

Programming Models and Environments on Systems with Modern Networks

788 Winter '08

-Tejus Gangadharappa

Papers

V. Tipparaju, G. Santhanaraman, J. Nieplocha, and D. K. Panda. Host-Assisted Zero-Copy Remote Memory Access Communication on InfiniBand. (IPDPS 04)

C. Coarfa, Y. Dotsenko, J. Mellor-Crummey, F. Cantonnet, T. El-Ghazawi, A. Mohanty, Y. Yao, An Evaluation of Global Address Space Languages: Co-Array Fortran and Unified Parallel C (PPOPP 2005)

C. Bell, et. al., Optimizing Bandwidth Limited Problems Using One-Sided Communication and Overlap (IPDPS '06)

Abstract

This paper describes how one-sided RMA communication model can be implemented efficiently using the Infiniband verbs layer.

ARMCI one-sided RMA operations are implemented on top of Infiniband.

Provides a host-assisted zero-copy method to perform one-sided RMA. (provides RMA for contiguous and non-contiguous data)

RMA model

An intermediate programming model between message-passing & shared memory

Benefits applications characterized by irregular data structures and dynamic access patterns.

Little sender/receiver co-ordination

Excellent potential for overlap of communication & computation

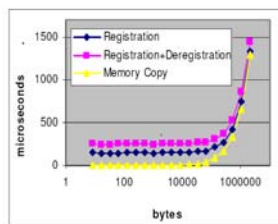
Preferred communication model in global address space languages

RMA over Infiniband

. A mismatch between user level RMA interfaces & Infiniband requirements w.r.t virtual memory

. RDMA write/read requires page locking (also called memory registration); a costly operation

. Memory registration vs Copying



Managing registered memory

Three techniques

. on-demand dynamic memory registration & deregistration

. copying data via pre-allocated registered buffers (termed host-assisted technique)
- improvement: - chunked data with pipelined copy/send overlaps

. providing user with a pre-registered memory allocation interface

Hybrid technique

Use the allocator interface provided, which tries to allocate pre-registered memory.

Maintain a table of registered chunks and VAPI memory key information.

If memory was allocated but could not be registered don't add it to the table.

Use the table to check if the memory used is registered (and if so, perform zero-copy RDMA read/write)

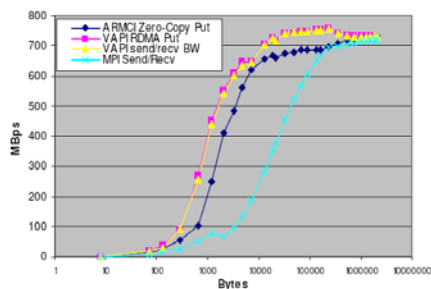
If unregistered, use copy based or dynamic registration approach

Evaluations

- . Dual processor 1 GHz Itanium2 (cluster 1)
- . 32 node dual processor P4 (cluster 2)
- . Mellanox A1 cards
- . 730 MBps Put 689 MBps Get (cluster1)
- . 830 Mbps Put 765 Mbps Get (cluster2)

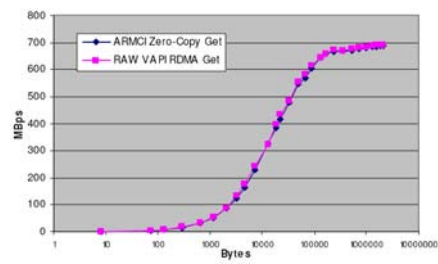
Performance of Put/Get

ARMCI Put bandwidth is lower than VAPI RDMA Put due to a difference in testing mechanism



Performance of Put/Get

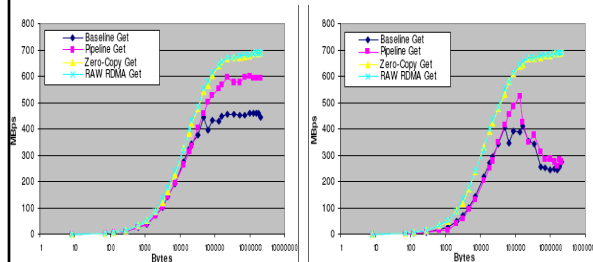
ARMCI Get performs very close to VAPI RDMA Get operation. (very little overhead)



Performance of Put/Get (protocols)

Comparison of zero-copy & buffered/pipelined implementations

Pipelined approach uses remote side CPU. Second chart shows effect on bandwidth with busy remote CPU



Host-assisted Zero-copy RMA

. A helper thread on the host assists RMA operations by initiating all operations and requires minimum CPU involvement

. Get -

- on networks without/unoptimized RDMA read client sends a request to helper thread. Helper thread issues an RDMA put

. Non-contiguous - Use SG VAPI ops (gather-send/scatter-recv)

. Put

- post request, thread posts a scatter-recv and acks back.
- source now posts its gather-send (achieving zero-copy)

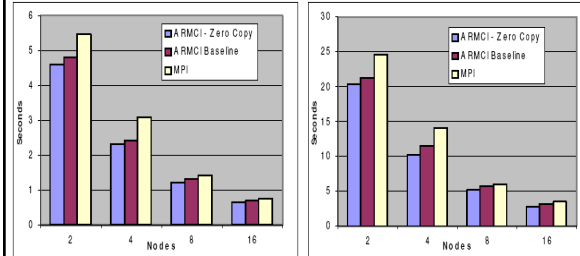
. Get

- post scatter-recv and send request to helper thread.
- thread posts gather-send

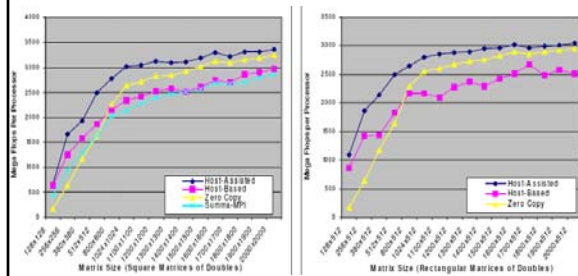
NAS MG/Matrix Mult

- NAS MG
modified to use ARMCI one-sided RMA ops
- SUMMA
host-based/buffered outperforms MPI by 44%
host-assisted zero copy outperforms by 18% to 80%.

NAS MG results



SUMMA results



Co-Array Fortran

- . Extension of F95 with constructs for SPMD parallel programming
- . `a(n,m)[*]` creates a shared co-array with 'nxm' local to each process
- . provides partitioned global address space
- . weakness - memory fences before and after procedure calls

Unified Parallel C

- . C extension for parallel programming (like OpenMP)
- . threads share a part of the address space
- . shared address space is partitioned and each fragment has an affinity to a thread
- . Supports dynamic memory allocations
- . supports non-blocking barriers (to overlap computation & synchronization)

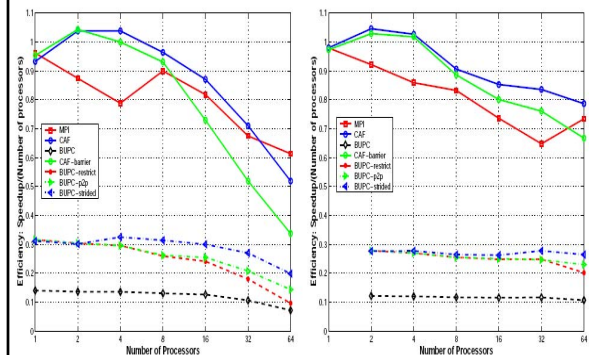
Evaluation

- . NAS MG multigrid kernel (discrete Poisson problem)
- . NAS CG (smallest eigenvalue of a sparse matrix using a conjugate gradient method)
- . NAS SP/BT (CFD applications)
- . 4 clusters
 - 92 HP Itanium2 with Myrinet
 - Alpha cluster (4 CPUs per SMP) with quadrics
 - 2 SGI numa platforms

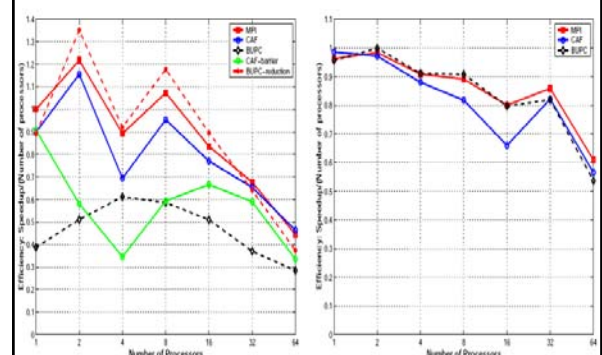
Evaluated models

CAF	Co-array Fortran
CAF-barrier	CAF with explicit barriers for synchronization
BUPC	Berkeley UPC with barrier synchronization
BUPC-restrict	local pointers are declared as 'restrict' (barrier synchronization)
BUPC-p2p	uses p2p messages for synchronization
BUPC-strided	uses strided data for bulk transfers (p2p sync)

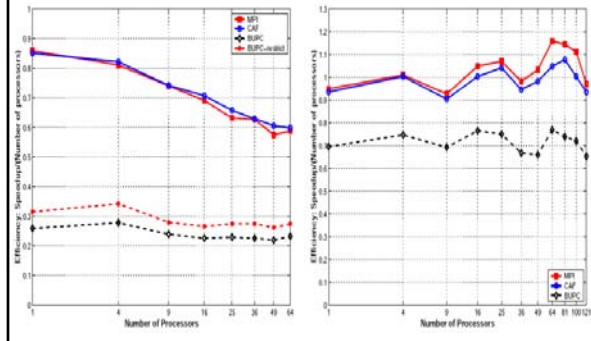
NAS MG (class A, class C, myrinet)



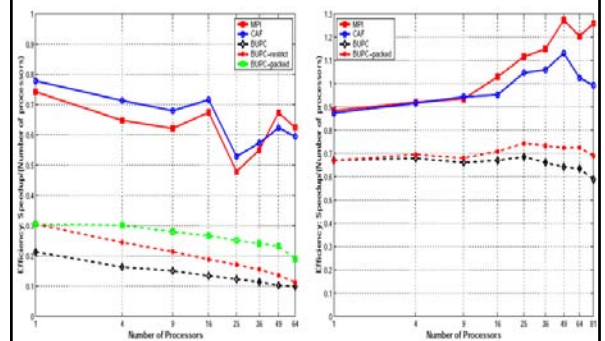
NAS CG (class C, class B, myrinet, quadrics)



NAS SP (class C, Itanium-Myrinet, Alpha-Quadrics)



NAS BT (class C, Itanium-Myrinet, class B, Alpha-Quadrics)



Conclusion

- . Both UPC and CAF can yield scalable performance
- . Barriers hurt performance (can be solved by using explicit point-to-point primitives)

Papers

V. Tipparaju, G. Santhanaraman, J. Nieplocha, and D. K. Panda.
Host-Assisted Zero-Copy Remote Memory Access
Communication on InfiniBand. (IPDPS '04)

C. Coarfa, Y. Dotsenko, J. Mellor-Crummey, F. Cantonnet, T. El-Ghazawi, A. Mohanty, Y. Yao, An Evaluation of Global Address Space Languages: Co-Array Fortran and Unified Parallel C (PPOPP 2005)

C. Bell, et. al., Optimizing Bandwidth Limited Problems Using One-Sided Communication and Overlap (IPDPS '06)

One sided communication

- . One sided communication is primary mode in in PGAS
- . Benefits of one-sided model is high for small messages where synchronization overhead is not amortized by transfer time
- . But one-sided model is beneficial to bandwidth-bound applications
- . Bulk messages have phases of communication & computation but one-sided support in new hardware allow de-coupling of synchronization & data transfer

Why one-sided ? Reason: Overlap

- . If computation can be found to overlap the communication latency then the communication latency will only be the cost of initiate & synchronize non-blocking communication
- . Evaluate one-sided method on the NAS FT problem that performs a 3D FFT

PGAS & GASnet

- One-sided primarily used in PGAS languages
- GASnet or Global-address space networking provides a language-independent, low-level networking layer that provides HPC primitives tailored for PGAS.
- GASnet provides decoupled one-sided point-to-point primitives
- GASnet interface has been natively implemented on Myrinet, Infiniband etc

MPI vs One-sided

- Sends & Receives have to be matched to complete transfer. Overhead of this matching is significant for small messages
- Active participation from application level code on both sides (excessive synchronization)
- One-sided has no synchronization or matching overhead
- Initiate provides complete information describing the data transfer

Latency advantages of one-sided

- Key advantage in one sided is that all relevant information about the operation is provided by the initiator
- MPI requires retrieving addresses after matching operation the two point to point operations (even Elan offload a noticeable latency difference exists)

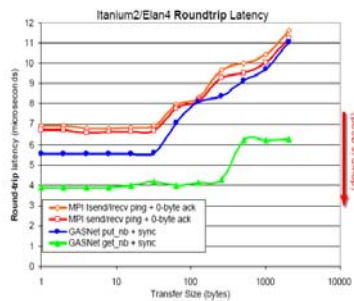


Figure 2. Latency of GASnet vs. MPI on Quadrics Elan4

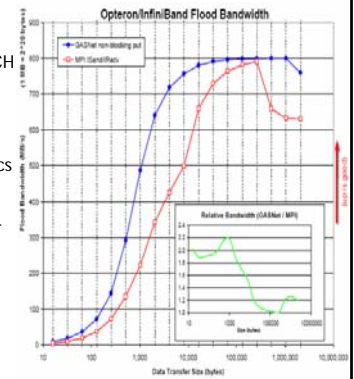
Flood bandwidth

shows flood bandwidth of GASnet infiniband vs MVAPICH

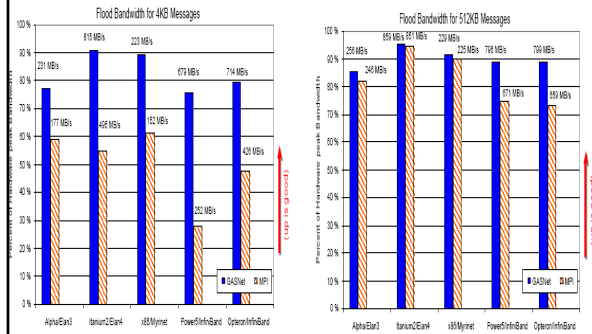
GASnet performs better due to 1-sided get/put ops better matched to RDMA ops vs MPI's 2-sided semantics

Memory pinning overhead in GASnet is managed better by Firehose algorithm

Firehose amortizes sync & pinning overhead across multiple memory operations



Flood bandwidth (2)



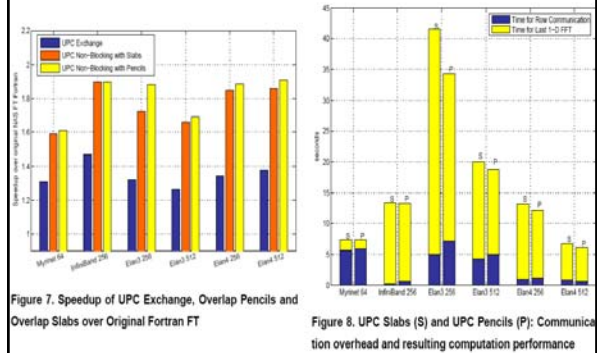
NAS FT Benchmark

- NAS FT implements a PDE using repeated FFT & IFT over 3 dimensions
- All-to-all, each thread exchanges its portion of the 3D FFT with every other thread
- After computing the FFT in one dimension, every thread locally transposes the data (exchanges), re-transposes and performs the remaining FFT

Optimizing FT with one-sided

- . Decompose the FFT computation & communication to smaller pieces to exploit overlap
- . Overlap slabs - is a method of decomposing 3D FFT to reduce the amount of time spent in communication-bound ops by overlapping the communication cost of sending previously computed slabs with computation of next slab
- . Overlap pencils - similar to slabs but further reduces granularity of communication & overlap
- . point-to-point messages of a single FFT row are sent while next row is computed

Results



Conclusions

- . Relative performance of UPC on GASnet vs MPI shows GASnet outperforms MPI in latency
- . GASnet achieves peak bandwidth for smaller message sizes than MPI

Thanks . . .