Modeling Speedup of SPMD Applications on the Intel Paragon: A Case Study *

E. Smirni[1] and E. Rosti[2]

[1] Department of Computer Science, Vanderibilt Univeristy
Nashville TN 37235, USA
esmirni@vuse.vanderbilt.edu
[2] Dipartimento di Scienze dell'Informazione
Università di Milano, Italy
rose@dsi.unimi.it

Abstract. Interconnection networks with nearly distance independent communication latency are a key feature of recent architectures. However, shared resources such as network channels can become bottlenecks that degrade performance and limit workload scalability.
In this paper, the problem of network contention in medium scale multicomputers is addressed. A queueing network model that predicts application speedup on the Paragon is presented. The model has been experimentally validated under the SUNMOS operating system. The experimental and modeling results are in good agreement and suggest ways to avoid internal network contention of communication intensive applications on the Paragon.

1 Introduction

A significant number of recent message-passing multiprocessors including the Intel Paragon use wormhole routing. Wormhole routing is a technique that guarantees high performance interconnection network and allows for multiprocessor scalability [DS86]. One of its distinctive features is that communication latency is nearly independent of the distance between the communicating processes. However, contention can degrade communication latency in networks with deterministic routing since message routing does not adapt to changes in the network load [AV94].

In this paper, we propose a queueing network model that succesfully predicts workload scalability of single program multiple data (SPMD) applications on the Intel Paragon. When a workload is scheduled for execution on a multiprocessor, it is desirable to rescrict any variance in execution time and consequently any variance in scalability. Factors that affect performance and variance in workload scalability include the number of processors assigned to each program (i.e.,

* This work was partially supported by sub-contract 19X-SL131V from the Oak Ridge National Laboratory managed by Martin Marietta Energy Systems, Inc. for the U.S. Department of Energy under contract no. DE-AC05-84OR21400 and by Italian M.U.R.S.T. 40% and 60% projects.

the partition size) and the partition geometry, the mapping of the communicating processes on the assigned partition, the available bandwidth, the system communication overhead, and the granularity of computation relative to communication. A communication intensive SPMD application is used to investigate the impact of such factors on program execution time. Even for a small number of assigned processors, internal channel contention can govern the execution time of the parallel workload and restrict its scalability. Experimental validation of the modeling results has been performed on a 512 node Paragon under the Sandia-University of New Mexico Operating System (SUNMOS) [MMR93]. SUNMOS has been selected because it offers better communication performance over the native OSF/1.

This paper is organized as follows. Section 2 presents the system hardware, the selected SPMD application, and the model used for performance prediction. Modeling results are presented in Section 3. The model experimental validation is presented in Section 4. Conclusions and future work are presented in Section 5.

2 The System

2.1 The Hardware Platform

The Paragon is an MIMD machine with distributed memory. The system is organized in a 2-dimensional, not necessarily square, grid topology. Bidirectional links connect adjacent nodes. Each node is composed of the iMRC mesh router chip and two i860 XP processors each with a 16 KB cache and a 32 MB main memory. One of the two processors, the application processor, is dedicated to computational work, while the second one, the communication processor, provides communication services. The iMRC mesh router chip is responsible for routing messages from one processor to another. The iMRC may route up to 5 message packets at once with no contention, offering a bandwidth of 200 MB/sec per port, for a total of 1 GB/sec.

The Paragon interconnection network is wormhole routed. With wormhole routing, each message is divided into small pieces of a few bytes called flits (flow control digits). The header flit (or flits) reserves the channels as it moves from source to destination. The remaining flits follow in a pipelined fashion. If the next required channel is busy, then the header flit is blocked until the channel becomes available. If the header flit blocks, the remaining flits are blocked at the intermediate nodes along the path. Only the tail flit can release the occupied channels. Deadlock is avoided at the routing level [NM93]. The wormhole routing on the Paragon is deterministic [DS86], meaning that the route established by the header flit is fixed and independent of the network load. Flits always travel first along the X dimension and then along the Y dimension.

2.2 The Workload

To investigate the impact of network contention on application speedup a communication intensive workload is selected. Matrix transposition [CE90] is a classic example of a communication intensive SPMD program. The matrix transposition algorithm is implemented as follows. Given that the number of processors assigned to a workload is p, the matrix is partitioned into p equal subblocks $M_{i,j}$, for $0 \leq i, j \leq \sqrt{p} - 1$, each stored on a distinct processor. For simplicity it is assumed that $p = m^2$, $m \geq 1$. During the computation phase, processes that own diagonal sublocks transpose their subblocks $M_{i,i}$ without communicating with other processes. Processes that own off-diagonal subblocks first perform a communication phase that interchanges their subblock $M_{i,j}$ with the subblock $M_{j,i}$ of their respective process. A computation phase follows that transposes the received subblock. Since all subblocks are of equal size, all processes perform an identical amount of computation. Processes that own non-diagonal subblocks transfer identical amounts of data.

The results presented in the following sections do not consider any network contention that is generated by message traffic of other applications. To avoid interaction with any external load (e.g., messages associated with other parallel programs running on the machine), and to focus on internal contention, only contiguous sets of processors are considered as possible partitions.

2.3 The Model

To model application speedup, a non-product form queueing network model is constructed [BCMP75]. The model is solved by simulation. Node stations model the application processes. Transposition of a diagonal subblock is modeled by a single visit to a diagonal node station. Customers' behavior at off-diagonal node stations mimics off-diagonal processes behavior. Upon arrival at an off-diagonal node station, a customer starts acquiring links for message transmission. Link acquisition and contention are explicitly modeled by enforcing exclusive access to the corresponding link resources. Link resources are released after communication takes place. When a subblock exchange is completed, a service demand that models subblock transposition is placed on the respective node station. Routing probabilities reflect the deterministic XY routing. The experimentally measured network bandwidth values are used as model parameters. Since the measured computation time and bandwidth have negligible variance, node service times and communication latency are modeled with constant distributions. Experimental validation of the model is presented in Section 4.

3 Simulation Results

3.1 Data Allocation

With SPMD workloads, the data placement problem can be considered as an alternative way to view the process mapping problem. Data placement may determine channel contention for a certain set of links because of the communication pattern induced among the application processes. To investigate up to what extent the relative placement of the communicating processes affects their execution time, three process mapping policies have been considered. The mapping policies and their respective communication patterns are illustrated in Figure 1.

The mesh mapping onto the $n \times x$ SUNMOS partition for $n = x = \sqrt{p}$ is illustrated in Figure 1(a). Each matrix subblock is placed onto its corresponding node in the partition (i.e., matrix subblock M_{ij}'s indices and node (i, j)'s indices coincide). With this mapping, network contention is generated. As an illustration, node (0,2)'s message carrying subblock M_{02} competes with nodes (0,1) and (0,3)'s messages for the same physical links.

In Figure 1(c) the worst process mapping is depicted on a one dimensional mesh. The communication pattern obtained maximizes the innermost link utilization, thus making it the bottleneck under particular conditions.

The optimal mapping, regardless of the physical partition geometry, allocates matrix subblocks that must be exchanged onto adjacent nodes so that no link contention occurs. Two possible optimal mappings on two and one dimensional dimensional meshes are depicted in Figures 1(b) and 1(d).

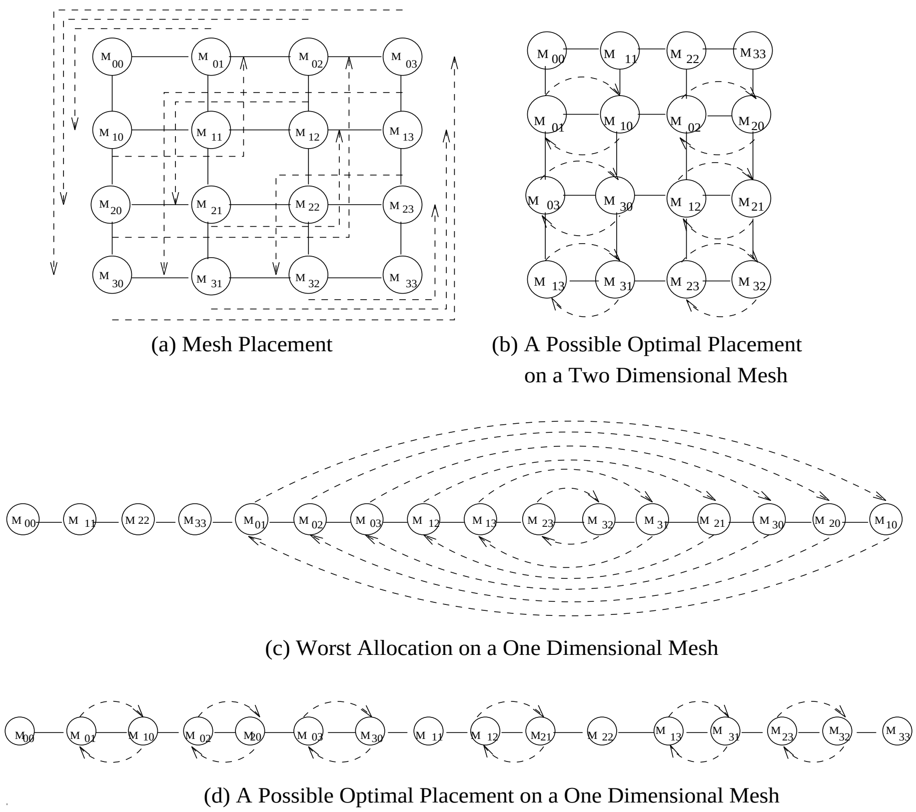

Fig. 1. Four data allocations for matrix transposition on 16 processors.

The speedup curves for a matrix of size 600×600 under the various mapping schemes are reported in Figure 2(a). As expected, the best speedup is obtained with the optimal mapping. With mesh mapping, and p assigned processors, the processes at the right upper and left bottom are the slowest. They have to compete for $\sqrt{p} - 2$ links along the most contended paths in the partition. The worst mapping yields the largest communication time and consequently the worst speedup. The processes that are placed at the two extremes of the worst mapping contend for $\frac{p-\sqrt{p}}{2} - 1$ links. For 4 allocated processors, the application performance is insensitive to the specific placement used as none of them causes any contention. As p increases, the gap among the three curves widens. The available bandwidth halves for each traversed link [SCRD94]. The transmission of smaller subblocks, that corresponds to messages with shorter link holding time, only partially mitigates the channel contention. At the most contended link in a 5×5 mesh (e.g., the link between nodes (0,0) and (0,1), or between (4,3) and (4,4)) the process on node (0,4) can only use 1/8 of the available bandwidth (i.e., $1/(2^{\sqrt{p}-2})$). Under the worst mapping, the processes that are furthest away from the most contended link enjoy only 1/512 of the available bandwidth (i.e., $1/(2^{\frac{p-\sqrt{p}}{2}-1})$).

The speedup curves for the various mappings for a matrix of size 1200×1200 are reported in Figure 2(b). Similarly to the 600×600 case, as the number of assigned processors increases, the difference between the worst and optimal placements becomes more pronounced. However, the gap in the speedup curves of the 1200×1200 case is not as wide as in the 600×600 case. Although the link holding time increases with the matrix size, the gap among the three mapping schemes decreases because of the higher increase in the ratio of computation to communication time of the workload. The computation part dominates and the workload execution time becomes less sensitive to network contention.

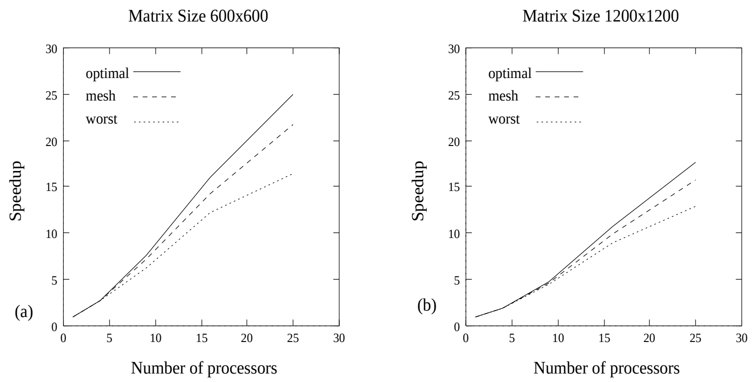

Fig. 2. Predicted speedup curves for three mapping schemes for a matrix of size 600×600 and a matrix of size 1200×1200.

3.2 Computation to Communication Granularity

In this section the effect of interleaving computation and communication is investigated. Transposition is performed in a step-wise fashion. Processes exchange a set of submatrix columns and transpose them before sending the next set of columns.

Smaller messages, combined with smaller amounts of computation between two message exchanges, are the key performance factors for this experiment. Interleaving computation and communication allows for the partial overlap of the application computation and communication phases.

In Figure 3(a) the speedup curves for a 600×600 matrix for the three data allocation strategies when the transposition is performed in two steps (dashed lines) are compared with the base case (solid lines). A speedup improvement by 7% is achieved for the optimal placement and it is attributed to the partial overlap of computation and communication. Higher improvements are achieved for the two other placements (namely, 16% for the mesh case and 30% for the worst case) because the computation that follows each message exchange decreases the injection rate of messages in the network. If the transposition is performed in four steps, (see Figure 3(b)), then the performance gain w.r.t. the base case is even larger and the application execution time becomes almost insensitive to the data placement. For the worst mapping case, the latency improvement is 54%.

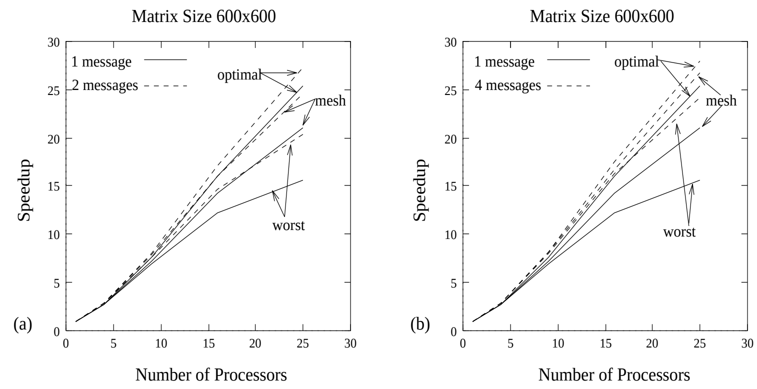

Fig. 3. Predicted speedup curves for three mapping schemes for a matrix of size 600×600 for various numbers of execution steps and different computation to communication granularities.

In general, it is noticed that the mapping schemes do not affect much the variance in workload scalability. As the number of exchanged messages increases, the gap in the speedups of the three mapping schemes decreases.

4 Experimental Validation

Experimental validation is performed on a 512 node Paragon, organized as a 16×32 mesh. Figure 4 compares predicted versus actual speedups of the base case algorithm for matrix sizes 600×600 and 1200×1200. The simulation model effectively captures the experimentally measured execution times of the matrix transposition workload. The model tends to overestimate speedup as contention increases but speedup behavior is qualitatively captured.

As both Figures 4(a) and 4(b) indicate, the model predicts the execution time of the application with the optimal and the mesh data placements more accurately than with the worst data placement. The same observation can be made with respect to the matrix size. The model prediction is tighter with larger matrices as a result of the high computation to communication ratio.

For the 1200×1200 matrix, the experimental and modeling curves overlap completely for the optimal and the mesh placements. For the 600×600 matrix, the maximum error for these two strategies is 6%. For the worst placement of the small matrix size, the error of the model lies between 8% and 12%. The use of constant distributions for the communication times of the link stations in the model accounts for the higher errors in the predicted behavior of the worst placement that generates the heaviest contention pattern.

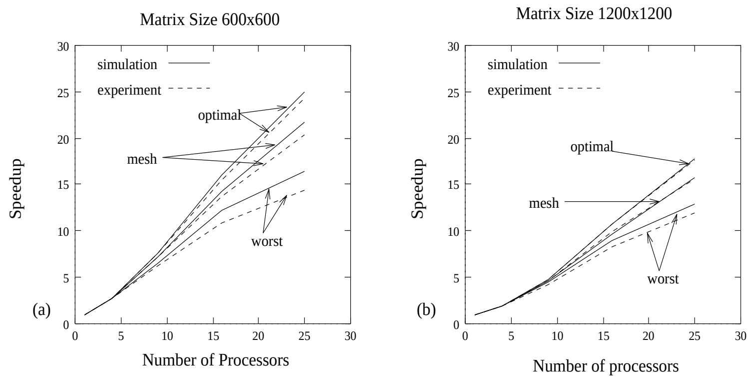

Fig. 4. Predicted and experimental speedup curves for three mapping schemes for a matrix of size 600×600 and a matrix of size 1200×1200.

5 Conclusions and Future Work

In this paper the performance implications of network contention on the scalability of SPMD applications executing on a wormhole routed multicomputer with deterministic routing have been presented. A communication intensive application was used to stress test the effects of network contention. Speedup has been predicted using a non-product form queueing network model parametrized with actual measurements. Modeling results are validated by experimentation on the Paragon under SUNMOS, an operating system that provides high communication performance.

It has been argued that the number of processors assigned to the application is more important than the partition geometry or the process mapping withing the assigned partition [LKD94]. In this paper, we have demonstrated that the data and/or process placement scheme can become a source of internal network contention and limit workload scalability, even for a small number of processors. Future work includes studying the impact on the workload execution time of external contention due to paths to shared system resource (e.g., RAID disks) traversing the given partition. An extention of this work is to further investigate workload scalability under such conditions.

Future work also includes the development of a more general model among whose parameters are the number of processors, the data/process mapping, and the task graph along with the temporal distribution of the workload's traffic pattern. Finally, the results of this work can be applied in processor scheduling policies for wormhole routed multiprocessors.

Acknowledgements
The helpful criticisms and suggestions provided by Larry Dowdy have significantly improved this paper.

References

[AV94] V.S. Adve, M.K. Vernon, "Performance analysis of mesh interconnection neworkswith deterministic routing," *IEEE Trans. on Parallel and Distributed Systems*, Vol. 5, No. 3, pp. 225-246, March 1994.

[BCMP75] F. Baskett, K.M. Chandy, R.R. Muntz, F. Palacios, "Open, closed, and mixed networks of queues with different classes of customers," *Journal of the ACM*, Vol. 22, No. 2, pp. 248-260, April 1975.

[CE90] S. Chittor, R. Enbody, "Performance degradation in large wormhole-routed interprocessor communication networks," in Proceedings of *1990 Int'l Conference on Parallel Processing*, vol. 1, pp. 424-428, 1990.

[DS86] W.J. Dally, C.L. Seitz, "The torus routing chip," *Journal of Distributed Computing*, Vol. 1, No. 3, pp. 187-196, 1986.

[LKD94] P. Krueger, T. Lai, V.A. Dixit-Radiya, "Job scheduling is more important than processor allocation for hypercube computers," *IEEE Trans. on Parallel and Distributed Systems*, Vol. 5, No. 5, pp 488-497, May 1994.

[NM93] L.M. Ni, P.K. McKinley, "A survey of wormhole routing techniques in direct networks," *IEEE Computer*, Vol. 26, No. 2, pp. 62-76, February 1993.

[MMR93] B. Maccabe, K.S. McCurley, and R. Riesen, "SUNMOS for the Intel Paragon: A Brief User's Guide," Sandia National Laboratories Tech. Report, 1993.

[SCRD94] E. Smirni, C.A. Childers, E. Rosti, L.W. Dowdy, "Thread Placement on the Intel Paragon: Modeling and Experimentation," Proc. of the *3rd Inter. Workshop on Modeling, Analysis, and Simulation of Computer and Telecommunication Systems*, pp 226-231, North Carolina, USA, 1995.