Proceedings of the World Congress on Engineering 2007 Vol I
WCE 2007, July 2 - 4, 2007, London, U.K.

Dynamic Graphs For The Dynamic Reconfiguration Of Groups Of Computers

C.B.Navarrete, E.Anguiano *

Abstract—**In the last years, the computers have increased their capacity of calculus and networks, for the interconnection of these machines, have improved until obtaining the actuals high rates of data transfering. The programs that nowadays try to take advantage of these new technologies cannot be written using the traditional techniques of programming, since most of the algorithms were designed for being executed in an only processor, in a nonconcurrent form instead of being executed concurrently in a set of processors working and communicating through a network. This paper aims to present the ongoing development of a new system for the reconfiguration of groupings of computers, taking into account these new technologies.**

Keywords: heterogeneous cluster computing, clusters reconfiguration, resources optimization

1 Introduction

Due to the continuous growing of the capabilities of networks, there are posed diverse optimization problems associated with algorithms for the design of networks. For a general optimization of a groups of machines, some methodology of dynamic allocation of resources is needed [1, 2, 3] being this, one of the main reasons of why nowadays exists a special interest in the search of new algorithms, able to replace the traditional methods whose efficiency and possibility of scaling in parallel to the architectures of processors make them inapplicable, in many cases, to very complex problems [4].

The main problem designing a parallel or distributed algorithm reside in the communication and synchronization of processes for its concurrent execution in different processors.

From the point of view of the heterogeneity of processors, a good parallel application for Heterogeneus Network of Computers (HNoC) must distribute computations unevenly taking into account at least the speeds of the processors. The efficiency of the parallel application also depends on the accuracy of estimation of the speeds of the processors of the HNoCs, which is difficult because the processors may have different speeds for different applications due to differences in the set of instructions, the number of instruction execution units, the number of registers, the structure of memory hierarchy and so on. [5]

From the point of view of the communications, one of the main problems of the implementation of networks of communication is the one of designing a network topology that could verify certain characteristics of trustworthiness, assuming this as the measurement that evaluates the probability of success in the communication between pairs of nodes. This is an influential factor in the quality of services offered to all the nodes. The evaluation of the exact parameters that determine the trustworthiness of a communication network is a NP-hard problem [6, 7]. For this reason, an optimization of the topology of the net is needed. It muss be assured that the temporary delay due to the communication and synchronization of the processes, is minor to the delay of processing the data by each processors. Also it is important to consider that when the problem is too much divided, the time necessary to communicate datas between the nodes and to synchronize them, exceeds the time of present computation of the CPU.

The common communication network is normally heterogeneous. The speed and bandwidth of the network, between different pairs of processors, may differ significantly. This makes the problem of optimal distribution of computations and communications across the HNoC much more difficult than across a dedicated cluster of workstations interconnected with a homogeneous high-performance network. Other issue is that the common communication network can use multiple network protocols for communication between different pairs of processors. A good parallel application should be able to use multiple network protocols between different pairs of processors [8].

Until this moment, we have considered only a static net-

*Email: carmen.navarrete@uam.es, eloy.anguiano@uam.es Higher Polytechnical School of Engineering. Universidad Autnoma de Madrid, SPAIN

work topology, function of the domain decomposition, defined before the execution of the algorithm or application in the HNoC; in those problems in which is not trivial to define a domain decomposition (i.e access to data in distributed databases) would be usefull to have a dynamic topology of communications, which varied around the distribution of the data and the different latencies from network, depending on the different connections between the nodes.

2 System Description

The model proposed for the reconfiguration of nodes of a HNoC will be based on a sufficiently ample language of communications. Using this language any node will be able to know in real time, what information contains any other node del system (data middleware), without having to communicate first with the master node of the system. This language also will allow the nodes to modify their rolls of master/slave depending on what value or data structure is needed and on what node has asked for it.

In order to be able to reconfigure the HNoC the language of communications will include dataframe of control as dataframe of data. The dataframes of data will contain the data needed for the execution of the algorithm in the HNoC whereas the control dataframes will be those with information about the runtimes and the comands and will be sent to know what nodes have what information and what hierarchy exists between the different nodes from HNoC.

Therefore, the HNoC will be represented as a directed dynamic graph in which the different connections will have weights of edges (cost of moving directly from one vertex to another one) equal to the different delays from network and those ones due to the overload of the processors of the nodes.

The vertex of the graph will represent each processors available at the HNoC. The information of each node of the graph will contain the effective load of the processor, considering this like the availability to execute other processes, as well as statistic value proportional to the execution time of the processes in the previous steps of the algorithm.

In the other hand, the edges of the graph will represent the connections between the available processors, according to a certain instant of the algorithm. These connections are statistically weighted according to the network delays, the time of transmission of the data throught the net that will depend as well on the network protocol and the physical layer used for this communication.

In addition to the nodes that are included in the HNoC, a super master node will exist as it will be the one which will administer the graphs that represents the different nodes and their communications. The super master node will maintain a resulting graph of all the existing communications between any two nodes of the HNoC and the execution times of each processor. With this information, and according to a statistical function, this node will generate an optimal graph, probably different, for each node of the HNoC. These graphs will be calculated based on the communications that each node needs to make with any other one and they will be calculated according to some algorithm of minimum path for graphs. Knowing that all the weights of the vertices and of the edges of the graph are positive values, we could use algorithms based on Dijkstra [9] or Bellman-Ford [10].

References

[1] R. Canal, J.M.l Parcerisa, and A. Gonzalez. Dynamic cluster assignment mechanisms. In *HPCA*, pages 133–, 2000.

[2] R. Bhargava and L. John. Improving dynamic cluster assignment for clustered trace cache processors, 2003.

[3] K. Amiri, D. Petrou, G. Ganger, and G. Gibson. Dynamic function placement in active storage clusters, 1999.

[4] A. Lastovetsky. Scientific programming for heterogeneus systems - bridging the gap between algorithms and applications. In *PARELEC'06 IEEE Proceedings*, pages 3–8, 2006.

[5] A. Lastovetsky and R Reddy. HeteroMPI: Towards a message-passing library for heterogeneous networks of computers. *Journal of Parallel and Distributed Computing*, 2005.

[6] M.O. Ball. Computing network reliability. 1979.

[7] J.S. Provan and M.O. Ball. The complexity of counting cuts and of computing the probability that a graph is connected. *SIAM Journal on Computing*, (12):777 – 788, 1983.

[8] J.Dongarra and A. Lastovetsky. *An overview of heterogeneous high performance and grid computing.* American Scientific Publishers, 2006.

[9] Edsger. W. Dijkstra. A note on two problems in connexion with graphs. *Numerische Mathematik*, 1:269–271, 1959.

[10] Richard Bellman. On a routing problem. *Quarterly of Applied Mathematics*, 16(1):87 – 90, 1958.