Current State of the Cray MPT Software Stacks on the Cray XC Series Supercomputers

May 10, 2017

Krishna Kandalla, Peter Mendygral, Nick Radcliffe, Bob Cernohous, Naveen Namashivayam, Kim McMahon, Christopher Sadlo and Mark Pagel

(**kkandalla**, pjm, nradcliff,bcernohous,nravi,kmcmahon,csadlo,pags)@cray.com

Agenda

- **Introduction & Motivation**
- Key Features and Optimization
- New and Upcoming Features in Cray MPT
- Q&A

Introduction & Motivation

- Intel KNL offers at least 64 cores per node, more than 2 TF double precision performance per chip
  Different from Xeon – wider vectors, slower cores, slower scalar processing

- KNL offers MCDRAM: On package High Bandwidth memory
  Software support necessary to manage specialized memory (such as huge page backed memory) on MCDRAM.

- MPI and SHMEM are popular parallel programming models.
  Implementations must be optimized and tuned carefully for the KNL architecture

- This talk summarizes some of the new features and optimizations in Cray MPT for current generation Cray XC Systems

- Cray MPT comprises of Cray MPI and Cray SHMEM software stacks that are highly tuned for XC Systems.

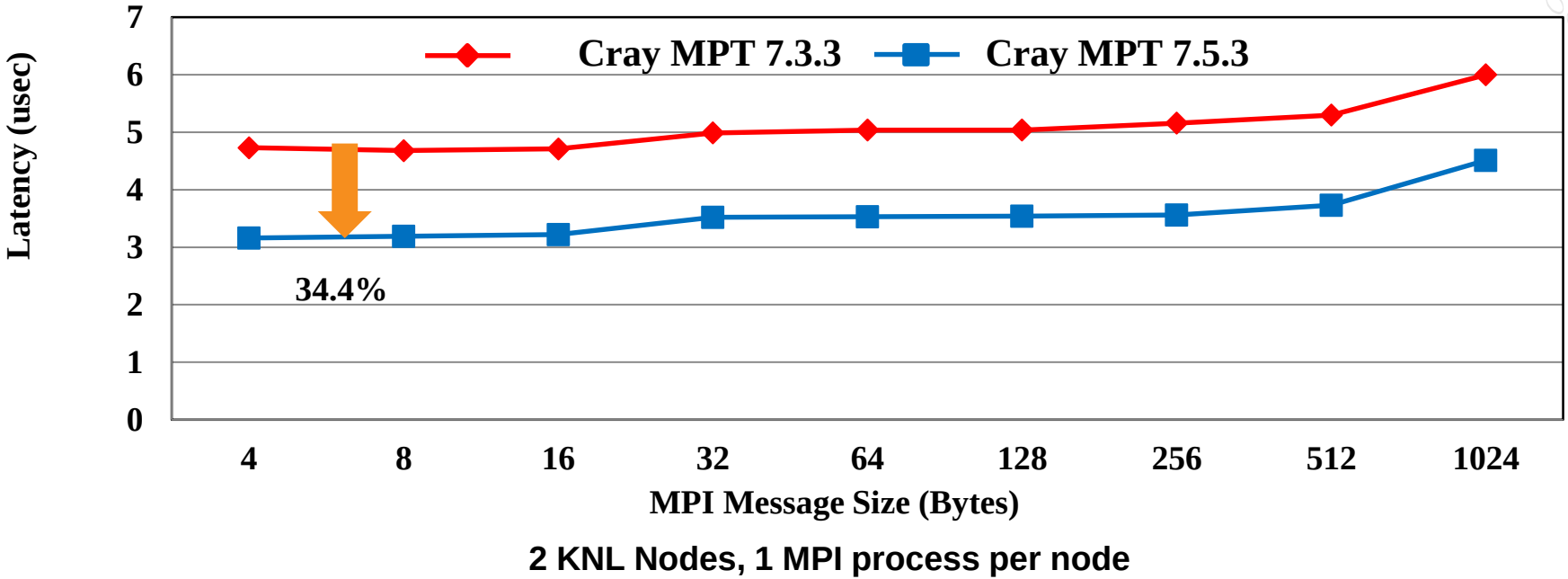
Agenda

- Introduction & Motivation
- **Key Features and Optimizations**
- New and Upcoming features in Cray MPT
- Q&A

Key Features And Optimizations

- **New Optimizations in Cray MPI to improve performance of point-to-point and collective operations for XC systems with KNL**
- **Performance and API Enhancements in Cray SHMEM**
- **New features in Cray MPI and Cray SHMEM to improve support for MCDRAM utilization**
- **Improved support for MPI_THREAD_MULTIPLE in Cray MPT**
  - Enhanced “Thread Hot” MPI-3 RMA capabilities on XC system with KNL
  - New locking impl. to improve multi-threaded pt2pt operations
- **Application-level performance studies on KNL**
  - WOMBAT and SNAP
- **Upcoming features in Cray MPT**

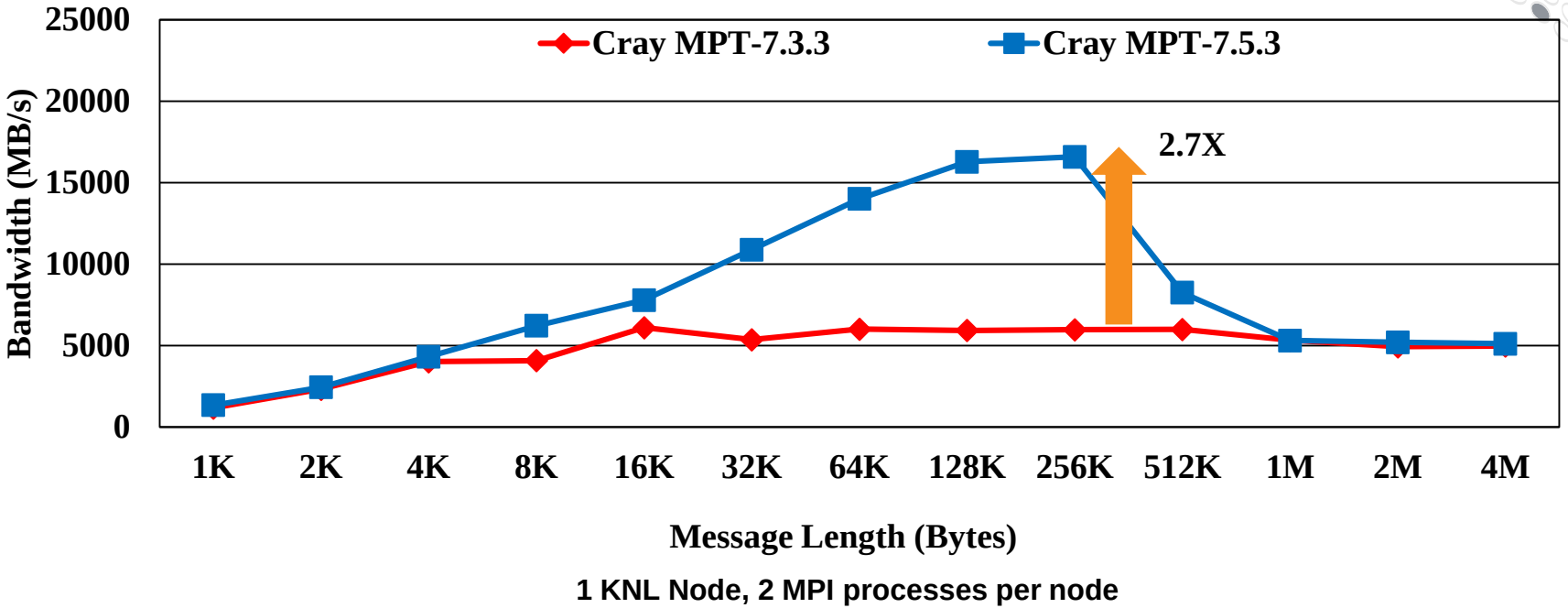
MPI Off-node Pt-2-Pt latency on XC (KNL)



2 KNL Nodes, 1 MPI process per node

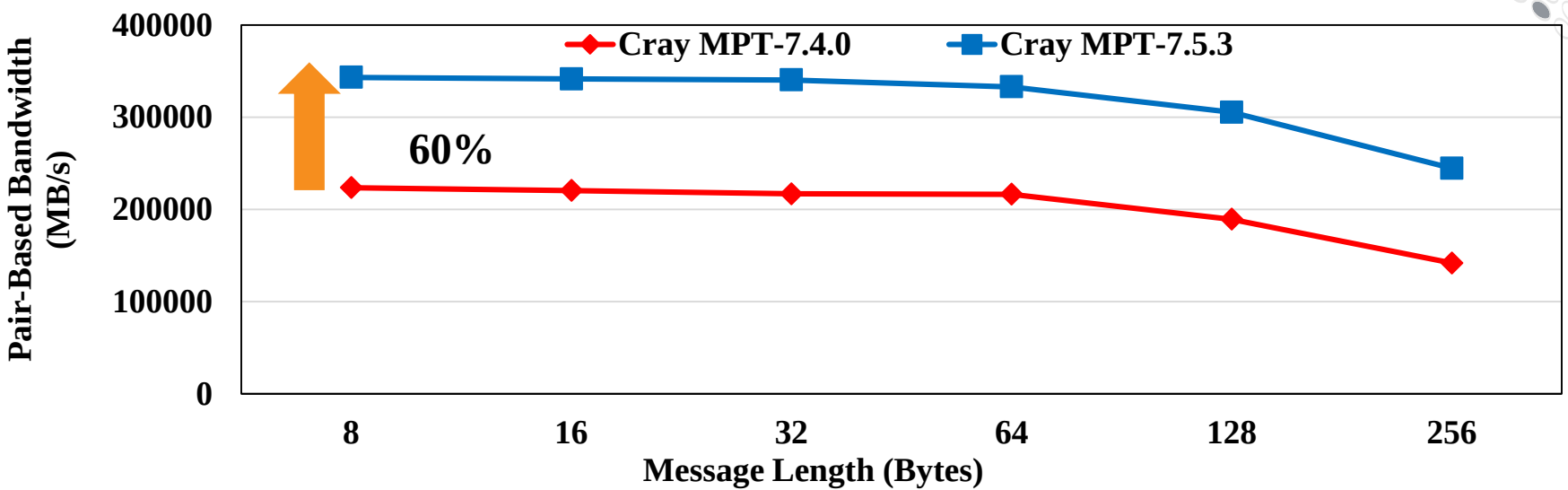
Cray MPT-7.5.3 has been optimized to simplify polling logic and reduce the number of instructions in the critical code-paths resulting in latency improvements up to **34%**

MPI On-node Pt-2-Pt Bandwidth on XC (KNL)



Cray MPT-7.5.3 relies on a new *memcpy*() implementation that is specifically tuned for the KNL processor. Improves on-node bandwidth by up to **2.7X**

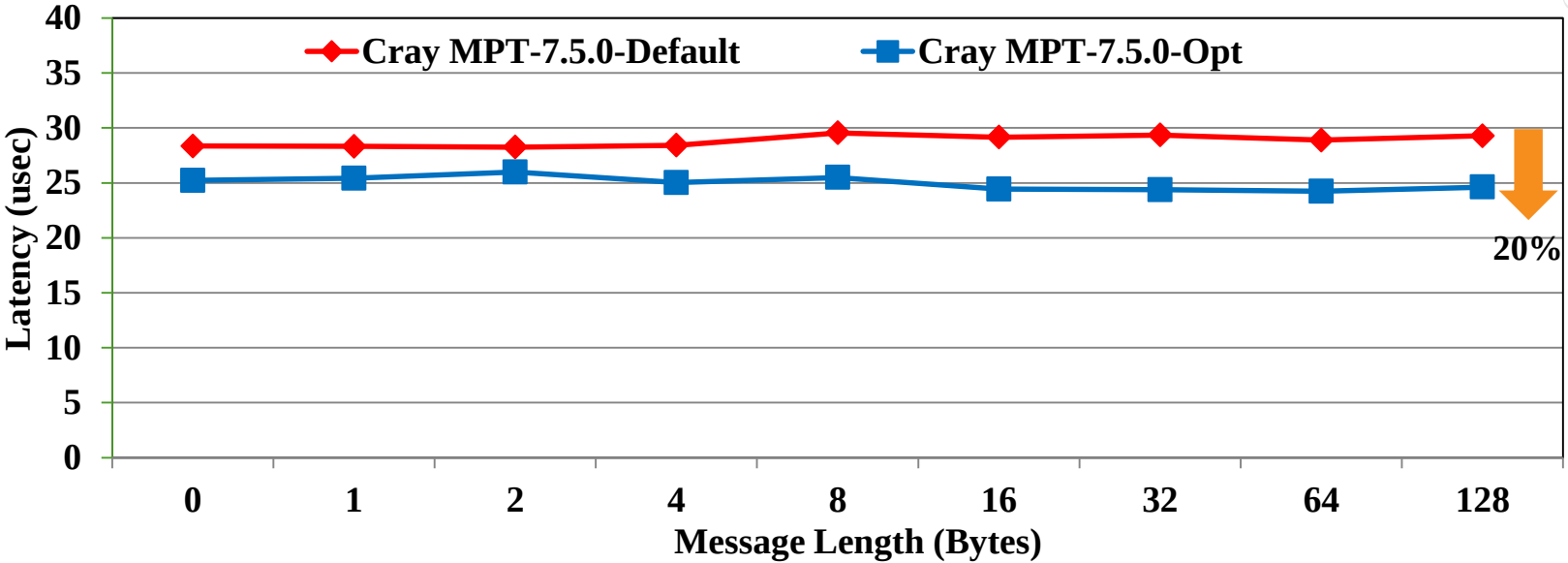
SMB Pair-Based Bandwidth with Cray MPI on XC (KNL)



32 KNL Nodes, 64 MPI processes per node (2,048 MPI Processes)
6 communicating pairs per process, craype-hugepages8M

Cray MPT-7.5.3 also outperforms Cray MPT-7.4.0 with the SMB Message Rate Benchmark on XC (KNL) systems by up to **60%**

Optimized MPI_Bcast on XC (KNL)



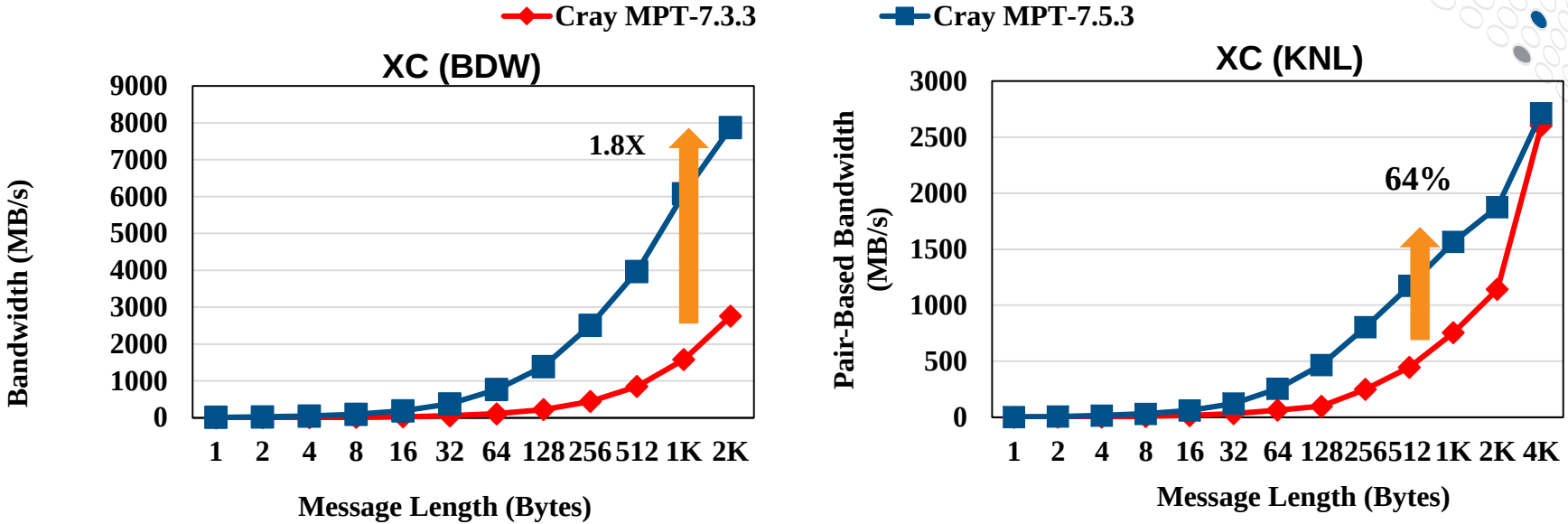
64 KNL Nodes, 64 MPI processes per node (4,096 MPI Processes)

Cray MPT-7.5.0 offers a new (non-default) optimization to improve the average communication latency reported by the osu_bcast.c benchmark by up to **20%** (MPICH_NETWORK_BUFFER_COLL_OPT = 0/1)

Key Features And Optimizations

- New Optimizations in Cray MPI to improve performance of point-to-point and collective operations for XC systems with KNL
- **Performance and API Enhancements in Cray SHMEM**
- **New features in Cray MPI and Cray SHMEM to improve support for MCDRAM utilization**
- **Improved support for MPI_THREAD_MULTIPLE in Cray MPT**
  - Enhanced “Thread Hot” MPI-3 RMA capabilities on XC system with KNL
  - New locking impl. to improve multi-threaded pt2pt operations
- **Application-level performance studies on KNL**
  - WOMBAT and SNAP
- **Upcoming features in Cray MPT**

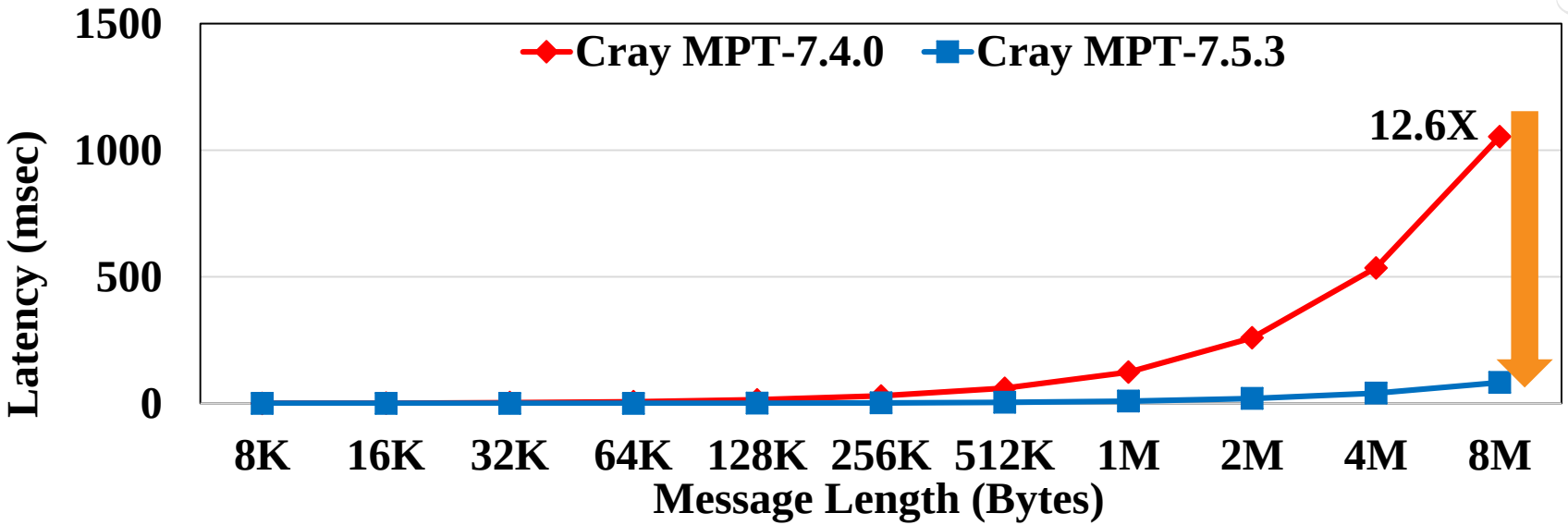
Cray SHMEM Bi-directional Put Bandwidth on XC



2 SHMEM PEs on two nodes. SOS Bi-Directional Write Bandwidth Benchmark

In Cray MPT-7.3.3 Shmem_put returns only after data is copied into the remote buffer
Cray MPT-7.5.3 is consistent with the current OpenSHMEM specification
(SHMEM_DMAPP_PUT_NBI=0 if the behavior in Cray SHMEM 7.3.3 is desired)

Cray SHMEM_Reduce on XC (KNL)



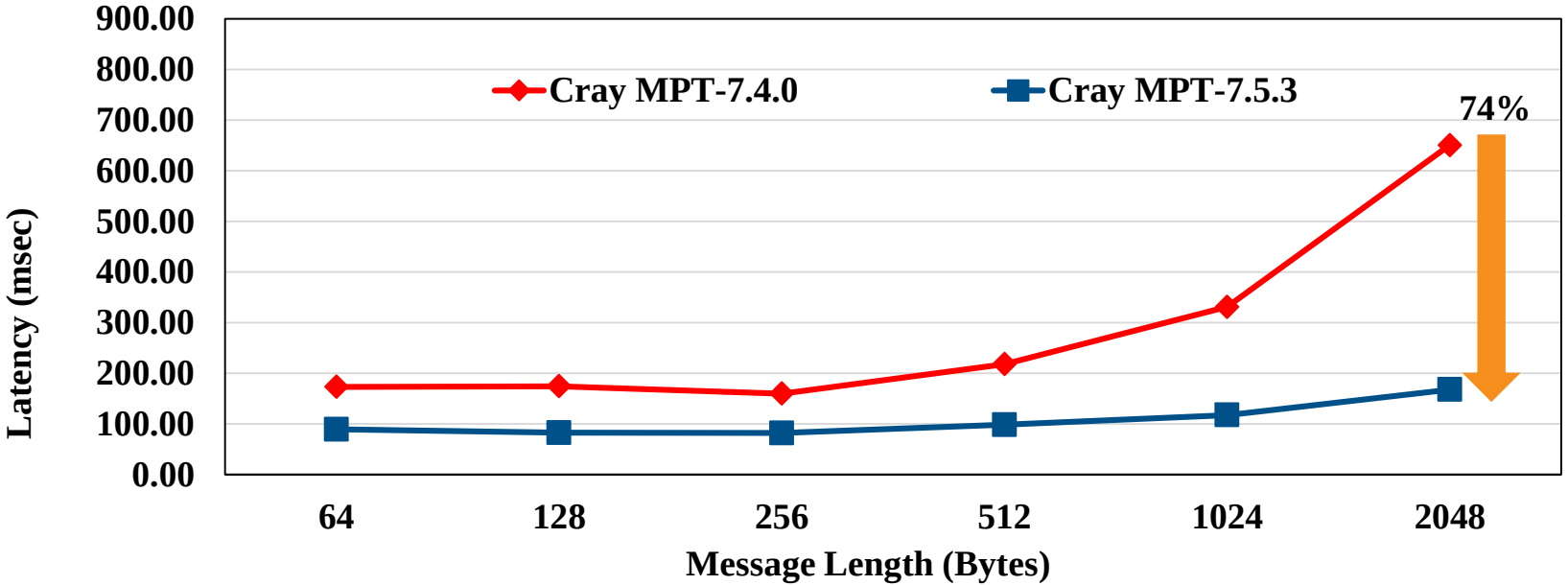
18,000 SHMEM PEs on 500 KNL nodes. PGAS All-Reduce micro-benchmark

Cray SHMEM optimizes *Active Set* based All-reduce for large data sizes by up to **12.6X**

SHMEM_USE_LARGE_OPT_REDUCE=0/1 (Default: 0)

SHMEM_REDUCE_CUTOFF_SIZE (Default: 16384)

SHMEM Team-Based Reduction on Cray XC (KNL)



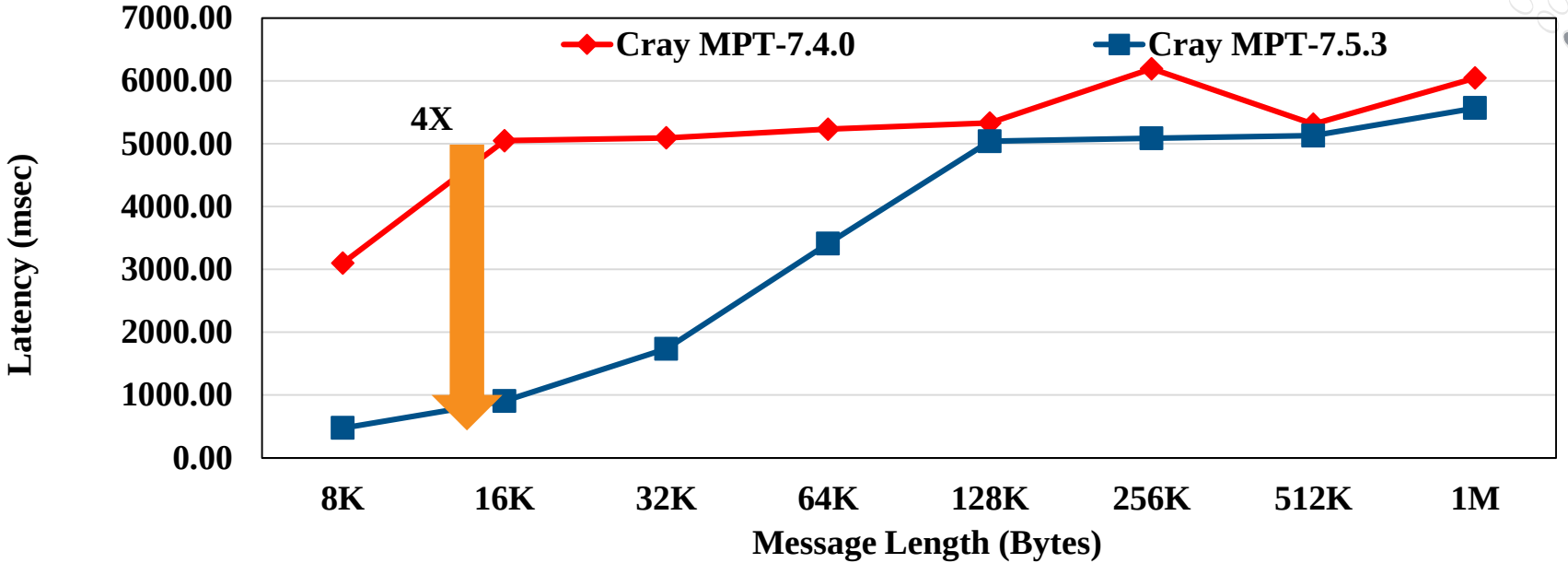
18,000 SHMEM PEs on 500 KNL nodes (36 Pes per node). PGAS micro-benchmark

Cray SHMEM 7.5.3 improves Team-based reduction operations via SMP optimizations

SHMEM_TEAM_SMP_REDUCE = 0/1 (Default: 0)

Cray SHMEM 7.5.3 is up to **74%** faster than Cray SHMEM 7.4.0 for small messages

SHMEM Team-Based Reduction on Cray XC (KNL)



18,000 SHMEM PEs on 500 KNL nodes (36 Pes per node). PGAS micro-benchmark

Cray SHMEM 7.5.3 improves Team-based reduction operations via SMP optimizations
SHMEM_TEAM_SMP_REDUCE = 0/1 (Default: 0)
Cray SHMEM 7.5.3 is up to **4X** faster than Cray SHMEM 7.4.0 for medium length msgs

API Extensions in Cray SHMEM

- Using a basic *Active Set* for work decomposition is not sufficient
- Apart from the existing color- and stride-based Team creation routines, Cray SHMEM supports creation of two- and three-dimensional Cartesian based Team Splits

2-Dimensional Cartesian splits	`shmemx_team_split_2d (`
	`shmem_team_t parent_team,`
	`int xaxis_range, int yaxis_range,`
	`shmem_team_t* xaxis_team,`
	`shmem_team_t* yaxis_team )`

3-Dimensional Cartesian splits	`shmemx_team_split_3d (`
	`shmem_team_t parent_team,`
	`int xaxis_range, int yaxis_range,`
	`int zaxis_range,`
	`shmem_team_t* xaxis_team,`
	`shmem_team_t* yaxis_team,`
	`shmem_team_t* zaxis_team )`

TABLE I

API Extensions in Cray SHMEM

- **Apart from *Active Set* based Collective operations, Cray SHMEM also supports the following *Team*-based collectives:**

SHMEM_BARRIER	shmemx_team_barrier (shmem_team_t team, long* pSync)
SHMEM_REDUCTIONS	shmemx_team_[TYPE]_[OPR]_to_all (shmem_team_t team, [TYPE]* dest, const [TYPE]* source, int nreduce, [TYPE]* pWrk, long* pSync)
SHMEM_ALLTOALL	shmemx_team_alltoall (void* dest, const void* source, int nelems, shmem_team_t team, long* pSync)
SHMEM_ALLTOALLV	shmemx_team_alltoallv (void* dest, size_t* dest_offsets, size_t* dest_sizes, const void* source, size_t* src_offsets, size_t* src_sizes, shmem_team_t team, long* pSync)

Key Features And Optimizations

- New Optimizations in Cray MPI to improve performance of point-to-point and collective operations for XC systems with KNL
- Performance and API Enhancements in Cray SHMEM
- **New features in Cray MPI and Cray SHMEM to improve support for MCDRAM utilization**
- **Improved support for MPI_THREAD_MULTIPLE in Cray MPT**
  - Enhanced “Thread Hot” MPI-3 RMA capabilities on XC system with KNL
  - New locking impl. to improve multi-threaded pt2pt operations
- **Application-level performance studies on KNL**
  - WOMBAT and SNAP
- **Upcoming features in Cray MPT**

KNL High Bandwidth Memory (MCDRAM)

- Several ways to allocate memory on MCDRAM for KNL
  - CCE or Intel Compiler directives
  - memkind API (hbw_malloc)
  - numactl
  - Explicit mmap/mbind OS calls (non-trivial for end users)

- But getting hugepage memory on MCDRAM is difficult
  - Using hugepages is recommended to achieve good performance on XC
  - memkind does NOT pay attention to the craype-hugepages modules
    - even if craype-hugepage module is loaded, memkind uses 4KB pages!
  - memkind API has some hugepage options
    - Only 2M and 1GB page sizes are supported in the API
    - ..but 1GB pages are not supported on CLE
  - CCE/Intel compiler directives can't request MCDRAM hugepages currently

- Cray MPI and SHMEM implementations offer new solutions to allow hugepage memory on MCDRAM.

Cray MPI support for MCDRAM on KNL

- **Cray MPI offers hugepage support for MCDRAM on KNL**
  - Exposed to the user via existing MPI library calls: MPI_Alloc_mem() or MPI_Win_Allocate()
  - Dependencies: memkind and NUMA libraries
  - **SYS_DEFAULT**: Memory affinity settings are determined by the default system settings (numactl options, for example) for a given job
  - Memory returned by MPI_Alloc_mem() can be used for allocating performance critical application data buffers.

- **This feature is exposed via env variables**
  - Users select: Affinity, Policy and PageSize
  - MPICH_ALLOC_MEM_AFFINITY = DDR or MCDRAM
    - DDR = allocate memory on DDR
    - MCDRAM = allocate memory on MDCRAM
    - Default behavior: **SYS_DEFAULT**
  - MPICH_ALLOC_MEM_POLICY = M/ P/ I
    - M = Mandatory: fatal error if allocation fails
    - P = Preferred: fall back to using DDR memory (default)
    - I = Interleaved: Set memory affinity to interleave across MCDRAM or DDR NUMA domains
  - MPICH_ALLOC_MEM_PG_SZ
    - 4K, 2M, 4M, 8M, 16M, 32M, 64M, 128M, 256M, 512M (default 4K)
  - MPICH_INTERNAL_MEM_AFFINITY
    - Controls the memory affinity of internal memory regions allocated by the Cray MPI implementation

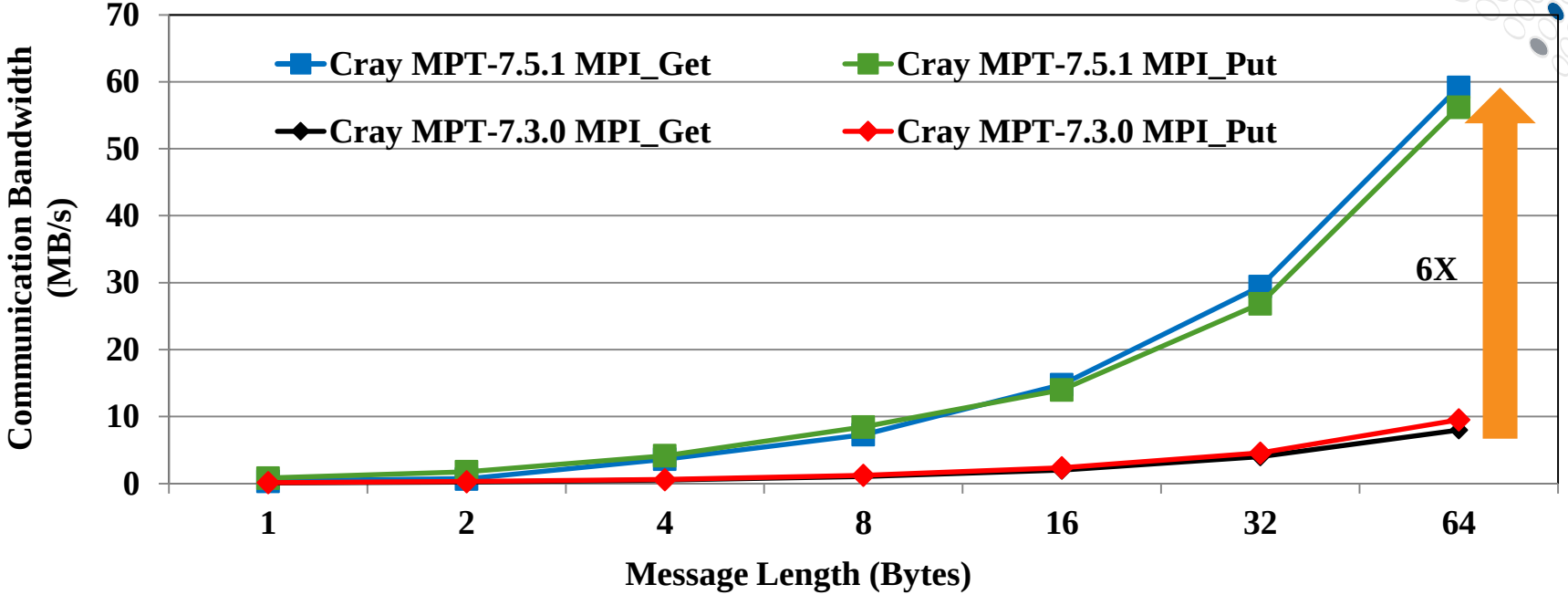
Cray SHMEM support for MCDRAM on KNL

- **SHMEM support for MCDRAM on KNL**
  - Joint efforts of Cray and Intel to define a common API for SHMEM to support different memory kinds
  - Dependency: libnuma library
  - Control memory placement of symmetric heaps (*Memory Partitions*) via env variables
  - New env variable: SMA_SYMMETRIC_PARTITION<ID>
  - User specifies: Size, Kind, Policy and PgSize
    - size=<any valid size based on available memory>
    - kind=D|Default|F|Fastmem (D=DDR, F=MCDRAM)
    - policy=M|Mandatory|P|Preferred|I|Interleaved
    - pgsize=<Supported pagesizes>
  - Can set up multiple partitions with different characteristics
  - Original shmalloc calls use memory from Partition1
  - Two new SHMEM API calls to create symmetric data objects specific to a memory partition.
    - **`void *shmemx_kind_malloc(size, partition_id)`**
    - **`void *shmemx_kind_align(alignment, size, partition_id)`**

Key Features And Optimizations

- New Optimizations in Cray MPI to improve performance of point-to-point and collective operations for XC systems with KNL
- Performance and API Enhancements in Cray SHMEM
- New features in Cray MPI and Cray SHMEM to improve support for MCDRAM utilization
- **Improved support for MPI_THREAD_MULTIPLE in Cray MPT**
  - Enhanced “Thread Hot” MPI-3 RMA capabilities on XC system with KNL
  - New locking impl. to improve multi-threaded pt2pt operations
- **Application-level performance studies on KNL**
  - WOMBAT and SNAP
- **Upcoming features in Cray MPT**

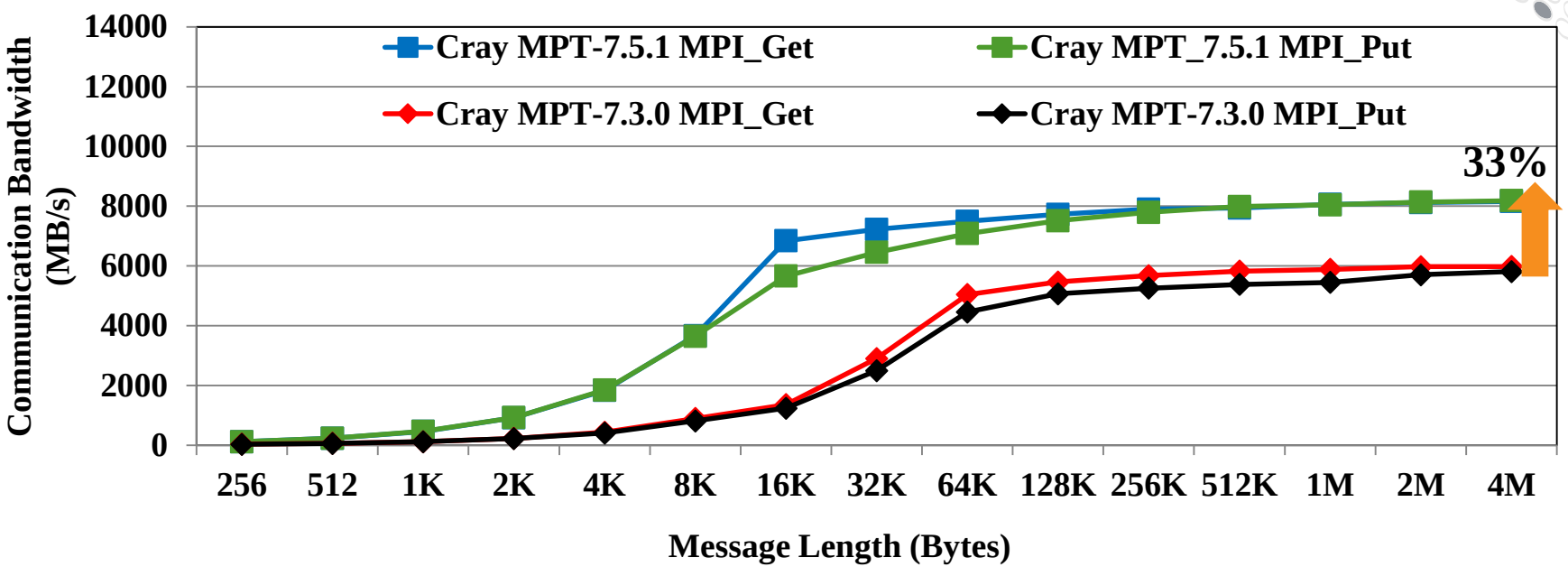
MPI_THREAD_MULTIPLE MPI-3 RMA on XC (KNL)



2 KNL Nodes, 1 MPI process per node, 64 communicating threads (Modified OSU 1-sided MicroBenchmarks)

Cray MPT-7.5.1 offers “Thread Hot” MPI-3 RMA communication tuned for XC (KNL) systems (Link against DMAPP and set MPICH_RMA_OVER_DMAPP=1)

MPI_THREAD_MULTIPLE MPI-3 RMA on XC (KNL)



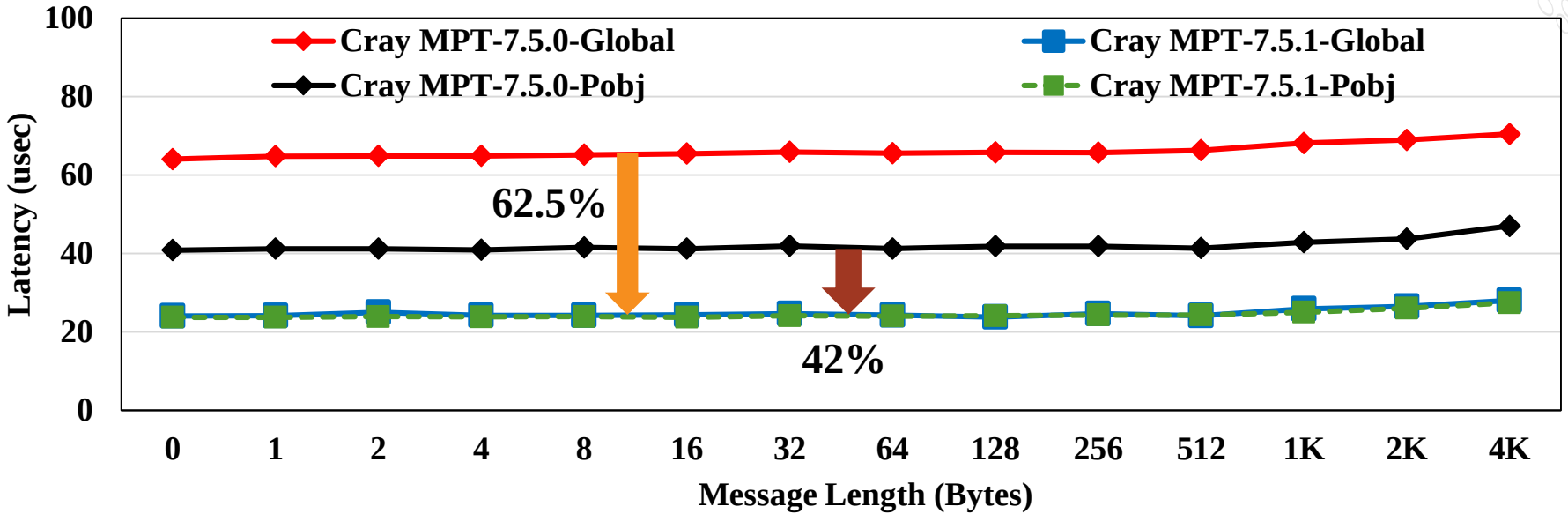
2 KNL Nodes, 1 MPI process per node, 64 communicating threads (Modified OSU 1-sided MicroBenchmarks)

Cray MPT-7.5.1 offers "Thread Hot" MPI-3 RMA communication tuned for XC (KNL) systems (Link against DMAPP and set MPICH_RMA_OVER_DMAPP=1)

Key Features And Optimizations

- New Optimizations in Cray MPI to improve performance of point-to-point and collective operations for XC systems with KNL
- Performance and API Enhancements in Cray SHMEM
- New features in Cray MPI and Cray SHMEM to improve support for MCDRAM utilization
- **Improved support for MPI_THREAD_MULTIPLE in Cray MPT**
  - Enhanced “Thread Hot” MPI-3 RMA capabilities on XC system with KNL
  - New locking impl. to improve multi-threaded pt2pt operations
- **Application-level performance studies on KNL**
  - WOMBAT and SNAP
- **Upcoming features in Cray MPT**

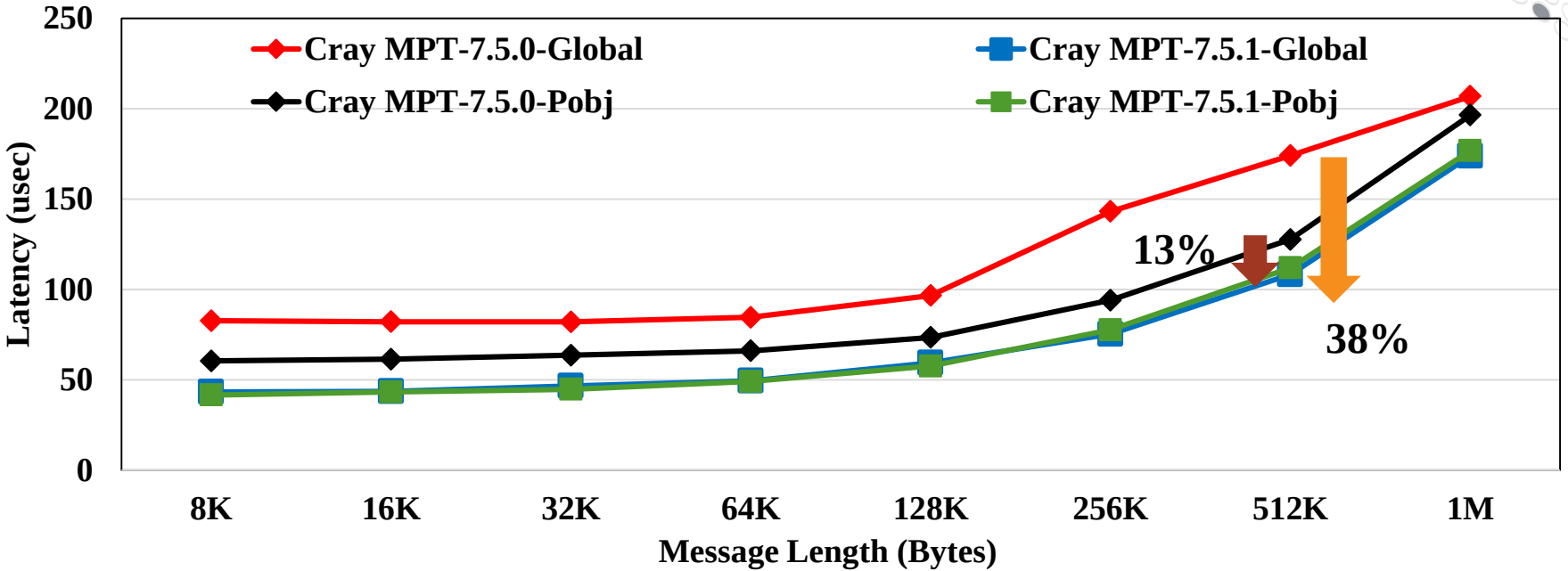
MPI_THREAD_MULTIPLE Pt-2-Pt on XC (KNL)



2 KNL Nodes, 1 MPI process per node, 16 communicating threads (osu_latency_mt.c Benchmark)

Cray MPT-7.5.1 offers a new locking implementation for MPI_THREAD_MULTIPLE support.
Global and Pobj versions in Cray MPT-7.5.1 are similar (osu_latency_mt.c)

MPI_THREAD_MULTIPLE Pt-2-Pt on XC (KNL)



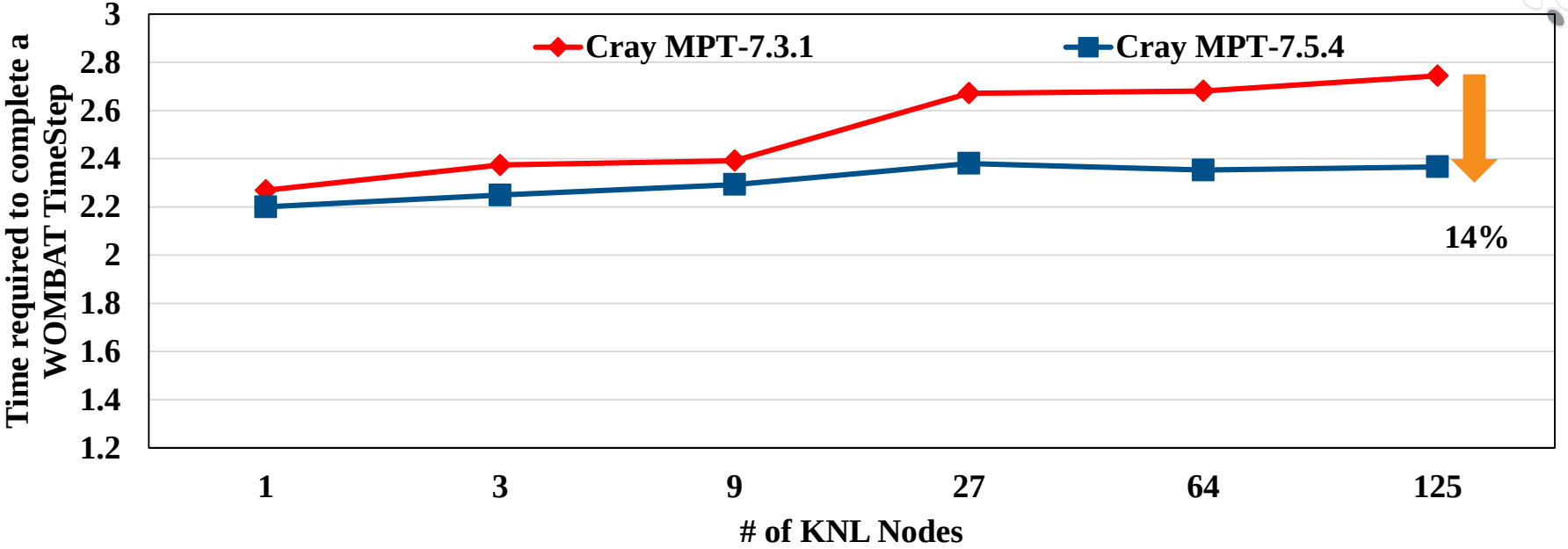
2 KNL Nodes, 1 MPI process per node, 16 communicating threads (osu_latency_mt.c Benchmark)

Cray MPT-7.5.1 offers a new locking implementation for MPI_THREAD_MULTIPLE support.
Global and Pobj versions in Cray MPT-7.5.1 are similar (osu_latency_mt.c)

Key Features And Optimizations

- New Optimizations in Cray MPI to improve performance of point-to-point and collective operations for XC systems with KNL
- Performance and API Enhancements in Cray SHMEM
- New features in Cray MPI and Cray SHMEM to improve support for MCDRAM utilization
- Improved support for MPI_THREAD_MULTIPLE in Cray MPT
  - Enhanced “Thread Hot” MPI-3 RMA capabilities on XC system with KNL
  - New locking impl. to improve multi-threaded pt2pt operations
- **Application-level performance studies on KNL**
  - WOMBAT and SNAP
- **Upcoming features in Cray MPT**

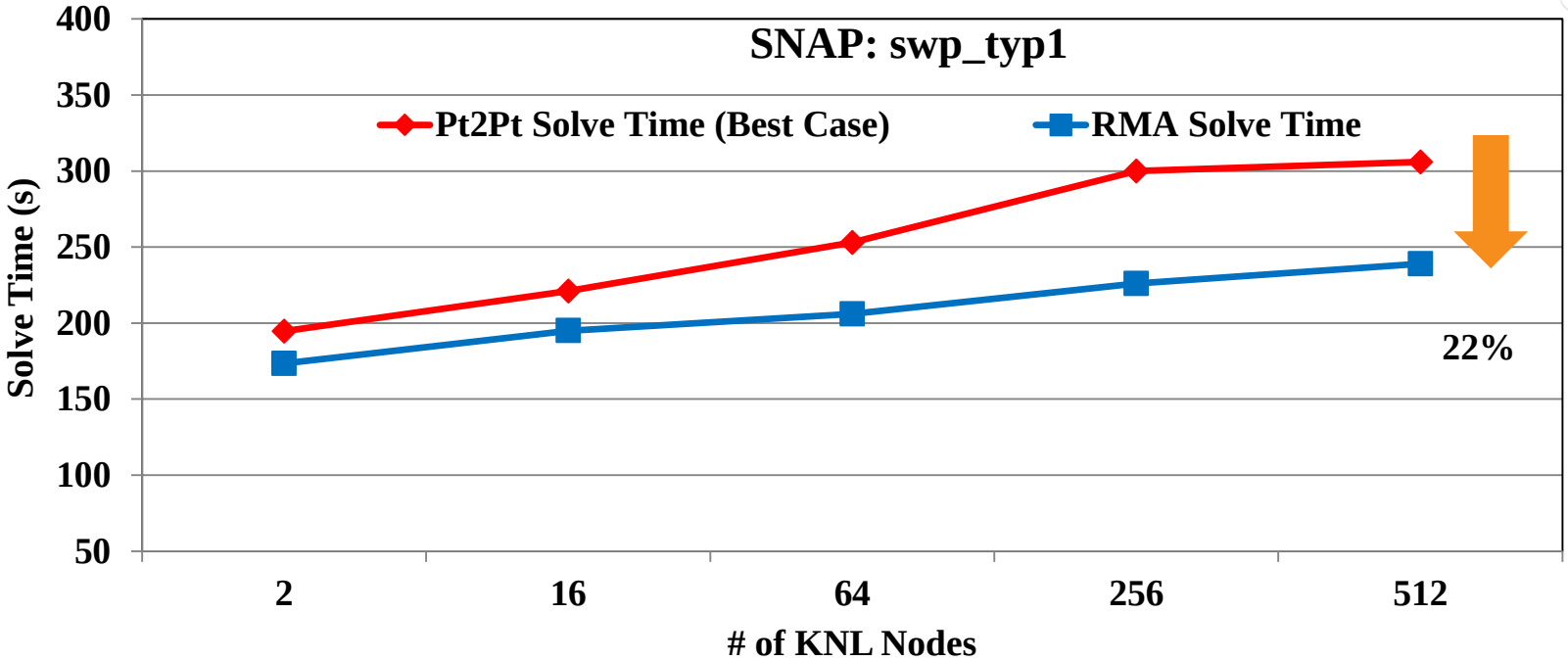
WOMBAT with Thread Hot MPI-3 RMA on Cray XC (KNL)



4 MPI processes per node, 16 communicating threads per rank

Thread Hot MPI-3 RMA in Cray MPT-7.5.1 improves the time required to perform a TimeStep in WOMBAT on Cray XC (KNL) systems by up to **14%**

SNAP with Thread Hot MPI-3 RMA on Cray XC (KNL)



RMA: 16 MPI processes per node, 8 OMP threads per rank, 2 HyperThreads per core

Thread Hot MPI-3 RMA Cray MPT-7.5.1 improves the Solve Time in SNAP by up to **22%**
Pt2pt Solve Time corresponds to Best Case runs with exhaustive tuning (#threads per rank; # ranks per node)

Agenda

- Introduction & Motivation
- Key Features And Optimizations
- **New and Upcoming Features in Cray MPT**
- Q&A

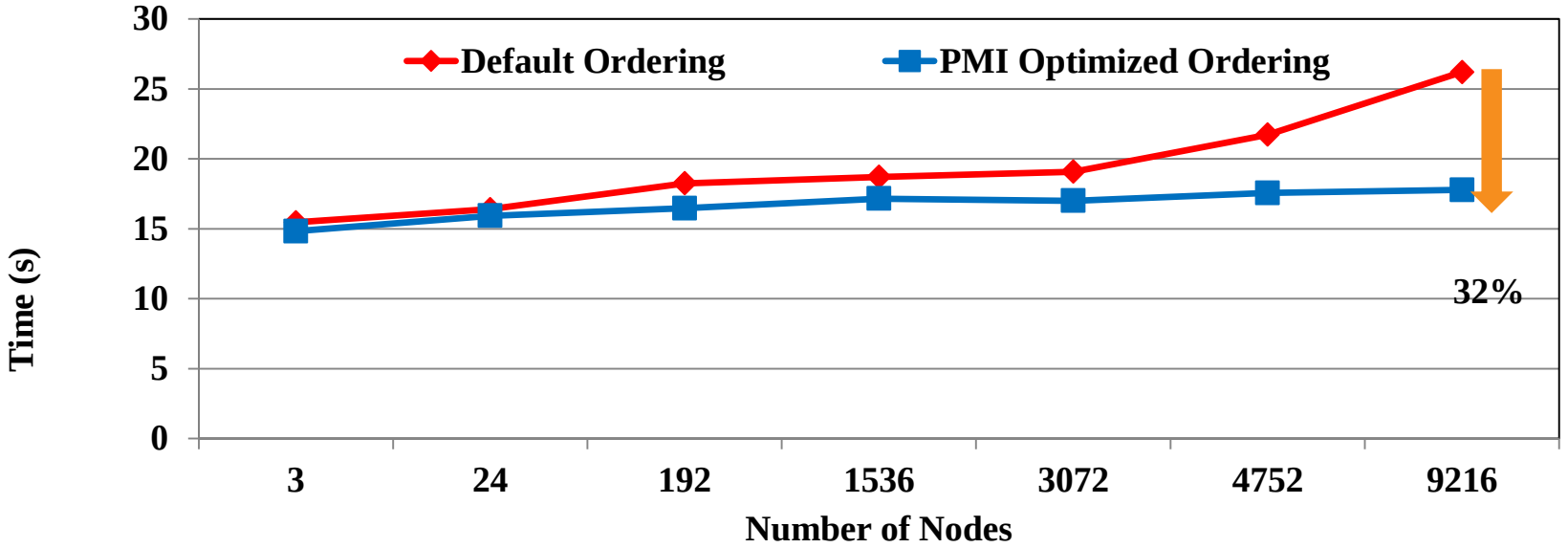
MPI-3 Dynamic Process Management (DPM) Support

- DPM support in Cray MPI will be rolled out incrementally during 2017 and early 2018
- Q2 2017 release will support MPI_Comm_connect/accept().
- New Env. Variable in Cray MPT-7.5.5:

  MPICH_DPM_SERVER=1 (for the server process)

  MPICH_DPM_CLEINT="<file path>" (for the client)
- Future (Q4 2017 and early 2018) Cray MPI releases will support MPI_Comm_spawn()

Improved Support for Network Topology-Awareness



MPICH_RANK_REORDER_METHOD=4 (to override the default process placement settings)
For a 3-D Cubic Grid, near-neighbor comm. pattern with 4,096 MPI processes:
MPICH_RANK_REORDER_OPTS="—ndims=3 –dims=16,16,16 --nearest"

Cray PMI offers topology-aware rank-reordering to improve communication performance on XC systems. Improves MiniGhost Execution time on Trinity (XC 40) by up to **32%**

New MPI I/O Optimizations in Cray MPT

- Lustre Lockahead is a new Cray enhancement in Lustre
- Significantly improves write performance for collective and shared-file I/O workloads
- Studies have demonstrated about **200%** improvements for small transfers and over **100%** benefits for larger transfers when compared to traditional Lustre
- Cray MPI leverages this feature to improve shared-file collective I/O, achieving more than **80%** of file per process performance.

  (M. Moore, P. Farrell and B. Cernohous, "Lustre Lockahead: Early Experience and Performance using Optimized Locking," Cray User Group (CUG) 2017)
- Cray MPT-7.5.3 offers improved timing statistics for different I/O phases (MPICH_MPIIO_TIMERS=1)

Summary & Conclusion

- New features and optimizations in Cray MPI and Cray SHMEM Software stacks for Cray XC systems
- Improved support for MPI_THREAD_MULTIPLE for MPI pt2pt and RMA operations on KNL
- WOMBAT and SNAP scaling studies on XC (KNL) systems with Thread Hot MPI-3 RMA optimizations in Cray MPI
- Summary of upcoming features in Cray MPT

Legal Disclaimer

Information in this document is provided in connection with Cray Inc. products. No license, express or implied, to any intellectual property rights is granted by this document.

Cray Inc. may make changes to specifications and product descriptions at any time, without notice.

All products, dates and figures specified are preliminary based on current expectations, and are subject to change without notice.

Cray hardware and software products 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.

Cray uses codenames internally to identify products that are in development and not yet publically announced for release. Customers and other third parties are not authorized by Cray Inc. to use codenames in advertising, promotion or marketing and any use of Cray Inc. internal codenames is at the sole risk of the user.

Performance tests and ratings are measured using specific systems and/or components and reflect the approximate performance of Cray Inc. products as measured by those tests. Any difference in system hardware or software design or configuration may affect actual performance.

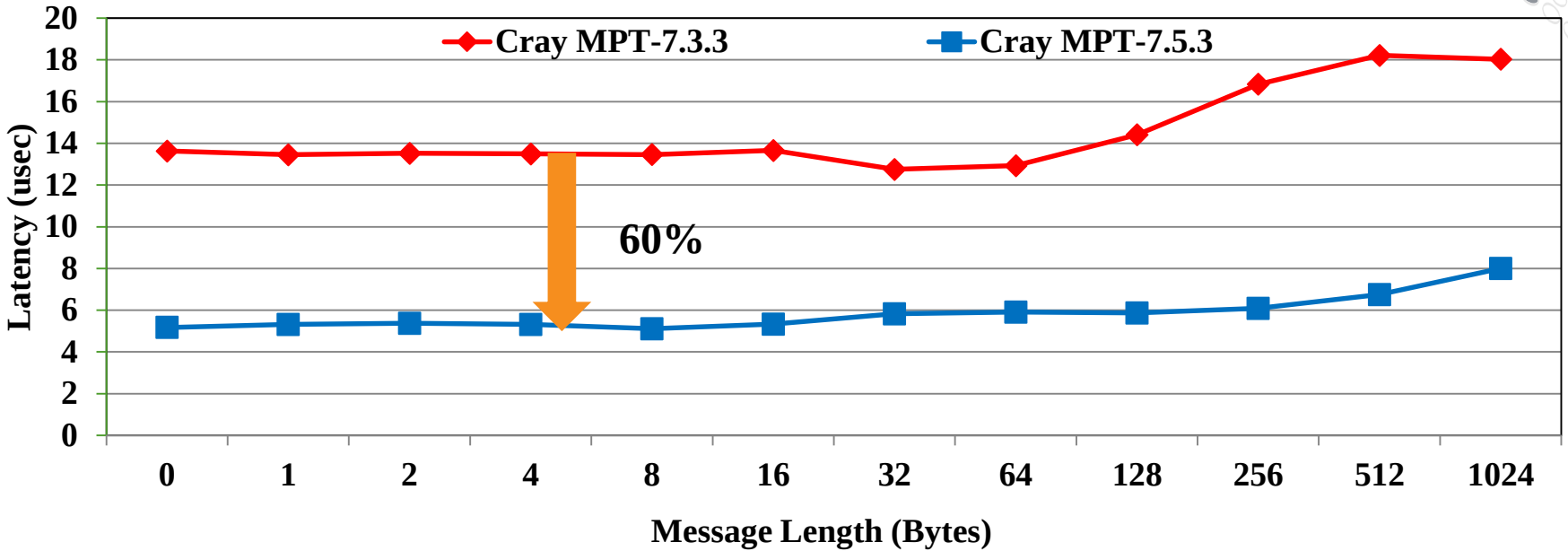
The following are trademarks of Cray Inc. and are registered in the United States and other countries: CRAY and design, SONEXION, and URIKA. The following are trademarks of Cray Inc.: APPRENTICE2, CHAPEL, CLUSTER CONNECT, CRAYPAT, CRAYPORT, ECOPHLEX, LIBSCI, NODEKARE, THREADSTORM, REVEAL. The following system family marks, and associated model number marks, are trademarks of Cray Inc.: CS, CX, XC, XE, XK, XMT, and XT. The registered trademark LINUX is used pursuant to a sublicense from LMI, the exclusive licensee of Linus Torvalds, owner of the mark on a worldwide basis. Other trademarks used in this document are the property of their respective owners.

Q&A

Krishna Kandalla, Ph.D.
kkandalla@cray.com

Backup Slides

MPI Multi-Pair Pt-2-Pt latency on XC (KNL)



2 KNL Nodes, 32 MPI processes per node (64 processes, 32 communicating pairs)

New enhancements in Cray MPT improve the performance of multi-pair communication patterns by up to **60%**