

Performance of the NAS Parallel Benchmarks on Grid Enabled Clusters

Philip J. Sokolowski

*Dept. of Electrical and Computer Engineering
Wayne State University
5050 Anthony Wayne Dr., Detroit, MI 48202
phil@wayne.edu*

Daniel Grosu

*Dept. of Computer Science
Wayne State University
5143 Cass Avenue, Detroit, MI 48202
dgrosu@cs.wayne.edu*

Abstract

As Grids become more available and mature in real world settings, users are faced with considerations regarding the efficiency of applications and their capability of utilizing additional nodes distributed over a wide area network. When both tightly coupled clusters and loosely gathered Grids are available, a cost effective organization will schedule applications that can execute with minimal performance degradation over wide-area networks on Grids, while reserving clusters for applications with high communication costs.

In this paper we analyze the performance of the NAS Parallel Benchmarks using both MPICH-G2 and MPICH with the ch_p4 device. We compare the results of these communication devices on both tightly and loosely coupled systems, and present an analysis of how parallel applications perform in real-world environments. We make recommendations as to where applications run most efficiently, and under what conditions.

1. Introduction

Grid enabled technologies are becoming more pervasive within many organizations [6] where low-cost, high-throughput computing is demanded. Many Grids are used to execute embarrassingly parallel tasks or to efficiently schedule and share resources within an organization. One ideal use for Grids, however, is to execute highly parallel applications typically deployed on large SMP systems or tightly coupled clusters. Since Grids contain many machines over wide-areas, interconnected by high-latency links, applications that perform optimally in this environment are those exhibiting low communication overhead. However, in some instances the overhead affects the application to a degree where the cost associated with communication is detrimental to the overall parallel runtime. While this communication overhead may not be of significance

on a tightly coupled cluster, the added latency and limited bandwidth of a grid environment may assert the dependency the application has on communication.

In many situations, an application with moderate communication overhead may be scalable enough that it can take advantage of additional processors outside of a traditional cluster. However, if the application is scaled beyond the confines of the cluster to include nodes on a campus-wide Grid, will it benefit or suffer from the additional processes? When does the communication overhead affect the application to a degree where it becomes inappropriate to execute it on a Grid? Are Grid based services like Globus [4] required, or is a simple device such as ch_p4 adequate? These are questions we set out to answer in this analysis of MPI [3] based applications on campus-wide Grids. Our analysis of application performance includes the examination of both the globus2 and ch_p4 devices as part of MPICH, and the evaluation of applications from the NAS Parallel Benchmark suite. A pair of sixteen node clusters with two different process mappings is evaluated, and an analysis of the results is provided.

Although the NAS Grid Benchmarks [5] are available, we choose to run the standard NAS parallel benchmarks because our goal is to provide a comparative analysis of the performance of MPI applications using globus2 and ch_p4 devices and not the performance analysis of applications that are written taking into account the grid infrastructure.

Organization

The paper is structured as follows. In Section 2 we describe our testing methodology. In Section 3 we discuss the application performance results for two different process mappings and two MPICH devices. In Section 4 we examine related work and draw conclusions.

2. Testing Methodology

We first define the hardware and software environment used in the benchmarks. The two clusters are dedicated solely to the experiment, as are the internal network

WSU Grid Network - Two Cluster Sites

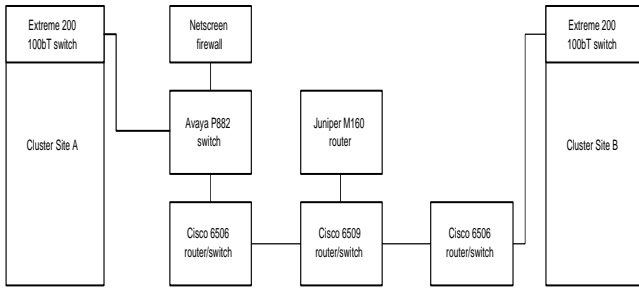


Figure 1. Wide-Area Campus Network Connecting Two Cluster Sites

switches used in within the clusters. The campus network is a real world production environment with shared usage. Although the wide-area network is shared, TCP tests show that the network is not saturated, and that bandwidth and latency performance is near ideal for a network of this construction. We then standardize on an implementation of MPI for both the `ch_p4` and `globus2` devices, and compile the NAS applications and execute them accordingly.

2.1 Hardware and Network

The hardware used for the experiments is based on Intel Xeon architecture running a distribution of NPACI Rocks (v3.1.0) Linux [9]. Both clusters consist of a master node and 16 compute nodes with 100bT Ethernet connections between the nodes. Each node is a dual processor Intel Xeon 2.6 GHz processor with 2.5GB of RAM. Each cluster uses a Summit 200-24 switch manufactured by Extreme Networks to provide connectivity between each of the compute nodes and the master node. Both cluster sites are identical and have no other applications running at the time of the experiment.

The network connecting the two cluster sites contains equipment by Avaya, Cisco, Netscreen, and Juniper. The network is a production environment with shared traffic. TCP test results indicate that the shared traffic on the production network did not interfere with application performance. The limiting devices in the network are the Netscreen firewall, which only operates at a speed of 100Mbps Full-Duplex, and the internal Extreme switches in the clusters themselves, which also operate at this speed. A complete diagram of the network connecting both cluster sites is detailed below in Figure 1.

Network traffic flows from Cluster Site A, through the Avaya P882 switch, through a VLAN through the Netscreen firewall, and back through the Avaya P882. At this point the

network bandwidth increases from 100Mbps to 1000Mbps. Traffic is next passed to a Cisco 6506, then to a Cisco 6509, and then routed over a VLAN through the Juniper M160 router. Traffic flows again through the Cisco 6509, to another Cisco 6506, and finally to Cluster Site B where the speed decreases back to 100Mbps. Analysis of this traffic flow identifies 100Mbps as the maximum bandwidth available to the experiment. Further, the latency of the wide area network is an order of a magnitude greater than the latency present within a single cluster [8], due primarily to the hierarchical nature of the network, and the many network devices that every packet must travel through between the two cluster sites.

2.2 Application Software

The software used in this work consists of NPACI Rocks v3.1.0 distributions with optimized 2.4.24 kernels and `gcc v3.2.3`. The MPI layer is based on two different interfaces, one compiled from source using MPICH v1.2.5.2 [1] with the `ch_p4` device, and the other, a pre-compiled distribution of MPICH-G2 v1.2.5-1a [7] using the `globus2` device. The MPICH-G2 package is distributed by the NSF Middleware Initiative (NMI) [10] and it is based on release 4 of the suite. Additionally, the Globus Toolkit v3.0.2 packaged from the NMI-R4 distribution is used to provide the underlying communication libraries and Grid enabled services required for MPICH-G2.

Applications are selected from the NAS Parallel Benchmark suite [2], and compiled from source using `gcc 3.2.3`. The applications chosen 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 LU decomposition of the 3-D compressible Navier-Stokes equations (LU). Version 2.4.1 of the NPB suite and class A problems are used. We have selected these applications because they accurately test both long and short distance data communication, integer speed, floating point speed, and parallelism of the applications.

Each application is first compiled from source and then executed on the two cluster sites using between 1 and 32 processes. Two different mappings of processes to nodes are implemented. The first mapping used is ‘sequential’, which maps processes first to the 16 compute nodes in Cluster A, and then to the 16 compute nodes in Cluster B. The second mapping performed is an ‘interleaved’ pattern where each consecutive processes is mapped to the next unused node in alternating clusters. These two mappings provide details on how application performance depends on the topology, and also provide details on the extensibility of an

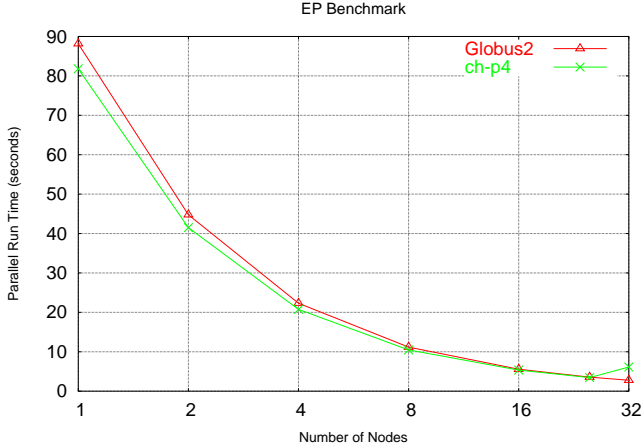


Figure 2. EP Performance

application when it is deployed beyond a single cluster. The interleaved mapping details application performance on a Grid where the physical location of compute nodes is varied and dynamic in nature. Each node has one process mapped to it. This intentional mapping is in place in order to evaluate the effect of communication on applications rather than the computational power of the nodes.

3. Performance Evaluation

We present a series of plots that detail the application parallel runtime vs. the number of processes (nodes) used by the application. We also present the same application for both sequential and interleaved mappings. There is a total of seven application benchmarks, performed with both the globus2 and ch_p4 device. We first begin with the analysis of the Embarrassingly Parallel (EP) 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 at all, therefore all applications should perform equally regardless of the communication device or process mappings. Figure 2 details the results of this benchmark.

We note that the performance for the EP application is as expected, with the overall runtime decreasing proportionally with each additional node. We note that for the ch_p4 interface, the initial execution time for one process is several seconds less than the globus2 device. We also note that the ch_p4 device performs slightly better than the globus2 device until 32 nodes are utilized. The explanation for this difference is that the ch_p4 device and globus2 device rely on a different set of communication and application libraries, with the globus2 libraries obtained from a pre-compiled distribution and the ch_p4 libraries compiled from source. Since the exact compilation flags and options are

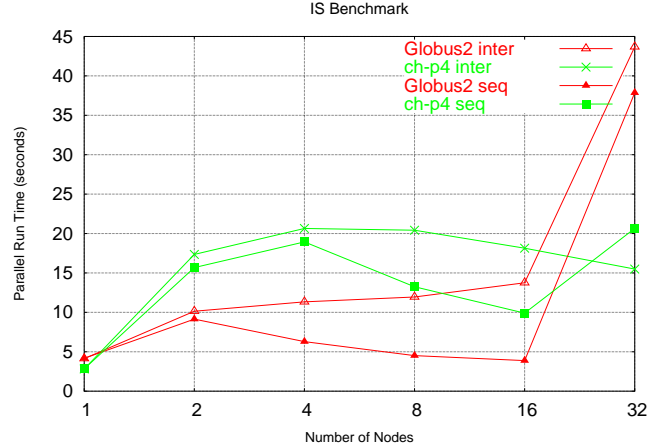


Figure 3. IS Performance

not documented for the NMI-4 distribution of the globus2 device, we can only obtain an approximation in comparable library optimality for the ch_p4 device. Given that this figure suggests the performance of the two devices are nearly identical for both devices when using multiple processes, we chose to disregard the differences in library optimality for further analysis. Readers are encouraged to note the dependency applications have on the optimality of the underlying libraries [12] and consider compiling applications and MPI layers from source using the best possible optimization options provided by the compiler.

Figure 3 demonstrates the performance of the Integer Sort (IS) algorithm, which involves a high amount of communication overhead. A careful analysis of the IS application shows that all communication is performed by collective operations, making IS a communication bound application [13]. The IS application over both mappings of processes performs significantly better on the globus2 device when less than 16 processes are used. Once 32 processes are utilized for the application, the ch_p4 device outperforms the globus2 device. Overall, we see that the communication overhead in the IS application causes the performance to degrade regardless of the process mapping or the communication device. Performance is better overall, however, when the processes are confined within one cluster, as we see from the results of the sequential mapping for 16 processes. From the performance of this application, we can conclude that communication intensive applications perform better on the globus2 device than the ch_p4 device. However, the communication overhead can be detrimental to an application if it is scaled out to too many processes, as it is evident in the 32 processor case.

Figures 4 and 5 demonstrate the performance characteristics of Block Tridiagonal (BT) and Scalar Pentadiagonal (SP) applications, which communicate using nonblocking

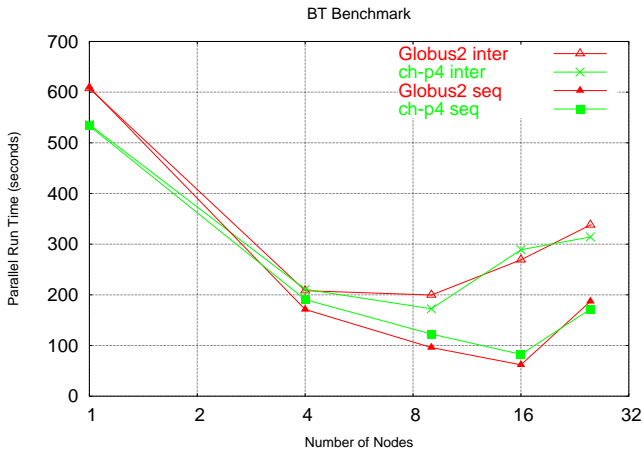


Figure 4. BT Performance

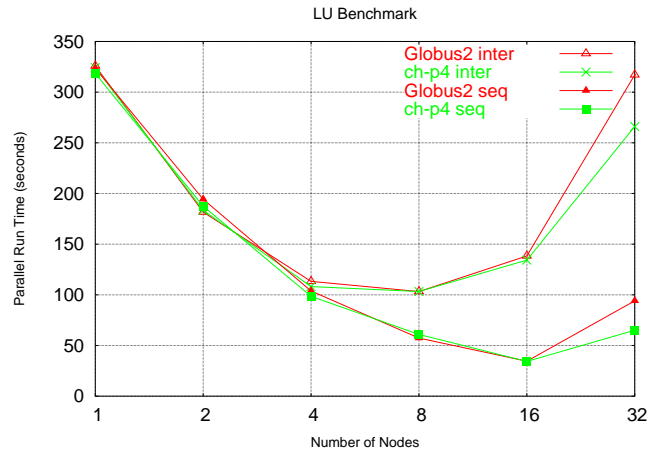


Figure 6. LU Performance

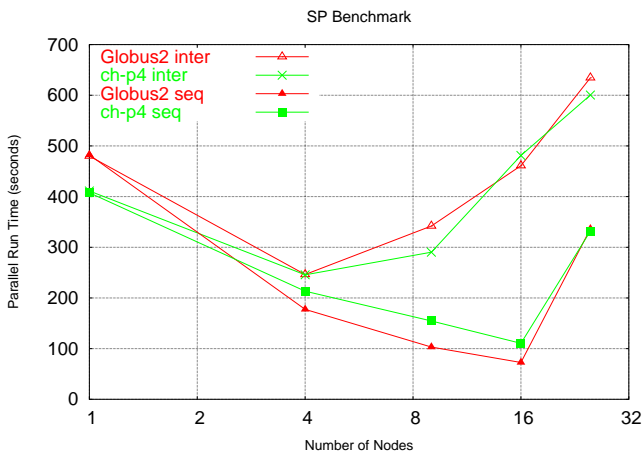


Figure 5. SP Performance

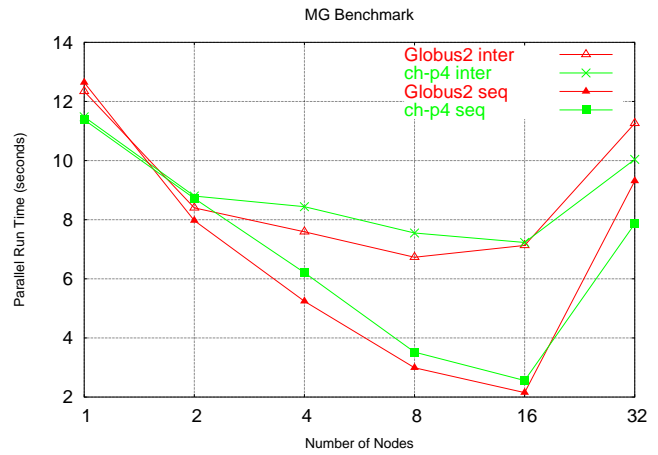


Figure 7. MG Performance

send operations. Because BT and SP require a square number of processes we scaled the system up to 25 out of 32 processes. The results of both applications for both the ch_p4 and globus2 devices are asymptotically similar regardless of the process mappings. In general, the globus2 device performed better than the ch_p4 device for sequential mappings. Performance for interleaved mappings varied, with some number of processes performing better on one device than another. In both applications, the performance of interleaved mappings was two to four times slower than that of sequential mappings. This performance degradation is due to the high latency introduced when using the wide-area campus network. Even though the campus network has the same bandwidth as the local network switch inside each cluster, the additional latency introduced is directly responsible for the decreased performance of the applications. This communication overhead is evident even in the sequen-

tial mapping once the application scales out to a pair of clusters. These types of applications will always perform better within a single cluster, and will have a significantly longer parallel runtime when extended to several clusters over a wide-area network, even with sequential process mapping. Both BT and SP utilize an all-to-all communication topology [11] which asserts the latency of the wide-area link between processes on the two clusters.

Figure 6 depicts the performance of the LU decomposition for Navier-Stokes equations, and Figure 7 details the performance of a V-cycle multigrid (MG) algorithm. These two applications use traditional blocking send operations for primary communication plus a small number of collective function calls. The figures show that in these types of applications, the globus2 device performs better than the ch_p4 device in most instances. When comparing the sequential process mapping with that of the interleaved map-

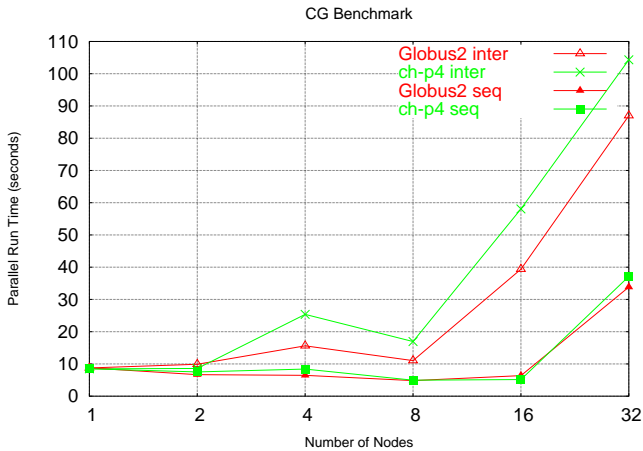


Figure 8. CG Performance

ping, the interleaved mapping causes the application to perform two to three times slower than the sequential mapping. Similar to the performance of applications that use non-blocking send operations, these applications do not scale well when extended to multiple clusters. Both LU and MG exhibit a ring topology for communication between processes [11], but with MG transmitting packets at a slightly higher rate, and with a larger average packet size than LU. The communication pattern, however, being the same between the two applications, is the predominant reason for the similar performance curves between the two applications.

Figure 8 depicts the performance of unstructured computations typically encountered on Grid systems. The application is designed to test long distance communication, and employs unstructured matrix vector multiplication. We see that this application performs asymptotically the same for both the globus2 and ch_p4 devices regardless of the process mapping. However, we observe that with interleaved mappings, the performance of the application is three times slower than that of a sequentially mapped process within a single cluster. Again, we notice the significant increase in runtime when a sequentially mapped application extends from 16 processes to 32 processes and spans multiple cluster sites. The results conclude that there is a sufficient number of messages such that the latency of the wide-area campus network introduced considerable communication overhead. This ultimately led to poor application performance once messages traveled outside of the cluster and through the wide-area network.

In these experiments with applications and two cluster sites, we note that in general, the globus2 device outperformed the ch_p4 interface, especially in situations where the communication overhead bound the application performance. In conditions where the application was embarrass-

ingly parallel, or computationally bound, the significance of using the globus2 device diminished, and the ch_p4 device even outperformed the globus2 device in some situations. In every situation, applications performed better on a single cluster rather than on a wide-area grid, or a pair of clusters. We see that sequentially mapped processes perform better than interleaved processes, even when both are extended to 32 nodes across a pair of clusters.

4. Conclusion

Many performance considerations of the NAS Parallel Benchmarks have been performed in the past, but none have considered them as a function of the MPI device over wide-area networks, and as a function of the process mapping as we have investigated in this paper. Many techniques are being developed to streamline the performance of message passing communication such as generating dynamic tree shapes for collective operations, and segmentation of large messages so that concurrent links in a wide-area network can be exploited [8]. MPICH-G2 uses information specified by users to create multilevel clustering of processes based on the underlying network topology, but this information must be provided by the user at execution time, and may not be available to the user.

In this paper, we analyzed and compared the performance of several applications when executed across both globus2 and ch_p4 devices within MPICH. We evaluated the performance of these applications with both sequential and interleaved mapping of processes across a pair of clusters separated by a wide-area campus network. We observed the importance of the underlying communication libraries and determined that in most conditions the globus2 device performs better than the ch_p4 device, and it is an ideal device to use on both Grids and individual clusters. We observed the impact of high-latency networks and the effect of process mappings on the latency. We observed that even with a fixed bandwidth either within a cluster, or between separate cluster sites, the latency alone is enough to degrade an application to a point where it is no longer cost effective to execute on a pair of clusters over wide-area links, regardless of process mapping.

References

- [1] Argonne National Laboratory. *MPICH - A portable implementation of MPI*. <http://www-unix.mcs.anl.gov/mpi/mpich>.
- [2] 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. Venkatakrisnan, and S. K. Weeratunga. The NAS parallel benchmarks. *Intl. Journal of Supercomputer Applications*, 5(3):66–73, Fall 1991.

- [3] M. P. I. Forum. MPI: A Message-Passing Interface Standard. *International Journal of Supercomputer Applications*, 8(3/4):165–414, 1994.
- [4] I. Foster, C. Kesselman, J. Nick, and S. Tuecke. The Physiology of the Grid: An Open Grid Services Architecture for Distributed Systems Integration. Open Grid Service Infrastructure WG, Global Grid Forum, June 2002.
- [5] M. Frumkin and R. F. Van der Wijngaart. NAS Grid Benchmarks: A Tool for Grid Space Exploration. *Cluster Computing*, 5(3):247–255, 2002.
- [6] G. Goth. Grid services architecture plan gaining momentum. *IEEE Internet Computing*, 6(4):7–8, 2002.
- [7] N. T. Karonis, B. Toonen, and I. Foster. MPICH-G2: A Grid enabled implementation of the Message Passing Interface. *Journal of Parallel and Distributed Computing*, 63:551–563, 2003.
- [8] T. Kielmann, H. E. Bal, and S. Gorch. Bandwidth-efficient collective communication for clustered wide area systems. In *Proc. of the 4th IEEE International Parallel and Distributed Processing Symposium*, pages 492–499, May 2000.
- [9] NPACI. *NPACI Rocks Cluster Distribution*. <http://rocks.npaci.edu/Rocks/>.
- [10] NSF Middleware Initiative. *NSF Middleware Initiative - Release 4*. <http://www.nsf-middleware.org/NMIR4/>.
- [11] J. Subhlok, S. Venkataramaiah, and A. Singh. Characterizing NAS benchmark performance on shared heterogeneous networks. In *Proc. of the IEEE International Parallel and Distributed Symposium*, April 2002.
- [12] 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.
- [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.