Efficient and Scalable NIC-Based Barrier over Quadrics and Myrinet with a New NIC-Based Collective Protocol

W. Yu*, D. Buntinas§, R.L. Graham+, and D.K. Panda*

The Ohio State University*

Argonne National Laboratory§

Las Alamos National Laboratory+

Presentation Outline

- Motivation of the collective protocol
- Barrier Algorithms
- Implementation over Myrinet and Quadrics
- Performance Evaluation
- Conclusions and Future Work

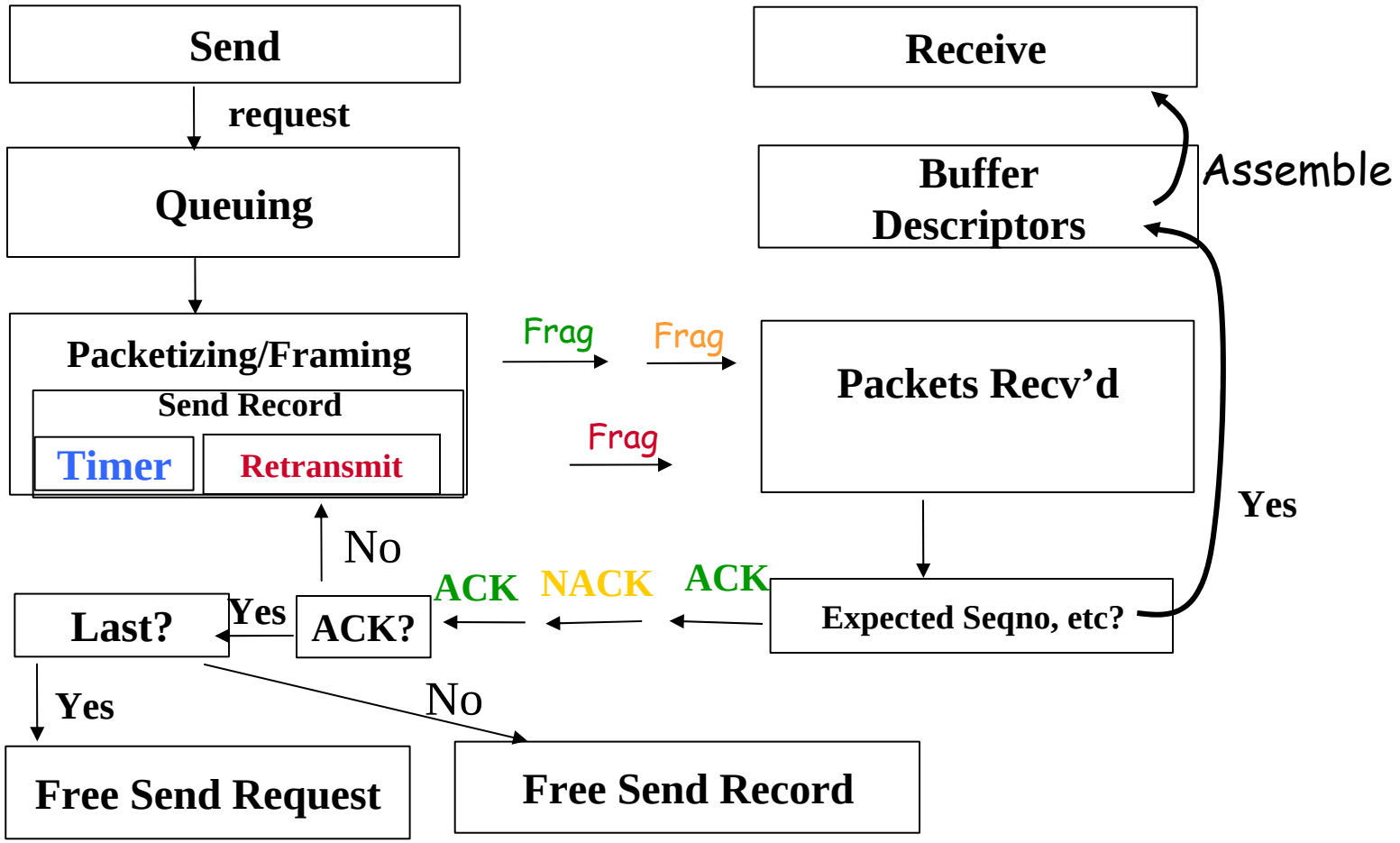
Motivation

- Programmable NIC processors
  - Myrinet, Quadrics, Alteon, etc.
  - Offloading Communication processing to NIC
- NIC-based collective operations
  - barrier, broadcast, reduce
- Unified NIC-based collective Protocol?
  - Minimize the resource requirement for scalability
  - Supporting multiple collective communication
  - Reduce communication processing and achieve better Performance

NIC Communication Processing

- Queuing
- Packetizing/Framing
- Bookkeeping
- Assembly
- Flow/Error Control

Point-to-Point Communication



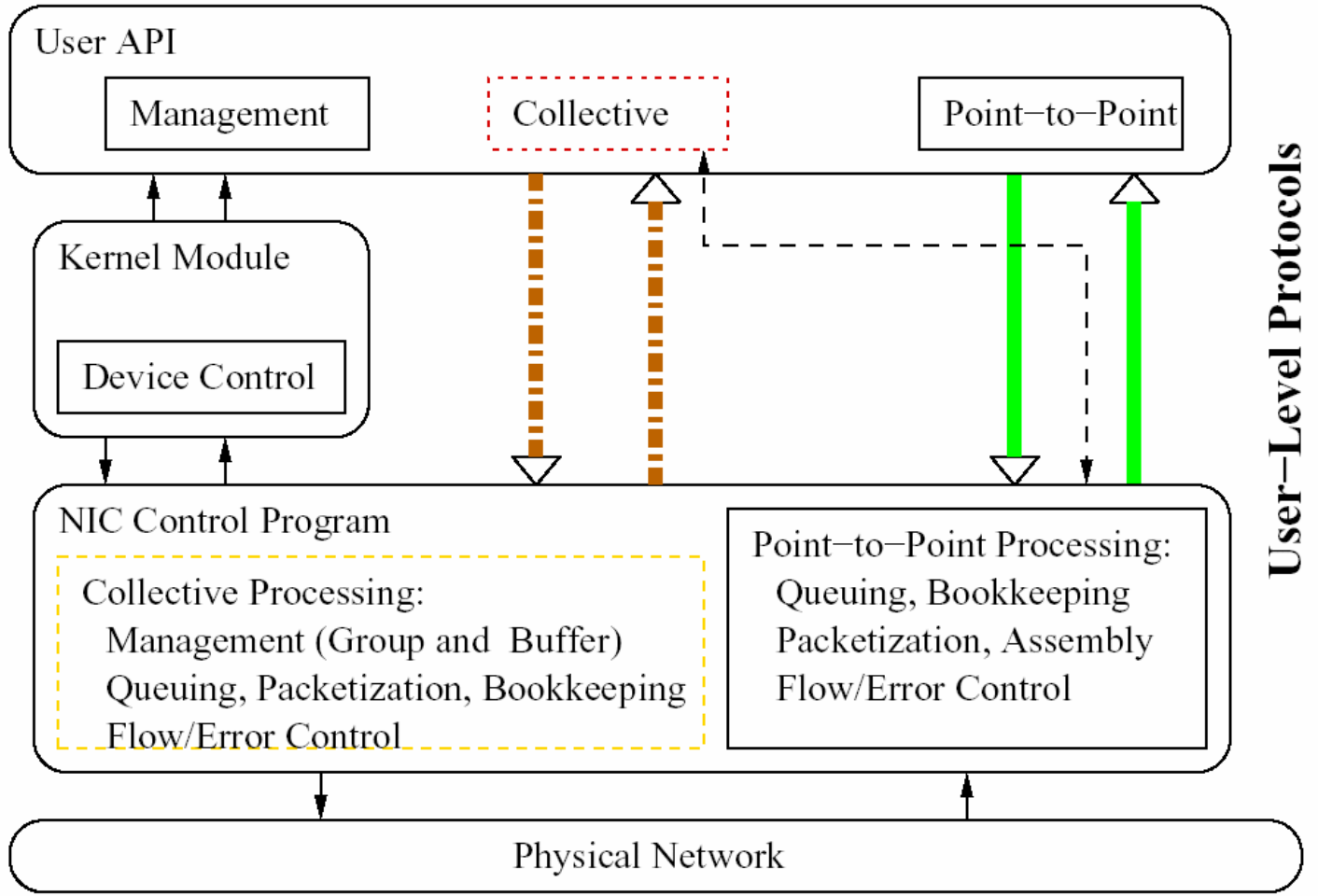
Problems with this Approach for Collective Communication

- Major problems
  - Delays in multiple processing steps
  - Redundant processing for collectives
- Queuing
  - Multiple Connection to a single NIC
  - Multiple Queues to different NICs
  - Results in delay if there are requests in the queues.
- Packetizing/Framing
  - Send buffers needs to be available
  - Packet the data by Copying/DMAing
  - Results in delay if send buffer is not immediately available

Problems with this Approach -- Continued

- Bookkeeping
  - A send record per packet, expecting an ACK
  - High resource requirement
- Assembly
  - Assemble packets into a message
- Flow/Error Control
  - Retransmit the packet if a send record timed out
  - Could also rewind the communication queues

Architecture of Our NIC-Based Collective Protocol



Design of A New NIC-based Collective Protocol

- Queuing
  - Group oriented
  - additional queues created for collectives
  - No need to go through Multiple queues
  - Better chaining of communication steps in a collective
- Packetizing/Framing
  - No need of send buffers for barrier,
  - Not if packets received or completed sending can be reused
  - Save copying/DMAing and skip the wait for free send bufers

Design of the Protocol --Continued

- Bookkeeping
  - One entry per collective operation
  - Scalable resource requirement
  - reduce bookkeeping overhead
- Assembly
  - No need for data assembly in a barrier
- Flow/Error Control
  - Collective flow/error control
  - Receiver-driven retransmission if the packet is not arrived
  - Reduce number of communication packets
  - Avoid rewinding the collective communication queues due to packet loss of other requests

Case Study with Barrier for Myrinet/GM

- Myrinet/GM:
  - Host-based barrier over unicast
  - Existing NIC-based barrier (IPDPS'01, CAC'01),
    - Exploit benefits of NIC programmability
    - Layered on top of point-to-point processing at the NIC
- Can collective processing improve barrier performance?
  - Reduced resource requirement
  - Reduced processing overhead
  - Enhanced chaining of communications steps
  - Reduced number of packets

Case Study with Barrier for Quadrics

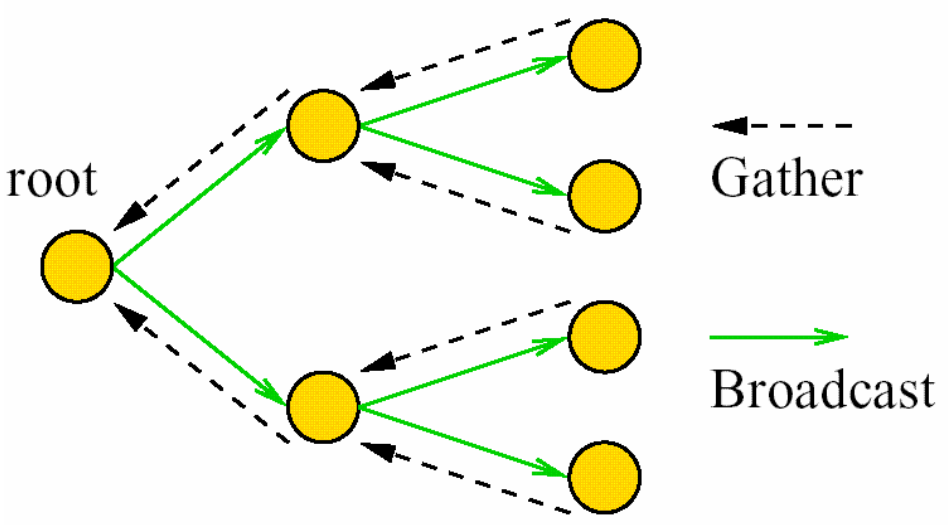
- Quadrics:
  - Hardware Barrier
    - Superior performance only on contiguous nodes (QsNet-II/Elan-4?)
  - Software tree-based barrier
    - Gather/Broadcast
    - Suboptimal performance
- Is the barrier with a NIC-based collective protocol beneficial?
  - Targeted to be on top of unicast communication
    - using no hardware broadcast
  - Hardwared packetizing/framing
  - No data transmission
  - Hardwared reliability
  - Possible benefits:
    - fast chaining of communication steps
    - Less host CPU involvement

Presentation Outline

- Motivation
- Barrier Algorithms
- Implementation over Myrinet and Quadrics
- Performance Evaluation
- Conclusions and Future Work

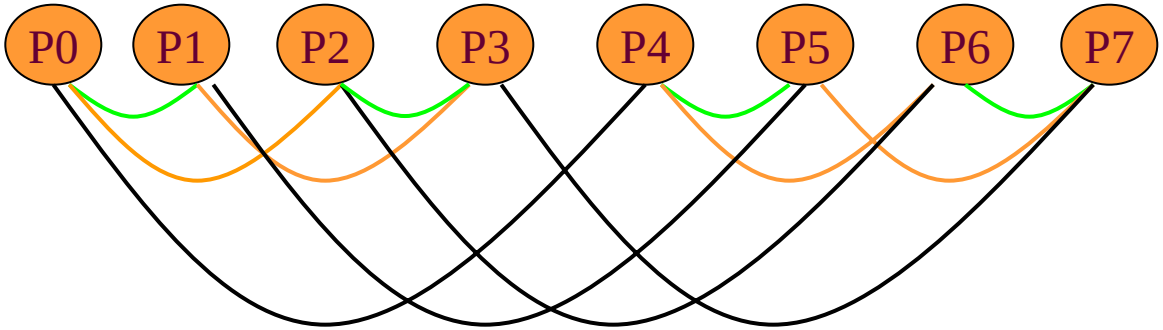
Barrier Algorithm -Gather/Broadcast

- Gather/Broadcast
  - N Processes
    - A tree-based topology to span all the nodes
    - Process 0 as the root
  - Synchronization
    - Process 0 gathers barrier messages through this tree
    - Process 0 broadcasts barrier messages
    - $2 * \lceil \log_2 N \rceil$ * UniDir (one-way message latency)



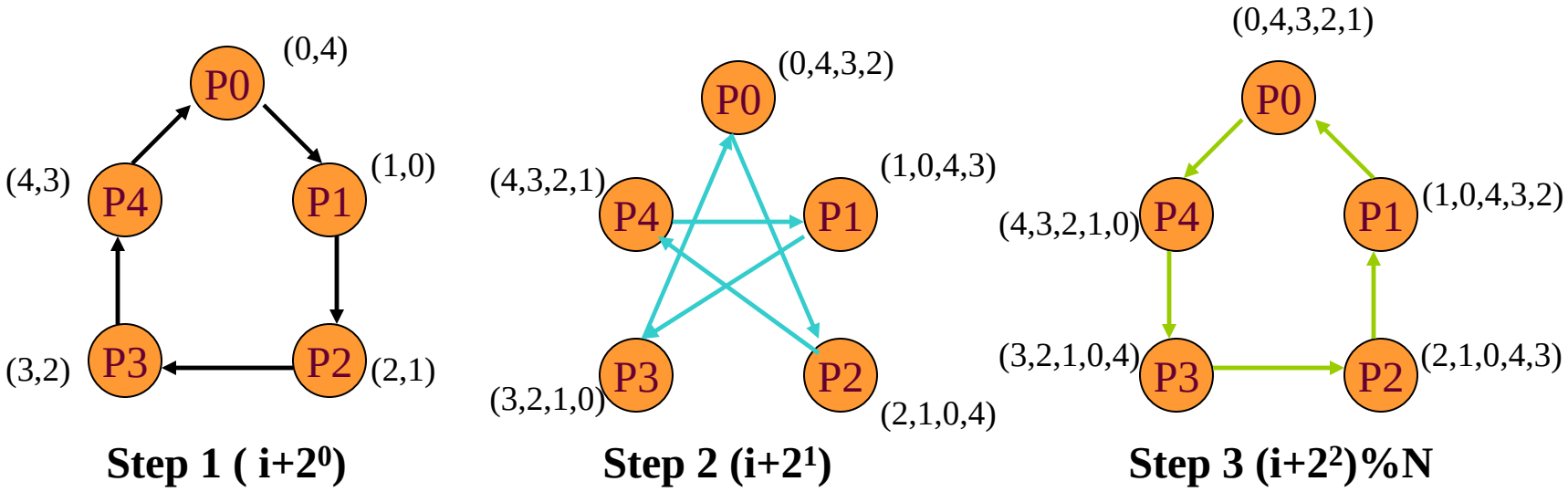
Barrier Algorithm -Pairwise Exchange

- Pairwise Exchange (recursive doubling)
  - Double the barrier info recursively through pairwise message exchange
  - Two extra one-way messages if not power of two
  - $\lfloor \log_2 N \rfloor$ * (BiDir) + 2 (UniDir)



Barrier Algorithm -Dissemination

- Dissemination (a variant of recursive doubling)
  - Double the barrier info recursively through chained dissemination
  - $\lceil \log_2 N \rceil$ * UniDir (send)
    - Other traffic: $\lceil \log_2 N \rceil$ * UniDir (recv)



Step 1 (i+2^0) **Step 2 (i+2^1)** **Step 3 (i+2^2)%N**

Presentation Outline

- Motivation
- Barrier Algorithms
- Implementation over Myrinet and Quadrics
- Performance Evaluation
- Conclusions and Future Work

Implementation of NIC-based Barrier over Myrinet

- Algorithms
  - Dissemination and Pair-wise Exchange
- A New NIC-based barrier over Myrinet
  - A separate queue for collective requests (barrier)
  - Use static packets for barrier messages
    - Immediate sending and fast forwarding
  - Bookkeeping and Error Control
    - A bit vector to record the arrival of barrier messages
    - Send a NACK to the corresponding sender if the barrier operation is timed out

Implementation of NIC-based Barrier over Quadrics

- Algorithms
  - Dissemination and Pairwise-exchange
- Simplified Processing
  - Chained RDMA together to propagate the barrier
    - Using normal RDM
    - Fast chaining of barrier messages
  - Using no NIC threads for barrier
  - Overlap the communication time of one set of barriers with the initialization time of another

Presentation Outline

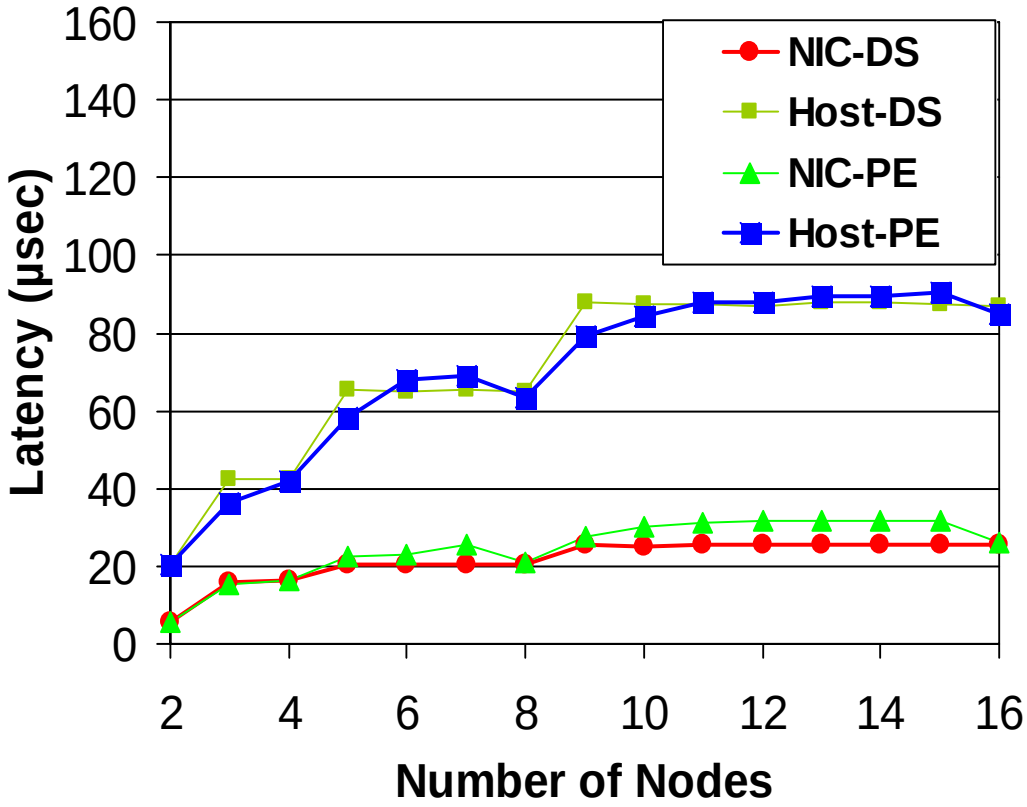
- Motivation
- Barrier Algorithms
- Implementation over Myrinet and Quadrics
- Performance Evaluation
- Conclusions and Future Work

Performance Evaluation

Experiment Testbed:

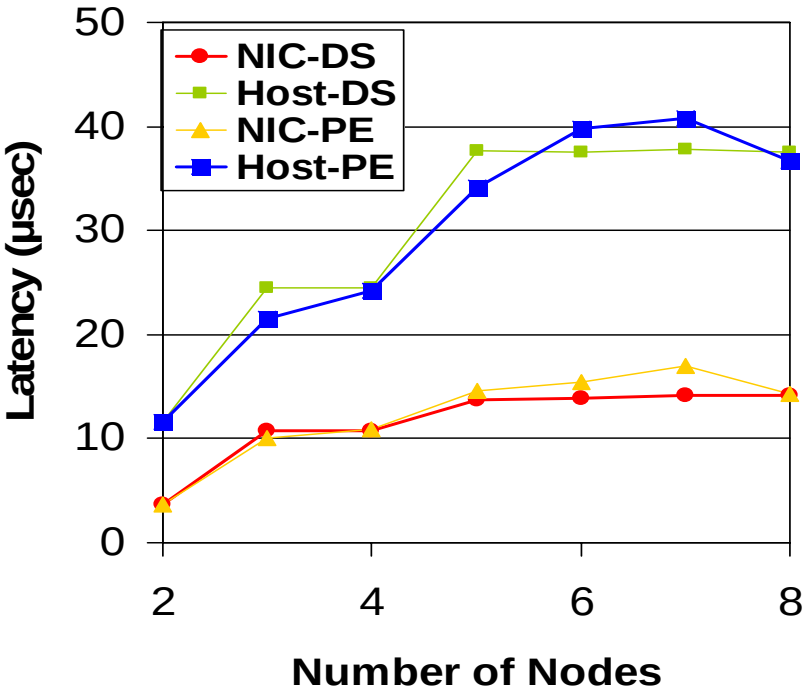
- Myrinet 2000 network
  - A 32 port switch
  - Myrinet cards
    - 133MHz LANai 9.1 processor
    - 225MHz LANai-X processor
- Quadrics Network, Elite-16 and Elan3
- 16-node Quad-SMP 700MHz Pentium III
- 8-node dual-SMP 2.4GHz Xeon

NIC-based Barrier over Myrinet - 16-node 700MHz



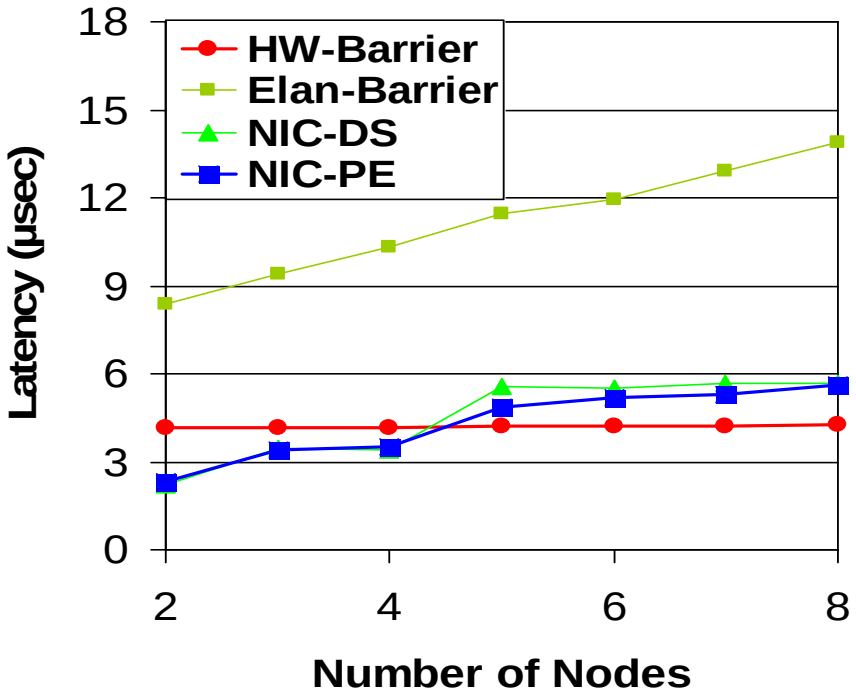
- The new NIC-based barrier improves latency 3.38 times
- Dissemination outperforms Pair-wise Exchange

NIC-based Barrier over Myrinet - 8-node 2.4GHz



- The new NIC-based barrier improves latency by a factor of 2.64 compared to the host-based barrier
- Dissemination outperforms performs Pair-wise Exchange

NIC-based Barrier over Quadrics - 8-node 700MHz



- The new NIC-based barrier improves latency by a factor of 2.48 compared to the host-based barrier
- Pair-wise Exchange outperforms dissemination

Analytical Model -- Myrinet

$$T_{barrier} = T_{init} + (\lceil \log_2 N \rceil - 1) * T_{trig} + T_{fin}$$

- Myrinet:
  - Myrinet 2000 network
  - 225MHz LANai-XP processor
  - dual-SMP 2.4GHz Xeon
- Latency Scaling:
  - $T_{barrier} = 3.6 + (\lceil \log_2 N \rceil - 1) * 3.5 + 3.84$
  - 38.94us barrier latency over 1024 nodes

Analytical Model -- Quadrics

$$T_{barrier} = T_{init} + (\lceil \log_2 N \rceil - 1) * T_{trig} + T_{fin}$$

- Quadrics:
  - Elan3 with QM-400 cards
  - Quad-SMP 700MHz Pentium III
- Latency Scaling:
  - $T_{barrier} = 1.9 + (\lceil \log_2 N \rceil - 1) * 2.12 + 0.4$
  - 21.38us barrier latency over 1024 nodes

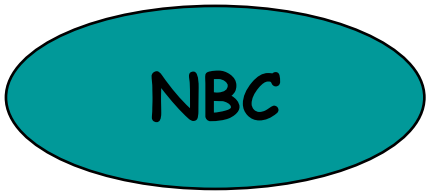
Presentation Outline

- Motivation
- Barrier Algorithms
- Implementation over Myrinet and Quadrics
- Performance Evaluation
- Conclusions and Future Work

Conclusion and Future Work

- A separate NIC-based collective protocol is beneficial
- Efficient Barrier operations are achieved
  - Myrinet:
    - Improve the host-based barrier latency by 3.38 times
    - Can achieve a barrier latency of 38.94us over 1024 nodes
  - Quadrics
    - Outperforms the existing software barrier by 2.48 times
    - Can achieve a barrier latency of 21.38us over 1024 nodes
- Future work:
  - NIC-based operations for other collectives
    - Gather, Allgather, and Alltoall
    - Allreduce, Reduce
  - Application-level performance

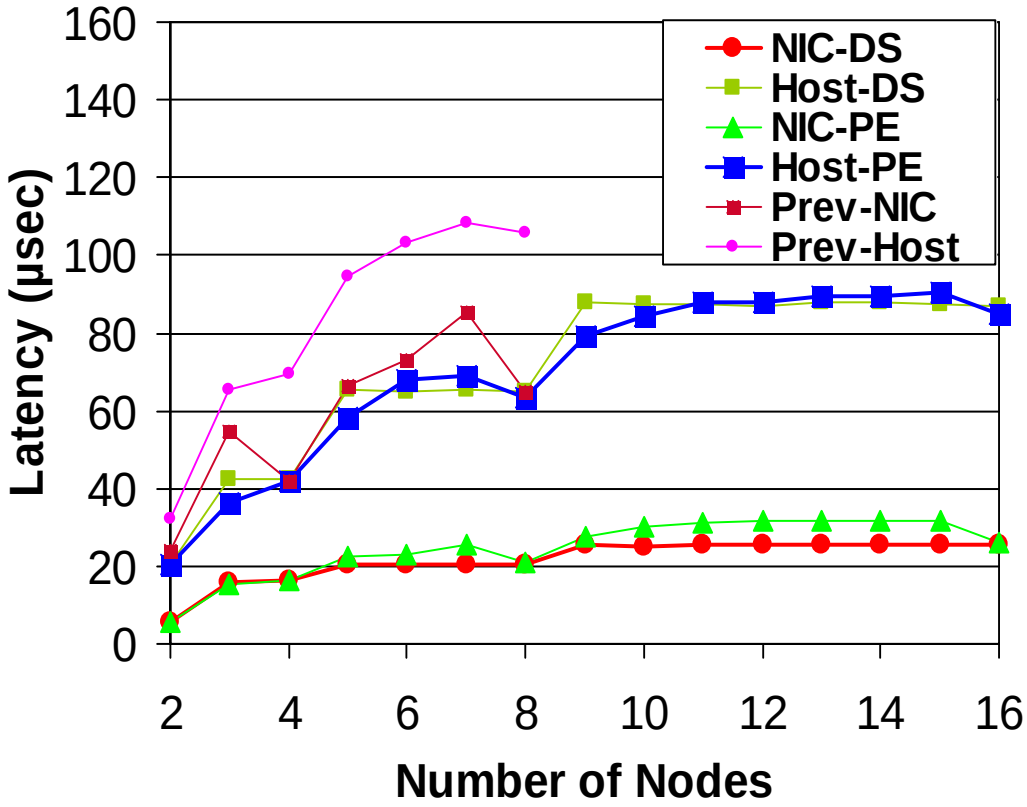
More Information



home page

http://www.cis.ohio-state.edu/~panda/
http://nowlab.cis.ohio-state.edu/

NIC-based Barrier over Myrinet - 16-node 700MHz



- The new NIC-based barrier improves latency 3.38 times
- Dissemination outperforms Pair-wise Exchange