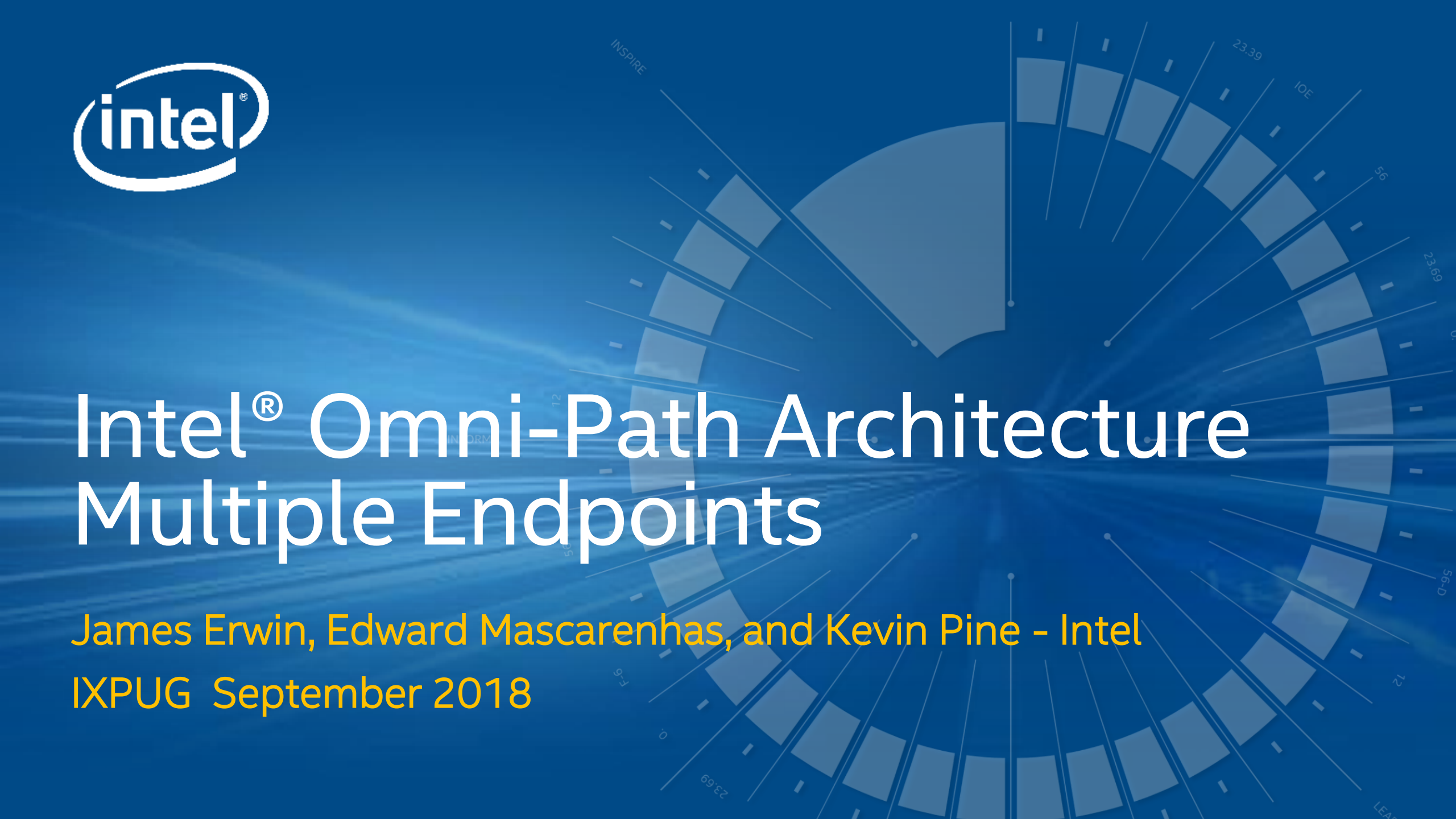


Intel® Omni-Path Architecture Multiple Endpoints

James Erwin, Edward Mascarenhas, and Kevin Pine - Intel

IXPUG September 2018

Outline

What is Multi-Endpoint (MEP)?

MEP Software Recipe

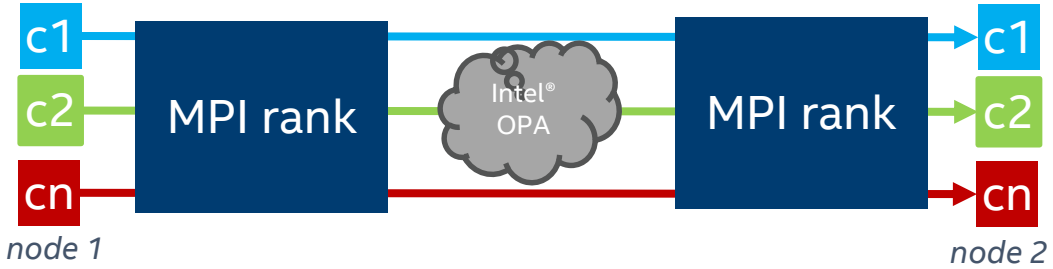
Simple Example for Allreduce

What is Multi-Endpoint (Multi-EP)?

- Hybrid MPI/OpenMP* codes are becoming more prevalent
  - Pure MPI applications are reaching limits of scalability
  - More MPI ranks = more surface to volume ratio and increased network transfer
- Multi-EP is the ability to use more than one thread per MPI rank
  - Enables MPI communication within OpenMP regions
  - Reduces fork-join[1] when MPI communication is required
  - Increases bandwidth per MPI rank because more physical cores are used to drive the communications

Traditional MPI

Multiple Endpoints



1. Mattson, T. and Meadows, L., "A 'Hands-on' Introduction to OpenMP," https://www.openmp.org/wp-content/uploads/omp-hands-on-SC08.pdf

*The OpenMP name and the OpenMP logo are registered trademarks of the OpenMP Architecture Review Board

Multi-EP Software Recipe

- Requires components from:
  - Intel® Omni-Path Fabric Suite Fabric Manager (IFS) version 10.5 or newer
  - OpenFabrics Interfaces (OFI) Libfabric version 1.5 or newer
  - Intel® MPI Library 2019 (or 2019 Technical Preview)
- Example execution, using 16 nodes, 1 MPI rank per node with 4 endpoints:
  - source /opt/intel/impi/2019.0.070/bin64/mpivars.sh release_mt -ofi_internal
  - export I_MPI_THREAD_SPLIT=1
  - export I_MPI_THREAD_RUNTIME=openmp
  - export PSM2_MULTI_EP=1
  - mpirun -np 16 -ppn 1 -hostfile 16nodes -genv I_MPI_FABRICS shm:ofi -genv OMP_NUM_THREADS=4 ./myapplication

PSM - Performance Scaled Messaging

Simple Example for Allreduce

```
#include <mpi.h>
#include <stdio.h>
#include <string.h>
#include <unistd.h>

int numThreads,nranks,myrank,size=1048576,niters=10240,i,ompsize
int provided; //used to confirm THREAD_MULTIPLE is supported

int main(int argc, char** argv) {

    // Initialize the MPI environment
    MPI_Init_thread(NULL, NULL, MPI_THREAD_MULTIPLE, &provided);

    // Get the number of processes
    MPI_Comm_size(MPI_COMM_WORLD, &nranks);

    // Get the rank of the process
    MPI_Comm_rank(MPI_COMM_WORLD, &myrank);
```

```
#pragma omp parallel
{
     numThreads = omp_get_num_threads();
}

    int x[size],y[size];
    ompsize=size/numThreads;

    for(i=0;i<size;++i) { x[i]=i; }

    MPI_Comm comm_mep[numThreads];
    for(int ip=0;ip<numThreads;++ip) {
       MPI_Comm_dup(MPI_COMM_WORLD,&(comm_mep[ip]));
    }
    memset(y,0,sizeof(y));

#pragma omp parallel
{
     for (int iter=0;iter<niters;++iter) {
        int ip=omp_get_thread_num();
        MPI_Allreduce(x+ip*ompsize,y+ip*ompsize, ompsize, MPI_INT,
                      MPI_SUM, comm_mep[ip]);
     }
}
    MPI_Finalize();
}
```

red indicates additions necessary for Multiple Endpoints

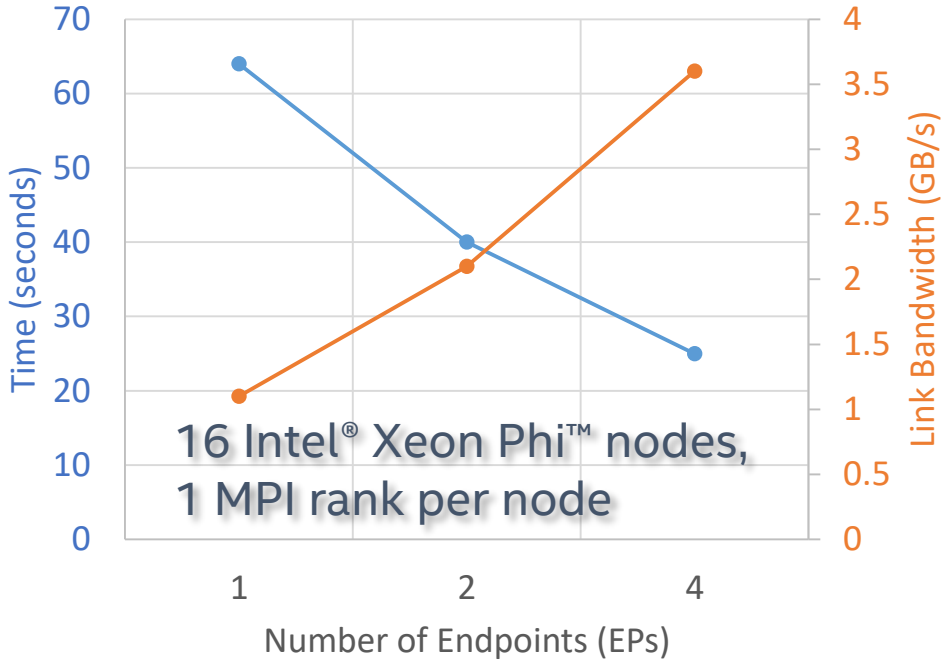
int x[size=1048576] array

4 End points

1 2 3 4 (ompsize=262,144 each)

Results & Conclusions

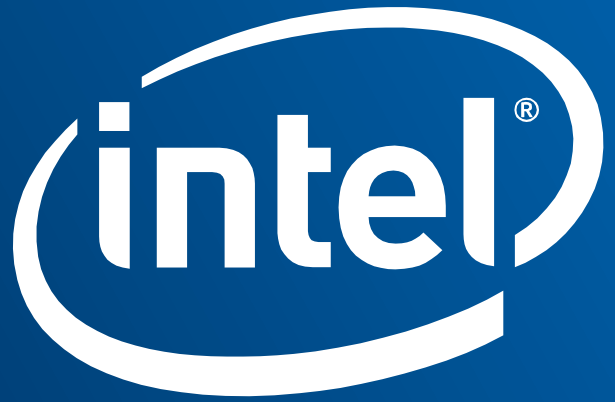
- **Wall clock time** is reduced:
  - 37% from 64 to 40 sec using 2 EPs
  - 60% from 64 to 25 sec using 4 EPs
- The **link bandwidth** of each node increases - higher network performance!
- Using Multi-EP with Intel® OPA only requires **a few easy steps**



- Multi-EP is being adopted by the industry to solve real world problems
  - "Multiple endpoints for improved MPI performance on a lattice QCD code" - https://dl.acm.org/citation.cfm?id=3176375
  - "Accelerating HPC codes on Intel(R) Omni-Path Architecture networks: From particle physics to Machine Learning" - http://arxiv.org/abs/1711.04883

Intel® Xeon Phi™ 7250 CPU. Intel® Turbo Boost and Hyper-Threading Technology enabled. Red Hat Enterprise Linux* Server release 7.4 (Maipo), Kernel: 3.10.0-693.21.1.el7.x86_64. BIOS: S72C610.86B.01.03.0018.012420182107, microcode: 0x1b6. CVE-2017-5753, CVE-2017-5715, and CVE-2017-5754 (Variants 1, 2, and 3) mitigated. Quadrant cluster mode, Flat memory mode. 16 GB MCDRAM, 96 GB DDR4 per node. Intel Fabric Suite (IFS 10.7.0.0.145). Intel® MPI Library 2019 Beta. source /opt/intel/impi/2019.0.070/bin64/mpivars.sh release_mt -ofi_internal; export I_MPI_THREAD_SPLIT=1 ; export I_MPI_THREAD_RUNTIME=openmp; mpirun -np 16 -ppn 1 -hostfile 16nodes -genv I_MPI_FABRICS shm:ofi -genv OMP_NUM_THREADS=4 -genv PSM2_MULTI_EP=1. Link bandwidth as reported by opatop tool for the highest utilized Intel® OPA Host Fabric Interface (HFI).

Performance results are based on testing as of Sept 20 2018 and may not reflect all publicly available security updates. See configuration disclosure for details. No product can be absolutely secure.



Thank you!

THIS SLIDE MUST BE USED WITH ANY SLIDES REMOVED FROM THIS PRESENTATION

NOTICES AND DISCLAIMERS

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

No computer system can be absolutely secure.

Tests document performance of components on a particular test, in specific systems. Differences in hardware, software, or configuration will affect actual performance. For more complete information about performance and benchmark results, visit http://www.intel.com/benchmarks .

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. For more complete information visit http://www.intel.com/benchmarks .

Intel® Advanced Vector Extensions (Intel® AVX)* provides higher throughput to certain processor operations. Due to varying processor power characteristics, utilizing AVX instructions may cause a) some parts to operate at less than the rated frequency and b) some parts with Intel® Turbo Boost Technology 2.0 to not achieve any or maximum turbo frequencies. Performance varies depending on hardware, software, and system configuration and you can learn more at http://www.intel.com/go/turbo.

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.

Cost reduction scenarios described are intended as examples of how a given Intel-based product, in the specified circumstances and configurations, may affect future costs and provide cost savings. Circumstances will vary. Intel does not guarantee any costs or cost reduction.

Intel does not control or audit third-party benchmark data or the web sites referenced in this document. You should visit the referenced web site and confirm whether referenced data are accurate.
Intel, the Intel logo, and Intel Xeon are trademarks of Intel Corporation in the U.S. and/or other countries.
*Other names and brands may be claimed as property of others.