The Archimedes Tool Chain and its Application to Earthquake Ground Motion Modeling

1997 Gordon Bell Award Entry Compiler Parallelization Category

David O'Hallaron, Jonathan Shewchuk†,
Jacobo Bielak, Omar Ghattas, Loukas Kallivokas, Heshing Bao, and Jifeng Xu‡
†Dept. of Computer Science, Carnegie Mellon University
‡Dept. of Civil and Environmental Engineering, Carnegie Mellon University

1 Executive Summary

The entry is an unstructured 3D finite element simulation of the ground motion during an aftershock from the January 17 1994 Northridge earthquake in the San Fernando Valley of Southern California. The application was automatically parallelized and compiled by the Archimedes tool chain into a C/MPI program, which was then compiled by the standard Cray optimizing C compiler and executed on 256 PEs of the Cray T3D. With 13.5 million nodes and 77 million tetrahedral elements, we believe the model is one of the largest unstructured codes produced to date, and it was produced automatically. Communication accounts for only 2% of the total running time, and the speedup is nearly linear. The application and its performance on T3D were first described in Supercomputing '96 [1].

2 Archimedes

Figure 1 shows the structure of the Archimedes tool chain. Each tool runs on Unix workstations. Starting from a sequential description of the solver written in a simple variant of the C language, a description of the boundaries of the region to be modeled, and a database of the material properties in the Los Angeles basin provided by Magistrale and Day at San Diego State (not shown), Archimedes produces a quality 3D mesh with the desired frequency resolution (Triangle/Pyramid), partitions the mesh using a geometric recursive bisection algorithm (Slice), parcels the mesh (Parcel), which involves computing the global-to-local node and element mappings, the local sparse matrix structures, and the local communication schedules that are induced by this partition) and finally producing a file that characterizes the parallel simulation.

The logic of the solver is described in a simple variant of sequential C with whole array syntax and a library of primitives for operations such as matrix vector product, inner products, identifying the corners of elements, and Gaussian quadrature routines for matrix assembly. In all there are over 100 of these primitives, and since users can add their own without recompiling anything, this number is growing daily. For example, a civil engineering graduation student at CMU is augmenting the existing primitives with a set of of primitives for nonlinear soils.

Contact author's email address: ohallaron@cs.cmu.edu.

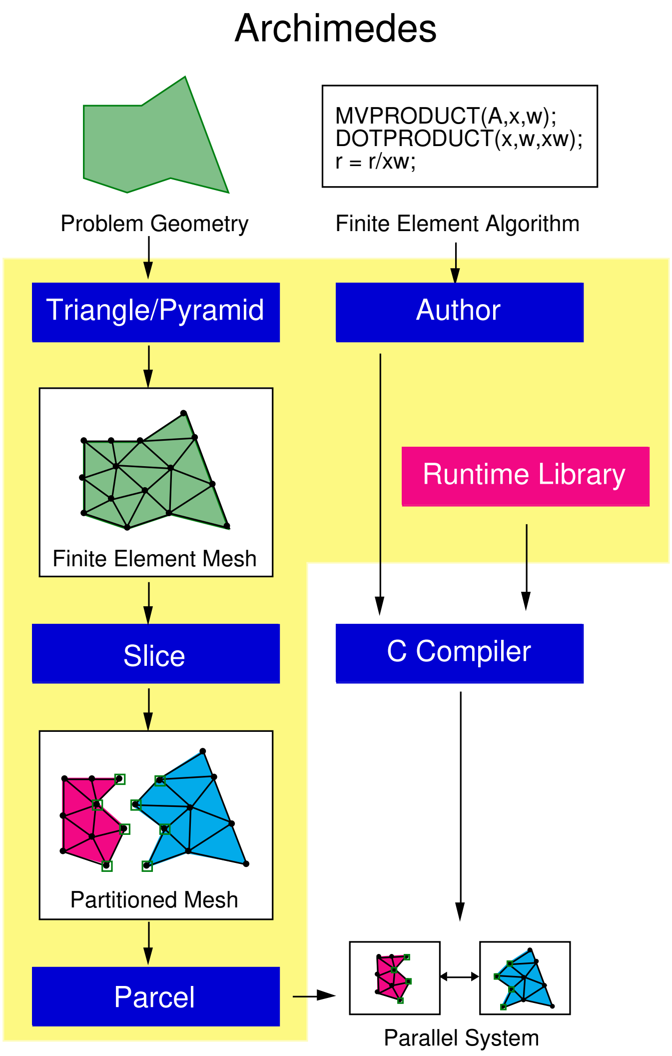

Figure 1: The Archimedes tool chain.

Archimedes compiles the solver (using a compiler called Author) into a standard C program, which is linked with the Archimedes and MPI runtime libraries and compiled using the vendor's C compiler. The parallelism is extracted by the partitioner and the parceler and recognized and supported by the various primitives. The translation by Author from a sequential program to a parallel program is straightforward: user defined primitives such as MVPRODUCT() are translated into C code supplied by the author of the primitive, and the whole array assignment statements are converted into loops over local vectors. The resulting object program is loaded onto each processor, after which it reads the file produced by Parcel, and then runs.

3 Earthquake application

The application simulates the seismic response of the San Fernando Valley in Southern California to an aftershock of the 1994 Northridge earthquake. The unstructured finite element simulation produced by Archimedes consists of 13.5 million nodes and 77 million linear tetrahedral elements, partitioned into 256 load balanced subdomains. The model covers a 50x50x10km region of the Earth, with a 1 Hz frequency resolution and a 200 meter spatial resolution. A smaller version of the San Fernando mesh with 1/10 Hz frequency resolution is shown in Figure 2.

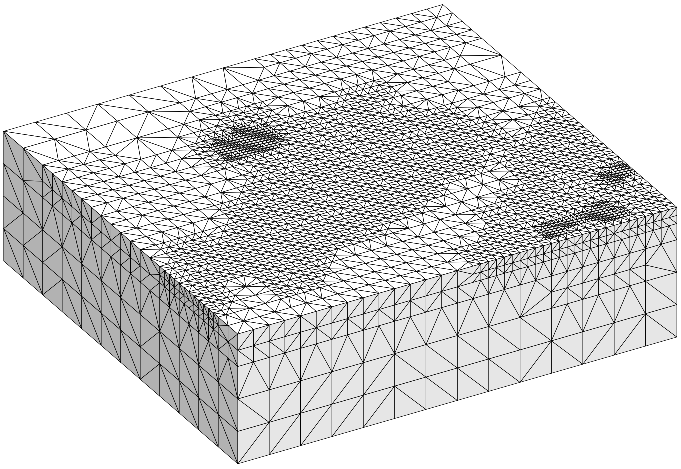

Figure 2: San Fernando earthquake model (1/10 Hz frequency resolution version)

The simulation uses 16,000 time steps to simulate 60 seconds of shaking. The solver is based on an explicit method although in the past we have also used Archimedes to compile solvers based on different methods such as conjugate gradients. The primary computation at each time step is a 40M x 40M (i.e., 13.5M x 3 dof) sparse matrix vector product (SMVP) operation which accounts for all of the interprocessor communication except for that related to file I/O.

4 Performance on Cray T3D

The simulation requires about 16 GBytes of DRAM (roughly 1Kbytes per node) on a Cray T3D. Running time on 256 T3D PEs is about 6.5 hours, with 16,679 secs (71%) due to local computations, 575 secs (2%) due to interprocessor communication during the SMVP operations, and 5995 secs (25%) due to file I/O. Sustained performance during the difficult SMVP operations is about 30 MFLOPS/PE, for a sustained rate of about 7.5 GFLOPS during the SMVP and a sustained overall rate of 3.5 GFLOPS (including all I/O, which includes reading a 1.5 GByte input file and writing a 15 GByte output file).

The application is too big to run on a sequential computer (it would require about 70 days to run on a 150 MHz Alpha 21064 processor with 15 GBytes of DRAM), so direct speedup comparisons are not possible. However results comparing different versions of this application produced by Archimedes show that the SMVP core scales nearly linearly if we hold the number of mesh nodes per PE roughly constant. This is shown graphically in Figure 3. In Figure 3 there are 5 different Quake applications, each produced by Archimedes. The name “sfx” denotes a model of the San Fernando Valley that can resolve a wave with a period of x seconds. For example, sf10 can resolve waves with periods of 10 seconds, or frequencies of up to 1/10 Hz. The “sf1b” application is our entry for the Bell Award. It is much bigger than the “sf1” application because it uses a more refined model of the San Fernando Valley with softer soils, and thus requires more points to achieve the desired 1 Hz frequency resolution.

We ran each application on the number of PEs (restricted to a power of 2) that would result in a workload of about 10,000 mesh nodes per PE. In some cases we are slightly above or below this target, and in teh

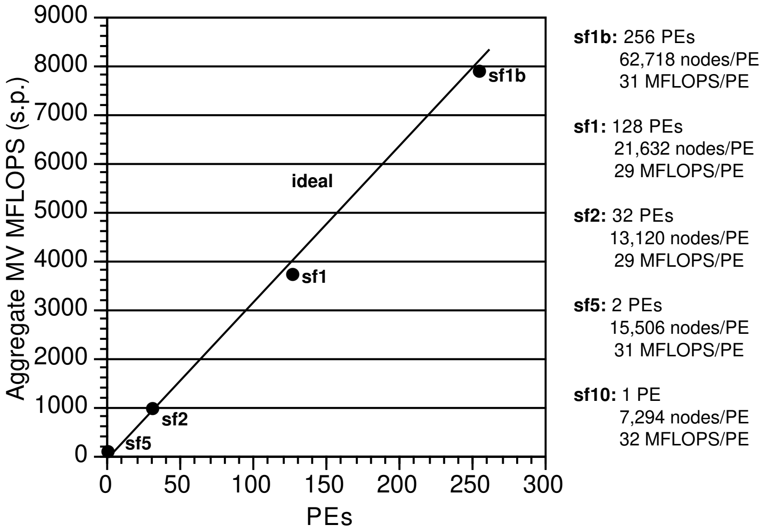

Figure 3: Scaling on the Cray T3D

case of sf1b we far exceed it, but in general the PEs have somewhat comparable workloads. Under these workloads, we can see from Figure 3 that the applications scale almost linearly, achieving a fairly constant sustained computational rate of 30 MFLOPS during the crucial sparse matrix vector product step (which accounts for all of the interprocessor communication in the application except for communication related to I/O). The bottom line is that versions of this application produced by Archimedes scale almost linearly.

References

[1] BAO, H., BIELAK, J., GHATTAS, O., O'HALLARON, D. R., KALLIVOKAS, L. F., SHEWCHUK, J. R., AND XU, J. Earthquake ground motion modeling on parallel computers. In *Proceedings of Supercomputing '96* (Pittsburgh, PA, Nov. 1996).