J. Plasma Physics (2006), *vol.* 72, *part* 6, *pp.* 841–844. © 2006 Cambridge University Press
doi:10.1017/S0022377806004995 Printed in the United Kingdom

MovieMaker: a parallel movie-making software for large-scale simulations

H. UEHARA, S. KAWAHARA, N. OHNO,
M. FURUICHI, F. ARAKI and A. KAGEYAMA
Earth Simulator Center, Japan Agency for Marine-Earth Science and Technology,
3173-25 Showa-machi, Kanazawa-ku, Yokohama 236-0001, Japan

(Received 15 August 2005 and accepted 24 January 2006)

Abstract. We have developed a parallel rendering software for scientific visualization of large-scale, three-dimensional, time development simulations. The goal of this software, MovieMaker, is to generate a movie, or a series of visualization images from one terabyte-scale data set within one night (or less than 12 h). The isocontouring, volume rendering, and streamlines are implemented. MovieMaker is a parallel program for the shared memory architecture with dynamic load balancing and overlapped disk I/O.

1. Introduction

The output data size of computer simulations is rapidly growing in accordance with the development of high-performance computers. On the Earth Simulator, the output data often reaches more than 1 GB per one time step for a single variable. Since a typical simulation may produce the output data for several variables for hundreds or sometimes thousands of time steps, the whole data set of one simulation job can reach more than 1 TB in total.

Three-dimensional, time development data should be visualized by a series of images, or a movie. However, making a scientific visualization movie from 1 TB of data in a reasonable time is beyond the ability of commercially available software today.

Therefore, we have decided to develop such a software by ourselves. The goal of the system, 'MovieMaker', is to enable us to produce a movie from 1 TB of simulation data within 12 h, or one night [1]. When the whole data set consists of 1024 time steps with 1 GB each, MovieMaker should be able to apply various visualization processes for each time step within about 40 s.

MovieMaker can handle two different kinds of visualization method, the polygon-based method and the volume-based method. This is useful when one should visualize a mixed set of scalar and vector fields; for instance, the volume rendering for a pressure field and the streamlines for a flow vector field. This can be contrasted with most other visualization tools developed so far [2–4]. Another feature of MovieMaker is its simple disk I/O. Yu et al. [2] pointed out the disk I/O cost in the parallel rendering system. They solved this problem by using the parallel disk I/O. On the other hand, we use the standard non-parallel disk I/O.

https://doi.org/10.1017/S0022377806004995 Published online by Cambridge University Press

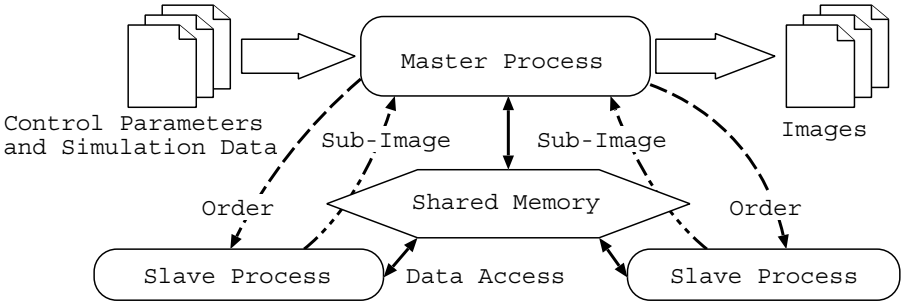


Figure 1. MovieMaker. Master/slave model for the parallel processing on the shared memory architecture is adopted.

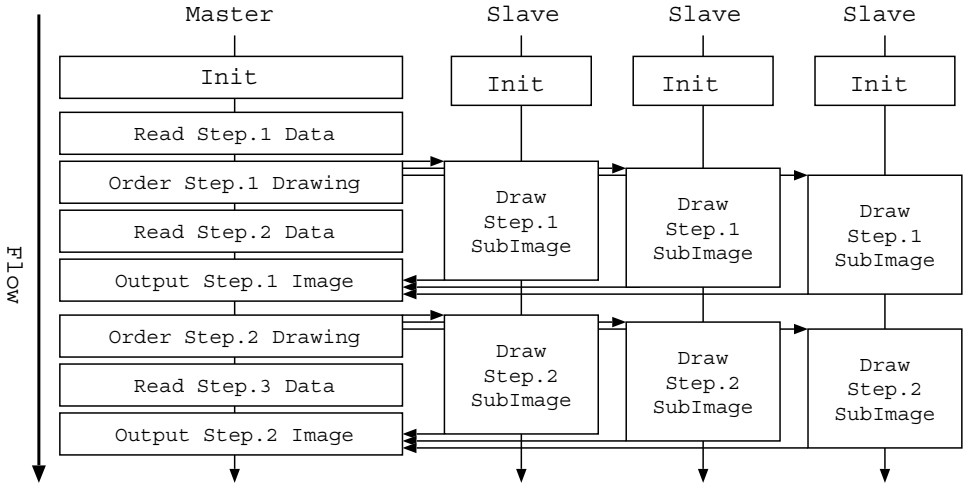


Figure 2. Flowchart of MovieMaker. Overlapping of the data read step and rendering step is achieved.

2. Development of MovieMaker

We have designed MovieMaker as a master/slave parallel rendering program for the shared-memory architecture. As shown in Fig. 1, the master process and slave processes share the simulation data stored in the shared memory area. The master process performs the following multi-tasks: (i) read a configuration file; (ii) read the simulation data into the shared-memory area; and (iii) control the slave processes keeping a good load balance. Slave processes perform rendering tasks following commands sent from the master process and then return partial images back to the master via shared memory. Interprocess communications (IPC) are performed with the Message Passing Interface (MPI). We have achieved dynamic load balancing in MovieMaker by an active monitoring and dynamic control of the slave processes.

In addition to the dynamic load balancing, we have also implemented task overlapping for efficient parallel rendering. Figure 2 shows a flowchart of MovieMaker. While the slave processes are working on rendering tasks for one specific time step of the simulation data, the master process reads the data of the next time step. Since it takes about one second to read a gigabyte scale data set for each time step

https://doi.org/10.1017/S0022377806004995 Published online by Cambridge University Press

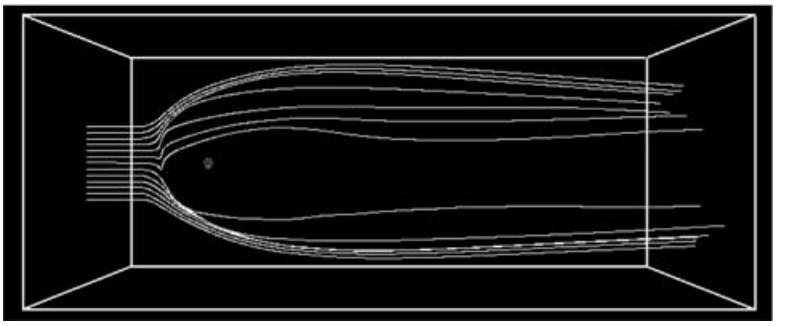


Figure 3. A sample image generated by MovieMaker using the volume rendering.



Figure 4. Same as Fig. 3 viewed from a different angle.

from the hard disk drive, the total read time for the whole movie data may reach about an hour. We could hide this read time in the rendering time of MovieMaker by applying the task overlapping method.

MovieMaker is implemented in C++ with OpenGL. OpenGL is used for the off-screen rendering. The present version of MovieMaker has three visualization methods: (1) volume-rendering; (2) isocontouring; and (3) streamlines. The volume rendering and the isocontouring are implemented based on the basic algorithms of ray casting and marching cubes, respectively. These three visualization methods can be used in juxtaposition. Fine-tuning of each visualization method is possible by controlling the visualization parameters specified by the configuration file.

MovieMaker, in the present version, adopts input data defined in the Cartesian geometry. The grid system should be rectilinear, i.e. uniform and/or non-uniform in each direction of x, y, and z. The maximum grid number is constrained only by the shared memory size.

In our computational environment with SGI Onyx 3800 (12 CPUs, 24 GB memory, 11 MPI processes), MovieMaker has achieved the target performance, i.e. processing 1 GB within 40 s. A performance test was applied to a geodynamo simulation data set with the grid size $640 \times 640 \times 640$ of 4-byte floating point. The average processing time to generate a single image by volume rendering, isocontouring, and streamlines are, 40, 33 and 10 s, respectively.

Figures 3 and 4 show sample images of the volume rendering visualization applied to the temperature distribution of a magnetohydrodynamic simulation of the Earth's magnetosphere. High-temperature regions are emphasized. The grid-size of the data is $502 \times 202 \times 202$, with the 4-byte floating-point data. The data size for one step is about 80 MB.

https://doi.org/10.1017/S0022377806004995 Published online by Cambridge University Press

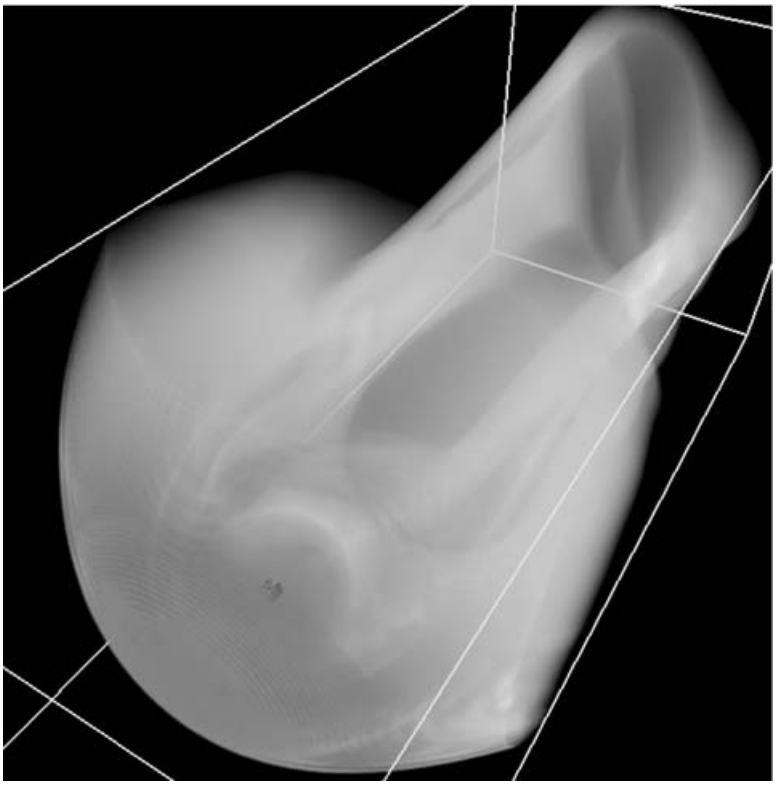


Figure 5. A sample image generated by MovieMaker using streamlines.

Figure 5 is another sample image generated by MovieMaker. This image shows the velocity of the Solar wind in the simulation of the magnetosphere, using the streamlines. The bow shock is clearly seen.

3. Conclusion

We have developed MovieMaker, a parallel movie-making software for scientific visualization of large-scale simulations. The main goal of MovieMaker is to make a movie from 1 TB of data within one night (or less than 12 h), and this goal has been successfully achieved for the basic visualization methods including the volume rendering, isocontouring, and streamlines.

References

[1] Uehara, H. et al. 2005 MovieMaker: high performance movie-making system for high-resolution data. *Proceedings of the Joint Meeting for Earth and Planetary Science*, J031-005 (CD-ROM) Japan Science Union: Tokyo, Japan.

[2] Yu, H. et al. 2004 I/O strategies for parallel rendering of large time-varying volume data. *Proceedings of the Parallel Graphics and Visualization Conference*, pp. 31–40 Eurographics.

[3] Ma, K. L. et al. 2003 Visualizing very large-scale earthquake simulations. *Proceedings of Supercomputing 2003* ACM.

[4] Livnat, Y. and Tricoche, X. 2004 Interactive point based isosurface extraction. *Proceedings of the IEEE Visualization Conference*, pp. 457–464 IEEE Computer Society.

https://doi.org/10.1017/S0022377806004995 Published online by Cambridge University Press