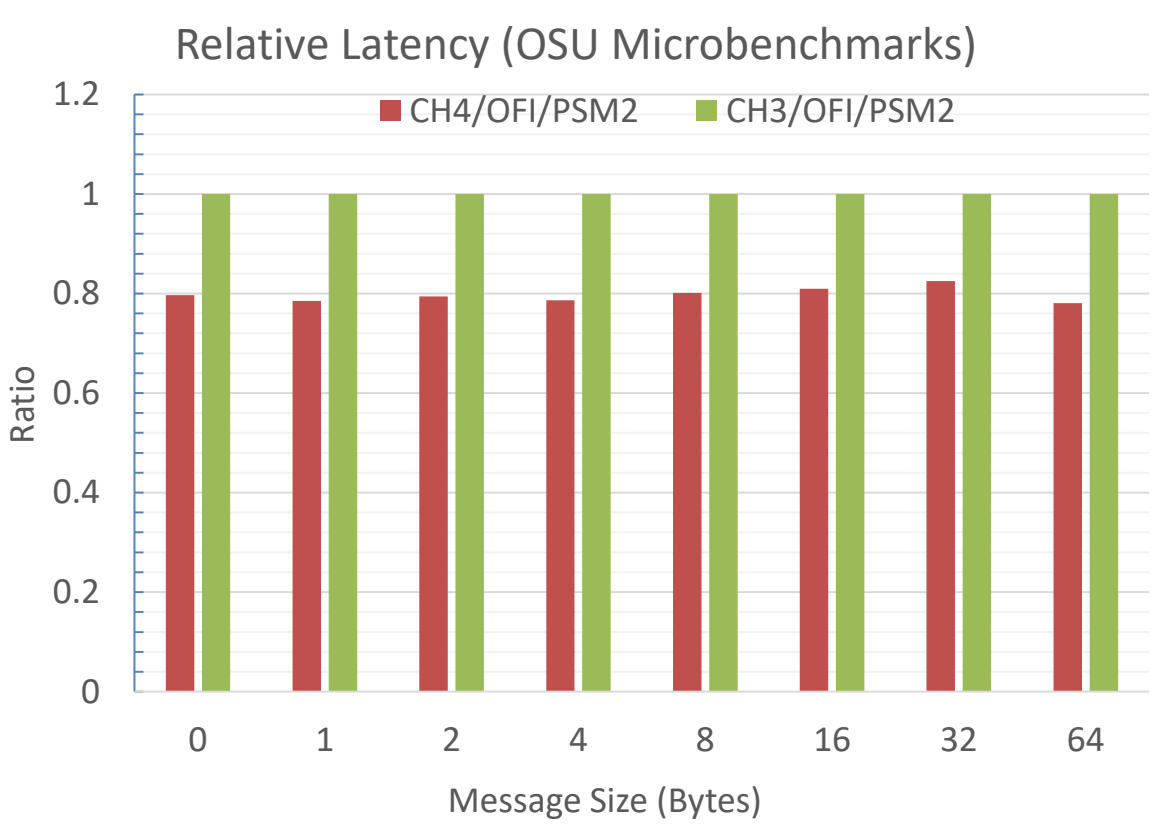
CH4 significantly reduces MPI instructions

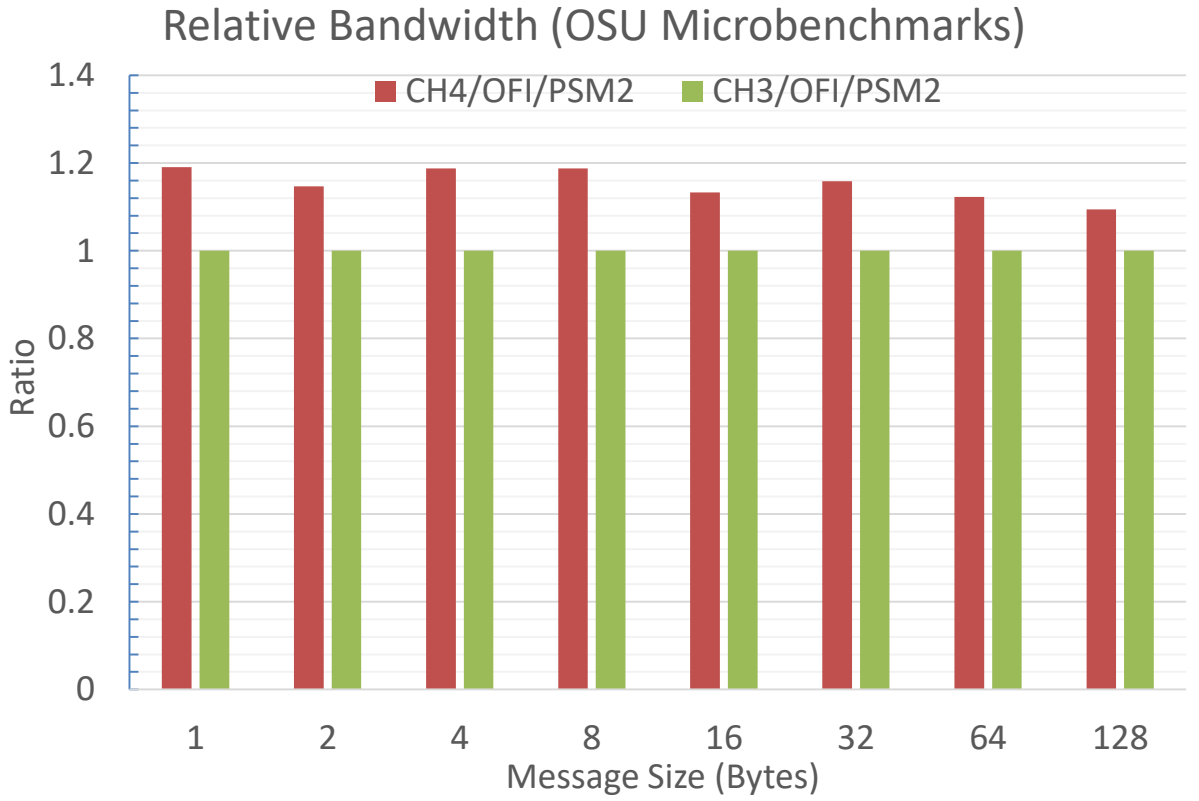


MPICH/CH4:

- 35 instructions with static build
- 85 instructions with dynamic build
- Data collected using GCC v5.2.1on Xeon E5-2697 (Haswell)

MPICH Performance Comparison (KNL + OPA)





Around 20% reduction in latency, gains in bandwidth for small messages

Intel Xeon Phi 7250 (Knights Landing) 1.4GHz; Intel Omni-Path; RHEL 7.3; GCC v5.3.1; OSU Micro-benchmarks 5.3

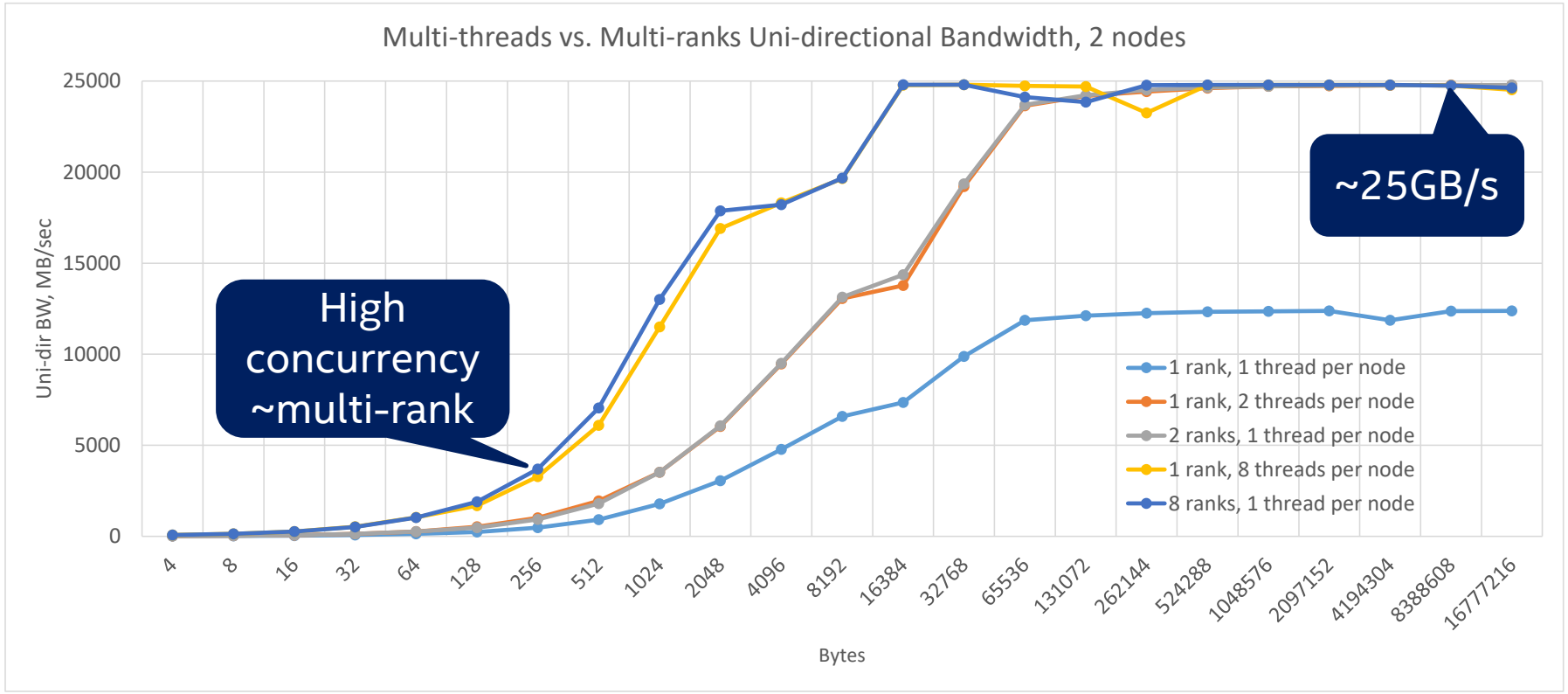
Thread-split usage model with Intel® MPI Library OpenMP* example

```
MPI_Comm split_comm[n];
int main() {
    int i, provided;
    MPI_Init_thread(NULL, NULL, MPI_THREAD_MULTIPLE, &provided);
    for (i = 0; i < n; i++)
        MPI_Comm_dup(MPI_COMM_WORLD, &split_comm[i]);
#pragma omp parallel for num_threads(n)
    for (i = 0; i < n; i++) {
        int j = i;
        MPI_Allreduce(MPI_IN_PLACE, &j, 1, MPI_INT, MPI_SUM, split_comm[i]);
    }
    MPI_Finalize();
}
```

Create thread-split communicators

Use thread-split communicator per thread

Early results with Intel® MPI Library prototype utilizing multiple endpoints



Uni-dir BW MT Benchmark (described on the right)
Intel® Omni-Path Architecture, dual-rail - Intel® Xeon® CPU E5-2699 v4 2.2GHz
This feature may be included in future versions of Intel MPI

Uni-dir BW MT benchmark

```
omp parallel
{
Rank #0, thread #i:
   n x Isend(msg, comm #i)
   Waitall(n x sreq)
   Recv(acc, comm #i)
Rank #1, thread #i:
   n x Irecv(msg, comm #i)
   Waitall(n x rreq)
   Send(acc, comm #i)
Barrier(comm #i)
omp barrier
}
```

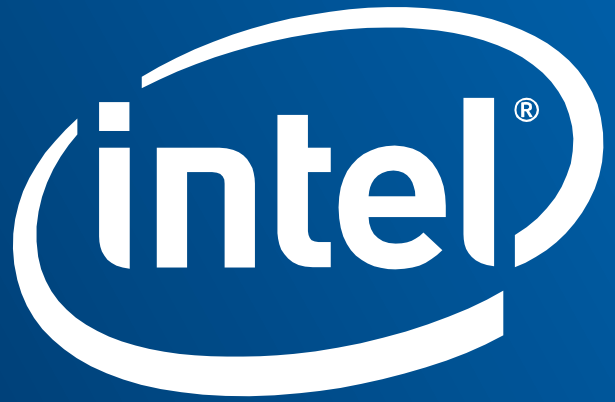
Legal Disclaimer & Optimization Notice

Software and workloads used in performance tests may have been optimized for performance only on Intel microprocessors. Performance tests, such as SYSmark and MobileMark, 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.

Optimization Notice

Intel’s compilers may or may not optimize to the same degree for non-Intel microprocessors for optimizations that are not unique to Intel microprocessors. These optimizations include SSE2, SSE3, and SSSE3 instruction sets and other optimizations. Intel does not guarantee the availability, functionality, or effectiveness of any optimization on microprocessors not manufactured by Intel. Microprocessor-dependent optimizations in this product are intended for use with Intel microprocessors. Certain optimizations not specific to Intel microarchitecture are reserved for Intel microprocessors. Please refer to the applicable product User and Reference Guides for more information regarding the specific instruction sets covered by this notice.

Notice revision #20110804


intel