System Software Laboratory

KU KONKUK UNIVERSITY

Kernel-Level Support for Scalable Intra-Node Collective Communications

Hyun-Wook Jin and **Joong-Yeon Cho**
System Software Laboratory
Dept. of Computer Science and Engineering
Konkuk University
jinh@konkuk.ac.kr

Contents

- MPI intra-node communication
- Intra-node collective communication
  - MPI_Bast()
  - MPI_Gather()
- Conclusions and future work

Multi/Many-Core Processors

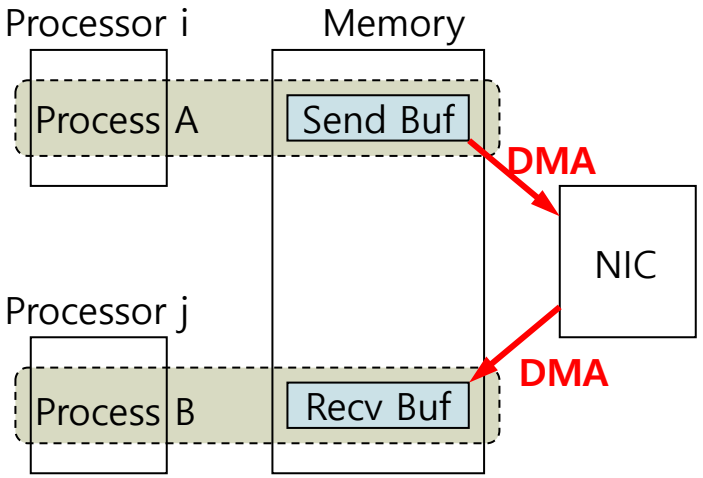
Xeon 5100 Series (Woodcrest)	Xeon 5500 Series (Gainestown)	Xeon 5600 Series (Westmere-EP)	Xeon E5-2600 Series (Sandy Bridge-EP)
2	4	6	8

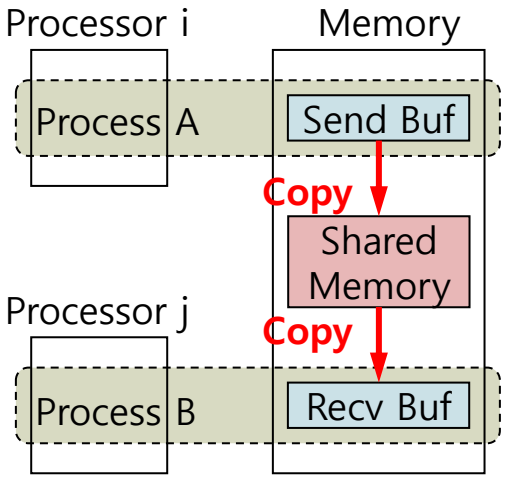
Xeon E5-2600 v2 Series (Ivy Bridge-EP)	Xeon E5-2600 v3 Series (Haswell-EP)	Xeon E7 v4 Family (Broadwell)	Xeon Platinum Series (Skylake)
12	18	24	28

Xeon Phi X100 Series (Knights Corner)	Xeon Phi 7200 Series (Knights Landing)
61	72

MPI Intra-Node Communication

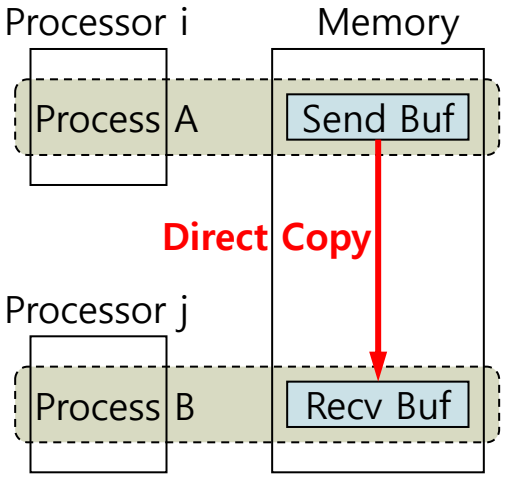
- Loopback
  - NIC provides a loopback path
  - Two DMAs
- Shared memory
  - Communicate through a memory area shared between MPI processes
  - Two data copies





MPI Intra-Node Communication

- Memory mapping
  - Directly move a message from source to destination buffer by means of kernel-level support
  - Single data copy
    - Beneficial for large messages



Kernel-Level Support for MMapping

- LiMIC2
  - Opened the era of one-copy intra-node communication
    - H-W. Jin, S. Sur, L. Chai, and D. K. Panda, "LiMIC: Support for High-Performance MPI Intra-Node Communication on Linux Cluster," In Proc. of CPP-05, Jun. 2005.
    - H.-W. Jin, S. Sur, Lei Chai, and D. K. Panda, "Lightweight Kernel-Level Primitives for High-Performance MPI Intra-Node Communication over Multi-Core Systems," In Proc. of IEEE Cluster 2007, Sep. 2007.
  - LiMIC2-0.5 was publicly released with MVAPICH2-1.4RC1 (Jun. 2009)
  - LiMIC2-0.5.6 is being released with the latest MVAPICH2
    - mvapich2-src]$ ./configure --with-limic2 [omit other configure options]
    - mvapich2-src]$ mpirun_rsh -np 4 -hostfile ~/hosts MV2_SMP_USE_LIMIC2=1 [path to application]

Kernel-Level Support for MMapping

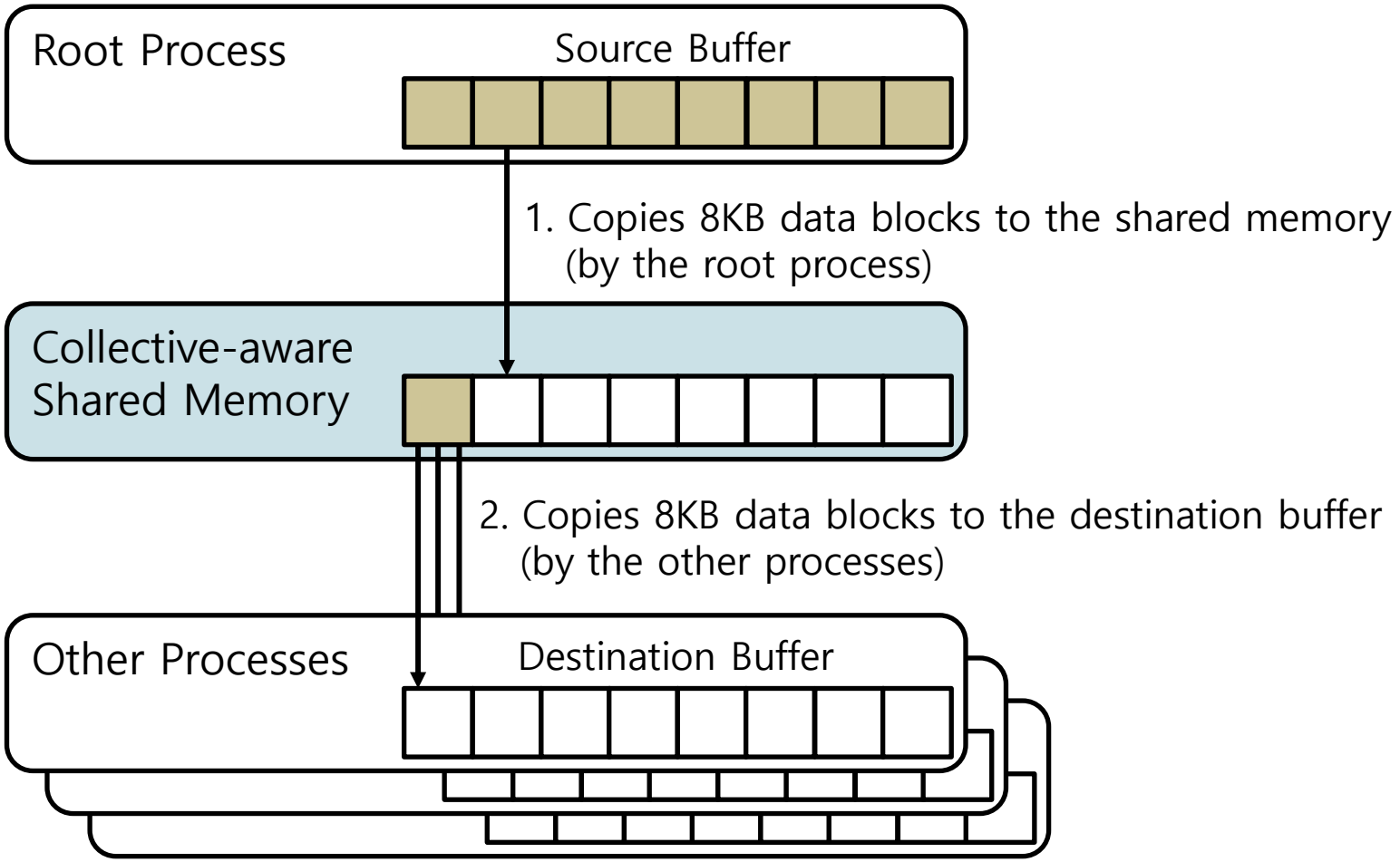
- CMA
  - In-kernel implementation + New system calls
    - J. Vienne, "Benefits of Cross Memory Attach for MPI Libraries on HPC Clusters," In Proc. of XSEDE 14, Jul. 2014.
  - Default intra-node communication channel for large messages in MVAPICH2
- XPMEM
  - Supports memory mapping to user-level address space
    - B. Kocoloski and J. Lange, "XEMEM: Efficient Shared Memory for Composed Applications on Multi-OS/R Exascale Systems," In Proc. of HPDC 2015, 2015.

Intra-Node Collective Communication

- MPI_Bcast()
  - Broadcasts a message from the root to all other processes of the communicator
    - One-to-Many: Root -> Other processes
  - MVAPICH2 (version 2.3) uses the collective-aware shared memory
- MPI_Gather()
  - Gathers together values from a group of processes
    - Many-to-One: All processes -> Root
  - MVAPICH2 (version 2.3) uses the kernel-level support (either CMA or LiMIC2) for large messages

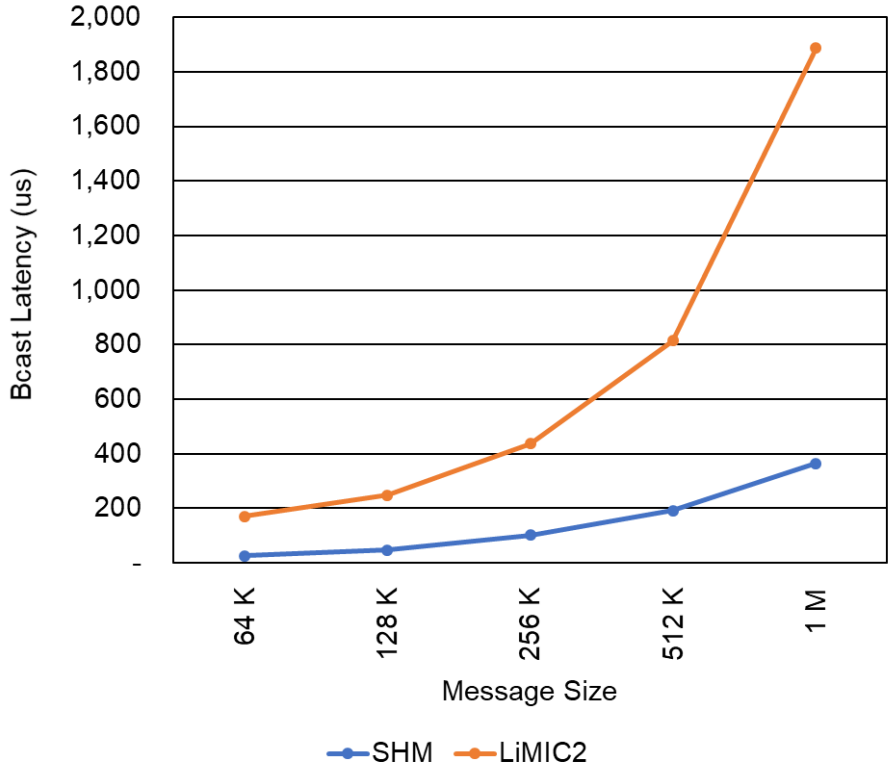
MPI_BCAST

MPI_Bcast() in MVAPICH2 (v.2.3)



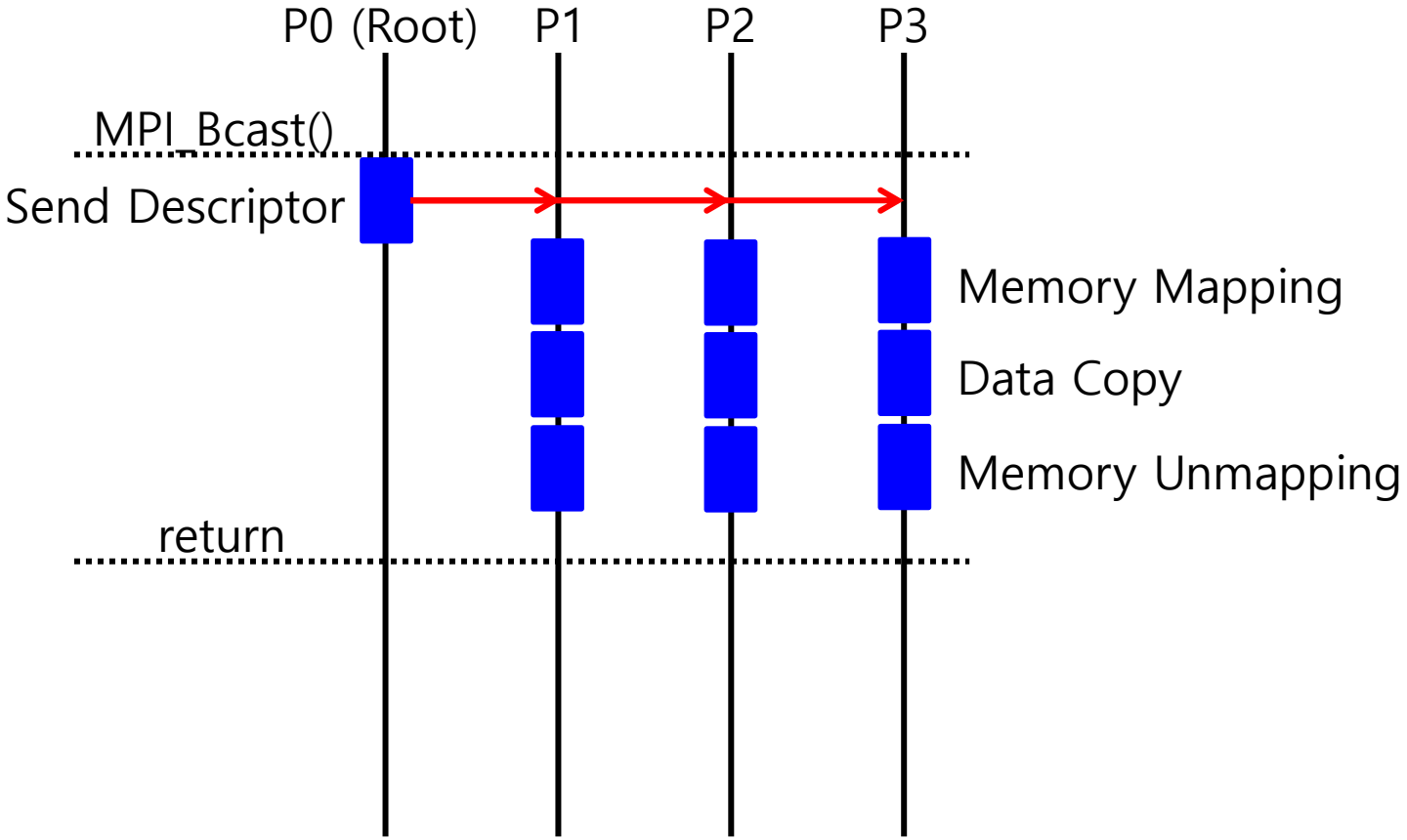
How bad is LiMIC2 for MPI_Bcast()?

- Experimentally applied LiMIC2 instead of shared memory
  - Shows higher latency up to 548%



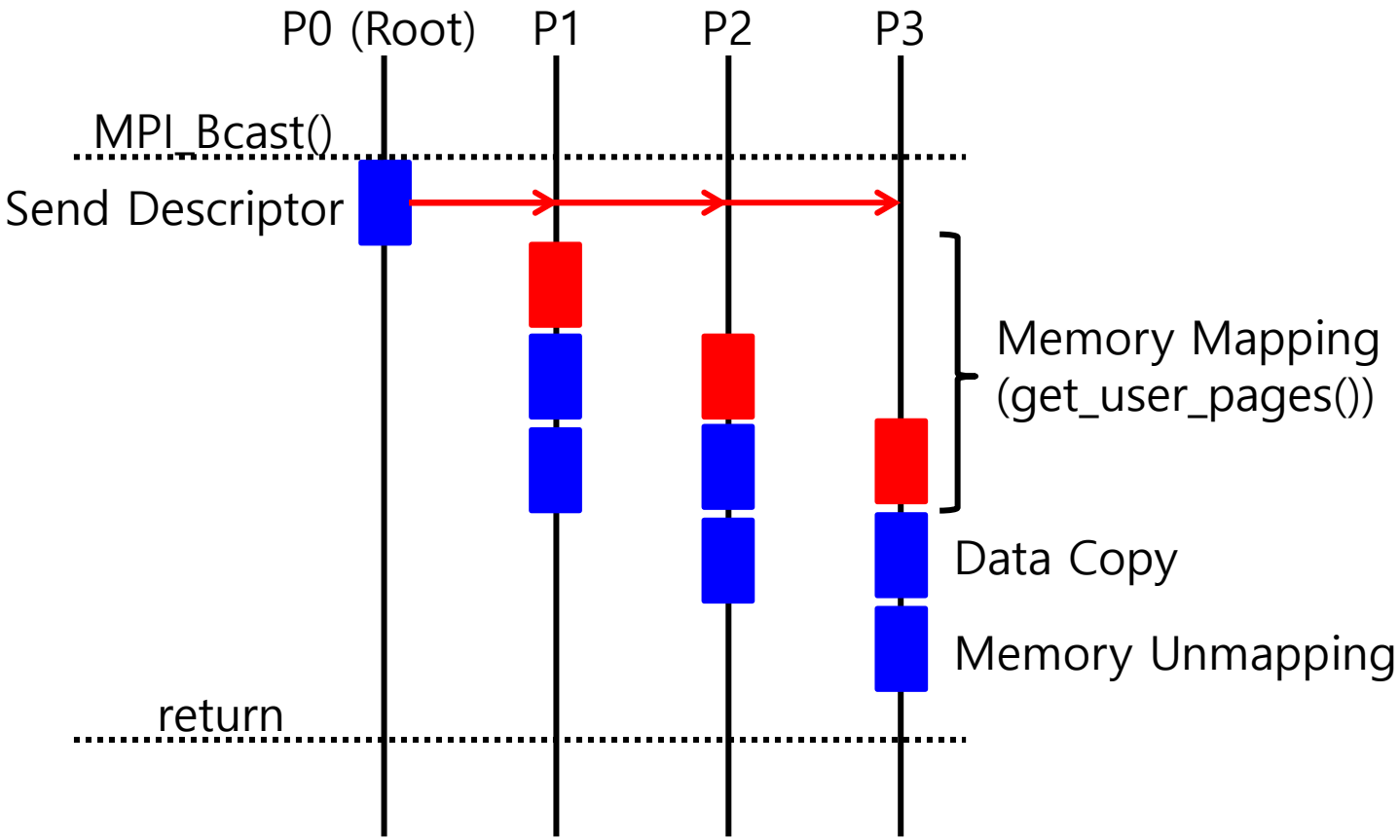
Why not to use LiMIC2 in MPI_Bcast()?

- What we expected…



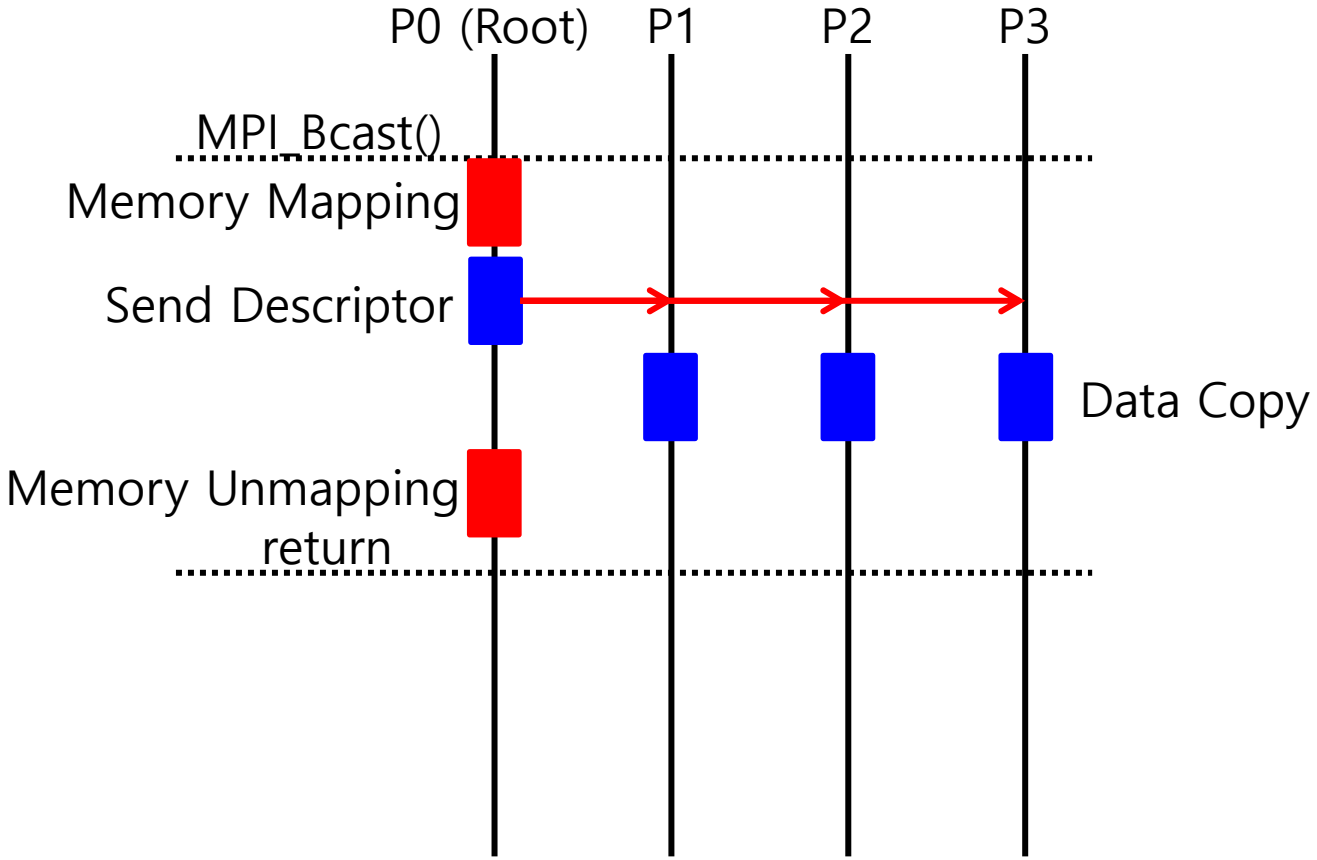
Why not to use LiMIC2 in MPI_Bcast()?

- What actually happened…



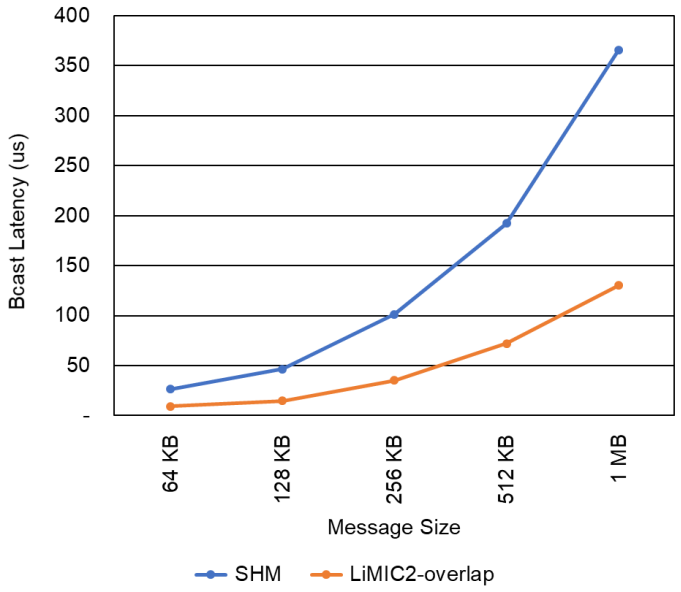
MPI_Bcast() with LiMIC2-overlap

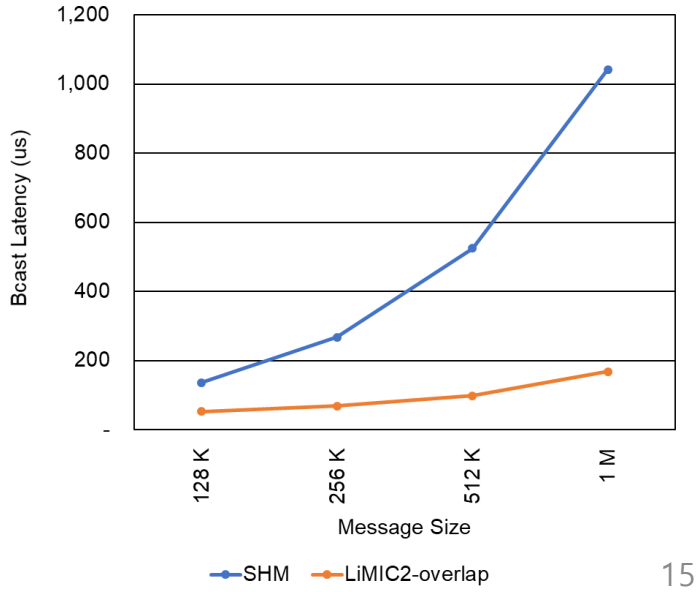
- The root performs memory mapping and the others reuse (share) the mapped area



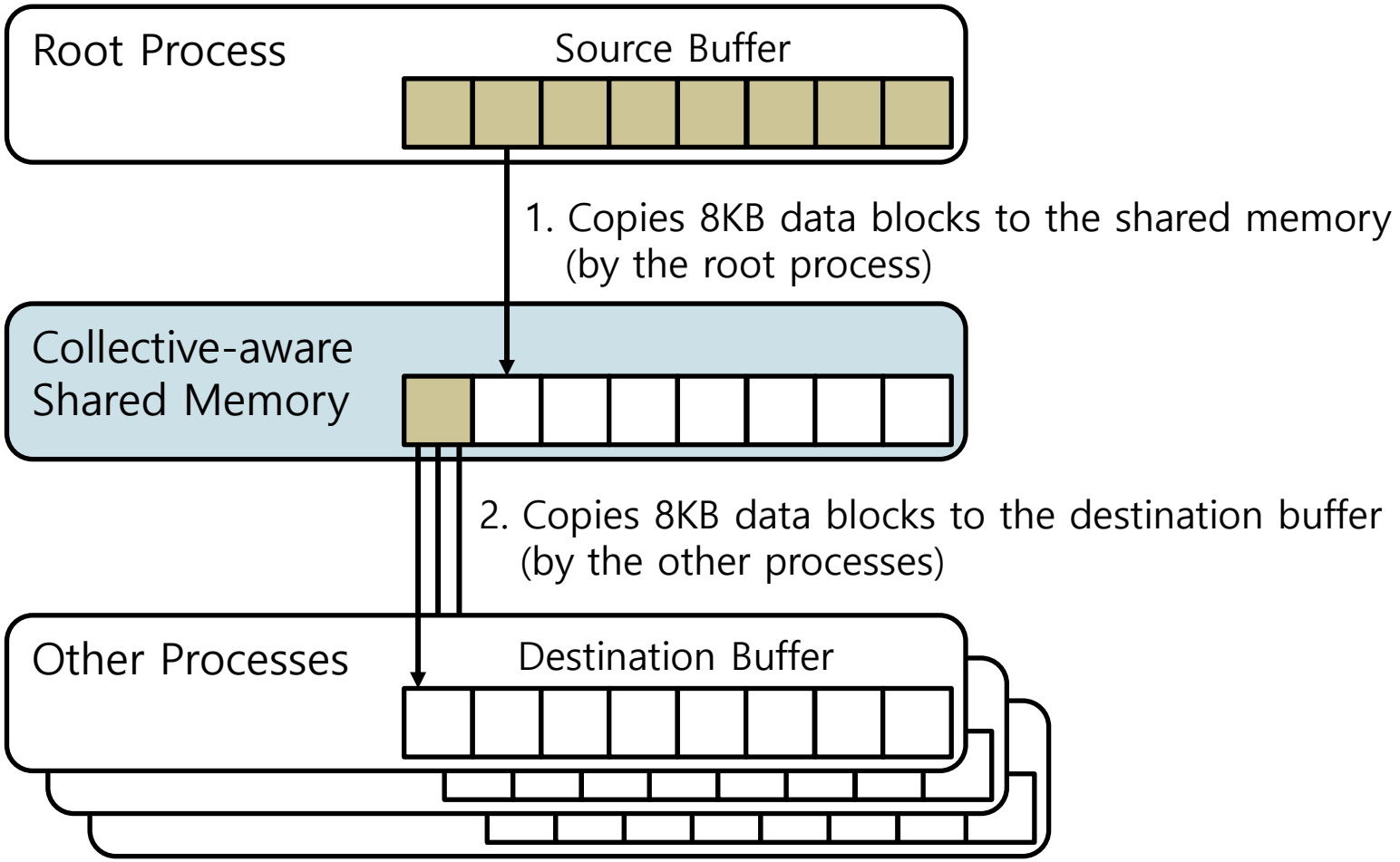
Preliminary Measurement Results

- 20-core system
  - Intel Xeon Haswell Deca-Core x 2
  - LiMIC2-overlap reduces the latency up to 68%
- 120-core system
  - Intel Xeon IvyBridge 15-Core x 8
  - LiMIC2-overlap reduces the latency up to 84%



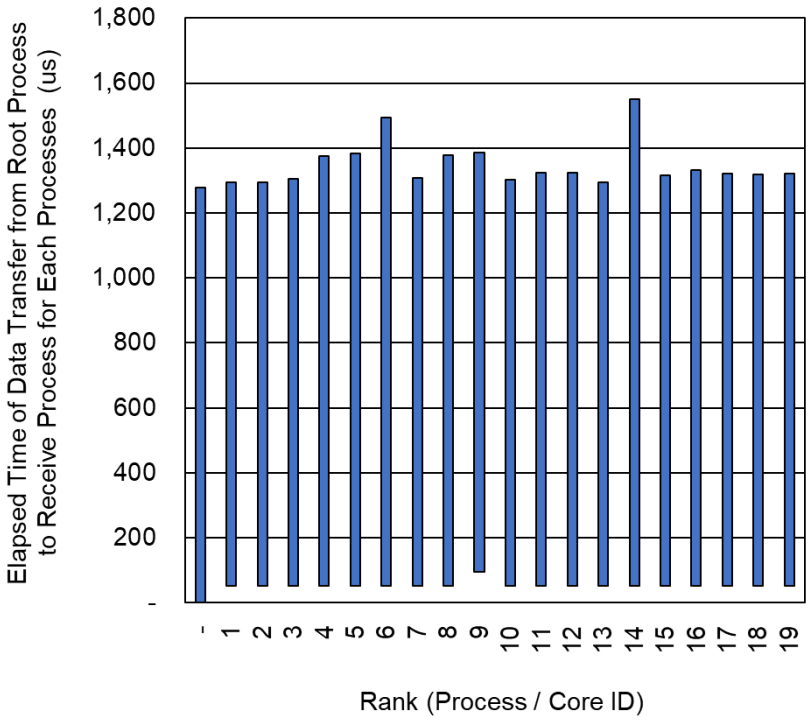


What's going on in MVAPICH2 Bcast?

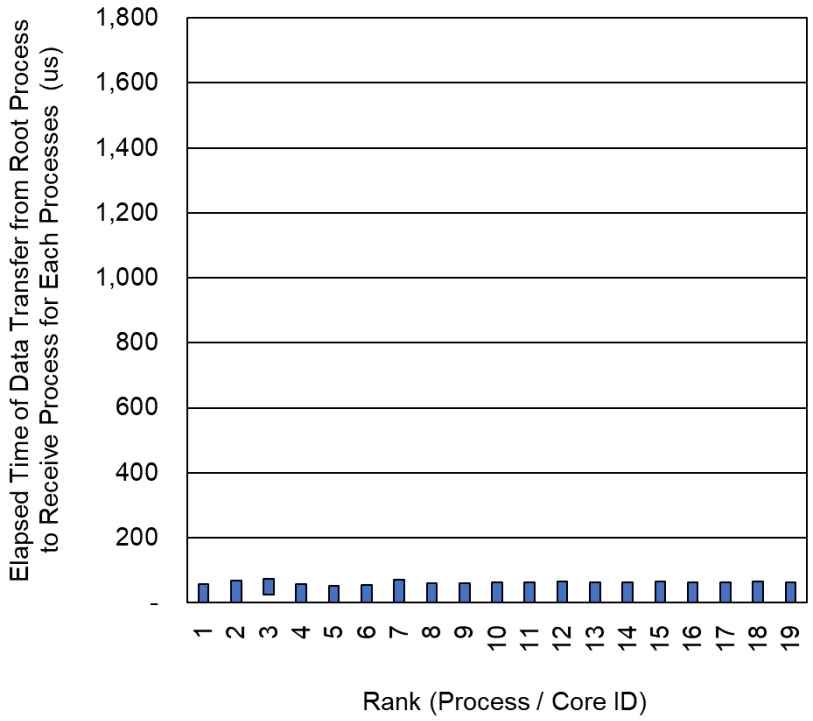


What's going on in MVAPICH2 Bcast?

- Collective-aware shared memory



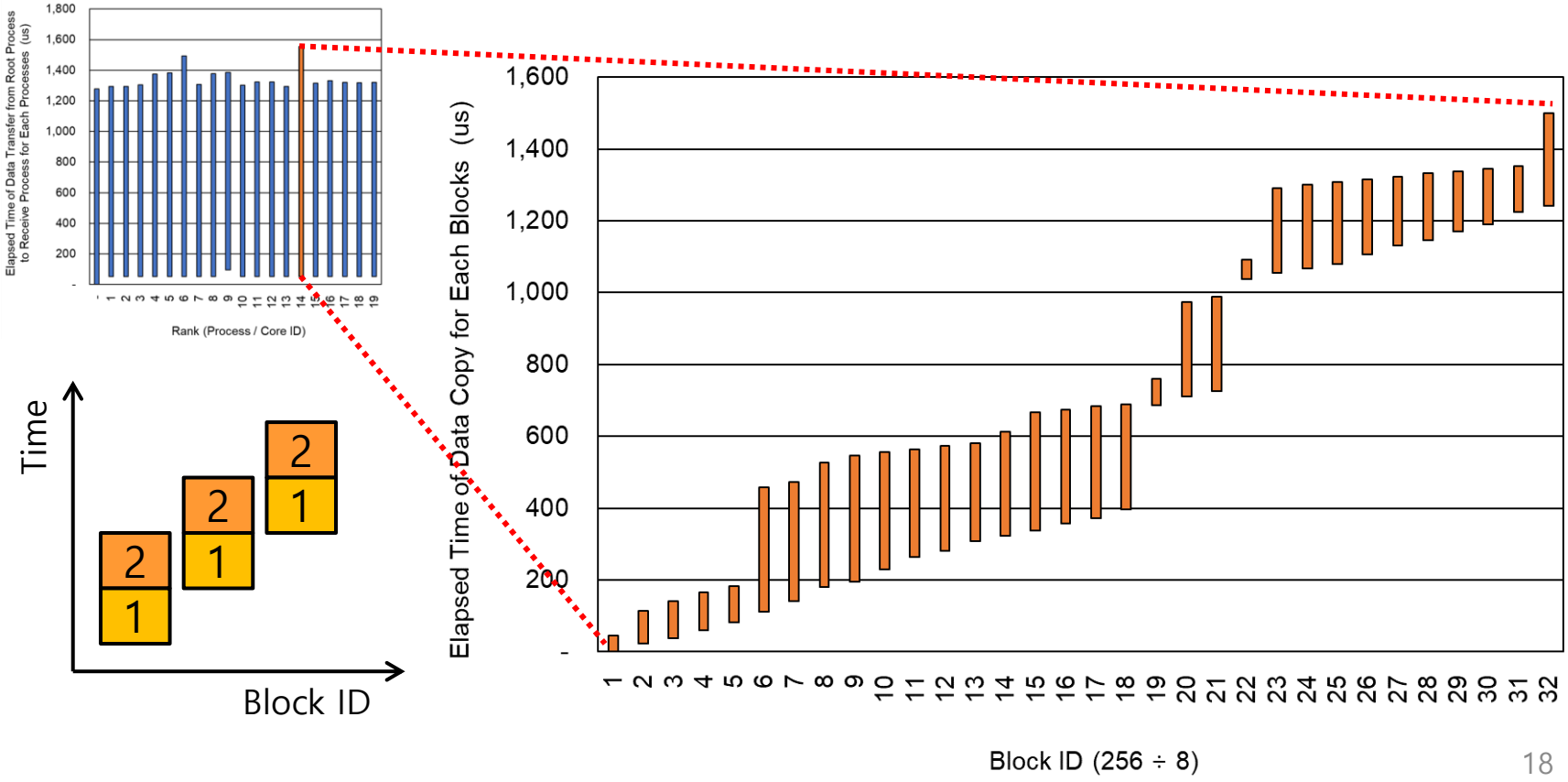
- LiMIC2-overlap



* Message size: 256KB
* Some profiling overheads are included

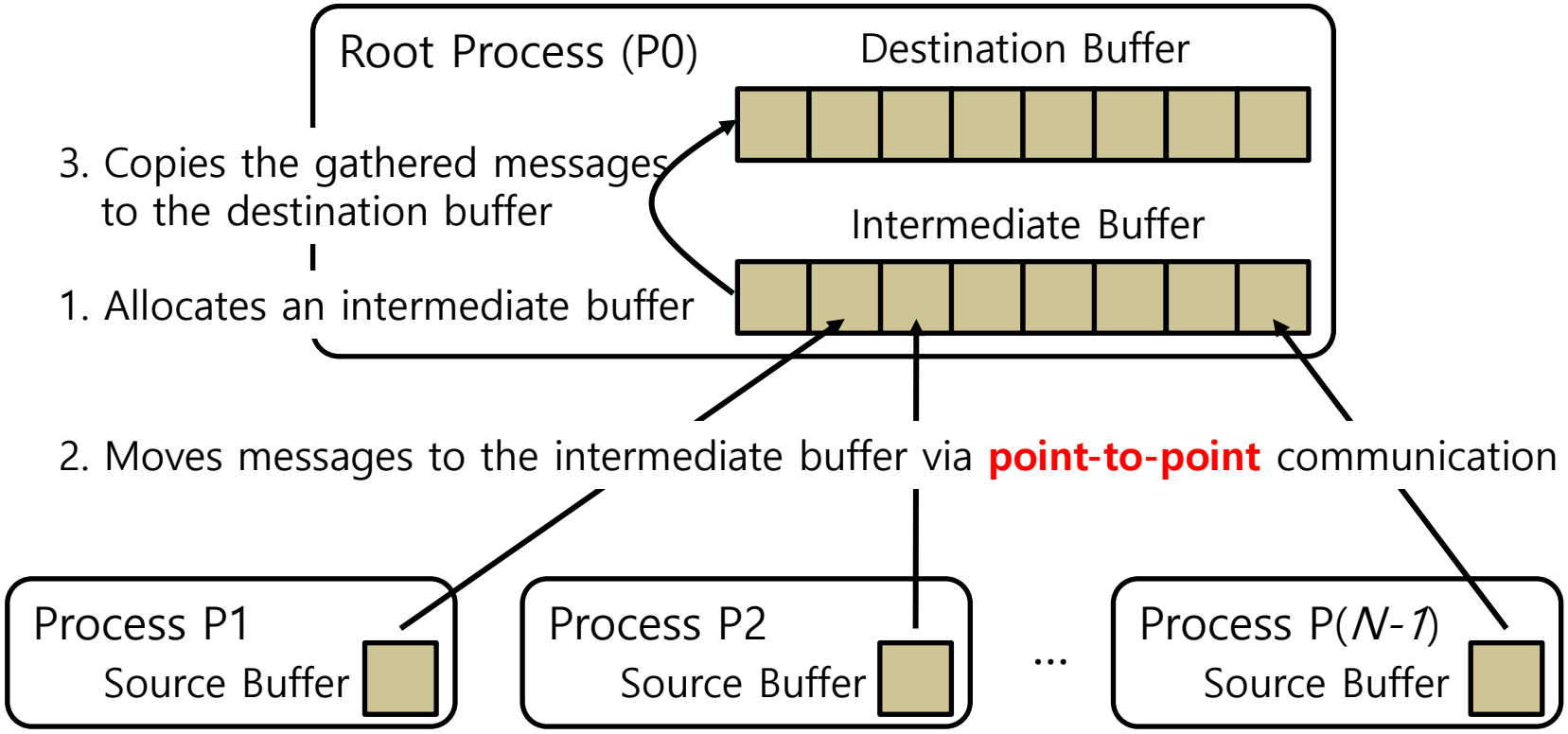
What's going on in MVAPICH2 Bcast?

- Data copy operations are not overlapped as much as expected

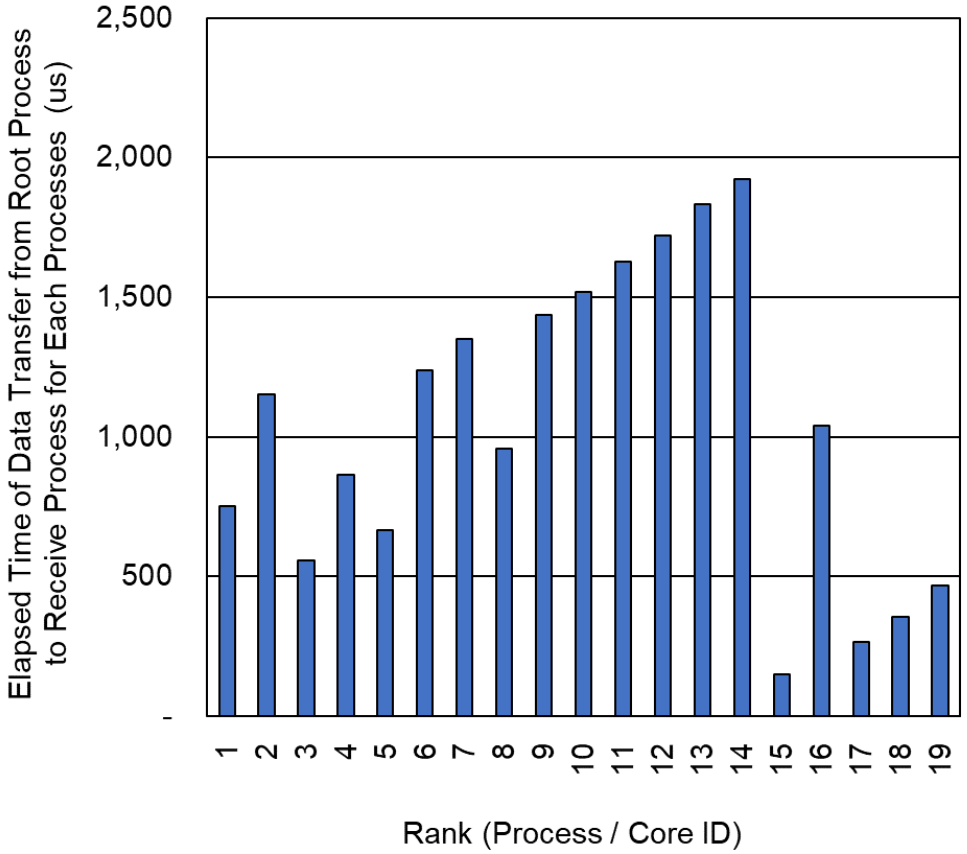


MPI_GATHER

MPI_Gather() in MVAPICH2 (v.2.3)

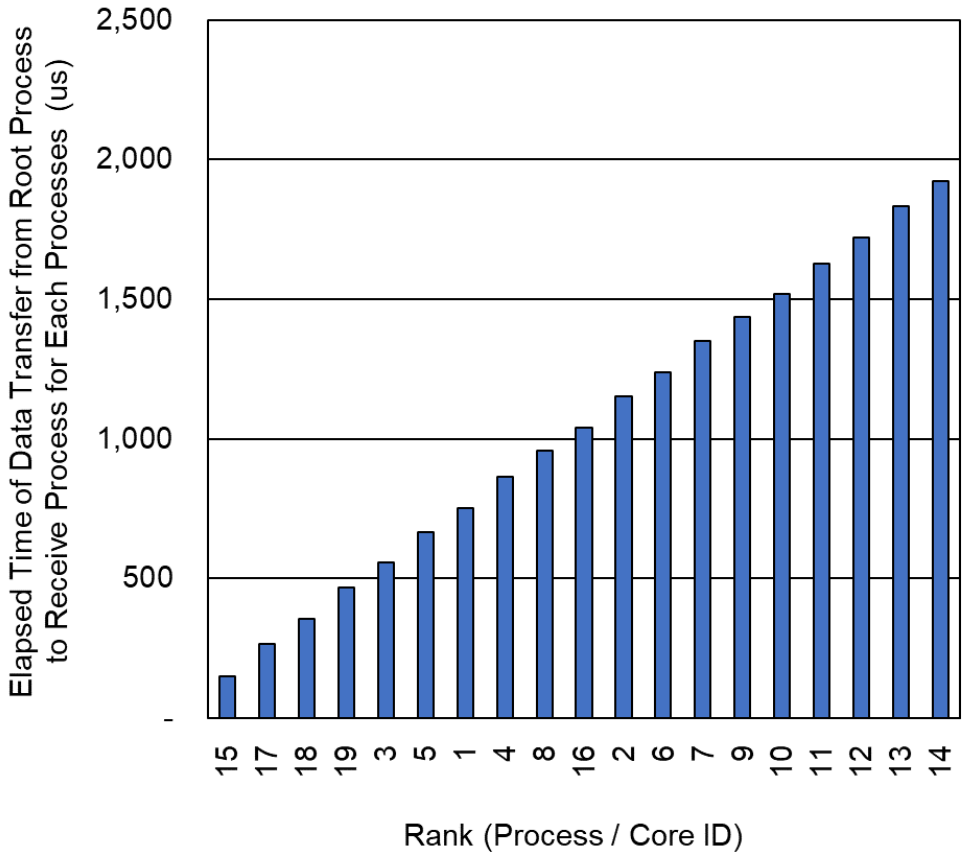


Is it OK to use LiMIC2 in MPI_Gather()?

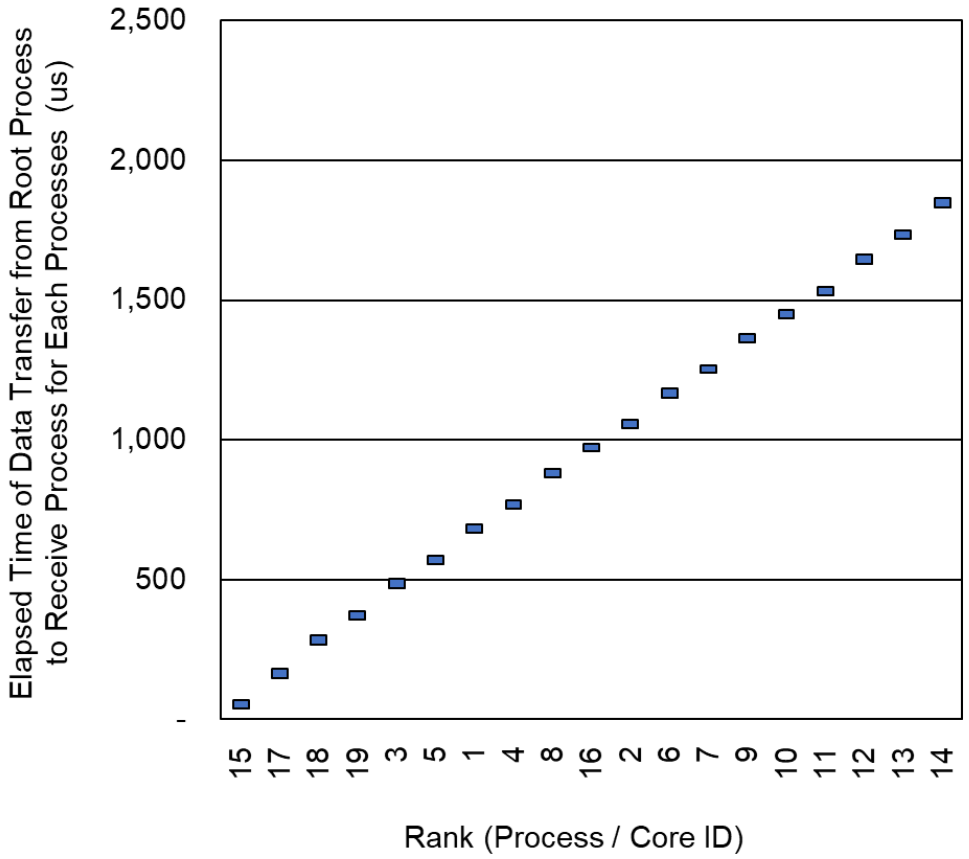


* Message size: 256KB
* Some profiling overheads are included

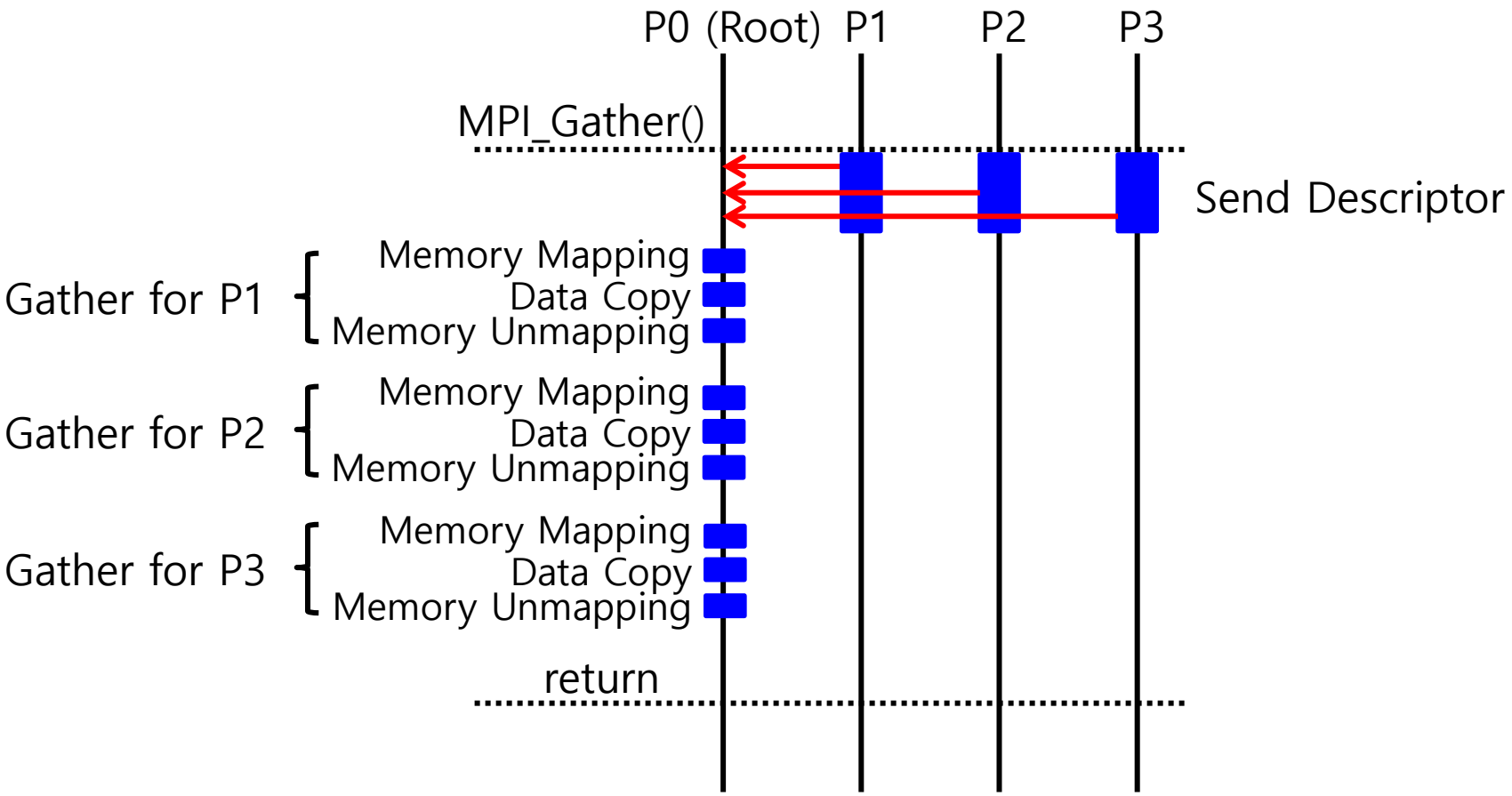
Is it OK to use LiMIC2 in MPI_Gather()?



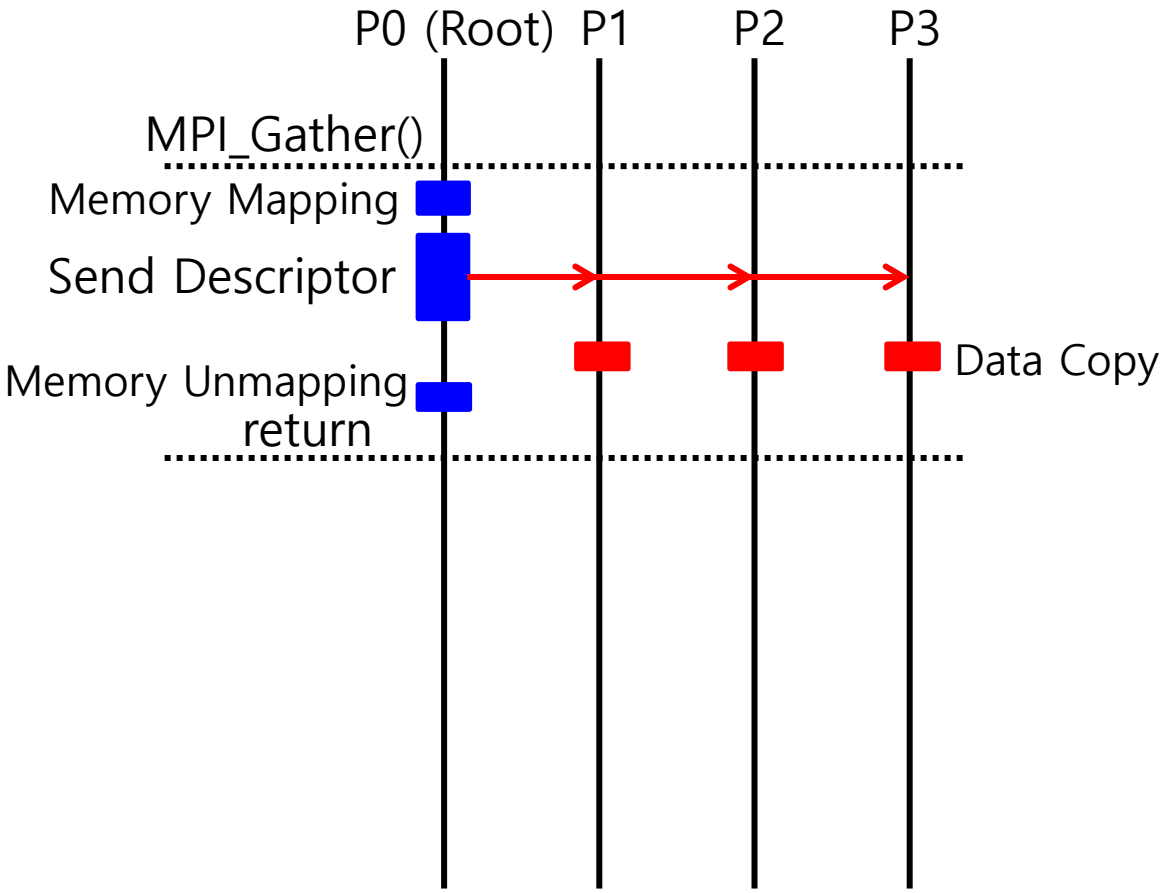
Is it OK to use LiMIC2 in MPI_Gather()?



Why not to use LiMIC2 in MPI_Gather()?

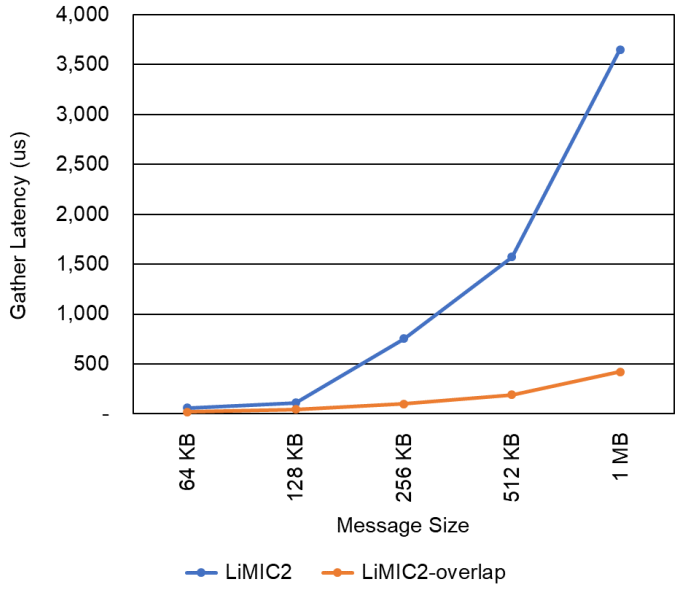


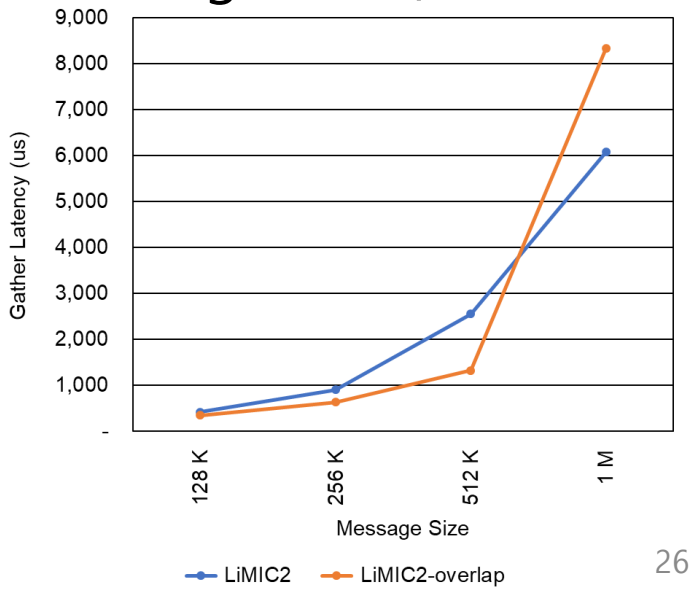
MPI_Gather() with LiMIC2-overlap



Preliminary Measurement Results

- 20-core system
  - LiMIC2-overlap reduces the latency up to 88%
- 120-core system
  - LiMIC2-overlap reduces the latency up to 50%
  - Different algorithms matter (e.g., binomial tree algorithm)





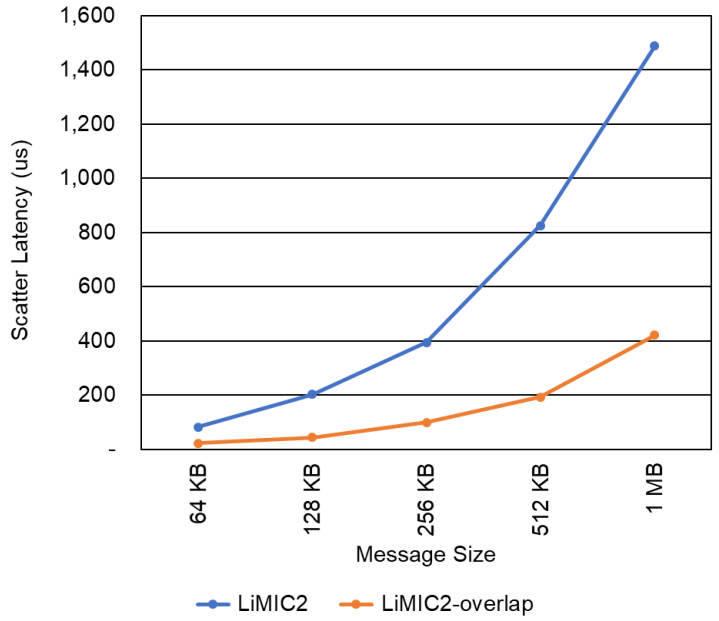
CONCLUSIONS

Concluding Remark

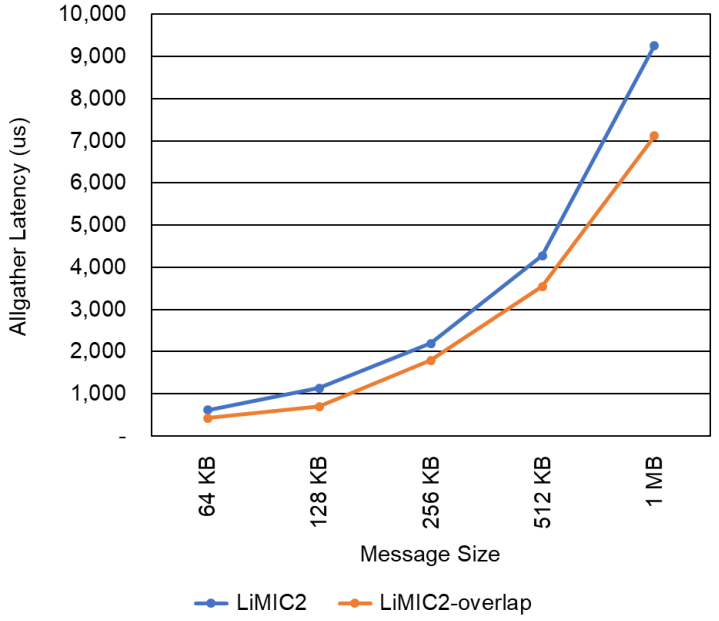
- Intra-node collective communication
  - MPI_Bcast()
    - One-to-Many communication
    - Implemented using collective-aware shared memory
  - MPI_Gather()
    - Many-to-One communication
    - Implemented using point-to-point
- LiMIC2-overlap
  - New interfaces
    - Memory mapping reuse
    - Flexibility of who can perform data copy
  - 84% improvement for MPI_Bcast()
  - 88% improvement for MPI_Gather()

Ongoing Work

- Other collectives
  - MPI_Scatter()
    - LiMIC2-overlap reduces the latency up to 78% on the 20-core system

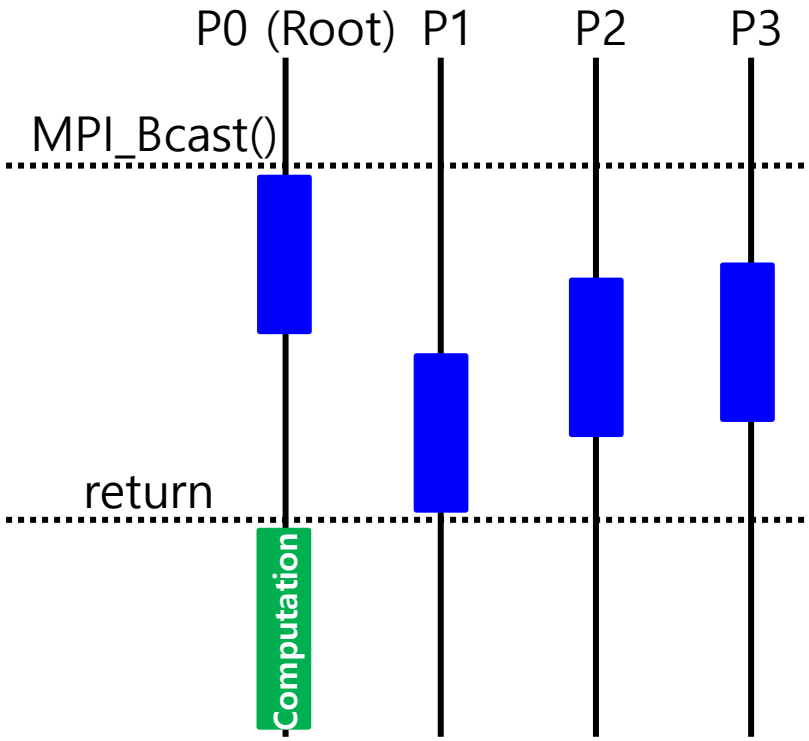


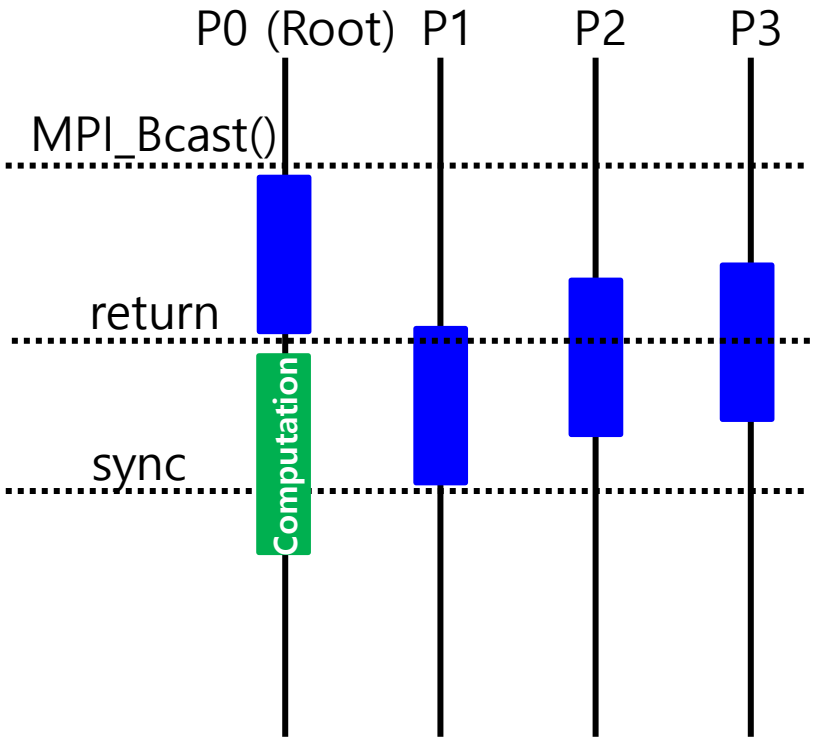
  - MPI_Allgather()
    - LiMIC2-overlap reduces the latency up to 38% on the 20-core system



Ongoing Work

- Overlapping between collective communication and computation





Future Work

LiMIC3

ParaMo 2019

- The 1st International Workshop on Parallel Programming Models in High-Performance Cloud
  - Co-located with Euro-Par 2019
  - Date: August 26, 2019
  - Venue: Göttingen, Germany

Thank You!

Ministry of Science and ICT

National Research Foundation of Korea

Institute of Information & Communications Technology Planning & Evaluation