Three-Level Hybrid Parallelization of Large-Scale Data Visualization for the Earth Simulator

Li Chen [1] Issei Fujishiro [1] [2] Kengo Nakajima [1]

(1) Research Organization for Information Science & Technology, 2-2-54, Nakameguro, Meguro-ku, Tokyo, 153-0061, Japan (e-mail: {chen, nakajima} @tokyo.rist.or.jp; phone: +81-3-3436-5271). (2) Ochanomizu University, Tokyo, 112-8610, Japan (e-mail: fuji@is.ocha.ac.jp; phone: +81-3-5978-5700).

Abstract

High parallel performance is the most distinguished feature of parallel visualization subsystem in GeoFEM. This paper describes some strategies we adopted to improve parallel performance of our subsystem for the Earth simulator. The three-level hybrid parallelization has been applied, including message passing for inter-SMP node communication, loop directives by OpenMP for intra-SMP node parallelization and vectorization for each processing element (PE). Good visualization images and high parallel performance have been obtained on Hitachi SR8000 for large unstructured datasets in GeoFEM, which shows the feasibility and effectiveness of our methods for the Earth Simulator.

Introduction

In 1997, the Science and Technology Agency of Japan began a 5-year project to develop a new supercomputer, the Earth Simulator [1]. The goal is the development of both hardware and software for predicting various Earth phenomena by computational simulations using a supercomputer. The Earth Simulator has shared memory symmetric multiprocessor (SMP) cluster architecture, and consists of 640 SMP nodes, where each SMP node consists of 8 vector processors. GeoFEM is known as a large-scale finite element analysis platform for solid earth simulation [2]. The present study was conducted as part of GeoFEM toward developing a parallel visualization subsystem for solid earth simulation.

As a part of the Earth Simulator software, our visualization subsystem should satisfy the following requirements:

- Powerful visualization functions;
- The ability to cope with large-scale datasets with a high parallel performance;
- Applicable to complicated unstructured grids;
- Suitable for the architecture of the Earth simulator to achieve the best performance on the supercomputer.

Towards these four targets, we have been developing a visualization subsystem (Fujishiro, 2001[3]) in GeoFEM, which has the following four features:

- Many visualization techniques are developed, for scalar, vector and tensor data fields, to reveal the features of datasets from many aspects;
- All the modules have been parallelized and obtained a high parallel performance;
- All of modules are based on unstructured grids, and can be extended to hybrid grids;
- Three-level hybrid parallel programming model is adopted in our modules for getting high speedup performance on the Earth Simulator.

Two aspects are highlighted in these four features. One is concurrent with computation due to the surprisingly huge simulation on the Earth Simulator; and the other is effective parallel performance optimization according to the architecture of the Earth Simulator's hardware. This paper will describe in detail about the two aspects and show the good results of our methods.

Concurrent with Computation

The simulation on the Earth Simulator is expected to be surprisingly huge, possibly up to terabytes scale. It is very difficult and time-consuming to transfer such kind of data to the client machines or save on the disks. Meanwhile, it is impossible to do visualization on the client machines due to the limitation of memory. Therefore, we implemented our subsystem so as to perform the concurrent visualization with computation on the same high-performance parallel computer, which can make full use of computational server's huge memory to complete visualization and avoid the limitations of storage capacity for large-scale data. Once computation modules finish one time-step computation, visualization modules will start immediately. Because visualization is usually much faster than computation process, during each time-step, multiple visualization methods can be done to generate visualization results by different parameters. Figure 1 shows the framework of the parallel visualization subsystem in GeoFEM.

Two kinds of output styles are provided. One is to output simplified graphic primitives to clients. On each client, the users can set viewing, illumination, shading parameter values,

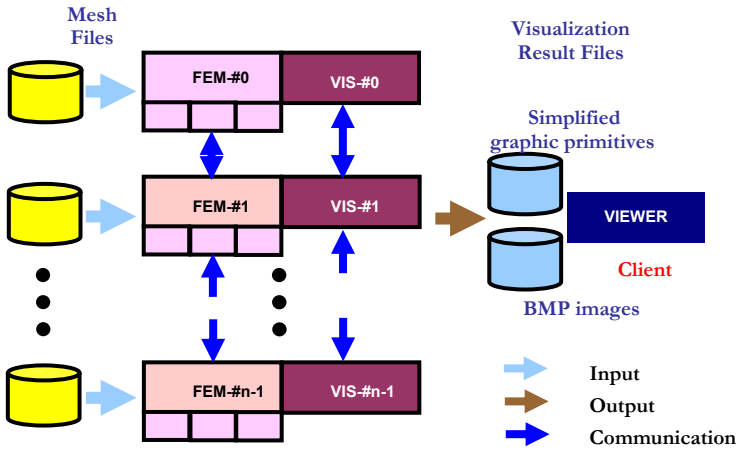

Figure 1: Framework of parallel visualization subsystem in GeoFEM

and so on, and display the graphic primitives by the GPPView software, which is also developed by the GeoFEM group [2]. On the computation server, the users only specify in the batch files, the visualization methods such as volume rendering, streamlines, and related parameters.

The second style is to directly output an image or a sequence of animation images to clients in the case that even the simplified geometric primitives on the Earth Simulator are still too large to transfer. Since the computational part usually need spend more than several days to several months to finish a large simulation with a large number of time-steps, and we cannot save the computational results due to the huge data size, it is better to generate many images by different visualization methods and parameter values to reveal the whole dataset from different aspects in one time, which can avoid the troubles of selecting suitable parameter settings for visualization methods.

Three-level Hybrid Parallelization for Visualization Subsystem

According to the memory hierarchical architecture on the SMP cluster machines, we applied the three-level hybrid parallelization for the visualization subsystem. That means:

- Inter-SMP node: MPI;
- Intra-SMP node: OpenMP for parallelization;
- Individual PE: Compiler directives for vectorization / pseudo vectorization.

Improve vector performance for each PE

Although vectorization can be done automatically by compiler, sometimes it is very difficult to reach a high vector performance if we do not organize the codes well to make it suitable for vector performance. For example, the sufficient long loop is very critical for the vector performance. We make this kind of loops as possible as we could by the following ways (Minami, 2001[4]):

- Combine some short loops into one long loop by reordering;
- Exchange the innermost and outer loop to make the innermost loop longer;
- Minimize load/store latency as possible as we could.

Accelerate parallelization in each SMP node by multi-coloring

Loop directives by OpenMP are used for the parallelization of intra-SMP nodes. In order to achieve efficient parallel performance by OpenMP, no dependency and data race are very critical. However, in our visualization subsystem, data race exists in some parts of codes. For example, in the parallel volume rendering module, before ray-casting grids, the gradient on each vertex need be computed first. The gradient is computed by the shape function of each mesh element. The pseudo-algorithm for getting gradients is as follows:

```
#pragma omp parallel
{
  for(i=0;i<num_element;i++) {
    compute Jacobian matrix of shape function;
    for(j=0; j<NUM_VERTEX_ELEMi ;j++) {
      for(k=0; k< NUM_VERTEX_ELEMi;k++)
        accumulate gradient value of vertex j contributed by vertex k;
  } } }
```

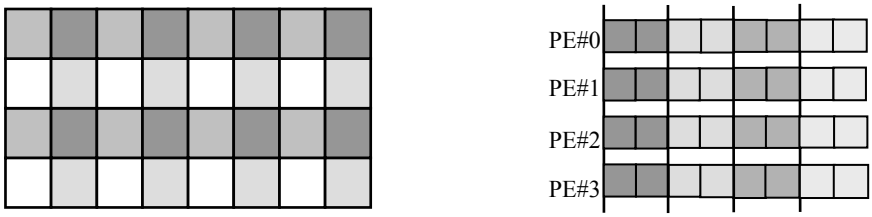

Figure 2: Multi-coloring for removing data race.

Obviously, there is a data race on accesses to the shared variable *gradient* because one vertex is often shared by some elements. Although it can be avoided by adding mutual exclusion synchronization or writing to a private variable in each thread and then copying to the share variable, they need either more extra time cost or more extra memory cost. In our parallelization, a multi-coloring strategy is adopted (Nakajima, 2001[5]) which can get rid of data race very easily. In this method, as shown in Figure 2, elements are colored by different colors so that each element is marked by the color different from all the colors of its adjacent elements. After coloring, the elements assigned by same color have no common vertex, so they can be parallelized without any data race by OpenMP. Meanwhile, the elements with different colors should be computed in a serial order.

Although it takes some extra time for multi-coloring, it can improve parallel performance much for large time-step dataset visualization. Note that the multi-coloring process only need be executed once at the first time-step.

Accelerate parallelization among SMP nodes by dynamic load repartition

MPI software is used for communication among SMP nodes in our system. Because the visualization modules need be concurrent with the computational part, it is very important to keep the consistent parallel style and data structure with computation. GeoFEM system adopts an object-space parallelism. The entire data domain is partitioned into distributed local datasets, and each partition is assigned to one SMP node. In order to reduce the communication among the SMP nodes, overlapped elements exist at each domain boundary.

However, this data distribution may not be suitable for most visualization cases. The rendered voxels often accumulate in small portions of the field, and their number changes greatly during visualization, which leads to very unbalance load distribution. Therefore, dynamic load repartition is very important to improve parallel efficiency.

For example, for parallel volume rendering, a scattered decomposition was adopted by some previous methods (Ma, 1997[6]), which can get a very good load balance. However, this method lost data coherence, which would slow down parallel performance much on the SMP cluster machine. Moreover, excessive storage and communication of intermediate results were needed in this method. It is not suitable for large-scale datasets. In order to keep the load balance, dynamic load repartition is done in our method. During the preprocessing of visualization, the data value of each vertex is fast scanned and the total number of vertices falling into the rendered range is counted. Then according to the number of rendered vertices on each SMP node, move a subvolume from one SMP node with a larger number of rendered vertices to another SMP node with a smaller one.

Experimental Results on Hitachi SR8000

We have applied our subsystem to visualize some unstructured large datasets generated by GeoFEM analysis modules on SR8000 (8 PEs, 8GFLOPS peak performance and 8GB memory for each node. 128 nodes (1024 PEs), 1.0TFLOPS peak performance and 1.0TB memory for total system).

Test 1: Parallel volume rendering a Pin Grid Array (PGA) dataset (Data courtesy of H. Okuda and S. Ezure in The University of Tokyo).

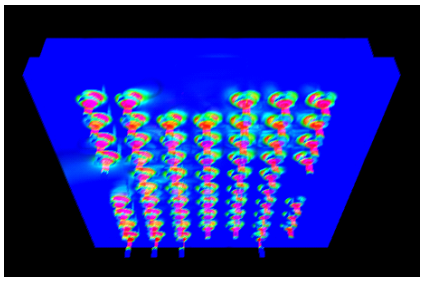

Figure 3: Parallel volume rendering image to show the equivalent scalar value of stress for a PGA dataset with 7,869,771 vertices and 7,649,024 grid elements.

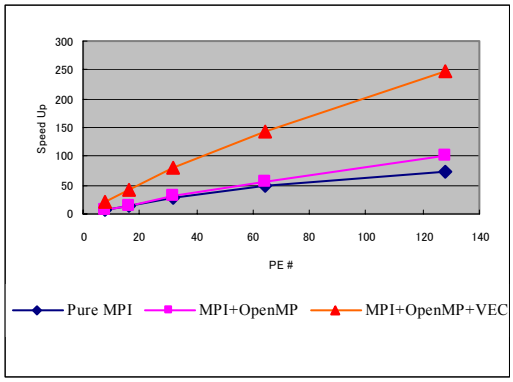

Figure 4: Comparison of speedup performance for the PGA dataset with 7,649,024 grid elements.

As shown in Figure 3, the parallel volume rendering module was applied to reveal the distribution of the equivalent scalar value of stress in a 3D unstructured PGA dataset, which has 7,869,771 vertices and 7,649,024 grid elements. We tested it on 1 node (8PEs), 2 nodes (16 PEs), 4 nodes (32 PEs), 8 nodes (64 PEs), 16 nodes (128 PEs) by MPI+OpenMP hybrid parallel module and three-level hybrid parallel module, and compared with the results of pure MPI on the same PE number with the same data size. The speedup performance is shown as Figure 4. To generate Figure 3, pure MPI took about 98.33, 48.79, 27.53, 15.39 and 10.40 seconds on 8 PEs, 16 PEs, 32 PEs, 64 PEs and 128 PEs, respectively, whereas MPI+OpenMP hybrid parallel took about 97.96, 48.73, 25.36, 13.34, and 7.48 seconds on 1 node, 2 nodes, 4 nodes, 8 nodes and 16 nodes, respectively. Furthermore, MPI+ OpenMP+ Vectorization hybrid parallel took 35.65, 17.44, 9.28, 5.29 and 3.07 seconds, respectively. We can see hybrid parallelization especially three-level hybrid parallelization can improve parallel performance much on the SR8000 machine.

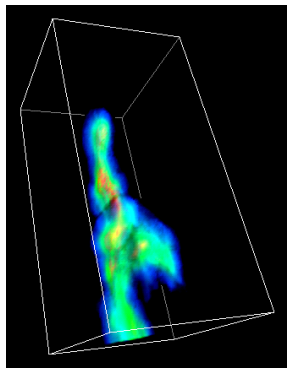

Figure 5: Parallel volume rendering image to show the concentration distribution for an underground water simulation dataset with about 10 million vertices and 100 timesteps.

Test 2: Parallel volume rendering an underground water dataset (Data courtesy of Kengo Nakajima in GeoFEM).

This 3D unstructured dataset simulated groundwater flow and convection/diffusion transportation through heterogeneous porous media for deep geological disposal of high-level radioactive waste. By using this dataset, we mainly tested the effectiveness of the dynamic load repartition strategy we adopted. In this dataset, many elements have no value in some time-steps. On SR8000 with 8 SMP nodes, for about 10 million vertices and 100 time-steps, without dynamic load repartition it took about 2.35 seconds for one time-step on average. After dynamic load reparation, it just took about 1.56 seconds on average for one time-step. This demonstrates the effectiveness of our dynamic load repartition method.

Conclusions and Future Work

This paper introduced the three-level hybrid parallelization of the large-scale unstructured data visualization for the Earth Simulator. Some techniques on improving the speedup performance of hybrid parallelization were described. Good results had been obtained on Hitachi SR8000 by applying the above techniques to some unstructured large datasets in GeoFEM, which shows the feasibility and effectiveness of our subsystem.

The hardware of the Earth Simulator has been successfully finished by NEC company in March this year. Up to now, it is the fastest supercomputer in the world with a peak performance of 35.6 Tflops. Our software will be installed and run on it this June. Because SR8000 has the similar architecture, our system is expected to achieve high parallel performance on the Earth Simulator as well. The future work will focus on the tests on the Earth Simulator and further improvements of the parallel performance.

Acknowledgments

This study is a part of the Solid Earth Platform for Large-Scale Computation project funded by the Japanese Ministry of Education, Culture, Sports, Science and Technology through Special Promoting Funds of Science & Technology.

References

[1] Earth Simulator Research and Development Center Web Site: http: // www.es.jamstec. go.jp/.

[2] GeoFEM Web Site: http://geofem.tokyo.rist.or.jp/.

[3] Fujishiro, I., et al., 2001, *Parallel visualization of gigabyte datasets in GeoFEM,* Journal of Concurrency and Computation: Practice and Experience (in print).

[4] Minami, K. and Okuda, H., 2001, *Performance optimization of GeoFEM on various computer architecture,* GeoFEM Report 2001-006, RIST/Tokyo.

[5] Nakajima, K. and Okuda, H., 2001, *Parallel iterative solvers for unstructured grids using Directive/MPI hybrid programming model for GeoFEM platform on SMP cluster architectures,* Journal of Concurrency and Computation: Practice and Experience (in print).

[6] Ma, K.-L., et al., 1997, *A scalable parallel cell projection volume rendering algorithm for 3D unstructured data*, Proc. of 1997 Symposium on Parallel Rendering, 95-104.