Total Exchange Performance Modelling under Network Contention

Luiz Angelo Barchet Steffenel, Grégory Mounié

Total Exchange Performance Modelling under Network Contention

Luiz Angelo Barchet-Steffenel* and Grégory Mounié

ID - IMAG Laboratory, MOAIS Team**
ZIRST 51, Avenue Jean Kuntzmann
F-38330 Montbonnot St. Martin, France
{Luiz-Angelo.Steffenel,Gregory.Mounie}@imag.fr

Abstract. One of the most important collective communication patterns for scientific applications is the *many to many*, also called *complete exchange*. Although efficient All-to-All algorithms have been studied for specific networks, general solutions like those found in well known MPI distributions are strongly influenced by the congestion of network resources. In this paper we present our approach to model the performance of the All-to-All collective operation. Our approach consists in identifying a contention factor that characterises the network environment, and using it to augment a contention-free communication model. This approach allows an accurate prediction of the performance of the All-to-All operation over different network environments with a small cost. Indeed, we demonstrate the accuracy of our approach by presenting our experiments with three different network environments, Fast Ethernet, Giga Ethernet and Myrinet.

1 Introduction

One of the most important collective communication patterns for scientific applications is the *many to many* (also called *complete exchange* [1]), in which each process holds P different data items that should be distributed among the P processes, including itself. An important example of this communication pattern is the All-to-All collective operation, where all messages have the same size m. The All-to-All operation is frequently used for matrix transposition, two-dimensional Fourier Transform, conversion between storage schemes (remapping of arrays in HPF compilers), shuffle permutation, N body problems and matrix-vector multiplication.

Although efficient All-to-All algorithms have been studied for specific networks structures like meshes, hypercubes, tori and circuit-switched butterflies, general solutions like those found in well-known MPI distributions rely on direct point-to-point communication among the processes. Because all communications are started simultaneously, the overall communication time of the MPI_AlltoAll operation is strongly influenced by the congestion of network resources.

* Supported by grant BEX 1364/00-6 from CAPES - Brazil
** This project is supported by CNRS, INPG, INRIA and UJF

Consequently, the performance modelling of the All-to-All operation is not a simple task. Indeed, most existing performance models are unable to reflect the impact of network contention, while others are too complex to be used in real situations.

In this paper we present a new approach to model the performance of the All-to-All collective operation. Our strategy consists in identifying a *contention factor* that characterises the network environment, and using it to augment a contention-free performance model. This approach allows the prediction of the performance of the All-to-All operation with efficiency and reduced cost. Indeed, to demonstrate our approach, we present the results we obtained with three different network environments, Fast Ethernet, Giga Ethernet and Myrinet.

The rest of this paper is organised as follows: Section 2 presents the definitions and the test environment we will consider along this paper. Section 3 presents a survey of performance modelling under communication contention. Section 4 presents our approach to model the performance of the All-to-All operation, validating it against experimental data. Finally, Section 5 presents our conclusions as well as the future directions of our work.

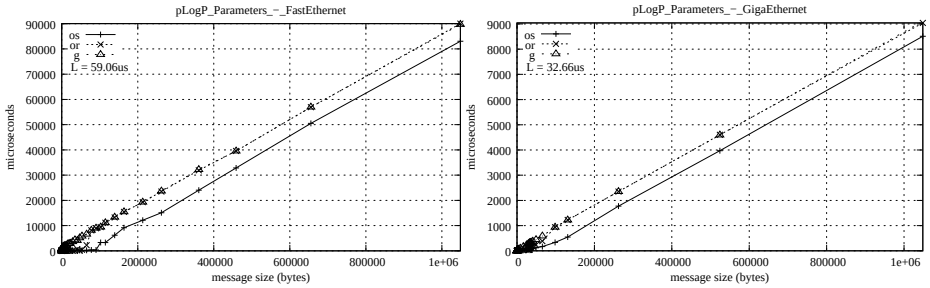
2 Performance Models and System Definitions

There are several performance models adequate to represent message-passing parallel programs, most of them based on *delay* [2], BSP [3] or LogP [4]. Although these last two performance models are equivalent in most circumstances [5], LogP is slightly more adapted to our problem as it includes the notion of finite network capacity, which is especially useful to reflect the network contention. Hence, in this paper we model collective communications using the *parameterised LogP* model (*pLogP*) [6], an extension of the LogP performance model that can accurately handle both small and large messages with a low complexity.

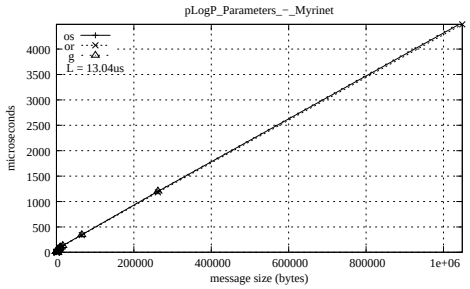
All along this paper we use $\boldsymbol{g(m)}$, $\boldsymbol{os(m)}$ and $\boldsymbol{or(m)}$ to respectively represent the communication gap, the send and the receive overheads of a message of size m, $\boldsymbol{L}$ as the communication latency between two nodes, and $\boldsymbol{P}$ as the number of nodes involved in the operation. The *pLogP* parameters used to feed our models were obtained with the MPI LogP Benchmark tool [7], and are presented in Figure 1.

The experiments were conducted on the **icluster-2**[1] cluster at the INRIA Rhône-Alpes computing centre and on the **IDPOT**[2] cluster at the ID-IMAG Laboratory. The icluster-2 contains 104 Bi-Itanium2 machines (900MHz, 3GB, Red Hat AS 3.0 with kernel 2.4.21smp) interconnected by a switched Fast Ethernet and a Myrinet network. The IDPOT cluster contains 48 Bi-Xeon machines (2.5GHz, 1.5GB, Debian with kernel 2.4.26smp) interconnected by a Giga Ethernet network. The experiments used LAM-MPI 7.0.4 and consisted on 100 measures for each set of parameters (message size, number of processes), from which the average values are considered in this paper.

[1] *http://i-cluster2.inrialpes.fr/*
[2] *http://idpot.imag.fr/* or *http://frontal38.imag.fr*



(a) Fast Ethernet (b) Giga Ethernet



(c) Myrinet

Fig. 1. pLogP parameters for the icluster-2 and IDPOT networks

3 Modelling the All-to-All Operation

In the *All-to-All* operation, every process holds $m \times P$ data items that should be equally distributed among the P processes, including itself. Because general implementations of the All-to-All collective communication rely on direct point-to-point communication among the processes, the network can easily become saturated, and by consequence, degrade the communication performance. Indeed, Chun and Wang [8] demonstrated, that the overall execution time of intensive exchange collective communications are strongly dominated by the network contention and congestive packet losses, two aspects that are very difficult to quantify. As a result, a major challenge on modelling the communication performance of the All-to-All operation is to represent the impact of network contention.

Unfortunately, most communication models like those presented by [1] are simple extensions of the *one-to-many* communication pattern (e.g. the Scatter operation, where a single process sends different messages of size m to each other process). By consequence, such models do not take into account the influence of the network contention, and therefore, are not accurate enough to predict the performances of an All-to-All operation.

Indeed, the development of contention-aware communication models is relatively recent, as shown by Grove [9]. For instance, one of the first models that considered the effects of resource contention was presented by Adve [10]. This

model considers that the total execution time of a parallel program is the sum of four components, namely:

$$T = t_{computation} + t_{communication} + t_{resource-contention} + t_{synchronisation}$$

While conceptually simple, this model was non-trivial in practice because of the non-deterministic nature of resource contention, and because of the difficulty to estimate average synchronisation delays.

In fact, the non-deterministic behaviour of the network contention is a major obstacle to modelling communication performance. A proposal to circumvent this limitation was introduced by Clement *et al.* [11], which suggested a way to account contention in shared networks such as non-switched Ethernet consisting in a contention factor γ that augments the linear communication model T:

$$T = l + \frac{b\gamma}{W}$$

where l is the link latency, b is the message size and W is the bandwidth of the link, and γ is equal to the number of processes. A restriction on this model is that it assumes that all processes communicate simultaneously, which is only true for a few collective communication patterns. Anyway, in the cases where this assumption holds, they found that this simple contention model greatly enhanced the accuracy of their predictions for essentially zero extra effort.

The principle of a contention factor is complemented by the work of Labarta et *al.* [12], that tried to approximate the behaviour of the network contention by considering that if there are m messages ready to transit, and only b available buses, then the messages are serialised in $\lceil \frac{m}{b} \rceil$communication waves.

Most recently, some works on contention-aware performance models have been published. LoGPC [13] is an extension of the LogP model that tries to determine the impact of network contention through the analysis of k-ary n-cubes. Unfortunately, the complexity of this analysis makes too hard the application of such model in practical situations.

Another approach was presented by Chun [8], in which the contention is considered as a component of the communication latency, and by consequence, their model uses different latency values according to the message size. Although easier to use than LoGPC, the model from Tam does not take into account the number of communicating processes, which is clearly related to the occurrence of network contention.

4 A Different Approach

Similarly to Clement *et al.* [11], we assume that the contention is sufficiently linear to be modelled. Our approach, however, consists on identifying theoretical performance bounds for the All-to-All operation and deriving a contention factor that fits our predictions with pre-existent experimental results. We consider that the network contention depends mostly on the physical characteristics of the

network (network cards, links, switches), and consequently, the ratio between the theoretical bounds and the practical results represents a "*contention signature*" of the network. Once identified the *signature* of a network, it can be used in further experiments to predict the communication performance.

In the case of the All-to-All operation, we explore the limitations of the *1-port* communication model. For instance, although a process can only send a message to a process each time, the *1-port* model allows a process to simultaneously send a message to one process and receive a message from another one. Hence, in a contention-free situation, a process would be able to access the network interface as soon as the precedent *send* operation returned (while the *receive* operation runs simultaneously in the background). In terms of pLogP parameters, this means that a contention-free process needs only g time units to simultaneously send a message to a process and receive a message from another one.

At the other hand, processes subjected to network contention may not be able to send and receive messages simultaneously. Due to the congestion of network resources, a process may not be able to overlap send and receive, and therefore, can be forced to serialise its communications. In pLogP terms, such processes need g time units to send each message, plus or time units to receive a message.

By consequence, a *Contention-Free* situation represents the capability to overlap send and receive operations with no extra cost, while in a *Contention* situation the processes need to serialise their transmissions due to the network contention. Thus, we model the All-to-All operation using these two situations as represented on Table 1. It worth noting that in real situations the performances of the All-to-All operation may exceed the predictions for the *Contention* case, as there are other factors that can influence the communications besides the physical environment. Even though, by defining a network signature based on the theoretical bounds, we are able to quantify the network contention effects regardless the sources of contention.

	Communication Models
Under Contention	$(P-1) \times g(m) + (P-1) \times or(m) + L$
Contention-free	$(P-1) \times g(m) + L$

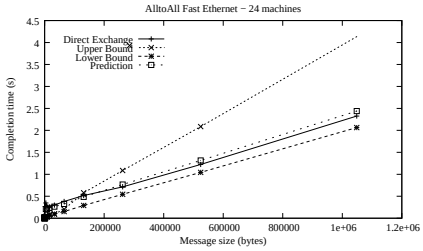
Table 1. Theoretical performance bounds for the *All-to-All* operation

4.1 Practical Results

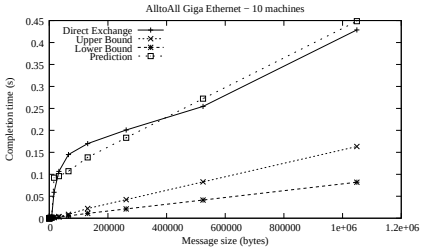
To illustrate our approach to predict the performance of the All-to-All operation in an environment subjected to network contention, we use a *Direct Exchange (DE)* algorithm similar to the current MPI_AlltoAll implementation from both LAM 7.0.4 and MPICH 1.2.5. In this algorithm, all nodes start to communicate simultaneously, but the contact list of each node is rotated to avoid overloading a single process each "round".

We present in Figure 2 an example with the measured performance for the *DE* algorithm as well as the predicted performance bounds. It can be observed that

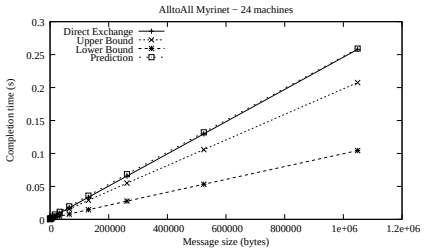
the completion time for the *DE* algorithm usually differs from the *Contention-free* case in a non-negligible amount, which clearly indicates the presence of network contention.



(a) Fast Ethernet



(b) Giga Ethernet



(c) Myrinet

Fig. 2. Theoretical bounds and performance predictions

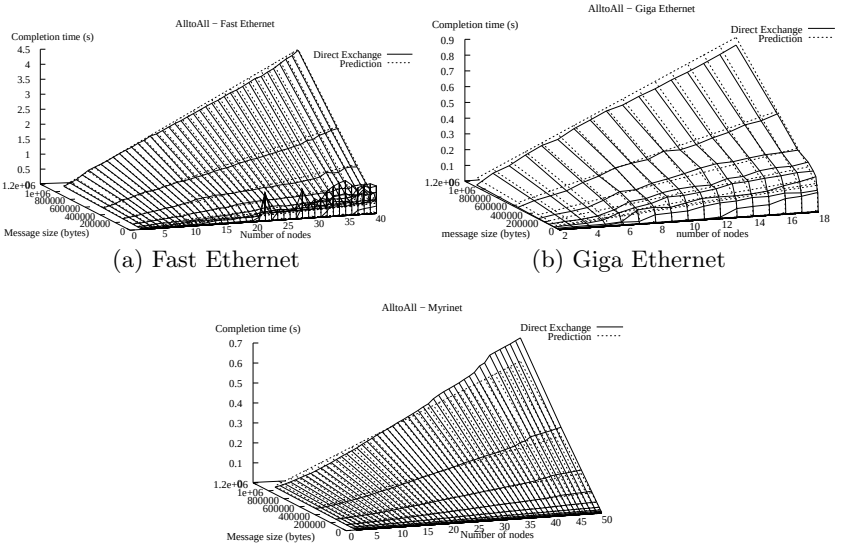
Next, we determine a contention factor γ between the predicted *Contention-free* and *Contention* performances such that the predictions fit the performance of the All-to-All operation. This contention factor γ is constant and depends only on the network characteristics (the *network signature*), whilst the *Contention-free* and *Contention* bounds depend on the number of processes and on the message size. In some cases, a supplementary factor δ, dependent on the number of processes P, may be necessary to represent additional costs like, for example, the overhead of message segmentation, buffer overflow or residual synchronisation delays. Hence, a performance model for the All-to-All operation can be defined by:

$$\begin{aligned} T &= Free + (Contention - Free) \ \times \gamma + (P-1) \times \delta \\ &= (P-1) \times g(m) + L + (P-1) \times or(m) \ \times \gamma + (P-1) \times \delta \\ &= (P-1) \times (g(m) + or(m) \ \times \gamma + \delta) + L \end{aligned}$$

Taking as basis the data from Figure 2, a contention factor γ that fits those performances is $\gamma = \frac{1}{10}$ for the Fast Ethernet network, $\gamma = \frac{9}{2}$ For the Giga Ethernet network and $\gamma = \frac{3}{2}$ for the Myrinet network. In the case of Fast Ethernet and Giga Ethernet networks we need a supplementary δ_P for messages larger than

2kB, mostly due to reception buffers overflow. Hence, we approximate the behaviour of Fast Ethernet networks with $\delta_P = 7ms * P$ while Giga Ethernet needs $\delta_P = 9ms * P$. Using these contention factor values as the *network signatures* of our clusters, we could accurately predict the performance of the All-to-All operation for a wide range of processes with no extra cost. Therefore, Figure 3 presents a comparison between our predictions and the measured performances for the All-to-All operation with both Fast Ethernet, Giga Ethernet and Myrinet networks.

It also worth noting the instabilities observed in the case of the Fast Ethernet network when dealing with small messages and a large number of processes. We believe that these instabilities are due to a problem with the TCP implementation on Linux, as previously discussed in a precedent work [14].



(a) Fast Ethernet (b) Giga Ethernet

(c) Myrinet

Fig. 3. Performance predictions for the *All-to-All* operation

5 Conclusions and Future Work

In this paper we present a new approach to model the performance of the All-to-All collective operation that is both simple and precise. Our method consists on identifying a contention factor that characterises the network environment, and using it to augment a contention-free performance model. This approach allows the prediction of the performance of the All-to-All operation over different network environments, with accuracy and reduced cost. Indeed, to demonstrate our approach, we present the results we obtained with three different network environments, Fast Ethernet, Giga Ethernet and Myrinet.

We intend to pursue our experiments on communication modelling using the GRID5000[3] facility, investigating the behaviour of collective communications with a larger number of machines and with other network environments such as InfiniBand, and more specifically to the complete exchange operations, to automate the definition of γ and δ for a given network. We are also interested in the study of contention effects in the domain of small messages, subjected to important performance variations as represented by the δ factor itself.

References

1. Christara, C., Ding, X., Jackson, K.: An efficient transposition algorithm for distributed memory computers. In: High Performance Computing Systems and Applications. (1999) 349–368
2. Rayward-Smith, V.J.: UET scheduling with unit interprocessor communication delays. Discrete Applied Mathematics **1**(18) (1987) 55–71
3. Valiant, L.G.: A bridging model for parallel computation. Communications of the ACM **33**(8) (1990) 103–111
4. Culler, D., Karp, R., Patterson, D., Sahay, A., Schauser, K.E., Santos, E., Subramonian, R., von Eicken, T.: LogP - a practical model of parallel computing. Communication of the ACM **39**(11) (1996) 78–85
5. Skillicorn, D., Hill, J., McColl, W.: Questions and answers about BSP. Scientific Programming **6**(3) (1997) 249–274
6. Kielmann, T., Bal, H., Gorlatch, S., Verstoep, K., Hofman, R.: Network performance-aware collective communication for clustered wide area systems. Parallel Computing **27**(11) (2001) 1431–1456
7. Kielmann, T., Bal, H., Verstoep, K.: Fast measurement of LogP parameters for message passing platforms. In: 4th Workshop on Runtime Systems for Parallel Programming. LNCS Vol. 1800 (2000) 1176–1183
8. Chun, A.T.T., Wang, C.L.: Contention-aware communication schedule for high-speed communication. Cluster Computing: The Journal of Networks, Software Tools and Application **6**(4) (2003) 337–351
9. Grove, D.: Performance Modelling of Message-Passing Parallel Programs. PhD thesis, University of Adelaide (2003)
10. Adve, V.: Analysing the Behavior and Performance of Parallel Programs. PhD thesis, University of Wisconsin, Computer Sciences Department (1993)
11. Clement, M., Steed, M., Crandall, P.: Network performance modelling for PM clusters. In: Proceedings of Supercomputing. (1996)
12. Labarta, J., Girona, S., Pillet, V., Cortes, T., Gregoris, L.: DiP: A parallel program development environment. In: 2nd Euro-Par Conference. Volume 2. (1996) 665–674
13. Moritz, C.A., Frank, M.I.: LoGPC: Modeling network contention in message-passing programs. IEEE Transactions on Parallel and Distributed Systems **12**(4) (2001) 404–415
14. Barchet-Estefanel, L., Mounié, G.: Fast tuning of intra-cluster collective communications. In: Proceedings of the Euro PVM/MPI 2004. LNCS Vol. 3241 (2004) 28–35

[3] *http://www.grid5000.org*