
intel
INTEL®
EXPERIENCE
DAY

DISCLOSURES

Performance results are based on testing as of August and September 2019 and may not reflect all publicly available security updates. See configuration disclosure for details. No product can be absolutely secure.

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 www.intel.com/benchmarks.

Optimization Notice

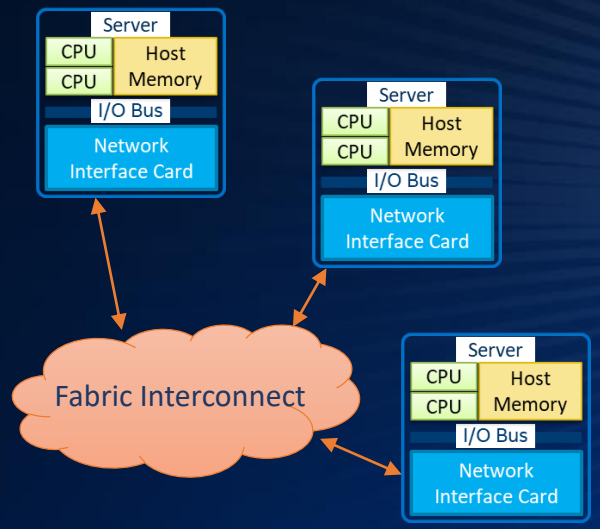
Intel's compilers may or may not optimize to the same degree for non-Intel microprocessors for optimizations that are not unique to Intel microprocessors. These optimizations include SSE2, SSE3, and SSSE3 instruction sets and other optimizations. Intel does not guarantee the availability, functionality, 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

HPC IN THE CLOUD

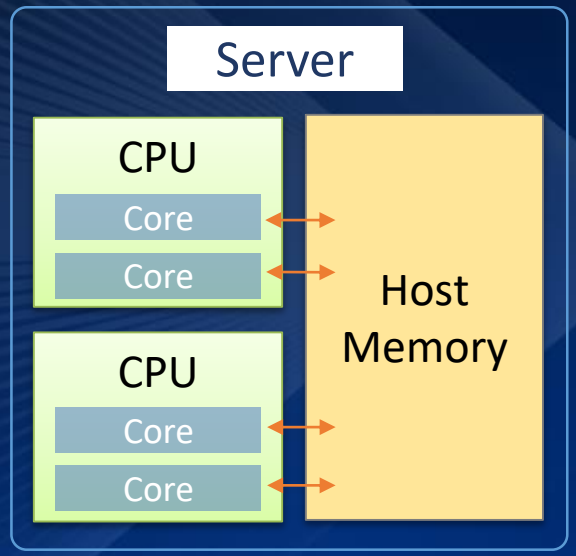
Новая архитектура Intel® MPI Library и интеграция с AWS

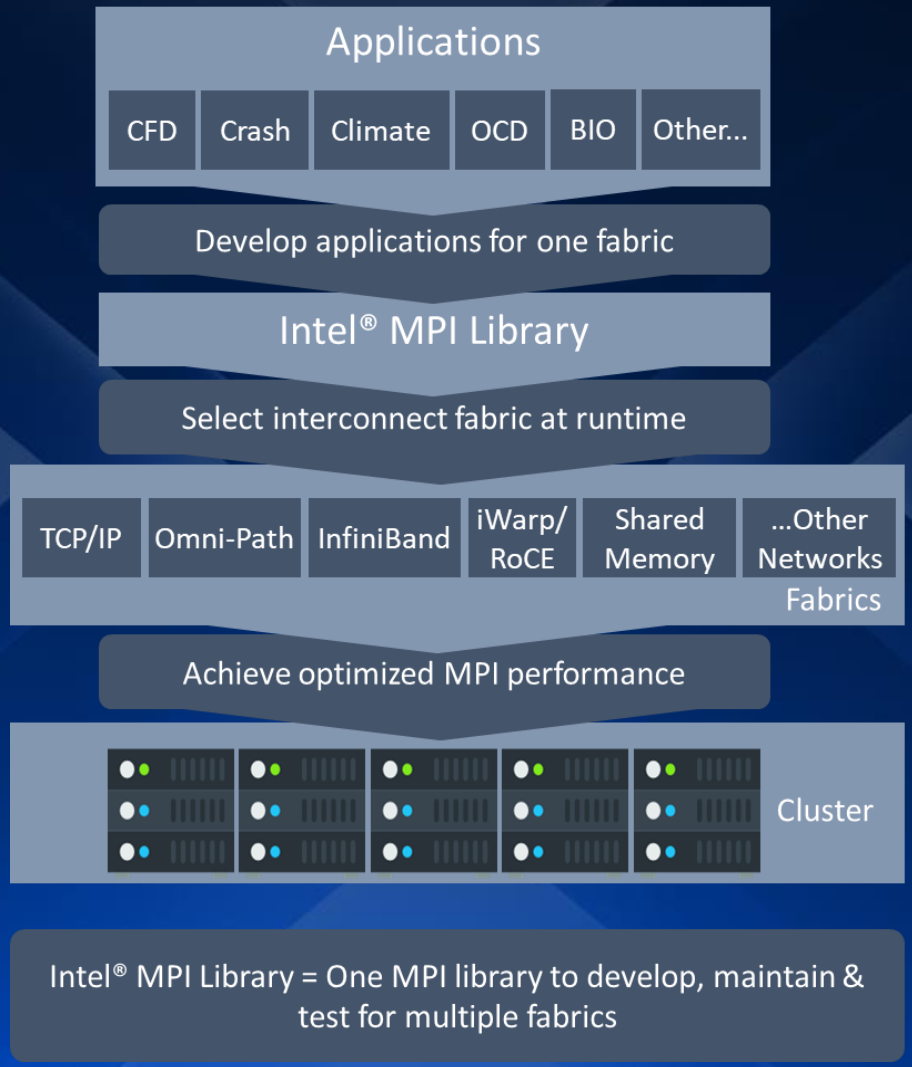
INTRODUCTION TO MPI

Traditional Parallel Cluster Distributed memory



Single-node Shared memory



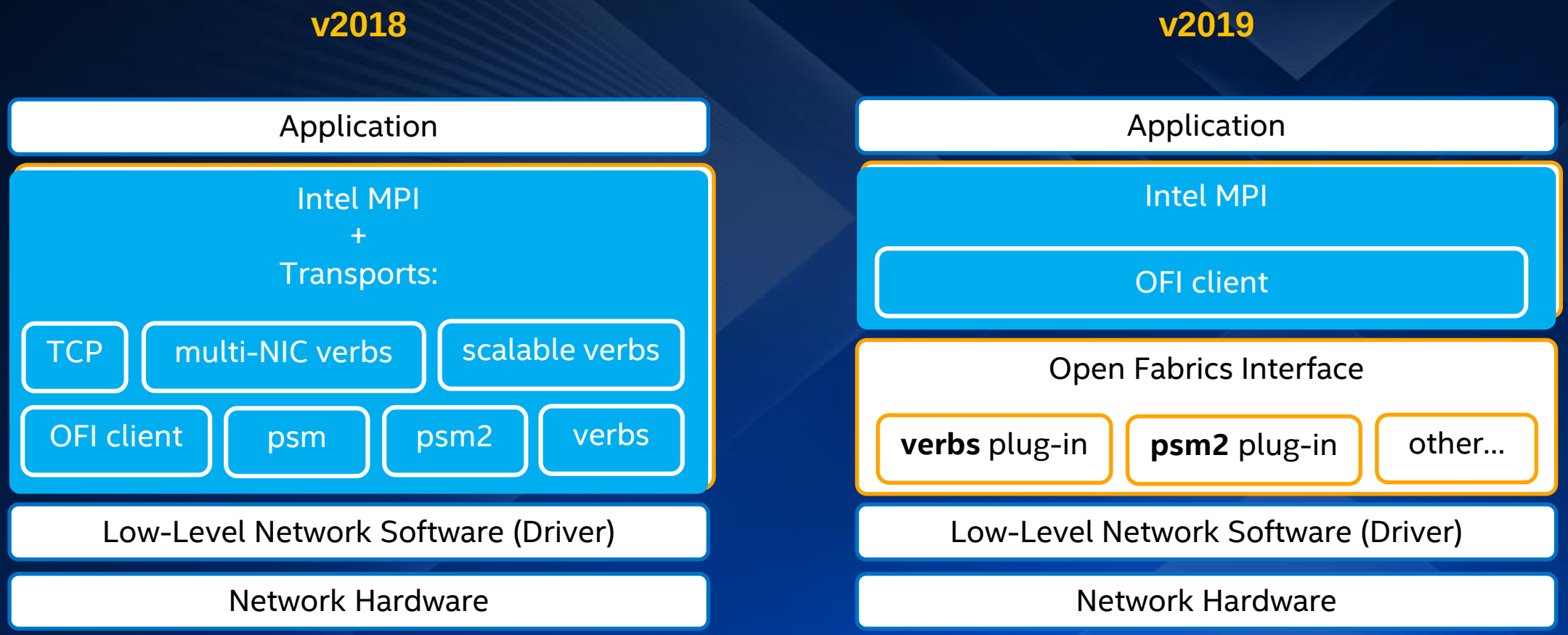


Fabric
Communication network designed to provide high-bandwidth and low-latency

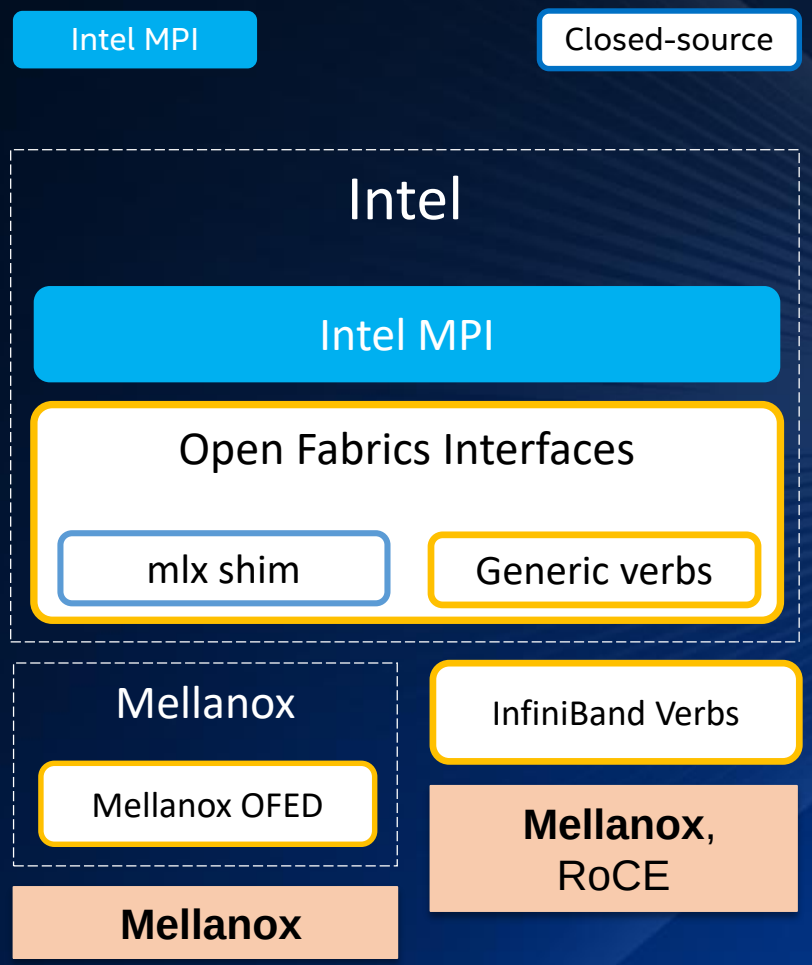
INTEL® MPI LIBRARY SOFTWARE ARCHITECTURE

Intel closed-source

Open-source



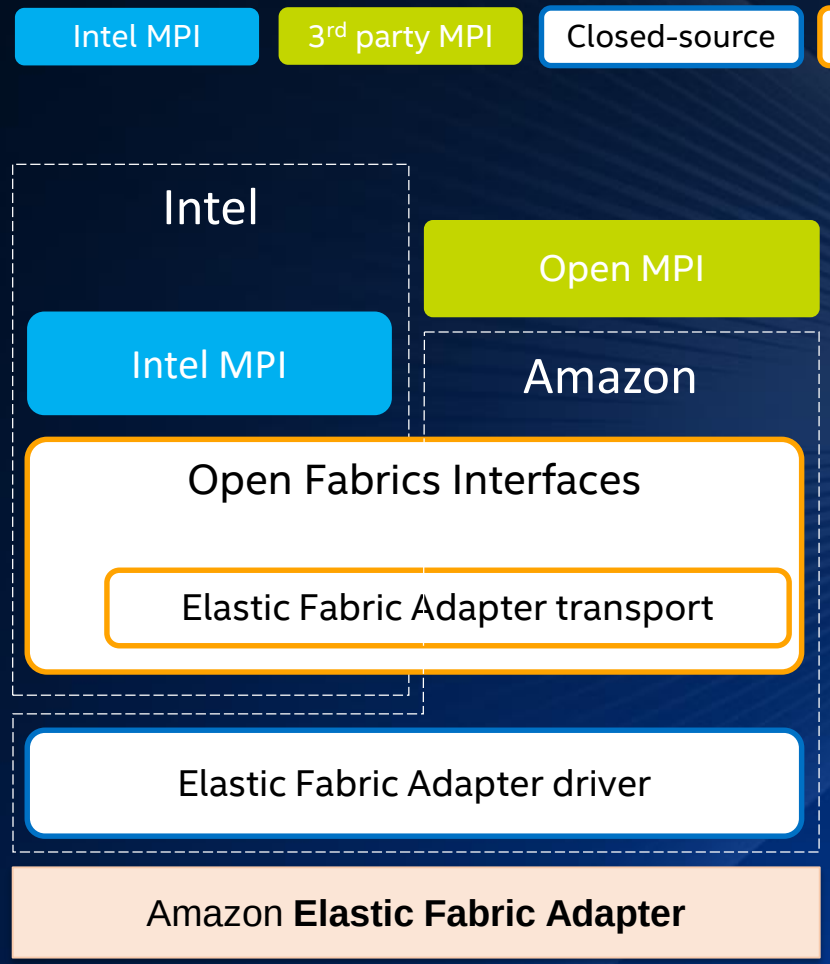
MELLANOX SUPPORT IMPROVEMENT



New OFI/mlx provider

- Part of IMPI 2019 U5 distribution (technical preview)
  - Available via **FI_PROVIDER=mlx**
- Default for Mellanox since IMPI 2019 U6
- Validated with Mellanox EDR/HDR
- Requires Mellanox OFED 4.5+

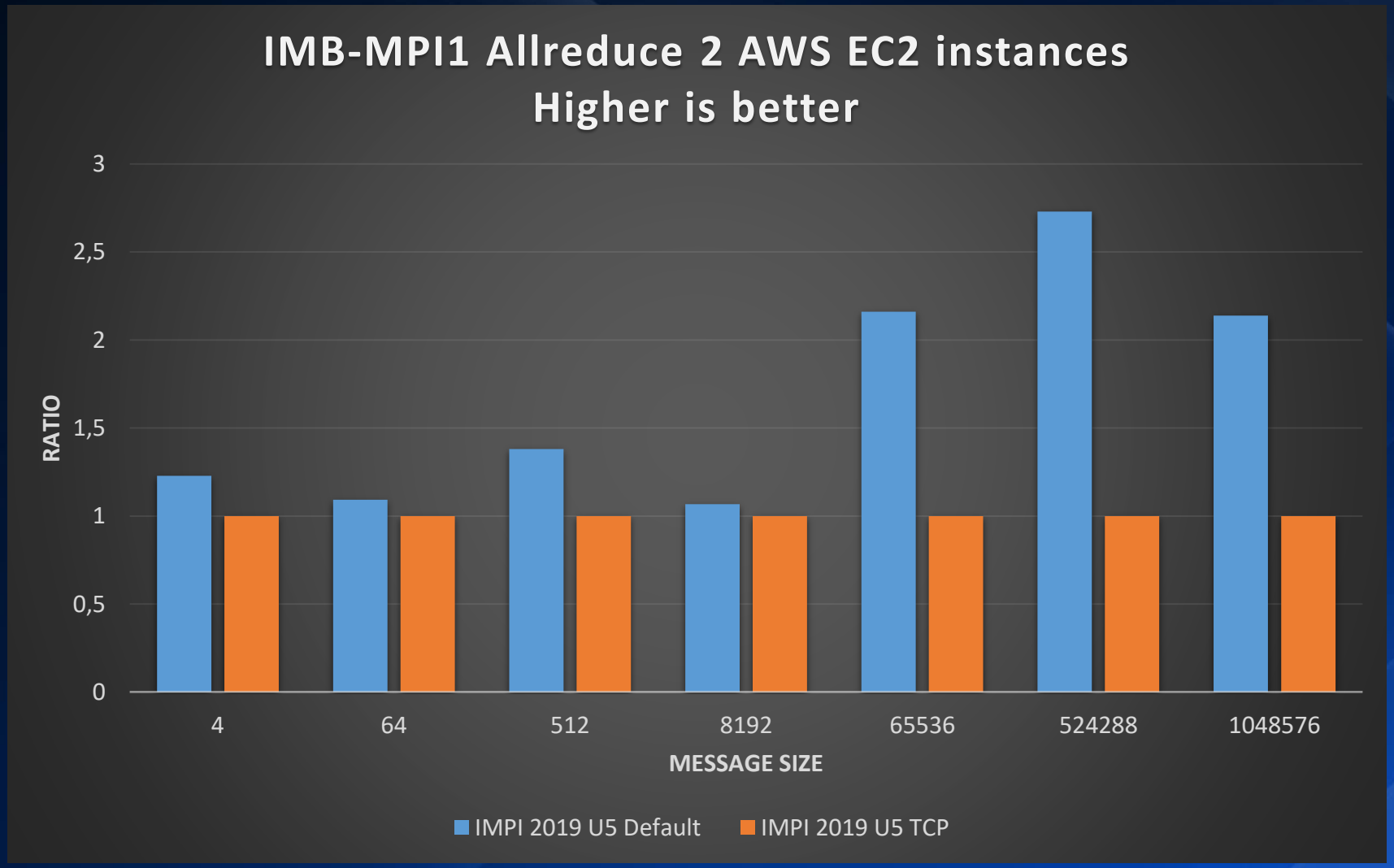
AMAZON AWS/EFA SUPPORT



- **New** OFI/efa provider
- IMPI 2019 U5
  - Performance tuning for EFA
  - Relies on AWS environment provided OFI/efa provider
- IMPI 2019 U6
  - Out of the box support of OFI/efa

https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/efa-start.html

AMAZON AWS/EFA PERFORMANCE. OFI/TCP VS OFI/EFA

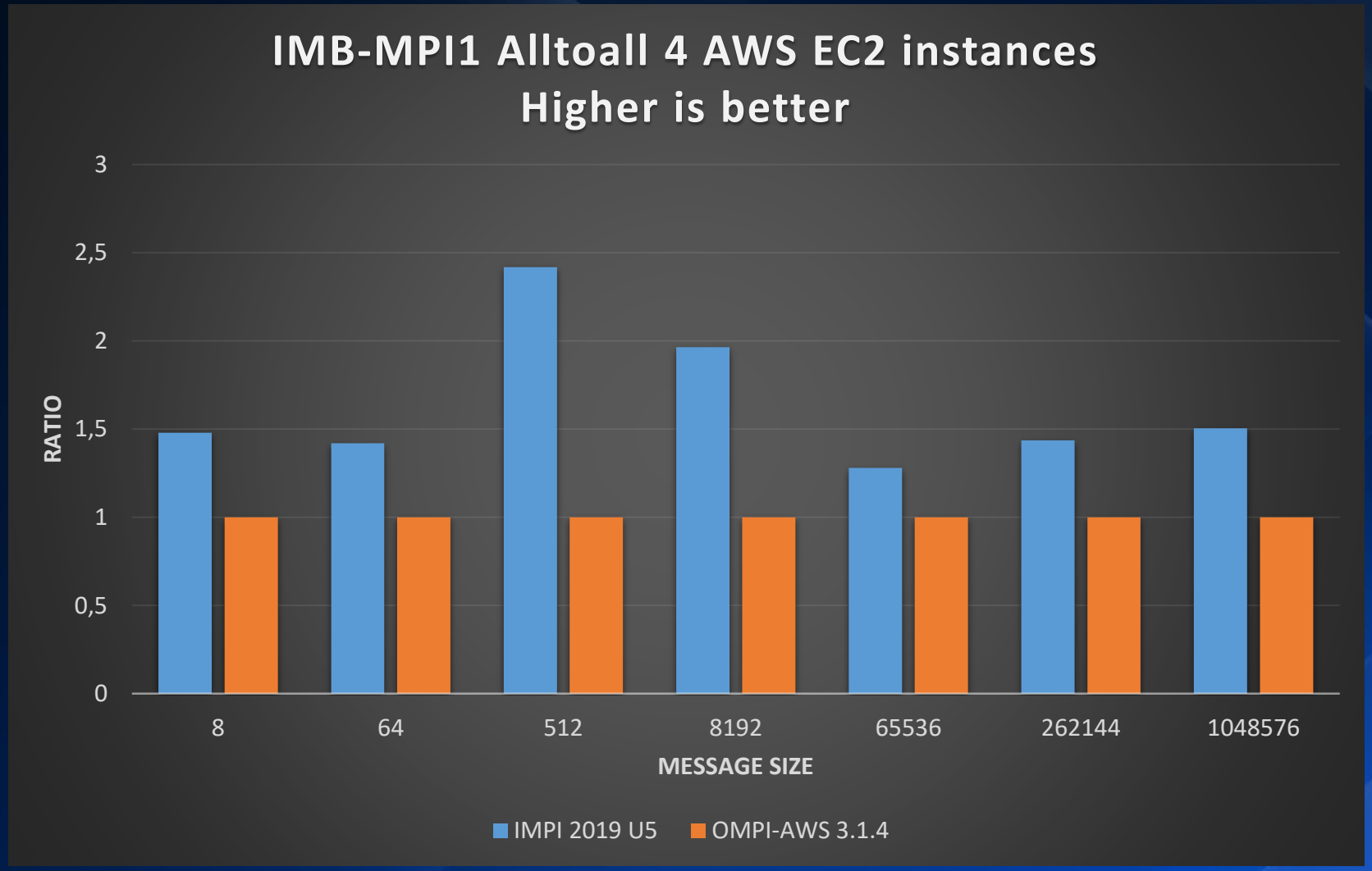


Performance results are based on testing as of June 2019 and may not reflect all publicly available security updates. See configuration disclosure for details. No product can be absolutely secure. Software and workloads used in performance tests may have been optimized for performance only on Intel microprocessors. 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.

Configuration:
Amazon Linux 2
c5n.18xlarge (278 ECUs, 72 vCPUs, 3 GHz, Intel Xeon Platinum 8124M, 192 GiB memory, Elastic Block Store, Elastic Fabric Adapter)
openmpi-3.1.4-2.amzn2.x86_64
Intel® MPI 2019 U5

https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/efa-start.html

AMAZON AWS/EFA PERFORMANCE

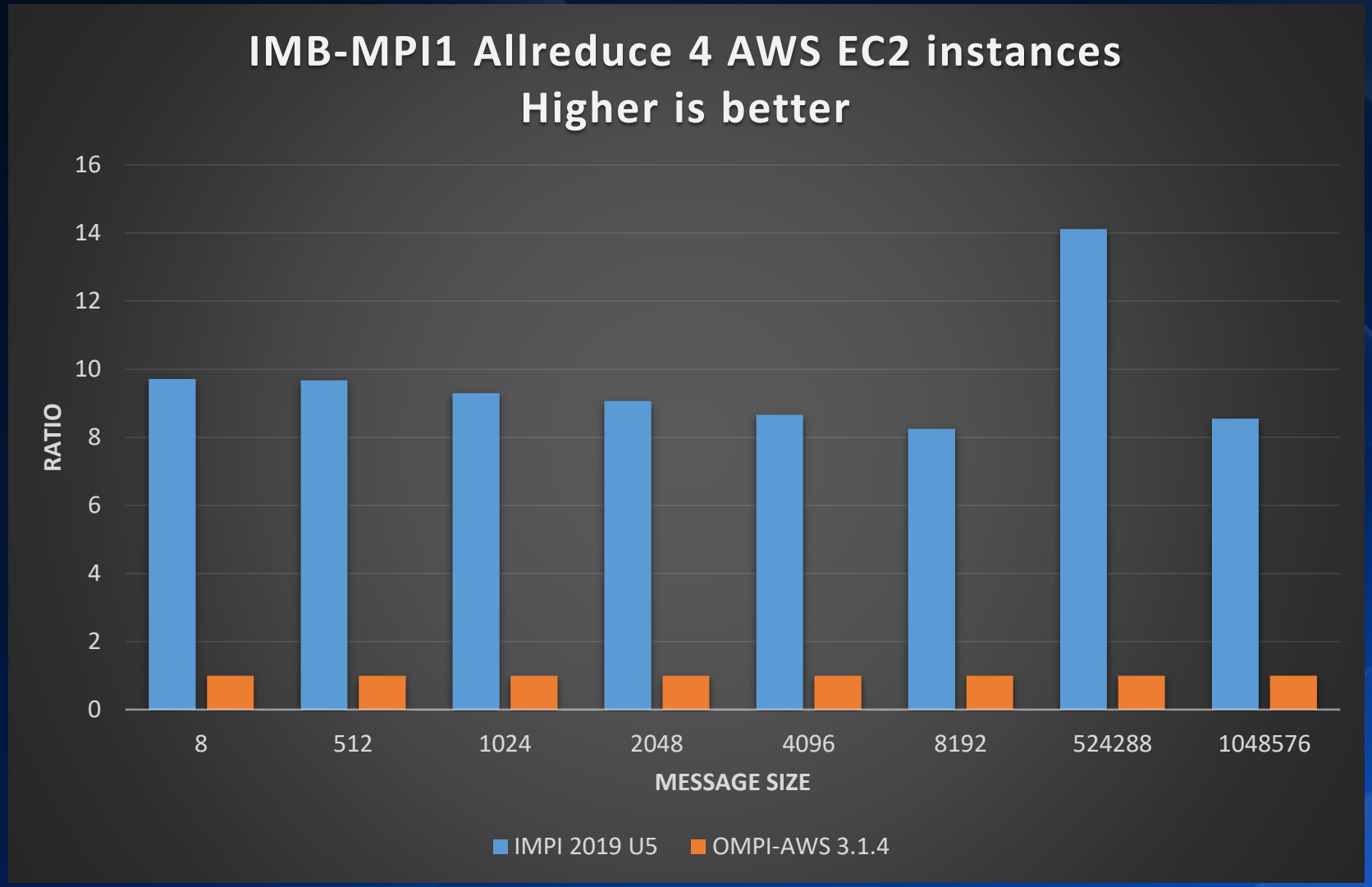


Performance results are based on testing as of June 2019 and may not reflect all publicly available security updates. See configuration disclosure for details. No product can be absolutely secure. Software and workloads used in performance tests may have been optimized for performance only on Intel microprocessors. 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.

Configuration:
Amazon Linux 2
c5n.18xlarge (278 ECUs, 72 vCPUs, 3 GHz, Intel Xeon Platinum 8124M, 192 GiB memory, Elastic Block Store, Elastic Fabric Adapter)
openmpi-3.1.4-2.amzn2.x86_64
Intel® MPI 2019 U5

https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/efa-start.html

AMAZON AWS/EFA PERFORMANCE



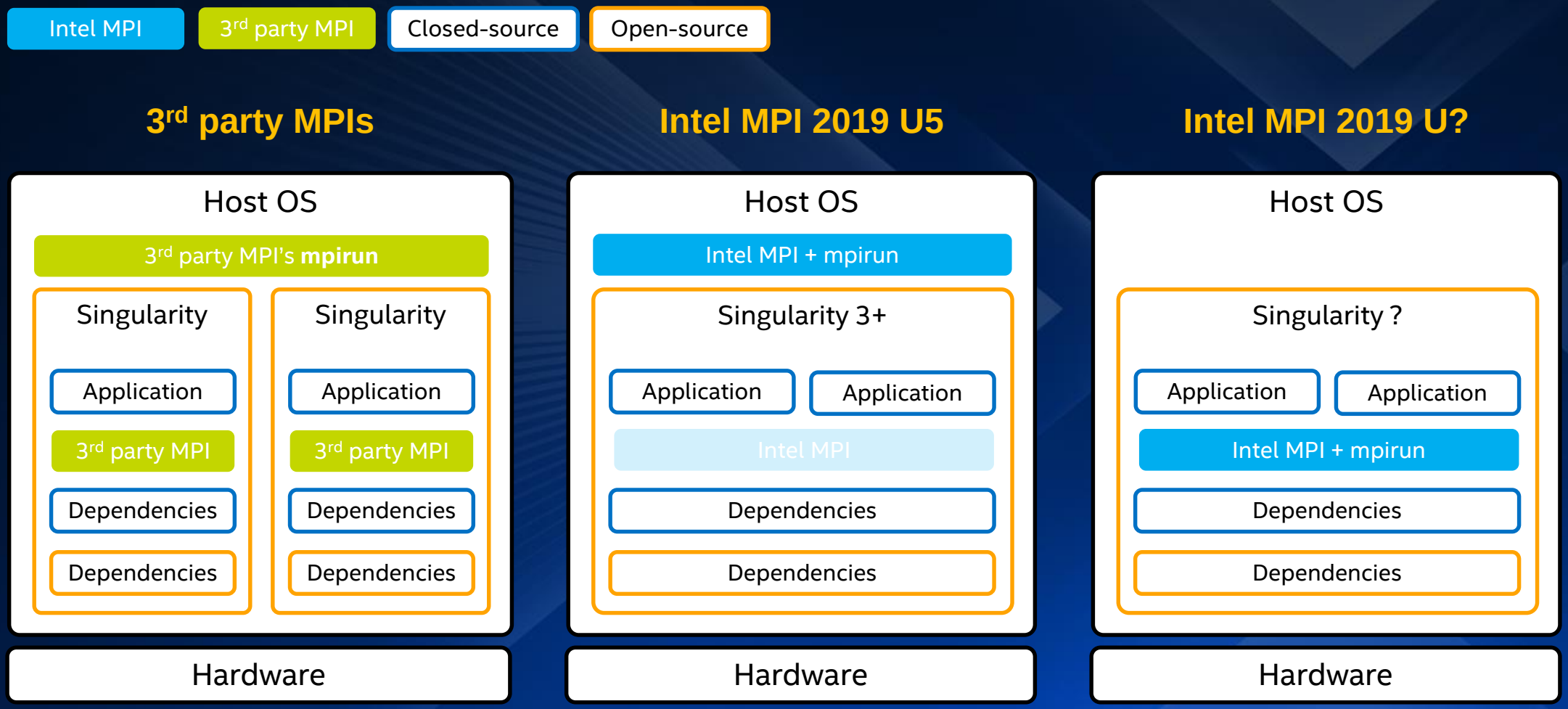
Performance results are based on testing as of June 2019 and may not reflect all publicly available security updates. See configuration disclosure for details. No product can be absolutely secure. Software and workloads used in performance tests may have been optimized for performance only on Intel microprocessors. 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.

Configuration:
Amazon Linux 2
c5n.18xlarge (278 ECUs, 72 vCPUs, 3 GHz, Intel Xeon Platinum 8124M, 192 GiB memory, Elastic Block Store, Elastic Fabric Adapter)
openmpi-3.1.4-2.amzn2.x86_64
Intel® MPI 2019 U5

https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/efa-start.html

SINGULARITY CONTAINER SUPPORT

Intel MPI | 3rd party MPI | Closed-source | Open-source



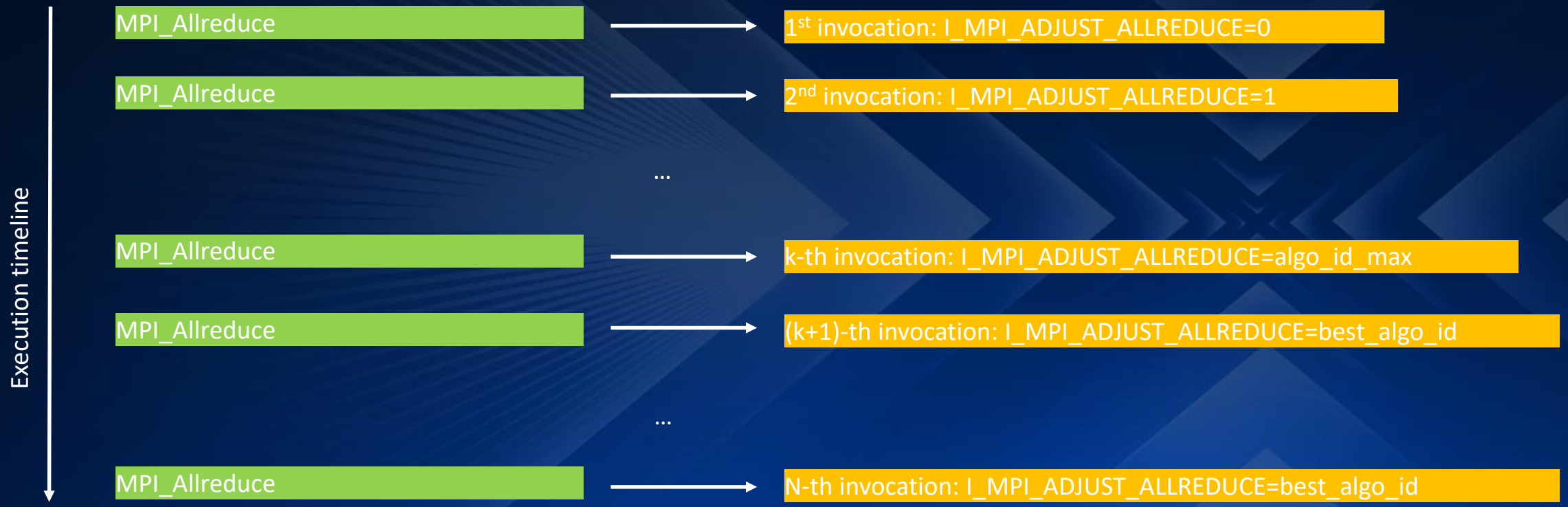
USABILITY FEATURES

- **New** auto tuning capability (**autotuner**)
- **New** spell checker logic
- **New** impi_info tool (MPI_T based)

```
$ I_MPI_PIN_DMAIN=socket mpirun -hosts host01,host02 -n 2 -ppn 1
IMB-MPI1 barrier
[0] MPI startup(): I_MPI_PIN_DMAIN environment variable is not
supported.
[0] MPI startup(): Similar variables:
        I_MPI_PIN_UNIT
        I_MPI_PIN
        I_MPI_PIN_DOMAIN
[0] MPI startup(): To check the list of supported variables, use
the impi_info utility or refer to https://software.intel.com/en-
us/mpi-library/documentation/get-started.
```

```
$ impi_info | head -10
 | NAME                                          | DEFAULT VALUE | DATA TYPE |
 ================================================================================
 |I_MPI_PIN                                      | on            | MPI_CHAR  |
 |I_MPI_PIN_SHOW_REAL_MASK                       | on            | MPI_INT   |
 |I_MPI_PIN_PROCESSOR_LIST                       | not defined   | MPI_CHAR  |
 |I_MPI_PIN_PROCESSOR_EXCLUDE_LIST               | not defined   | MPI_CHAR  |
 |I_MPI_PIN_CELL                                 | unit          | MPI_CHAR  |
 |I_MPI_PIN_RESPECT_CPUSET                       | on            | MPI_CHAR  |
 |I_MPI_PIN_RESPECT_HCA                          | on            | MPI_CHAR  |
 |I_MPI_PIN_DOMAIN                               | auto:compact  | MPI_CHAR  |
```

AUTOTUNER – APPLICATION DRIVEN TUNING



GET STARTED WITH AUTOTUNER

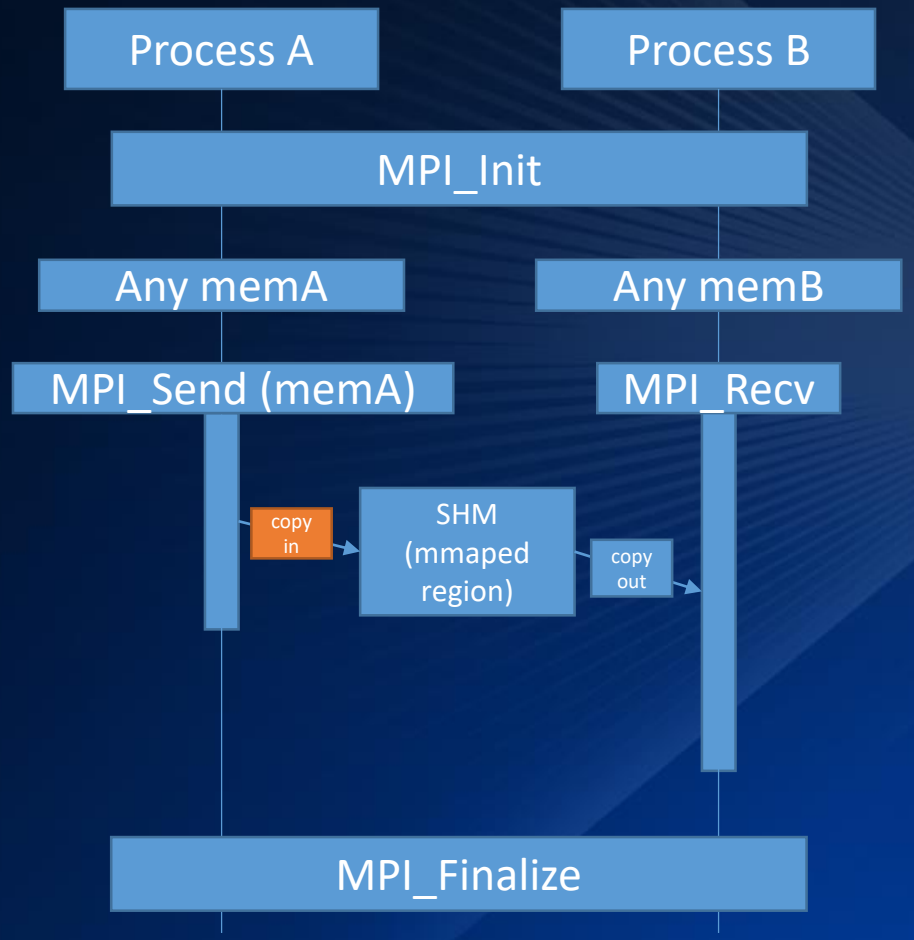
Step 1 – Enable autotuner and store results (store is optional):

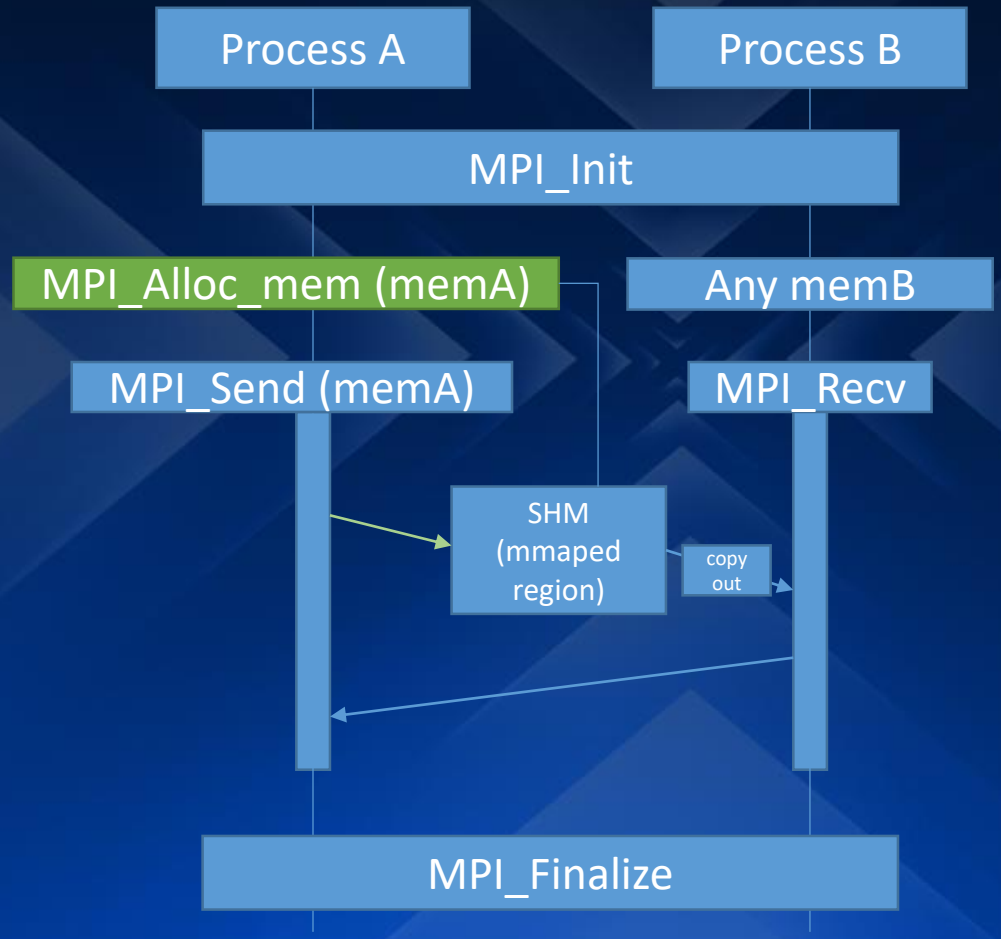
```
$ export I_MPI_TUNING_MODE=auto
$ export I_MPI_TUNING_BIN_DUMP=./tuning_results.dat
$ mpirun -n 96 -ppn 48 IMB-MPI1 allreduce -iter 1000,800 -time 4800
```

Step 2 – Use the results of autotuner for consecutive launches (optional):

```
$ export I_MPI_TUNING_BIN=./tuning_results.dat
$ mpirun -n 96 -ppn 48 IMB-MPI1 allreduce -iter 1000,800 -time 4800
```

SHM HEAP OVERVIEW





GET STARTED WITH SHM HEAP

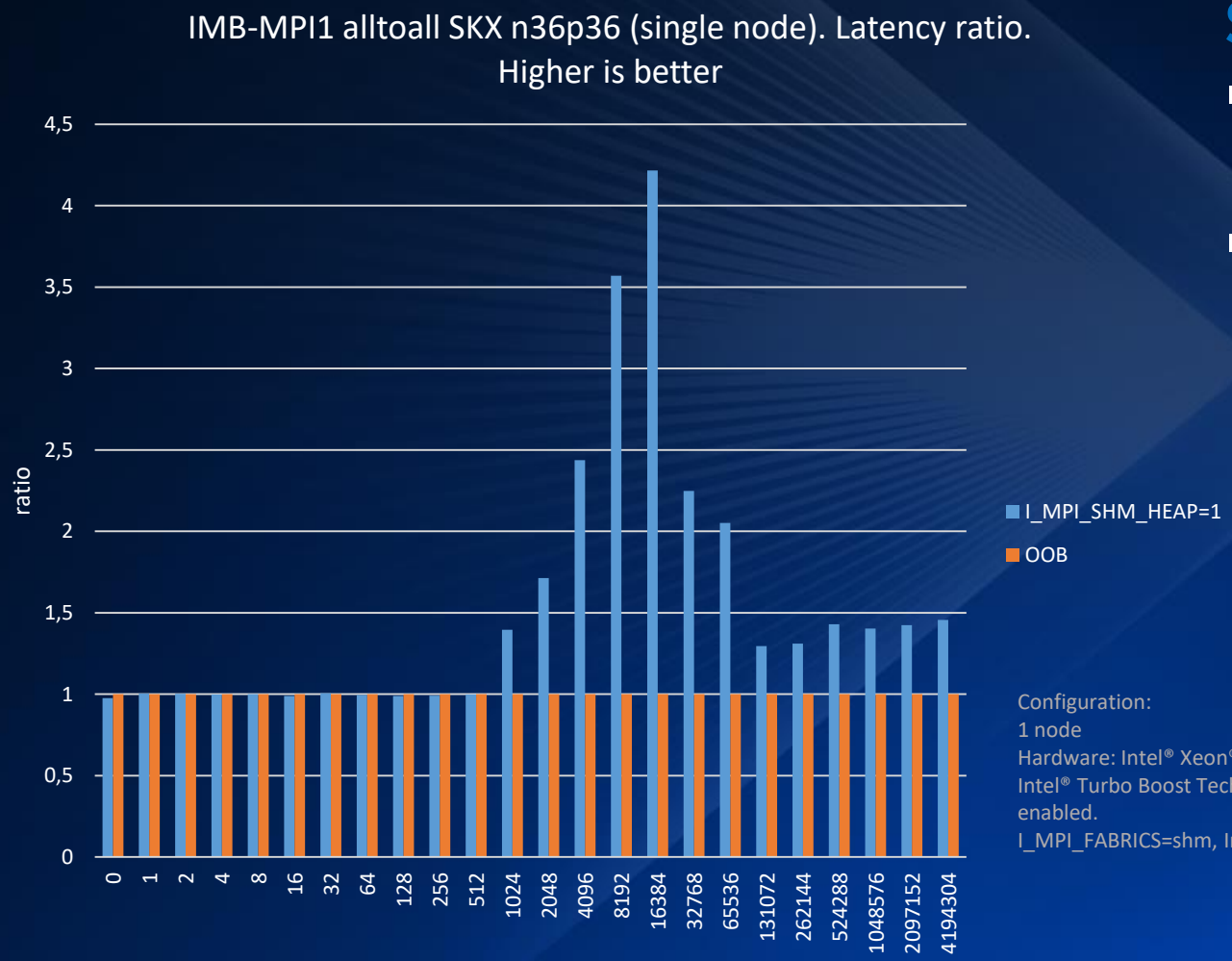
Basic way 1 (application w/ MPI_Alloc_mem) – Enable SHM HEAP

```
$ export I_MPI_SHM_HEAP=1
$ mpirun -n 36 -ppn 36 IMB-MPI1 alltoall -iter 1000,800 -time 4800
```

Basic way 2 (application w/o MPI_Alloc_mem) – Use proxy library to replace malloc with MPI_Alloc_mem:

```
$ export I_MPI_SHM_HEAP=1
$ export LD_PRELOAD=$I_MPI_ROOT/intel64/lib/libmpi_shm_heap_proxy.so
$ mpirun -n 36 -ppn 36 IMB-MPI1 alltoall -iter 1000,800 -time 4800
```

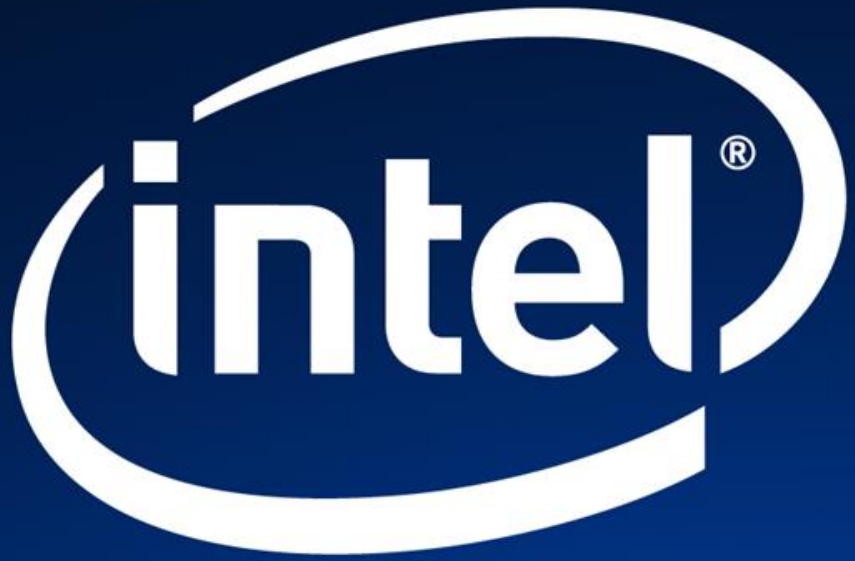
SHM HEAP



SHM pt2pt acceleration

- **Removes copy-in** phase from pt2pt communication
- **Efficient memory allocation** mechanism

Configuration:
1 node
Hardware: Intel® Xeon® Gold 6140 CPU @ 2.30GHz; 192 GB RAM.
Intel® Turbo Boost Technology and Hyperthreading Technology enabled.
I_MPI_FABRICS=shm, Intel® MPI Library 2019 Update 5


intel