HAL
open science

Modeling Network Contention Effects on All-to-All Operations

Luiz Angelo Steffenel

To cite this version:

Luiz Angelo Steffenel. Modeling Network Contention Effects on All-to-All Operations. 2006, pp.CD-ROM. hal-00089242

HAL Id: hal-00089242

https://hal.science/hal-00089242v1

Submitted on 16 Aug 2006

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Modeling Network Contention Effects on All-to-All Operations

Luiz Angelo Steffenel
Université Nancy-2, LORIA, AlGorille Team
LORIA - Campus Scientifique - BP 239
F-54506 Vandoeuvre-lès-Nancy Cedex, France
barchet@loria.fr

Abstract

One of the most important collective communication patterns used in scientific applications is the complete exchange*, also called* All-to-All*. Although efficient* complete exchange *algorithms have been studied for specific networks, general solutions like those available in well-known MPI distributions (e.g. the MPI_Alltoall operation) are strongly influenced by the congestion of network resources. In this paper we present an integrated approach to model the performance of the All-to-All collective operation. Our approach consists in identifying a contention signature that characterizes a given network environment, 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 architectures with a small overhead. This approach is assessed by experimental results using three different network architectures, namely Fast Ethernet, Gigabit Ethernet and Myrinet.*

Keywords. *Network Contention, MPI, Collective Communications, Performance Modeling*

1. Introduction

One of the most important collective communication patterns for scientific applications is the *total exchange* [8] (also called *All-to-All*), in which each process holds n different data items that should be distributed among the n processes, including itself. An important example of this communication pattern is the All-to-All operation, where all messages have the same size m.

Although efficient All-to-All algorithms have been studied for specific networks structures like meshes, hypercubes, tori and circuit-switched butterflies [8][7][20][14], 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, architecture independent algorithms are strongly influenced by the saturation of network resources and subsequent loss of packets - the network contention.

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 signature* that characterizes a given network environment. Using such *contention signature*, we are able to accurately predict the performance of the All-to-All operation, with an arbitrary number of processes and message sizes. To demonstrate our approach, we present experimental results obtained with three different network architectures (Fast Ethernet, Gigabit Ethernet and Myrinet). We believe that this model can be extremely helpful on the development of application performance prediction frameworks such as PEMPIs [17], but also in the optimization of grid-aware collective communications (e.g.: LaPIe [3, 5] and MagPIe [15]).

This paper is organized as follows: Section 2 presents a survey of performance modeling under communication contention. In section 3 we discuss the impact of network contention on the performance of total exchange algorithms, presenting experimental data that exhibit and characterize this influence. Section 4 presents the network models used in this paper, and in section 5 we formalize the *total exchange* problem, as well as some performance lower bounds. In Section 6 we present a preliminary approach to model the performance of the All-to-All operation. This approach is extended in Section 7, where we propose a strategy to characterize the *contention signature* of a given network and for instance, to predict the performance of the All-to-All operation. Section 8 validates our model against experimental data obtained on three different network architectures (Fast Ethernet, Gigabit Ethernet and Myrinet). Finally, Section 9 presents some conclusions and the future directions of our work.

1-4244-0328-6/06/$20.00 ©2006 IEEE.

2. Related Works

In the *All-to-All* operation, every process holds $m \times n$ data items that should be equally distributed among the n processes, including itself. Because general implementations of the All-to-All collective communication rely on direct point-to-point communications among the processes the network can easily become saturated, and by consequence, degrade the communication performance. As a result, a major challenge on modeling 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 Christara [8] and Pjesivac-Grbovic [19] are simple extensions of the *one-to-many* communication pattern that do not take into account the potential effects of network contention. Indeed, these works usually represent the All-to-All operation as parallel executions of the Scatter operation, as presented by the expression below:

$$T = (n-1) \times (\alpha + \beta m) \tag{1}$$

The development of contention-aware communication models is relatively recent, as shown by Grove [12], mostly because of the non-deterministic behavior of the network contention. To circumvent these restrictions, some authors suggested a few techniques to adapt the existing models. As consequence, Bruck [6] suggested the use of a *slowdown factor* to correct the performance predictions. Similarly, Clement *et al.* [10] introduced a technique that 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:

$$T = l + \frac{b\gamma}{W} \tag{2}$$

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 enhanced the accuracy of their predictions for essentially zero extra effort.

The use of a contention factor was supported by the work of Labarta et *al.* [16], that tried to approximate the behavior of the network contention by considering that if there are m messages ready to be transmitted, and only b available buses, then the messages are serialized in $\lceil \frac{m}{b} \rceil$ communication waves.

Most recently, some works tried to design contention-aware performance models. For instance, LoGPC [18] presents 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 to include contention-specific parameters in the performance models was presented by Chun [9]. In his work, the contention is considered as a component of the communication latency, and by consequence, his model uses different latency values according to the message size. Although easier to use than LoGPC, the model from Chun does not take into account the number of messages passing in the network nor the link capacity, which are clearly related to the occurrence of network contention.

3. Impacts of Network Contention

The simplest approach to implement the All-to-All operation, called here *Direct Exchange*, considers that each process communicates directly with each other one. This strategy is currently used to implement the *MPI_Alltoall* operation in both LAM-MPI[1] and MPICH[2] libraries. In this strategy, communications are scheduled in successive rounds where each process p_i sends a message to a process p_j, whilst receiving a message from p_k, as described in Algorithm 1.

Algorithm 1 Direct Exchange Algorithm

```
for t=1 to n-1 do
   do in parallel for all i (0 ≤ i < n)
      p_i sends the message addresses to p_{i+t mod n}
      p_i receives the message from p_{i-t mod n}
```

To prevent the overloading of a single receiver, this technique rotates destination processes at each round. Nevertheless, our preliminary experiences (described in detail in [4]), suggest that the overload of the receiver is not enough to induce additional resource contention. Thus, the performance slowdown observed during the execution of the All-to-All operation is almost exclusively due to the saturation of the network, which causes packet loss. This observation is corroborated by the work of Grove [12], who already pointed out that contention originates mostly because of network overload, which forces message drops on bottleneck devices (switches, routers, etc.).

In order to better evaluate the presence of network contention on local area networks, we conducted some

[1] *http://www.lam-mpi.org*
[2] *http://www-unix.mcs.anl.gov/mpi/mpich1/*

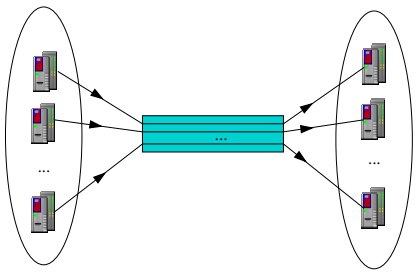

Figure 1. Stressing a broadband wide-area connection

experiences to stress the network. This approach is usually employed to measure the effective bandwidth of broadband wide-area connections, as presented by Fig. 1: several point-to-point connections are started simultaneously, flooding the link. As the TCP/IP protocol tries to evenly share the bandwidth among the connections, computing the aggregate throughput allows us to determine the effective bandwidth of the network link, or in our case, the overload caused by the contention.

Indeed, we evaluate the average bandwidth through the opening of several point-to-point connections in a Gigabit Ethernet network. We compute the aggregate bandwidth allocated to these connections during the transmission of large data files (32 MB), and gradually increasing the number of simultaneous point-to-point connections to saturate the network. In a preliminary analysis, we observe that the average throughput is drastically reduced, as presented in Fig. 2.

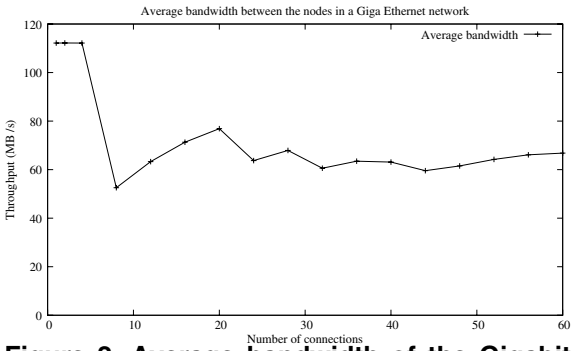

Figure 2. Average bandwidth of the Gigabit Ethernet network when simultaneous connections send a 32 MB message

Indeed, when analyzing the time each individual connection needs to send this 32MB message, as present in Fig. 3, we observe that connections do not behave identically. Actually, most connections finish their transmission in a reasonable time (as the average *completion time* indicates), but some point-to-point connections require almost six times longer to finish their transmission. This behavior can be explained by a recurrent phenomenon of packet loss that affects a reduced number of connections. Indeed, the slowdown observed in some connections is mostly related to the time required to detect the loss of TCP packets and their subsequent retransmission, as explained by Grove [12].

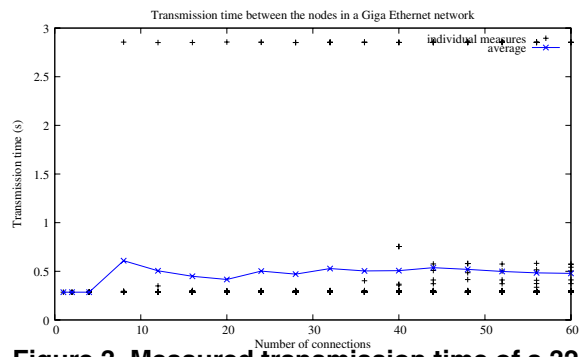

Figure 3. Measured transmission time of a 32 MB message in a Gigabit Ethernet network

4. Network Models Definition

In this section we present the communication, transmission, and synchronization models used in this work. We assume that the network is fully connected. These models can be used to approximately model most current parallel machines with distributed memory.

Communication Model: The links between pairs of processes are bidirectional, and each process can transmit data on at most one link and receive data on at most one link at any given time.

Transmission Model: We use Hockney's notation [13] to describe our transmission model. Therefore, the time to send a message of size $w_{i,j}$ from a process p_i to another process p_j, is $\alpha + w_{i,j}\beta$, where α is the start-up time (the latency between the processes) and $\frac{1}{\beta}$ is the bandwidth of the link. As in this paper we assume that all links have the same latency and bandwidth, and because we only investigate the regular version of the MPI_Alltoall operation where all messages have the same size m, $\forall i, \forall j, w_{i,j} = m$, and therefore the time to send a message from a process p_i to a process p_j is $\alpha + m\beta$.

Synchronization Model: We assume an asynchronous communication model, where transmissions from different processes do not have to start at the

same time. However, all processes start the algorithm simultaneously. This synchronization model corresponds to the execution of the MPI_Alltoall operation, used as reference in this work.

The total time for an algorithm is the difference between the start time and the time at which all processes are finished. We consider that message splitting is not allowed, then a message can only be sent in a single transmission.

There are also two possibilities for message forwarding: either messages are transmitted directly from the source to the destination, or messages are forwarded along a path of intermediate processes in a store-and-forward manner. When using store-and-forward, the entire message must be received at each intermediate node before forwarding it. Because the store-and-forward approach only behaves well for situations where the latency dominates the bandwidth [11], which usually is not the case, we consider only direct connections between the source and the destination process.

5. Problem Definition

In the *total exchange* problem, n different processes hold each one n data items that should be evenly distributed among the n processes, including itself. Because each data item has potentially different contents and sizes according to their destinations, all processes engage a total exchange communication pattern. Therefore, a total exchange operation will be complete only after all processes have sent their messages to their counterparts, and received their respective messages.

Formally, the total exchange problem (TEP for short) can be described using a weighted digraph $dG(V, E)$ of order n with $V = \{p_0, ..., p_{n-1}\}$. This digraph is called a message exchange digraph or MED for short. In a MED, the vertices represent the process nodes, and the arcs represent the messages to be transmitted. An integer $w(e)$ is associated with each arc $e = (p_i, p_j)$, representing the size of the message to be sent from process p_i to process p_j. Note that there is not necessarily any relationship between a MED and the topology of the interconnection network.

The port capacity of a process for transmission is the number of other processes to which it can transmit simultaneously. Similarly, the port capacity for reception is the number of other processes from which it can receive simultaneously. We will concentrate on the performance modeling problem with all port capacities restricted to one for both transmitting and receiving. This restriction is well-known in the literature as *1-port full-duplex*.

5.1. Notation and lower bounds

In this section, we present theoretical bounds on the minimum number of communications and on the bandwidth for the general message exchange problem. The number of communications determines the number of start-ups, and the bandwidth depends on the message weights.

Given a MED $dG(V; E)$, we denote the in-degree of each vertex $p_i \in V$ by $\Delta_r(p_i)$, and the out-degree by $\Delta_s(p_i)$. Let $\Delta_r = \max_{p_i \in V}\{\Delta_r(p_i)\}$ and $\Delta_s = \max_{p_i \in V}\{\Delta_s(p_i)\}$.

Since our model does not assume any additional overhead to provide synchronization, we can compute the following straightforward bound on the number of start-ups.

Claim 1. *The number of start-ups needed to solve a message exchange problem on a digraph $dG(V; E)$ without message forwarding is at least* $\max(\Delta_s, \Delta_r)$.

Given a MED $dG(V, E)$, the bandwidth bounds are determined by two obvious bottlenecks for each vertex - the time for it to send its messages and the time for it to receive its messages. Each vertex p_i has to send messages with sizes $\{w_{i,j} \mid j = 0 \ldots n-1\}$. The time for all vertices to send their messages is at least $t_s = \max_i \sum_{j=0}^{n-1} w_{i,j}\beta$. Similarly, the time for all vertices to receive their messages is at least $t_r = \max_j \sum_{i=0}^{n-1} w_{i,j}\beta$.

Claim 2. *The time to complete a personalized exchange is at least* $\max\{t_s, t_r\}$.

We can combine the claims about the number of start-ups and the bandwidth when message forwarding is not allowed.

Claim 3. *If message forwarding is not allowed, and either the model is synchronous or both maxima are due to the same process, the time to complete a personalized exchange is at least* $\max(\Delta_s, \Delta_r) \times \alpha + \max\{t_s, t_r\}$.

Because in this paper we do not assume messages forwarding, the fan-in and fan-out of a process must be $(n-1)$. Further, as we consider messages to be the same size and the network to be homogeneous, we can simplify Claim 3 so that the following bound holds.

Proposition 1. *If message forwarding is not allowed, and all messages have size* m, *and both bandwidth and latency are identical to any connection between two different processes p_i and p_j, the time to complete a total exchange is at least* $(n-1) \times \alpha + (n-1) \times \beta m$.

Proof. The proof is trivial, as the time to complete a total exchange is at least the time a single process needs to send one message to each other process. ■

6. Throughput under Contention Approach

The simplest approach to model the performance of a communication pattern subjected to network contention is to obtain two different parameter sets: one for the "contention-free" situations and one for the contention situations. This approach was partially employed in Chun's [9] work, who suggested the use of a latency parameter L that depends on the message size (and therefore, on the contention this message may cause). Although interesting, that approach has a main drawback: it does not consider the number of simultaneous connection, which can induce network contention even if messages are relatively small.

Therefore, in this paper we initially consider a different approach to measure the impact of network contention using a *gap per byte* (β) parameter that presents two different states: a *contention gap* β_C and a *contention-free gap* β_F parameters, obtained from experimental measurements. These two parameters are used in different proportions to model the average *gap per byte* used in performance models.

Therefore, we use the values presented in Fig. 3 to obtain the parameters β_F and β_C ($8.502 \times 10^{-9} s/byte$ and $8.498189 \times 10^{-8} s/byte$, respectively). Supposing that at most one of each two connections will be delayed due to contention, we can apply a simple proportion

$$\beta = (1 - \rho) \times \beta_F + \rho \times \beta_C \quad (3)$$

with $\rho = 0.5$ to obtain a synthetic bandwidth value $\beta = 4.6742 \times 10^{-8} s/byte$. Using this value of β with the performance model from Proposition 1 gives the following approximation, as presented in Fig. 4. Please note how different are the predictions of the theoretical lower bound, which uses only the contention-free gap parameter.

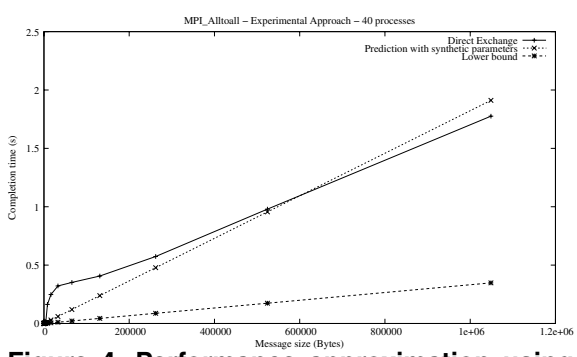

Figure 4. Performance approximation using two bandwidth parameters β_C and β_F

This approach, however, has several drawbacks. First, it requires a more complex procedure to measure the parameters β_F and β_C, as we shall saturate the network. Moreover, as we need to send large messages to stress the network, the average gap per byte may not correspond to the transmission time of small MPI messages, which are usually dominated by the "envelop" size instead of the message size. Indeed, as we observe in Fig. 4, the transmission cost of small messages increases rapidly, becoming linear only when messages are larger than 64 KB. Therefore, a better solution to model the performance of the All-to-All operation should keep the good aspects of this approach (the synthetic β parameter), while minimizing the measure cost and adapting the model to messages of different sizes. In the next section we present our proposal to cope with these aspects.

7. Contention Signature Approach

From the previous section we learn that measuring the parameter β from a saturate network allows us to predict the performance of the All-to-All operation, especially when message sizes are large. While this approach seems to be adequate from an experimental viewpoint, it has two main drawbacks, the cost to acquire the parameters β_C and β_F, and the inaccuracy in the case of small messages.

To cope with this problem and to model the contention impact on the performance of the All-to-All operation, we adopt an approach similar to Clement *et al.* [10], which considers the contention sufficiently linear to be modeled. Our approach, however, tries to identify the behavior of the All-to-All operation with regard to the theoretical lower bound (Proposition 1) on the *1-port* communication model. In our hypothesis, the network contention depends mostly on the physical characteristics of the network (network cards, links, switches), and consequently, the ratio between the theoretical lower bound and the real performance 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, provided that the network infrastructure does not change.

Initially, we consider communication in a contention-free environment. In this case, a process that sends messages of size m to $n-1$ processes needs at least $(n-1) \times \alpha + (n-1) \times m\beta$ time units. Further, by the properties of the *1-port* communication model, The total communication time of the All-to-All operation must be at least $(n-1) \times \alpha + (n-1) \times m\beta$ time units if all processes start communicating simul-

taneously, as stated by Proposition 1 (note that this model corresponds to the models used by Christara [8] and Pjesivac-Grbovic [19]).

In the case of the All-to-All operation, however, the intensive communication pattern tends to saturate the network, causing message delays and packet loss that strongly impact on the communication performance of this collective communication. In this network congestion situation, traditional models such as those presented by Christara [8] and Pjesivac-Grbovic [19] do not hold anymore, even if the communication pattern has not changed.

Therefore, our approach to model the performance of the MPI_Alltoall operation, in spite of the influence of network contention, consists on determining a *contention ratio* γ that express the relationship between the theoretical performance (lower bound) and the real completion time. For simplicity, we consider that this *contention ratio* γ is constant and depends exclusively on the network characteristics; however, this parameter is still related to the number of processes and the message sizes, as the lower bound depends on these values. Therefore, the simplest way to integrate this *contention ratio* γ in our performance model would be as follows:

$$T = ((n-1) \times (\alpha + m\beta)) \times \gamma \tag{4}$$

7.1. Non-linear aspects of the network contention

Although the performance model augmented by use of the *contention ration* γ proved to be quite accurate (see [1][2]), we observe nonetheless that some network architectures are still subject to performance variations according to the message size. To illustrate this problem, we present in Fig. 5, a detailed mapping of the communication time of the MPI_Alltoall operation in a Gigabit Ethernet network. We observe that the communication time does not increases linearly with the message size, but instead, present a non-linear behavior that prevents our model to accurately predict the performance when dealing with small messages.

Although in this paper we are not interested on determining the causes of these disturbances, there are three main scenarios that can explain this behavior: MPI sending policy, buffer capacity or synchronization of processes.

In spite of the real cause of this non-linearity, we propose an extension of the *contention ratio* model to better represent this phenomenon. Therefore, we augment the model with a new parameter δ, which depends on the number of processes but also on a given message size M. As a consequence, the association of a linear and an affine equation can define a more realistic performance model for the MPI_Alltoall operation, as follows:

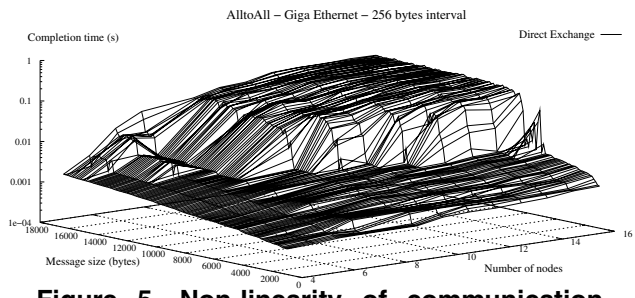

Figure 5. Non-linearity of communication cost with small messages

$$T = \begin{cases} ((n-1) \times (\alpha + m\beta)) \times \gamma & if\, m < M \\ ((n-1) \times ((\alpha + m\beta)) \times \gamma + \delta) & if\, m \geq M \end{cases} \tag{5}$$

8. Validation

To validate the approach proposed in this paper, this section presents our experiments to model the performance of MPI_Alltoall operation using three different network architectures, Fast Ethernet, Gigabit Ethernet and Myrinet. As previously explained, our approach consists on comparing the expected and real performance of the MPI_Alltoall operation, using as sample a predefined number of nodes n'; the relationship between these two measures allows us to define the γ and δ parameters. These two parameters, γ and δ, correspond to the "network contention signature" and allow us to accurately predict the performance of the MPI_Alltoall operation.

To obtain these parameters, we compare the sample data obtained from both theoretical lower bound and experimental measure, when varying the message size. Indeed, the lower bound comes from Proposition 1, with parameters α and β obtained from a simple point-to-point measure. The parameters γ and δ are obtained through a linear regression with the Generalized Least Squares method, comparing at least four measurement points in order to better fit the performance curve.

The different experiments presented in this paper represent the average of 100 measures for each set of parameters (message size, number of processes), and were conducted over two clusters of the Grid'5000 platform[3]:

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

The *icluster2* cluster, located at INRIA-Rhone-Alpes[4], composed of 104 dual Itamium2 nodes at 900 MHz. Three different networks interconnect *icluster2* nodes: a Fast Ethernet network (5 Fast Ethernet switches - 20 nodes per switch - interconnected by 1 Gigabit Ethernet switch), a Gigabit Ethernet network (not used in our experiments) and a Myrinet 2000 network (one 128 ports M3-E128 Myrinet switch). All machines have Red Hat Enterprise Linux AS release 3, with kernel version 2.4.21. In our tests we used LAM-MPI 7.1.2beta and the *gm* driver version 2.0.21.

The *GdX* (GriD'eXplorer) cluster, hosted by IDRIS[5] and operated by INRIA-Futurs[6] / LRI[7] teams. This cluster includes 216 nodes with dual AMD Opteron processors at 2 GHz and a Broadcom Gigabit Ethernet network. Software versions are: Debian Linux kernel 2.6.8 and LAM-MPI 7.1.2beta.

8.1. Fast Ethernet

In the case of the Fast Ethernet network, the measured completion time is just a little superior to the expected lower bound, as presented in Fig. 6. Indeed, this relatively small difference must be considered in the light of the retransmission policy: although the communication latency (and therefore the timeouts) is relatively small (around 60 μs), the reduced bandwidth of the links minimizes the impact of the retransmission of a lost packet. More important, we observe that the experimental measure behave like an affine equation, showing a start-up cost usually not considered by the traditional performance model; this start-up cost corresponds to the δ parameter proposed in our model.

From this data, we were able to calculate a *contention ratio* $\gamma = 1.0195$, which demonstrates that communication delays related to the loss of TCP packets are not the main factor that influences the performance in the case of the Fast Ethernet network. Instead, the most important factor in this case is the affine factor δ. From the same data, we determined $\delta = 8.23\,ms$ for messages larger than $M = 2\,kB$, which means that each simultaneous communication induces an overload of $8.23\,ms$ to the completion time of the All-to-All operation. Applying both γ and δ parameters we were able to approximate our predictions from the real measures as depicted by Fig. 6. Indeed, these parameters can be used to accurately predict the performance of the MPI_Alltoall operation with an arbitrary number of processes, we demonstrate in Fig. 7.

[4] *http://www.inrialpes.fr/i-cluster2/*
[5] *http://www.idris.fr/*
[6] *http://www-futurs.inria.fr/*
[7] *http://www.lri.fr/*

We observe indeed that our error rate is usually smaller than 10% when there are enough processes to saturate the network, as presented in Fig. 8, characterizing the application domain of our model (saturated networks).

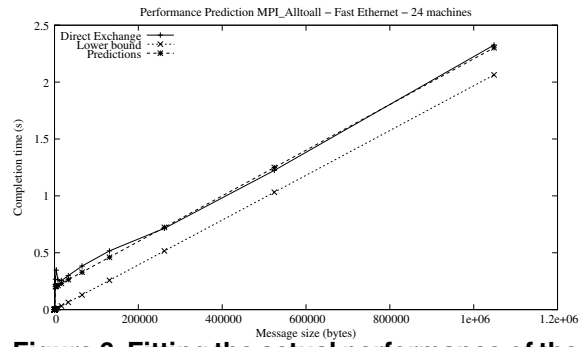

Figure 6. Fitting the actual performance of the MPI˙Alltoall operation

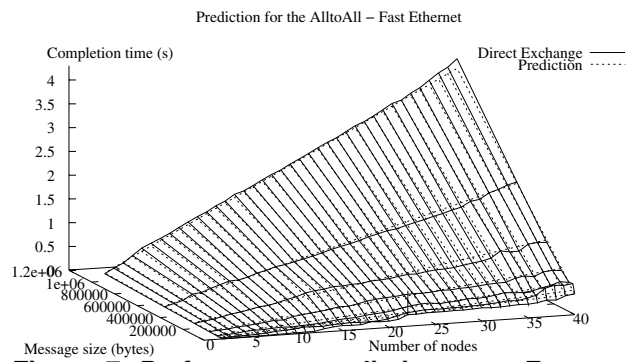

Figure 7. Performance prediction on a Fast Ethernet network

8.2. Gigabit Ethernet

In the case of the Gigabit Ethernet network, as presented in Fig. 9, we observe a clear difference between the theoretical lower bound and the measured values, much larger than in the case of the Fast Ethernet network. Indeed, the occurrence of network contention induces a retransmission delay that penalizes the completion time in a high transfer rate environment.

Another important analysis on the case of the Giga Ethernet network relates to the difference between the theoretical lower bound and the measured values. We observe that this difference is no more constant as observed with the Fast Ethernet, and varies according to the message size. Instead, messages with more than a few KB pay a considerable start-up cost, as predicted by the performance model we propose in Section 7.

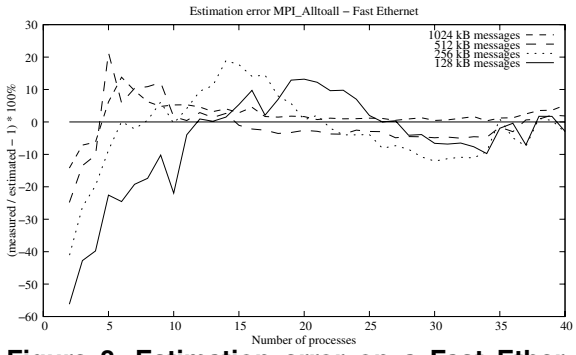

Figure 8. Estimation error on a Fast Ethernet network when varying the number of processes

To compute the *contention ratio* γ and a *start-up cost* δ, we use sample data for an arbitrary number of processes. Indeed, we chose in this example the results for an execution of the All-to-All operation with 40 processes (one by machine), as presented in Fig. 9. Using linear regression on these data we obtain $\gamma = 4,3628$ and $\delta = 4,93\,ms$ (to be used only for messages larger than $M = 8\,kB$). As a result, the performance predictions from our model correspond to the curve presented on Fig. 10. As in the case of the Fast Ethernet network, the error rate is quite small when the network becomes saturate, as presented in Fig. 11, even when we consider different message sizes.

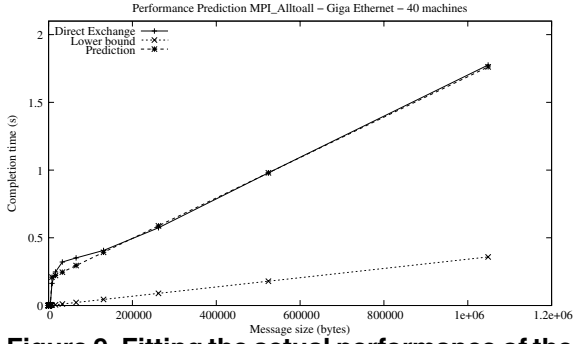

Figure 9. Fitting the actual performance of the MPI˙Alltoall operation

8.3. Myrinet

Although the two previous experiments give important proofs on the validity of our modeling method, they share many similarities on both network architecture and transport protocol (TCP/IP). To ensure that our method is not bounded to a specific infrastructure, we chose to validate our performance model also in

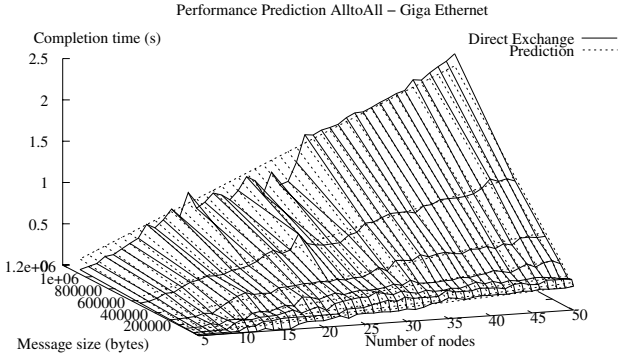

Figure 10. Performance prediction on a Gigabit Ethernet network

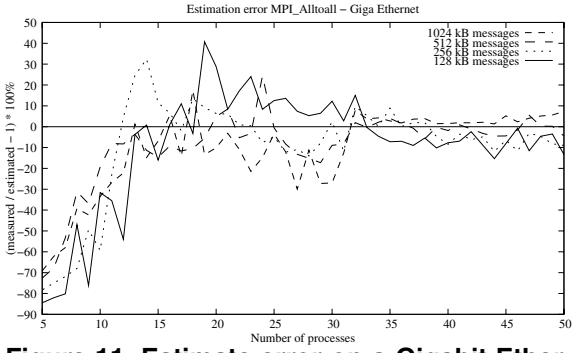

Figure 11. Estimate error on a Gigabit Ethernet network when varying the number of processes

a Myrinet network, using the *gm* transport protocol. Because of the *Myrinet+gm* stack differs considerably from the Ethernet+TCP/IP stack, any systematic behavior introduced into our sampling data by these architectures should be exposed.

Therefore, Fig. 12 presents the completion time of the All-to-All operation with a group of 24 processes. We can observe that contention affects this network in a same way as in the previous experiments, even if the *start-up* cost for the Myrinet network is almost inexistent (one of the main characteristics of the *Myrinet+gm* stack).

Hence, we were able to fit the performance of a 24-processes All-to-All operation as presented in Fig. 12 using only the *contention ratio* $\gamma = 2,49754$ (as the linear regression pointed a *start-up cost* δ smaller than 1 microsecond). When applying this factor to an arbitrary number of machines, as presented in Fig. 13, we observe that our performance predictions hold with a reasonable error rate. Indeed, a close look at the error rate (Fig. 14) indicates that the network becomes really saturate only when there are more then 40 com-

municating processes, and therefore the observed error is not related to the model itself but to the choice of the sample data.

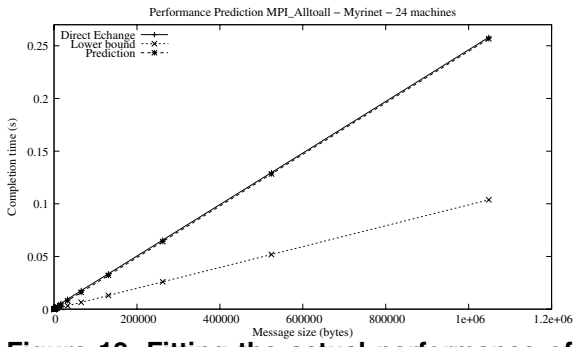

Figure 12. Fitting the actual performance of the MPI˙Alltoall operation

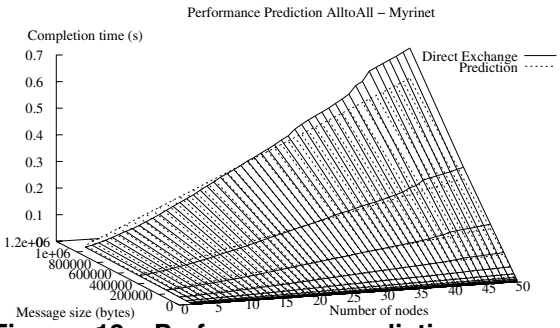

Figure 13. Performance prediction on a Myrinet network

9. Conclusions and Future Works

In this paper we address the problem of modeling the performance of *Total Exchange* communication operations, usually subject to important variations caused by network contention. Because traditional performance models are unable to predict the real completion time of an All-to-All operation, we try to cope with this problem by identifying the *contention signature* of a given network. In our approach, two parameters γ and δ are used to augment a linear performance model in order to fit the real performance of the MPI_Alltoall operation. Because these parameters characterize the network contention behavior and are independent of the number of communicating processes, they can be used to accurately predict the communication performance when there are enough communicating processes to saturate the network. Indeed, we demonstrate our approach through experiments conducted on three different network architectures, Fast Ethernet, Gigabit Ethernet and Myrinet.

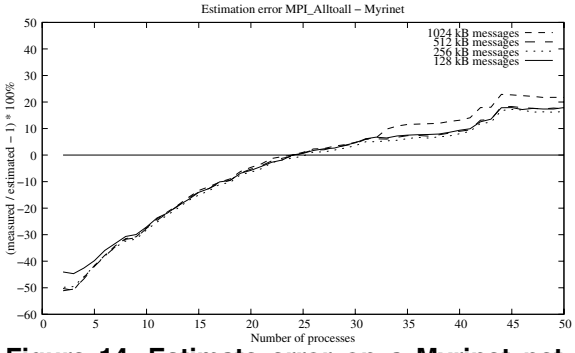

Figure 14. Estimate error on a Myrinet network when varying the number of processes

We intend to pursue our experiments on communication modeling using the GRID5000[8] facility, validating and extending our model under different network architectures like Infiniband. Indeed, we expect to extend our models to other collective communication operations, which are especially affected by contention when scaling up to a grid level. Further, we plan to investigate the contention modeling in the domain of small messages, which are still subjected to important performance variations despite the improvements from our performance model, and also to propose an intermediate performance model for half-saturate networks.

References

[1] L. Barchet-Estefanel and G. Mounié. Performance characterisation of intra-cluster collective communications. In *Proceedings of the 16th Symposium on Computer Architecture and High Performance Computing (SBAC-PAD 2004)*, pages 254–261, 2004.

[2] L. Barchet-Estefanel and G. Mounié. Prédiction de performances pour les communications collectives. In *Proceedings du 16ème Rencontre Francophone du Parallélisme (RenPar'16)*, pages 101–112, 2005.

[3] L. A. Barchet-Estefanel. *LaPIe: communications collectives adaptées aux grilles de calcul*. PhD thesis, INPG, France, 2005.

[4] L. Barchet-Steffenel and G. Mounié. Total exchange performance modelling under network contention. In *Proceedings of the 6th International Conference on Parallel Processing and Applied Mathematics*, LNCS Vol. 3911, pages 100–107, 2005.

[5] L. A. Barchet-Steffenel and G. Mounié. Scheduling heuristics for efficient broadcast operations on grid environments. In *Proceedings of the Performance Modeling, Evaluation and Optimization of Parallel and Dis-*

[8] *http://www.grid5000.fr*

tributed Systems Workshop - PMEO'06 (associated to IPDPS'06). IEEE Computer Society, 2006.

[6] J. Bruck, C.-T. Ho, S. Kipnis, E. Upfal, and D. Weathersby. Efficient algorithms for all-to-all communications in multiport message-passing systems. *IEEE Transactions on Parallel and Distributed Systems*, 8(11):1143–1156, November 1997.

[7] C. Calvin, S. Perennes, and D. Trystram. All-to-all broadcast in torus with wormhole-like routing. In *Proceedings of the IEEE Symposium on Parallel and Distributeed Processing*, pages 130–137, 1995.

[8] C. Christara, X. Ding, and K. Jackson. An efficient transposition algorithm for distributed memory computers. In *Proceedings of the High Performance Computing Systems and Applications*, pages 349–368, 1999.

[9] A. T. T. Chun. *Performance Studies of High-Speed Communication on Commodity Cluster*. PhD thesis, University of Hong Kong, 2001.

[10] M. Clement, M. Steed, and P. Crandall. Network performance modelling for PM clusters. In *Proceedings of Supercomputing*, 1996.

[11] A. Goldman, D. Trystram, and J. G. Peters. Exchange of messages of different sizes. *Journal of Parallel and Distributed Computing*, 66(1):1–18, 2006.

[12] D. Grove. *Performance Modelling of Message-Passing Parallel Programs*. PhD thesis, University of Adelaide, 2003.

[13] R. Hockney. The communication challenge for MPP: Intel paragon and meiko cs-2. *Parallel Computing*, 20:389–398, 1994.

[14] L. V. Kalé, S. Kumar, and K. Varadarajan. A framework for collective personalized communication. In *Proceedings of the International Parallel and Distributed Processing Symposium (IPDPS'03)*, 2003.

[15] T. Kielmann, H. Bal, S. Gorlatch, K. Verstoep, and R. Hofman. Network performance-aware collective communication for clustered wide area systems. *Parallel Computing*, 27(11):1431–1456, 2001.

[16] J. Labarta, S. Girona, V. Pillet, T. Cortes, and L. Gregoris. DiP: A parallel program development environment. In *Proceedings of the 2nd Euro-Par Conference*, volume 2, pages 665–674, 1996.

[17] E. T. Midorikawa, H. M. Oliveira, and J. M. Laine. PEMPIs: A new metodology for modeling and prediction of MPI programs performance. In *Proceedings of the SBAC-PAD 2004*, pages 254–261. IEEE Computer Society/Brasilian Computer Society, 2004.

[18] C. A. Moritz and M. I. Frank. LoGPC: Modeling network contention in message-passing programs. *IEEE Transactions on Parallel and Distributed Systems*, 12(4):404–415, 2001.

[19] J. Pjesivac-Grbovic, T. Angskun, G. Bosilca, G. E. Fagg, E. Gabriel, and J. J. Dongarra. Performance analysis of MPI collective operations. In *Proceedings of the Wokshop on Performance Modeling, Evaluation and Optimisation for Parallel and Distributed Systems (PMEO), in IPDPS 2005*, 2005.

[20] Y. Yang and J. Wang. Optimal all-to-all personalized exchange in multistage networks. In *Proceedings of the International Conference on Parallel and Distributed Systems (ICPADS'00)*, pages 229–236, 2000.