Efficient Hardware Multicast Group Management over InfiniBand

Amith R. Mamidala, Hyun-Wook Jin and
Dhabaleswar K. Panda

Department of Computer Science and Engineering
Ohio State University

{mamidala,jinhy,panda}@cse.ohio-state.edu

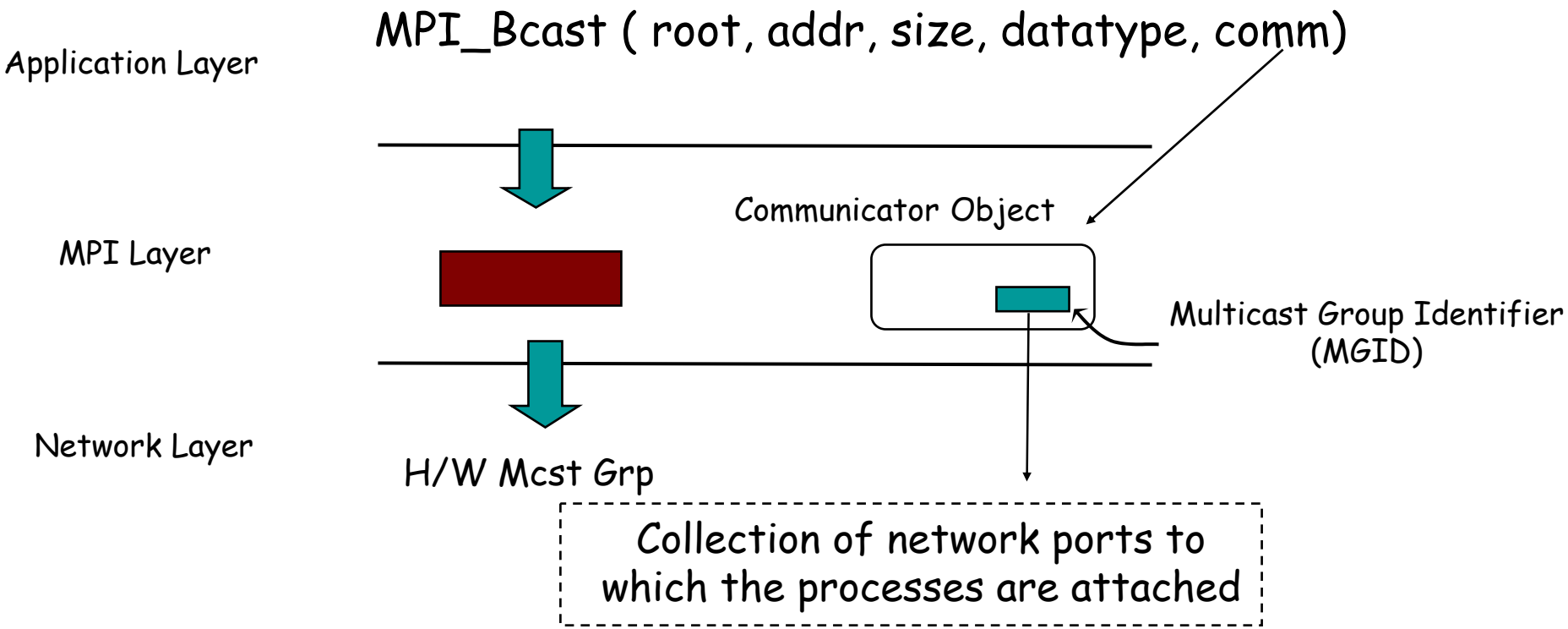
Presentation Outline

- Background
- Problem Statement
- Detailed Design
- Performance Evaluation
- Conclusions & Future Work

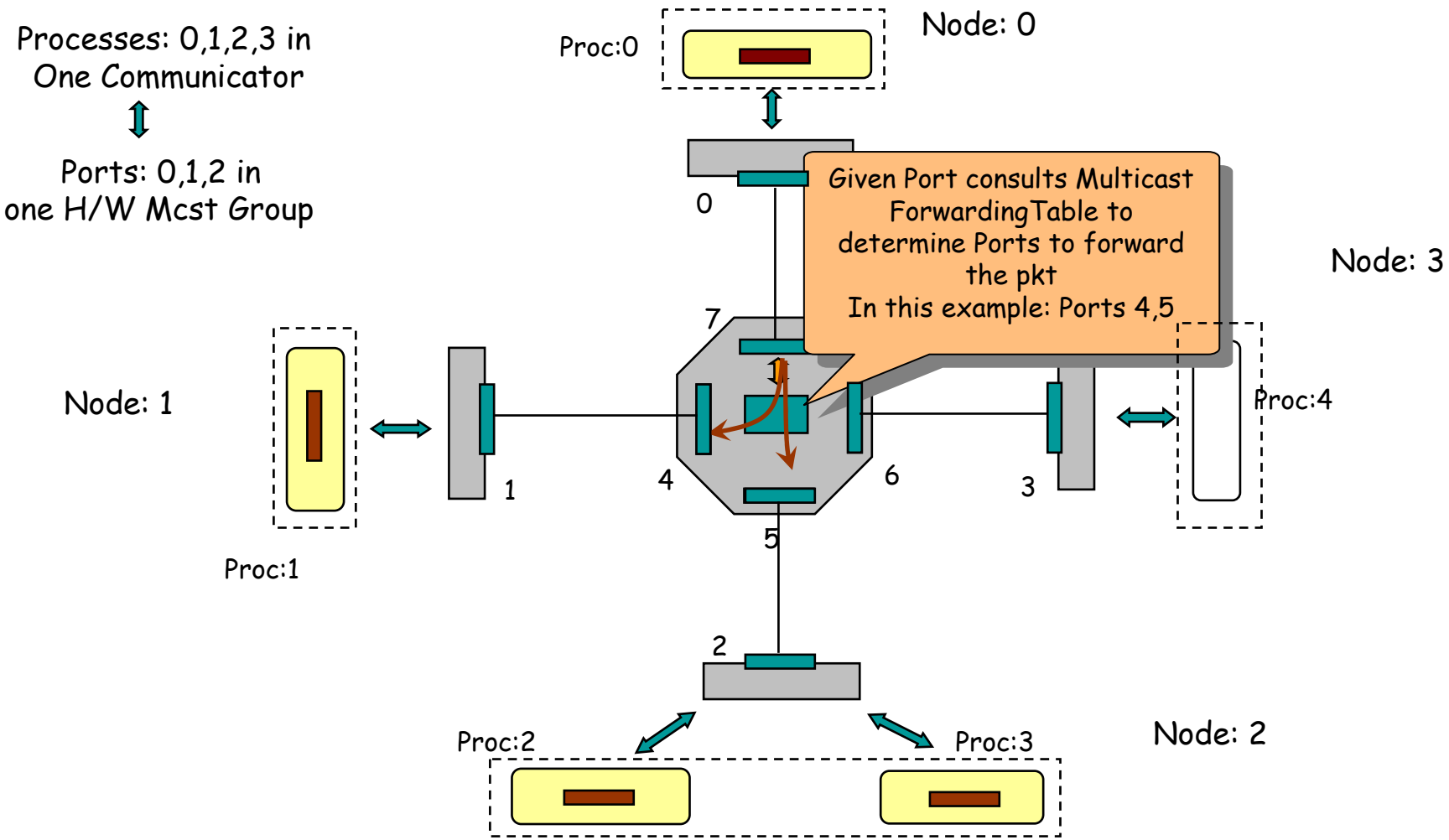
Background

- MPI Collectives
  - take advantage of network features
- InfiniBand (IBA)
  - H/W Mcst Support
- Efficient and Scalable Collectives
  - J. Liu, A. Mamidala, D.K. Panda, "Fast and Scalable MPI-Level Broadcast using InfiniBand's Hardware Multicast Support", IPDPS 04
  - A. Mamidala, J.Liu, D.K. Panda, "Efficient Barrier and Allreduce on IBA clusters using hardware multicast and adaptive algorithms", Cluster 04

MPI_Bcast over H/W Mcst



MPI Communicator & H/W Mcst Grp



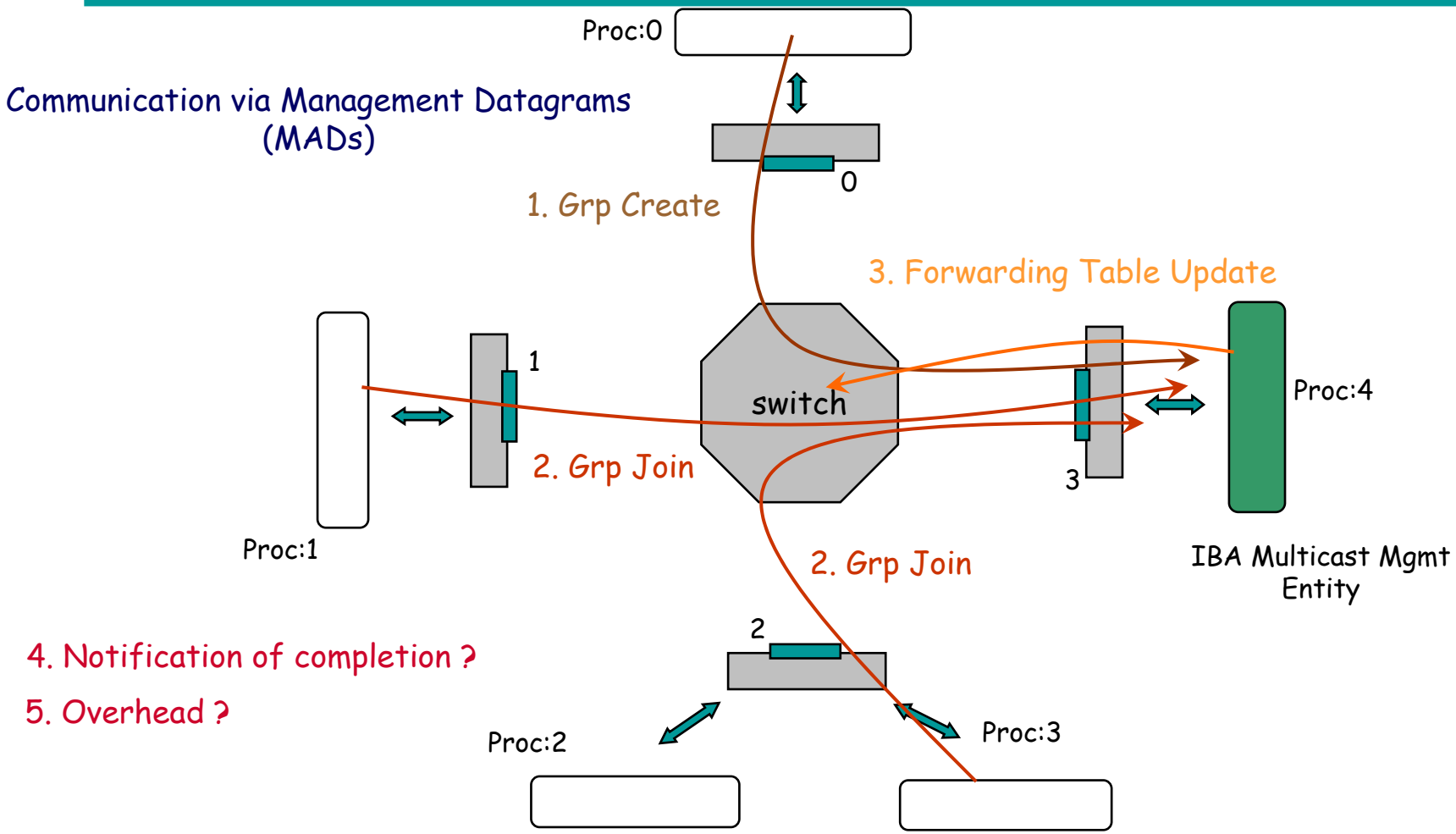
Presentation Outline

- Background
- Problem Statement
- Detailed Design
- Performance Evaluation
- Conclusions & Future Work

Limitations

- Earlier study using single communicator
  - MPI_COMM_WORLD
  - Static initialization
- Need of supporting multiple communicator
  - Fluent, etc.
  - Dynamic Process Management (MPI-2)
- Efficient mechanism for supporting multiple communicator
  - H/W Mcst Groups constructed on the fly
  - involves an external management entity

InfiniBand H/W Mcst Grp Construction



Research Challenges

- Mapping Communicators to H/W Mcst Grps
  - No explicit mechanism defined for notifying MPI
- Overhead of mapping
  - High Forwarding Table Update overhead
  - Especially critical for large communicator and cluster sizes
- Can we develop an efficient framework for mapping MPI Communicators to H/W Mcst Grps?

Presentation Outline

- Background
- Problem Statement
- Detailed Design
- Performance Evaluation
- Conclusions & Future Work

Design Alternatives

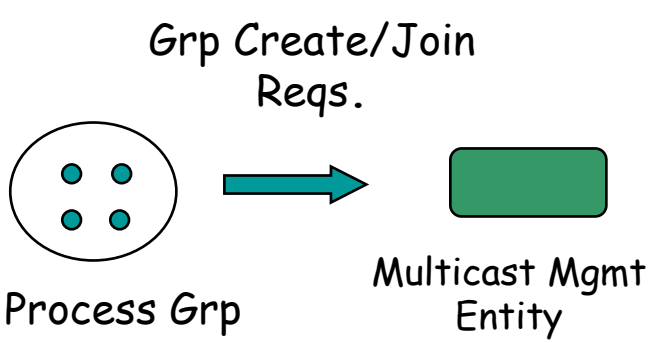
- Propose three design alternatives:
  - Basic approach
    - Notification
    - Incurs Forwarding Table Update overhead
  - Lazy approach
    - Overlapping Notification and Forwarding Table Update
  - H/W Mcst Grp Pool approach
    - Hiding Forwarding Table Update
    - No Notification phase required

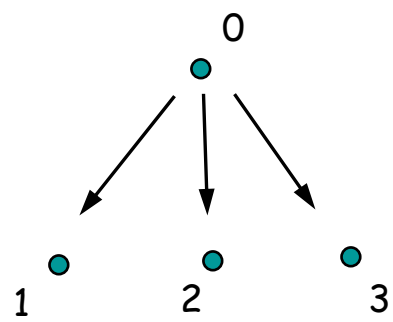
Basic Approach

Step: 1

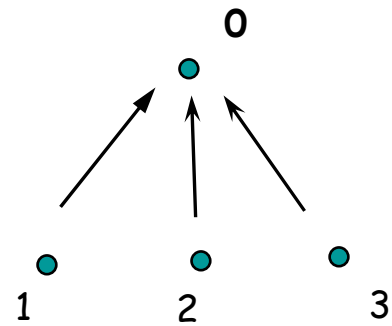
Step: 2

Step: 3





Multicast Ping Message
Over new H/W Mcst Grp



Post Acks after
Receiving Ping Message

Notification

Limitation

- Notification
  - MPI_Comm_create
  - Simple to implement
- Disadvantage
  - Notification is a blocking call
  - High overhead for large communicators
    - Forwarding Table Update
- Question
  - Can we overcome the high overhead of Forwarding Table Update?

Lazy Approach

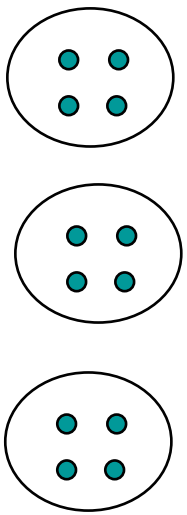
- Step 1: Same as Earlier Approach
- Step 2: Notification
  - non blocking
  - Overlapping Forwarding Table Update
- Implementation
  - Root returns immediately after posting ping mesg
  - Ack posting and collection done asynchronously
  - Time stamps stored in the communicator object
  - Ptp fall back for communication

Drawbacks

- Disadvantages:
  - Utility of H/W Mcst Grp reduced
  - Like to use it as soon as possible
  - Overlapping does not solve the problem
- Question:
  - Can we avoid the high overhead of the Forwarding Table Update to make H/W Mcst Grps readily available?

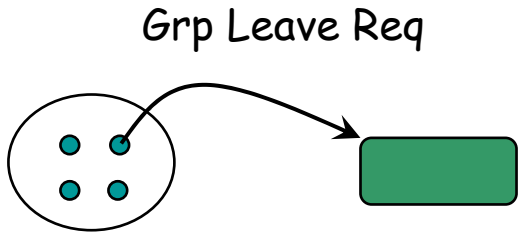
H/W Mcst Grp Pool Design

Step: 1



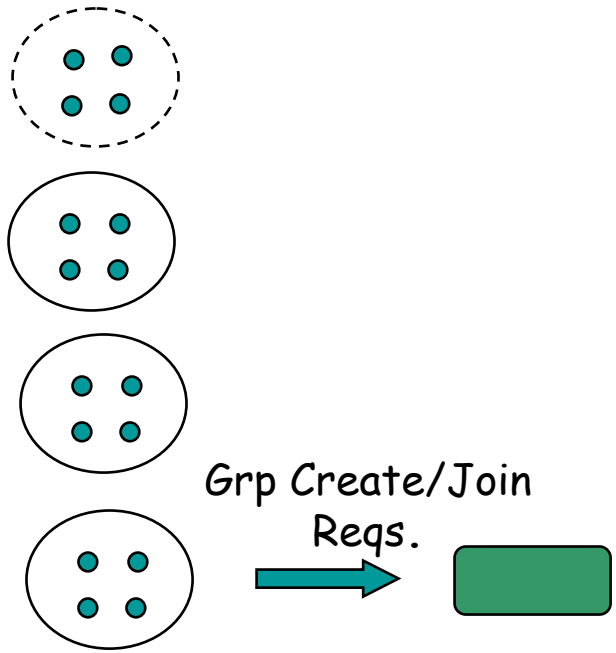
Initial H/W
Mcst Grp Pool

Step: 2



Non-participating
Nodes leave H/W Mcst Grp

Step: 3



Replenishing H/W Mcst Grp Pool

Benefits

- Advantages:
  - H/W Mcst Grp can be immediately used
  - Most of the job out of the critical path
- Implementation
  - Notification required for newly replenished groups
  - The size of the H/W Mcst Grp Pool can be tuned for different applications

Presentation Outline

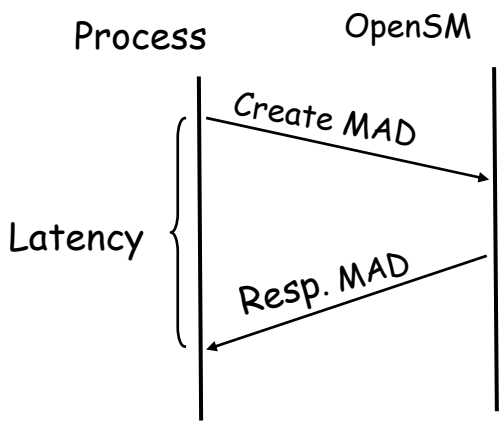
- Background
- Problem Statement
- Detailed Design
- Performance Evaluation
- Conclusions & Future Work

Experimental TestBed

- Cluster of Intel Xeon 2.66 MHz,512 KB L2 Cache, MT23108 IBA HCAs
- OpenSM: Multicast Mgmt Entity (Subnet Manager & Subnet Administrator)

Experiments

1. Effect of different parameters such as the number of outstanding MADs and transaction timeout values of OpenSM on notification
2. Latency of Basic H/W Mcst Grp Operations



Experiments contd..

3. Effective MPI_Bcast Latency for different iteration count

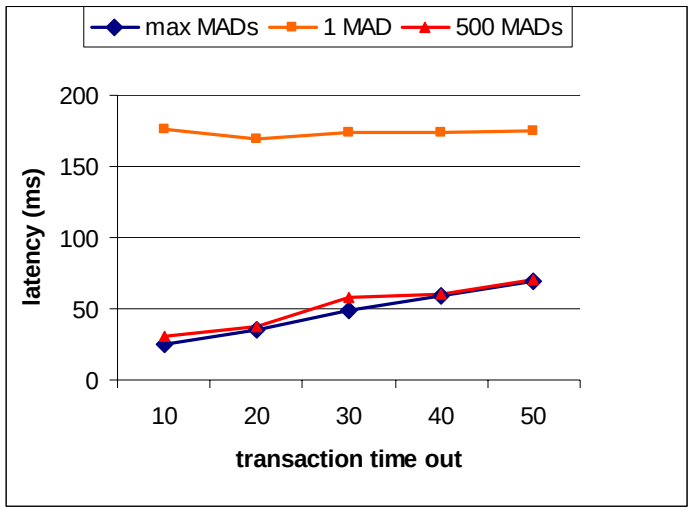
Comm. sizes
- 1. MPI_Comm_create
- 2. MPI_Bcast } Iter. Count

4. Effective MPI_Bcast Latency for different computation time

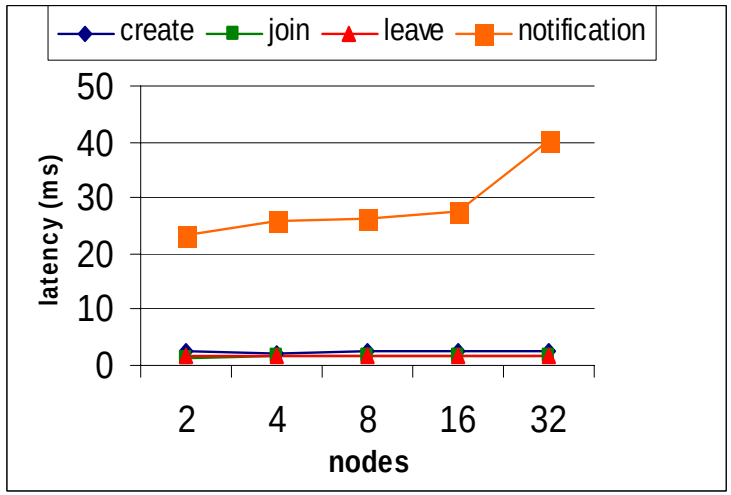
Comm. sizes
- 1. MPI_Comm_create
- 2. Computation
- 3. MPI_Bcast } 1000 Iter.

Latencies of basic Multicast Grp Set-up operations

Effect of different parameters of OpenSM on multicast testing



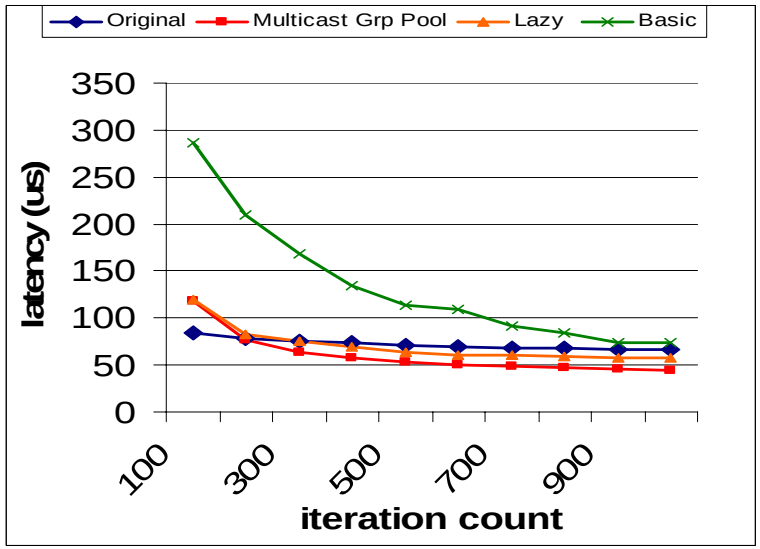
Latency of basic set-up operations



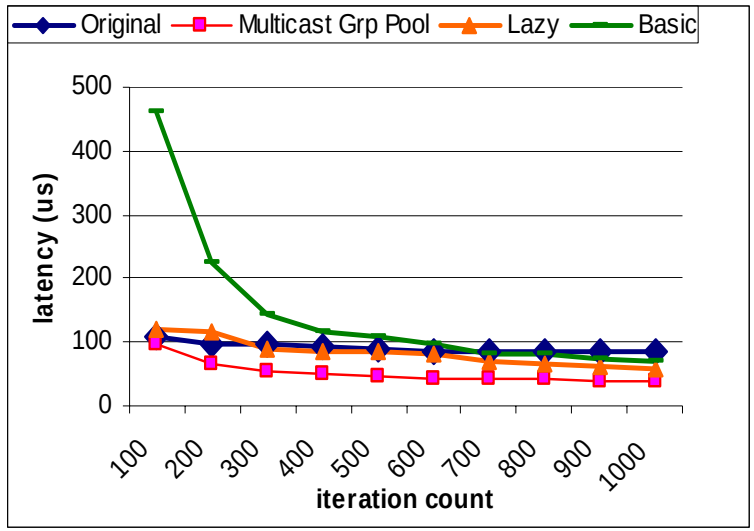
- Latency of issue of multicast grp set-up operations much smaller compared to setting up routing entries

Effective MPI_Bcast Latency

Effective Latency of MPI_Bcast: 16 nodes



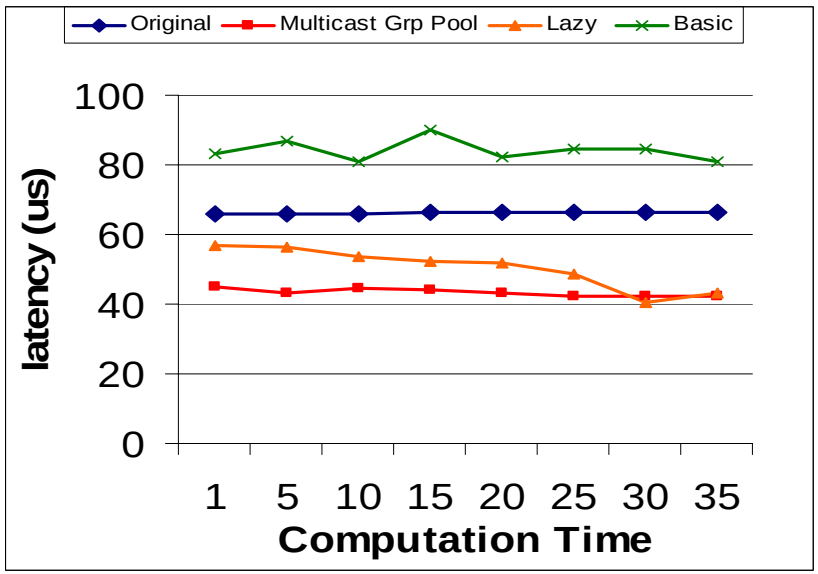
Effective Latency of MPI_Bcast: 32 nodes



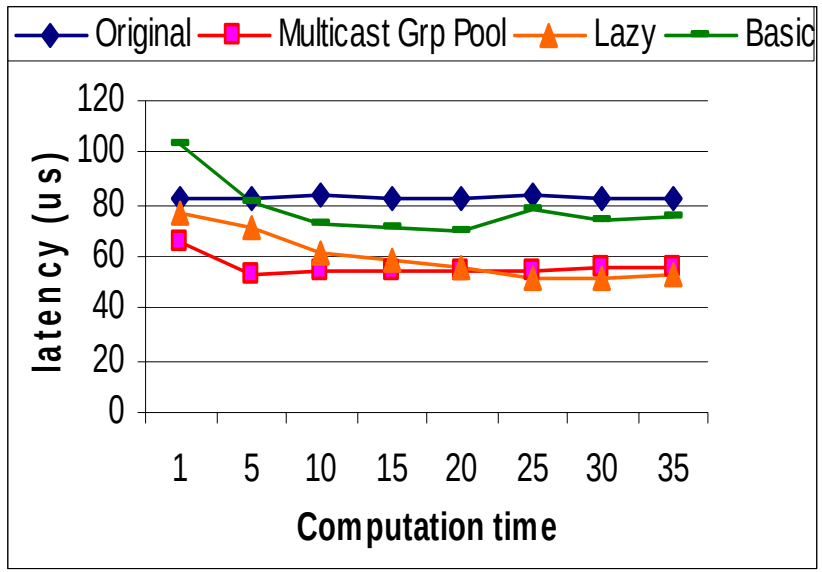
- Multicast Grp pool improves effective latency by a factor of as much as 2.42 for 16 nodes and 4.8 for 32 nodes compared to Basic approach

Effective MPI_Bcast Latency

Effective Latency of MPI_Bcast: 16 nodes



Effective Latency of MPI_Bcast: 32 nodes



- Multicast Grp pool improves effective latency by a factor of as much as 1.95 for 16 nodes and 1.80 for 32 nodes compared to Basic approach

Presentation Outline

- Background
- Problem Statement
- Detailed Design
- Performance Evaluation
- Conclusions & Future Work

Conclusions & Future Work

- Design Alternatives for efficiently mapping MPI Communicators to H/W Mcst Grps
- Three designs proposed:
  - Basic, Lazy and H/W Mcst Pool
  - H/W Mcst Pool design performs best
- Evaluated performance using OpenSM
- Future work: Application Level Evaluation (Fluent, Pallas)
- Evaluation with Gen2 verbs

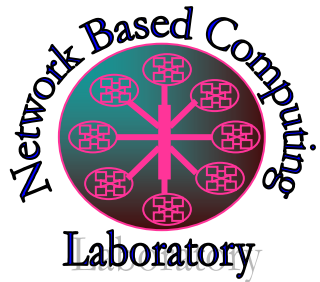
Acknowledgements

Our research is supported by the following organizations

- Current Funding support by

- Current Equipment donations by

Web Pointers



http://www.cse.ohio-state.edu/~panda/

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

MVAPICH Web Page

http://nowlab.cse.ohio-state.edu/projects/mpi-iba/

Questions ?