A preliminary evaluation of the hardware acceleration of the Cray Gemini Interconnect for PGAS languages and a comparison with MPI

Hongzhang Shan, Nicholas J. Wright, John Shalf and Katherine Yelick
CRD and NERSC
Lawrence Berkeley National Laboratory, Berkeley, CA 94720
{hshan, njwright, jshalf, kayelick}@lbl.gov

Marcus Wagner and Nathan Wichmann
Cray Inc. 380 Jackson Street, Suite 210, St. Paul, MN 55101
marcus, wichmann@cray.com

ABSTRACT

The Gemini interconnect on the Cray XE6 platform provides for lightweight remote direct memory access (RDMA) between nodes, which is useful for implementing partitioned global address space languages like UPC and Co-Array Fortran. In this paper, we perform a study of Gemini performance using a set of communication microbenchmarks and compare the performance of one-sided communication in PGAS languages with two-sided MPI. Our results demonstrate the performance benefits of the PGAS model on Gemini hardware, showing in what circumstances and by how much one-sided communication outperforms two-sided in terms of messaging rate, aggregate bandwidth, and computation and communication overlap capability. For example, for 8-byte and 2KB messages the one-sided messaging rate is 5 and 10 times greater respectively than the two-sided one. The study also reveals important information about how to optimize one-sided Gemini communication.

Categories and Subject Descriptors

B.8.2 [**Performance Analysis and Design Aids**]; D.1.3 [**Concurrent Programming**]

General Terms

Languages, Performance

Keywords

PGAS, MPI, Messaging Rate, CAF, Performance

1. INTRODUCTION

The classic parallel programming model, MPI, faces several new challenges on petaflop computing platforms, which are dominated by multicore-node architectures [2, 4]. To address these challenges, researchers are starting to investigate other programming models to understand whether they could replace or be used in combination with MPI. Among these studied programming models, the Partitioned Global Address Space (PGAS) family of languages show great promise as the near-term alternative to MPI. Co-Array Fortran (CAF) [3] and Unified Parallel C (UPC) [1] are two representative examples of such languages.

Hopper is a 1.28 PF peak Cray XE6 computing platform recently installed at NERSC. The defining feature of this platform is the custom interconnect, called Gemini, which provides a hardware accelerated global address space and allows remote direct memory access (RDMA) from any node to any other in the system. In this work, we will investigate what the effect of this special Gemini hardware support for global address space and one-sided messaging is upon the performance of the PGAS languages.

2. PERFORMANCE RESULTS

2.1 Messaging Rate

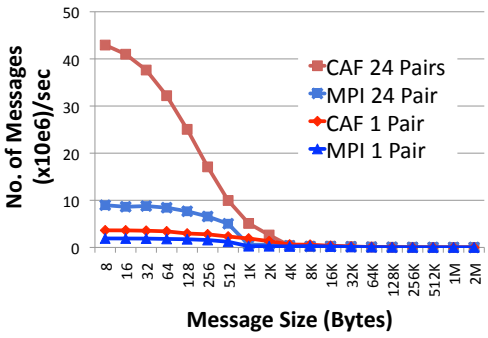

Figure 1: The messaging rate for MPI and CAF measured using 1 and 24 communicating pairs per node.

Our results, shown in Fig. 1, show that in the bandwidth limit, with large messages, MPI and PGAS performance is identical. For medium-sized and small messages, the lower overhead of the single-sided PGAS messaging allows CAF to deliver up to eight times more messages per second.

2.2 Computation / Communication Overlap

We develop an independent micro-benchmark to measure the capability of different languages to overlap communication and computation. A metric called overlapped_fraction

PMBS'11, November 13th, 2011, Seattle, Washington, USA.
ACM 978-1-4503-1102-1/11/11.

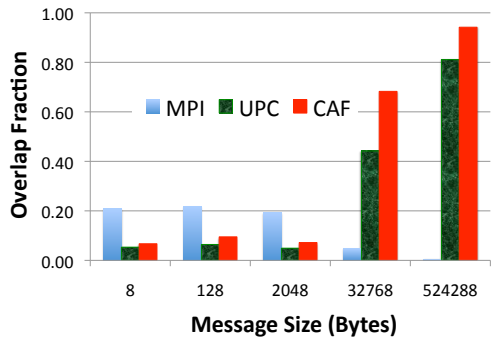

Figure 2: The overlap capability of MPI, UPC, and CAF.

is computed using following formula:

$$overlapped\_fraction = 1 - \left(\frac{T_{TotalRunningTime} - max(T_{Comp}, T_{Comm})}{min(T_{Comp}, T_{Comm})} \right)$$

where T_{comp} is the computation time and T_{comm} is the communication time. In the case that the runtime is equal to the maximum of the separate measurements of computation and communication the overlap is perfect. This fraction represents the amount of work that it was not possible to overlap. The results in Fig. 2 show that MPI exhibits some overlap capability for small messages and almost no overlap for large ones. On the contrary, CAF and UPC demonstrate excellent overlap capability for large messages.

2.3 NAS FT

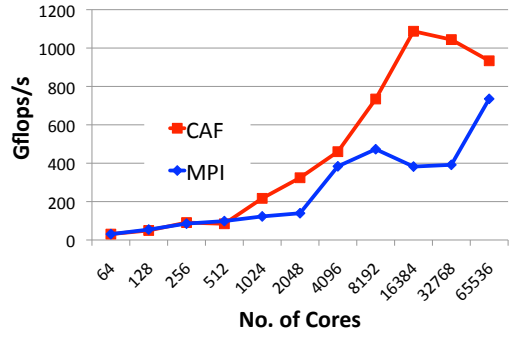

Figure 3: The performance of NAS FT for Class B for the CAF and MPI versions.

For the CAF implementation, the MPI_Alltoall has been replaced by one-sided get operations. CAF exhibits much better performance and better scalability up to 16K cores. The CAF performance is about 2.6 times better than the MPI result when 16K cores are used.

2.4 Stream

We developed a version of the STREAM benchmark using CAF. Fig. 4 show that the RDMA operations executed by the Gemini allow very high STREAM copy bandwidths to be achieved. However, for for the remaining operations (Scale, Add and Triad.), their performances are significantly lower using the naive implementation. In order to improve their performance, we used larger message sizes and pipelined the messages to overlap the communication with computation. The performances of the optimized version are over three times better than those of the naive version.

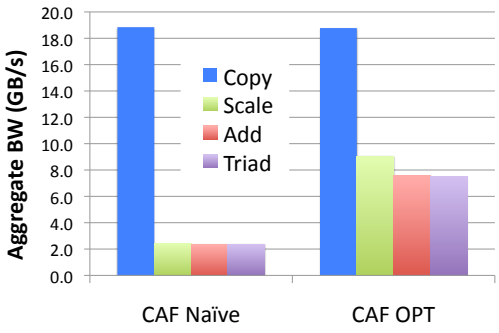

Figure 4: The STREAM bandwidth between two nodes using 1 pair.

3. SUMMARY AND CONCLUSIONS

In this work we evaluated the performance of PGAS languages on a Cray XE6 high-performance computing platform for which the Gemini interconnect provides direct support for a globally addressable memory and hardware accelerated one-sided messaging. We examined the performance in terms of bandwidth, message rate, and capability to overlap computation with communication. The results demonstrated that with this special hardware acceleration, PGAS languages can outperform MPI, especially for messages a few KB in size, and therefore provide a viable alternative. However, they also show that simply swapping MPI calls for equivalent PGAS constructs may not necessarily be the optimal path forward for achieving good performance with PGAS, as the performance in the bandwidth limit is identical to that of MPI. Codes may need to be modified to send smaller messages more frequently than one would with MPI in order to achieve the greatest benefit from using PGAS languages, Our future work will focus on converting existent scientific applications into PGAS codes and study their performance on Hopper. The full paper can be found in [5].

4. REFERENCES

[1] Carlson, W. W., Draper, J. M., Culler, D. E., Yelick, K., Brooks, E., and Warren, K. Introduction to UPC and language specification. In *Tech. Rep. CCS-TR-99-157 May* (May 1999).

[2] Geist, A. Sustained petascale: The next MPI challenge. In *EuroPVMMPI* (October 2007).

[3] Numrich, R. W., and Reid, J. Co-array Fortran for parallel programming. In *ACM SIGPLAN Fortran Forum, vol. 17, no. 2, pp. 1 31* (August 1998).

[4] Challenges for the message passing interface in the petaflops era. `www.cs.uiuc.edu/homes/wgropp/bib/talks/tdata/2007/mpifuture-uiuc.pdf`.

[5] Shan, H., Wright, N., Shalf, J., Yelick, K., Wagner, M. and Wichmann, N. A preliminary evaluation of the hardware acceleration of the Cray Gemini Interconnect for PGAS languages and a comparison with MPI. *SIGMETRICS Performance Evaluation Review 40(2)* (2012)