SAND2011-4617C

The Impact of Injection Bandwidth Performance on Application Scalability

Kevin T. Pedretti, Ron Brightwell, Doug Doerfler, K. Scott Hemmert, and James H. Laros III

Sandia National Laboratories*
Albuquerque, NM, 87185, USA
{ktpedre, rbbrigh, dwdoerf, kshemme, jhlaros}@sandia.gov

Abstract. Future exascale systems are expected to have significantly reduced network bandwidth relative to computational performance than current systems. Clearly, this will impact bandwidth-intensive applications, so it is important to gain insight into the magnitude of the negative impact on performance and scalability to help identify mitigation strategies. In this paper, we show how current systems can be configured to emulate the expected imbalance of future systems. We demonstrate this approach by reducing the network injection bandwidth performance of a 160-node, 1920-core Cray XT5 system and analyze the performance and scalability of a suite of MPI benchmarks and applications.

Keywords: bandwidth configurability, benchmarking, exascale co-design

1 Introduction

An individual compute node's network injection bandwidth to computational performance ratio is an important metric used when designing and comparing large-scale parallel computers. The ratio must be high enough to support the breadth of applications intended to run on a system, but not so high that bandwidth, which is expensive, both in terms of dollars and power, goes unused and is wasted. Determining the right design point is a significant challenge.

Historically, a rule of thumb has been that a "balanced" system should support one byte per second of network I/O injection bandwidth for each floating point operation per second (FLOPS) computed. Such a system would be generally applicable and support a broad range of applications. Unfortunately, a variety of factors has resulted in system balance ratios decreasing over time and the trend is predicted to accelerate. A recent DARPA-sponsored study [10] concluded that even with optimistic technology scaling assumptions, realizing the next milestone of exa-FLOPS capable parallel computers will require that system balance targets be reduced by more than an order of magnitude from what

* Sandia National Laboratories is a multi-program laboratory managed and operated by Sandia Corporation, a wholly owned subsidiary of Lockheed Martin company, for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-AC04-94AL85000.

is considered acceptable today. As a point of comparison, Table 1 lists the network injection to compute balance ratios of two prior-generation systems along side the hypothetical exascale system forecast in the report.

Table 1: Comparison of system balance over time.

System	ASCI Red	Jaguar-PF	Exascale Prediction
Year	1997	2009	2015
System Peak TFLOPS	3.15	2,332	1,029,901
Node Inj. Bandwidth (GB/s)	0.400	2.2	3.68
Node Peak GFLOPS	0.666	124.8	4,600
Net/Compute Balance Ratio	**0.6**	**0.018**	**0.0008**

With this hardware prediction in view, it is important to gain more insight into how this change in system balance will affect existing application software, the design of future applications, and, ultimately, provide feedback to influence the design requirements of future exascale computers as part of the co-design process [2]. The approach we are currently pursuing is to leverage the extensive configurability of existing supercomputer platforms, specifically the Cray XT and XE architectures, to emulate the expected behavior of future systems. The resulting experimental platform is then used to run important production applications with realistic input problems to observe how they respond to the change in system balance. Our initial efforts have focused on application sensitivity to network injection bandwidth, since it plays a critical role in determining how the processor and network interface are connected and the level of integration required. It is one of the earliest hardware design decisions that has arisen.

The remainder of this paper is organized as follows. Section 2 describes the relevant aspects of our test platform. Our approach is then described in Sect. 3, followed by results in Sect. 4. Section 5 discusses work that is related to our experimental approach. Finally, conclusions are given in Sect. 6.

2 Test Platform

All of our injection bandwidth experiments to date have been performed on the "XTP" Cray XT5 system at Sandia. This system consists of 160 12-core compute nodes, each with 32 GB of memory and a Cray SeaStar [4] network interface. Each node connects to its SeaStar via a point-to-point HyperTransport link, and each SeaStar then connects to the node's six nearest neighbors via proprietary Cray point-to-point network links. The overall network topology forms a 3-D torus.

The XTP system is considerably smaller than the systems we ultimately seek to target, but is enlightening nonetheless. Our work on XTP has validated the soundness of our approach and produced a set of baseline scaling results for our

target applications. Moving on to larger-scale Cray XT and XE systems can leverage the XTP work directly, and is expected to be straightforward.

3 Approach

In order to reduce a compute node's network injection bandwidth, we are leveraging source-level access to the Cray XT and XE *coldstart* boostrap infrastructure, which serves the same purpose as a traditional BIOS. A separate instance of coldstart runs on each compute node, but all instances are the same and configure each of the compute nodes identically.

Very early in the power-on sequence, coldstart initializes the HyperTransport (HT) point-to-point link that connects the Opteron host processor to the SeaStar network interface. The speed of this link determines a compute node's injection bandwidth capability into the network fabric. The Opteron processor supports several HT links, but we only degrade the single link that connects to the SeaStar. All other HT links continue to operate at full speed.

By default, coldstart configures the Opteron-to-SeaStar HT link for the maximum speed supported by both the Opteron and SeaStar. We modify this negotiation process to select one of the alternative speeds listed in Table 2, which is the cross-product of the settings supported by the Opteron and the settings supported by the SeaStar. The basic configuration parameters for the HT link are width, either 8-bits or 16-bits wide, and operating frequency, 200 MHz, 400 MHz, or 800 MHz. This results in the theoretical peak uni-directional HT link bandwidths listed in the table. However, HT and SeaStar protocol overheads limit the achievable injection bandwidth to far less than the theoretical peak.

Table 2: Possible Opteron-to-SeaStar HT link bandwidth configurations.

Link Frequency & Width	8-bit	16-bit
200 MHz	400 MB/s	800 MB/s
400 MHz	800 MB/s	1600 MB/s
800 MHz	1600 MB/s	3200 MB/s

3.1 Benchmarks and Applications

Our analysis includes several benchmarks and applications. We briefly describe them here.

Communication Micro-Benchmarks: We use a standard MPI ping-pong micro-benchmark to measure latency and bandwidth between two nodes over the network and an MPI streaming bandwidth benchmark to measure an entire node's injection bandwidth using all cores simultaneously. The ping-pong bandwidth benchmark is part of the Intel MPI Benchmark Suite and the streaming bandwidth test is from Ohio State University. These benchmarks are primarily used to verify that the injection bandwidth degradation performs as expected.

HPC Challenge (HPCC) Benchmarks: The HPC Challenge [12] benchmarks are commonly used to evaluate the performance of supercomputers. Although they are not production applications, they have been constructed to represent a disjoint set of application characteristics common in scientific computing. We selected four of these benchmarks: HPL, RandomAccess, PTRANS, and FFT. The High Performance Linpack (HPL) benchmark measures the floating point performance obtained when solving a dense linear system of equations. RandomAccess measures the rate of random updates to a large pool of 64-byte integer values. The `RA_SANDIA_NOPT` version of the algorithm was used, which results in communication between processes consisting primarily of 8 KB messages. The PTRANS benchmark performs a parallel matrix transpose and is network bandwidth intensive. FFT measures the floating-point performance obtained for a double-precision, complex, one-dimensional discrete Fourier transform (DFT).

Applications: Four applications were evaluated for their sensitivity to network injection bandwidth: CTH, Sage, xNOBEL, and Charon. These codes were selected due to their importance to the Advanced Simulation and Computing (ASC) program within the U.S. Department of Energy and because of their ability to scale to very large-scale machines. The combination of these particular applications also comprise a disjoint set of communication patterns that are representative of a large majority of other ASC applications. Realistic input problems were used in a weak scaling fashion, resulting in computational work per core being held roughly constant regardless of scale. We briefly describe each of these application below.

CTH [5] is a multi-material, large deformation, strong shock wave, solid mechanics code. The test problem used for this study was a 3D shaped charge simulation discretized to a rectangular mesh. In this configuration, inter-process communication consists primarily of large, multi-megabyte sized messages communicated among neighboring nodes. CTH is therefore limited primarily by point-to-point network bandwidth.

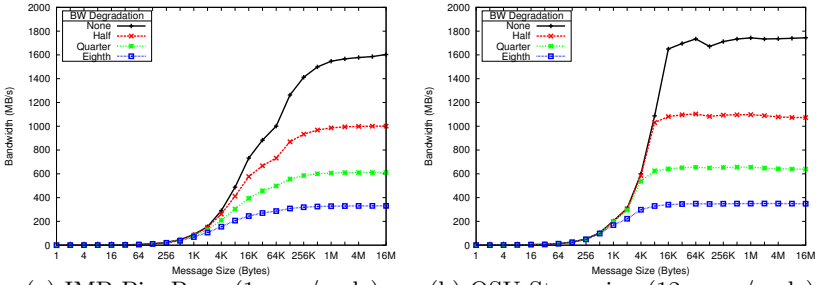
SAGE [14] is a multidimensional, multi-material Eulerian hydrodynamics code. SAGE inter-process communication consists primarily of a bulk-synchronous gather/scatter abstraction, which aggregates messages into large doubly-indexed arrays. Message sizes are generally in the hundreds of kilobytes to one megabyte range.

xNOBEL [7] is a one-,two-,or three-dimensional, multi-material Eulerian hydrodynamics code. It was developed for solving a variety of high-deformation flow of materials problems, with the distinguishing characteristic of being able to model high explosives. The problem used for this study was a shape charge simulation in two dimensions. Network communication consists of relatively small messages in the tens of bytes to hundreds of kilobytes.

Charon [11] is a semiconductor device simulation code. The problem used for this study is a 2D steady-state drift-diffusion simulation for a bipolar junction transistor. Charon is sensitive to small message latency for point-to-point and global reduction operations.

4 Results

Measured network bandwidth between two adjacent nodes is plotted in Fig. 1 for varying levels of injection bandwidth degradation. The labels in this figure and all others in this paper use the following convention: *None* = (800 MHz, 16 bit), *Half* = (400 MHz, 16 bit), *Quarter* = (200 MHz, 16 bit), and *Eighth* = (200 MHz, 8 bit). The bandwidth curves for the streaming test (Fig. 1b) ramp up much more quickly and reach slightly higher asymptotic levels than the ping-pong test (Fig. 1a). Applications that send few message at a time will behave more like what is observed for the ping-pong test, while applications that send many messages at a time will behave more like the streaming test. Small message latency was also measured for each configuration and found to be approximately the same as with no degradation (< 1.0 μs).



(a) IMB PingPong (1 core/node) (b) OSU Streaming (12 cores/node)

Fig. 1: MPI micro-benchmark uni-directional bandwidth measurements.

The resulting network injection bandwidth to compute balance ratios achieved using our test system are listed in Table 3. These ratios were calculated using the asymptotic maximum node-level bandwidths measured in Fig. 1b (in GB/s) divided by the peak 115.2 GFLOPS capability of each compute node. The lowest achieved balance ratio of 0.0030 is a factor of five worse than the full-speed baseline, but still a factor of 3.75 higher than the exascale system prediction listed in Table 1. Nevertheless, the test system exhibits a significant decrease in network-to-compute balance that can be leveraged to characterize application sensitivity to injection bandwidth.

Table 3: Achieved network injection bandwidth to compute balance ratios using experimental platform.

Inj. BW Degradation	Net/Compute Balance Ratio	Factor Worse
None	0.0151	–
Half	0.0094	1.6
Quarter	0.0056	2.7
Eighth	0.0030	5.0

HPCC and application results are presented as performance degradation relative to full injection bandwidth, and are therefore unit-less. For example, a value of 1.5 indicates that the performance measured with the degraded injection bandwidth configuration was 1.5 times slower than with no degradation. In order to obtain the highest level of imbalance possible in practice, we used all twelve processor cores per compute node, running one MPI process per core. Therefore, the number of nodes used for each data-point in Figs. 2 and 3 is obtained by dividing the x-axis label by twelve and rounding up to the nearest integer.

Results for the HPCC benchmarks are shown in Fig. 2. While network performance is important, its impact on HPL performance is secondary compared to computational capability, which we do not modify. As shown in Fig. 2a, our experimental results confirm this, showing relatively little performance degradation for HPL (note the reduced y-axis range compared to the other plots).

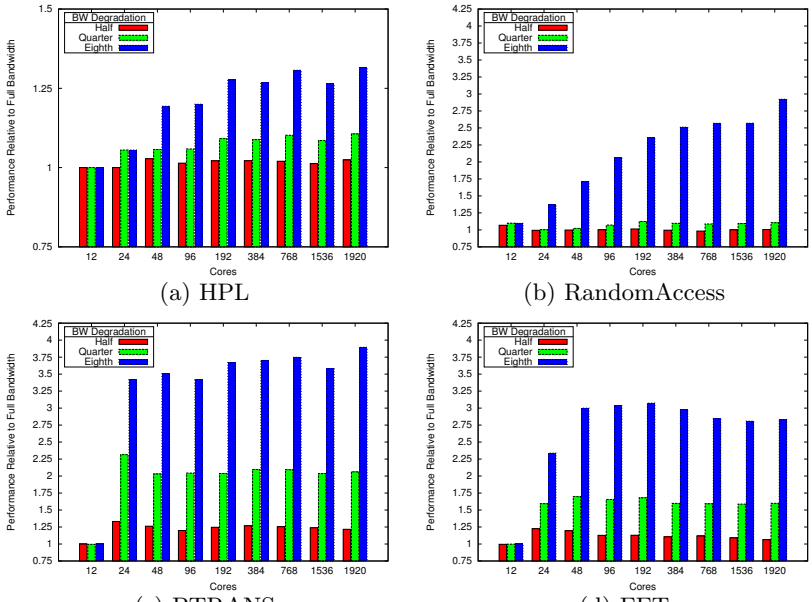
The results shown in Fig. 2b for RandomAccess were surprising to us. Half and quarter configurations result in almost no performance penalty, while eighth results in much larger penalties that appear to grow with scale. Our current theory is that 8 KB `MPI_Sendrecv` operations are at a point on the bandwidth curve that has roughly the same bandwidth for none, half, and quarter configurations, and much less for eighth. Such an effect can be seen for 4 KB messages in Fig. 1b, and we hypothesize that something similar is happening here for 8 KB messages. Further investigation at larger-scale is required.

As expected for a network bandwidth bound benchmark like PTRANS, the effect of injection bandwidth degradation shown in Fig. 2c is dramatic. The degradation factors relative to full injection bandwidth for half, quarter, and eighth configurations are approximately 1.25, 2, and 3.8, respectively, and the penalties do not appear to grow with scale. These degradation factors are reasonably close to the theoretical maximums listed in Table 3. FFT demonstrates behavior similar to PTRANS, but with lower degradation magnitudes.

Results for the applications evaluated are shown in Fig. 3. CTH is observed to be relatively insensitive to half injection bandwidth, moderately sensitive to quarter bandwidth, and significantly sensitive to eighth bandwidth (Fig. 3a). The performance degradations appear to be increasing with scale, but we believe this is an artifact of the communication to computation ratio increasing logarithmically with scale, as observed for CTH in [6]. Larger-scale experiments are needed to confirm this.

SAGE demonstrates a performance degradation profile similar to CTH, but of a lesser degree (Fig. 3b). SAGE's message sizes are generally not as large as CTH's, and are likely falling at a point on the message size vs. bandwidth curve where the differences between the injection bandwidth configurations are not as pronounced. As with CTH, larger-scale experiments are required to determine if performance degradation continues to increase.

xNOBEL demonstrates different scaling behavior than was observed for CTH and SAGE (Fig. 3c). Half and quarter injection bandwidth configurations have very little impact on performance. However, with one-eighth speed injection



(a) HPL (b) RandomAccess

(c) PTRANS (d) FFT

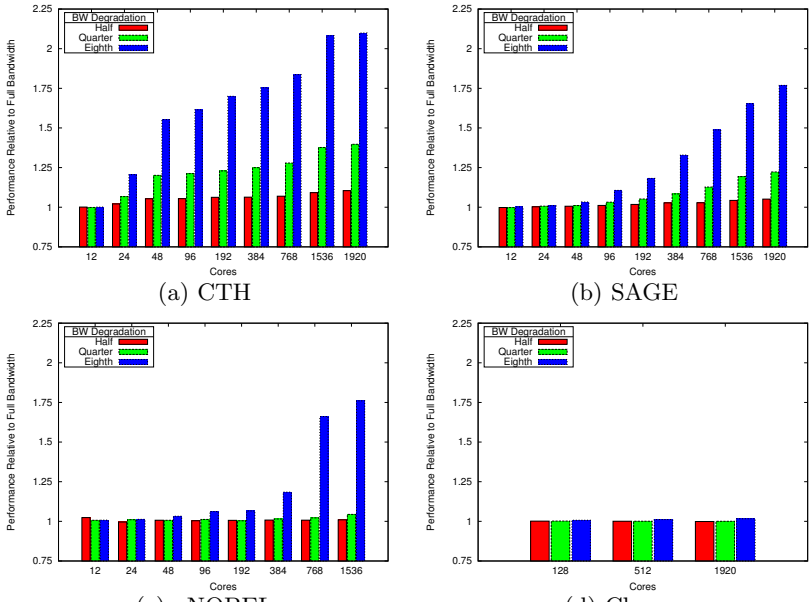
Fig. 2: HPCC Results

bandwidth, significant performance degradation is observed but only at the largest scales tested. We believe this to be caused by xNOBEL suddenly losing the ability to significantly overlap communication and computation due to the severely degraded injection bandwidth. If true, this demonstrates the importance of providing sufficient network performance so that well-designed applications can maximize their ability to overlap communication and computation.

As expected for a latency bound application, Charon is unaffected by injection bandwidth degradation (Fig. 3d). For the test problem used, the average message size is less than 1 kilobyte, which, according to Fig. 1, results in essentially identical performance no matter what injection bandwidth configuration is used. Given Charon's insensitivity to injection bandwidth, it would clearly be advantageous to save power by using a low-bandwidth, low-latency interconnect.

5 Related Work

Parameterized analytic performance models have been developed for a number of large-scale parallel applications [1, 9] and been shown to closely track experimental measurements. While clearly worthwhile, a downside to analytic modeling is that it is time consuming, requiring expert-level knowledge of a given application, and is necessarily application specific. Our in-situ experimental-based approach can be complementary to analytic modeling by accelerating the application understanding process and by assisting in model validation.



(a) CTH (b) SAGE

(c) xNOBEL (d) Charon

Fig. 3: Application Results

Simulation-based approaches are highly configurable and flexible. The downside of this approach is often performance – slowdowns of 1000x or more are common, and increase with the simulation's fidelity. Model-based simulators can be much faster, but make simplifying assumptions. In contrast, our approach allows full applications to be evaluated in real-time. However, a drawback of our approach is that it can generally only slow-down the performance parameter being studied. The Structural Simulation Toolkit (SST) [13] is an example of a simulation platform targeted at simulating large-scale parallel computers. Our approach could be used to assist in validating SST component models and other simulation tools.

Finally, experimental methods such as our approach use existing systems and real applications to evaluate the impact of the parameter under study. Software-based de-tuning was used in [6] to characterize the performance penalty with scale of various levels of operating system interference. Recently, results from this study were reproduced via simulation [8], demonstrating the important relationship between simulation and experiment. Another software-based experimental approach is described in [3], where a special MPI library was used to degrade network performance. Our approach is similar, but uses direct hardware manipulation instead of software techniques. This eliminates the potential of introducing artificial software-induced overhead.

6 Conclusions and Future Work

Future exascale computing systems are expected to have significantly reduced network and memory bandwidth relative to computational performance than current systems. In this paper, we have demonstrated how existing large-scale parallel computers can be used to more closely emulate the expected imbalance of future systems. The resulting experimental platform can then be used for evaluating application sensitivity to the parameters under study. Our specific focus has been on network injection bandwidth, but many other hardware parameters can be examined as well. Results from our application scaling studies on a 160-node, 1920-core Cray XT5 system indicate that some applications experience sudden drops in performance at certain network injection bandwidth thresholds. Our ongoing work involves performing much larger-scale experiments to determine if the trends observed continue at higher core counts. We hope to leverage the empirical results obtained to assist in the development and validation of application models and simulation tools. Ultimately, we seek to gather information on the system balance requirements of existing highly-scalable parallel applications and leverage this insight to guide the design of future exascale supercomputers.

References

1. Alam, S.R., Vetter, J.S.: An analysis of system balance requirements for scientific applications. In: ICPP '06: Proceedings of the International Conference on Parallel Processing (2006)
2. Alvin, K., Barrett, B., Brightwell, R., Dosanjh, S., Geist, A., Hemmert, S., Heroux, M., Kothe, D., Murphy, R., Nichols, J., Oldfield, R., Rodrigues, A., Vetter, J.: On the path to exascale. International Journal of Distributed Systems and Technologies 1(2), 1–22 (May 2010)
3. Ang, J.A., Barnette, D., Benner, B., Goudy, S., Malins, B., Rajan, M., Vaughan, C.: Supercomputer and cluster performance modeling and analysis efforts: 2004-2006. Tech. rep., Sandia National Laboratories Technical Report, SAND2007-0601 (2007)
4. Brightwell, R., Hudson, T., Pedretti, K., Underwood, K.D.: SeaStar interconnect: Balanced bandwidth for scalable performance. IEEE Micro 26(3), 41–57 (May/June 2006)
5. E.S. Hertel, J., Bell, R., Elrick, M., Farnsworth, A., Kerley, G., McGlaun, J., Petney, S., Silling, S., Taylor, P., Yarrington, L.: CTH: A Software Family for Multi-Dimensional Shock Physics Analysis. In: Proceedings of the International Symposium on Shock Waves. pp. 377–382 (July 1993)
6. Ferreira, K.B., Bridges, P., Brightwell, R.: Characterizing application sensitivity to OS interference using kernel-level noise injection. In: SC '08: Proceedings of the International Conference on High-Performance Computing, Networking, Storage, and Analysis (November 2008)
7. Gittings, M., Weaver, R., Clover, M., Betlach, T., Byrne, N., Coker, R., Dendy, E., Hueckstaedt, R., New, K., Oakes, W.R., Ranta, D., Stefan, R.: The rage radiation-hydrodynamic code. Computational Science & Discovery 1(1), 015005 (2008)

8. Hoefler, T., Schneider, T., Lumsdaine, A.: Characterizing the influence of system noise on large-scale applications by simulation. In: SC '10: Proceedings of the International Conference for High Performance Computing, Networking, Storage, and Analysis (November 2010)
9. Hoisie, A., Johnson, G., Kerbyson, D.J., Lang, M., Pakin, S.: A performance comparison through benchmarking and modeling of three leading supercomputers: Blue Gene/L, Red Storm, and Purple. In: SC '06: Proceedings of the International Conference on High-Performance Computing, Networking, Storage, and Analysis (November 2006)
10. Kogge, P.M., et al.: Exascale computing study: Technology challenges in achieving exascale systems. Tech. rep., University of Notre Dame CSE Department Technical Report, TR-2008-13 (September 2008)
11. Lin, P.T., Shadid, J.N., Sala, M., Tuminaro, R.S., Hennigan, G.L., Hoekstra, R.J.: Performance of a parallel algebraic multilevel preconditioner for stabilized finite element semiconductor device modeling. Journal of Computational Physics 228, 6250–6267 (September 2009)
12. Luszczek, P., Dongarra, J., Koester, D., Rabenseifner, R., Lucas, B., Kepner, J., Mccalpin, J., Bailey, D., Takahashi, D.: Introduction to the hpc challenge (HPCC) benchmark suite. Technical Report (http://icl.cs.utk.edu/projectsfiles/hpcc/pubs/hpcc-challenge-benchmark05.pdf) (March 2005)
13. Rodrigues, A.: Programming Future Architectures: Dusty Decks, Memory Walls, and the Speed of Light, chap. 3, pp. 56–81. University of Notre Dame (2006)
14. Weaver, R., Gittings, M.: Massively parallel simulations with DOE's ASCI supercomputers: An overview of the Los Alamos Crestone project. In: Adaptive Mesh Refinement - Theory and Applications, Lecture Notes in Computational Science and Engineering, vol. 41, pp. 29–56. Springer Berlin Heidelberg (2005)