

PERFORMANCE CONSIDERATIONS FOR NETWORK SWITCH FABRICS ON LINUX CLUSTERS

Philip J. Sokolowski
Department of Electrical and Computer Engineering
Wayne State University
5050 Anthony Wayne Dr.
Detroit, MI 48202
email: phil@wayne.edu

Daniel Grosu
Department of Computer Science
Wayne State University
5143 Cass Avenue
Detroit, MI 48202
email: dgrosu@cs.wayne.edu

ABSTRACT

One of the most significant components in a cluster is the interconnection network between computational nodes. A majority of today's clusters use either switched Fast Ethernet, Gigabit Ethernet, or a specialized switch fabric to connect nodes. However, the use of these specialized switch fabrics may not necessarily benefit the users, and in some cases they perform only slightly better than commodity Ethernet standards. In this paper we investigate the necessity of a high speed interconnect for Linux based clusters and the performance differences between Myrinet and Ethernet connected networks. We benchmark applications and network performance in a dedicated environment with varying workloads and processing elements. We show that for certain workloads, a high speed interconnect is not required. We also examine the increased complexity of introducing high-speed switch fabrics, the importance of their underlying device drivers, and their usefulness with real applications. Finally, we make recommendations as to which network to use given a certain workload and available computation nodes.

KEY WORDS

cluster computing, performance evaluation, message-passing.

1 Introduction

Parallel and distributed computing systems have quickly become commonplace in arenas where historically, large-scale systems of this type were reserved for specialty applications and implementations. Specifically, Linux clusters constructed out of commodity hardware, utilizing freely available application programming interfaces are quickly becoming mainstream replacements for large, expensive, proprietary systems [1]. However, in order to implement a successful, user-friendly cluster, one must construct an environment in which many inexpensive nodes are linked together efficiently and soundly. Creating this environment within the confines of commodity hardware can be a daunting challenge. Stability, performance, complexity, and cost efficiency can be diverging characteristics. In our recent deployment of several NPACI Rocks [2] Linux clusters, we

examined a single component that affected all four of these factors, that being the interconnect fabric that linked nodes together on a network. We believe that the lessons learned from our implementation of a Rocks Linux cluster are beneficial to the scientific community, and are presented in this paper.

Our motivation behind writing this paper is that the interconnect fabric for our environment is the second most expensive component of the system and one of the most complex components to implement. Future implementations of Linux clusters at our institution and others can benefit from the benchmarks and recommendations made in this paper. Further, one can use this information in determining the benefits of establishing a computational grid [3].

The goals of this paper are to detail the complexity of interconnect switch fabrics for Linux clusters, to benchmark the performance of different switch fabrics available to us, and to make general recommendations regarding a particular switch fabric's preference over another for a given workload.

From a hardware perspective, two important factors affecting the performance of the parallel applications are: i) the performance of the node that is executing the computational work, and ii) the performance of the network that connects computational nodes together. Depending on the granularity of the problem, the significance of one of these factors will generally outweigh another. Fine-grained applications, which consist of a large number of small tasks, will assert the performance of the network interconnects. Coarse-grained applications with a small number of large tasks will prefer local node performance. In most parallel computing paradigms, the total cost of a system must be balanced between these two components, with the appropriate budget for network interconnects and computational nodes given the type of work that will be executed.

The problem we have faced in the deployment of our systems is the heterogeneous nature of the applications the cluster must accommodate. Further, there are several different clusters with some having access to Myrinet connected nodes, others with 100bT Ethernet, and still others with Gigabit Ethernet interconnects. Future clusters will undoubtedly introduce more switch fabrics such as VIA

[4], InfiniBand [5], or Quadrics [6]. It is important to understand the limitations and nature of these different interconnects in order to determine where a particular application will run most cost efficiently. Therefore, we benchmark the latency and bandwidth of Message Passing Interface (MPI) code [7] on Myrinet, 100bT Ethernet, and Gigabit Ethernet with the benchmarking suite MPBench [8]. We also analyze the performance of several parallel applications from the NASA Advanced Supercomputing suite [9] to benchmark the parallel runtime of an application when executed across different networks.

In past work, common switch fabrics used in today’s parallel systems have been evaluated [10] for many different protocols in tightly coupled systems. These evaluations provide useful information when designing a parallel system of many commodity nodes. Our approach is to evaluate three of the most commonly used switch fabrics for clusters, and determine if a particular type of network fabric is optimal for an application using MPI. This approach is logical and explicit, yet it is difficult to find published results on the performance of MPI applications, especially with modern processing elements and networks. Further, many benchmarks only evaluate network performance and not application performance, which have widely varying characteristics. In this paper, we examine the performance of the network fabric, and how it affects applications.

Organization

The paper is structured as follows. In Section 2 we describe our testing methodology. In Section 3 we discuss the performance results. In Section 4 we draw conclusions and present future directions.

2 Testing Methodology

We first define the hardware and software environment used in the benchmarks. We then standardized on an implementation of MPI (MPICH 1.2.5.2) and compile the MPI libraries with specific devices for each network fabric. We then compile the MPBench benchmark and NAS applications using a specific MPI library for a given network. Lastly, we dedicate all of the hardware used in this operation by terminating any running applications, then starting each benchmark.

2.1 Hardware and Software

The hardware in these experiments is based on Intel Xeon architecture running a distribution of NPACI Rocks (v3.1.0) Linux. One cluster, consisting of a master node and 16 compute nodes with both Myrinet and 100bT Ethernet connections between the nodes is used in the experiment. The 100bT Ethernet connection is replaced with a Gigabit Ethernet connection once benchmarks for the 100bT network are completed. The 100bT switch is a Summit 200-24 switch manufactured by Extreme Networks and the Gigabit network contains a pair of Avaya P332 se-

ries switches up-linked together with nodes interleaved between both switches.

The software used in this work consists of not only the network benchmarking suite and application suite, but also a series of middleware components. The critical software is determined to be the Linux kernel, Myrinet drivers, compilers, and MPICH specifically compiled for the appropriate network interface. These components are identical on all nodes. The Linux kernel itself is a demodularized version of the kernel that is distributed with NPACI Rocks v3.1.0. The Myrinet M3F-PCIX-D series interface cards use version 2.1.1 of the Myrinet device drivers. The Myrinet enabled benchmarks use a specialized version of MPICH-GM (ver 1.2.5..12) which provides the `ch_gm` interface, an interface that allows the use of Glenn’s Messages (GM) for MPI communications instead of standard TCP/IP, which is normally used with `ch_p4` interfaces. All compilations are performed with `gcc v3.2.3`. The specifics of the hardware and software are detailed in Table 1.

Component	Rocks Cluster
Number of Nodes	16
Interconnect 1	Myrinet
Interconnect 2	100bT Switched Ethernet
Interconnect 3	Gigabit Ethernet
CPU	2 x 2.6 GHz Xeon
RAM	2.5 GB
OS Version	NPACI Rocks 3.1.0
MPI Version 1	MPICH 1.2.5.2
MPI Version 2	MPICH-GM 1.2.5..12
Myricom Drivers	2.1.1

Table 1. Hardware and software configuration.

2.2 Benchmarking Software

The benchmarking software used to collect bandwidth and latency results presented in this paper is MPBench [8], (10/23/03 release), which is a freely available tool and part of the LLCbench (Low-Level Characterization Benchmarks) suite. MPBench is selected because is a widely-accepted, mature benchmarking tool that incorporates standard MPI routines. Second, MPBench is easy to implement, and can be easily re-compiled for specific implementations of MPI. Third, MPBench produces output data that can be easily graphed and analyzed. Eight different MPI benchmarks were obtained. However, only benchmarks concerning latency and bandwidth with message sizes ranging from 4 to 4194304 bytes were used in this paper.

The *bandwidth measurement* describes how much traffic a network can manage and the size of the message it can accommodate. The *latency measurement* describes how long it takes a send operation to reach the destination and be processed. Latency and bandwidth are highly susceptible to the implementation of the message-passing protocol, as we will see later. Therefore, it is optimal to use an

implementation of MPI on a cluster that has been specifically optimized for the cluster’s architecture.

The applications chosen from the NAS Parallel Benchmarks (NPB) are based on a Multigrid algorithm (MG), Integer Sort algorithm (IS), Embarrassingly Parallel applications (EP), Conjugate Gradient methods (CG), solutions of multiple independent systems of non diagonally dominant, scalar, pentadiagonal equations (SP) and block tridiagonal equations (BT), and an LU decomposition of the 3-D compressible Navier-Stokes equations (LU). Version 2.4.1 of the NPB suite is used. Due to space limitations, every application’s performance graph is not detailed in this paper. These applications are chosen because they accurately test both long and short distance data communication, integer speed, floating point speed, and parallelism of the applications. Class A problem sizes are used in order to assert the communication, rather than computational dependencies of each benchmark. Details on the applications can be found in the NPB white paper [9].

In order to draw successful conclusions regarding the suitability of a particular interface for a particular type of work, we test different interfaces with different implementations of MPICH and their respective protocols. We test Myrinet interfaces, 100bT Ethernet interfaces, and Gigabit Ethernet interfaces using the `ch_p4` interface with TCP/IP. We also test Myrinet interfaces with the `ch_gm` interface and Glenn’s Messages. We chose to test message sizes ranging from four bytes to four Megabytes, and we also vary the node count by powers of two starting with two nodes up through 16 nodes. Each node has one process mapped to it. We intentionally set this mapping because we are interested in evaluating the effect of network interfaces on applications rather than the computational power of the nodes. The following combinations of tests are performed, as listed in Table 2.

MPICH	Protocol	Network Interface	# of Nodes
MPICH 1.2.5.2	TCP/IP(<code>ch_p4</code>)	100bT Ethernet	2, 4, 8, 16
	TCP/IP(<code>ch_p4</code>)	Gigabit Ethernet	2, 4, 8, 16
	TCP/IP(<code>ch_p4</code>)	Myrinet (PCIX-D)	2, 4, 8, 16
MPICH-GM 1.2.5..12	GM(<code>ch_gm</code>)	Myrinet(PCIX-D)	2, 4, 8, 16

Table 2. MPICH versions, interfaces, protocols and number of nodes used in the benchmarks.

3 Performance Evaluation

First, we present a series of plots that depict the latency and bandwidth of the benchmarked configurations. There are a total of four different tests, each depicting latency and bandwidth as metrics. All of the results for combinations of two, four, eight, and 16 nodes are overlaid into one plot. Results are presented in a logarithmic scale. Secondly, we present plots for four of the application benchmarks: EP, IS, BT, and LU.

3.1 Performance on Fast Ethernet

We first examine the performance of Fast Ethernet using the `ch_p4` interface. The bandwidth and latency vs. packet size is presented in Figure 1 and Figure 2.

Here we can clearly see that regardless of the number of nodes involved, both the bandwidth and latency exhibit similar behavior. However, it is important to notice both the trends and limitations of Fast Ethernet and TCP/IP. The bandwidth of Fast Ethernet is poor relative to other switch fabrics, and the latency is very high when messages exceed 16K in size. We also see that the two components diverge, as for every increase in bandwidth, there is also an increase in latency. The most drastic jump in latency occurs when the message size exceeds 8K. The best trade-off of bandwidth for latency appears to occur when messages are less than 8K in size. Therefore, Fast Ethernet may be sufficient for fine-grained workloads containing messages smaller than 8K, and latencies no greater than 32 microseconds.

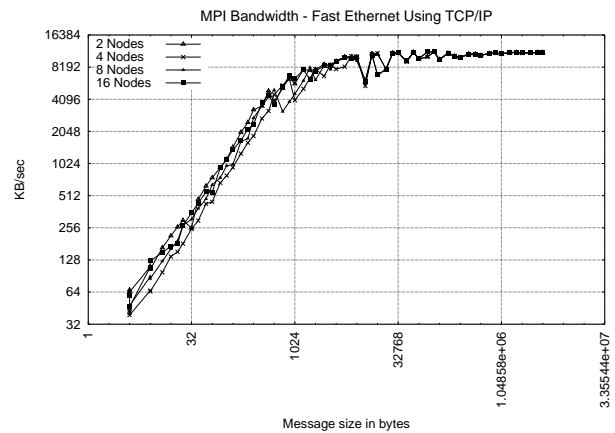


Figure 1. 100bT Ethernet Bandwidth With TCP/IP Message Passing Protocol (`ch_p4`).

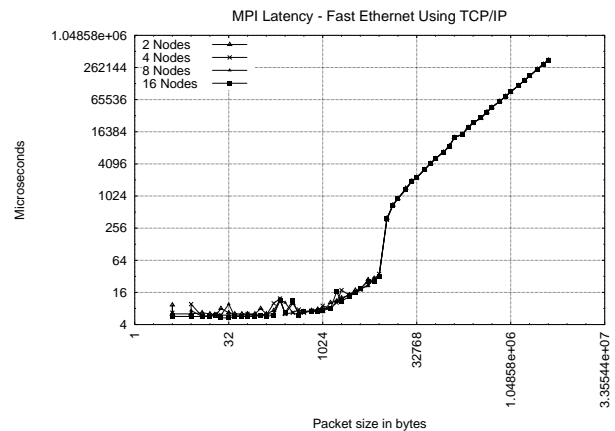


Figure 2. 100bT Ethernet Latency With TCP/IP Message Passing Protocol (`ch_p4`).

3.2 Performance on Gigabit Ethernet

We next evaluate the performance of Gigabit Ethernet using the `ch_p4` interface. The plots for Gigabit Ethernet (Figure 3 and Figure 4) at first seem extremely similar to that of Fast Ethernet. However, there is a massive increase in maximum bandwidth for large messages in Gigabit Ethernet. For messages less than 1K, Gigabit Ethernet's bandwidth is similar to Fast Ethernet, but for significantly large messages, it is eight to ten times greater.

When we consider the latency of smaller messages (less than 10K) the performance of Gigabit Ethernet is nearly the same as Fast Ethernet. However, messages larger than 10K assert latency advantages of Gigabit Ethernet. We observe a considerable improvement in Gigabit Ethernet's latency after this point, with it being five times faster than Fast Ethernet. We conclude that applications with messages smaller than 10K will perform more cost effectively on Fast Ethernet. For message sizes exceeding 10K, Gigabit Ethernet is more suitable.

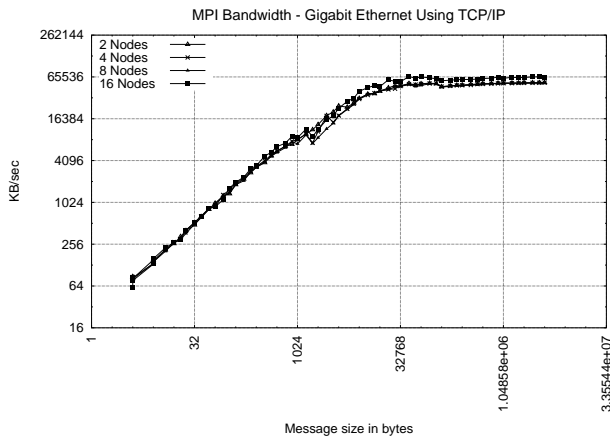


Figure 3. 1000bT Ethernet Bandwidth With TCP/IP Message Passing Protocol (`ch_p4`).

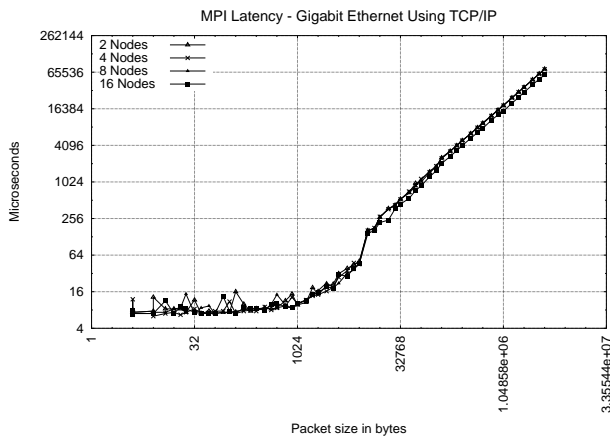


Figure 4. 1000bT Ethernet Latency With TCP/IP Message Passing Protocol (`ch_p4`).

3.3 Performance on Myrinet

We next evaluate the performance of a Myrinet network using the `ch_p4` interface. Figure 5 and Figure 6 detail the bandwidth and latency of a Myrinet network using TCP/IP. We see from these two figures that the bandwidth and latency are better than both Gigabit and Fast Ethernet, but not to the degree we would expect when examining Myrinet's published standards [11]. The reason for this deficiency is that Myrinet's performance is dependent on OS-bypass features, which are only available with the vendor-supplied GM protocol. When Myrinet devices are used with MPICH's standard `ch_p4` interface, they emulate Ethernet devices. Traffic must then flow from the application through the OS kernel, to the GM device driver. When GM is used as a device (`ch_gm`) instead, the traffic flows directly from the application to the interface, bypassing the OS. Thus, the kernel configuration, CPU performance, and IP protocol stack are introduced as factors that affect performance.

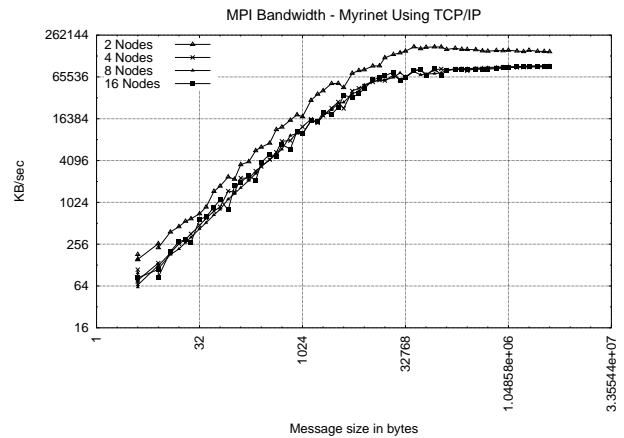


Figure 5. Myrinet Bandwidth With TCP/IP Message Passing Protocol (`ch_p4`).

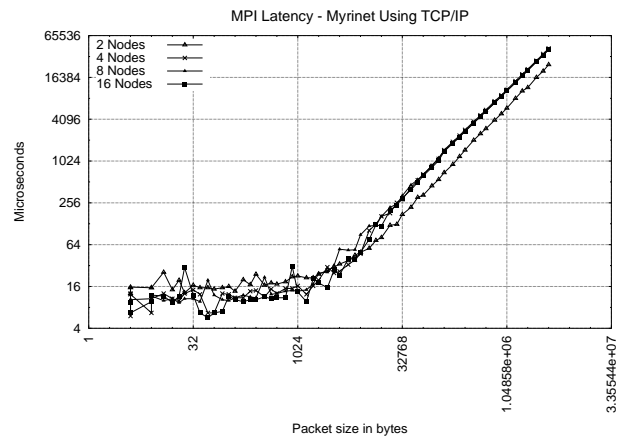


Figure 6. Myrinet Latency With TCP/IP Message Passing Protocol (`ch_p4`).

In Figure 7 and Figure 8 we evaluate the performance of a Myrinet network using the `ch_gm` interface. Here we clearly see the advantage of the GM protocol and its associated MPICH interface. For messages less than 4K in size, Myrinet is able to consistently sustain bandwidth performance better than Fast Ethernet and Gigabit Ethernet, and also maintain a latency less than eight microseconds. For larger messages over 32K in size, the bandwidth increases quickly, and the latency grows at about the same rate as Gigabit Ethernet but always remains significantly lower. We conclude that the MPI implementation is a critical component that must be implemented correctly for a given switch fabric technology. Without the vendor's supplied version of MPI, the performance of Myrinet is poor when compared to its published specification [12] and can even be outperformed by commodity networks.

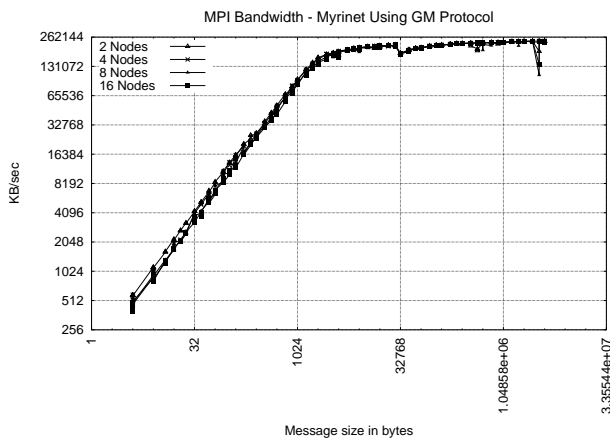


Figure 7. Myrinet Bandwidth With GM Message Passing Protocol (`ch_gm`).

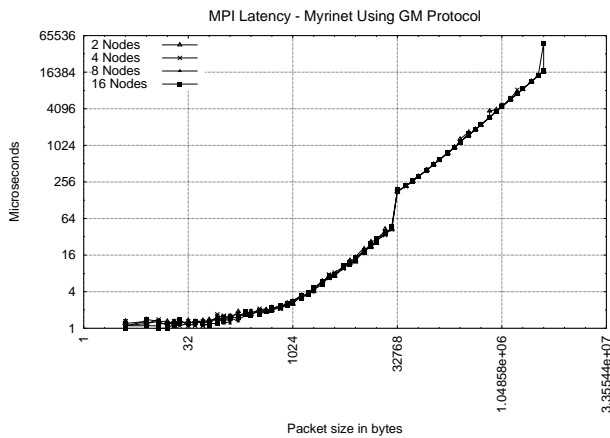


Figure 8. Myrinet Latency With GM Message Passing Protocol (`ch_gm`).

3.4 Network Performance Comparison

In Figure 9 and Figure 10, we compare Myrinet-GM, Fast Ethernet, and Gigabit Ethernet. In this last set of results, we have overlaid and normalized the performance metrics of the three switch fabrics. The latency performance of all three networks for messages around 8K in size are comparable. However, it is clear that Myrinet always outperforms Gigabit Ethernet and Fast Ethernet. The two Ethernets are similar in performance in both latency and bandwidth for messages less than 8K. It is not until we start passing messages larger than this that we see a significant difference between Gigabit Ethernet and Fast Ethernet. For a small range of message sizes where latency is more important to the application than bandwidth, Fast Ethernet is adequate.

3.5 Application Performance

We now discuss the performance of real applications over the four different interfaces. We first began with the analy-

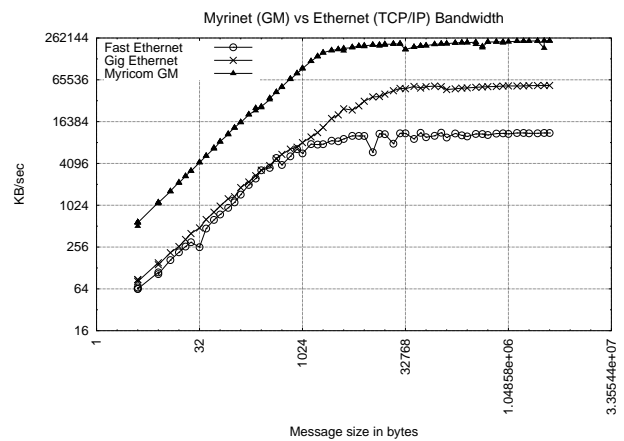


Figure 9. Bandwidth of Myrinet, Fast Ethernet, and Gigabit Ethernet.

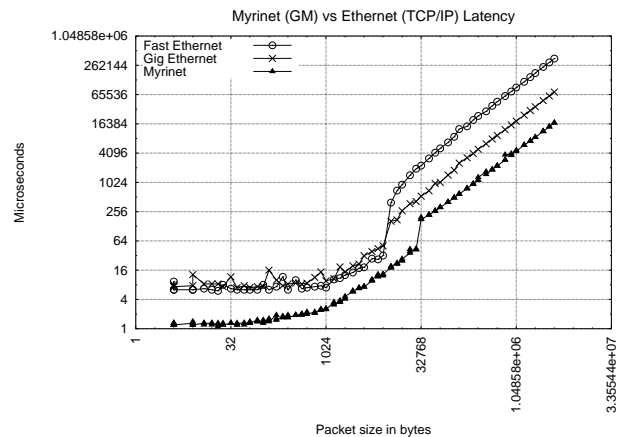


Figure 10. Latency of Myrinet, Fast Ethernet, and Gigabit Ethernet.

sis of the Embarrassingly Parallel application, which sets a baseline for the analysis of the communication overhead in the other applications. Unlike the other applications, EP has no communication overhead. If all communication devices and libraries for the four MPI devices tested in this paper perform similarly under the EP application, all other applications should perform equally regardless of the communication libraries or MPI implementation used. Figure 11 details the results of this benchmark.

We note that the performance of EP application over all interfaces is nearly identical, which demonstrates that all applications and devices tested use communication libraries that exhibit the same levels of optimality. MPI performance is dependent on the compiler and selected compile options, as all MPI applications depend on the underlying communication libraries. All applications in this paper have been compiled in the same manner using gcc and g77.

Figure 12 demonstrates the performance of the Integer Sort algorithm, which involves a high amount of communication overhead. A careful analysis of the IS algorithm shows that all communication in this benchmark is

by collective operations, and is a communication bound application [13]. This is distinct in the performance of the application over the Fast Ethernet interface where the high latency times of the switch fabric incur a very high communication cost and force the application to have a significantly higher runtime. The same behavior is demonstrated with the Myrinet interface using TCP/IP, where the communication performance is a function of both the hardware and the underlying device drivers. On interfaces such as Myrinet and Gigabit Ethernet, the performance was asymptotically similar, with Myrinet and GM being the best performer.

Figure 13 demonstrates the performance characteristics of Block Tridiagonal application, which communicates using nonblocking send operations. This application, as well as the Scalar Pentadiagonal application behave similarly, but the difference in overall application runtime is not as dependent on the interface used as in the other applications. Overall, Gigabit Ethernet and Myrinet GM perform nearly identically. However, the known deficiencies in the GM interface's communication libraries need to be further investigated before declaring that Gigabit Ethernet is a vi-

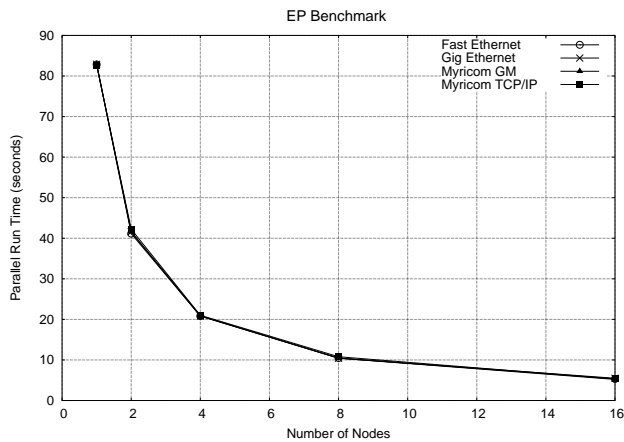


Figure 11. EP Performance on Four Interfaces.

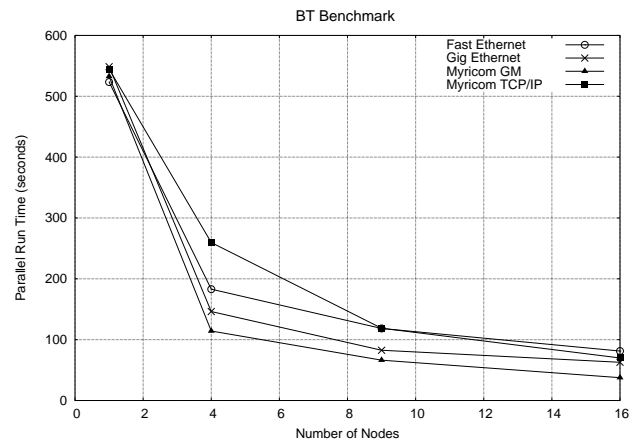


Figure 13. BT Performance on Four Interfaces.

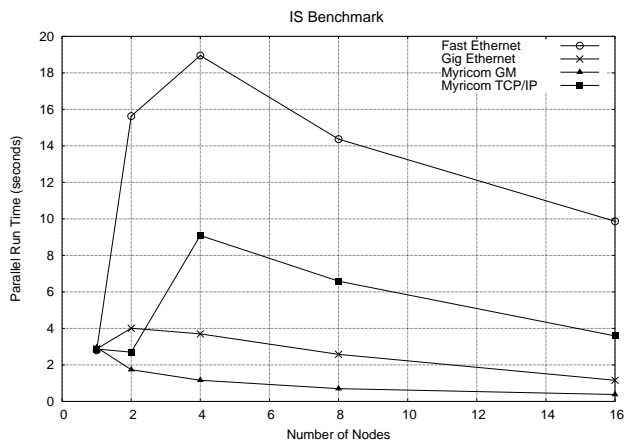


Figure 12. IS Performance on Four Interfaces.

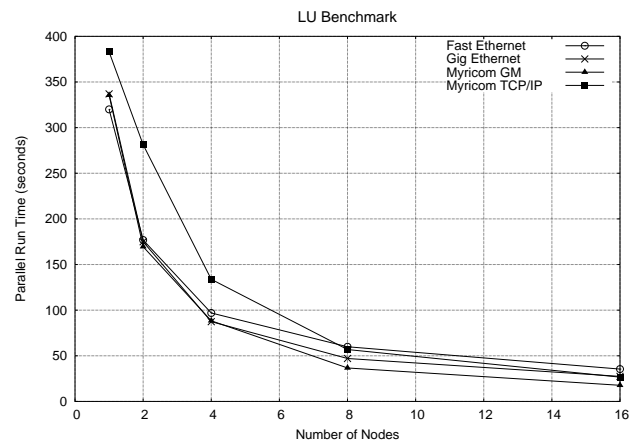


Figure 14. LU Performance on Four Interfaces.

able alternative to Myrinet GM in these types of applications.

Figure 14 depicts the performance of the LU decomposition for Navier-Stokes equations, which also behaves similarly to the MG application (not presented here). These two applications use traditional blocking send operations for primary communication plus a small number of collective function calls. The figure shows that in these types of applications, Myrinet GM is the best performer, and both Gigabit Ethernet and Fast Ethernet can be viable contenders for performance vs. cost considerations.

These benchmarks demonstrate how finely-grained applications run well on Gigabit Ethernet and Myrinet, but not on Fast Ethernet. It also demonstrates that Fast Ethernet is useful on coarsely-grained and moderately-grained applications. When we examine the three applications, we see clear indications of the performance of the network interface on the application. All applications were run on the same cluster, with the entire system solely dedicated to each application as it executed. In communication intensive applications such as IS and CG, the latency of the network introduces a significant amount of communication overhead. In applications such as LU and MG where the performance is computationally bound, we witness diminished network impact on overall parallel runtime.

4 Conclusion

In this paper, we have analyzed and compared the bandwidth and latency of Fast Ethernet, Gigabit Ethernet, and Myrinet. We demonstrated that the message-passing protocol used for a specific switch fabric is critical, and that specific vendors require specific implementations of their message-passing routines. We have shown that for certain applications that have small messages, the performance of Gigabit Ethernet and Fast Ethernet is similar, and if the application is computationally bound, these are sufficient fabrics. We demonstrated that some applications that have a significant amount of communication with larger messages can perform poorly on Fast Ethernet, and if cost is a factor, Gigabit Ethernet can still be a viable alternative to a custom switch fabric. We identified Myrinet as the invariably optimal technology to use from the three evaluated, but only when used with the GM protocol. We indicated that the underlying communication libraries for MPI have an important role in the performance of an application, and optimization of these libraries at compilation time is essential. We also identified the ideal message sizes for applications such that they would be highly portable and perform well, regardless of the switch fabric implemented on the cluster.

Many performance considerations of the NAS Parallel Benchmarks have been performed in the past, but none have considered them as a function of the switch fabric as we have investigated here. Given the appropriate hardware configurations, one could test many different types of computational nodes, network types, and applications. One potential evaluation would be to consider several dif-

ferent Linux kernels and quantify the effect of the OS on performance. Another useful test would evaluate several compilers and their effect on the MPICH communication libraries. In the near future, we will be considering specialized versions of MPICH using the globus2 interface on several clusters. We will also benchmark future switch fabrics (such as InfiniBand, AVI, and Quadrics) as they become available to us.

References

- [1] T. Sterling. Launching into the future of commodity cluster computing. In *Proc. of the IEEE International Conference on Cluster Computing*, page 345, September 2002.
- [2] NPACI. *NPACI Rocks Cluster Distribution*. <http://rocks.npaci.edu/Rocks/>.
- [3] G. Goth. Grid services architecture plan gaining momentum. *IEEE Internet Computing*, 6(4):7–8, 2002.
- [4] J. Wu, J. Liu, P. Wyckoff, and D. Panda. Impact of on-demand connection management in MPI over VIA. In *Proc. of the IEEE International Conference on Cluster Computing*, pages 152–159, September 2002.
- [5] D. Pendery and J. Eunice. Infiniband Architecture: Bridge over Troubled Waters. Research Note, Infiniband Trade Association, April 2000.
- [6] F. Petrini, W. C. Feng, A. Hoisie, S. Coll, and E. Frachtenberg. The Quadrics Network (QsNet): High-Performance Clustering Technology. In *Proc. of Hot Interconnects 9*, pages 125–130, August 2001.
- [7] Message Passing Interface Forum. Mpi: A message-passing interface standard. *International Journal of Supercomputer Applications*, 8(3/4):165–414, 1994.
- [8] P. Mucci, K. London, and J. Thurman. The MPBench Report. <http://icl.cs.utk.edu/projects/lcbench/mpbench.pdf>, November 1998.
- [9] D. H. Bailey, E. Barszcz, J. T. Barton, D. S. Browning, R. L. Carter, L. Dagum, R. A. Fatoohi, P. O. Frederickson, T. A. Lasinski, R. S. Schreiber, H. D. Simmon, V. Venkatakrishnan, and S. K. Weeratunga. The NAS parallel benchmarks. *Intl. Journal of Supercomputer Applications*, 5(3):66–73, Fall 1991.
- [10] D. Turner and X. Chen. Protocol-dependent message-passing performance on linux clusters. In *Proc. of the IEEE International Conference on Cluster Computing*, pages 187–194, September 2002.
- [11] Myricom Inc. *GM 2.0 API Performance with M3F-PCIXD Myrinet/PCI-X Interfaces*. <http://www.myri.com/myrinet/performance/index.html>.
- [12] J. Hsieh, T. Leng, V. Mashayekhi, and R. Rooholamini. Architectural and Performance Evaluation of GigaNet and Myrinet Interconnects on Clusters of Small-Scale SMP Servers. *Proc. Supercomputing 2000*, November 2000.
- [13] F. Wong, R. P. Martin, R. H. Arpaci-Dusseau, and D. Culler. Architectural Requirements and Scalability of the NAS Parallel Benchmarks. *Proc. of the 1999 ACM/IEEE Conference on Supercomputing*, November 1999.