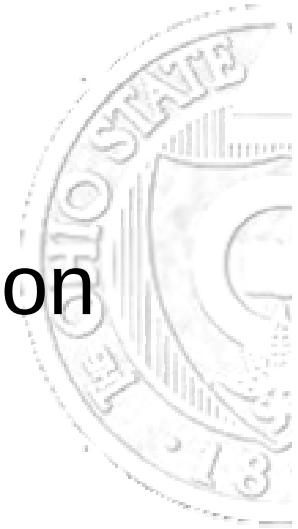
NETWORK-BASED COMPUTING LABORATORY



Designing High Performance and Scalable MPI Intra-node Communication Support for Clusters

Lei Chai Albert Hartono Dhabaleswar. K. Panda

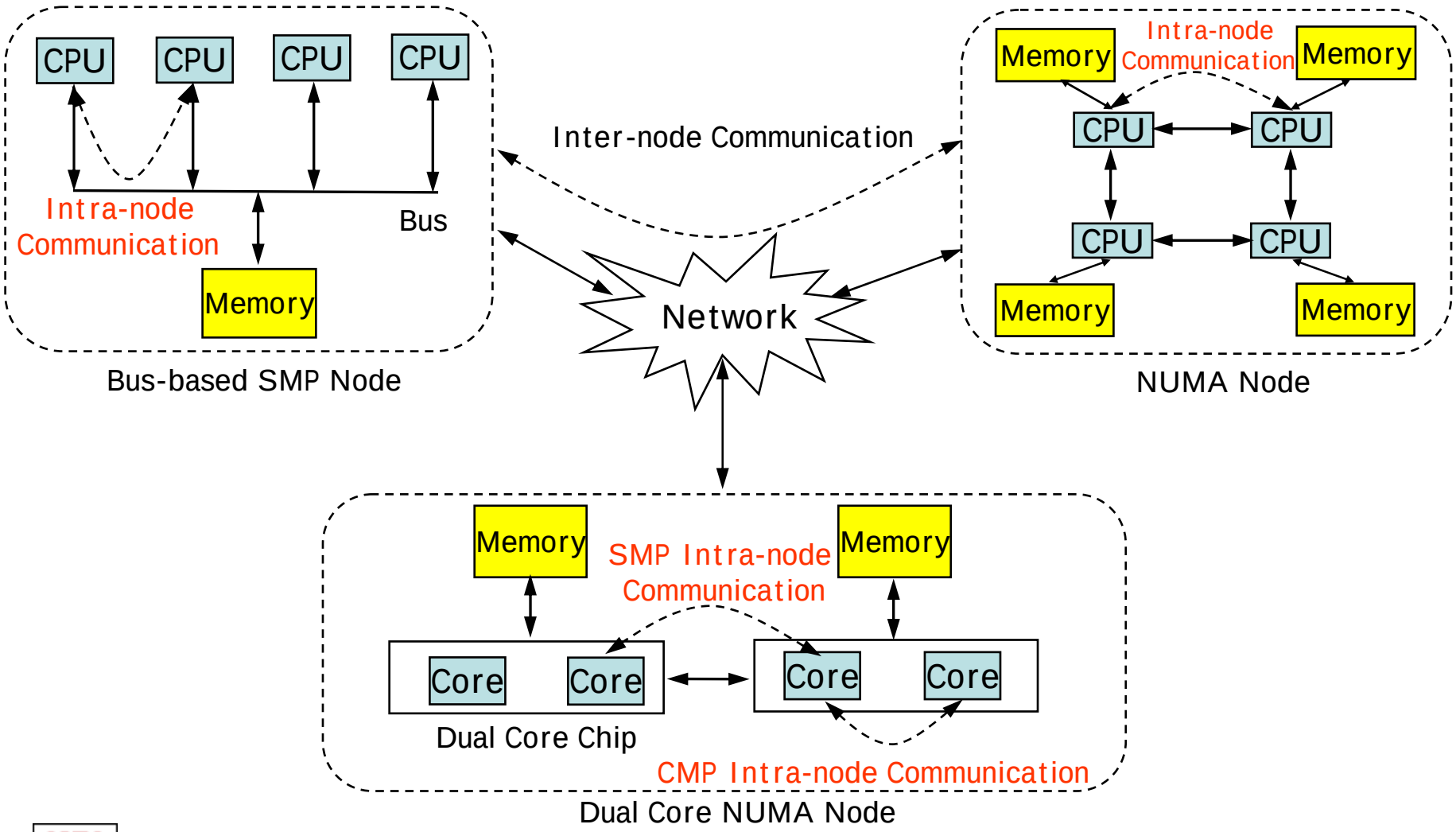
Computer Science & Engineering Department
The Ohio State University

OHIO STATE

Outline

- Introduction and Motivation
- Background
- Design Description
- Performance Evaluation
- Conclusions and Future Work

SMP Based Cluster



Motivation

- Advances in processor and memory architecture
  - NUMA systems
  - Multi-core systems
- Good scalability
  - Large SMP systems available
  - E.g Sun's Niagara 2 System has 8 cores on the same chip and can run 64 threads simultaneously
- MPI intra-node communication more critical!
- Goals:
  - To improve MPI intra-node communication performance
  - To reduce memory usage

Outline

- Introduction and Motivation
- Background
- Design Description
- Performance Evaluation
- Conclusions and Future Work

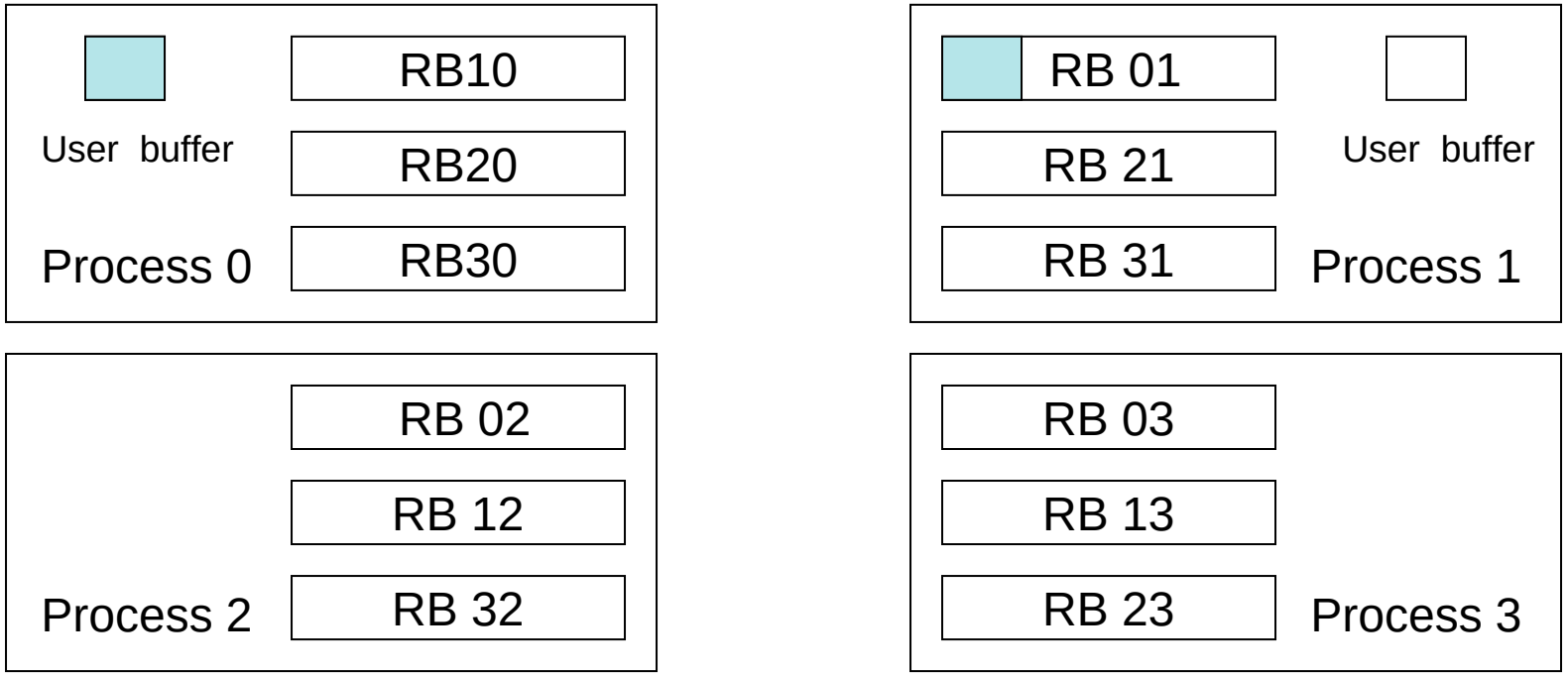
MPI Intra-node Communication

- Existing approaches
  - NIC based loop back
  - Kernel assisted memory mapping
  - User space memory copy
- Advantages of user space memory copy
  - Good performance
  - Portability
- User space memory copy is deployed by many MPI implementations
  - MVAPICH
  - MPICH-MX
  - Nemesis

MVAPICH

- MVAPICH: High performance MPI on InfiniBand clusters developed by OSU
  - Based on MPICH
  - MVAPICH and MVAPICH2 are currently being used by more than 405 organizations worldwide
  - Latest release: MVAPICH-0.9.8 & MVAPICH2-0.9.5
  - http://nowlab.cse.ohio-state.edu/projects/mpi-iba/index.html

Intra-node Communication Design in MVAPICH



- RBxy: Receive Buffers
  - Buffers shared between processes
  - x: sender, y: receiver

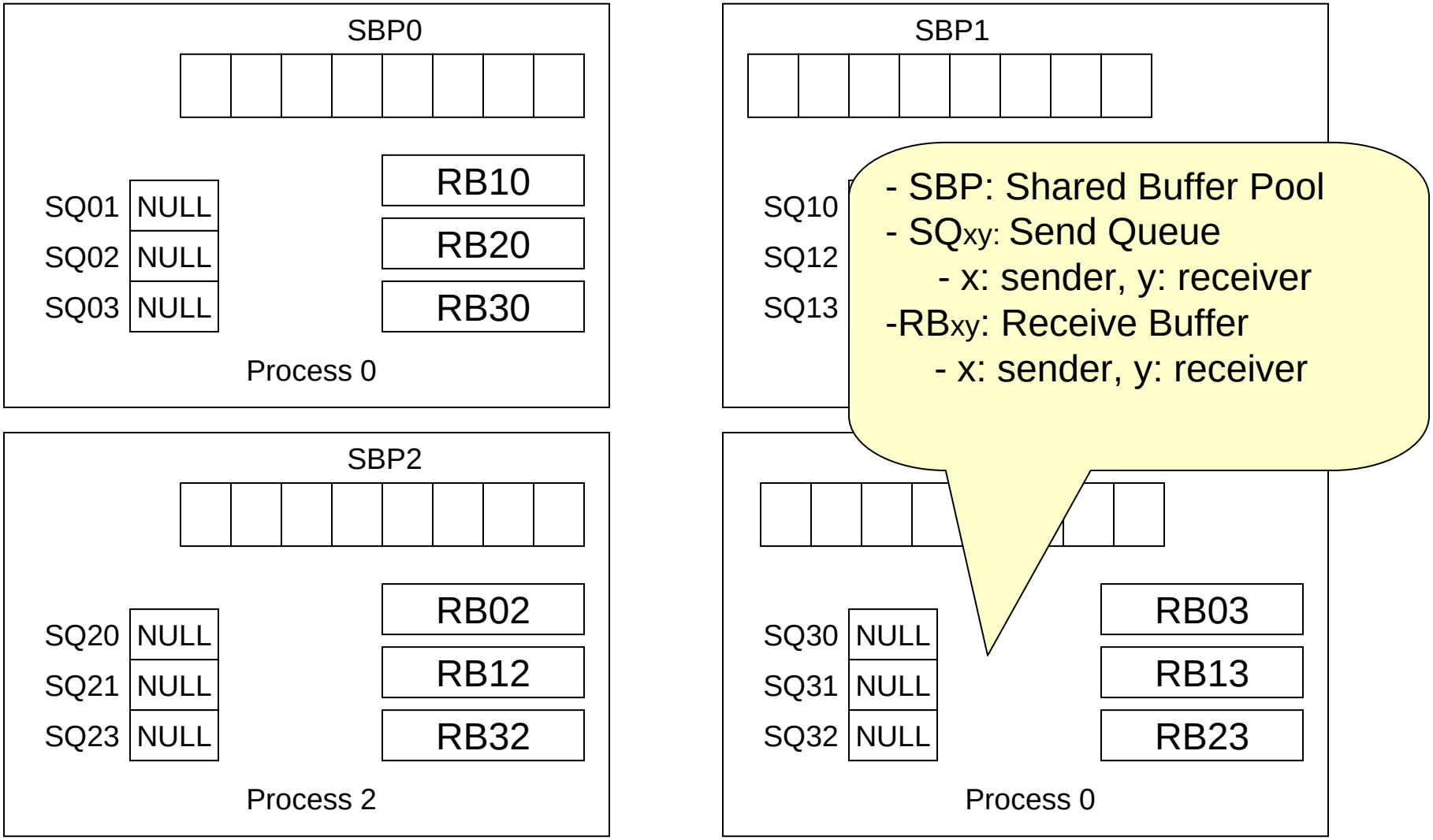
Analysis of the Current Design

- Advantages
  - Lock-free
  - Messages in-order
- Flaws
  - Large memory usage
    - Not scalable
  - Inefficient in cache utilization
    - Need to walk through the receive buffer
    - Performance is not optimized

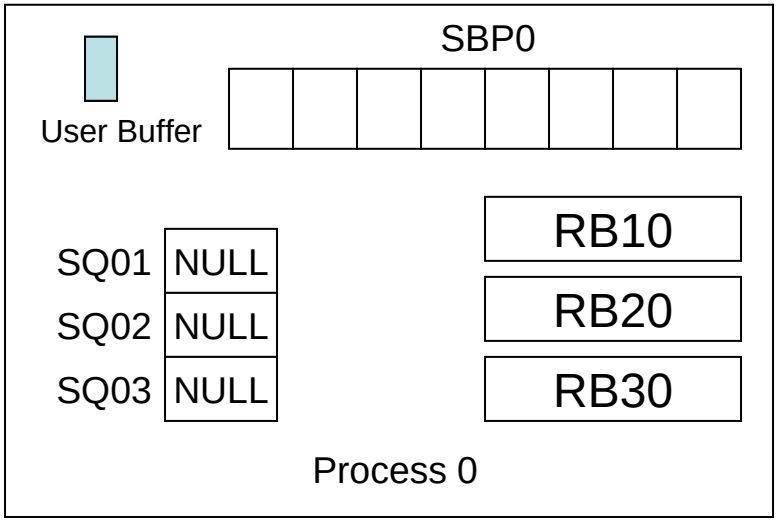
Outline

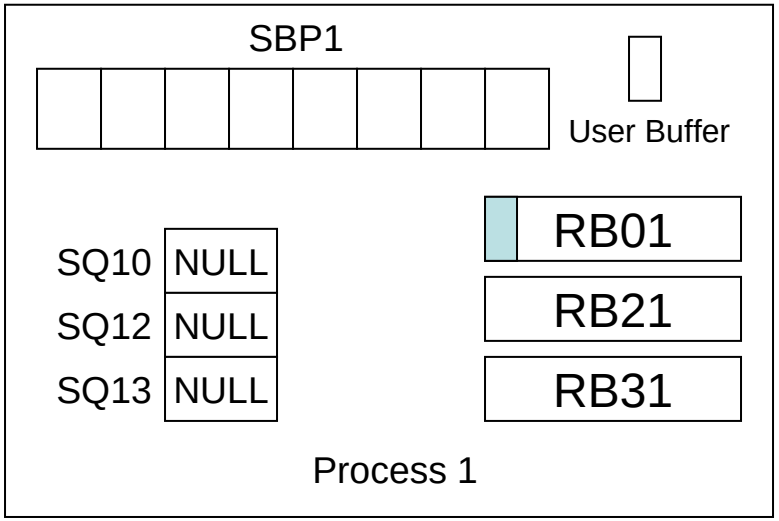
- Introduction and Motivation
- Background
- Design Description
- Performance Evaluation
- Conclusions and Future Work

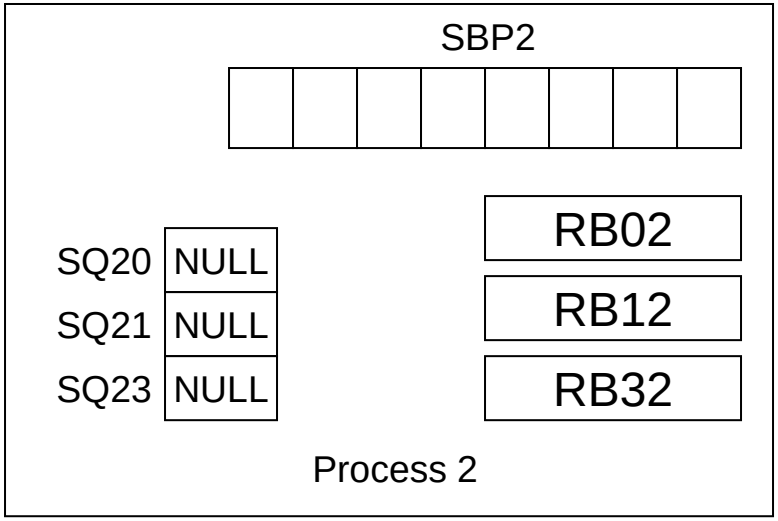
Data Structures

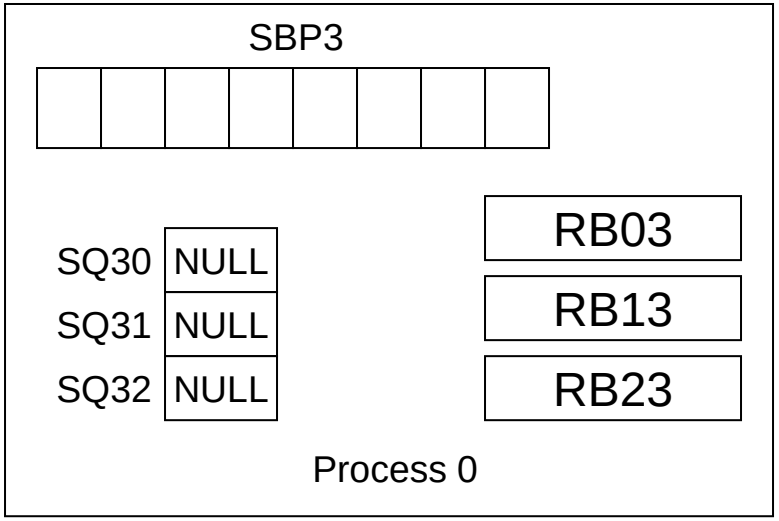


Small Message Transfer

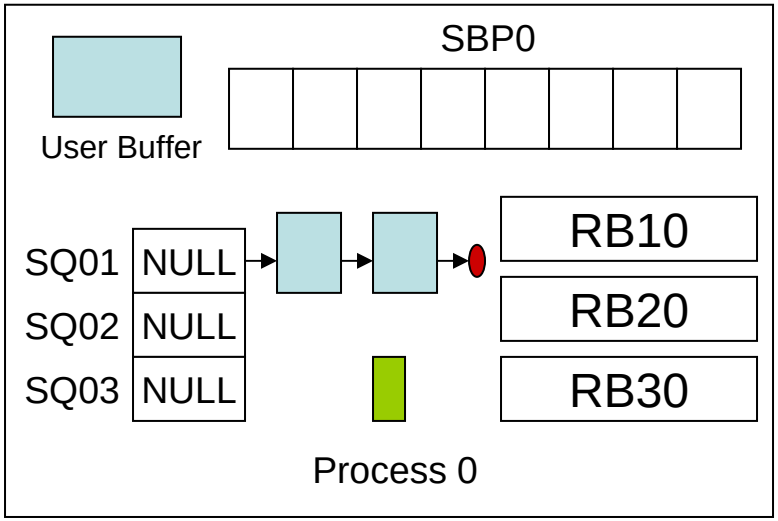


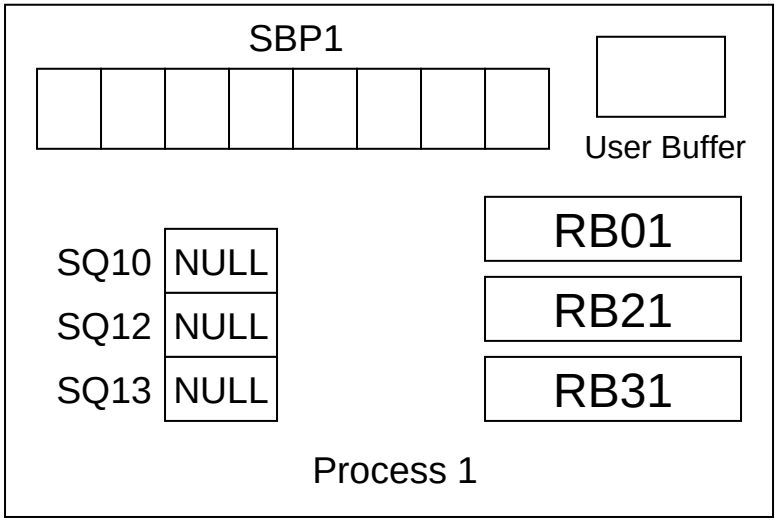


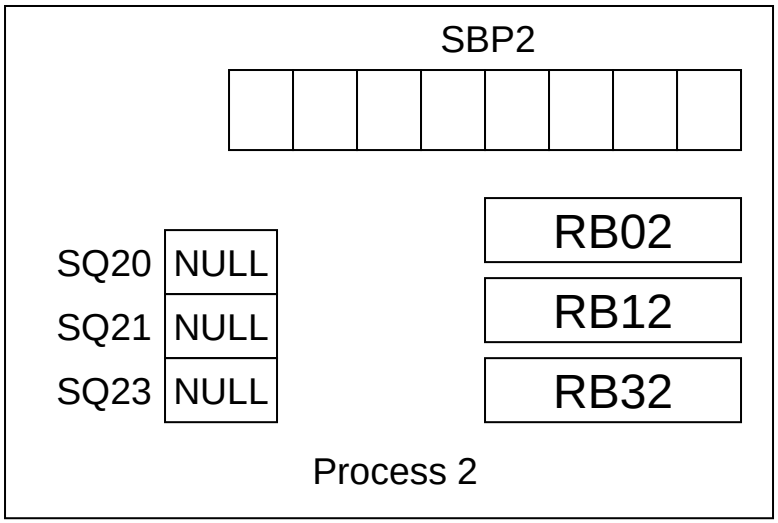


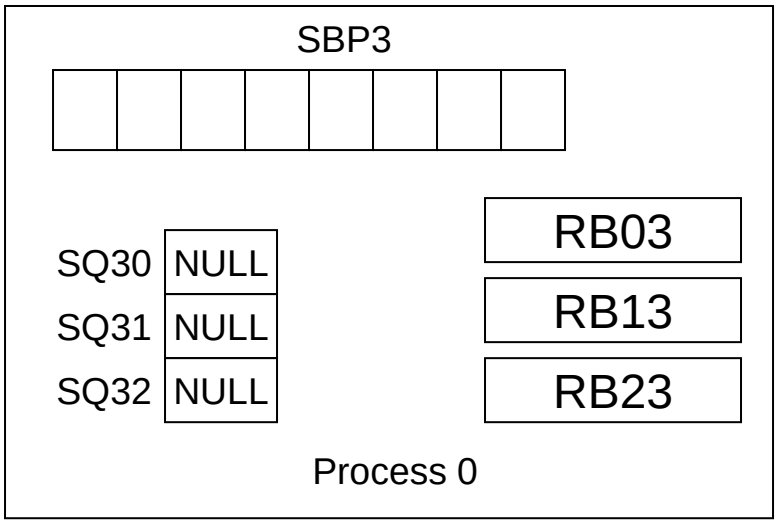


Large Message Transfer









Analysis of the New Design

- Lock free
- Messages in-order
  - Control messages are going through receive buffers
- Efficient in cache utilization
  - Small messages: small receive buffer, likely in the cache
  - Large messages: chances of buffer reuse improved
- Efficient memory usage
  - Receive buffers become smaller
  - Large message buffers are shared among all the connections

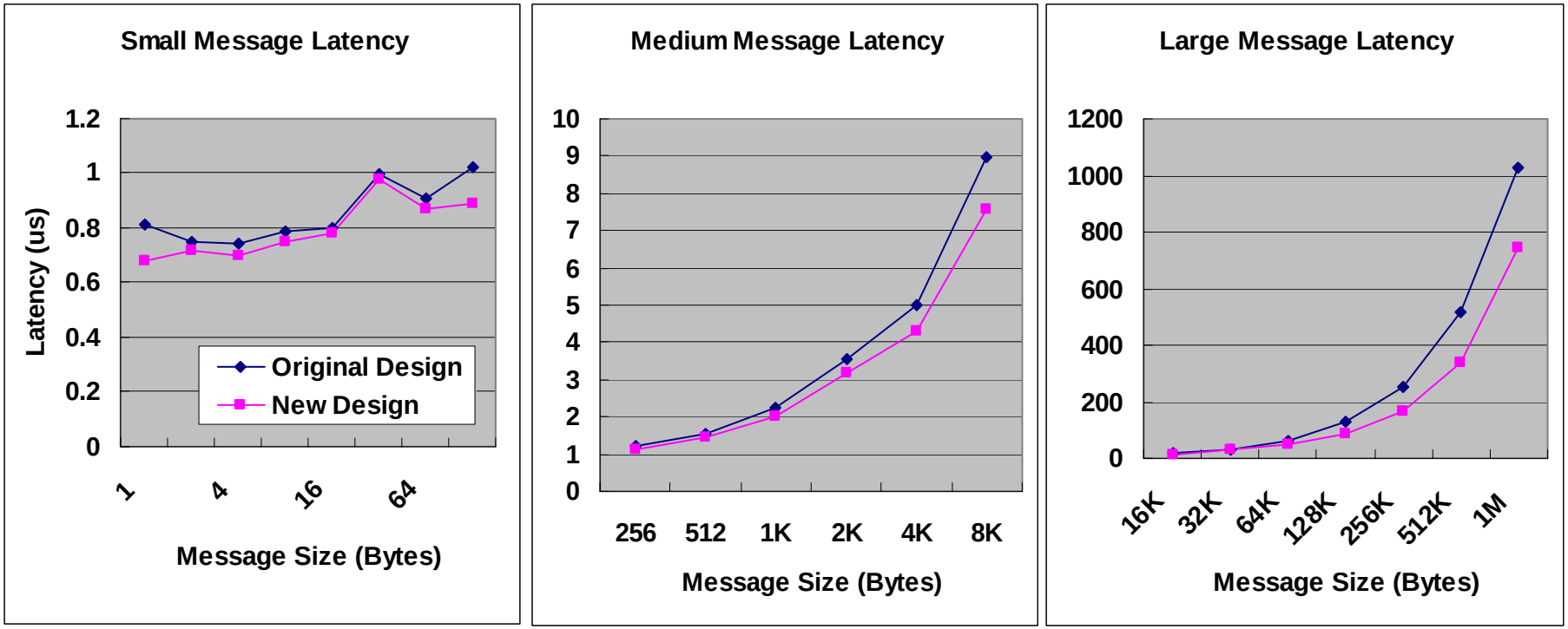
Outline

- Introduction and Motivation
- Background
- Design Description
- Performance Evaluation
- Conclusions and Future Work

Experimental System Setup

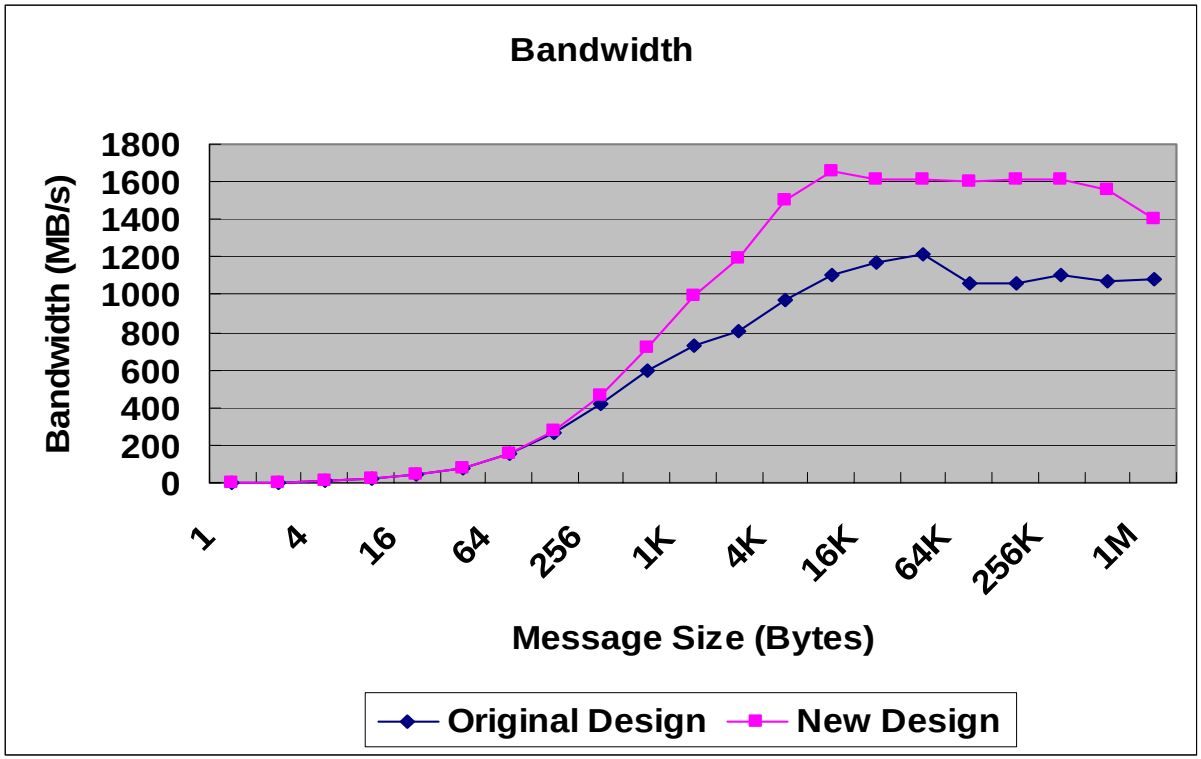
- NUMA Cluster
  - Two nodes connected by InfiniBand
  - Each node has four AMD Opteron processors, 2.0GHz
  - 1MB L2 cache
  - Linux 2.6.16
- Multi-core Cluster
  - Two nodes connected by InfiniBand
  - Each node has four dual-core AMD Opteron processors, 2.0GHz
    - Two cores per chip, two chips in total
  - Each core has 1MB L2 cache
  - Linux 2.6.16

Latency on NUMA Cluster



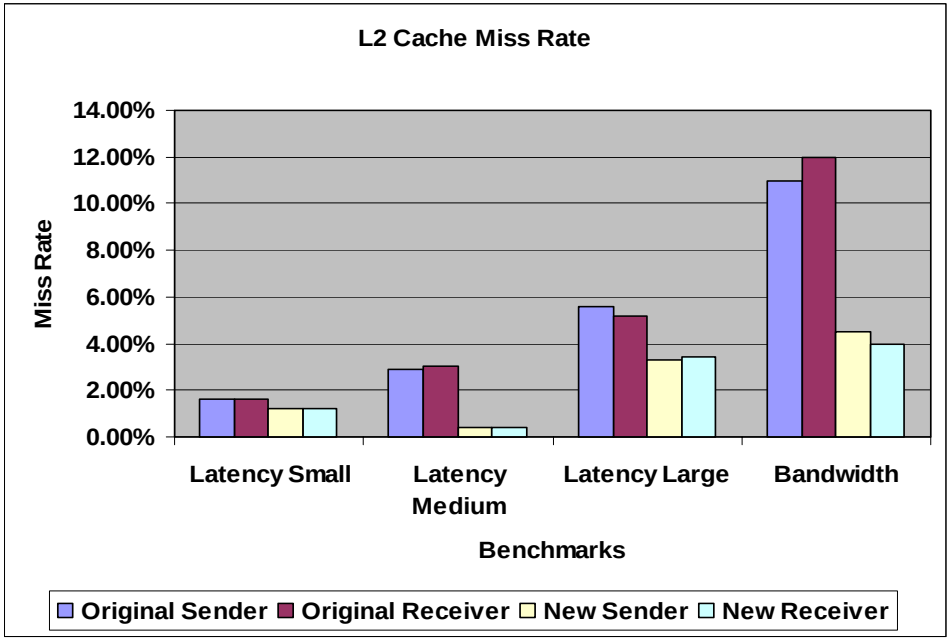
- Latency for small and medium messages is improved by up to 15%
- Latency for large messages is improved by up to 35%

Bandwidth on NUMA Cluster



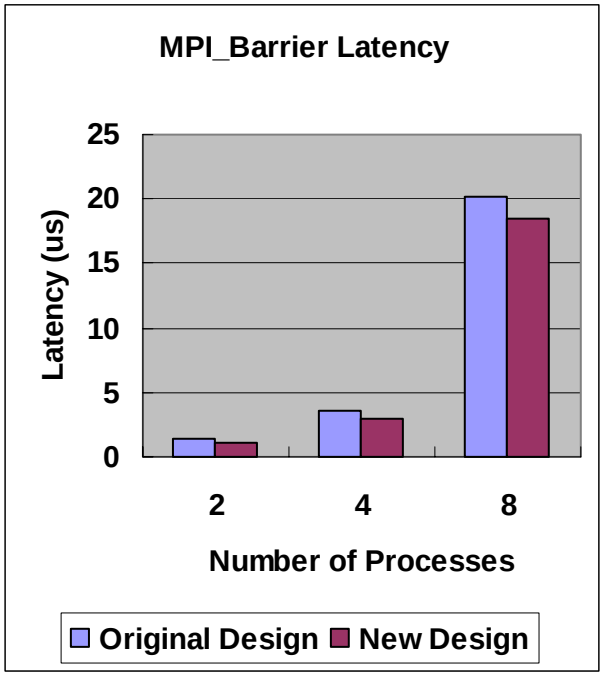
- Bandwidth is improved by up to 50%

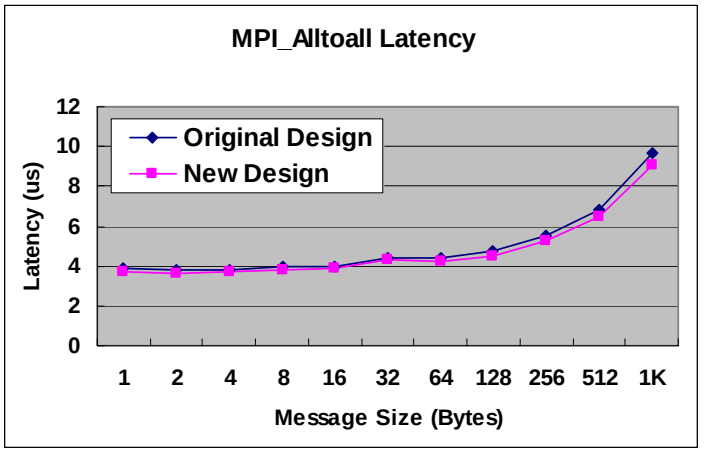
L2 Cache Miss Rate

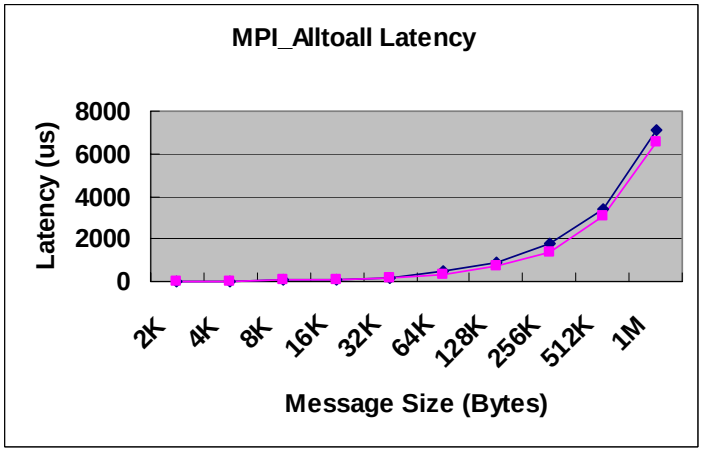


- Tool: Valgrind
- The improvement in latency and bandwidth comes from better L2 cache utilization

Collectives on NUMA Cluster

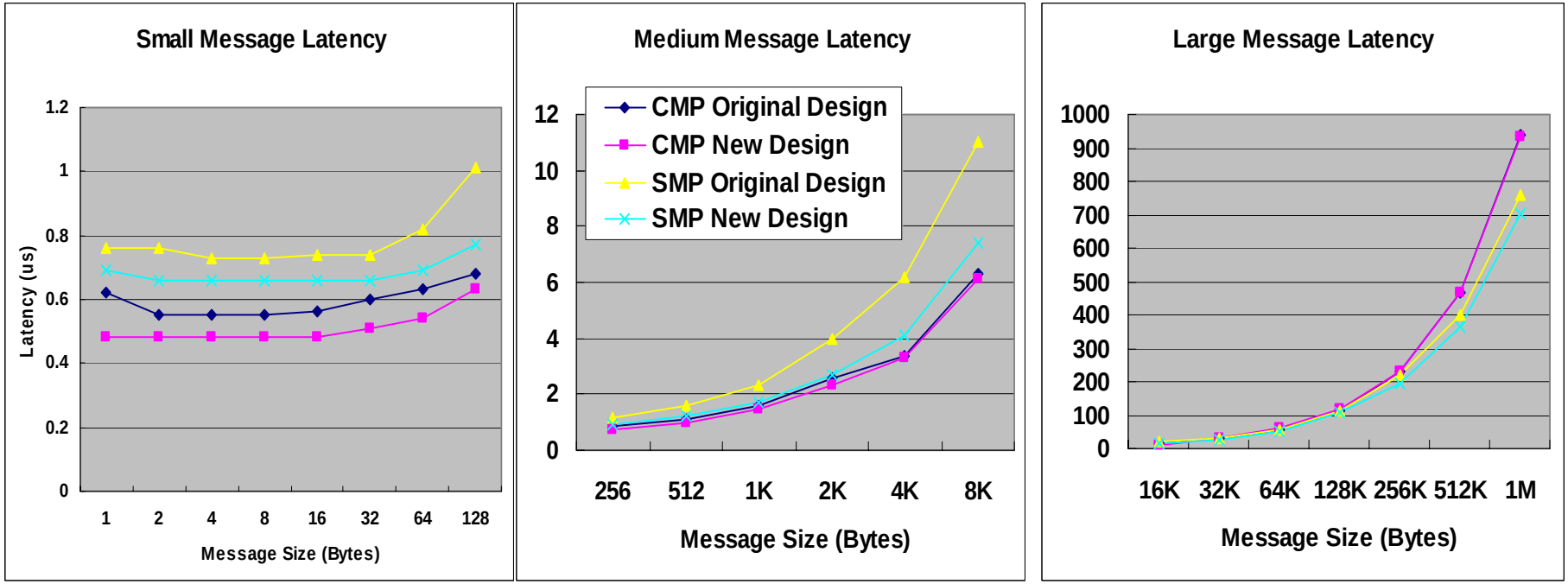






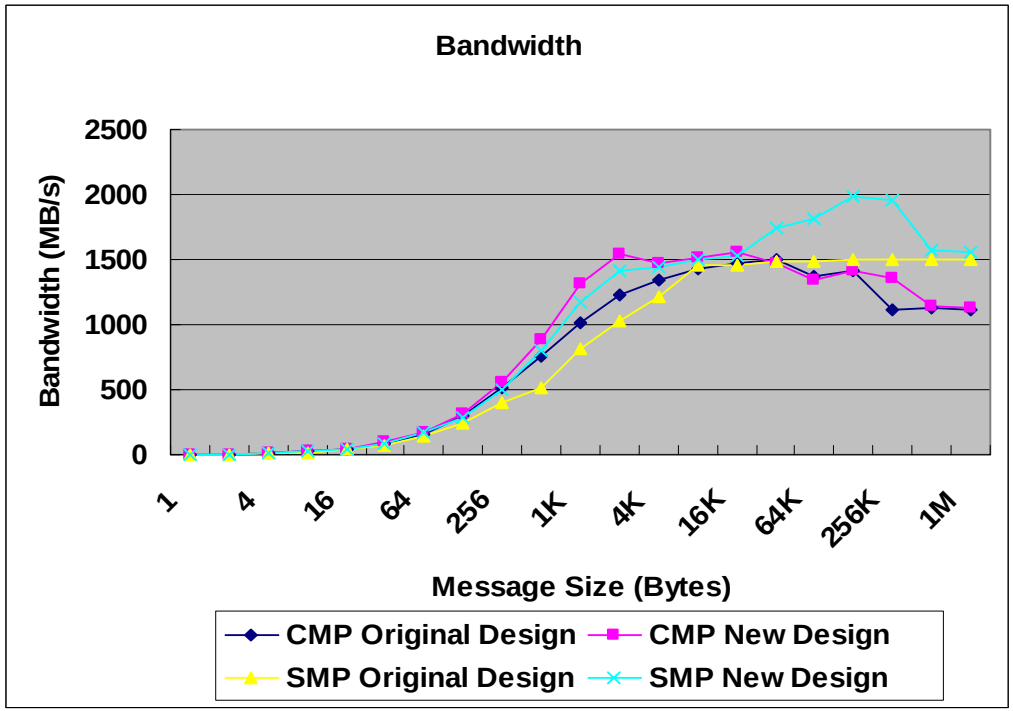
- MPI_Barrier latency is improved by up to 19%
- MPI_Alltoall latency is improved by 10%

Latency on Multi-core Cluster



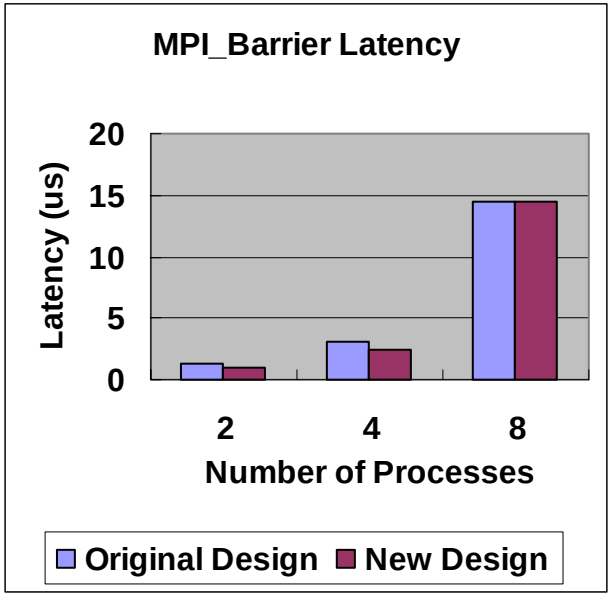
- CMP latency is lower than SMP latency for small messages, but higher for large messages
  - Cache transaction vs. memory contention
- The new design improves SMP latency for all the messages
- The new design improves CMP latency for small messages

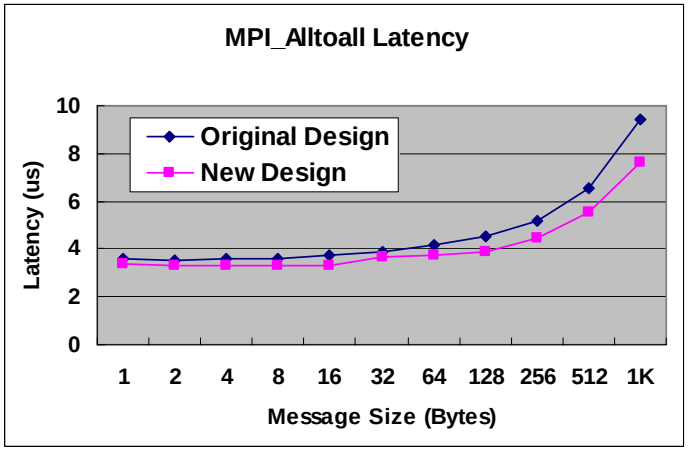
Bandwidth on Multi-core Cluster

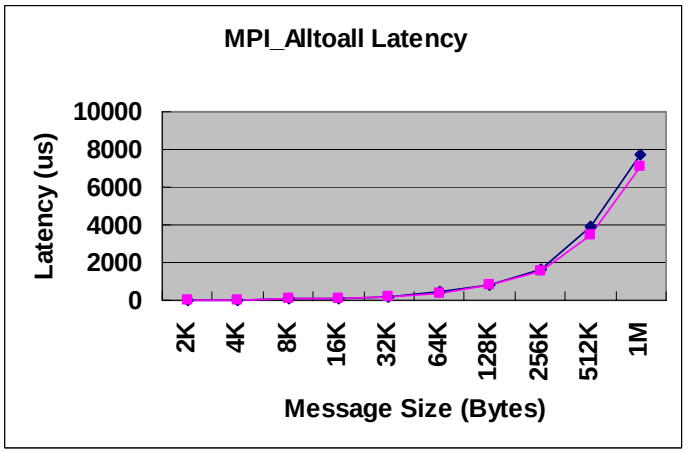


- The new design improves SMP bandwidth significantly
- The new design also improves CMP bandwidth for small and medium messges

Collectives on Multi-core Cluster







- The new design improves collective performance on multi-core cluster

Outline

- Introduction and Motivation
- Background
- Design Description
- Performance Evaluation
- Conclusions and Future Work

Conclusions

- Designed and implemented high-performance and scalable MPI intra-node communication support
  - Lock free
  - Efficient cache utilization
  - Efficient memory usage
- Evaluated on NUMA and multi-core systems
  - Both point-to-point and collective performance has been improved significantly

Future Work

- Application level study
- Evaluation on larger systems
- Further optimizations on multi-core systems

Acknowledgements

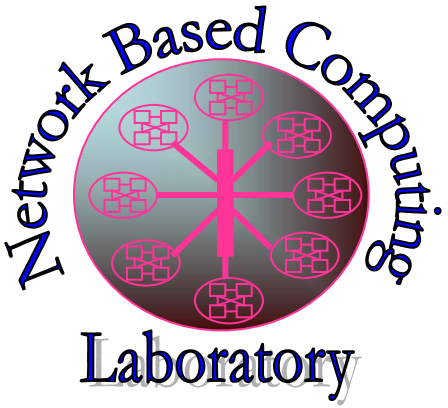
Our research is supported by the following organizations

- Current Funding support by

- Current Equipment support by


Thank you

{chail, hartonoa, panda}@cse.ohio-state.edu



Network-Based Computing Laboratory

http://nowlab.cse.ohio-state.edu/