"Architectural Considerations, Software Support and Compiler Issues in Multi-Computer Implementation"

12. PERSONAL AUTHOR(S)
Dharma P. Agrawal & Jon Mauney

13a. TYPE OF REPORT
Final

13b. TIME COVERED
FROM 9/5/85 TO 10/31/88

14. DATE OF REPORT (Year, Month, Day)
December 28, 1988

15. PAGE COUNT
28

16. SUPPLEMENTARY NOTATION
The view, opinions and/or findings contained in this report are those of the author(s) and should not be construed as an official Department of the Army position, policy, or decision, unless so designated by other documentation.

18. SUBJECT TERMS (Continue on reverse if necessary and identify by block number)

19. ABSTRACT (Continue on reverse if necessary and identify by block number)
We have studied the problem of efficient execution of programs on parallel computers and on the B-HIVE architecture in particular and the software support required, concentrating on programming language implementation.

In a loosely-coupled multiprocessing environment, in which processors communicate over a generalized hypercube or similar network, the cost of sharing data among processes is quite high. We have investigated techniques of minimizing the total communication cost in parallel programs. We have also looked into ways of introducing parallelism in combinatorial problems and observed the impact of randomization on the system speedup. Besides the speedup, the network connectivity and data distribution in parallel & distributed systems play a very important role in determining the system performance. We have computed the reliability of such systems under different environment. In addition, we have studied the design of topologies with limited connections and which could be appropriate for both LAN & MAN applications.

20. DISTRIBUTION / AVAILABILITY OF ABSTRACT

21. ABSTRACT SECURITY CLASSIFICATION
Unclassified

22a. NAME OF RESPONSIBLE INDIVIDUAL
Dharma P. Agrawal

22b. TELEPHONE (Include Area Code)
(919) 737-3984

22c. OFFICE SYMBOL
ECE
Architectural Considerations, Software Support and Compiler Issues in Multi-Computer Implementation

Final Report
by

Dharma P. Agrawal
Professor,
Department of Electrical and Computer Engineering
North Carolina State University, Box 7911
Raleigh, NC 27695-7911
Tel: (919)737-3984

and

Jon Mauney
Assistant Professor,
Department of Computer Science
North Carolina State University, Box 8206
Raleigh, NC 27695-8206
Tel: (919)737-7889

Dated December 28, 1988

Research Agreement No. DAAG29-85-k-0236

submitted to

U.S. Army Research Office
Electronics Division
P.O. Box 12211
Research Triangle Park, NC 27709-2211

The View, Opinions, and/or findings contained in this report are those of the authors and should not be construed as an official Department of Army position, policy, or decision, unless so designated by other document.
1 Statement of Problem Studied

We have studied the problem of efficient execution of programs on parallel computers — and on the B-HIVE architecture in particular — and the software support required, concentrating on programming language implementation.

In a loosely-coupled multiprocessing environment, in which processors communicate over a generalized hypercube or similar network, the cost of sharing data among processes is quite high. We have investigated techniques of minimizing the total communication cost in parallel programs. We have also looked into ways of introducing parallelism in combinatorial problems and observed the impact of randomization on the system speedup. Besides the speedup, the network connectivity and data distribution in parallel & distributed systems play a very important role in determining the system performance. We have computed the reliability of such systems under different environment. In addition, we have studied the design of topologies with limited connections and which could be appropriate for both LAN & MAN applications.

2 Summary of the most important results

We are interested in techniques for arranging parallel code onto loosely-couple processors so as to minimize communication overhead and maximize processing speed. This is a synthesis problem, similar to the problem of optimization and code-generation in a traditional compiler. It is very different from the analysis problem of discovering potentially parallel operations in the program. Much previous research has centered on designing programming languages that express parallelism explicitly, and on program analysers that discover implicit parallelism in sequential codes. Our work is independent of these issues and compatible with either approach.

We use a medium grain parallelism model to minimize communication overhead [1-10]. A medium grain model is shown to be an optimum way of merging fine grain operations into parallel tasks such that the parallelism obtained at the small grain level is retained and communication overhead is decreased. Our “vertical partitioning” and scheduling techniques have been evaluated by the simulation of ten EISPACK subroutines [7]. The vertical partitioning model clearly outperforms the model without the vertical partitioning.
A new communication model has been introduced [1], allowing additional overlap between computation and communication. Simulation results indicate [8] that the medium grain communication model shows promise for automatic parallelization for a loosely-coupled multiprocessor system.

Since most of the computation in a program is performed inside loops, parallelization of loop structures is an important topic and has been extensively studied in the past. The particular program of loop execution on a loosely-coupled parallel processor has not received much attention, however. We introduce a compensated loop scheduling that adjusts to the delays involved in distributed data for parallel execution. We also propose a nested loop scheduling that allows heterogeneous loop allocation [11]. Heterogeneous loop allocation adjusts to dependencies within loops and in some cases permits better utilization of processors in the inner loops. Additional experimentation [12] has been done to do parallel recursive least squares computation in distributed memory multiprocessors. All these problems have been studied for efficient execution of programs on parallel computers in general, and on the B-HIVE architecture [13,14] in particular. We have shown that for some optimization problems, such as backtracking [15], and branch and bound [16], a randomized search algorithm yields high performance on a parallel processor. A major advantage of randomized search is that very little interprocessor communication is required.

There are many other factors that influence the performance of a parallel system. The impact of single faults on the system performance of a class of cluster-based multiprocessors have been considered in [17]. The effect of network connectivity and data distribution on various reliability and performance parameters have been studied in [18-21]. A comparison of various fault-tolerant multistage interconnection networks, has been performed in [22,23]. The requirements of LANs and MANs in terms of connectivity, are different than a general parallel systems and appropriate topologies for such doubly connected multidimensional networks and their characteristics, have been covered in [24-27]. Thus, the research results encompasses more than the areas just described in the proposal.

3 List of all publications and technical reports


4 List of all participating scientific personnel

4.1 Faculty
1. Dr. Dharma P. Agrawal
2. Dr. Jon Mauney
3. Dr. Suresh Rai (visiting)

4.2 Graduate Students
1. Mr. Ja-Song Leu, completed Ph.D., July 1987.
3. Mr. A. Kumar, doctoral student, expected to finish in Summer 1989.
4. Mr. Sukil Kim, doctoral student, expected to finish in Summer 1989.
5. Mr. T.Y. Chung, doctoral student, expected to finish in Summer 1989.
7. Mr. I.O. Mahgoub