LA-UR-93- 435

LA-UR--93-435

DE93 007325

Los Alamos National Laboratory is operated by the University of California for the United States Department o

FEB 1 1 1993

OSTI

TITLE MASSIVELY PARALLEL HYDRODYNAMICS ON UNSTRUCTURED GRIDS

AUTHOR(S): Manjit S. Sahota, T-3
Harold E. Trease, X-3

SUBMITTED TO: 1993 SCS Simulation Multiconference, Key Bridge Marriott, Arlington, Virginia, March 29–April 1, 1993

DISCLAIMER

This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.

By acceptance of this article, the publisher recognizes that the U.S. Government retains a nonexclusive, royalty-free license to publish or reproduce the published form of this contribution, or to allow others to do so, for U.S. Government purposes.

The Los Alamos National Laboratory requests that the publisher identify this article as work performed under the auspices of the U.S. Department of Energy.

MASTER

Los Alamos National Laboratory
Los Alamos, New Mexico 87545

FORM NO. 836 R4
ST. NO. 2629 5/81

RESTRICTED TO U.S. ONLY

MASSIVELY PARALLEL HYDRODYNAMICS ON UNSTRUCTURED GRIDS

Manjit S. Sahota and Harold E. Trease
Los Alamos National Laboratory
Los Alamos, New Mexico 87545

ABSTRACT

We investigate the feasibility of using massively parallel Single Instruction Multiple Data (SIMD) and Multiple Instruction Multiple Data (MIMD) computer architectures for explicit three-dimensional unstructured-grid finite-element Lagrange and finite-volume free-Lagrange hydrodynamics. We compare the Cray-YMP, Connection Machine 200 (CM-200), and CM-5 computational times required for the foregoing algorithms. Although we observe very large global communication penalty (~40-95% depending upon the problem) on the CM for these algorithms, moderate to good speedups over a single-processor Cray-YMP are obtained. Also, when the Fortran 90 code written for the CM is ported back to the Cray, we observe ~30-40% speedups over the original vectorized Cray coding. To reduce the global communication cost, we investigate the possibility of using an automatic generalized domain decomposition technique for MIMD architectures (CM-5) currently under development at the Thinking Machines Corporation.

INTRODUCTION

A major impediment to efficiently implementing unstructured-grid hydrodynamics on distributed-data massively parallel computer architectures is the global data communication due to the unstructured nature of the grid. A large fraction of the computer time is spent in global communication instead of performing the actual calculations. We have observed over 90% of the total central processor unit (CPU) times being spent on global communication on the Thinking Machines Corporation's (TMC's) Connection Machines 200 and 5 (CM 200 and CM 5). In spite of this overwhelming communication penalty, our algorithms appear to run several factors faster on the CM compared to a single processor Cray YMP. Thus there is a significant potential of further enhancing the computational efficiency by either minimizing the extent of global communication by rearranging the data, or by increasing the efficiency of the global communication. We have attempted to achieve both in the current work.

We mapped two three-dimensional hydrodynamically different algorithms onto the CM machines. The first algorithm employs the free-Lagrange method (FLM) (Mandell and Trease 1989; Painter and Marshall 1991; Sahota 1991a, 1991b; Sahota and Trease 1991; Trease 1991) that uses a dynamic variable-connectivity median mesh for control volumes. However, all hydrodynamic computations are done at a tetrahedral level by partitioning and accumulating tetrahedral results at the computational cell centers (Sahota 1991a; Sahota and Trease 1991). All variables are cell centered.

The second algorithm uses the finite-element method (FEM) on a staggered Lagrangian tetrahedral grid with eroding slide lines and offers a multitude of rate-dependent plasticity models (Johnson, Colby, and Vavrick 1979; Johnson and Stryk 1987; Johnson and Stryk 1989). The elements are defined by tetrahedra where all thermodynamic variables are centered. The vertices of tetrahedra represent nodes where velocities are defined.

The next section describes our implementation of the foregoing methodologies to the massively parallel environment and delineates our results. The Domain Decomposition section provides a way of minimizing global communication. Although specifically aimed for the CM, this approach will be found useful on any MIMD computer including workstation clusters. Finally, our major conclusions are presented in the Conclusions section.

MASSIVELY PARALLEL IMPLEMENTATION AND RESULTS

As the original FLM algorithm was developed with vectorization and parallelization on the Cray YMP, its mapping onto the massively parallel environment was relatively straightforward with the exception of dynamic mesh reconnections, which were inherently serial on the Cray. (Serial reconnections on the Cray were not considered a serious drawback as only ~10% of the computational time was spent in reconnections.) A new algorithm had to be developed for parallel recon nections in the massively parallel environment (Trease 1991). However, although efficiently vectorized for the Cray, the FEM was developed to also run on a variety of computer architectures including the scientific workstations in any number of space dimensions with different coordinate systems. As a result, we had to rewrite major portions of the algorithm for the CM. (The eroding slide lines were not ported to the CM, but the work is in progress.)

After restructuring the two algorithms in the Fortran 90 standards and mapping them onto the CM, we ported them back to the Cray-YMP. We observed speedups of ~30-40% over the original Cray versions. The results of our comparisons are based on the new faster Cray run times.

Table I compares the run times relative to the Cray YMP for our sample calculations. The numbers in Table I are normalized to the new Cray times. Parallelization on the Cray was done using the Autotasking only. Thus, some further improvement in the speedup on the multiprocessor Cray is possible. All calculations on the CM were performed using the double precision (64 bit) arithmetic. Because of the unavailability of the full CM, some of the CM calculations in Table I were carried out on a part of the machine and the results were estimated for the full machine. However, we have performed numerous calculations on the different CM partitions and have found the results to be surprisingly linear for the sizes of the sample problems used to generate Table I.

TABLE I: NORMALIZED RUN TIMES FOR DIFFERENT COMPUTER ARCHITECTURES

Computer Architecture	**Normalized CPU Time**	
	FLM	**FEM**
1-Processor Cray-YMP	1	1
8-Processor Cray-YMP	5	-[a]
CM-200 (2048 Nodes) with Fortran 90 Gather/Scatter	2	2
CM-200 (2048 Nodes) with CMSSL Library Gather/Scatter to optimize the message delivery process.	[b]	8
CM-5 (1024 Sparc Nodes without Vector Units)	4	11

a. Inconclusive results because the calculations were done in a timeshared environment.

b. Message delivery optimizations are of limited value because reconnections in the FLM require reevaluation of the message delivery trace, the optimization of which is expensive.

We note from Table I that the speedup for CM-200 over Cray is about the same (a factor of ~2) for the two algorithms for straight Fortran 90 coding. The fourth row in Table I shows the result of preestablishing the communication paths using the TMC's CMSSL library. We precomputed and used message delivery optimizations for basic data motion and combining operations. We selected the Fastgraph option, which optimizes the use of CM-200 hypercube wires by scheduling the use of individual wires by each message. This resulted in a surprizing speedup of an additional factor of four over the straight Fortran 90 gather/scatter. We tried this optimization for the FEM only because this algorithm uses a fixed connectivety and the communication had to be established only once. We do not know at this stage if any significant improvement in the FLM run time can be obtained by preestablishing communication because changes in the grid connectivity require reestablishing the communication path, which is quite expensive for the Fastgraph option. There are other options that are faster in preestablishing the communication paths, but they are not as efficient as the Fastgraph on CM-200.

We also tried several message delivery optimizations on the CM-5 with only limited success.

The last row of Table I shows the results for CM-5 without the Vector Units (with Sparc chip). Work is in progress for assessing the performance of of the Vector Units. (There are four Vector Units to a Node.) The FLM does not fare as well as we expected. Althouth a speedup of a factor of 11 over a single processor Cray YMP appears impressive, it is only a factor of 1.4 over the speed of CM-200. However, we are optimistic that higher speedups will be attained with the Vector Units. The primary reason for the FEM running so much faster than the FLM is because we significantly reduced the global indirect communication in the FEM by copying the desired equation of state data to the elements at the start of the problem.

DOMAIN DECOMPOSITION

Special consideration must be given to the minimization of global communication for unstructured grids because of indirect addressing. For unstructured grids, the communication costs related to indirect addressing are minimized by taking advantage of the CM-5's dedicated hardware for local indirect address ing within each processing element. To fully utilize this hardware feature, we must use a mesh reordering technique to give us the minimum global communication versus local communication ratio. The technique that we use to do local domain decomposition is called nested spectral dissection (Pothen, Simon, and Wang 1992), also known as recursive spectral bisection (Simon 1991). The idea behind this type of local domain decomposition is quite simple. A random unstructured mesh is reordered such that, for a given set of processors, an (near) optimal mapping in communication space is achieved where the amount of interprocessor communication is minimized (and thus the amount of intraprocessor communication is maximized). Thus, the code takes advantage of the fast local communication hardware provided on the CM-5. Another beneficial side-effect of maximizing the local processor communication is that there are fewer interprocessor messages to deliver; thus making more effective use of the global router due to the reduced message traffic. The data structures used in the two algorithms are all one-dimensional and the programming language used is Fortran 90 (Connection Machine Fortran, CMF). The arrays are smeared across the machine such that each processor has (approximately) the same amount of work to do (i.e., we let Fortran 90 load balance the work to be done). We use the CMF global/local programming model to handle computing tasks that are not well balanced for Fortran 90.

Figure 1 shows a three-dimensional grid for an octant of a sphere for the FLM. In order to limit global communication to the vicinity of each node, we reordered tetrahedra using the TMC's CMSSL partitioning library. Figure 2 shows 16 clusters of tetrahedra corresponding to the grid shown in Fig. 1. Each cluster is then loaded onto a different processor, thereby minimizing global communication by maximizing local processor communication. We are currently in the process of evaluating the improvements in the run times using such an approach. Some initial data supplied by the TMC show speedups of ~8 over the original global communication (D.M. Fraser and J. Myczkowski, TMC, private communication).

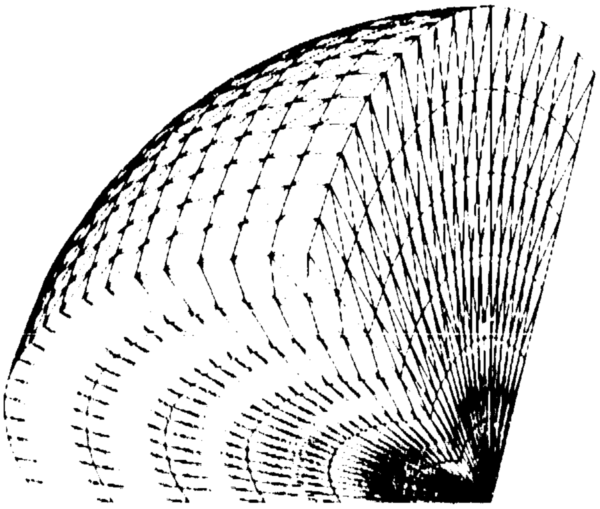

Fig. 1. Three dimensional tetrahedral grid for the octant of a sphere.

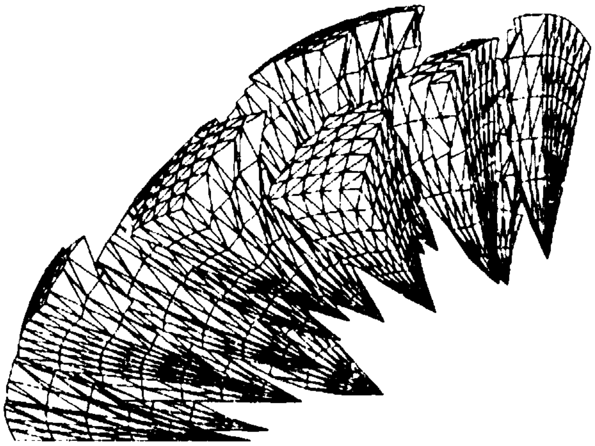

Fig. 2. Automatic domain decomposition of the tetrahedral grid shown in Fig. 1 into 16 clusters.

It is important to note the advantage of the foregoing approach over the conventional approaches used on structured grids. A manual domain decomposition on unstructured grids is extremely cumbersome and impractical. This automated approach requires minimal interaction by the user. Although in theory, for the FLM method, the domain decomposition is required each time the mesh reconnections are made, we have noticed only a slow deterioration of the original decomposition with time. Thus, new domain decomposition may be infrequently required during the course of calculation.

Such a domain decomposition sets an upper limit on the average value of the communication costs because volume communication costs (i.e., within processor communicaiton) increase faster than surface communication costs (i.e., global communication costs). As a result, after a certain limit the absolute value of the global communication costs do not increase in proportion to the problem size. Therefore, as the problem gets larger the fraction of time spent in global communication should actually go down; and perhaps for vary large problems unstructured grid grind times may even become comparable to those for the structured grids.

CONCLUSIONS

We assessed the feasibility of using massively parallel SIMD and MIMD computer architectures of the CM family on unstructured-grid Lagrangian and free-Lagrangian three-dimensional explicit hydrodynamics. In spite of a heavy penalty in global communication cost (~40-95%), we still observe some significant speedups on the CM compared to a single-processor Cray-YMP (factors ranging from 2 to 11 depending upon the problem and the algorithm). Porting the algorithm back to Cray also results in a speedup of ~30-40% over the original Cray code.

Preestablishing global communication trace on the CM-200 results in an additional factor of four improvement in run time, however such a trace setup has a large initial penalty and may be prohibitively expensive for dynamic grids. Also, this approach does not work as well on CM-5.

Automatic domain decomposition has a significant future potential, not only for the CM-5, but also for other computer architectures including the workstation clusters. Such an approach sets an upper limit in absolute terms on the global communication cost. As a result, the communication cost may tend to vanish for very large problems. The research on [illegible] domain decomposition and the assessment of the performance of the CM-5 Vector Units continues.

ACKNOWLEDGEMENTS

This work was performed under the auspices of the U.S. Department of Energy by the Los Alamos National Laboratory (LANL) under Contract W-7405-ENG-36. We thank F.L. Addessio and P.J. Maudlin, LANL; G.R. Johnson and R.A. Stryk, Alliant Techsystems Inc.; and D.M. Fraser and J. Myczkowski, TMC; for technical discussions.

REFERENCES

Johnson, G.R.; D.D. Colby; and D.J. Vavrick. 1979. "Three Dimensional Computer Code for Dynamic Response of Solids to Intense Impulsive Loads." *International Journal of Numerical Methods in Engineering*, Vol. 14.

Johnson, G.R. and R.A. Stryk. 1987. "Eroding Interface and Improved Tetrahedral Element Algorithms for High Velocity Impact Computations in Three Dimensions." *International Journal of Impact Engineering*, Vol. 5.

Johnson, G.R. and R.A. Stryk. 1989. "Dynamic Three-Dimensional Computations for Solids, with Variable Nodal Connectivity for Severe Distortions." *International Journal of Numerical Methods in Engineering*, Vol. 28.

Mandell, D.A. and H.E. Trease. 1989. "Parallel Processing a Three-Dimensional Free-Lagrange Code: A Case History." *The International Journal of Supercomputer Applications*, Vol. 3, No. 2, pp. 92-99.

Painter, J.W. and J.C. Marshall. 1991. "3-D Reconnection and Fluxing Algorithms." *Lecture Notes in Physics, Advances in the Free-Lagrange Method*, Springer-Verlag Press, pp. 139-148.

Pothen, A.; H.D. Simon; and L. Wang. 1992. "Spectral Nested Dissection." Pennsylvania State University report CS-92-01.

Sahota, M.S. 1991a. "An Explicit-Implicit Solution of the Hydrodynamic and Radiation Equations." *Lecture Notes in Physics, Advances in the Free-Lagrange Method*, Springer-Verlag Press, pp. 57-65.

Sahota, M.S. 1991b. "Delaunay Tetrahedralization in a Three-Dimensional Free-Lagrangian Multimaterial Code." *Lecture Notes in Physics, Advances in the Free-Lagrange Method*, Springer-Verlag Press, pp. 130-138.

Sahota, M.S. and H.E. Trease. 1991. "A Three-Dimensional Free-Lagrange Code for Multimaterial Flow Simulations." *Proceedings of the ASME-JSME International Symposium on Liquid-Solid Flows*, Portland, Oregon, June 24-27.

Simon, H. 1991. "Partitioning of Unstructured Problems for Parallel Processing." *Proceedings of Conference on Parallel Methods on Large Scale Structural Analysis and Physics Applications*, Pergammon Press.

Trease, H.E. 1991. "Parallel Nearest Neighbor Calculations." *Lecture Notes in Physics, Advances in the Free-Lagrange Method*, Springer-Verlag Press, pp. 149-156.