

COMMUNICATION CHARACTERISTICS OF MESSAGE-PASSING SCIENTIFIC AND ENGINEERING APPLICATIONS

Reza Zamani

Ahmad Afsahi

Department of Electrical and Computer Engineering

Queen's University

Kingston, ON, Canada K7L 3N6

{zamanir, ahmad}@ee.queensu.ca

Abstract

Communication performance is an important factor that affects the performance of message-passing parallel applications running on clusters. A proper understanding of communication behaviour of parallel applications will help designing better communication subsystems and MPI libraries in the future. It will also help application developers to maximize their application performance on a target architecture.

This paper examines the message passing communication characteristics of three applications (BT-MZ, SP-MZ, and LU-MZ) in the NAS Multi-Zone parallel benchmark suite as well as two applications (SPEC_{env} and SPEC_{seis}) in the SPEC_{hpc2002} suite. Our study considers both point-to-point and collective communications. For point-to-point communications, we quantify the message type, message frequency, message size, and message destinations. For collectives, we examine their type, frequency, and payload. Our results show that the applications studied have diverse communication patterns and that they are mostly sensitive to the changes in the system size and the problem size. All applications use only a few collective operations, while SPEC applications use them frequently with very large payloads. Overall, our work helps in a better understanding of the communication workloads in the current and emerging parallel applications.

Keyword

Communication Characteristics, Message-Passing, Parallel Applications, MPI, Clusters

1. Introduction

Clusters of workstations/multiprocessors have been regarded as cost-effective platforms for high-performance computing. In such systems, communication performance is an important factor that affects the performance of parallel applications. Many factors influence the performance of communication subsystems. Specifically, communication hardware and its services, communication system software and libraries, and the user environment (multiprogramming,

multiuser) are the major sources of the communication overhead.

Most parallel applications running on clusters use the *Message Passing Interface* (MPI) [1] as the basic foundation for explicit message passing. MPI provides the application developer with a rich set of functionality. Depending on the nature of the application, its algorithm, the set of MPI functionality used, and the workload and system size, MPI parallel applications exhibit a wide range of communication behaviour. In essence, their behaviour plays a key role in performance. A proper understanding of the communication patterns of such applications will help application developers to maximize their application performance in a given environment. It will also help system designers to come up with better communication architectures as well as optimized MPI libraries in the future.

Recently, two suites of applications have been introduced: the *NAS Multi-Zone (NPB-MZ)* suite [2] from NASA, and the *SPEC_{hpc2002}* suite [3] from Standard Performance Evaluation Corporation (SPEC). These suites are widely used for performance evaluations of parallel architectures and as of today are the most trusted benchmarks. Applications in these suites can be run with MPI, OpenMP [4], or mixed MPI-OpenMP.

Several researchers in the past have investigated the communication behaviour of parallel applications [5-11]. Along the same path, we examine the MPI characteristic of the three applications (*BT-MZ*, *SP-MZ*, and *LU-MZ*) in the *NPB-MZ* suite as well as two applications (*SPEC_{seis}* and *SPEC_{env}*) in the *SPEC_{hpc2002}* suite in terms of their point-to-point and collective communications. We quantify the message type, message frequency, message size, and message destinations for point-to-point communications, as well as the type, frequency, and payload for collective operations. We also evaluate the impact of different number of processors as well as different problem sizes on the communication characteristics of these applications. Overall, our experiments reveal that the applications studied in this paper have diverse communication patterns, and that they are sensitive to the changes in the system size and the problem size.

The rest of this paper is organized as follows. In Section 2, we introduce the applications. Section 3 describes our experimental methodology. In Section 4, communication characteristics of the applications are presented. Related work is presented in section 5. Finally, we conclude our paper in section 6.

2. Applications

The applications studied in this paper include the NAS Multi-Zone (NPB-MZ) suite [2], and the SPEChpc2002 suite [3]. NPB-MZ consists of three simulated computational fluid dynamics (CFD) applications: BT-MZ, SP-MZ and LU-MZ. SPEChpc2002 consists of three real applications: SPECseis, SPECenv, and SPECchem. An overview of these applications is presented in Table 1.

Table 1. Overview of applications.

Application	Field	Language	Lines
BT-MZ	Computational fluid dynamics	FORTRAN	4600
SP-MZ	Computational fluid dynamics	FORTRAN	4100
LU-MZ	Computational fluid dynamics	FORTRAN	4500
SPECseis	Seismic processing	C and FORTRAN	20,000
SPECenv	Climate modeling	C and FORTRAN	143,000
SPECchem	Computational chemistry	C and FORTRAN	100,000

2.1 NAS Multi-Zone

The NAS Parallel Benchmark (NPB) [12] is a set of 8 benchmark programs designed to help evaluate the performance of parallel computers. It has five kernels, and three simulated CFD applications: BT, SP, and LU. The NPB-MZ, introduced in July 2003, is an extension of the NPB suite that involves solving the application benchmarks BT-MZ, SP-MZ and LU-MZ on collections of loosely coupled discretization meshes. Many important scientific problems feature several levels of parallelism, and this property is not reflected in NPB. To remedy this deficiency, the multi-zone versions were created.

BT-MZ uses an implicit algorithm to solve 3-dimensional compressible Navier-Stokes equations. The finite-differences solution to the problem is based on an Alternating Direction Implicit (ADI) approximate factorization. SP-MZ has a similar structure to BT-MZ. The finite-differences solution to the problem is based on a Beam-Warming approximate factorization. LU-MZ uses symmetric successive over-relaxation (SSOR) method to solve a seven-block-diagonal system.

Applications in the NPB-MZ suite can be run with MPI, OpenMP, or mixed MPI-OpenMP. There are five different workload classes available: small (S), workstation (W), large (A), larger (B, and C; C is larger than B), and largest (D).

2.2 SPEChpc2002

In December 2002, SPEC introduced the SPEChpc2002 suite. The benchmarks in the suite are derived from real high performance computing applications and measure the overall performance of high-end computer systems.

SPEChpc2002 consists of three applications: SPECseis, SPECenv, and SPECchem. SPECseis represents an industrial application that performs time and depth migrations used to locate gas and oil deposits. SPECenv is based on a weather research and forecasting model called WRF. SPECchem is based on a quantum chemistry application called GAMESS (General Atomic and Molecular Electronic Structure System).

SPECchem and SPECenv support MPI, OpenMP and mixed MPI-OpenMP parallelism. SPECseis supports MPI, and OpenMP parallelism. There are two different workload classes available: small (S), and medium (M). Basic requirements for SPEChpc2002 are UNIX (Linux) with 2GB of memory, up to 100GB of disk, and a set of compilers.

3. Experimental Methodology

This paper examines the MPI characteristics of the BT-MZ, SP-MZ, and LU-MZ applications (classes B and C) in the NPB-MZ suite as well as the SPECseis, and SPECenv applications (classes S and M) in the SPEChpc2002 suite in terms of their point-to-point and collective communications (we will report their mixed-mode characteristics in a future work). For point-to-point communications, we quantify the number of messages sent per process, the average message size sent per process, the cumulative distribution function (CDF) of the message sizes, the number of distinct message destinations per process, and the distribution of message destinations for the root process (process zero). For collective communications, we present the type and the number of collectives used in the applications, as well as their payloads. We also evaluate the impact of the problem size and the system size on the communication characteristics of the applications.

The results reported here are not from simulation, but gathered through the actual application run, and independent of the target machine. We ran our applications on a dedicated 32-processor cluster at the Parallel Processing Research Laboratory at the Queen’s University. Our cluster consists of four Dell PowerEdge 6650s, each having four Intel Xeon MP 1.4GHz processors with 256 KB cache, and eight Dell PowerEdge 2650s, each having 2 Intel Xeon MP 2.0GHz processors with 512KB cache. Each node has 512MB of RAM per processor, and runs Linux RedHat 9 with SMP kernel version 2.4.24. The nodes are interconnected by the latest Myrinet network using the two-port “E-card” network interfaces. We used version 7.1 of the Intel compilers, and MPICH 1.2.5..10 [13] for compiling, building, and running the benchmarks.

We did not change the source code of the applications. Instead, we wrote our own profiling code using the wrapper facility of the MPI to gather the communication traces of

the applications. We did this by inserting monitor operations in the profiling MPI library for the communication related activities. Note that gathering the communication traces of the applications does not affect their communication patterns.

4. Communication Characteristics

Communication characteristics of an application may significantly affect the application performance in a given environment. Table 2 shows the various MPI function calls used in the applications studied in this paper. It is evident that these applications use a small subset of the MPI functions. This is consistent with previous studies on other applications [5]. Applications in the NPB-MZ use non-blocking point-to-point function calls that allow them to overlap their communication with the computation. SPECenv uses only non-blocking receive function, while SPECseis uses blocking functions. The applications studied use up to three different collective communication operations. In the following, we examine the characteristics of the point-to-point and collective communications of our applications.

Table 2. MPI functions usage in the applications.

Routines	SPECseis	SPECenv	BT-MZ	SP-MZ	LU-MZ
MPI_Send	√	√			
MPI_Recv	√	√			
MPI_Isend					√
MPI_Irecv		√	√	√	√
MPI_Wait		√			
MPI_Waitall			√	√	√
MPI_Bcast	√	√	√	√	√
MPI_Barrier	√		√	√	√
MPI_Reduce		√	√	√	√

4.1 Point-to-Point Communication

Point-to-Point communication is the simplest type of communication among processes in message-passing applications. It can be easily supported by blocking/non-blocking send and receive primitives. However, their characteristics play a key role in the performance of the applications that use them.

Communication properties of message-passing parallel applications can be categorized by the *temporal*, *volume*, and *spatial* attributes of the communications [10, 6]. The temporal attribute of communications characterizes the rate of message generations, and the rate of computations in the applications. We will discuss the temporal attributes in another paper. The volume of communications is characterized by the number of messages, and the distribution of message sizes in the applications. The spatial attribute of communications is characterized by the distribution of message destinations.

Figure 1 shows the number of messages sent per process for the applications under different problem sizes and number of processes. It is evident that the number of messages sent in the BT-MZ, SP-MZ, and LU-MZ is

decreasing (except for some of the cases where the number of processes is two) with the increasing number of processes. This trend is consistent for both classes B, and C. However, contrary to the NAS-MZ benchmarks, the number of messages sent for the SPECenv and SPECseis applications show a different trend, where they actually increase when the number of processes increases. This trend is consistent for both small and medium workloads.

An interesting observation is that the processes in the LU-MZ (except for C2) and SPECseis send equal number of messages to their destinations, where this is not the case for the other applications. Among the five applications, SPECenv has the largest number of messages sent per process. For instance, with 16 processes, each process in SPECenv sends 90,000 messages on average, while each process in the SP-MZ, BT-MZ, SPECseis, and LU-MZ sends roughly 13,000, 10,000, 5,000 and 1,000 messages, respectively.

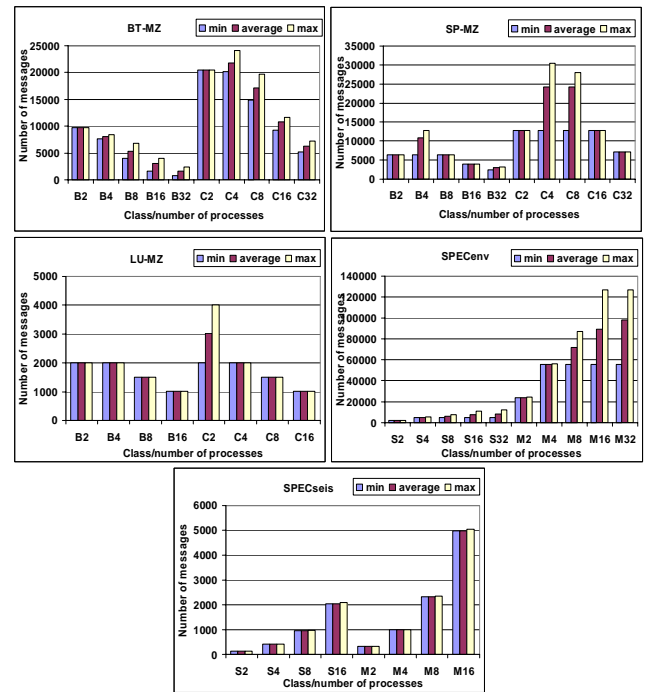


Figure 1. Number of messages sent per process.

Figure 2 presents the average message size per send in the applications. One can easily see that the average message size for the SPEC applications, for both small and medium workloads, becomes smaller as the number of processes increases. In contrast, the average message size for the NAS-MZ application benchmarks increases.

BT-MZ, and SP-MZ roughly use the same sort of message sizes, between 120KB to 180KB for the BT-MZ, and between 100KB to 165KB for the SP-MZ. However, LU-MZ uses larger message sizes, especially for the larger class C with message sizes between 640KB to 800KB. SPEC applications, especially for the small class S, use the least average message size among the all applications. An

overall observation is that the LU-MZ is more bandwidth-bound than the other applications.

The total message volume that a process sends over the network is roughly equal to the average message size per send times the number of messages sent per process. Although not shown here, the total message volume sent by each process in the SPEChpc2002 applications is very different for small and medium classes. However, this is not the case for the classes B and C in the NAS-MZ. SPECenv, medium class, has the largest message traffic per process on the network, roughly between 4000MB to 7000MB. However, for the small class, each process sends roughly 140MB to 230MB. SPECseis seems to have the lightest message traffic per process on the network with 110MB to 130MB of data for the medium class, and around 8MB for the small class. For the C class, each process in BT-MZ, SP-MZ, and LU-MZ sends 1100MB to 3500MB, 1100MB to 3800MB, and 780MB to 1850MB, respectively.

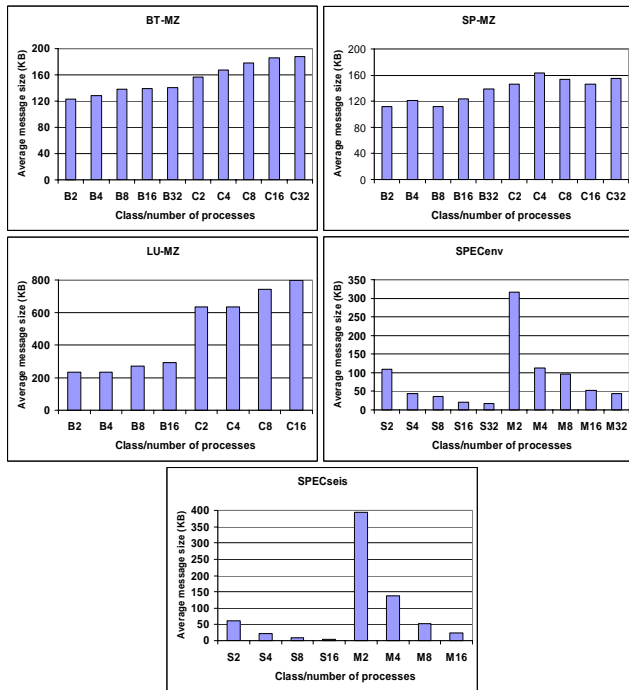


Figure 2. Average message size per send.

The *cumulative distribution function* (CDF) of message sizes provides greater detail about the sizes of messages sent in an application. Figure 3 presents the CDF of the message sizes for the applications under different system and problem sizes. Note that the horizontal axis for the SPECseis is in logarithmic scale. From the graphs, it can be seen that the BT-MZ and SPECenv use a large number of different message sizes, while the other applications use only a few message sizes. BT-MZ uses up to 21 different message sizes in class C (16 in class B). The shortest and the longest messages are 42KB, and 446KB, respectively. SPECenv uses up to 70 different messages sizes in class S (50 in class M). It uses both short messages (as small as 4

bytes) and very long messages (as large as 3129KB for the class M). Thus, SPECenv is very much sensitive to both latency and bandwidth of the interconnect

The distribution of the message sizes sent by the SP-MZ, and LU-MZ are bimodal. That is, they only use two different message sizes. Message sizes for the SP-MZ, and LU-MZ suggest that these two applications are bandwidth-bound. SPECseis uses five different message sizes each for both classes. It uses small messages (including zero-byte messages) as well as very large messages (up to 32768KB). This shows that SPECseis is more sensitive to the bandwidth than to the latency of the interconnect.

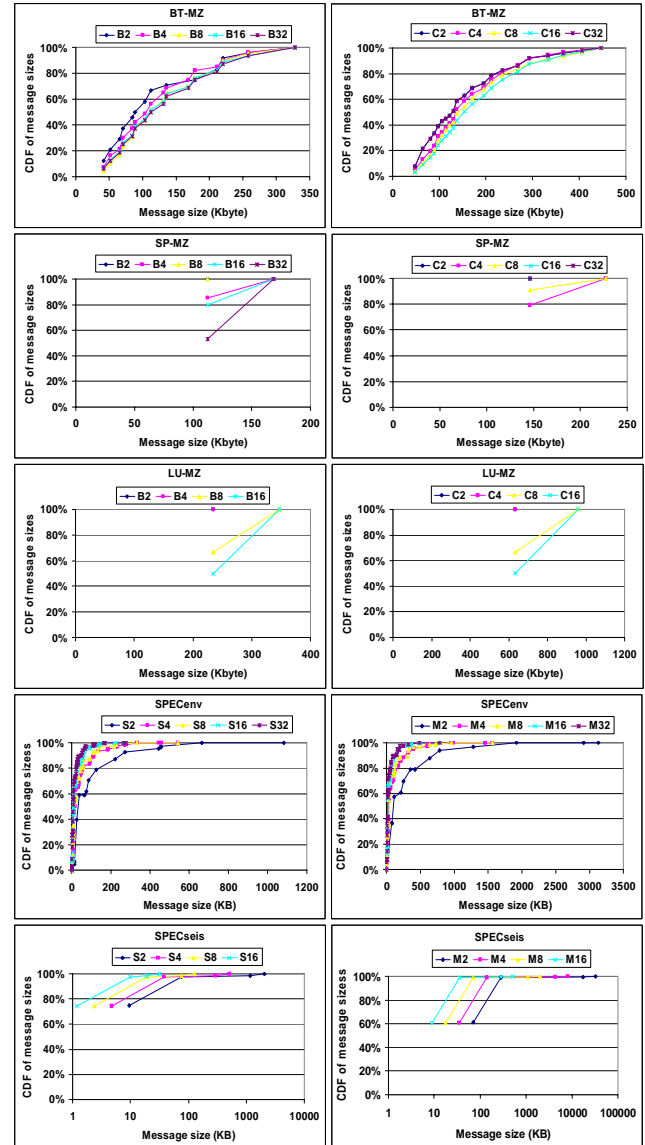


Figure 3. Cumulative distribution function of message sizes.

Spatial behaviour is characterized by the distribution of message destinations [10, 6]. We have studied the number of message destinations for each process in the applications. Figure 4 shows that the number of message destinations per process does not change with the workload

for the LU-MZ, and SPEC applications. Processes in the LU-MZ, SP-MZ, and SPECenv (except for some of the processes) have a few favourite communication partners. This is consistent with previous studies for other applications [5, 6]. However, processes in BT-MZ (especially the C class) and SPECseis communicate with most of the remaining processes.

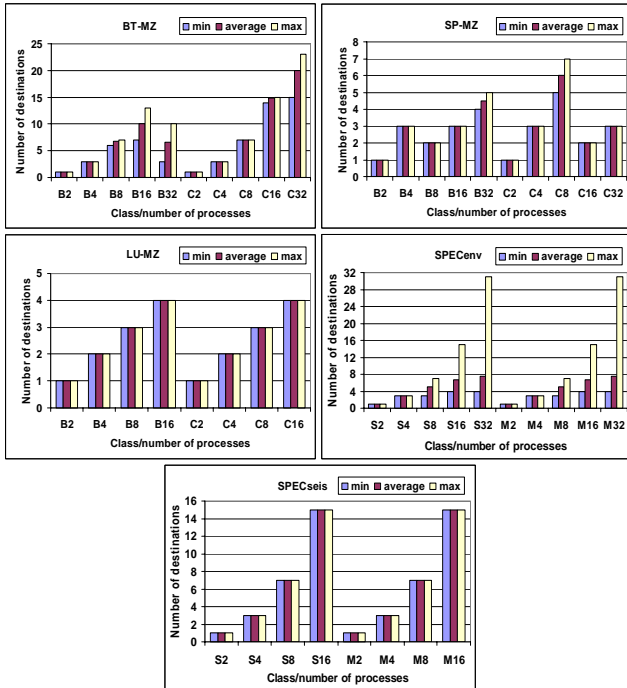


Figure 4. Number of message destinations per process.

As stated earlier, not all processes in the applications communicate with all other processes. Usually, process zero (root process) is responsible for distributing the data and verifying the results. This makes it a favorite destination for other processes. Figure 5 shows the distribution of message destinations for process zero of the applications running with 16 processes. The root process in the SP-MZ and LU-MZ only communicates with a subset of all other processes, while in the SPECenv, SPECseis, and the BT-MZ (class C) it communicates with all other processes. Interestingly, process zero in the SPECseis communicates uniformly with all other processes.

4.2 Collective Communication

Table 3 presents the type, frequency, and the payload (in byte) of the collective operations used in the applications studied in this paper. Broadcast, barrier and reduce are the only collective primitives used in these applications. The reduce primitive in the SPECenv uses the “sum” operation. NPB-MZ applications use the “sum”, as well as the “max” operation. SPECenv uses a large number of broadcast operations with very large payloads. Our finding for the SPEC collective characteristics is strikingly different from previous research on other applications [5].

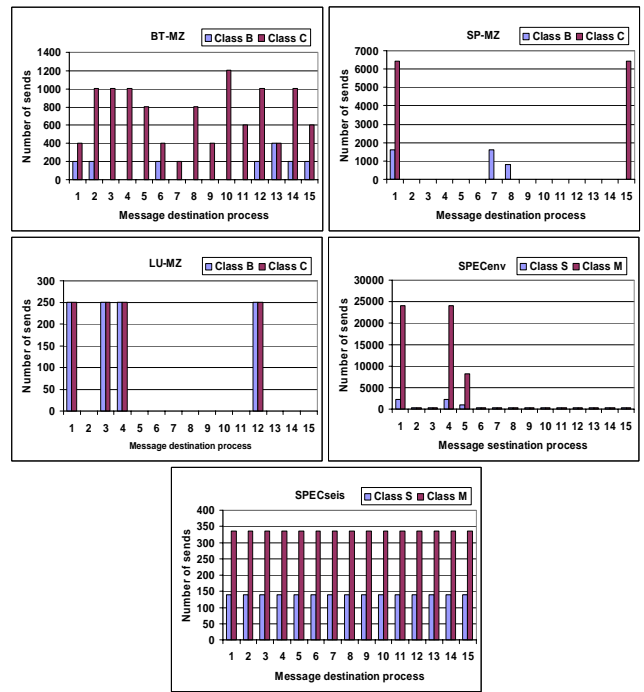


Figure 5. Distribution of message destinations for process 0 (16 processes).

5. Related Work

Various research groups have characterized the communication patterns of applications under different parallel programming paradigms [5-11]. Vetter and Mueller [5] presented the MPI point-to-point and collective communications as well as floating-point characteristics of some applications in the ASCI Purple suite, and the SAMRAI application. Kim and Lilja [6] quantified the characteristics of some kernels and applications in MPI and PVM as well as their execution times. They also introduced the concept of locality for send/receive communication calls using the LRU heuristics. Afsahi and Dimopoulos extended the notion of communication locality to the message destinations, and message reception calls using the LRU, LFU and FIFO policies [7]. They then devised different message predictors. Wong and his colleagues [8] studied the NPB benchmarks. Chodnekar and his associates [10] considered the inter-arrival time of messages, and message volume in message-passing and shared-memory applications. Karlsson and Brorsson [9] compared the communication patterns of some applications in SPLASH and NPB benchmarks under MPI and ThreadMark. Cypher and his colleagues [11] studied some application benchmarks that use explicit communication.

6. Conclusions and Future Work

This paper examined the MPI characteristics of small-to large-scale scientific applications in terms of their point-to-point and collective communications. We quantified the number of messages sent, the average message size per send, the cumulative distribution function of the message

sizes, the number of distinct message destinations, and the distribution of message destinations. For collective communications, we presented the type, frequency, and the payload. We also evaluated the impact of the problem size and the system size on the communication behaviour of the applications.

Table 3. Collective communications (per process).

Applications	# processes	#Broadcast & payload (byte)	#Reduce & payload (byte)	#Barrier
BT-MZ (B & C)	2-32	3 (96)	3 (704)	2
SP-MZ (B & C)	2-32	3 (96)	3 (704)	2
LU-MZ (B & C)	2-16	7 (512)	4 (768)	2
SPECseis (S & M)	2-16	38 (23312)	-	20
SPECenv (S)	2-32	946 (6631224)	1 (16)	-
SPECenv (M)	2-32	2247 (102597148)	1 (16)	-

We found that the applications studied have diverse communication characteristics. Those include very small to very large messages, frequent to infrequent messages, various distinct message sizes, set of favourite destinations, and regular versus irregular communication patterns. Some applications are sensitive to the bandwidth of the interconnect, while others are latency-bound as well. Our evaluation also revealed that most applications are sensitive to the changes in the system size and the problem size. We discovered all applications use only a few collective operations. However, SPEC applications use them frequently with very large payloads. Overall, the information provided in this work will help system designers, application developers, and library/middleware designers to better understand the current and future communication workloads of parallel applications.

This study verifies that message-passing applications communicate intensively. Therefore, they will benefit from improvements in the interconnect hardware and their features as well as the communication system software and libraries. Collective communications such as broadcast, barrier, and reduce are expensive operations. Thus, it is essential to optimize their implementation in hardware and/or software in the future computer systems.

We intend to continue our study by gathering the locality characteristics as well as the timing profiles of the applications in our cluster. We would like to study the relationship between the application characteristics and their performance taking into account the features of the interconnect such as the parameters of the LogP model, and the buffer reuse impact on performance, as shown in [14].

Acknowledgements

We would like to thank the anonymous reviewers for their comments and suggestions. This work was supported

by grants from the Natural Sciences and Engineering Research Council of Canada (NSERC), Canada Foundation for Innovation (CFI), and Ontario Innovation Trust (OIT).

References

- [1] MPI: A Message Passing Interface Standard, Version 1.2, Message Passing Interface Forum, 1997.
- [2] R.F. Van der Wijngaart & H. Jin. NAS Parallel Benchmarks, Multi-Zone Versions, *NAS Technical Report NAS-03-010*, NASA Ames Research Center, 2003.
- [3] SPEC HPC2002 suite, (<http://www.spec.org/hpc2002/>).
- [4] OpenMP C/C++ Application Programming Interface, Version 2.0, 2002, (<http://www.openmp.org/specs/>).
- [5] J.S. Vetter & F. Mueller, Communication Characteristics of large-scale scientific applications for contemporary cluster architectures, *Journal of Parallel and Distributed Computing* 63, 2003, 853-865.
- [6] J. Kim & D.J. Lilja, Characterization of Communication Patterns in Message-Passing Parallel Scientific Application Programs, *Proc. CANPC'98, Workshop on Communication, Architecture, and Applications for Network-based Parallel Computing*, 1998.
- [7] A. Afsahi & N.J. Dimopoulos, Efficient Communication Using Message Prediction for Clusters of Multiprocessors, *Concurrency and Computation: Practice and Experience*, 14(10) 2002, 859-883.
- [8] F.C. Wong, R.P. Martin, R.H. Arpaci-Dusseau & D.E. Culler, Architectural requirements and scalability of the NAS parallel benchmarks, *Proc. 1999 ACM/IEEE conference on Supercomputing*, 1999.
- [9] S. Karlsson, & M. Brorsson, A Comparative Characterization of Communication Patterns in Applications using MPI and Shared Memory on an IBM SP2, *Proc. CANPC'98, Workshop on Communication, Architecture, and Applications for Network-based Parallel Computing*, 1998.
- [10] S. Chodnekar, V. Srinivasan, A. Vaidya, A. Sivasubramanian, & C. Das, Towards a Communication Characterization Methodology for Parallel Applications, *Proc. 3rd International Conference on High Performance Computer Architecture*, 1997, 310- 321.
- [11] R. Cypher, A. Ho, S. Konstantinidou, & P. Messina, Architectural Requirements of Parallel Scientific Applications with Explicit Communication, *Proc. 20th International Symposium on Computer Architecture*, 1993.
- [12] D.H. Bailey, T. Harsis, W. Saphir, R.V. der Wijngaart, A. Woo, & M. Yarrow, The NAS parallel benchmarks 2.0: *NAS technical report NAS-95-020*, NASA Ames Research Center, 1995.
- [13] W. Gropp, E. Lusk, N. Doss, & A. Skjellum, A High-Performance, Portable Implementation of the MPI Message Passing Interface Standard, *Parallel Computing*, 22(6), 1996, 789-828.
- [14] Y. Qian, A. Afsahi, & R. Zamani, Myrinet Networks: A Performance Study, *Proc. NCA04, 3rd IEEE International Symposium on Network Computing and Applications*, 2004, 323-328.