Efficient Implementation of MPI-2 Passive One-Sided Communication on InfiniBand Clusters

W. Jiang, J. Liu, H. Jin, D.K. Panda
D. Buntinas, R. Thakur, W.Gropp

The Ohio State University
Argonne National Laboratory

Outline

- Bac   round
- Pro lem Statement
-   esi n and Implementation
- E perimental   esults
- Conclusion

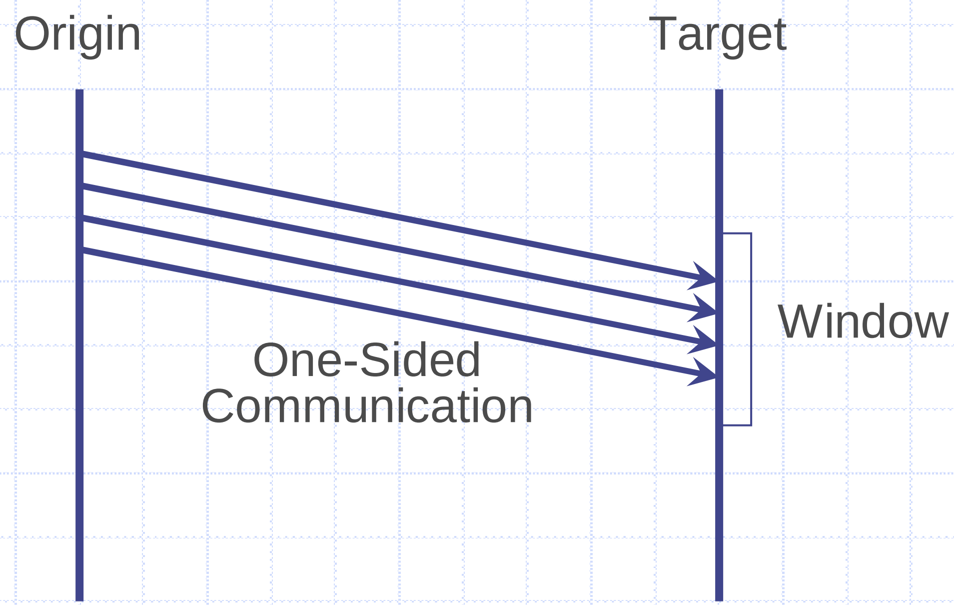
- n i n ag Pa ing n a
- a u
  - On-Sided Communication
  - Dynamic Process Management
  - Parallel I/O support
- P H n a i n P L

One sided Communication

- a ing a i n i n ui d a
- a a a i n a id d ini ia a i n
- u i uni a i n a n
- u d i an in ain a a i a i n

One i e uni ti n

- **Terminology**
  - Origin
  - Target
  - Window
- **Communication**
  - MPI_Put
  - MPI_Get
  - MPI_Accumulate
- **Synchronization**
  - Active target
  - Passive target

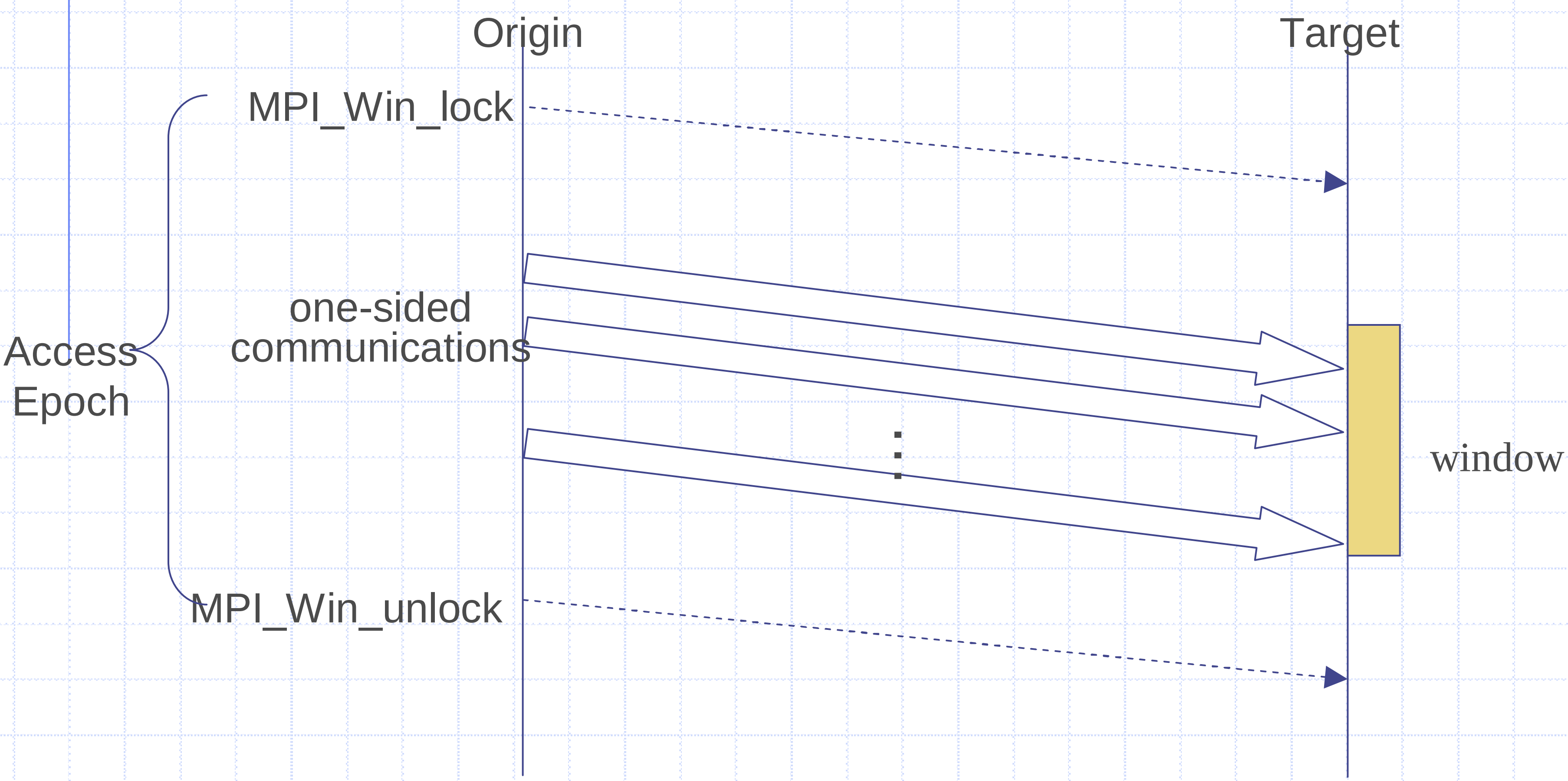


- *MPI-2 one-sided communication decouples synchronization from communication functions*

n ni ti n in One i e uni ti n

- i n ni a i n
  - The target is actively involved in synchronization
  - Previously implemented active mode one sided communication using RDMA based communication on InfiniBand
    - *High Performance MPI-2 One-Sided Communication over InfiniBand. Weihang Jiang, Jiuxing Liu, Hyun-Wook Jin, Dhabaleswar K. Panda and William Gropp and Rajeev Thakur . CCGRID[2004]*
- Pa i n ni a i n
  - The target is not explicitly involved in synchronization
  - Main focus of our work here

One i e uni ti n it i e n ni ti n



n ini n

- n indu anda d
- P id ig an
  - 5 usec latency
  - 10Gbps throughput
- d an d a u
  - Remote Direct Memory Access (RDMA)
    - RDMA write
    - RDMA read
  - Atomic operations
    - Compare and Swap
    - Fetch and Add

Outline

- Bac   round
- Pro lem Statement
-   esi n and Implementation
- Evaluation   esults
- Conclusion

o em tatement

- esi n e i ient assive syn hroni ation y ta in a vanta e o n ini an ’s a van e eat res
- ore i erent a roa hes an strate ies
- va ate these a roa hes

Outline

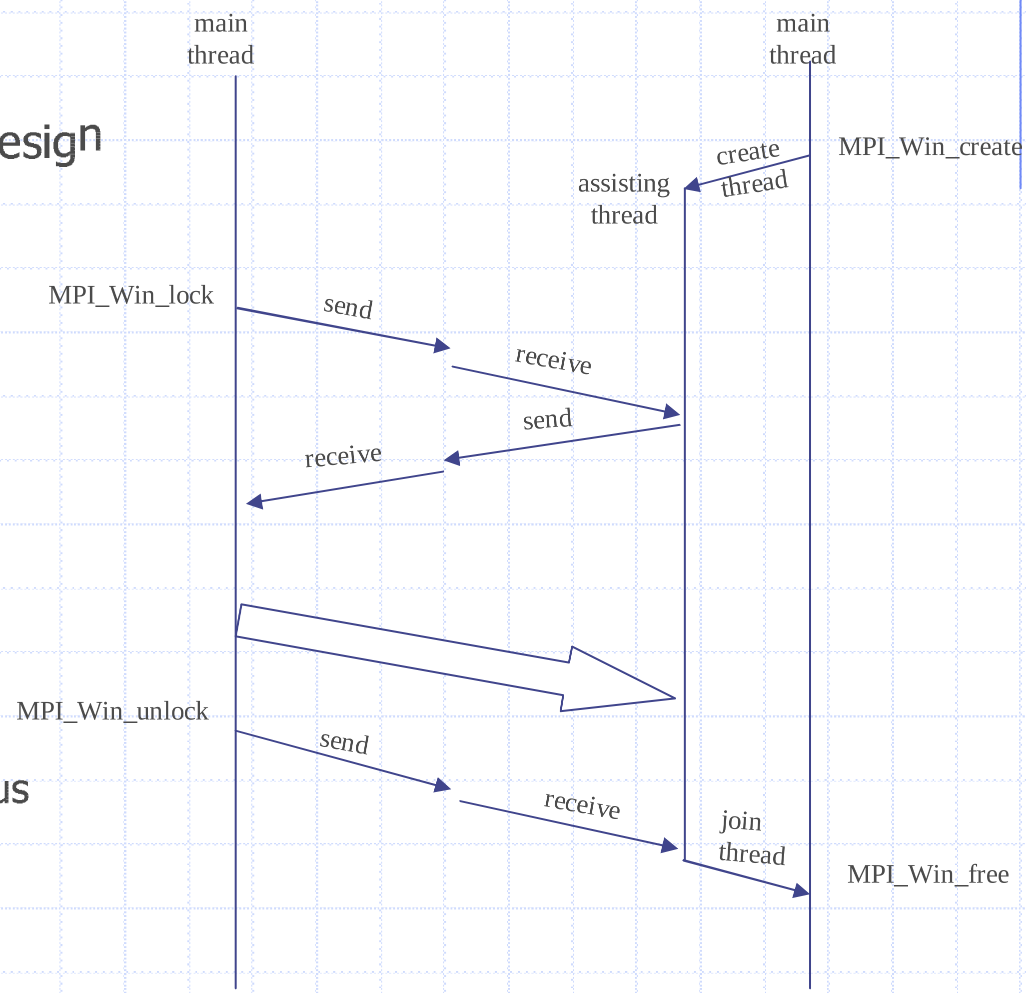
- Bac   round
- Pro lem statement
-   esi n and Implementation
- E perimental   esults
- Conclusion

esi n

- esi n C allen es
  - o Overhea or syn hroni ation
  - n e en ent ro ress
  - on rrent o ni ation
- o road approac es
  - Threa ase esi n
  - to i O eration ase esi n

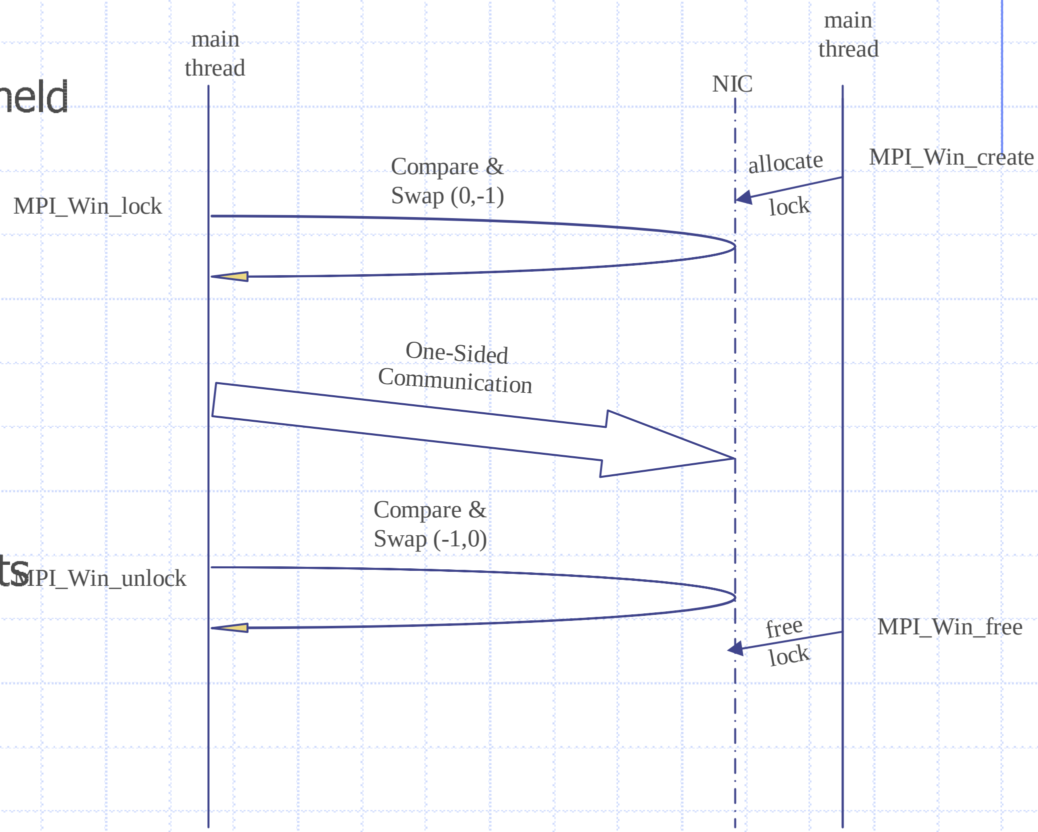
e e e i n

- Variations of thread based design
  - Dedicated thread
  - Multiple thread
  - Event Driven thread
- Pros:
  - Simple and portable
- Cons:
  - CPU cycle consuming
  - Can not handle simultaneous requests well



t i O e ti n e e i n

- "Test & Set" based design
  - Flag used to indicate if a lock is held
  - Taking advantage of Infiniband atomic operations
- Pros:
  - Simple
  - Can handle simultaneous requests well (with hardware support)
- Cons:
  - Large number of messages



tomic Operation Based esi n 2

J.M. Mellor-Crummey and Michael L. Scott

- Proposed for shared-memory environment
- Implement MCS by taking advantage of Infiniband atomic operations

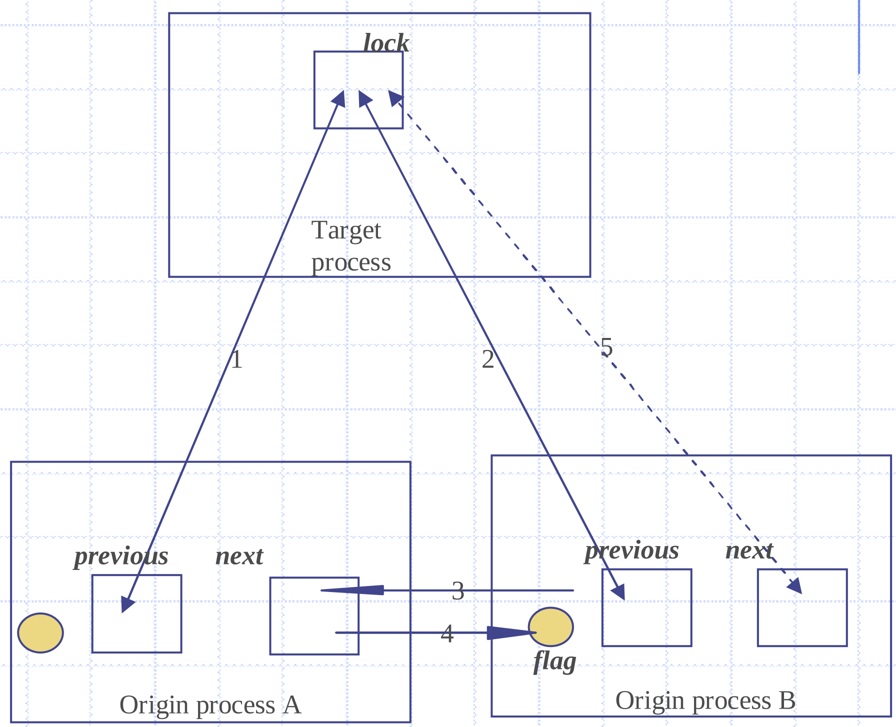
Pros : Scalable to large number of clients

Cons : Needs atomic swap operation

t i O e ti n e e i n

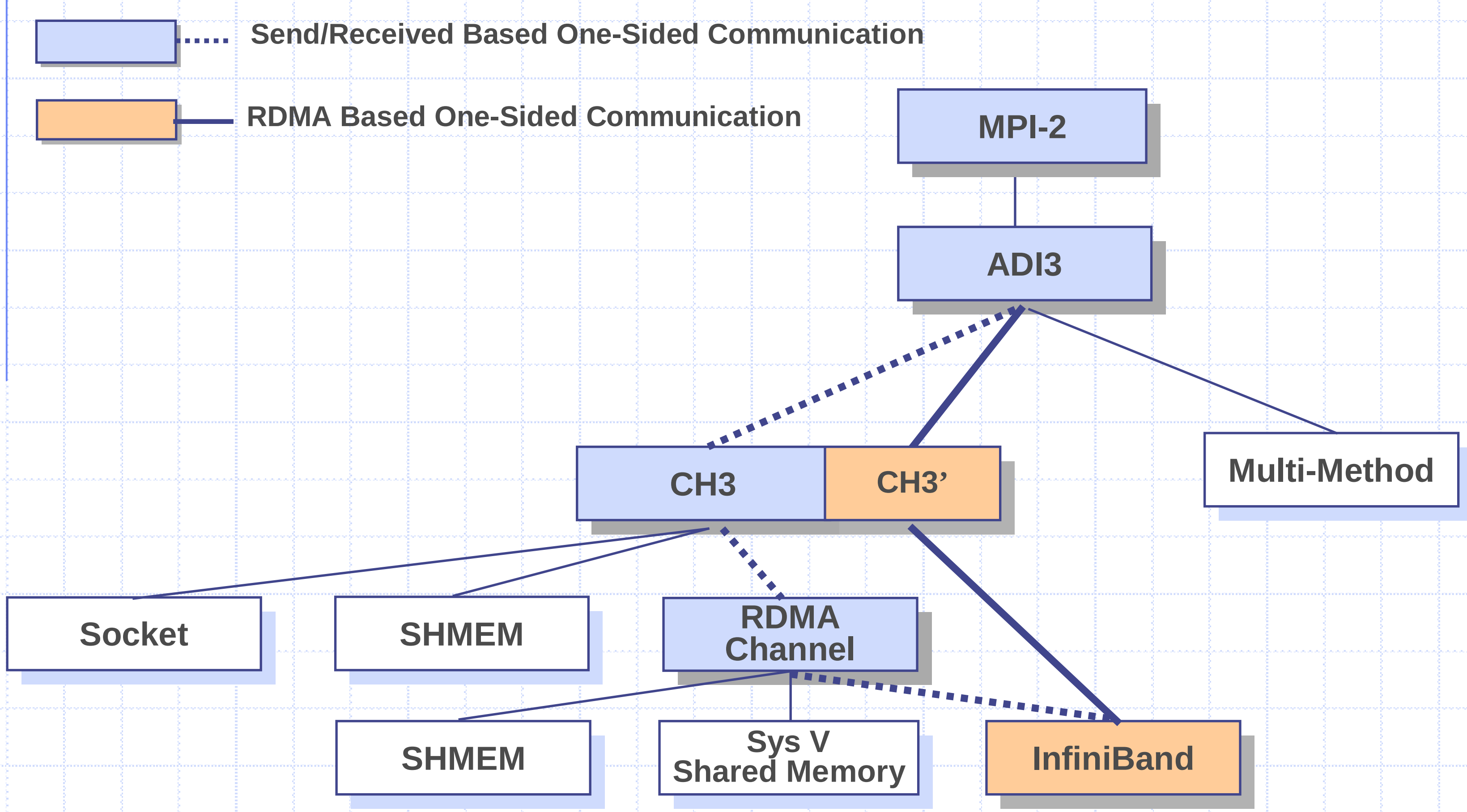
MCS

- Maintains a distributed queue
- Origin process maintains flag, previous and next
- Target process maintains lock
- Avoid spinning on the remote memory
- A scalable synchronization algorithm
  - Needs SWAP for efficient implementation



e

One sided design



Outline

- Bac round
- Motivation
- esi n and Implementation
- E perimental esults
- Conclusion

Experimental Test bed

- u i n d
  - Dual Intel Xeon 2.40 GHz processors
  - PCI-X 64-bit 133MHz interfaces
  - 512K L2 cache and a 400 MHz front side bus
- an n iniH Dua P H
- n ini a ig P n ini and i
- Linu d Ha . i . . n , .

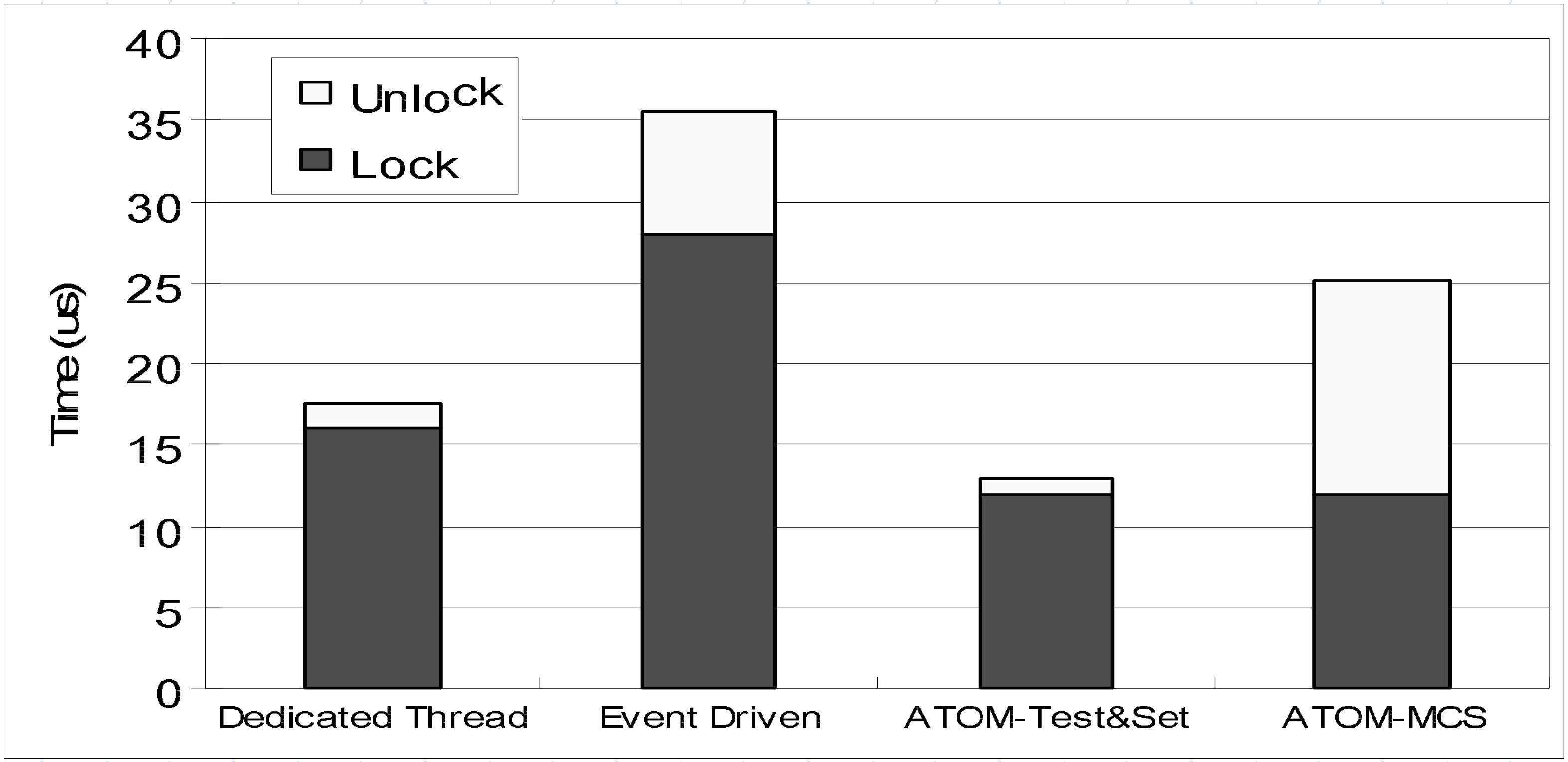
Experiments nd ted

- Synchronization Overhead
- Synchronization e ay
- e a e o e ity
- onc rrency
- Overhead in re ence o o tin hread
- S ode

Synchronization Overhead

- One process calls only MPI-2 passive synchronization functions (MPI_Win_lock and MPI_Win_unlock) on a window on the other process for multiple iterations
- We then report the time taken for each iteration

Synchronization Overhead

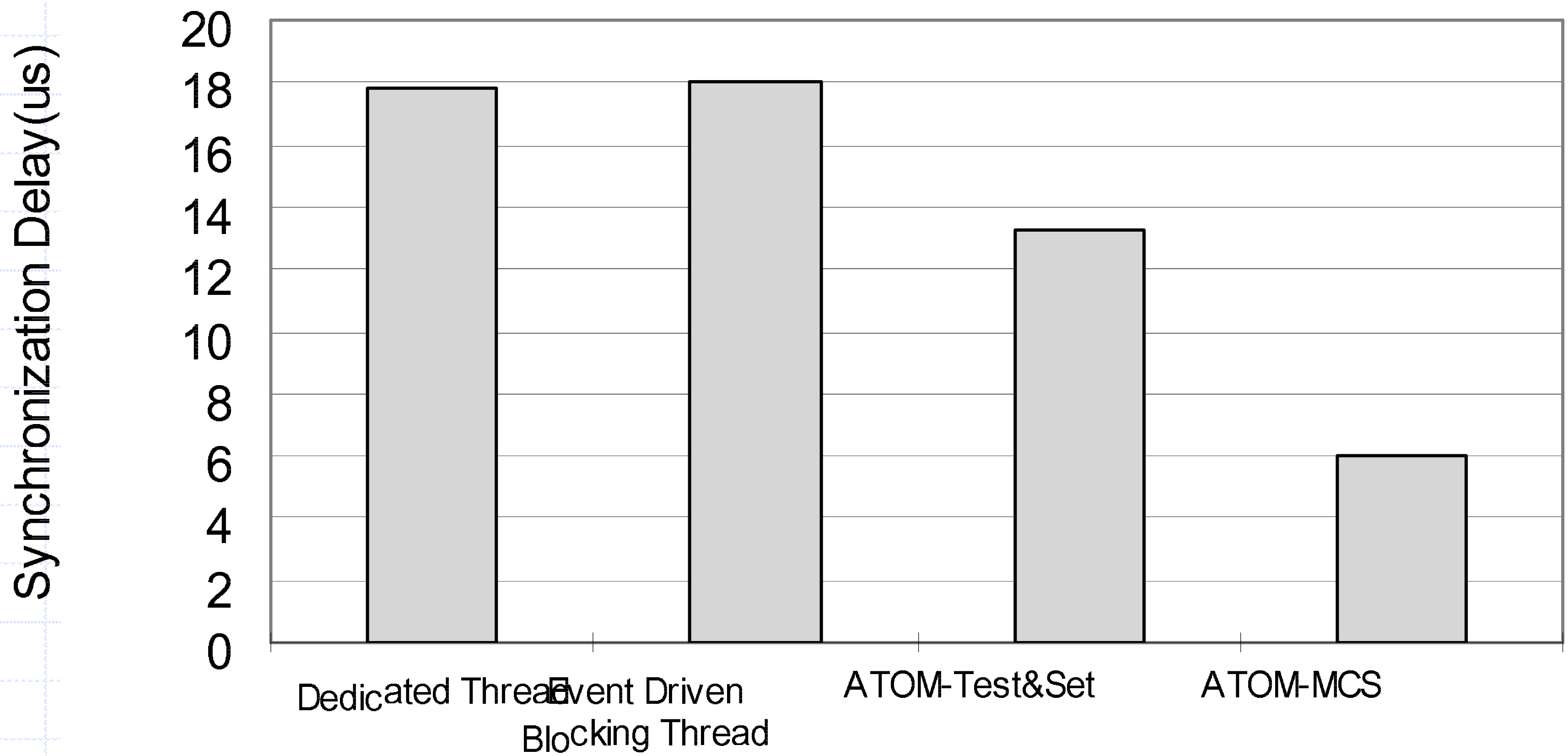


- Event driven brings heavy overhead
- "Test&Set" based design has best performance (12.8usecs)

n r ni ati n ela

- o ori in roce e and one tar et roce and the t o ori in ee co etin or the a e oc or ti e iteration
- o ori in roce e ho d the oc a ternative y
- ecord the de ay to chan e oc o ner hi

n r ni ati n ela

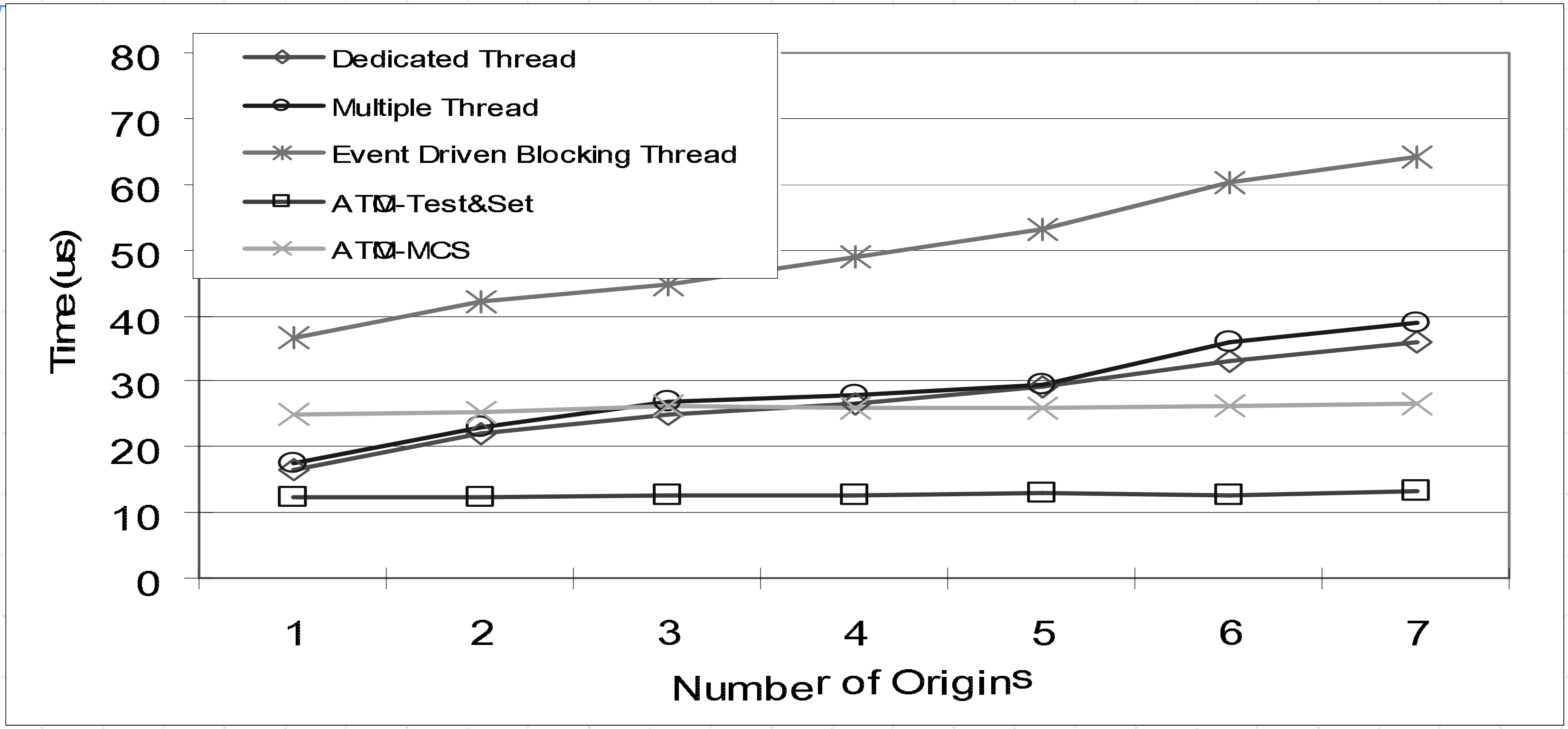


- ATOMIC-MCS shows the best synchronization delay
- The lock can be transferred to next process with a single message

n rren Test

- In the tar et process multiple windows are created and the num er of windows is e ual to the num er of ori in processes
- In each iteration each ori in process calls only MPI_Win_lock and MPI_Win_unlock on the correspondin tar et window
- We then report the avera e time spent on each iteration

n rren

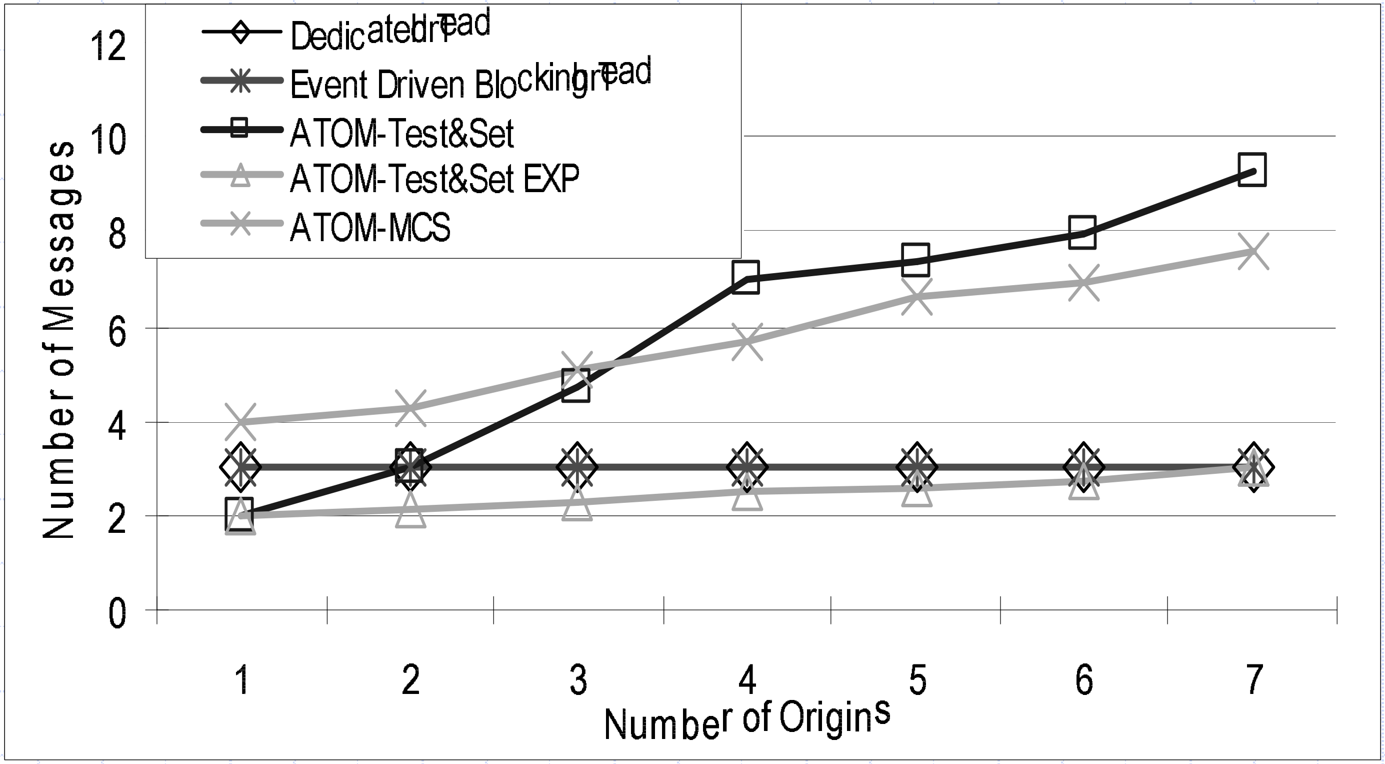


- The hardware support for atomic based design helps in handling concurrency with almost no increase in response time
- The response time of thread based designs increase with number of origins

essa e mplexit

- he test uses multiple ori in processes and one tar et process and all the ori in processes compete for the same lock on a tar et window
- We calculate the avera e num er of messa es e chan ed etween one ori in process and one tar et process

essa e mplexit

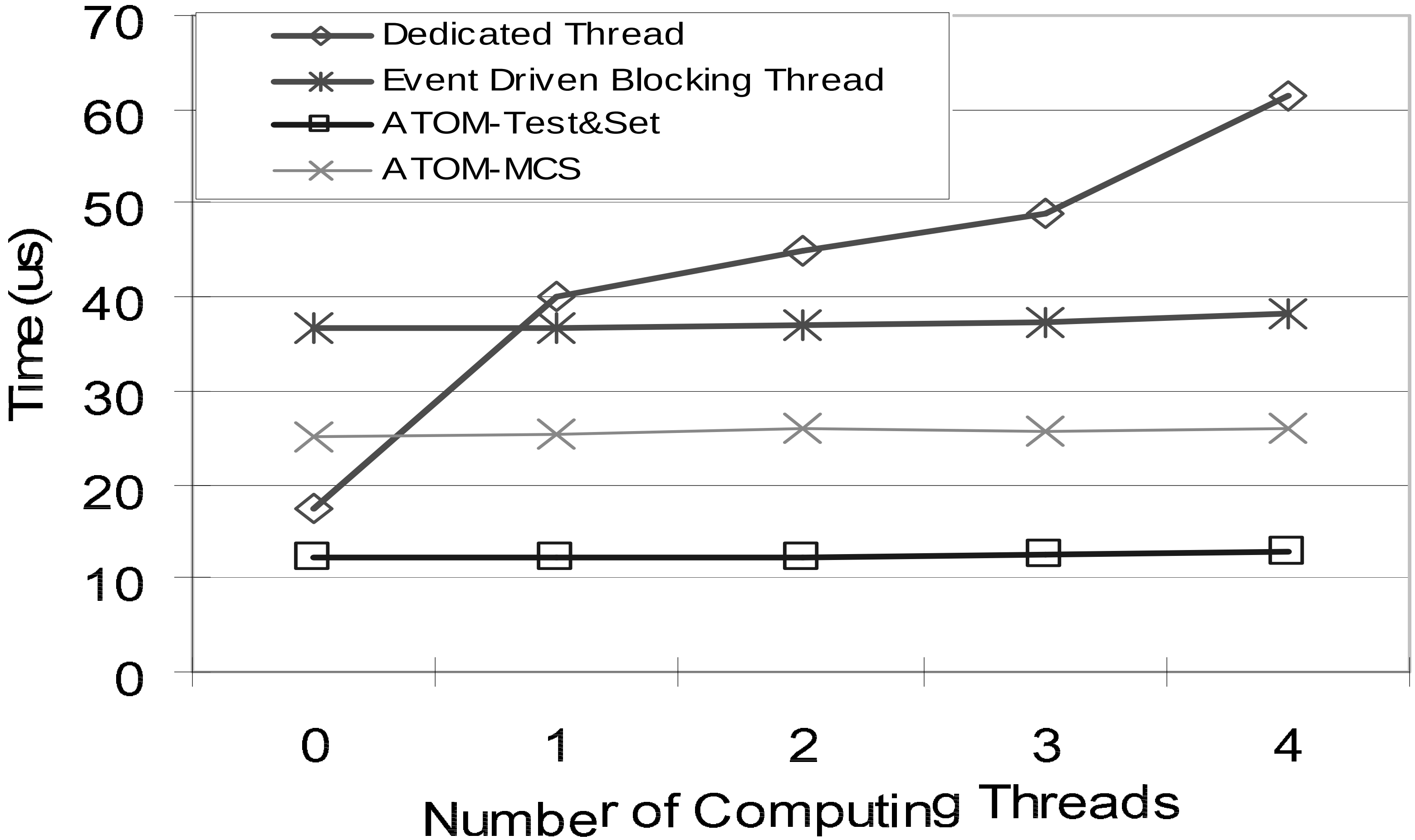


- Without atomic swap, MCS can not achieve its benefits
- With exponential back-off, the “Test&Set” design has low message complexity

mp tin T read

- e have one tar et roce and one ori in roce
- he tar et roce a n ti e co tin thread
- he ori in roce ca ynchronization nction e ea re the overhead in the re ence o the e co tin thread

utin e

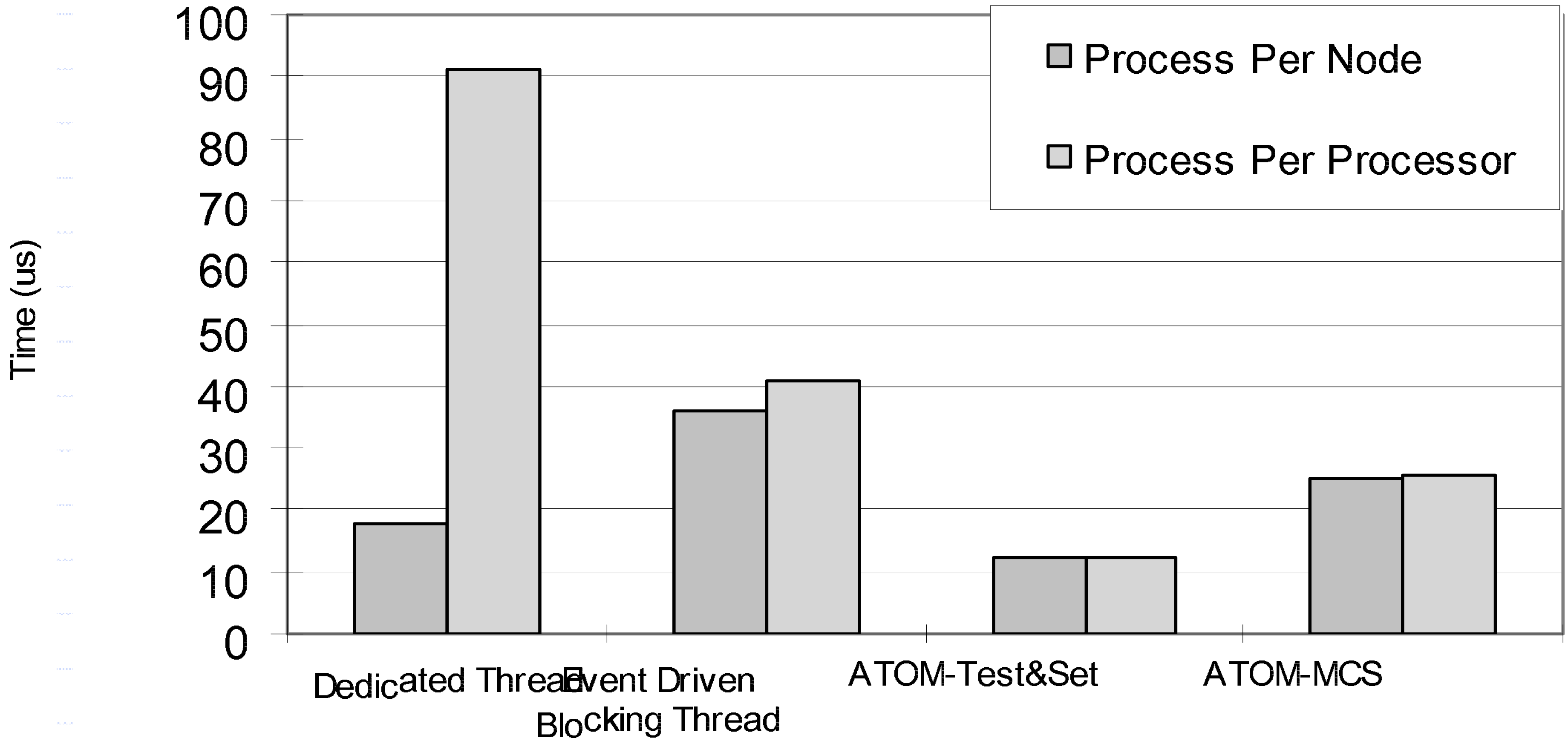


- Atomic based designs performance is not affected by computing threads
- The performance degrades for dedicated thread based design

s de

- o ori in processes and t o tar et processes
- Ori in processes call s nc roni ation functions on tar et processes
- P t o ori in tar et processes in t e different nodes
- SMP t o ori in tar et processes in t e same dual-CP nodes

v S ode



- The atomic based designs outperform the thread based design
- More suitable for SMP mode runs

Outline

- Bac   round
- Motivation
- esi n and Implementation
- E perimental   esults
- Conclusion

n l si n

- i . a d D ign an a i a i n ni a i n
  - Achieves lower overhead (~12.8 us)
- Reasonable Message Complexity (less than3)
- Less CPU utilization

So t are i tri tion

- P H
  High performance MPI-2 Implementation over InfiniBand based on MPICH2
- u n i n P H . a
- d an gani a i n ing P n ini and
- W i a P H . . in a
  - This release will support active one-sided communication
  - Passive one-sided communication will be incorporated in the next release
- n a . i . i a . du i i a ind .

t re r

- Da a u n id d uni a i n
  - Packing/Unpacking
  - InfiniBand specific datatype solutions
    - Multi RDMA
    - Gather/Scatter
    - Hybrid
  - i a i n a ua i n

Thank You!

For more information, please visit the

NBC **Home Page**

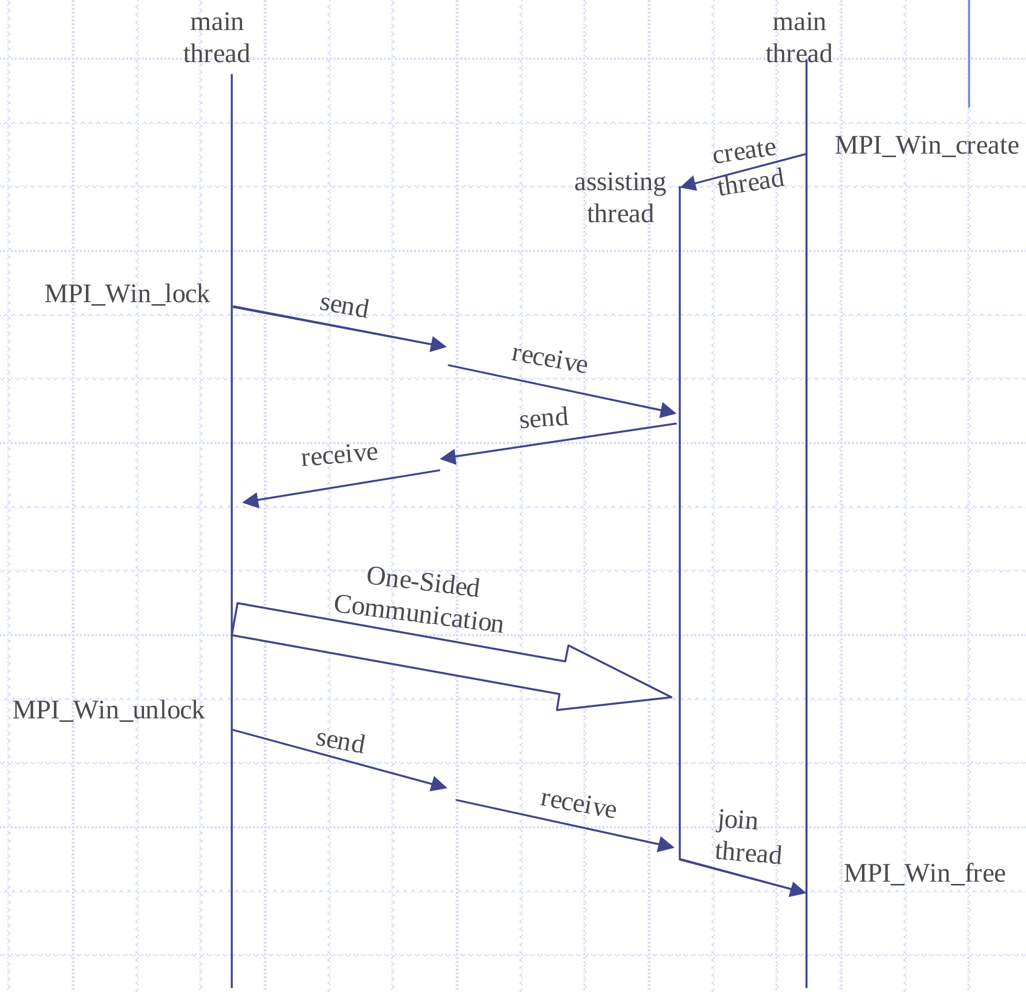
http://nowlab.cis.ohio-state.edu

Network Based Computing Laboratory

The Ohio State University

re i s r

- i h er or ance One Sided o nication over n ini and eihan ian i in i y n oo in ha a e ar anda and i ia ro and a eev ha r rid
- M - ased implementation can ac ieve
  - etter er or ance or one ided co nication and active ynchronization


main
thread
main
thread
MPI_Win_create
create
thread
assisting
thread
MPI_Win_lock
send
receive
send
receive
One-Sided
Communication
MPI_Win_unlock
send
receive
join
thread
MPI_Win_free