ALGORITHM DESIGN FOR SCIENTIFIC COMPUTATION FOR HIGHLY
PARALLEL MULTIPROCESSOR SYSTEMS(U) PURDUE RESEARCH
FOUNDATION LAFAYETTE IN D GANNON 1987 AFOSR-TR-87-1454

UNCLASSIFIED AFOSR-85-0123

Dennis Gannon
Dept. of Computer Science
Indiana University, Bloomington Indiana

Project Summary

As part of the AFOSR Fast Algorithms Initiative, the project focused on the design of parallel algorithms and the related software design problems associated with multiprocessor systems. The research work was divided into two phases. The primary emphasis of the first phase was to study new algorithm ideas for solving the large numerical linear algebra problems associated with two and three dimensional elliptic P.D.E. problems. The work in the second phase of the research was directed toward understanding the software mechanisms needed to "map" these algorithms to existing parallel computers. In the following paragraphs we detail our work in both areas.

One area of algorithm design research focused on the problem of deriving sound theoretical results for an experimental multigrid solver. This algorithm (invented by Gannon and Van Rosendale in 1983) uses a novel technique for exploiting concurrency. Most multigrid methods work by sequentially solving a sequence of "easier" problems on a nested family of grids. For parallel computation these methods suffer from the problem that very small "coarse" grids do allow much concurrency. If one parallelized the work on the large "fine" grids, then these coarse grids rapidly become a bottleneck. In terms of complexity analysis, a problem on a \( n \times n \) grid has a parallel computation time of \( O(\log^2(n)) \). Our concurrent multigrid method works by splitting the problem so that all grid levels may be solved in parallel rather than sequentially as in the standard scheme. If you experiment with this technique is that unlike the standard methods which have the property that the spectral radius of the convergence is independent of the grid size, the concurrent multigrid scheme shows a slow degradation in spectral radius. Experimental work showed that this new method was still superior for large numbers of processors, but until last year we had no proof of these claims. As part of the work on this project we showed that the algorithm can be formulated so that the parallel complexity is \( O(\log(n) \log(\log(n))) \). Though this may not look like much of a difference, for large \( n \) it is a substantial improvement. This result has been recently published in the Journal of Parallel and Distributed Computing.

Another area where we did work on basic algorithm research was to look at applying the ideas developed for parallel numerical methods to problems in computer graphics. In particular, we looked at multiprocessor ray tracing for non-shared memory machines. One very simple solution to this problem is to divide pixel space among the processors and run all computations independently. For systems with very large numbers of very simple processors, this requires that the physical data base must be duplicated in each processor. Using ideas taken from systolic array theory, we showed that it is possible to build a pipelined tree to do ray tracing such that each processor is involved only with one geometric primitive and no duplication of the data base is needed. This work resulted in a Ph.D. Thesis.

The second phase of our research focused on the problem of programming parallel algorithms on highly concurrent systems. The basic problem here is that while an algorithm designer may be able to express his ideas as highly parallel algorithms, each different parallel computer requires that the programmer express the concurrency in a different way. Some machines want tightly organized vectors of parallel operations while others want large independently executing tasks. It is often
As part of the AFOSR Fast Algorithms Initiative, the project focused on the design of parallel algorithms and the related software design problems associated with multiprocessor systems. The research work was divided into two phases. The primary emphasis of the first phase was to study new algorithm ideas for solving the large numerical linear algebra problems associated with two and three dimensional elliptic P.D.E. problems. The work in the second phase of the research was directed toward understanding the software mechanisms needed to "map" these algorithms to existing parallel computers. In the following paragraphs we detail our work in both areas.
harder for the parallel programmer to decide which aspects of concurrency to ignore than it is to find the concurrency in the first place. For large numerical problems we have been exploring an idea first proposed by Jack Dongarra. Write algorithms in terms of a library of extended linear algebra primitives called the BLAS2 and BLAS3. These go beyond the basic scalar-vector BLAS operations to include vector-matrix and matrix-matrix operations. Then, on each parallel machine, we design the fastest possible parallel implementation of this BLAS2-3 library. While this may sound easy, it is very difficult for non-shared memory machines. Working with graduate student Jairo Panetta, we designed a set of "systolic blas" for message based non-shared memory machines where the local memories are mapped to part of the address space of a host processor. (This includes the Pringle/CHiP system at Indiana and Washington as well as the Thinking Machines, Connection Machine.) This work was published in a SIAM volume as well as a part of Panettas Ph.D. Thesis.

The final phase of this work is still under way. We attempting to encapsulate the techniques used by programmers to restructure computation to take advantage of the special features of a particular machine or special library like the BLAS2-3. The objective is to build a set of tools, and finally an expert system, that can be used by programmers of these special purpose computers. Preliminary results here look very promising.

Individuals Supported by the Project

In addition to the principle investigator, there were four graduate students supported by this project. Three have completed Ph.D.s, and the other will soon.

1. Jairo Panetta, Ph.D. Dec. 1985. His thesis was described above. He is currently in Brazil (his home country), but he is still collaborating with people at Argonne and Purdue.
2. Yeou-Huei Hwang, Ph.D. June 1985. His work is on parallel algorithms for computer graphics and is described above. He is currently working for Bell Labs.
3. Alejandro Kapauan, Ph.D. Dec. 1985. This student was responsible for the design and construction of the Pringle Multiprocessor system that was the basis of the work described above. He is currently working for Bell Labs.
4. Ko-Yang Wang, Ph.D. expected May 1987. His basic work on expert systems for parallel programming is described above. A paper giving an initial study of the problem is cited below. He will probably go to work for IBM Research at Yorktown Heights.

Publications Based on Research Supported by the Grant.

END DATE
FILMED
JAN 1988