On the Design of Communication-Aware Task Scheduling Strategies for Heterogeneous Systems *

J. M. Orduña, V. Arnau, A. Ruiz, R. Valero
Departamento de Informática
Universidad de Valencia
E-mail: Juan.Orduna@uv.es

José Duato
D.I.S.C.A.
Universidad Politécnica de Valencia
E-mail: jduato@gap.upv.es

Abstract

Many research activities have focused on the problem of task scheduling in heterogeneous systems from the computational point of view. However, an ideal scheduling strategy would also take into account the communication requirements of the applications and the communication bandwidth that the network can offer. In this paper, we first propose a criterion to measure the suitability of each allocation of network resources to each parallel application, according to the communication requirements. Second, we propose a scheduling technique based exclusively on this criterion that provides a near-optimal mapping of processes to processors according to the communication requirements. Evaluation results show that the use of this scheduling technique fully exploits the available network bandwidth, greatly improving network performance. Therefore, the proposed scheduling technique may be used in the design of communication-aware scheduling strategies for those situations where the communication requirements are the system performance bottleneck.

1. Introduction

Networks of Workstation (NOWs) have become powerful and flexible systems that are nowadays being used as low-cost parallel computers [3, 4, 11]. The incremental expansion capabilities provided by NOWs usually makes these systems to become heterogeneous as they grow. Effectively, using switch-based interconnects [5, 21, 13, 15] different types of workstations and personal computers (possibly with different computing power) can be interconnected.

In order to fully exploit the computing power of heterogeneous systems, a lot of research has focused on solving the *NP-complete* problem of efficiently scheduling diverse groups of tasks to the machines that form the system [19, 17, 12, 18, 23, 24]. Nevertheless, these proposals only focus on computational aspects, and they do not consider communication cost, thus assuming that the communication subsystem provides enough bandwidth in any case. However, as the computational power of new processors increases, the interconnection network in these heterogeneous systems may become the system bottleneck, particularly when executing applications with huge network bandwidth requirements, like multimedia applications, video-on-demand applications, etc. Therefore, the mapping problem must be studied not only from the computational point of view but also from the communication point of view. Given a heterogeneous system (that may be formed by different groups of interconnected homogeneous systems) and given a certain set of different (parallel or sequential) applications from different users, an ideal scheduling strategy would map the processes to processors taking into account both the computational and the communication requirements of the applications running on the machine. The scheduler would choose either a computation-aware or a communication-aware task scheduling strategy depending on the kind of requirements that leads to the system performance bottleneck.

In order to develop a communication-aware task scheduling strategy for parallel applications on heterogenous systems, several problems must be solved. First, the communication requirements of the applications running on the machine must be measured or estimated. On the other hand, the available network resources must also be characterized. Additionally, some criterion is needed to measure the suitability of each allocation of network resources to each of the parallel applications, according to their communication requirements. Based on this criterion, some scheduling technique based on the computational requirements as well as on the communications requirements must be developed. Finally, this technique must be integrated with process scheduling, in order to be used only when the communication requirements are the ones that lead to the system performance bottleneck.

In a previous paper [2], we proposed a model of communication cost for characterizing the network resources of any given irregular topology. In this paper, we first pro-

*Supported by the Spanish CICYT under Grant TIC97–0897–C04–01

pose a criterion to measure the suitability of each allocation of network resources to each parallel application, according to the communication requirements. Second, we propose a scheduling technique based on this criterion that provides a near-optimal mapping of processes to processors according to the communication requirements. In this first approach we have considered the problem under simplified assumptions (all the applications generate only intracluster traffic, one process per processor, all the processes have the same communication requirements) in order to quickly analyze the behavior of the proposed scheduling technique. The purpose is to make a first evaluation of the network performance improvement that can be achieved when this technique is used. We have only measured throughput improvement to asses how much we can improve the utilization of communication services for a given network hardware.

Due to the huge complexity of solving both the problem of measuring the communication requirements of the applications running on the machine and the problem of integrating the proposed scheduling technique with process scheduling, we will leave the solving of these problems for future work.

The rest of the paper is organized as follows: Section 2 establishes the problem of characterizing irregular networks, proposes a model of communication cost and briefly discusses some existing heuristic search methods used for task scheduling. Section 3 briefly describes the proposed model of communications cost. Section 4 describes the proposed criterion to measure the suitability of a given allocation of network resources to the applications running on the machine, and also the proposed task scheduling technique based on this criterion. Section 5 shows the performance evaluation results obtained with the proposed method. Finally, Section 6 presents some concluding remarks.

2. Background

In massively parallel computers (MPP's), interconnection networks have been traditionally characterized by their topological properties, such as number of nodes, bisection width and diameter. However, these properties do not provide information about the arrangement of the links, one of the most important issues when characterizing irregular topologies. Therefore, these properties cannot be used to measure the communication cost for irregular topologies. Additionally, the routing algorithm may also seriously affect the performance of irregular topologies by determining the traffic distribution in the network. Consider, for example, the up*/down* routing scheme used in Autonet networks [21]. In this scheme some minimal paths are forbidden for routing, and the routing algorithm tends to overload links located near the root switch. As a result, the network may saturate prematurely. Therefore, both the topology and the routing algorithm must be considered when characterizing irregular networks.

In a previous paper, we proposed a new model of communication cost between nodes, *the table of equivalent distances* [2]. This model takes into account only the topology of the network and the routing algorithm, and is totally independent of the traffic pattern. The strong correlation of the model with network performance makes it a valid basis for network characterization methods that do not depend on the traffic pattern. Additionally, this model may be used as the basis for an efficient mapping of processes to processors, since it provides a metric based on internode distance.

We have defined a criterion based on the table of distances to measure the suitability of a given allocation of network resources to the applications running on the machine. This criterion is the resultant intracluster and intercluster network bandwidth relationship for a given mapping. Additionally, we have studied the use of this criterion together with the application of heuristic search methods in order to find the best scheduling technique based on the communication requirements. A wide variety of heuristics, such as Opportunistic Load Balancing (OLB) [1, 12], User-Directed Assignment (UDA) [1, 12], Fast Greedy [1], Min-min or Max-min [1, 12, 16] have been applied to task scheduling. In particular, we have studied the application of three of them: Genetic Simulated Annealing [7, 22], A^* heuristic [17, 20], and Tabu search [14]. The Genetic Simulated Annealing heuristic is a combination of the Genetic Algorithm search method [23, 25] and the Simulated Annealing method [20]. Genetic Algorithms works with "chromosomes" (a chromosome would be in our case a possible mapping of processes to processors) and a target function to evaluate each chromosome. This method iteratively produces mutations in the chromosomes and selects only a portion of the best evaluated mutations. On other hand, Simulated Annealing is an iterative technique that considers only one possible solution (mapping) at a time. However, this method uses a procedure that probabilistically allows poorer solutions to be accepted in order to perform a more exhaustive search. The A^* heuristic is a tree search method that prunes the tree according to a cost function, until a leaf (mapping) is reached. Finally, the Tabu search is a heuristic search method that keep tracks of areas of the solution space already explored.

In this paper we present the application of the Tabu search method. With this heuristic technique we obtained the best results in the search of a good mapping. The proposed algorithm provides a near-optimal mapping of processes to processors for any given set of processes running on the machine and for any given network topology. Evaluation results show that this method significantly improves network performance by assigning the processes with higher communication requirements to the network areas with higher bandwidth. Furthermore, this scheduling technique is applicable to both regular and irregular topologies, providing a general basis for communication-aware task scheduling strategies.

3. Model of Communication Cost

Our model of communication cost between nodes proposes a simple metric, the *equivalent distance* between each pair of nodes (in what follows we will refer to a switching element as a node). This metric measures the cost of communicating two nodes without explicitly considering traffic pattern [2]. The method used to compute the equivalent distance between two nodes n_i and n_j is based on an analogy with electric circuits. First, only the links belonging to shortest paths between n_i and n_j are considered. The remaining links are removed from the network. Note that we only consider paths supplied by the routing algorithm used. then, assuming that all the links in the network have the same cost, each link is replaced with a unit resistor. Finally, the equivalent distance between n_i and n_j is computed as the equivalent resistance between them.

The application of this method produces a *table* T_N *of* $N \times N$ *equivalent distances*, where N is the number of nodes in the network. In this table, the element T_{ij} represents the equivalent distance between node i and node j. The table of distances does not satisfy the triangular inequality, and thus it does not define a metric space. In other words, we cannot find a metric space in which we can represent the nodes with the equivalent distances between them. Therefore, we cannot use classical clustering methods based on Euclidean metric distances. However, the table of distances provides a measurement of internode distance, and it is highly correlated with network performance, as shown in [2].

4. A Communication-Aware Scheduling Technique

Based on the table of distances, we first propose a criterion for measuring the quality of each allocation of network resources to the applications running on the machine. Second, we propose a heuristic search method to establish the optimal mapping of processes to processors for any given set of applications running on the machine and for any given network topology. From a general point of view, we can consider that each application belongs to a different user. Therefore, we can assume that the processes belonging to the same application may intensively communicate between them, but they will not communicate at all with processes from other applications. Therefore, we can group the processes running on the machine, forming a set of logical clusters of processes, where each logical cluster is formed by the processes belonging to each application. Additionally, in order to quickly analyze the behavior of the proposed scheduling technique we will also assume that there exists only one process per processor and that all the processes have the same communication requirements. With these simplifying assumptions, the proposed algorithm intends to provide a network partition adapted to any existing set of logical clusters.

4.1. Quality Function

When all of the traffic generated by the set of logical clusters is intracluster traffic (we are assuming that each application belongs to a different user) then the processes should be assigned to the processors in such a way that they fully exploit the communication links existing in the network. Therefore, it is necessary to define a metric of the communication bandwidth achieved by each one of the possible mappings of processes to processors, in order to select the best mapping.

Starting from the table of distances, we have defined two distinct and complementary global quality functions, the *similarity* and the *dissimilarity* functions. The first function measures the intracluster distances, and the second one measures the intercluster distances. Since these distances can be considered as the inverses of the intra and intercluster bandwidth, respectively, these functions provide a relationship between the intracluster and intercluster bandwidth. It must be noticed that both functions must take into account the mapping of processes to processors. Since we are assuming that all of the traffic generated by the set of logical clusters is intracluster traffic and that there exists only one process for each processor, the assignment of processes to processors will determine the destinations for the messages generated by each network node, and also the communication cost for a given topology. In this sense, we will denote as a *cluster* the subset of nodes (network switches) assigned to each set of logical clusters (for the sake of simplicity, we have also assumed that on one hand all the network nodes are equal and have the same number of processors attached to them, and on the other hand all the logical clusters are formed by a number of processes that results in an integer multiple of network nodes when they are mapped to the processors). A given mapping of processes to processors will determine a network partition formed by the union of all the clusters.

Let a network partition P be formed by M clusters $A_1, A_2, \cdots, A_M$, and let a cluster A_i be formed by x_i nodes $a_1, a_2, \cdots, a_{x_i}$ ($x_i < N$, where N is the number of nodes in the network). In general terms, nodes $a_1, a_2, \cdots, a_{x_i}$ form a cluster A_i only if the processes of a given logical cluster are assigned to this set of network nodes. That is, a node a_j will belong to a cluster A_i if any of the processes assigned to a_j belongs to a certain logical cluster α and the rest of the nodes that form A_i also execute processes belonging to α. Under these conditions, the cluster quality function for cluster A_i is defined as

$$F_{A_i} = \sum_{k=1}^{x_i-1} \sum_{j=k+1}^{x_i} T^2_{a_k a_j} \tag{1}$$

where T_{ij} is the distance from node i to node j in the table of distances. If a cluster A_i contains x_i nodes, then

F_{A_i} is defined as the quadratic sum of all intracluster distances. It must be noticed that for these functions we are considering the distances in the table of distances.

The similarity global function for the final partition is defined as

$$F_G = \frac{\dfrac{\sum_{i=1}^{M} F_{A_i}}{\sum_{i=1}^{M} \dfrac{x_i\,(x_i-1)}{2}}}{\dfrac{\sum_{i=1}^{N-1}\sum_{j=i+1}^{N} T_{ij}^2}{\dfrac{N\,(N-1)}{2}}} \tag{2}$$

where M is the number of clusters in the final partition, F_{A_i} represents the cluster similarity function for each cluster A_i and the term

$$\sum_{i=1}^{M} \frac{x_i\,(x_i-1)}{2} \tag{3}$$

is the total number of intracluster distances in partition P. F_G is computed as the sum of all the F_{A_i} values divided by the total number of intracluster distances existing in partition P and normalized by the quadratic average value of all of the distances between the network nodes. Thus, a value of F_G greater than 1 means that the final mapping shows a greater intracluster communication cost than when mapping processes to processors randomly (without computing any scheduling technique), while values for F_G close to 0 mean that the obtained mapping shows a very small intracluster communication cost, compared to the quadratic average distance of network nodes.

For the dissimilarity global function we define the cluster dissimilarity function D_{A_i} for a cluster A_i as

$$D_{A_i} = \sum_{k=1}^{x_i}\sum_{j=1}^{N-x_i} T_{a_k j}^2 \quad \forall j \notin A_i \tag{4}$$

That is, D_{A_i} is defined as the quadratic sum of all intercluster distances from nodes in cluster A_i to all the nodes in the rest of the clusters. The dissimilarity global function is defined as

$$D_G = \frac{\dfrac{\sum_{i=1}^{M} D_{A_i}}{\sum_{i=1}^{M} x_i\,(N-x_i)}}{\dfrac{\sum_{i=1}^{N-1}\sum_{j=i+1}^{N} T_{ij}^2}{\dfrac{N\,(N-1)}{2}}} \tag{5}$$

where M represents the number of clusters in the final partition and D_{A_i} represents the cluster dissimilarity function for each cluster A_i. D_G is computed as the sum of all the D_{A_i} values divided by the total number of existing intercluster distances in partition P and normalized by the quadratic average value of all of the distances between the network nodes. Thus, a value of D_G close to 1 means that the obtained mapping has an intercluster communication cost very close to the communication cost of considering each network node as a cluster. Higher values for D_G mean that the obtained mapping shows greater intercluster distances (the clusters are better defined than when considering each network node as a cluster).

F_G and D_G provide a measurement of the intracluster and intercluster communication costs, respectively. Therefore, they can be considered as inversely related to the intracluster and intercluster bandwidth, respectively. Thus, the quotient of D_G divided by F_G provides the relationship between the intracluster and intercluster bandwidth. We will denote this relationship as the *clustering coefficient* C_c. The scheduling technique should maximize this coefficient in order to provide the best mapping of processes to processors, since the processes in the set of logical clusters are assumed to have a 100% of intracluster communication and no intercluster communication at all.

4.2. Scheduling Technique

We start from a given set of logical clusters that will define a space of solutions Ω, consisting of all the possible mappings of processes to processors for the existing network topology. This space is associated with a target function F that assigns a cost to each particular solution (mapping of processes to processors) $P \in \Omega$. Since the mapping of processes to processors has been shown to be a $NP-Complete$ problem [10, 16], we have applied a heuristic search method to find the best mapping for the existing set of logical clusters. The heuristic search method must find a particular solution P_0 such that

$$F(P_0) \leq F(P) \;\; \forall\, P \in \Omega$$

We have used as a target function F for each mapping P the global similarity function F_G, as defined in Section 4.1.

Since this function represents the intracluster communication cost, when minimizing this target function we are also maximizing the clustering coefficient C_c, that is, the intracluster/intercluster bandwidth relationship. In this way the heuristic search leads to a mapping that will provide almost the best network performance.

We have tried several of the heuristic search methods proposed in [6], and we have obtained the best results for a variant of the Tabu Search method [14]. This heuristic provided the same or better clustering coefficients than other methods with higher computational cost. Given a certain mapping of processes to processors P_i (that will determine a network partition with M clusters $A_1, A_2, \cdots, A_M$), our proposal searches another mapping P_{i+1} consisting of P_i with the permutation between two nodes belonging to different network clusters that results in the greatest decrease in the function F. Therefore, $F(P_{i+1}) \leq F(P_i)$. However, when the target function enters a local minimum, then there will not exist any permutation that decreases the value of F. In order to continue the search, the Tabu method establishes that in this case the next iteration P_{i+1} will consist of P_i with the permutation that results in the smallest increase of the target function F. Additionally, the inverse permutation to the one that led from P_i to P_{i+1} is forbidden for a given number h of iterations (the name of the method is derived from these "Tabu movements"). The search must end when F reaches its minimum value. However, there is no way to ensure that the actual minimum is the minimum value for F.

In particular, we have started the Tabu search with a random mapping. We have computed the Tabu search either until the same local minimum has been reached three times or until the method has searched for 20 iterations. At this point, the value $F(P_{MIN})$ considered as minimum and its corresponding mapping P_{MIN} are stored, and another seed (random mapping) $P_{i'}$ is used, thus continuing the search from another different starting point. After repeating this process 10 times, for small size networks (up to 16 switches) the minimum obtained by this method was the same value $F(P_0)$ that the one obtained with an exhaustive search. For larger size networks we could not perform an exhaustive search, due to the huge computational power it required. However, we were not able to find any lower value for F with any other heuristic search method.

Figure 1 shows the value of $F(P_i)$ for the Tabu search performed in a 16-switch network. In this figure, the total iteration number is shown in the X-axis. The value of F at the ten different starting points of the search method form the peak values of F in the figure. It can be seen that the value for F rapidly decreases in the first few iterations after a starting point. It is worth mentioning that the minimum value for F is not reached from all the starting points. In this example it is only reached from the third, fifth, and sixth starting points. On other hand, from the seventh starting point, 20 iterations are searched without reaching 3 times the same local minimum.

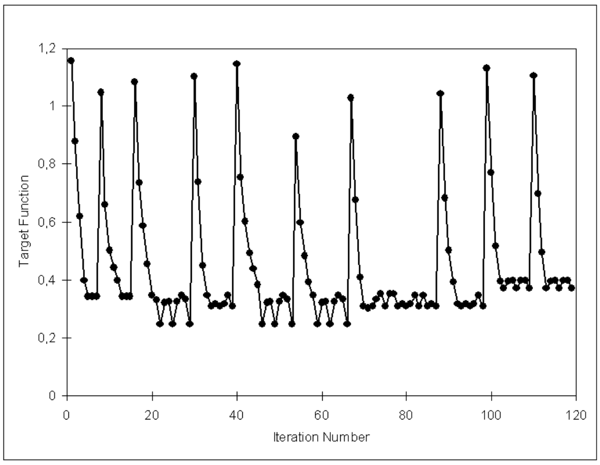

Figure 1. Tabu search in a 16-switch network

5. Performance Evaluation

In this section we are going to study the improvement in network performance that the proposed scheduling technique can provide, as well as the correlation between the clustering coefficient and network performance. In order to achieve these goals, we have evaluated the performance of several network topologies with random mappings of processes to processors and also with the mapping provided by the scheduling technique. We have computed the clustering coefficient for each one of the considered mappings, in order to show that this quality function really provides an "a priori" measurement of the network performance that a given mapping can achieve. This performance study assumes that all the communication between processors is intracluster communication and all the processors transmit the same amount of information.

We have evaluated the performance of several irregular networks by simulation. The evaluation methodology used is based on the one proposed in [8]. The most important performance measures are latency and throughput. The message latency lasts since the message is injected in the network until the last flit is received at the destination node. Throughput is the maximum amount of information delivered per time unit (maximum traffic accepted by the network). Traffic is the flit reception rate. Latency is measured in clock cycles. Traffic is measured in flits per switch per cycle. Our simulator models the network at the flit level.

5.1. Network Model and Message Generation

The network is composed of a set of switches. The network topology is irregular and has been generated randomly. However, for the sake of simplicity we imposed three restrictions. First, we assumed that there are exactly 4 workstations connected to each switch. Second, two neighboring switches are connected by a single link. Finally, all the switches in the network have the same size. We assumed 8-port switches. Therefore, each switch has 4 ports available to connect to other switches. From these 4 ports, three of them are used in each switch when the topology is generated. The remaining port is left open. We have evaluated networks with a size ranging from 16 switches (64 workstations) to 24 switches (96 workstations). For some network sizes, several distinct topologies have been analyzed.

In order to show that the proposed scheduling technique can increase network performance, we have evaluated each network with several distinct mappings of processes to processors. For the sake of simplicity, we have assumed a fixed pool of N processes (where N is also the number of workstations in the network) grouped into 4 clusters with $X = \frac{N}{4}$ processes, where X is also multiple of four (since we are assuming one process per processor). With this assumption we ensure that each cluster of processes can be mapped on an integer number of network switches. Each process is assumed to send all of the generated messages to processes in the same logical cluster of processes. For each network, we have performed the Tabu search until it has provided the best network partition for the 4 clusters of $\frac{N}{4}$ processes, computing the corresponding mapping. Additionally, we have computed several random mappings for each considered network.

5.2. Simulation and Correlation Results

(5,6,8,15) (0,1,11,12) (3,9,10,14) (2,4,7,13)

Figure 2. 4-cluster partition obtained for a 16-switch network

Figure 2 shows the obtained 4-cluster partition for a 16-node (switch) network. Since for this network size the set of logical clusters is formed by 4 clusters of 16 processes each, the ideal network partition would be formed by clusters of 4 nodes (switches) each. The scheduling technique provides a network partition formed by exactly four clusters of four nodes each. Additionally, we computed 9 different randomly generated mappings.

Figure 3 shows network performance for the mapping provided by the scheduling technique (OP label), compared with the network performance obtained by several randomly generated mappings (R_i labels) (Although we run network simulations for all of the generated mappings, we have not included all of them in this figure for the sake of clearness). For each one of the considered mappings, the network was simulated from low traffic (simulation point $S1$) to saturation (simulation point $S9$). The clustering coefficient obtained for each mapping is shown on the right side of each plot label. As can be seen, the network through-

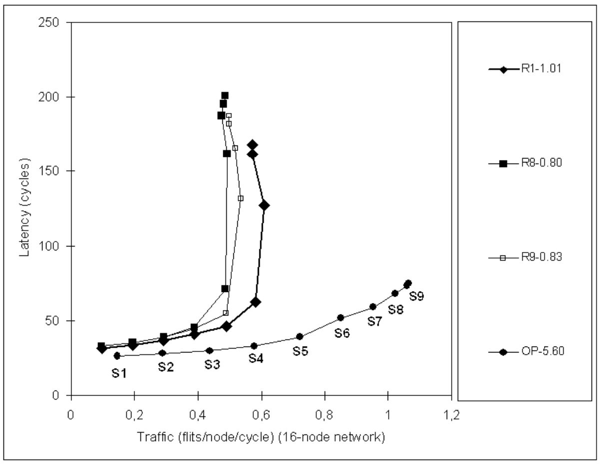

Figure 3. Simulation results for a 16-switch network

put obtained with the mapping provided by the scheduling technique is about a 85% higher than the network throughput obtained with any of the randomly generated mappings. These results show that the proposed scheduling technique can improve network performance when the logical set of clusters is well defined. On the other hand, the value of the clustering coefficient C_c is clearly lower for randomly generated mappings, showing that this function is directly related to network performance.

Figure 4 shows the partition provided by the scheduling technique for a 24-switch network. This network has been especially designed with four interconnected rings of 6 nodes, in order to test if for well defined topologies the scheduling technique was able to find the correct network clusters. In this case, the scheduling technique was able to identify the mentioned topology.

(1,2,3,4,5,6) (7,8,9,10,11,12) (18,19,20,21,22,23) (0,13,14,15,16,17)

Figure 4. 4-cluster partition obtained for a especially designed 24-node network

Figure 5 shows the network performance for the mappings performed on the 24-switch network. The scheduling technique provided one mapping (labeled as OP), and we have also generated three randomly generated mappings (R_i labels). This figure clearly shows that the network performance obtained with the mapping provided by the scheduling technique is much higher than the performance obtained with randomly generated mappings. In this case the network throughput obtained with the mapping provided by the scheduling technique is five times higher than the network throughput obtained with any of the randomly gen-

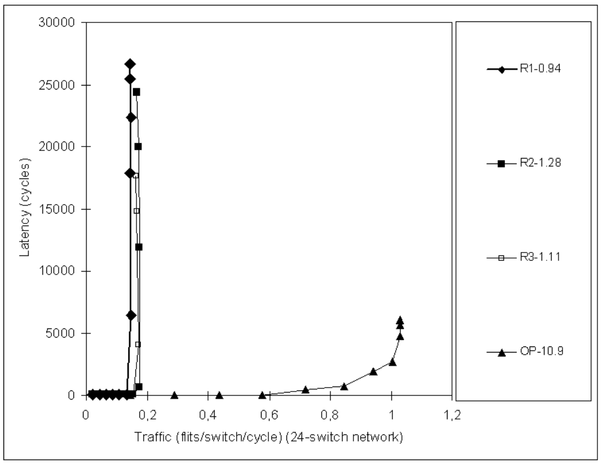

Figure 5. Simulation results for the specially designed 24-switch network.

erated mappings. It is worth mentioning that the clustering coefficient obtained for the mapping provided by the scheduling technique is higher for this network topology than the one for the 16-switch network topology. This higher value shows that this 24-switch network contains very well defined clusters.

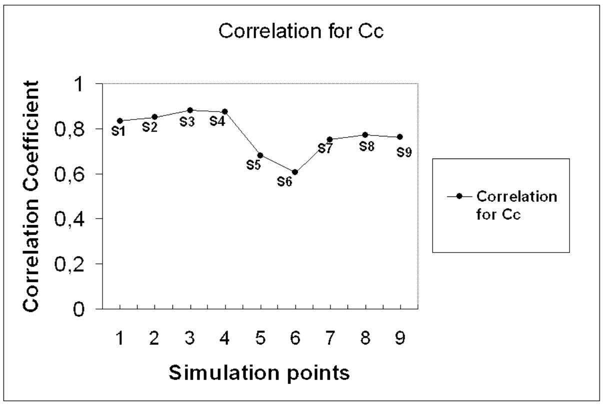

Figure 6. Correlation of clustering coefficient C_c with network performance.

Additionally, we have studied the correlation of the clustering coefficient C_c with network performance. Since the logical clusters are perfectly defined (all the generated traffic is intracluster traffic), it is expected that a higher value of C_c corresponds to better network performance (effectively, Figures 3 and 5 show that the network performance is directly related to the value of C_c). Figure 6 shows the correlation between coefficient C_c and the network performance obtained for all the mappings for the 16-node network (see Figure 3), from low load (point S1) to deep saturation (point S9) . The correlation coefficient is about 85% for the simulation points corresponding to low load (points S1 to S4). For simulation points corresponding to network saturation (simulation points S7 to S9) the correlation coefficient is about 75 %, showing that this coefficient is highly correlated with network performance even under deep saturation. The correlation index obtained for the simulation points S5 and S6 is not significant, due to the very different network saturation levels for different mappings corresponding to these simulation points. For example, for the traffic level at simulation point S6, the network is under deep saturation with the random mappings, while it is still not saturated with the mappings provided by the scheduling technique.

Although they are not shown here due to space limitations, we have also studied this correlation index for other network examples. The correlation index for any of the considered networks was higher than 70% for simulation points at both low network load and network saturation. These results validate the clustering coefficient as an "a priori" measure of relative network performance.

6. Conclusions and Future Work

In this paper, we have proposed a scheduling technique consisting of the application of a heuristic search method using a search criterion based on the table of distances [2]. This scheduling technique intends to be a first step towards scheduling strategies for heterogeneous systems that take into account both the computational and the communication requirements of the applications running on the machine. The purpose is to evaluate the network performance improvement that a communication-aware scheduling strategy can achieve.

We have evaluated the network performance for the mapping of processes to processors provided by the proposed scheduling technique, comparing it with the network performance obtained with randomly generated mappings of processes to processors. Network throughput is greatly improved when using the mapping provided by the proposed scheduling technique. When the network topology has better defined clusters of processors, then the improvement in network performance achieved with the proposed scheduling technique is also higher. Thus, the proposed scheduling technique may be used as a valid task scheduling strategy when the network is the system bottleneck in a heterogeneous system.

We have also studied the correlation between the proposed search criterion and network performance, showing that when the set of logical clusters is formed by jobs with only intracluster communication then this criterion is highly correlated with network performance. In this case the criterion really measures the effectiveness of communications, and therefore it can be used as an "a priori" measure of relative network performance.

As for future work, we plan to study the aspects of the design of a scheduling strategy that have not been studied in this paper: the measurement of the communication requirements of the applications running on the cluster, and the integration of the proposed scheduling technique with process scheduling. Also, we plan to study the use of the proposed scheduling technique in a more realistic environment, eliminating the simplifying assumptions made in this paper.

References

[1] R. Armstrong, D. Hensgen and T. Kidd, "The Relative Performance of various Mapping Algorithms Is Independent of Sizable Variances in Run-Time Predictions", in *Proceedings of* 7^{th} *IEEE Heterogeneous Computing Workshop (HCW'98)*, March 1998, pp. 79-87.

[2] V. Arnau, J.M. Orduña, A. Ruiz, J. Duato, "On the Characterization of Interconnection Networks with Irregular Topology: A New Model of Communication Cost", in *XI IASTED International Conference on Parallel and Distributed Computing and Systems (PDCS'99)*, November 1999.

[3] F. Berman, R. Wolski, S. Figueira, J. Schopf and G. Ghao, "Application-Level Scheduling on Distributed Heterogenous Networks", in *Proceedings of Supercomputing*, 1996.

[4] F. Berman, R. Wolski, "Scheduling from the Perspective of the Application", in *Proceedings of Symposium on High Performance Distributed Computing*, 1996.

[5] N. J. Boden, D. Cohen, R. E. Felderman, A. E. Kulawik, C. L. Seitz, J. Seizovic and W. Su, "Myrinet - A Gigabit per Second Local Area Network," *IEEE Micro*, pp. 29–36, February 1995.

[6] T. D. Braun, H. J. Siegel et al., "A Comparison Study of Static Mapping Heuristics for a class of Meta-tasks on Heterogeneous Computing Systems", in *Proceedings of* 8^{th} *IEEE Heterogeneous Computing Workshop (HCW'99)*, April 1999, pp. 15-29.

[7] H. Chen, N. S. Flann and D. W. Watson, "Parallel Genetic Simulated Annealing: A massively prallel SIMD approach", in *IEEE Transactions on Parallel and Distributed Computing*, Vol. 9, No. 2, Feb. 1998, pp. 126-136.

[8] J. Duato, "A new theory of deadlock-free adaptive routing in wormhole networks," *IEEE Trans. Parallel and Distributed Systems*, vol. 4, no. 12, pp. 1320–1331, December 1993.

[9] J. Duato, S. Yalamanchili, L. Ni, *Interconnection Networks: An Engineering Approach*, *IEEE Computer Society Press, 1997*.

[10] D. Fernández-Baca, "Allocating modules to processors in a distributed system", in *IEEE Transactions on Software Engineering*, Vol. SE-15, No. 11, Nov 1989, pp. 1427-1436.

[11] I. Foster and C. Kesselman, *The Grid: Blueprint for a New Computing Infrastructure*, *Morgan and Kauffman Publishers, 1998*.

[12] R.F. Freund, M. Gherrity, S. Ambrosious et al., "Scheduling in Multi-User, Heterogeneous Computing Environments with SmartNet", in *Proceedings of* 7^{th} *IEEE Heterogeneous Computing Workshop (HCW'98)*, March 1998, pp. 184-199.

[13] M. Galles, "Spider: A high speed network interconnect", in *IEEE Micro*, vol. 17, no. 1, pp. 34–39, January-February 1997.

[14] F. Glover, M. Laguna, , *The Grid: Tabu Search*, *Kluwer Academic Publishers, 1998*. ISBN: 0-7923-8187-4

[15] R. Horst, "TNet: A Reliable System Area Network," *IEEE Micro*, vol. 15, no. 1, pp. 37–45, February 1995.

[16] O. H. Ibarra and C. E. Kim, "Heuristics Algorithms for Scheduling Independent Tasks on Nonidentical processors", in *Journal of the ACM*, Vol. 24, No. 2, April 1977, pp. 280-289.

[17] M. Kafil and I. Ahmad, "Optimal Task Assignment in Heterogeneous Distributed Computing Systems", in *IEEE Concurrency*, Vol. 6, No. 3, 1998, pp. 42-51.

[18] M. Maheswaran, S. Ali, H. Siegel, D. Hensgen and R. Freund, "Dynamic Matching and Scheduling of a Class of Independent Tasks onto Heterogenous Computing Systems", in *Proceedings of* 8^{th} *IEEE Heterogeneous Computing Workshop (HCW'99)*, April 1999, pp. 30-44.

[19] S. C. S. Porto and C. C. Ribeiro, "A Tabu Search Approach to Task Scheduling on Heterogeneous Processors under Precedence Constraints", in *Int. Journal of High Speed Computing*, Vol. 7, No. 1, 1995, pp. 45-71.

[20] S. J. Russell and P. Norvig, *Artificial Intelligence: A Modern Approach*, *Prentice-Hall, Englewood Cliffs, 1995*.

[21] M. D. Schroeder et al., "Autonet: A high-speed, self-configuring local area network using point-to-point links," Technical Report SRC research report 59, DEC, April 1990.

[22] P Shroff, D. Watson, N. Falnn and R. Freund "Genetic Simulated Annealing for Scheduling Data-Dependent Tasks in Heterogeneous Environments", in *Proceedings of* 5^{th} *IEEE Heterogeneous Computing Workshop (HCW'96)*, April 1996, pp. 98-104.

[23] H. Singh and A. Youssef, "Matching and Scheduling Heterogenous Task Graphs using Genetic Algorithms", in *Proceedings of* 5^{th} *IEEE Heterogeneous Computing Workshop (HCW'96)*, April 1996.

[24] H. Topcuoglu, S. Hariri and M.Y. Wu, "Task Scheduling Algorithms for Heterogenous Processors", in *Proceedings of* 8^{th} *IEEE Heterogeneous Computing Workshop (HCW'99)*, April 1999, pp. 3-14.

[25] S. C. S. L. Wang, H. J. Siegel, V. P. Roychowdhury and A. A. Maciejewski, "Task Matching and Scheduling in heterogeneous Computing Environmets using a Genetic-algorithm-based Approach ", in *Int. Journal of High Speed Computing*, Vol. 7, No. 1, 1995, pp. 45-71.