Evaluating The Potential of Cray Gemini Interconnect for PGAS Communication Runtime Systems

Abhinav Vishnu [#1], Monika ten Bruggencate [#2], and Ryan Olson [#2]
[#1] High Peformance Computing Group,
Pacific Northwest National Laboratory,
902 Battelle Blvd, Richland, WA
[#2] Cray, Inc

***Abstract*—The Cray Gemini Interconnect has been recently introduced as the next generation network for building scalable multi-petascale supercomputers. The Cray XE6 systems, which use the Gemini Interconnect are becoming available with Message Passing Interface (MPI) and Partitioned Global Address Space (PGAS) Models such as as Global Arrays, Unified Parallel C, Co-Array Fortran and Cascade High Performance Language. These PGAS models use one-sided communication runtime systems such as MPI-Remote Memory Access, Aggregate Remote Memory Copy Interface and proprietary communication runtime systems. The primary objective of our work is to study the potential of Cray Gemini Interconnect by designing application specific micro-benchmarks using the DMAPP userspace library. We design micro-benchmarks to study the performance of simple communication primitives and application specific micro-benchmarks to understand the behavior of Gemini Interconnect at scale. In our experiments, the Gemini Interconnect can achieve a peak bandwidth of 6911 MB/s and a latency of 1μs for get communication primitive. Scalability tests for atomic memory operations and shift communication operation up to 65536 processes show the efficacy of the Gemini Interconnect.**

I. Introduction

The Cray Gemini Interconnect [1] has been recently proposed to be the next generation Interconnect for designing scalable multi-petascale systems. The Gemini Interconnect is the primary architectural difference between the Cray XE6 systems and the previous-generation XT-based supercomputers, which use the SeaStar Interconnect [2]. Cray Gemini is a custom system-on-a-chip architecture ASIC which has a 3D Torus Interconnect, and connects to its six nearest neighbours.

Message Passing Interface (MPI) [3], [4], the *de facto* model for writing parallel applications has become available on the Cray XE6 systems. Partitioned Global Address Space (PGAS) Models such as Global Arrays [5], Unified Parallel C (UPC) [6], Co-Array Fortran [7] and Cascade High Performance Language (Chapel) [8] have also become available. These PGAS models use one-sided communication runtime systems, such as MPI-Remote Memory Access (RMA) [4], Aggregate Remote Memory Copy Interface (ARMCI) [9] and proprietary one-sided communication runtime systems.

The primary objective of our work is to design micro-benchmarks motivated from application case studies using the Cray DMAPP [10] userspace library. We specifically design the communication micro-benchmarks for PGAS models and one-sided communication runtime systems. The intended outcome of this study is to provide designers of one-sided communication runtime systems with an in-depth performance analysis of performance parameters of the Cray Gemini Interconnect. Our study includes designing micro-benchmarks for simple one-sided communication primitives (put, get, and atomic memory operations (AMOs)) with various inter-process configurations, handling contiguous and non-contiguous datatypes, simultaneous request of message transfers, scalability analysis of atomic memory operations for dynamic load balancing and shift communication operation with different routing options. In our experiments, the Cray Gemini Interconnect can achieve a peak bandwidth of 6911 MB/s and a latency of 1μs for get communication primitive. Scalability tests on atomic memory operations and shift communication operation up to 65536 processes show the efficacy of the Cray Gemini Interconnect. We plan to release the tests in the near future, which may be extended to study the use-cases not studied in this paper. They may also be used for making design choices with PGAS applications and communication runtime systems.

The rest of the paper is organized as follows. In section II, we present the related work. In section III, we present an in-depth performance of Cray Gemini Interconnect with communication primitives. We conclude and present future directions in section V.

II. Related Work

Performance analysis of high-speed Interconnects has been widely studied by researchers in the last couple of decades. Liu et al. performed the comparison of InfiniBand [11], Myrinet [12] and Quadrics [13] using micro-benchmarks [14]. Liu concluded that InfiniBand outperforms Myrinet and Quadrics for simple micro-benchmarks. However, the performance comparison does not show a significant performance improvement on NAS Parallel Benchmarks [15]. Govindaraju et al. presented design of IBM High Performance Switch (HPS) Interconnect and performance evaluation using simple micro-benchmarks [16], [17]. IBM HPS provides Remote

Direct Memory Access (RDMA) capability on connectionless transport semantics. Petrini et al. presented an evaluation of Quadrics Interconnect using simple user-level access benchmarks [13]. The article presents an in-depth performance analysis of leveraging offloaded collective communication capabilities provided by Quadrics Interconnect. Leveraging the communication and scalability primitives provided InfiniBand, Panda et al. have presented design of Message Passing Interface (MPI) using InfiniBand DDR Interconnect, InfiniBand QDR Interconnect and Offloaded collective communication primitives using InfiniBand QDR Interconnect [18], [19], [20].

Alverson et al. have studied the performance of the Cray Gemini Interconnect with simple micro-benchmarks [1]. The above work provides an in-depth micro-architecture of the Gemini System Interconnect, with initial performance evaluation. The objective of our work is to analyze the performance of Cray Gemini Interconnect in much more detail at the userspace level with primary focus on PGAS programming models and associated communication runtime systems.

III. Performance Evaluation

In this section, we present performance evaluation of the Cray Gemini Interconnect using micro-benchmarks. The micro-benchmarks evaluate the performance of simple primitives such as put, get, and atomic memory operations (AMOs). Using dynamic load balancing counters as a use-case, we study the scalable performance of AMOs. We also study the performance of non-contiguous datatypes supported by Distributed Memory Applications (DMAPP) [10]. Lastly, the performance of shift communication pattern is studied, comparing the performance of routing options provided by the Cray Gemini Interconnect. We begin with a description of our Experimental Testbed.

A. Experimental Testbed

We use the NERSC Hopper Phase II system [21] for performance evaluation. The Hopper Phase II system is a Cray XE6 system, which has 6384 nodes with two twelve-core AMD Magny-Cours 2.1 GHz processors per node. Each node has 24 cores providing a total of 153,216 cores in the NERSC Hopper Phase II system. A total memory of 32 GB DDR3 1333 MHz is available per node. Each AMD core has an independent L1 and L2 cache of sizes 64KB and 512 KB, respectively. A 6MB L3 shared cache connects six magny-cours processors. The Hopper Phase II system provides a peak performance of 1.28 PF/s. Figure 1 shows a block diagram of the AMD Magny-Cours processor in Hopper Phase II supercomputer.

The Cray Gemini Interconnect is the most significant difference in Cray XE6 systems in comparison to the previous generation Cray XT systems. A block diagram of Cray Gemini ASIC is shown in Figure 2. Each Gemini ASIC has two Network Interface Cards (NICs), and a 48-port YARC router. The NICs within an ASIC are connected to a Netlink block, which provides internal routing between the NICs. A total of eight links connect a Netlink block to a router. Each Gemini

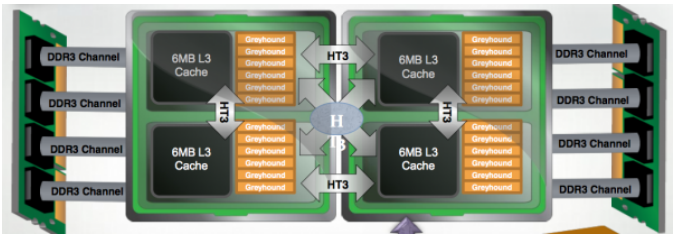

Fig. 1. Block Diagram of AMD Magny-Cours Processor [21]

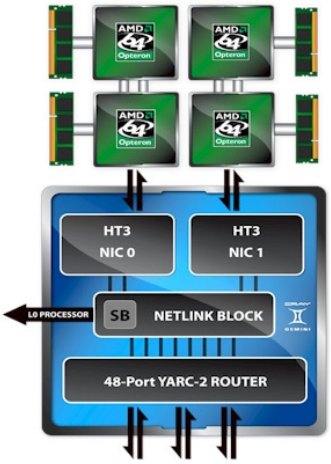

Fig. 2. Block Diagram of Gemini ASIC [21]

ASIC is connected to other ASICs using a 3D Torus topology. A total of eight links each are used to connect in the 'x' and 'z' dimensions, and four links are used in the 'y' dimension [1].

The Cray Gemini Interconnect supports multiple routing options. Each routing algorithm is dimension-order, but provides different flexbility in using the links in each dimension. The adaptive dimension-order routing allows the packets to be scheduled on lightly loaded links adaptively [1], while hash deterministic routing provides lesser flexibility. (More details on the exact routing algorithm for hash-deterministic are not publically available).

B. Evaluation Methodology

We evalute the one-sided communication primitives (put, get, atomic memory operations) for multiple configurations:

- Intra-NIC: The communicating processes are scheduled on the same socket and communicate using the same NIC.
- Inter-NIC, Intra-Gemini ASIC: The processes are scheduled on nodes sharing the same Gemini ASIC.
- Inter-ASIC: The processes are scheduled on nodes attached to different ASICs.

The Intra-NIC configuration has other possible combinations, such as processes scheduled on different sockets. However, a communication runtime system is expected to use load/stores and/or shared memory based based communication for processes scheduled on the same node, whenever possible. The intra-NIC configuration is primarily for pedantic purposes.

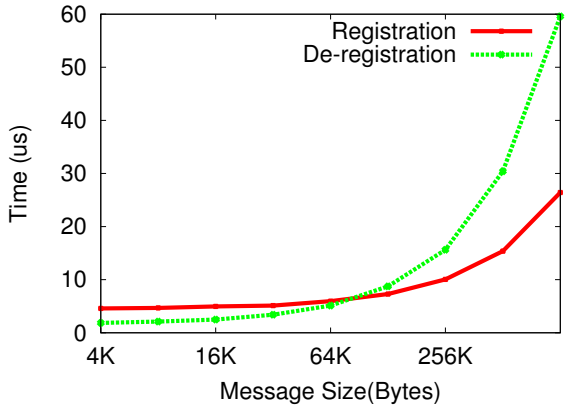

Fig. 3. Cost of Registration and De-registration, 4K Pagesize

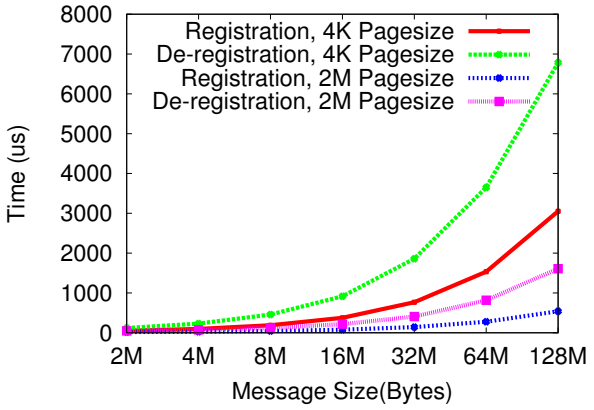

Fig. 4. Cost of Registration and De-registration, Large Pagesize

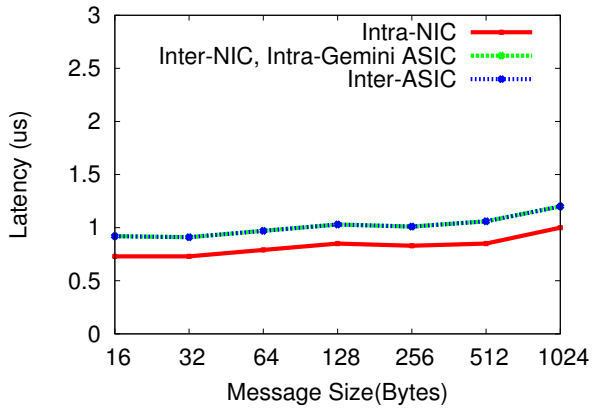

Fig. 5. Put Latency

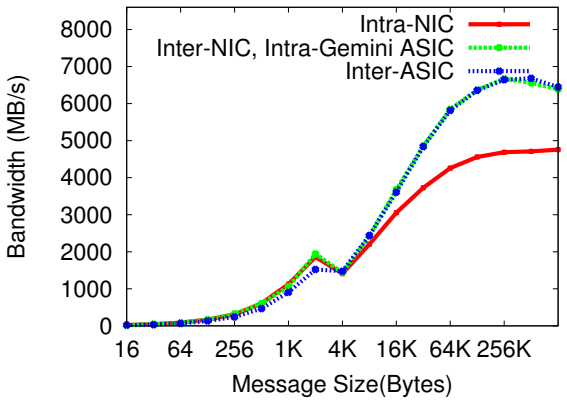

Fig. 6. Put Bandwidth with Different Configurations

With Inter-ASIC configuration, other factors may impact the overall performance, including the hop-distance and dimension(s). For each of the tests involving exactly two nodes, we ensure that the processes are scheduled in either the 'x' or 'z' dimension with a distance of one-hop. For each of the tests, we use a relaxed ordering protocol for HyperTransport and adaptive routing for Inter-ASIC communication, unless specified otherwise. The DMAPP [10] layer ensures the location consistency even for relaxed ordering protocol, ruling out the possibility of getting a stale data for a get operation, which immediately follows a put to the same memory region.

C. Performance Evaluation of Simple Primitives

In this section, we present a performance evaluation of Cray Gemini Interconnect by designing micro-benchmarks of simple primitives. We use Distributed Memory Applications (DMAPP) [10] interface for designing these micro-benchmarks. DMAPP is not intended for end applications, but for designing one-sided communication runtime systems.

1) Memory Allocation and Registration: The Gemini Interconnect supports symmetric memory allocation through the DMAPP Interface [10]. This precludes a need for on-the-fly registration and address exchange for the distributed data structures of PGAS models such as GA, UPC and CAF. The local buffers for communication may not be allocated using a symmetric heap and may require on-the-fly registration. Figures 3 and 4 show the cost of on-the-fly registration for pagesizes of 4Kbytes and 2Mbytes, respectively. The registration cost increases linearly for each of the pagesizes. For buffers of size greater than 2Mbytes, the cost of registration for 2Mbytes pagesize is 3.8 times faster than the 4Kbytes page size. However, the cost of deregistration is more than the cost of registration. These costs of registration and de-registrations are an important factor in deciding the optimal pagesize, and total size of local buffers to be registered, since the cost of on-the-fly deregistration is expensive. To limit the registration overhead which an application would otherwise incur as the result of on-the fly memory registration, DMAPP implements an internal memory registration cache. For achieving best performance, PGAS runtime systems should avoid small dynamic memory registrations in favor for larger ones.

2) Put Communication Primitive : Figure 5 shows the latency of put communication primitive with three different configurations. For Inter-NIC configurations, a latency of .9μs is observed. The DMAPP blocking interface provides location consistency - a return from a blocking interface (put, get,

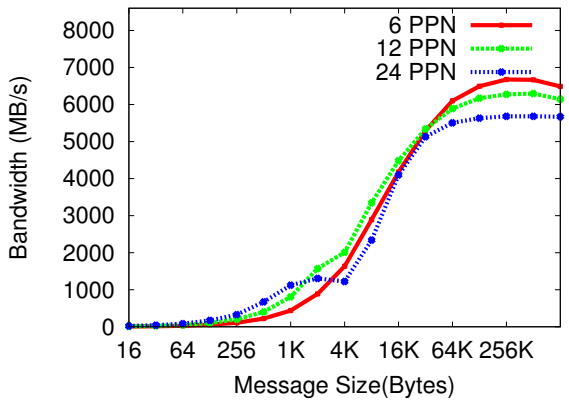

Fig. 7. Multi-Put Bandwidth

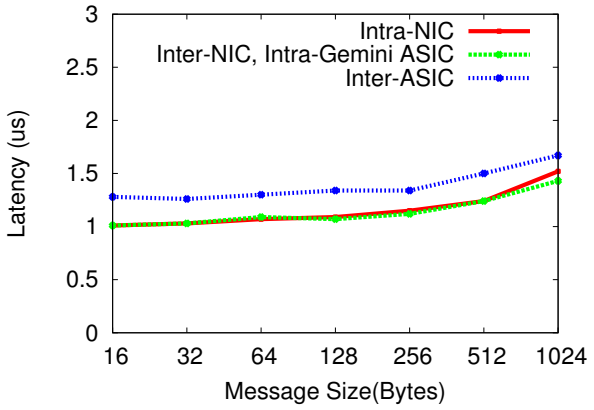

Fig. 8. Get Latency with Different Configurations

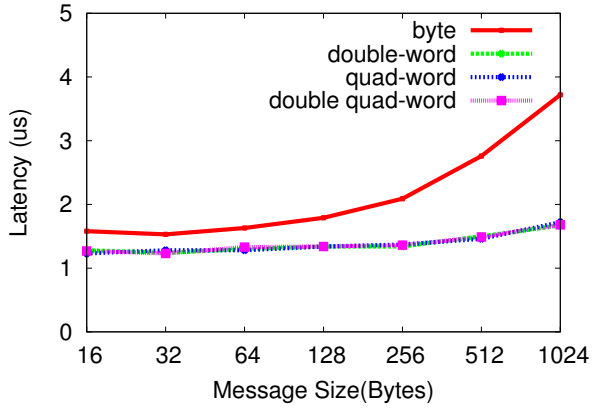

Fig. 9. Get Latency with Different Datatypes, Inter-Gemini ASIC

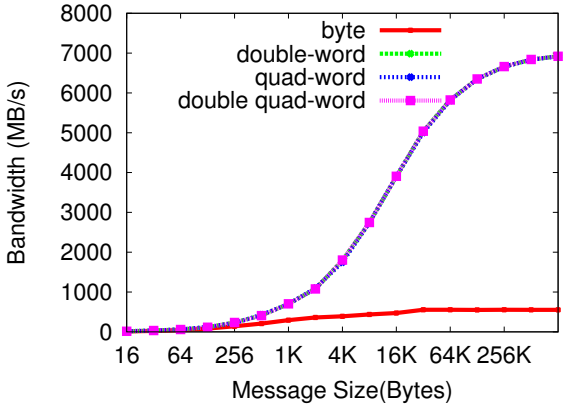

Fig. 10. Get Bandwith with Different Datatypes, Inter-Gemini ASIC

AMOs) indicates that source buffer is reusable and the result of the operation is globally visible. The Intra-NIC latency is .72μs.

Figure 6 shows the bandwidth achieved by the put communication primitive. The test is designed to run several hundred iterations with exactly one outstanding put from a source to a target. The Inter-NIC configurations achieve a peak bandwidth of 6681 MB/s. The Intra-NIC configuration achieves 4758 MB/s. The Intra-NIC performance is limited by the achievable bandwidth from HyperTransport, since the DMA engine has to read from memory and write to the memory on the same link simultaneously. DMAPP [10] provides a tunable threshold for using Programmed I/O (PIO) and DMA. In our experiments, the messages greater than the size of 2Kbytes use DMA, which results in a slight performance drop, as observed from the Figure 6.

Figure 7 shows the aggregate bandwidth of Inter-ASIC communication using the put primitive with 6, 12 and 24 processes per node (PPN). In every iteration of the test, each process on one node performs a put operation on the memory of the symmetric process on other node. The Inter-ASIC configuration is used for performance evaluation. Simultaneous communication is frequently observed with applications such as STOMP [22] and NWChem [23], which use PGAS models, in addition to the legacy MPI applications. The small messages - bounded by the injection rate, show the best performance with 12 and 24 PPN. For large message sizes, a significant contention is observed for 12 and 24 PPN, while 6 PPN achieves a peak aggregate bandwidth of 6675 MB/s.

3) Get Communication Primitive: Figure 8 shows the latency of the get primitive with different configurations. A latency of 1.24μs is observed for Inter-ASIC configuration. The other configurations show a latency of 1μs. Completion semantics of blocking get primitive follow location consistency, as discussed in the previous section.

Figure 9 shows the latency of a blocking get operation for different datatypes in the Inter-ASIC configuration. Performance of double-word (4 bytes), quad-word (8 bytes) and double quad-word (16 bytes) is identical and almost constant at 1.2μs for up to 512 bytes. The performance of datatype byte is considerably worse. Cray Gemini Interconnect requires that local and remote addresses and length of PIO get must be at least double-word aligned. The DMAPP API however supports get with byte granularity and must use a properly aligned temporary buffer to work around the hardware restriction, which leads to poor performance for byte-aligned data transfer.

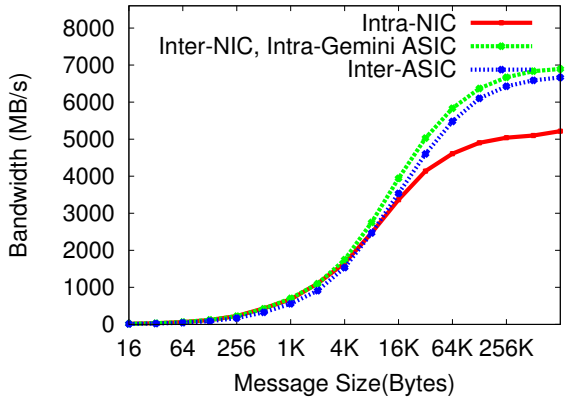

Fig. 11. Get Bandwith with Different Configurations

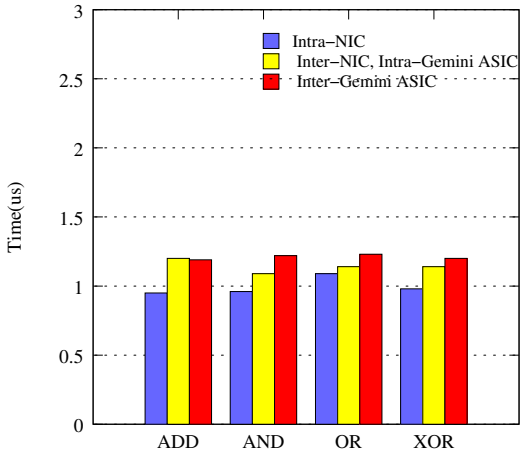

Fig. 12. No-Fetch, Atomic Operations

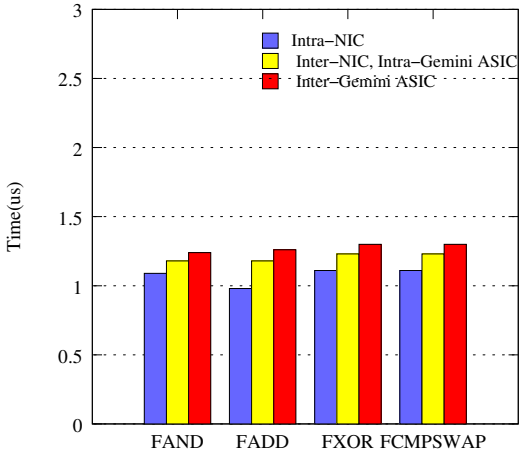

Fig. 13. Fetch, Atomic Operations

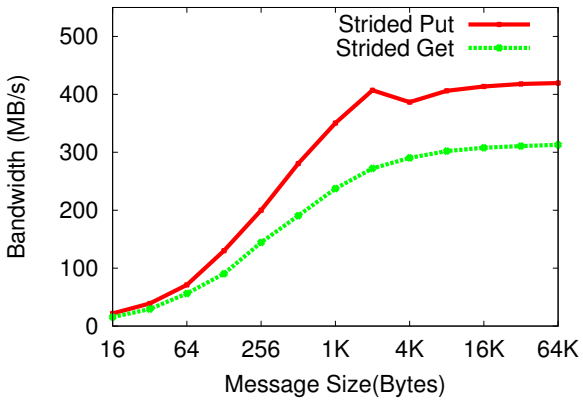

Fig. 14. Strided Bandwidth Performance

For double-word, quad-word and double-word-alignment, the peak bandwidth achieved is 6919 MB/s, as shown in the Figure 10.

Figure 11 shows the bandwidth of get communication primitive with different configurations. For large messages, we observe similar trends as the put communication primitive. The Intra-NIC configuration achieves a peak bandwidth of 5215 MB/s, due the simultaneous use of bi-directional HyperTransport. The Inter-NIC configurations achieve a peak bandwidth of 6893 MB/s.

4) Atomic Memory Operations: Figures 12 and 13 show the performance of the atomic memory operations (AMOs) with different configurations. Cray Gemini supports quad-word atomic operations only. The AMOs which do not return the original value (ADD, AND, OR and XOR) have a latency of 1.24μs for Inter-NIC, Inter-ASIC configuration. The atomic operations which return the original value (FAND, FADD, FXOR, FCOMPSWP) have a latency of 1.25μs for Inter-NIC, Inter-ASIC configuration. The AMOs are non-coherent with the HyperTransport(s) which connect the individual Magny-Cours sockets.

D. Emulating Application Specific Communication Pattern Microbenchmarks

In this section, we design micro-benchmarks to emulate the communication patterns incurred by scientific applications. These micro-benchmarks study the performance of uniformly non-contiguous (strided) operations used by applications in computational chemistry [23], and sub-surface modeling [22]. A micro-benchmark to design accumulate operation using AMOs is also presented, which emulates the native implementation of accumulate operation with help from the asynchronous agent . To study the scalability of the Gemini interconnect, we design two micro-benchmarks motivated from application domains - use of AMOs for dynamic load balancing counters [23] and study the impact of routing options in the Cray Gemini Interconnect for shift communication pattern. The shift communication operation is frequently observed in many PGAS and MPI applications in forms of collective communication primitives and kernels including Fourier Transform.

Figures 14 and 15 show the bandwidth and latency observed for two processes with Inter-ASIC configuration. For the test,

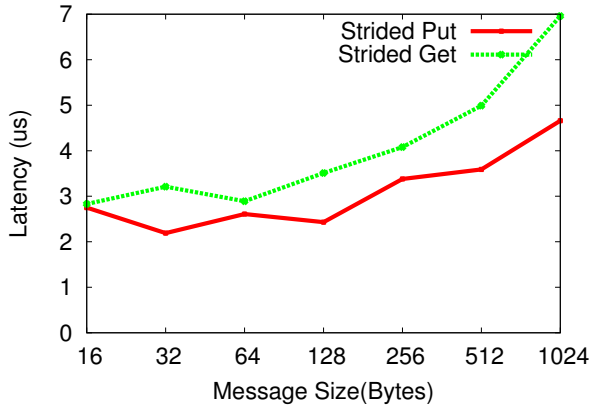

Fig. 15. Strided Latency Performance

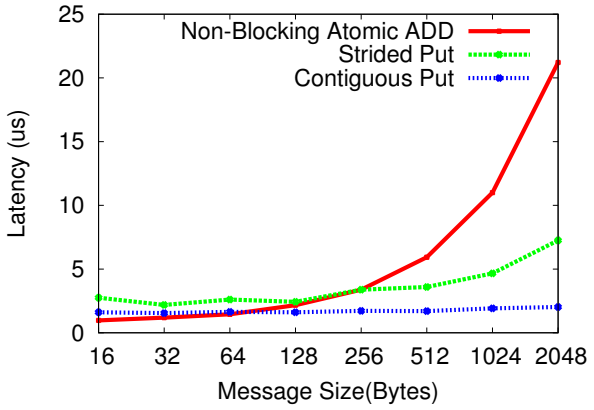

Fig. 16. Designing Accumulates using ADD Operation

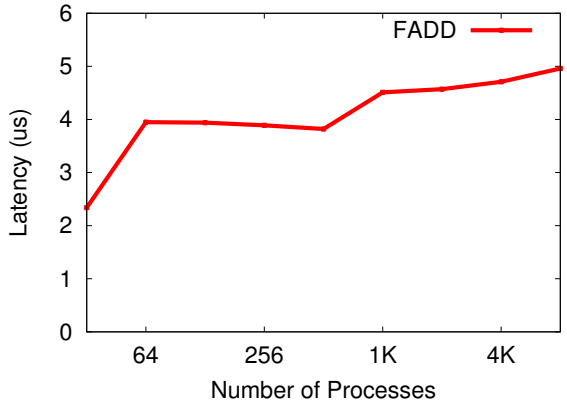

Fig. 17. Scalability Performance of Atomic FADD

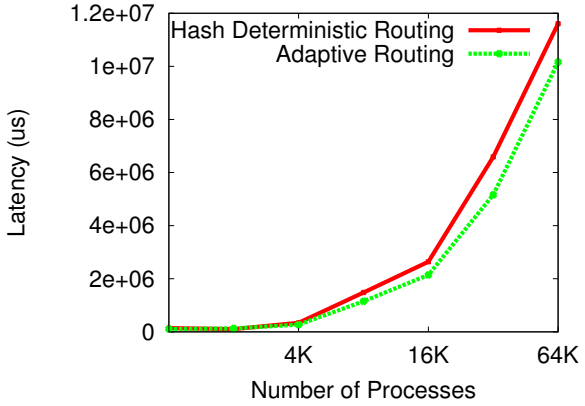

Fig. 18. Performance of Shift Communication Pattern

a stride of 128 bytes was used and the contiguous data size was increased, as shown on the horizontal axis. Strided communication is heavily used by NWChem [23], where a block of data in multi-dimensional global address space is requested from a task [5]. To optimize the performance of this datatype, typically a copy-based protocol with is used with help from a remote asynchronous progress agent [9]. While Gemini provides support for native strided communication, the observed bandwidth is much lower than the peak bandwidth observed with contiguous datatypes in the previous section. The peak bandwidth observed is 312 MB/s for get communication primitive, and 420 MB/s for put communication primitive.

Figure 16 shows the latency of an accumulate operation for contiguous data designed using AMOs (ADD). The accumulate operation is designed using non-blocking ADD AMOs. We compare the performance to contiguous and strided put with the performance of accumulate operation. For one-sided communication runtime systems like ARMCI [9], the accumulate operation is implemented using a remote asynchronous agent. The latency of an accumulate operation for contiguous data is bounded from below by the performance of the contiguous put communication primitive. We observe that for up to 128bytes, the latency of contiguous put and an accumulate operation using AMOs is similar. With increasing message size, the latency of non-blocking AMO ADD increases significantly, while the of contiguous put increases slightly.

Figure 17 shows the performance of atomic FADD with increasing number of processes. The test is designed to study the efficacy of Cray Gemini Interconnect AMOs for a use-case of dynamic load balancing counter. The hardware atomics are cached on the Gemini NIC (non-coherent with the system bus), resulting in excellent scalable performance. For applications in computational chemistry [23], the excellent scalability of atomic operations is critical. The startup and completion phases of dynamic load balancing can be effectively implemented with hardware AMOs in Gemini.

Figure 18 shows the performance evaluation of the shift-communication pattern benchmark using up to 65536 processes [24]. The shift-communication benchamrk is a representative of many collective communication operations in MPI [3], [4]. The shift communication pattern is also used to design transpose operation in PGAS models such as UPC [6] and GA [5]. The Gemini Interconnect supports multiple routing algorithms, including hash-deterministic, and adaptive. The Gemini Interconnect uses dimension-order routing for hash-deterministic and adaptive routes, but provides complete

flexibility in using multiple links within each dimension for adaptive routing. The routing algorithm is not fully adaptive, but provides better utilization of the lightly loaded links [1]. We observe that the adaptive routing outperforms the hash-deterministic routing by 12% for 32768 and 65536 processes, respectively. A super-linear increase in latency is observed from 16384 to 32768 processes, since the number of 'y' links used in the shift communication increase significantly. The 'y' links support only half the peak bandwidth of 'x' and 'z' links, resulting in super-linear latency increase.

IV. Acknowledgments

The PNNL part of the research described in this paper was conducted under the Laboratory Directed Research and Development (LDRD) Program, a multiprogram national laboratory operated by Battelle for the U.S. Department of Energy under Contract DE-AC05-76RL01830.

The Cray part of the research is supported by the Defense Advanced Research Projects Agency (DARPA) under its Agreement No. HR0011-07-9-0001. Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the DARPA.

V. Conclusions and Future Work

The primary objective of our work is to design micro-benchmarks motivated from application case studies using the Cray DMAPP [10] userspace library specifically designed for PGAS models and one-sided communication runtime systems. The intended outcome of this study is to provide designers of one-sided communication runtime systems with an in-depth performance analysis of performance parameters with the Cray Gemini Interconnect. To meet this objective, our study includes designing micro-benchmarks for one-sided communication primitives (put, get, and atomic memory operations) with various inter-process configurations, handling contiguous and non-contiguous data-types, simultaneous request of message transfers, scalability analysis of atomic memory operations for dynamic load balancing and shift communication operation with different routing options. The Gemini Interconnect can achieve a peak bandwidth of 6911 MB/s and a latency of 1μs for get communication primitive. Scalability tests for atomic memory operations and shift communication operation up to 65536 processes shows the efficacy of the Cray Gemini Interconnect.

We plan to use this study to design efficient communication protocols for one-sided communication runtime systems such as ARMCI [9] and MPI-RMA [4]. We also plan to study the performance of these communication runtime systems with applications in computational chemistry [23] and sub-surface modeling [22].

References

[1] R. Alverson, D. Roweth, and L. Kaplan, "The Gemini System Interconnect," in *International Symposium on High Performance Interconnects 2010*, aug. 2010, pp. 83 –87.

[2] R. Brightwell, K. T. Pedretti, K. D. Underwood, and T. Hudson, "SeaStar Interconnect: Balanced Bandwidth for Scalable Performance," *IEEE Micro*, vol. 26, no. 3, pp. 41–57, 2006.

[3] W. Gropp, E. Lusk, N. Doss, and A. Skjellum, "A High-Performance, Portable Implementation of the MPI Message Passing Interface Standard," *Journal of Parallel Computing*, vol. 22, no. 6, pp. 789–828, 1996.

[4] A. Geist, W. Gropp, S. Huss-Lederman, A. Lumsdaine, E. L. Lusk, W. Saphir, T. Skjellum, and M. Snir, "MPI-2: Extending the message-passing interface," in *Euro-Par, Vol. I*, 1996, pp. 128–135.

[5] J. Nieplocha, R. J. Harrison, and R. J. Littlefield, "Global Arrays: A Nonuniform Memory Access Programming Model for High-Performance Computers," *Journal of Supercomputing*, vol. 10, no. 2, pp. 169–189, 1996.

[6] P. Husbands, C. Iancu, and K. A. Yelick, "A Performance Analysis of the Berkeley UPC Compiler," in *International Conference on Supercomputing*, 2003, pp. 63–73.

[7] R. W. Numrich and J. Reid, "Co-array Fortran for parallel programming," *SIGPLAN Fortran Forum*, vol. 17, pp. 1–31, August 1998.

[8] B. Chamberlain, D. Callahan, and H. Zima, "Parallel Programmability and the Chapel Language," *International Journal on High Performance Computing Applications*, vol. 21, no. 3, pp. 291–312, 2007.

[9] J. Nieplocha and B. Carpenter, "ARMCI: A Portable Remote Memory Copy Library for Distributed Array Libraries and Compiler Run-Time Systems," in *Lecture Notes in Computer Science*. Springer-Verlag, 1999, pp. 533–546.

[10] M. Bruggencate and D. Roweth, "DMAPP: An API for One-Sided Programming Model on Baker Systems," *Cray Users Group (CUG)*, aug. 2010.

[11] InfiniBand Trade Association, "InfiniBand Architecture Specification, Release 1.2," October 2004.

[12] N.J. Boden, D. Cohen, R.E. Felderman, A.E. Kulawik, C.L. Seitz, J.N. Seizovic, and W. Su, "Myrinet: A Gigabit-per-second Local Area Network," *IEEE Micro*, vol. 15, no. 1, pp. 29–36, 1995.

[13] F. Petrini, W. Feng, A. Hoisie, S. Coll, and E. Frachtenberg, "The Quadrics Network: High-Performance Clustering Technology," *IEEE Micro*, vol. 22, no. 1, pp. 46–57, 2002.

[14] J. Liu, B. Chandrasekaran, W. Yu, J. Wu, D. Buntinas, S. P. Kini, D. K. Panda, and P. Wyckoff, "Microbenchmark Performance Comparison of High-Speed Cluster Interconnects." *IEEE Micro*, vol. 24, no. 1, pp. 42–51, 2004.

[15] D. H. Bailey, E. Barszcz, J. T. Barton, D. S. Browning, R. L. Carter, D. Dagum, R. A. Fatoohi, P. O. Frederickson, T. A. Lasinski, R. S. Schreiber, H. D. Simon, V. Venkatakrishnan, and S. K. Weeratunga, "The NAS Parallel Benchmarks," in *The International Journal of Supercomputer Applications*, no. 3, 1991, pp. 63–73.

[16] R. K. Govindaraju, P. H. Hochschild, D. G. Grice, K. J. Gildea, R. Blackmore, C. A. Bender, C. Kim, P. Chaudhary, J. Goscinski, J. Herring, S. Martin, and J. Houston, "Architecture and Early Performance of the New IBM HPS Fabric and Adapter," in *International Conference on High Performance Computing*, 2004, pp. 156–165.

[17] R. Sivaram, R. K. Govindaraju, P. H. Hochschild, R. Blackmore, and P. Chaudhary, "Breaking the Connection: RDMA Deconstructed," in *International Symposium on High-Performance Interconnects*, 2005, pp. 36–42.

[18] S. Sur, A. Vishnu, H.-W. Jin, W. Huang, and D. K. Panda, "Can memory-less network adapters benefit next-generation infiniband systems?." in *Hot Interconnects*, 2005, pp. 45–50.

[19] A. Vishnu, B. Benton, and D. K. Panda, "High Performance MPI on IBM 12x InfiniBand Architecture," in *International Workshop on High-Level Parallel Programming Models and Supportive Environments, held in conjunction with IPDPS '07 (HIPS'07)*, 2007.

[20] A. Vishnu, A. R. Mamidala, H.-W. Jin, and D. K. Panda, "Performance Modeling of Subnet Management on Fat Tree InfiniBand Networks using OpenSM." in *Proceedings of First International Workshop on System Management Techniques, Processes, and Services, held in conjunction with IPDPS'07*, 2005.

[21] "NERSC Hopper Phase II System," http://www.nersc.gov/users/computational-systems/hopper.

[22] Subsurface Transport over Multiple Phases, "STOMP," http://stomp.pnl.gov/.

[23] R. A. Kendall, E. Aprà, D. E. Bernholdt, E. J. Bylaska, M. Dupuis, G. I. Fann, R. J. Harrison, J. Ju, J. A. Nichols, J. Nieplocha, T. P. Straatsma, T. L. Windus, and A. T. Wong, "High Performance Computational Chemistry: An Overview of NWChem, A Distributed Parallel

Application," *Computer Physics Communications*, vol. 128, no. 1-2, pp. 260–283, June 2000.

[24] A. Vishnu, M. J. Koop, A. Moody, A. R. Mamidala, S. Narravula, and D. K. Panda, "Hot-Spot Avoidance With Multi-Pathing Over InfiniBand: An MPI Perspective," in *Cluster Computing and Grid*, 2007, pp. 479–486.