

# Computational Science Experiences on the Intel Touchstone DELTA Supercomputer \*

RICK STEVENS

Mathematics and Computer Science  
Argonne National Laboratory  
Argonne IL 60439-4844

## Abstract

*Argonne National Laboratory's involvement in the Concurrent Supercomputing Consortium (CSC)—the consortium that owns and operates the Touchstone DELTA System—has been motivated primarily by the need to make high-performance computing resources available to our computer science research groups and to those researchers using large-scale computation in their scientific work. Since early May 1991 when the Touchstone DELTA System (DELTA) was first installed at Caltech, Argonne has been actively using it to investigate a variety of scientific problems. In this short paper I describe some of our experiences and progress in global climate modeling, computational biophysics, and computational quantum chemistry. At the end of the paper I have tried to draw some conclusions regarding the usability of machines like the DELTA.*

## 1 Overview

With the installation of the DELTA those of us working on MIMD parallel programming tools and applications finally had access to a computer favorably comparable to the largest machines employed in scientific computing. To say that we had been working all these years for that moment would be a slight exaggeration, but it would have captured the spirit of the moment last spring when the machine ran the first real program. In many ways the existence DELTA is tangible proof that a large-scale general purpose parallel system is able to compete head on with traditional vector-supercomputing.

The DELTA, due to its large memory, large parallel filesystem and peak performance, has marked a

\*This research supported in part by the Applied Mathematical Sciences subprogram of the office of Energy Research, U.S. Department of Energy, under contract W-31-109-Eng-38.

turning point in the modern evolution of supercomputers. We are in the new world. But we've arrived much faster than anyone had expected and we are not wholly prepared. At Argonne we've been developing a number of programs to use the DELTA, at last count no less than 15 projects are underway. In this short paper I will describe only a few. Our work in global climate modeling has been formed around a very active collaboration with Oak Ridge National Laboratory (ORNL) and the National Center for Atmospheric Research (NCAR). A new effort in computational biophysics is rapidly producing a series of models for the DELTA and finally I will briefly introduce some work in computational chemistry, a discipline where Argonne has had a long and distinguished effort.

## 2 Global Climate Modeling

During the past five years, significant progress has been made in the development of massively parallel computers. Indeed, they are now surpassing conventional vector-supercomputers on a routine basis for many applications. Many supercomputer vendors are expected to offer massively parallel products in the teraflop range by the mid-nineties. In the field of climate modeling the challenge is to develop new models that can exploit massive parallelism and thus take advantage of the latest hardware developments.

At Argonne we are developing a series of models that are explicitly designed to run efficiently on parallel hardware. For moderate-scale parallel machines (50-500 processors), a straightforward decomposition of the model dynamics is generally sufficient to achieve high computational efficiency. In these parallel models the "model physics" component is left as a column process contained within one processor. If, however, we increase the number of processors while holding the model resolution fixed (to decrease the time-to-solution for a particular problem), it becomes appar-

ent that a three-dimensional decomposition will be needed. Unlike the dynamics kernel, the model physics components contain dozens of algorithms with widely varying communications requirements. Thus, an interesting and challenging set of parallel programming problems arises. Decreasing the granularity of the parallel model is significant, since we sometimes are interested in the fastest time-to-solution at relatively modest resolutions.

Maintaining an effective load balance in the parallel model is also important. A number of methods have been developed for determining the magnitude of the load imbalance and for statically and dynamically adjusting the allocation of processors to model physics columns [6].

Coupling an atmospheric model to an ocean model and biosphere model is the next step to producing a fully integrated earth systems model. Such an integrated model is essential if we wish to fully understand the interaction of ocean circulation and the biosphere with any potential global change. Today, ocean models are routinely developed in isolation from atmospheric models and are difficult to couple, physically and numerically. Coupling these in the context of a massively parallel implementation is a serious challenge and we are developing tools and techniques to solve these problems.

At Argonne we have work underway developing new numerical methods for climate modeling, developing parallel implementations of existing methods and developing new schemes for coupling models. Related efforts are underway to develop parallel implementations of meso-scale models such as MM4 and biosphere models such as BATS.

## 2.1 Test Codes and New Methods

As a first stage in our strategy to parallelize CCM2 for the DELTA, we developed two test codes. The first was a simple spectral-transform barotropic vorticity model (BVE) from [7]. This code was used to understand the basic parallel algorithms needed and to debug program development tools. Later a more complicated model from the National Center for Atmospheric Research (NCAR) was used as a test code. This code, a spectral-transform shallow water code (SWE), has been used extensively for performance analysis and modeling and for benchmarking runs on the DELTA [4]. It contains the same basic dynamics elements as the much larger CCM2. It has been very helpful to have a series of codes of increasing complexity (BVE  $\approx$  600 lines, SWE  $\approx$  6000 lines and CCM2  $\approx$  60,000 lines) on which to test parallelization meth-

ods and tools. The climate modeling group at Argonne has developed parallel versions of each of these models using Fortran and PCN (a parallel language developed jointly by Argonne and Caltech [2]).

At the same time parallel implementations of existing numerical methods were being developed, new numerical methods were also being investigated. A particularly successful system was developed by P'liang Chern and Ian Foster [3]. In this so-called "Icosahedral Method" the sphere was tessellated into an icosahedron and a control volume method was used to integrate the shallow-water equations. A parallel version of this method was one of the first applications to run on the DELTA. It is implemented in C and PCN, obtained about 70% parallel efficiency and achieved nearly two gigaflops sustained performance. This code later became part of the DELTA acceptance test and part of Intel's internal MPP test suite. It has been used to investigate the vortex merging behavior exhibited by the interaction of Rossby-Haurwitz waves with zonal flow.

## 2.2 Parallel CCM2

A major goal of the DOE-CHAMMP program is to develop massively parallel implementations of leading global climate modeling systems. The focus of the joint Argonne, NCAR, Oak Ridge development effort is CCM2, the latest version of NCAR's series of CCMs (Community Climate Models). CCM2 is designed to run on Cray class multiprocessors with a small number of shared memory processors. Its standard resolution is T42 (roughly 300km) with about 20 vertical layers. The standard resolution produces a Gaussian mesh of 64 by 128 or 8192 cells. However the model can run at much higher resolutions and indeed that is one of the primary reasons for producing a massively parallel implementation. We expect to run with resolutions of greater than T169 (about 80km) on the DELTA in the near future. A typical run of the parallel CCM2 would simulate 10 years and produce over 200 Gigabytes of data. We expect to complete such runs on the DELTA in less than one week of machine time.

NCAR's CCM2 uses a spectral transform algorithm with semi-Lagrangian transport for the model dynamics. Our initial parallel implementation decomposes the problem in the two horizontal dimensions, leaving vertical processes unchanged. This means that we need a parallel FFT, parallel Legendre transform and parallel semi-Lagrange. From extensive performance modeling [4] it was determined that the most efficient parallel implementation would use about equal numbers of processors in each horizontal dimension. Sev-

eral different physical mappings of processors are being tried, including mapping the FFTs in blocks to avoid contention in the network.

### 2.3 Parallel MM4

Other efforts at Argonne, particularly in the Environment Assessment and Information Systems Division have been focusing on the use of both global and regional long term climate simulations for policy related studies. One interesting problem is that of driving a meso-scale (regional) model with boundary data from a GCM (General Circulation Model). Several members of the climate modeling group have produced a parallel implementation of the Penn-State/NCAR MM4 model. MM4 is a regional weather model capable of high-resolution forecasts over a limited area, typically 5,000km square. Since MM4 is a regional model it must be driven by boundary data. This data can come from observations or from a global model. An interesting set of experiments are currently underway that will run MM4 nested inside a GCM with one-way coupling. This coupling allows the GCM to "force" the meso-scale model but does not permit the meso-scale to affect the global model. Using this coupled model one might be able to determine regional effects of global change.

MM4 is a finite-difference model with a standard resolution of 80km. Two parallel versions of this model have been developed at Argonne. A one-dimensional version suitable for small numbers of processors (8-10) and a two-dimensional version suitable for 50-100 processors. The model is currently running on the DELTA and with a small number of processors is performance competitive to a Cray XMP. Multiple parallel MM4 jobs are currently run on the DELTA by partitioning the machine into subpartitions. Future plans call for MM4 to be run simultaneously with a GCM that will send it's boundary data directly to MM4 in real time.

## 3 Computational Biophysics

Many computational problems in molecular biology require modeling and reasoning about constrained systems. A particularly interesting case occurs in the reconstruction of biomolecular structures from x-ray crystallographic data—a highly labor-intensive process requiring many iterations from a probable guess to numerical simulation to comparison with experimental data and subsequent refinement of the predicted structure. The refinement process is currently one of the most severe bottlenecks preventing a major

increase in the number of protein structures solved. A related problem is that of predicting the macroscopic behavior of new proteins. The combination of molecular structure modeling and macro-behavior modeling (interaction dynamics and polymerization rates, etc.) will allow us to combine all available experimental and computational data in the attack on the protein conformation problem.

A major component of Argonne's program in computational biophysics is to accelerate this structure refinement process by combining automated reasoning methods and parallel simulation to the problem of deducing a probable protein structure subject to certain restraints and constraints. Our eventual goal is to offer the crystallographer a set of tools that will significantly reduce the time currently needed for the reconstruction of biomolecular structures from crystallographic data.

### 3.1 Parallel Molecular Dynamics

Our approach combines numerical simulation with symbolic computation. As part of this effort we are developing parallel versions of molecular dynamics (MD) models including AMBER. These parallel systems will be used as components in a larger system, a system that will use MD when needed and will also apply to the unknown structure rules of bond angles, conserved structures, and similarities to known structures. The use of a symbolic or reasoning program to setup and run parallel numerical simulations and then apply or disregard the results to the refinement of the unknown structure is a novel feature of our approach.

That these various programs need to be parallel is evident from the difficulty of applying MD or other nonlinear optimization methods to large molecules on sequential computers and that we wish each of these methods to be used in concert—under the control of the reasoning program—to refine a single structure in as short a time as possible. We have selected the initial numerical components of the combined numerical/symbolic Automated Refinement System (ARS). PROLSQ and AMBER are the nonlinear optimization program and molecular dynamics programs chosen respectively. Parallelization of PROLSQ and AMBER have begun.

### 3.2 Protein Interactions Kinetics

We are also developing high-performance simulations of protein-protein macro-behavior. We have developed a parallel implementation of KSYM a protein-protein interaction kinetics program for the DELTA.

This model predicts the rate of protein interactions and can be used to model the development of polymers and other aggregate protein structures. The parallel implementation uses a number of self-scheduling macro-tasks each of which is a parallel explicit finite-difference model. The parallel implementation is a combination of Fortran, message passing and PCN and runs in excess of four gigaflops. It has been used to model the effects of changing pH on polymerization rates for Bence Jones proteins.

We have also made progress in the development of an initial test strategy for applying automated reasoning techniques to the X-ray refinement problem. Our initial approach is to develop automated capability for the manipulation and composition of Hendrickson's stereochemical restraint functions and the application of these composite functions to control the numerical refinement program PROLSQ.

#### 4 *ab initio* Chemistry

The ANL Theoretical Chemistry Group forms the theoretical half of Argonne's Chemical Reaction Dynamics Program. This research program is primarily concerned with the qualitative and quantitative understanding of chemical reaction dynamics, particularly relating to combustion processes.

To fully exploit machines like the DELTA, parallel algorithms must be developed that allow the user the flexibility to trade memory for time. On sequential or shared-memory computers, often a user will not be concerned with the ability to utilize all of available memory with the same algorithm (i.e. he/she can simply increase the size of the problem to fit the machine without changing the algorithm). To date many parallel chemistry implementations have not had this memory monotonicity property. This is particularly true in cases where data structures have been replicated rather than distributed.

In an attempt to design truly scalable algorithms we have started a new project to investigate the parallel scalability of *ab initio* quantum chemistry methods, including; direct self consistent field (SCF) Hartree-Fock, configuration interaction (CI), and multi-reference configuration interaction (MRCI) methods on large-scale massively parallel processors (MPP). Our main focus is on machines with  $O(10^3)$  to  $O(10^4)$  processors. Performance models are being developed and will be used to evaluate the scalability of each method and each will be extensively tested on the DELTA. Our primary goal is to understand

and to make predictions about which parallel computational chemistry algorithms are appropriate for various MPP systems. In addition we are implementing selected methods using portable parallel programming techniques (PCN, P4 and TCGMSG) to provide a testbed for benchmark computations.

Our initial focus has been on Direct SCF Hartree-Fock methods and will be extended to CI, MRCI and multipole methods. This effort underway since August of 1991 is working in four areas. (1) Developing a comprehensive set of performance models for important quantum chemistry methods and demonstrate which methods have the best parallel scaling properties. In each case we are examining a multitude of alternative data decompositions, each allowing us to vary the granularity of the parallel algorithm. From these models (which are in terms of the chemistry problems input and the machine parameters) we can determine the likelihood of obtaining good parallel program efficiency. (2) In many parallel algorithms we have determined that a number of recurring parallel program structures can be found. These recurring structures are called "templates" or "motifs". Often one can describe the complete computation as a composition of these high-level motifs. When this is the case, building the performance models can be simplified by using models of these pre-defined motifs. (3) Several existing course-grain parallel implementations of these methods have been constructed using a set of tools known as TCGMSG. Part of this effort is to port these implementations to the DELTA. (4) To test the specific predictions made with the performance models and develop a set of benchmark calculations that can run with a variety of methods on the DELTA.

Work has started on a complete performance model of a fine-grain parallel implementation of direct-SCF Hartree-Fock. In this performance model we assume the Fock-matrix is completely distributed across the machine. (i.e. with no replication) All existing parallel SCF methods replicate the Fock-matrix and thus are limited to basis sets of approximately 400 on machines like the DELTA. The distributed Fock-matrix approach should allow up to several thousand basis functions which would significantly improve the range of problems open to direct-SCF methods while maintaining high (85%-97%) parallel efficiency.

A preliminary implementation of a "template" based SCF test program has been completed and run on the DELTA. This code will be used to evaluate the technique of using "templates" for parallel code development.

## 5 Conclusions

Our principle conclusions during the past six months of use of the DELTA are as follows:

**Portability.** Essentially complete portability from the iPSC-860 to the DELTA has been achieved and in many instances no code changes are needed to get adequate performance. The use of high-level language tools such as PCN have allowed applications to be moved unchanged from several platforms to the DELTA.

**Input/Output.** I/O is still the great unsolved problem, both from the systems implementation standpoint and from the programming model standpoint. Our primary concerns are throughput achievable in real applications and reliability. Our ongoing projects are serving as excellent testbeds to refine the requirement for I/O systems.

**Topology.** The transition from a hypercube to the mesh topology has not had significant impact on the development or porting of codes.

**Data volume.** The average size of data sets has increased significantly from our immediate past. We now routinely must deal with single files on the order of a gigabyte and complete data sets for a particular run in the hundreds of Gigabytes. This has strained the DELTA file manipulation systems and our local computer infrastructure.

**Compilers.** Per node performance is still the number one problem in obtaining good overall applications performance. The majority of our codes are not communications bound. Thus improving basic compiler technology is absolutely necessary.

**Visualization.** Visualization has become a critical part of the computing environment and must be designed into the strategy for each application. Effective remote visualization is not yet feasible in most cases. Large datasets also make visualization difficult at local sites without large-scale data handling facilities.

## 6 Acknowledgements

The work reported here is primarily the work of others. In particular I would like to acknowledge I'liang Chern, Ian Foster and Steve Hammond for the Icosahedral Method; John Drake (ORNL), Ian Foster, John Michalakes and Pat Worley (ORNL) for the parallel implementation of SWE and CCM2; Fred Stevens for the protein interaction kinetics program; Ian Foster, Robert Harrison, Rik Littlefield (PNL) and Rusty Lusk and others for various efforts in parallel quantum chemistry codes.

## References

- [1] Bath, L. et al. *Users' Guide to NCAR CCM1*, National Center for Atmospheric Research, Boulder, Colorado. June, 1987.
- [2] Chandy, K., Taylor, S., "An Introduction to Parallel Programming", Jones and Bartlett Publishers, Boston, 1991.
- [3] Chern, I., Foster, I., "Parallel Control Volume Method for Shallow Water Equations" Proceedings of SIAM Parallel Processing Seminar, March 1991.
- [4] Foster, I., Gropp, W., Stevens, R., "The Parallel Scalability of the Spectral Transform Method", To appear Month Weather Review, March 1992.
- [5] Hack, J. (NCAR). CCM2 status report presented during meeting of CHAMMPions working group (ANL, NCAR, ORNL) at NCAR. January, 1991.
- [6] Michalakes, J., "Analysis of Workload and Load Balancing Issues in the NCAR Community Climate Model" (*ANL/MCS TM 144*). Mathematics and Computer Sciences Division, Argonne National Laboratory, Chicago. January, 1991.
- [7] Washington, W., Parkinson, C., "An Introduction to Three-Dimensional Climate Modeling", University Science Books, Mill Valley, California, 1986.