Final Report for
Project # DE-FC02-06ER25755
Dhabaleswar K. (DK) Panda and P. Sadayappan
The Ohio State University

1 Executive Summary

In this report, we describe the research accomplished by the OSU team under the Pmodels2 project. The team has worked on various angles: designing high performance MPI implementations on modern networking technologies (Mellanox InfiniBand (including the new ConnectX2 architecture and Quad Data Rate), QLogic InfiniPath, the emerging 10GigE/iWARP and RDMA over Converged Enhanced Ethernet (RoCE) and Obsidian IB-WAN), studying MPI scalability issues for multi-thousand node clusters using XRC transport, scalable job start-up, dynamic process management support, efficient one-sided communication, protocol offloading and designing scalable collective communication libraries for emerging multi-core architectures. New designs conforming to the Argonne's Nemesis interface have also been carried out.

All of these above solutions have been integrated into the open-source MVAPICH/MVAPICH2 software [24]. This software is currently being used by more than 2,100 organizations worldwide (in 71 countries). As of January '14, more than 200,000 downloads have taken place from the OSU Web site. In addition, many InfiniBand vendors, server vendors, system integrators and Linux distributors have been incorporating MVAPICH/MVAPICH2 into their software stacks and distributing it. Several InfiniBand systems using MVAPICH/MVAPICH2 have obtained positions in the TOP500 ranking of supercomputers in the world. The latest November '13 ranking include the following systems: 7th ranked Stampede system at TACC with 462,462 cores; 11th ranked Tsubame 2.5 system at Tokyo Institute of Technology with 74,358 cores; 16th ranked Pleiades system at NASA with 81,920 cores;

Work on PGAS models has proceeded on multiple directions. The Scioto framework, which supports task-parallelism in one-sided and global-view parallel programming, has been extended to allow multi-processor tasks that are executed by processor groups. A quantum Monte Carlo application is being ported onto the extended Scioto framework. A public release of Global Trees (GT) has been made, along with the Global Chunks (GC) framework on which GT is built. The Global Chunks (GC) layer is also being used as the basis for the development of a higher level Global Graphs (GG) layer. The Global Graphs (GG) system will provide a global address space view of distributed graph data structures on distributed memory systems.

The research has already led to 35 publications. The investigators have also given several invited talks and tutorials where latest research results have been presented.

2 Technical Contributions and Results

The project consists of research, design and development along two broad directions: 1) High Performance and Scalable MPI and 2) Partitioned Global Address Space Models.

High Performance and Scalable MPI

The following contributions were made along the first direction during the course of the project:

- **Study of MPI scalability issues:** The ever increasing demand for more computational power by scientific applications has lead to the increase in scale of compute clusters. Clusters with thousands of processing cores are already deployed and even larger clusters with tens-of-thousands of cores are in the planning stages. The MPI library utilized by the scientific applications should accordingly scale both in terms of performance delivered and resources consumed. The OSU team has been pursuing this direction of research. We have explored the viability of using the InfiniBand Unreliable Datagram as a scalable transport for MPI. Our designs and experimental results presented in [12] show that the performance and resource consumption of the MPI library could be significantly improved at large scale (with 8K-16K processors). In addition, our research in using a message-coalescing approach geared towards reduction of memory consumption in reliable connection oriented models [14] reveals that significant savings in resource consumption can be obtained while not sacrificing end application performance.

- **MPI implementations on modern networking technologies:** InfiniBand is emerging as an open-standard interconnect for designing next generation high performance clusters. For the last several years, OSU has been engaged in designing high performance MPI (MVAPICH with MPI-1 semantics and MVAPICH2 with MPI-2 semantics) [24] for InfiniBand-based clusters. As the InfiniBand networking standard matures, the next-generation of hardware are being released. Two of the prominent new network-interface adapters are the ConnectX-2

architecture by Mellanox technologies with support for QDR (Quad Data Rate) and offload capabilities and adapter from QLogic using on-loading technique. The ConnectX-2 QDR architecture is geared towards providing 40Gbps communication performance with PCI 2.0 interface technology. Our study and analysis in [33] reveals the capabilities of this adapter with the emerging Nehalem systems. We have also designed and developed an optimized implementation of MPI for QLogic's InfiniPath adapter using QLogic's PSM interface for both MVAPICH and MVAPICH2 stacks.

- **Scalable MPI Point-to-point (Two-sided and One-Sided) Communication:** The ever increasing demand for more computational power by scientific applications has lead to the increase in scale of compute clusters. Clusters with thousands of processing cores are already deployed and even larger clusters with tens-of-thousands of cores are in the planning stages. The MPI library utilized by the scientific applications should accordingly scale both in terms of performance delivered and resources consumed. Thus, it is essential to provide best and scalable designs and implementations for inter-node and intra-node communication (both one-sided and two-sided) with good overlap of computation with communication.

  1. We have explored the viability of using multiple InfiniBand transports (UD, RC and XRC) and studied various trade-offs (performance, reliability, and memory scalability) [13, 10] in inter-node communication. Based on this study, a hybrid design has been also proposed [9] which scales to 50-60K cores with constant memory footprint for the MPI library.
  2. A new TupleQ design has been proposed [11] to achieve zero-copy communication. These designs provide the best performance on large multi-core clusters with minimum memory footprint.
  3. Efficient design and implementation for MPI-2 one-sided communication has been proposed in [26] and [27]. These designs takes into account the atomic property for both inter-node and intra-node communication, and provides alternative schemes for fence synchronization, respectively.
  4. A new lock-free asynchronous rendezvous design has been proposed in [15]. This design provides asynchronous progress and overlap of computation with communication for large data transfers.
  5. Efficient design and implementation for MPI-2 one-sided communication operations using passive synchronization [28], atomic read-modify-write [29], and shared-memory backed windows [25] have been carried out.
  6. A general purpose protocol onload engine has been designed [16] for multi-core systems to achieve overlap of computation and communication in MPI operations.
  7. A new kernel-level design for one-sided operation has been designed [17]. This provides true one-sided support for one-sided communication using one-sided support. This paper has received a **Best Paper Award**.
  8. A new design for Argonne's Nemesis interface on InfiniBand has been designed [20]. This design takes advantage of efficient intra-node communication in the Argonne's Nemesis interface.

- **Scalable collective communication over large scale multi-core InfiniBand clusters:** As large scale InfiniBand multi-core clusters are being increasingly deployed, several challenges emerge pertaining to scalability and performance of collective operations. Collective Communications exhibit varying communication patterns and behaviors and accordingly, intelligent design decisions are required which guarantee high performance and scalable resource usage. The utility of RDMA semantics for collective operations was studied in [22]. Techniques such as message aggregation were proposed to cut-down the number of network operations and improve network utilization [21]. New shared-memory-based designs have been proposed [6] to achieve high performance allgather operations for multicore clusters. Optimized designs with collective offload support has been proposed in [7]. In [8, 32], topology-aware schemes have been exploited to deliver optimized performance for scatter, gather and broadcast operations. In [5], new designs are proposed to implement collectives with reduced power consumption.

- **Scalable Job Launching:** As HPC clusters are becoming very large (50-100K cores), starting-up MPI jobs itself is becoming the bottleneck - sometimes taking 15-20 minutes. Since modern-day clusters mostly use multi-core processors, we have designed a new scalable job launching framework. This framework and the follow-up optimizations [30, 31] are able to significantly cut down the MPI job-startup time. For example, on the TACC ranger, this framework is able to launch 32K MPI processes in just 90 seconds.

- **Dynamic Process Management:** MPI-2 specification provides dynamic process management (DPM) features. Not many MPI stacks support this feature. Also, there are no standard benchmarks to evaluate the performance

of dynamic process management in an MPI stack. We have designed [4] a high-performance InfiniBand support for MVAPICH2 and also have introduced a set of benchmarks to evaluate the performance of DPM feature on any MPI stack.

- **Congestion avoidance with InfiniBand:** For clusters, fat tree has become the most popular interconnection topology, due to its multi-pathing capabilities. However, even with fat tree, hot-spots may occur in the network depending upon the route configuration between end nodes and communication patterns in the application. To make matters worse, the deterministic routing nature of InfiniBand limits the application from effective use of multiple paths transparently and avoid the hot-spots in the network. To alleviate this situation, we have designed an MPI functionality which provides hot-spot avoidance for different communication patterns, without a-priori knowledge of the pattern [34]. We have leveraged LMC (LID Mask Count) mechanism of InfiniBand to create multiple paths in the network, and studied its efficiency in creation of contention free routes. Our evaluation with NAS Parallel Benchmarks and collective communication primitives shows significant improvement compared to the current state-of-the-art designs.

- **Topology Agnostic Routing in InfiniBand and Congestion avoidance:** For clusters, fat tree has become the most popular interconnection topology, due to its multi-pathing capabilities. However, even with fat tree, hot-spots may occur in the network depending upon the route configuration between end nodes and communication patterns in the application. To make matters worse, the deterministic routing nature of InfiniBand limits the application from effective use of multiple paths transparently and avoid the hot-spots in the network. To alleviate this situation, we have designed an MPI functionality which provides network agnostic routing, without a-priori knowledge of the pattern [35].

- Hybrid MPI and Stream Processing Model: As stream processing is becoming important for many applications, we have designed a new hybrid MPI and stream processing model [23] to support these emerging applications.

Partitioned Global Address Space models

The team has made the following contributions along the second direction during the course of the project:

- **The Scioto framework [1, 2, 3]** supports task-parallelism in one-sided and global-view parallel programming models. It provides lightweight, locality aware dynamic load balancing and can interoperate with existing parallel models including MPI, SHMEM, CAF, and Global Arrays. Through task parallelism, the Scioto framework provides an approach to address the issues of load imbalance and heterogeneity as well as dynamic mapping of computations on multicore architectures.

  Scioto provides the most scalable work-stealing based implementation of load balancing that we are aware of. On the Chinook system at PNNL, scalability up to over 10,000 cores was achieved [2]. During this year, enhancement of the Scioto system has enabled multi-level parallelism, where intra-task parallelism within a Scioto task can be utilized by executing it on a GA processor group. This system is being used to port a quantum Monte Carlo code (developed by Theresa Windus at Iowa State University). It is expected that the port over Scioto will enable it to achieve better load balancing and scalability than the current implementation in NWChem.

  The Global Trees (GT) framework provides a multi-layered logical interface to a global address space view of distributed tree data structures, while providing scalable performance on distributed memory systems. The Global Trees (GT) library provides a global view of distributed linked tree structures and a set of routines which operate on these structures.

  Work during this year has resulted in the development of instrumentation to characterize GT access patterns, and a visualization GUI to display summarized statistics and characteristics of GT access patterns. For applications with iterative loops with similar access patterns in successive iterations (for example, the Barnes Hut algorithm for N-body simulation), node allocation based on profiled access patterns has been shown to result in good speedup [19]. A public release of GT has been made available. The Global Chunks (GC) layer, on which GT is built, is now being used to develop Global Graphs (GG), a high-level PGAS abstraction for general linked data structures.

- **Global Trees:** We have designed and prototyped the Global Trees (GT) framework for system which provides a multi-layered logical interface to a global address space view of distributed tree data structures, while providing scalable performance on distributed memory systems.

  The Global Trees (GT) library provides a global view of distributed linked tree structures and a set of routines which operate on these structures. We extend existing global view approaches with a framework that provides

support for parallel computations over irregular and dynamic tree structures. The GT design approach is based on two key insights. First, tree-based algorithms are easily expressed in a fine-grained manner, but data movement must be done at a much coarser level of granularity for good performance. Second, since GT is focused on a single data abstraction, this allows us to exploit attributes unique to tree structures and to provide optimized routines for common operations.

Global Trees is a run-time library and programming interface which provides a global address view for tree-based data structures on distributed memory clusters. In contrast to traditional distributed shared memory systems, GT is designed specifically to support dynamic linked data structures and as a result is able to focus on providing efficient access for applications which use these structures. Global Trees automates the allocation, distribution, and communication of shared data. The main ideas incorporated into GT are summarized below:

1. Efficient Fine-Grained Data Access: Our technique combines the ease of programming in a shared memory environment with the efficiency of coarse-grained data movement during a dynamic computation. Tree data is grouped into chunks, from which subsequent data accesses can be performed without communication. Each process involved in the parallel computation may independently and asynchronously access global tree data with no explicit cooperation from other processes. While GT provides fine-grained node-level data access similar to shared memory programming models, internally the system is aware of the data distribution and is able to exploit data structure specific knowledge to improve locality and yield good performance.
2. Tree Structure Optimizations: We exploit the linked nature of of tree structures when resolving portable global references. Global pointers which reside inside the same chunk as the referenced node are modified to portable pointers which achieve nearly the same performance as normal pointer dereferencing.
3. High Level Operations on Distributed Trees: The Global Trees framework provides parallelized common tree operations which are optimized to take advantage of locality information known to the runtime.
4. Application-driven Customization: Our approach provides a set of data structures and operations which may be used to implement generalized tree algorithms. When high performance is critical, GT permits developers to directly influence runtime behavior and provides performance and profiling statistics to tune the application.

The GT prototype has been empirical evaluated with the Barnes-Hut benchmark from the SPLASH-2 parallel benchmark suite. Performance and scalability were found to be considerably better than Intel's Cluster OpenMP on a cluster of 64 nodes. Details of the framework and the experimental evaluation may be found in a report that has been submitted for publication [18].

- **Taskpool model for independent tasks:** The basic execution model of Global Arrays (GA) in its current distribution is MPI-like, i.e. there are P GA processes (typically equal to the number of physical processors for execution) that are "long-lived" and exist through the parallel program's execution. Like an MPI process, each GA process also has persistent "local" state in its copy of all local variables. While it improves upon MPI in providing a convenient global view of large arrays, the above GA model does not provide any better support for load balancing than MPI - the programmer must do it explicitly. In order to provide system-supported load balancing, the taskpools extension to the GA model is being developed.

  A taskpool is a set of tasks, where each task's inputs and outputs are specified as portions of global arrays. A task in the taskpool can perform arbitrary local computation, but using the *GetfromGlobal* + *ComputeLocal* + *PuttoGlobal* model. Thus, its input operands are all portions of global arrays, and its outputs specify portions of global arrays, but within the body of a task's code, no references to global arrays are permitted. In the first prototype of taskpools, all tasks in the pool are independent. The taskpool model was created primarily to provide system support for load balancing. Since the tasks explicitly specify the portions of global arrays that they need to copy in or update, the system can perform locality-aware load balancing of tasks among the processors. Affinity of processors to memory - with multi-core and SMP nodes - can be exploited in this load balancing process, without needing to impose a two-level (e.g. MPI+OpenMP) programming model.

  We have implemented the taskpool model for independent tasks over the GA/ ARMCI interface. A simple global lock-based dynamic load balancing scheme has been implemented. We have also developed an hierarchical load-balancing scheme for SMP and multi-core systems.

3 Personnel Supported

Eight graduate students, one programmer (part-time) and two post-docs (part-time) have been supported by this grant.

4 Invited Tutorials and Talks Presented

The investigators have delivered the following keynote talks, invited tutorials and talks where results from this research have been presented.

1. D. K. Panda and S. Sur, *InfiniBand and 10-Gigabit Ethernet for Dummies*, Int'l Supercomputing Conference (ISC '11), June 19, 2011.
2. D. K. Panda and S. Sur, *Designing High-End Computing Systems with InfiniBand and High-Speed Ethernet*, Int'l Supercomputing Conference (ISC '11), June 19, 2011.
3. D. K. Panda, *Designing High-End Computing Systems and Programming Models with InfiniBand and High-Speed Ethernet* Int'l Conference on Supercomputing (ICS '11), May 31, 2011.
4. D. K. Panda and S. Sur, *Designing Cloud and Grid Computing Systems with InfiniBand and High-Speed Ethernet*, Int'l Conference on Cluster and Grid Computing (CCGrid '11), May 23, 2011.
5. D. K. Panda, P. Balaji and S. Sur, *InfiniBand and Ethernet Architectures for Scientific and Enterprise Computing: Opportunities and Challenges*, Int'l Symposium on High Performance Computing Architecture (HPCA '11), Feb. 12, 2011.
6. D. K. Panda, *Networking Technologies for Clusters: Where do We Stand and What Lies Ahead?*, Keynote Talk, Int'l Conference on Parallel and Distributed Systems (ICPADS '10), Shanghai, China, Dec 9, 2010.
7. D. K. Panda, P. Balaji and S. Sur, *InfiniBand and 10-Gigabit Ethernet for Dummies*, Int'l Conference on Supercomputing (SC '10), Nov. 15, 2010.
8. D. K. Panda, P. Balaji and S. Sur, *Designing High-End Computing Systems with InfiniBand and High-Speed Ethernet*, Int'l Conference on Supercomputing (SC '10), Nov. 15, 2010.
9. D. K. Panda, *Networking Technologies for Exascale Computing Systems: Opportunities and Challenges*, Keynote Talk, HPC China Conference, Beijing, Oct 28, 2010.
10. D. K. Panda, *Design of Collectives and One-Sided Operations in MPI and their Impact on Application-Level Performance and Scalability*, Keynote Talk, HPC Advisory Council Workshop, Beijing, China, Oct 27, 2010.
11. D. K. Panda, P. Balaji and S. Sur, *Designing High-End Computing Systems and Programming Models with InfiniBand and High-speed Ethernet*, Int'l Conference on Partitioned Global Address Space (PGAS '10), October 15, 2010.
12. D. K. Panda, P. Balaji and S. Sur, *Designing High-End Computing Systems with InfiniBand and High-speed Ethernet*, Int'l Conference on Cluster Computing (Cluster '10), September 20, 2010.
13. D. K. Panda, P. Balaji and S. Sur, *Designing High-End Computing Systems with InfiniBand and High-speed Ethernet*, Hot Interconnect (HOTI), August 20, 2010.
14. D. K. Panda, *InfiniBand Software Networking Technologies*, Discovery 2015 Workshop, Oak Ridge National Laboratory, July 13, 2010.
15. D. K. Panda, P. Balaji and S. Sur, *Designing High-End Computing Systems with InfiniBand and High-speed Ethernet*, Int'l Conference on Supercomputing (ICS), June 1, 2010.
16. D. K. Panda and P. Balaji, *Designing High-End Computing Systems with InfiniBand and 10-Gigabit Ethernet*, Int'l Supercomputing Conference (ISC), May 30, 2010.
17. D. K. Panda, *MVAPICH/MVAPICH2 Update, Future Plans and Path Towards Exascale*, Open Fabrics Sonoma Workshop, March 16, 2010.
18. D. K. Panda, *Designing High Performance, Scalable and Fault-Resilient MPI Library for Modern Clusters*, Pacific Northwest National Library (PNNL), February 23, 2010.
19. D. K. Panda and P. Balaji, *InfiniBand and Ethernet Architectures for Scientific and Enterprise Computing: Opportunities and Challenges*, Int'l Symposium on High-Performance Computer Architecture (HPCA-16), Jan 10, 2010.

20. D. K. Panda, *New Research Areas in HPC Technologies Development*, Center for Development of Advanced Computing (C-DAC), Pune, India, January 8, 2010.

21. D. K. Panda, *Evolution of HPC Interconnects in next 5 years and their Role in Peta/Exascale Systems*, Center for Development of Advanced Computing (C-DAC), Pune, India, January 8, 2010.

22. D. K. Panda, *MVAPICH2 Project: Latest Status and Future Plans*, BOF on MPICH2, in conjunction with Supercomputing (SC '09), Nov 19, 2009.

23. D. K. Panda, *MVAPICH/MVAPICH2 Project: Latest Status and Future Plans*, Presentation at Mellanox Booth, Supercomputing Conference (SC '09), Nov. 17, 2009.

24. D. K. Panda, P. Balaji and M. Koop, *InfiniBand and 10-Gigabit Ethernet for Dummies*, Int'l Conference on Supercomputing (SC '09), Nov. 15, 2009.

25. D. K. Panda, P. Balaji and M. Koop, *Designing High-End Computing Systems with InfiniBand and 10-Gigabit Ethernet*, Int'l Conference on Supercomputing (SC '09), Nov. 15, 2009.

26. Bruce Palmer, Manojkumar Krishnan, Sriram Krishnamoorthy, and P Sadayappan, *Parallel Programming Using the Global Arrays Toolkit*, Cluster 2009, New Orleans, LA, September 4, 2009.

27. D. K. Panda, P. Balaji and M. Koop, *Designing High-End Computing Systems with InfiniBand and 10-Gigabit Ethernet*, IEEE Cluster (Cluster '09), September 4, 2009.

28. D. K. Panda, *Networking Technologies for Clusters: Where do We Stand and What Lies Ahead?*, Plenary Talk, Int'l Conference on Cluster Computing (Cluster '09), Sept. 2, 2009.

29. D. K. Panda, P. Balaji and M. Koop, *InfiniBand and 10-Gigabit Ethernet for Dummies*, jointly with P. Balaji and M. Koop, Hot Interconnect (HotI '09), August 25, 2009.

30. D. K. Panda, P. Balaji and M. Koop, *Designing High-End Computing Systems with InfiniBand and 10-Gigabit Ethernet*, jointly with P. Balaji and M. Koop, Hot Interconnect (HotI '09), August 25, 2009.

5 Web Sites

The following websites contain links to the papers, presentations and software resulting from this research:

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

References

[1] J. Dinan, S. Krishnamoorthy, B. Larkins, J. Nieplocha, and P. Sadayappan. Scioto: A framework for global-view task parallelism. In *Proc. International Conference on Parallel Processing (ICPP 2008)*, 2008.

[2] J. Dinan, B. Larkins, S. Krishnamoorthy, J. Nieplocha, and P. Sadayappan. Scalable work stealing. In *Proc. Supercomputing (SC 2009)*, 2009.

[3] James Dinan. *Scalable Task Parallel Programming in the Partitioned Global Address Space*. PhD thesis, Ohio State University, Department of Computer Science and Engineering, June 2010.

[4] T. Gangadharappa, M. Koop, and D. K. Panda. Designing and Evaluating MPI-2 Dynamic Process Management Support for InfiniBand. In *Int'l Workshop on Parallel Programming Models and Systems Software for High-End Computing*, September 2009.

[5] K. Kandalla, E. Mancini, S. Sur, and D. K. Panda. Designing Power-Aware Collective Communication Algorithms for InfiniBand Clusters. In *Int'l Conference on Parallel Processing (ICPP)*, September 2010.

[6] K. Kandalla, H. Subramoni, G. Santhanaraman, M. Koop, and D. K. Panda. Designing Multi-Leader-Based Allgather Algorithms for Multi-core Clusters. In *International Workshop on Communication Architecture for Clusters, held in conjunction with IPDPS '09*, May 2009.

[7] K. Kandalla, H. Subramoni, J. Vienne, K. Tomko, S. Sur, and D. K. Panda. Designing Non-blocking Broadcast with Collective Offload on InfiniBand Clusters: A Case Study with HPL . In *19th Annual Symposium on High Performance Interconnects(HotI '11)*, August, 2011.

[8] K. Kandalla, H. Subramoni, A. Vishnu, and D. K. Panda. Designing Topology-Aware Collective Communication Algorithms for Large Scale InfiniBand Clusters: Case Studies with Scatter and Gather. In *Int'l Workshop on Communication Architecture for Clusters (CAC 10), in conjunction with IPDPS '10*, April 2010.

[9] M. Koop, T. Jones, and D. K. Panda. MVAPICH-Aptus: Scalable High-Performance Multi-Transport MPI over InfiniBand. In *International Parallel and Distributed Processing Symposium (IPDPS08)*, April 2008.

[10] M. Koop, R. Kumar, and D. K. Panda. Can Software Reliability Outperform Hardware Reliability on High Performance Interconnects? A Case Study with MPI over InfiniBand. In *International Conference on Supercomputing (ICS08)*, June 2008.

[11] M. Koop, J. Sridhar, and D. K. Panda. TupleQ: Fully-Asynchronous and Zero-Copy MPI over InfiniBand. In *International Parallel and Distributed Processing Symposium (IPDPS09)*, May 2009.

[12] M. Koop, S. Sur, Q. Gao, and D. K. Panda. High Performance MPI Design using Unreliable Datagram for Ultra-Scale InfiniBand Clusters. In *International Conference on Supercomputing (ICS07)*, page To Appear, 2007.

[13] M. Koop, S. Sur, and D. K. Panda. Zero-Copy Protocol for MPI using InfiniBand Unreliable Datagram. In *IEEE Cluster (Cluster 07)*, September 2007.

[14] Matthew J. Koop, Terry Jones, and Dhabaleswar K. Panda. Reducing Connection Memory Requirements of MPI for InfiniBand Clusters: A Message Coalescing Approach. In *International Symposium on Cluster Computing and the Grid*, pages 495–504, 2007.

[15] R. Kumar, A. Mamidala, M. Koop, G. Santhanaraman, and D. K. Panda. Lock-free Asynchronous Rendezvous Design for MPI Point-to-point Communication. In *EuroPVM/MPI*, page to be presented., September 2008.

[16] P. Lai, P. Balaji, R. Thakur, and D. K. Panda. ProOnE: A General Purpose Protocol Onload Engine for Multi- and Many-Core Architectures. In *International Supercomputing Conference (ISC09)*, June 2009.

[17] P. Lai, S. Sur, and D. K. Panda. Truly One-Sided MPI-2 RMA Intra-node Communication on Multi-core Systems. In *Int'l Supercomputing Conference (ICS)*, June 2010.

[18] B. Larkins, J. Dinan, S. Krishnamoorthy, A. Rountev, and P. S adayappan. Global trees: A framework for linked data structures on distributed memory parallel systems. In *Submitted to Supercomputing (SC 2008)*, 2008.

[19] D. Brian Larkins. *Efficient Run-time Support for Global View Programming of Linked Data Structures on Distributed Memory Parallel Systems*. PhD thesis, Ohio State University, Department of Computer Science and Engineering, June 2010.

[20] M. Luo, S. Potluri, P. Lai, E. P. Mancini, H. Subramoni, K. Kandalla, S. Sur, and D. K. Panda. High Performance Design and Implementation of Nemesis Communication Layer for Two-sided and One-Sided MPI Semantics in MVAPICH2. In *Int'l Workshop on Parallel Programming Models and Systems Software for High-End Computing (P2S2 '10), in conjunction with ICPP '10*, September 2010.

[21] Amith R. Mamidala, Debraj De, Abhinav Vishnu, Sundeep Narravula, and Dhabaleswar K. Panda. Scalable Collective Communication for Next-Generation Multicore Clusters with InfiniBand . In *Technical Report No. OSU-CISRC-6/07-TR49*, 2007.

[22] Amith R. Mamidala, Sundeep Narravula, Abhinav Vishnu, Gopalakrishnan Santhanaraman, and Dhabaleswar K. Panda. On Using Connection-Oriented vs. Connection-Less Transport for Performance and Scalability of Collective and One-Sided Operations: Trade-offs and Impact. In *Symposium on Principles and Practices of Parallel Programming*, pages 46–54, 2007.

[23] E. P. Mancini, G. Marsh, and D. K. Panda. An MPI-Stream Hybrid Programming Model for Computational Clusters. In *Int'l Symposium on Cluster Computing and the Grid (CCGrid)*, May 2010.

[24] Network-Based Computing Laboratory. MVAPICH/ MVAPICH2: MPI-1/ MPI-2 for InfiniBand and iWARP. `http://mvapich.cse.ohio-state.edu`.

[25] S. Potluri, H. Wang, V. Dhanraj, S. Sur, and D. K. Panda. Optimizing MPI One Sided Communication on Multi-core InfiniBand Clusters using Shared Memory Backed Windows. In *EuroMPI*, 2011.

[26] G. Santhanaraman, P. Balaji, K. Gopalakrishnan, R. Thakur, W. Gropp, and D. K. Panda. Natively Supporting True One-sided Communication in MPI on Multi-core Systems with InfiniBand. In *International Symposium on Cluster Computing and the Grid*, 2009.

[27] G. Santhanaraman, T. Gangadharappa, S. Narravula, A. Mamidala, and D. K. Panda. Design Alternatives for Implementing Fence Synchronization in MPI-2 One-sided Communication on InfiniBand Clusters. In *Int'l Conference on Cluster Computing (Cluster)*, September 2009.

[28] G. Santhanaraman, S. Narravul, and D. K. Panda. Designing Passive Synchronization for MPI-2 One-Sided Communication to Maximize Overlap. In *International Parallel and Distributed Processing Symposium (IPDPS08)*, April 2008.

[29] G. Santhanaraman, S. Narravula, A. Mamidala, and D. K. Panda. MPI-2 One Sided Usage and Implementation for Read Modify Write operations: A case study with HPCC. In *EuroPVM/MPI*, September 2007.

[30] J. Sridhar, M. Koop, J. Perkins, and D. K. Panda. ScELA: Scalable and Extensible Launching Architecture for Clusters. In *Int'l Conference on High Performance Computing (HiPC 08)*, December 2008.

[31] J. Sridhar and D. K. Panda. Impact of Node Level Caching in MPI Job Launch Mechanisms. In *EuroPVM/MPI*, September 2009.

[32] H. Subramoni, K. Kandalla, J. Vienne, S. Sur, B. Barth, K. Tomko, R. McLay, K. Schulz, and D. K. Panda. Design and Evaluation of Network Topology-/Speed-Aware Broadcast Algorithms for InfiniBand Clusters. In *IEEE Cluster '11*, 2011.

[33] H. Subramoni, M. Koop, and D. K. Panda. Designing Next Generation Clusters: Evaluation of InfiniBand DDR/QDR on Intel Computing Platforms. In *Int'l Symposium on Hot Interconnects*, August 2009.

[34] Abhinav Vishnu, Matthew J. Koop, Adam Moody, Amith R. Mamidala, Sundeep Narravula, and Dhabaleswar K. Panda. Hot-Spot Avoidance With Multi-Pathing Over InfiniBand: An MPI Perspective. In *International Symposium on Cluster Computing and the Grid*, pages 479–486, 2007.

[35] Abhinav Vishnu, Matthew J. Koop, Adam Moody, Amith R. Mamidala, Sundeep Narravula, and Dhabaleswar K. Panda. Topology Agnostic Hot-Spot Avoidance with InfiniBand. In *Concurrency and Computation: Practice and Experience, Best Papers from CCGrid '07*, page in press., 2007.