NETWORK-BASED COMPUTING LABORATORY

Design and Implementation of Key Proposed MPI-3 One-Sided Communication Semantics on InfiniBand

Sreeram Potluri, Sayantan Sur, Devendar Bureddy and Dhabaleswar K. Panda

Network-Based Computing Laboratory
Department of Computer Science and Engineering
The Ohio State University, USA

Introduction

- Reduced synchronization overheads, simultaneous use of powerful system resources - key on modern clusters
- Better support through one-sided communication in MPI-2
- Optimized implementation in MVAPICH2
- Limitations in semantics – hindered its wider acceptance
- RMA working group proposed several extensions as part of the MPI-3 effort
- Efficient implementation is crucial – to highlight their performance benefits, encourage their wide-spread use
- Can the new semantics be implemented with high performance in MVAPICH2?

Overview

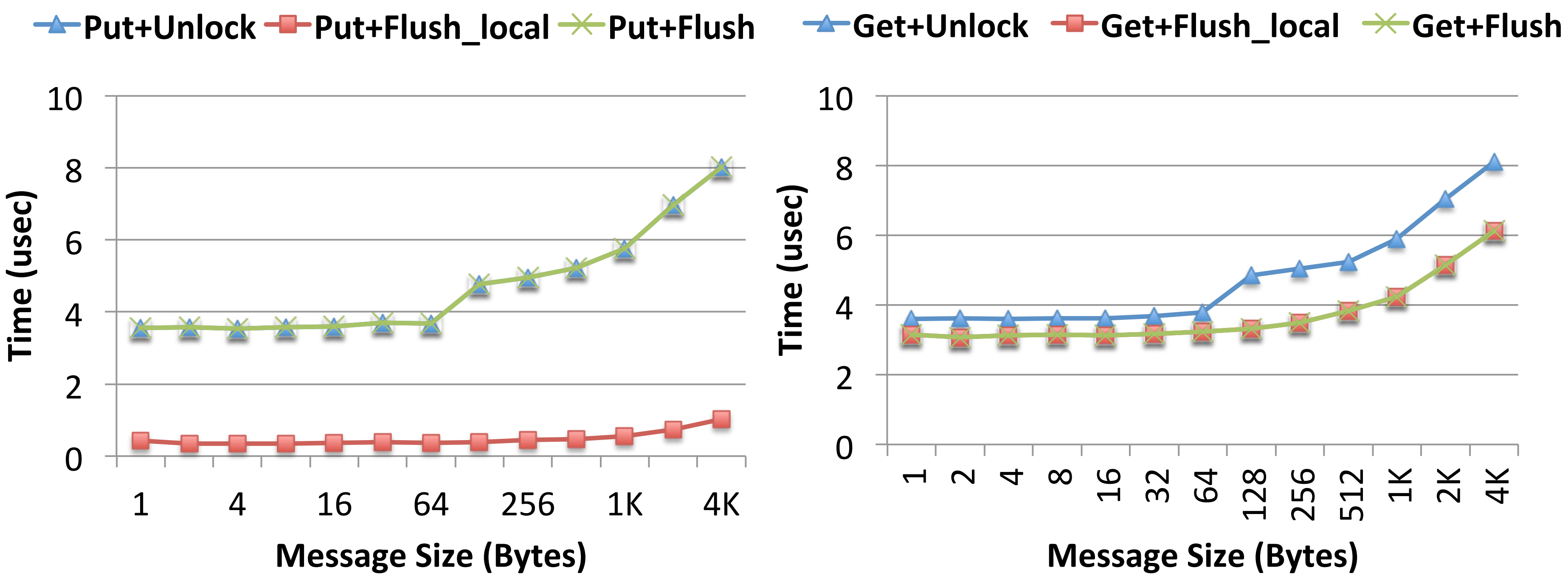
MPI-3 One Sided Communication

Synchronization	Communication	Window Creation
• Lock_all, Unlock_all • Win_flush, Win_flush_local, Win_flush_all, Win_flush_local_all • Win_sync	• Get_accumulate • Rput, Rget, Raccumulate, Rget_accumulate • Fetch_and_op, Compare_and_swap	• Win_allocate • Win_create_dynamic, Win_attach, Win_detach
Separate and Unified Windows	Accumulate Ordering	Undefined Conflicting Accesses

Flush Operations

- Local and remote completions bundled in MPI-2 one-sided communication model
- Handled using synchronization operations, requires closure of an epoch
- Overhead in scenarios that require only local completions
- Considerable overhead on networks like IB - semantics and cost of local and remote completions are different
  - RDMA Reads and Atomic Ops: CQ event means both local and remote completions
  - RDMA Writes: CQ event only means local completion. Remote completion requires a follow up Send/Recv exchange or an atomic operation.
- Flush operations allow for more efficient check for completions

Flush Operations



- Local completion of Put is efficient using flush
- Completion does not require closure of the epoch

8-core Intel Westmere Nodes connected with InfiniBand QDR IB

Request Based Operations

- Current semantics provide bulk synchronization
- Lack of a way to request completion of individual operations, without closing an epoch
- Does not serve well for fine grained computation and communication overlap
- Request based operations (MPI_Rput, MPI_Rget, and others) return an MPI Request, can be polled for completion
- Added GCP(Get-Compute-Put) Benchmarks in the OSU suite to highlight their benefits

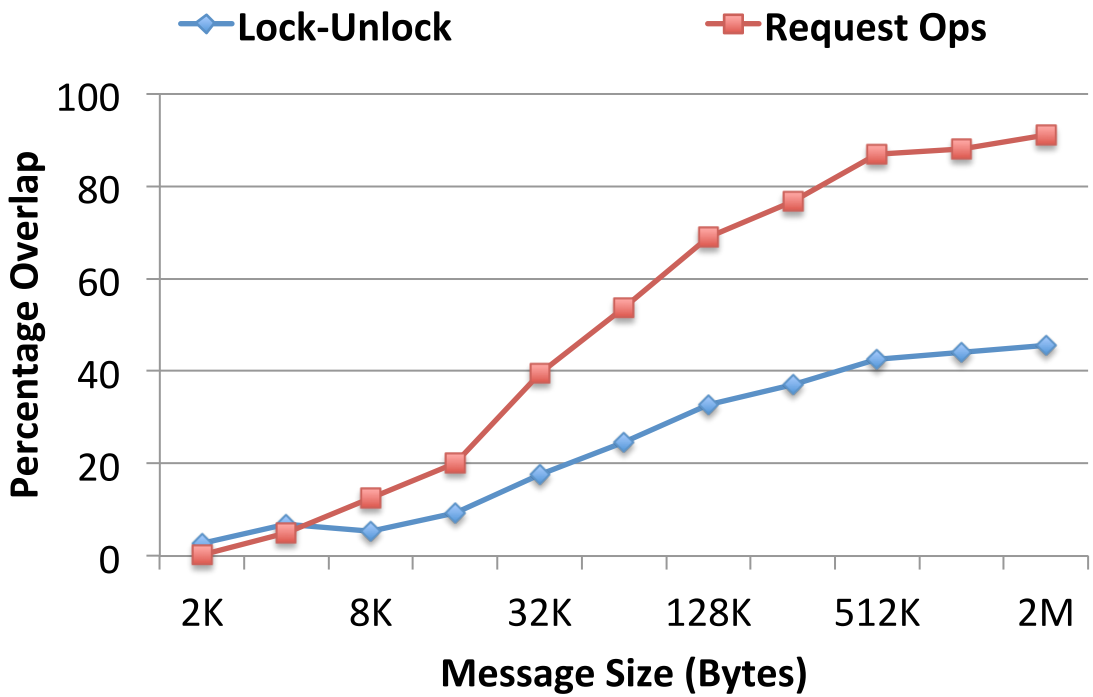
Request Based Operations

GCP Benchmark

No Overlap	Overlap using Lock-Unlock	Overlap using Request Ops

MPI_Win_lock
for i in 1, N
    MPI_Get (ith Block)
end for
MPI_Win_unlock
 	
Compute (N Blocks)
 	
MPI_Win_lock
for i in 1, N
    MPI_Put (ith Block)
end for
MPI_Win_unlock
	MPI_Win_lock
	for i in 1, N
	    MPI_Get (ith Block)
	end for
	MPI_Win_unlock
	
	MPI_Win_lock
	for i in 1, N
	    Compute (ith Block)
	    MPI_Put (ith Block)
	end for
	MPI_Win_unlock
		MPI_Win_lock
		for i in 1, N
		    MPI_Rget (ith Block)
		end for
		
		MPI_Wait_any (get requests)
		while a get request j completes
		    Compute (jth Block)
		    MPI_Rput (jth Block)
		    MPI_Wait_any (get requests)
		end while
		MPI_Wait_all (put requests)
		MPI_Win_unlock

Request Based Operations



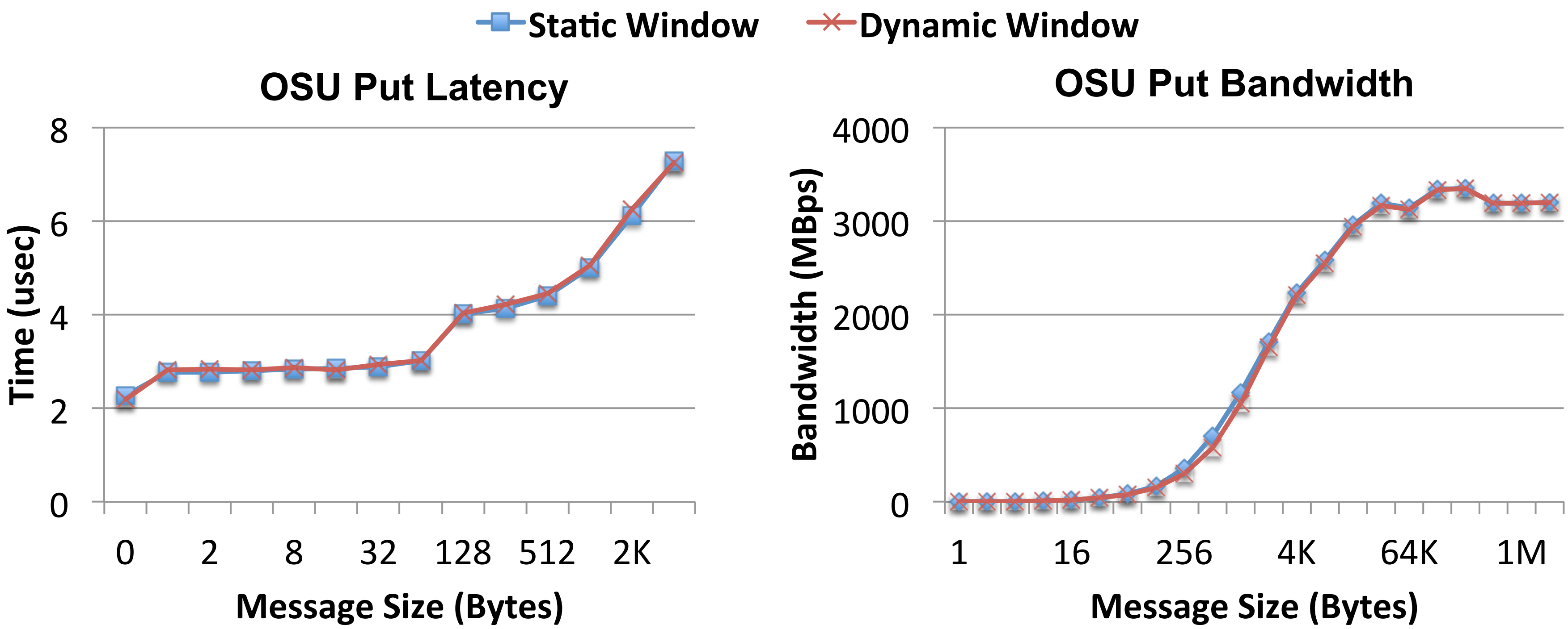
- Request based operations provide superior overlap

8-core Intel Westmere Nodes connected with InfiniBand QDR IB

Dynamic Windows

- Creation of a window is collective on communicator
- A process can attach or detach memory to the window dynamically
- User has to manage exchange and correct use of address information
- MPI Implementations on IB have to manage dynamic exchange of key information to use RDMA
- MVAPICH2 uses a pull model – request-for-info sent when the first operation is issued on a region, information is cached
- Request is piggy-backed onto the first data packet for small and medium message sizes

Dynamic Windows



- Dynamic windows can provide performance similar to static windows
- Key exchange overhead is amortized

8-core Intel Westmere Nodes connected with InfiniBand QDR IB

Conclusion and Future Work

- First implementation of features from the proposed one-sided communication semantics for MPI-3
- Highlighted their benefits
- Working towards a complete implementation of the proposed MPI-3 one-sided communication standard
- Modifying application benchmarks to show how real-world applications can benefit from the proposed extensions

Thank You!

{potluri, surs, bureddy, panda}@cse.ohio-state.edu

Network-Based Computing Laboratory

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

MVAPICH Web Page

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