Scalable Algorithms for Complete Exchange on Multi-Cluster Networks

Alfredo Goldman
Department of Computer Science, University of São Paulo
Rua do Matão, 1010 CEP 05508-900 - São Paulo — Brazil
e-mail: gold@ime.usp.br

Abstract

Inside a dedicated parallel computer the communication times were generally modeled in the same way, independently of which processors communicate. In a network where the links among the computers are heterogeneous, or in hierarchical clusters, this might not be true anymore. Computers that have faster links, or are closer to each other should be able to exchange messages faster. These differences on communication times should be considered, not only for attributing tasks to the processors but also in global synchronization/communication. The goal of this paper is to study irregular all-to-all communications in a network of dedicated clusters.

1 Introduction

The first works which addressed the communication problem were either too specific, or not realistic. Some works assumed more general models, with messages of different sizes and totally connected networks. In [2], Coffman et al. studied the problem of exchanging messages of different sizes, but they did not allow preemption on messages sending, once a communication starts it cannot be interrupted. In [1], Choi and Hakimi studied the same problem, allowing transmissions of messages on several, not necessarily consecutive, sends on a synchronous networks. Finally, in [3] the authors studied a more realistic model with the exchange of messages of different sizes in an asynchronous network where preemption was allowed.

In these works [1, 2, 3], the communication times were given by the linear cost model. Which is well adapted for most of interprocessor communication when there is no congestion. In this model, to send a message, there is a startup (β) and the transmission time depends inversely on the bandwidth (γ). The parameters β and γ were the same for all processors pairs. In this work we intend to introduce another parameter in the model, the main idea being that communication does not have to behave similarly for all processor pairs.

2 Problem and Model

In order to model a multi-cluster network (MCN), we suppose that there are two connection levels, local and distant. The local communications will model the communications inside a same cluster, and the distant communications the exchange of data between different clusters.

We assume that each computer in the system has a limited capacity of communication, that is, each machine can transmit at most one message and receive at most one message simultaneously. This limitation is well known in the literature as one-port full-duplex. We also assume synchronous sends, that is a send is only complete after the message reception.

There are two different approaches. In [6] the authors considered the difference between the end of the send and the end of the receive operations for a communication, we do not consider this difference on our model. It is interesting to remind that on [6] there were no explicit consideration on congestion. Conversely, in [5] the congestion problem for the complete exchange is solved at the packet level. Intuitively, the assumed model can be viewed as an intermediate model.

3 Homogeneous environment

The goal of this section is to give an overview of the known algorithms to schedule communications on homogeneous environments. A detailed description of these algorithms can be found in [3].

The main algorithms are: FP - Fixed Pattern Algorithm, where each processor executes $n-1$ communication phases. In each communication phase t, processor p_i sends its message to processor $p_{i+t \bmod n}$.

HL - Hypercube-like Algorithm, this algorithm was first proposed for hypercubes. This algorithm does message forwarding and does not split the messages.

There are also three matrix algorithms. The main idea of the matrix algorithms is to provide the knowledge of the whole communication matrix on each processor, and then to apply an algorithm which provides better communication phases. The algorithms can be written as:

```
all-to-all exchange of the message sizes
global strategy determination
strategy execution
```

There are three main strategies: Max-Min which computes communication phases where the smallest message size is maximized, Max-Sum where the sum of the messages size is maximized on each communication phase and Unif. The main idea of the Unif is to obtain communication phases where all the messages have the same size [3].

4 Scheduling of Communications on MCN

Obviously, the algorithms of the previous section can be used, without any change for a MCN. In this section we propose two strategies for scheduling MCN communications: to adapt the previous algorithms to the new environment and a modification on the algorithm given in [6].

Adapting the algorithms. As the fixed pattern and hypercube like algorithms do not use the information on the message sizes, they only can use information on which links are faster than others. So it should be convenient to group on the communication phases only local, or distant communications.

On the other side, the other algorithms can be adapted. To do that, we have to introduce additional costs for the messages to be transmitted between clusters. To implement this cost, we have to consider the differences on the communication parameters. Instead of using the communication matrix M we will use the expected time matrix, where for each message its communication time is computed [4].

The communication phase is found using the time matrix, but only the information on the communication pattern (the pairs of processors involved on transmission-reception) is used. The algorithm is given below:

```
build from M the expected time matrix T
while the comm. matrix M is not done
  find a communication phase C_T on T
  use the comm. pattern C_T to construct C_M
  send the messages on C_M
  subtract from M the messages on C_M
```

The substitution of the communication matrix by a time matrix can be applied directly to MM and MS, but in order to use Unif we have to search a more complicated strategy. To be able to use the same idea we should know the number of times that each message will be split before the beginning of the algorithm. We propose a different way to solve this problem. The main idea is to reconstruct the time matrix after each communication phase [4]. To avoid empty message transmission when the remaining messages are small, we consider that they cannot be split anymore.

Two level algorithms. The main idea is to solve the problem level by level, first the messages are exchanged on each cluster, and messages to each other cluster are centralized on one node. Then the nodes that received the messages for other clusters exchange the messages on the MCN. Finally these nodes broadcast the messages to the final destinations [6]. We propose that any of the algorithms of Section 3 can be used in the first two steps. The best choice depends on the parameters for local and distant communication. A two-level algorithm does message forwarding, it does message splitting depending on the choice of the algorithm.

5 Conclusion

From the simulations we have done, we conclude that the algorithms proposed for the exchange of messages on MCNs provide performances up to 20% better than the algorithms for homogeneous networks. For MCNs the algorithms choice depends mostly on the distant communication parameters, for small start-ups adapted Unif has the best performances, for big start-ups HL is still the best algorithm. Otherwise, the adapted MS provides the best results.

For future works it would be interesting to generalize the MCN model to consider more than two kinds of communications, each one with its own start-up and bandwidth, and to generalize the algorithms to the n-port model.

References

[1] H. Choi and S.L. Hakimi. Data transfer in networks. *Algorithmica*, 3:223–245, 1988.

[2] E.G. Coffman, M.R. Garey, D.S. Johnson, and A.S. Lapaugh. Scheduling file transfers. *SIAM J. Comput.*, 14(3):744–780, August 1985.

[3] A. Goldman, J. Peters, and D. Trystram. Exchange of messages of different sizes. In *IRREGULAR'98*, LNCS 1457, Springer-Verlag, 194–205, 1998.

[4] A. Goldman. Algorithms for Complete Exchange on Multi-Cluster Networks. Manuscript (www.ime.usp.br/~gold/papiers/cluster02.ps).

[5] A. Tam and C-L. Wang. Contention-Free Complete Exchange Algorithm on Clusters. Cluster 2000, 2000, Saxony, Germany, 57-64.

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