HAL
open science

Predictive models for bandwidth sharing in high performance clusters

Jerome Vienne, Maxime Martinasso, Jean-Marc Vincent, Jean-François Mehaut

To cite this version:

Jerome Vienne, Maxime Martinasso, Jean-Marc Vincent, Jean-François Mehaut. Predictive models for bandwidth sharing in high performance clusters. Proceedings of the IEEE Cluster Conference, 2008, Tsukuba, Japan. hal-00953618

HAL Id: hal-00953618

https://inria.hal.science/hal-00953618v1

Submitted on 25 Mar 2014

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.

Predictive models for bandwidth sharing in high performance clusters

Vienne Jérôme #*1, Martinasso Maxime #*1, Vincent Jean-Marc #1, Méhaut Jean-François #1

Laboratoire LIG equipe MESCAL, ZIRST 51 avenue Jean Kuntzmann
38330 Montbonnot Saint-Martin, France
* *BULL SA, 1 rue de Provence,*
BP 208 38432 Echirolles Cedex, France
1 firstname.lastname@imag.fr

***Abstract*—Using MPI as communication interface, one or several applications may introduce complex communication behaviors over the network cluster. This effect is increased when nodes of the cluster are multi-processors, and where communications can income or outgo from the same node with a common interval time. Our goal is to understand those behaviors to build a class of predictive models of bandwidth sharing, knowing, on the one hand the flow control mechanisms and, on the other hand, a set of experimental results. This paper present experiences that show how is shared the bandwidth on Gigabit Ethernet, Myrinet 2000 and Infiniband network before to introduce the models for Gigabit Ethernet and Myrinet 2000 networks.**

I. Introduction

Improving processor capabilities by increasing the number of cores produces sharing affects over computer components among core requests. Furthermore, understanding resource-sharing phenomenon becomes one most issues for actual parallel architectures.

These new behaviors of resource sharing are difficult to analyze and to predict. Simultaneously executions of application tasks create concurrent access over network. Their effects lead to performance leaks of bandwidth network segmentation among communications

In this article, we present an analysis of concurrent behaviors for Gigabit Ethernet, Myrinet 2000 and Infiniband using message passing library MPI. This analysis aim)s to the definition of predictive models based on the concept of bandwidth sharing.

The definition of predictive models has many objectives.

On the one hand, it leads to a better understanding of concurrency phenomenons, and, if we apply models on a scientific application, it allows us to identify periods of application performance loss. On the other hand, the definition for each network of one model allows us to easily compare the performances of an application.

Those 2 points can be important elements to help an HPC integrator to propose a network solution for a set of applications.

After a quick overview of network communication models, section III will describe the flow control mechanism of each interconnect. Section IV will be devoted to our approach, which will lead to our models in section V. Finally, we will present the evaluation of our models in section VI using synthetic benchmarks and the application Linpack[1] before to conclude.

II. Network Communication Models

Network communication can use MPI primitives to send or receive messages. MPI performance was investigated in [2] for high performance networks such as Myrinet or Infiniband. The authors showed the impact of buffer reuse, intra-node communication latency, and memory usage against the communication performance of several MPI primitives. If we focus on cluster of SMP nodes, more MPI performance studies were presented in [3], for different kind of platform.

A popular communication model developed is the LogP model [4] and is version LogGP model [5] for long messages. Both models use a linear model characterized by 4 parameters: L (as delay), o (as overhead), g (as bandwidth) and P (as number of processors) LogGP model introduce a new parameters G (as gap per byte). Impacts of each parameters was analyzed in [6], with methods to measure model parameters. Like these two models, and if we focus on wormhole model, a basic approach to predict communication delays is to a linear model featured by an overhead cost and a communication rate factor (applied to message length and path network). In network based on wormhole communications, this kind of models is sufficient in case of each communication is independent and does not share any network resource. Although, as one or several MPI applications, composed by several MPI tasks, can spread messages through overlapped communication time, these linear models poorly predict communication delays.

A first approach of communication sharing effects was introduced by [7]. The authors predict communication delay with a linear model taking into account the path sharing over a Myrinet network of workstation. Their study was based on the protocols GM[8] and BIP[9] but without MPI as user interface. Their model gives a first approach of sharing network resource. Communications are modeling by a piece-wise linear equation, and in case of sharing path, this equation is multiplied by the maximum number of communications within the sharing conflict. They evaluate their model against two communication schemes with synchronous sends. Their models gives good results for some communications of their schemes, but the authors do not provide more insight about communication

influences and so for communication delays predicted with low accuracy.

III. Flow control mechanism

Flow control is one of the most important factors for regulating access (and thus concurrency) to network resources. In fact, when a sharing conflict occurs over the network path, it is handled by the flow control of the physical network. Thus different policies of flow control lead to different behaviors of communications and thus to different communication performances.

A. Gigabit Ethernet

The flow control mechanism of Gigabit Ethernet was defined by the IEEE 802.3x committee for full-duplex Ethernet. By this standard, when a receiver becomes congested, it can send a pause frame to the source which therefore stops sending packets for a specific period of time. The receiver can also send a frame to inform the source to begin sending data again. However, using TCP over Gigabit Ethernet, the reliability of packets will be increased thanks to the concept of window size and the TCP's sliding windows mechanism. But such flow control has an important impact on the communication time.

B. Myrinet

Myrinet flow control is based on cut-through routing of packets. Such technology blocks packets while the communication channel is unavailable. It avoids the need for packet buffering.
Two chips of the Network Interface Card (NIC) perform the flow control. To achieve cut through routing, Myrinet NIC is based on a Stop & Go flow control protocol. In case of concurrency, receiver injects Stop or Go control messages into the network to inform senders to stop or resume the communication flow.

C. Infiniband

InfiniBand provides flow control mechanisms at different levels:

- a *static rate control* mechanism prevents a high speed link from overrunning a low speed link.
- a *credit-based flow-control mechanism* ensures the integrity of the connection. With InfiniBand, packets will not be transmitted until a sufficient memory space is available in the receiving buffer. The destination issues credits to signal available buffer space, after which the packets are transmitted. This mechanism is used for communication between interconnects and switches and between switches. A switch has a small buffer for each port or a global buffer.

IV. Approach

In this section, we will focus on our methodology to study the bandwidth sharing. To study this sharing, we had to defined the different kind of possible conflict that appears when communications start in the same time on the same node. Then, we had to develop a communication method to analyze the network behavior when these conflicts appear. The last part of this section will be focus on experimental conflict results obtain on a set of schemes.

A. Notion of Conflict

A conflict represents communications which share resources through simple pattern. As communications can income or outgo from clusters nodes, one communication can be seized by one of the following elementary conflicts:

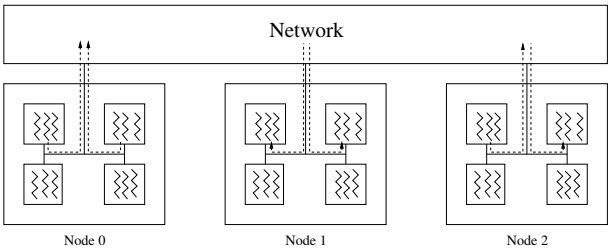


Fig. 1. Concurrent communication schemes

- Outgoing Conflict $C_{\leftarrow X \rightarrow}$ (node 0 in figure 1) where the communication only outgoes with other outgoing communications from a node X.
- Income Conflict $C_{\rightarrow X \leftarrow}$ (node 1 in figure 1) where the communication incomes with only other incoming communications to a node X.
- Income/Outgo Conflict $C_{\rightarrow X \rightarrow}$ or $C_{\leftarrow X \leftarrow}$ (node 2 in figure 1) in which a communication outgoes (resp. incomes) with other incoming (resp. outgoing) communications.

B. Communication Method and Measurements

Several way of implementing communication between MPI tasks can lead to different network performance. As consequence, it is interesting to introduce in this section our communication method of the different benchmarks used to analyze network behavior.
Sending is done through blocking send defined by the standard *MPI_Send* primitive. To synchronize MPI tasks between them, we used a MPI synchronization barrier, we highlight to the reader that using synchronization barrier gives implicitly an order for the tasks to continue their executions.
Cache effects can also influence measurements. To avoid cache effects, we executed several not-measured communication before each benchmark.
Measured time is done at the source task, starting before the MPI send and ending when the MPI send method terminates. Message length corresponds to the length specified in the *MPI_Send* primitive, and does not correspond to the effective length send over the network (MPI implementation add a small envelope to the message). Thus, effective message length are always greater than specified length and a 0-specified length is not meaningless. If one MPI task has to receive two messages from two others tasks, we used the MPI flag *MPI_ANY_SOURCE* in the receive function, to avoid a fixed

order of receive.
To study the bandwidth sharing behavior, we develop a software using this method. The parameters of the software are:

- Iteration number of MPI_SEND
- Referential time that is given by measuring the time of a MPI_Send of 20 MB from a node 0 to a node 1 without other communications.
- Description of the communication task scheme using a specific description language.

At the end, the software give us the penalty for each communication task. Be T_{ref} the referential time and T_i the time of the task i to send 20 MB then penalty P_i is:

$$P_i = \frac{T_i}{T_{ref}}$$

More the penalty is important more the time of the communication task will be important and more the performance of the cluster will be bad.

C. Experimentations

This section deals with congestion behavior comparison on three architectures: Gigabit Ethernet, Myrinet 2000, Infiniband (Infinihost III). For these experimentations, we used 3 different clusters:

- For Gigabit Ethernet: A cluster IBM eServer 326 composed of 53 nodes with 2 AMD Opteron 248 at 2GHz inside each node. Each node has 4 GB of memory and has a BCM 5704 Gigabit ethernet card. The MPI version is MPICH.
- For Myrinet 2000: A cluster IBM eServer 325 composed of 72 nodes with 2 AMD Opteron 246 at 2 GHz inside each node. Each node has 2 GB of memory and has a Myrinet 2000 card. The MPI version is MPI MX.
- For Infinihost III: A cluster BULL Novascale composed of 26 nodes with 2 Intel Woodcrest (4 cores/node) at 2,4 GHz inside each node. Each node has 4 GB of memory and has a Infinihost III card. The MPI version is MPIBULL2 (based on MVAPICH 1.0).

All clusters use a fat tree topology for the network. Each network has a different behavior when congestion happens, depending of the control flow. We will study those different behaviors on different communication schemes by looking penalties with our software.

Figure 2 can be describe like this:

- On the left, different communication schemes
- On the right, the penalties compute resulting of the scheme and of the network architecture for each communication task.

For example, for the second communication scheme, node 0 send in the same time a message of 20 MB to the node 1 (task a) and the node 2 (task b). For each communication time, the real time is 1.5 time more long on Gigabit ethernet compare to a single communication.
We choose these schemes to illustrate the evolution of the bandwidth sharing when we just add 1 communication task each time.

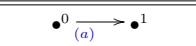
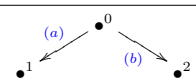
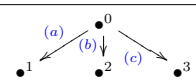
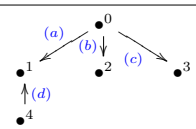
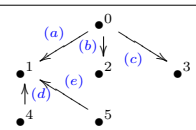
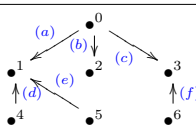
Communication schemes	Network		
	Gigabit Eth.	Myrinet	Infiniband InfinihostIII
	a = 1	a = 1	a =1
	a = 1.5 b = 1.5	a =1.9 b = 1.9	a = 1.725 b = 1.725
	a = 2.25 b = 2.25 c = 2.25	a = 2.8 b = 2.8 c = 2.8	a = 2.61 b = 2.61 c = 2.61
	a = 2.15 b = 2.15 c = 2.15 d = 1.15	a = 2.8 b = 2.8 c = 2.8 d = 1.45	a = 2.61 b = 2.61 c = 2.61 d = 1.14
	a = 4.4 b = 2.6 c = 2.6 d = 2.6 e = 2.6	a = 4.4 b = 4.2 c = 4.2 d = 2.5 e= 2.5	a = 3.663 b = 3.66 c = 3.66 d = 2.035 e = 2.035
	a = 4.4 b = 2.0 c = 3.3 d = 2.6 e = 2.6 f = 1.4	a = 4.5 b = 4.5 c = 4.5 d = 2.5 e= 2.5 f = 1.3	a = 3.935 b = 3.935 c = 3.935 d = 1.995 e = 1.995 f = 1.01

Fig. 2. Result of penalties depending of network

At first, we can see that 3 different flow controls protocols give different behaviors. Gigabit Ethernet with TCP appears to be the best architecture for concurrence management. Because when you add one communication in your scheme, whole communications won't be impacted.
But we shouldn't forget the initial bandwidth to really compare the results. Even if Gigabit ethernet seems to have a better "sharing behavior", Infiniband will probably stay the faster interconnect whatever the communication scheme.

V. Models

In this section we will present 2 finished congestion models for Gigabit Ethernet and Myrinet interconnect finalizing our previous work [10], [11].

A. Gigabit Ethernet

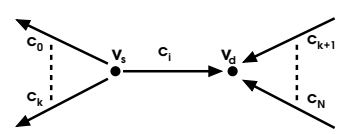


Fig. 3. Detail scheme of a communication.

Gigabit Ethernet congestion model is based on a quantitative approach (parameters + measures) due to distinctive feature from manufacturer. Looking at figure 3, the scheme represents a situation in a graph. This scheme pushes forward a

communication c_i coming from v_s to v_d. This communication is in conflict with the outgoing communications c_0 to c_k on the node v_s and incoming communications c_{k+1} to c_N. We will note $\Delta_o(v_s)$ the outgoing degree of the node v_s (that represents the number of outgoing communications) and $\Delta_i(v_d)$ the incoming degree of the node v_d (that represents the number of incoming communications).
As shown in our previous section, conflicts with an important number of communication cause important penalties. So, for a conflict, we define 2 set of communications: communication strongly penalized and the other ones.

Definition 1: Let c_i be a communication corresponding to an arc (v_s, v_d) of a graph G, having as degrees $\Delta_o(i) = \Delta_o(v_s)$ and $\Delta_i(i) = \Delta_i(v_d)$. Let C_o be the set of communications having the same source as c_i in the graph G. Let C_i be the set of communications having the same destination as c_i in the graph G.
Then if $\Delta_i(i) = max\{\Delta_i(j), \forall j \mid c_j \in C_o\}$, we say that c_i belongs to the set C_o^m of strongly slow outgoing communications. Reciprocally, we say that $c_i$belongs to the set of strongly slow incoming communications C_i^m if $\Delta_o(i) = max\{\Delta_o(j), \forall j \mid c_j \in C_i\}$.
We will note $card(C_i^m)$ (resp. $card(C_o^m)$), the number of communications belong to the set C_i^m (resp. C_o^m). To fix the penalty of a communication c_i, we compute two values. The first one represents penalties involved by the conflict in emission, noted p_o. The second one represents penalties involved by the conflict in reception, noted p_i.

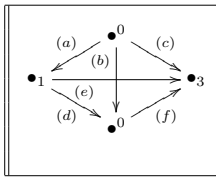
$$p_o = \begin{cases} 1 \text{ if } \Delta_o(i) = 1 \\ \begin{cases} \text{if } c_i \in C_o^m : \ \Delta_o(i) \times \beta \times (1 + \gamma_o(\Delta_o(i) - card(C_o^m))) \\ \text{then : } \ \Delta_o(i) \times \beta \times (1 - \gamma_o / card(C_o^m)) \end{cases} \end{cases}$$

$$p_i = \begin{cases} 1 \text{ if } \Delta_i(i) = 1 \\ \begin{cases} \text{if } c_i \in C_i^m : \ \Delta_i(i) \times \beta \times (1 + \gamma_i(\Delta_i(i) - card(C_i^m))) \\ \text{then : } \ \Delta_i(i) \times \beta \times (1 - \gamma_i / card(C_i^m)) \end{cases} \end{cases}$$

Last, the penalty associated to a communication c_i is:

$$p = max(p_o, p_i)$$

We used 3 parameters (β, γ_i , γ_o). They characterize the specificity of the card.
The parameter β describes the penalty get by the resource sharing. To estimate β, we use simple outgoing conflicts. We evaluate penalties of this conflict by increasing the number of outgoing communication. Then we divide the values that we get by the number of communication. For the Figure 2, we can see that β=0.75 ($\frac{1.5}{2} = \frac{2.25}{3} = 0.75$).



Coms	Measured T [s]	Predicted T [s]
a	0.095	0.095
b	0.099	0.095
c	0.118	0.113
d	0.068	0.069
e	0.099	0.103
f	0.103	0.103

Fig. 4. Verification of parameters, size of communication at 4MB.

To evaluate γ_i and γ_o, we use conflict from Figure 4 where each communication send the same amount of data. Following the scheme, the parameter γ_o leads strongly the communication a of node 0 and the parameter γ_i the communication f of node 3. Basing on communication times (t_a) and (t_f), we can deduce values of both γ. We obtain:

- $\gamma_o = 1 - t_a/(3 \times \beta \times t_{ref})$
- $\gamma_i = 1 - t_f/(3 \times \beta \times t_{ref})$

with t_{ref} the necessary time to send the same amount of data as (a) or (f) but without concurrency. On figure 4, we obtain $\gamma_i = 0.036$ and $\gamma_o = 0.115$.

B. Myrinet

Myrinet congestion model is based on a descriptive approach, a first quantitative approach was study in [11]. As Myrinet NIC used a Stop & Go flow control protocol, we will consider 2 states for a communication: $send$ and $wait$. Using those 2 states, we look up to determine all the possible combinations of communication states from the graph. For that, we use only one rule: *When a communication is in state "send", each communication having the same source node or the same destination node becomes in state "wait"*. Figure 5 shows an example of this rule. For the original graph introduces, we can find 5 different sets of communication states. On these schemes, a communication in state "send" is represented by a solid arrow, and a communication in state "wait" is represented by a dotted arrow.

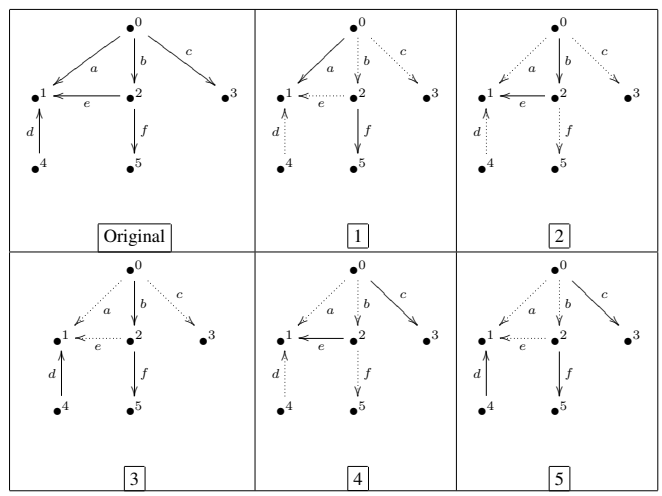


Fig. 5. Example of the various set of communication states

	Communications					
	a	b	c	d	e	f
Sum	1	2	2	2	2	3
Minimum	1	1	1	2	2	2
penalty	5	5	5	2.5	2.5	2.5

Fig. 6. penalty calculation for figure 5

When all sets of state are determined, we compute the emission coefficient defined as the number of states "send" for a communication. Then, we analyze the set of emission

coefficients of outgoing communications from a node, and we associate for each of the communication the minimal emission coefficient. In fact, we consider the worst case where each outgoing communication, of a same node, is as much slow as the slowest communication. It is because we consider that each communication is sharing the network card fairly.
Finally, the penalty of a communication is computed by dividing the number of state sets by the emission coefficient. An example is given at Figure 6 using the set of communication state of the Figure 5.

VI. Evaluation of predictive models

A. Simulator

In order to compute the prediction of each models, we build a simulator. This simulator has different kind of parameters:

- One or more application represented by a sequence of event. There are two kind of events: compute events and communication events. A compute event is composed only by the time of computation when a communication event is defined by the numbers of source and destination task and by the size of data to send (or receive).
- The definition of the cluster including for each node the number of core, the number of node etc... Node are numbered by iterative way starting by 0.
- Scheduling of tasks on nodes. It can be user defined or using Round-Robin scheduling
- The definition of the kind of model and the parameters: latency, bandwidth, size of buffer, etc...

With all these informations, the simulator will give us for each task, the duration of all events and total time, the kind of conflicts, the average penality, the size of communication etc...

B. Evaluation method

The evaluation of the models is based on the comparison between predicted times T_p and measured times T_m for various graphs. In the case of the synthetic graphs, this comparison is carried out by calculating the relative error (E_{rel}) for each communication.
Moreover, to measure the total error of the graph, we compute the average of the absolute errors (E_{abs}) of the communications of the graph. Both errors are measured as a percentage. For a graph constitutes by N communications and a communication called c_k, formulas calculating these errors are presented below:

$$E_{rel}(c_k) = \frac{T_p - T_m}{T_m} \times 100$$

$$E_{abs}(G) = \frac{1}{NR} \sum_{k=1}^{NR} |E_{rel}(c_k)|$$

The relative error gives a fine view of the accuracy of models.It allows, among others, to show if the model has an optimistic (negative error) or pessimistic (positive error) behavior. On the other hand, the average of the absolute errors gives a global view of the accuracies on the given graph.
The use of the absolute error avoids behaviors of compensation between relative errors, and therefore a more exact description of the predictions. Measurements are carried out by the program presented in VI-A.
In the case of graph coming from applications, we compute for each task the amount of times of the predicted and measured communications. For a task t_i, these sums are noted $S_m = \sum_{c_k \in t_i} T_m$ for the measured communications and $S_p = \sum_{c_k \in t_i} T_p$ for the predicted communications. Relying on these values, we compute the absolute error for a task t_i:

$$E_{abs}(t_i) = |\frac{S_p - S_m}{S_m} \times 100|$$

C. Synthetic Benchmark

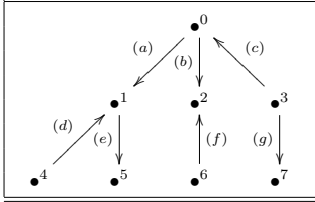
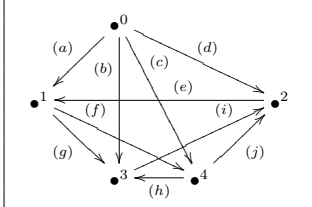
Schemes	Communication times			
MK1	com.	T_m	T_p	Error (E_{rel})
	a	0.087	0.089	2.3
	b	0.087	0.089	2.3
	c	0.070	0.071	1.4
	d	0.052	0.053	1.9
	e	0.037	0.035	-5.4
	f	0.051	0.053	3.9
	g	0.070	0.071	1.4
	Average of absolute errors E_{abs} = 2.6			
MK2	com.	T_m	T_p	Error (E_{rel})
	a	0.164	0.177	7.9
	b	0.164	0.177	7.9
	c	0.164	0.177	7.9
	d	0.164	0.177	7.9
	e	0.043	0.053	23.2
	f	0.086	0.085	-1.2
	g	0.087	0.085	-2.3
	h	0.108	0.101	-6.5
	i	0.108	0.101	-6.5
	j	0.059	0.073	23.7
	Average of absolute errors E_{abs} = 9.5			

Fig. 7. Accuracy of Myrinet 2000 model with synthetic graphs

At the beginning, to evaluate our models and the simulator, we use synthetic graph. Synthetics graphs allow us to explore the relevance of models. It is interesting to determine the evolution of errors to discover for each conflict the weakness and the strong of each models. The Figure 7 show us an example with a tree (MK1) and a complete graph (MK2) results on our Myrinet model. Concerning Myrinet and Gigabit Ethernet on trees, our model is often pessimistic (but with enough efficacy) specially when a conflict outgo and another one income in the same time for Myrinet. It should be happen due to the use of the full duplex link.
The only big difference come from complete graph, we can see that both have a good evaluation but the gigabit ethernet model seems to be optimistic compare to Myrinet model that seems to be pessimistic.

D. Linpack

To finish the evaluation of our models we use Linpack with a communication scheme where each task n send message to the task $n + 1$ for a problem size of 20500. To get the events, we modified MultiProcessing Environment (MPE) library integrate in Mpich. The impact of tracing is very low because the average cost of the tracing was of 0,7%.

The Figures 8 and 9 show the result for our Myrinet and Gigabit Ethernet Models. The diagrams represent the measured and predicted times and the line the absolute error for each MPI task.

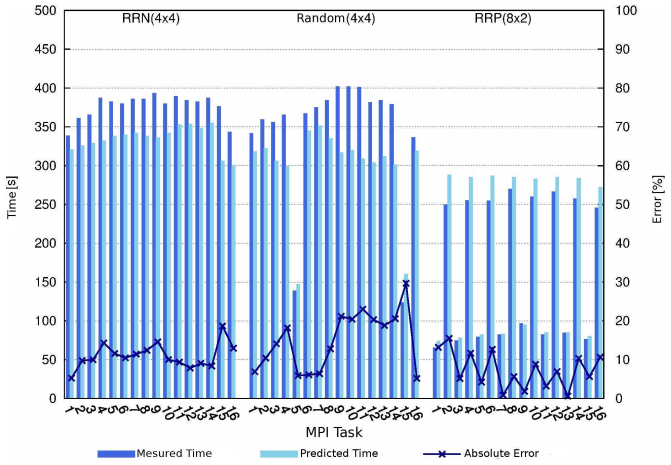


Fig. 8. Evaluation of Gigabit Ethernet model using HPL, size 20500

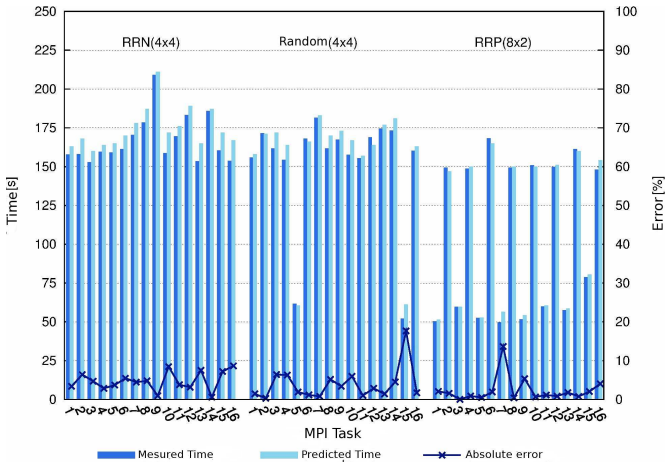


Fig. 9. Evaluation of Myrinet model using HPL, size 20500

We tested models using 3 possible scheduling:

- RRN (Round-Robin per Node): MPI tasks are assigned cyclicly between each nodes.
- RRP (Round-Robin per Processor): MPI tasks are assigned filling first the nodes.
- Random: MPI tasks are assigned randomly.

We can see that Myrinet model is globally accurate, and the errors should be caused by memory congestion. Gigabit Ethernet are a bit less accurate than Myrinet. It could be explain by the variability of Gigabit network behaviors. Compare to Myrinet, Gigabit Ethernet doesn't use technique as RDMA or OS-bypass. These techniques limit the influence of the system in the management of communications.

This method highlights two problems: the influence of the resource memory and propagation of the error. These two problems are responsible for the loss of accuracy obtained. However, the predictions under these conditions remain satisfactory.

VII. Conclusion

In this paper, we compared and studied the penalties of Myrinet 2000 and Gigabit Ethernet with Infiniband. With the study of these penalties and the knowledge of flux control protocols, we established models of these interconnects. Those 2 models was integrated inside a simulator to study the accuracy of them. A set of synthetic benchmark has been tested to prove and accurate our models, before to test them on real application.

These models allow us to predict performance loss due to bandwidth sharing and compare the performance on the same application on Myrinet and Gigabit Ethernet networks.

We are actually working to improve again these models with a deep study of some specific cases and to test our models on nodes with 8 and 16 cores to extend them. We are working too on the model of the Infiniband InfinihostIII and ConnectX interconnect on BULL Novascale clusters.

References

[1] J. Dongarra, P. Luszczek, and A. Petitet, "The linpack benchmark: past, present and future." *Concurrency and Computation: Practice and Experience*, vol. 15, no. 9, pp. 803–820, 2003.

[2] J. Liu, B. Chandrasekaran, J. Jiang, S. Kini, W. Yu, D. Buntinas, P. Wyckoff, and D. Panda, "Performance comparison of mpi implementations over infiniband myrinet and quadrics," 2003. [Online]. Available: citeseer.ist.psu.edu/liu03performance.html

[3] K. Al-Tawil and C. A. Moritz, "Performance modeling and evaluation of MPI," *Journal of Parallel and Distributed Computing*, vol. 61, no. 2, pp. 202–223, 2001. [Online]. Available: citeseer.ist.psu.edu/453861.html

[4] D. E. Culler, R. M. Karp, D. Patterson, A. Sahay, E. E. Santos, K. E. Schauser, R. Subramonian, and T. von Eicken, "Logp: a practical model of parallel computation," *Commun. ACM*, vol. 39, no. 11, pp. 78–85, 1996.

[5] A. Alexandrov, M. F. Ionescu, K. E. Schauser, and C. Scheiman, "LogGP: Incorporating long messages into the LogP model for parallel computation," *Journal of Parallel and Distributed Computing*, vol. 44, no. 1, pp. 71–79, 1997. [Online]. Available: citeseer.ist.psu.edu/alexandrov95loggp.html

[6] R. P. Martin, A. Vahdat, D. E. Culler, and T. E. Anderson, "Effects of communication latency, overhead, and bandwidth in a cluster architecture," in *ISCA*, 1997, pp. 85–97. [Online]. Available: citeseer.ist.psu.edu/martin97effects.html

[7] S. C. Kim and S. Lee, "Measurement and prediction of communication delays in myrinet networks," *J. Parallel Distrib. Comput.*, vol. 61, no. 11, pp. 1692–1704, 2001.

[8] Myricom Inc., "GM: A Message-Passing System For Myrinet Networks," 2003, http://www.myri.com/scs/GM-2/doc/html/.

[9] L. Prylli and B. Tourancheau, "Bip: A new protocol designed for high performance networking on myrinet," in *IPPS/SPDP Workshops*, 1998, pp. 472–485.

[10] M. Martinasso and J.-F. Méhaut, "Model of conccurent mpi communications over smp clusters," HAL-INRIA, Tech. Rep. 00071352, May 2006.

[11] M. Martinasso and J.-F. Méhaut, "Prediction of communication latency over complex network behaviors on SMP clusters," in *Proceedings of the 2nd European Performance Engineering Workshop, EPEW 2005*, ser. Lecture Notes in Computer Science, vol. 3670. Versailles: Springer-Verlag, Sep. 2005, pp. 172–186.