Regular Paper

An Evaluation of the Communication Cost of Parallel Processing in Real-Time Simulations Using an Image-Composition Device

Masato Ogata,[†] Kagenori Kajihara,[†] Takaaki Kikukawa[†] and Takafumi Terada[†]

We have been developing a volumetric computing graphics cluster system in the form of a PC cluster with its rendering and calculation performance enhanced by graphics boards and dedicated devices. The goal of the system is to perform real-time simulation for a practical surgical simulator. A space-partition scheme for parallel processing inevitably requires data communications for both simulation and visualization. We studied the effects of both types of communications through experiments. On the basis of the results, we discuss a performance model and propose a performance metric for time-restricted processing. To provide an example, we evaluated our VGCluster system by using the proposed metric. The metric shows the effect of sustaining scalability by using a dedicated image-composition device.

1. Introduction

A wide variety of phenomena, such as motion of fluids, gases, and climatic variations, lend themselves naturally to volumetric representation. Large-scale and precise simulations using volumetric models have been conducted in order to study complex phenomena in the natural world. In such simulations, the use of interactive simultaneous simulation and visualization with parameters varying "on the fly" is becoming important for the intuitive understanding of these phenomena. Such interactivity can put humans into a closed loop of modeling and simulation, and they can interact to design and manipulate the model with feedback from visualized results.

The advent of personal computers and network technologies has made PC clusters more powerful and less expensive. Accordingly, we have posed the rather naive question of whether a real-time simulation system with a visualization capability can be implemented by using traditional PC clusters without any dedicated hardware for communications. This has been our primary motivation for this research.

In this paper, we study the effects of communications not only on parallel visualization but also on parallel simulations in order to develop a practical surgical simulator that requests processing of a 512^3 grid. To study the feasibility of the final goal, we conducted experiments using a deformable model that involves simultaneous processing with simulation and visualization. Here, we evaluate the results and discuss a performance model and an evaluation metric for time-restricted processing.

The remainder of this paper is organized as follows. In Section 2, we describe previous work. In Section 3, we discuss parallel processing of simultaneous simulation and visualization. In Section 4, we present the configuration of the VGCluster cluster systems. In Section 5, we describe our experiments. In Section 6, we discuss the performance model and a proposed metric. Finally, we present conclusions and ideas for future work in Section 7.

2. Previous Work

In many simulations, supercomputers are commonly used to compute numerical models. The numerical results are then forwarded to a high-end graphics system, such as SGI Onyx, to be visualized off-line[1)~3)]. The Earth Simulator[3)] is famous for its use of this scheme. Although traditional supercomputers are extremely powerful for calculations, they are not suitable for visualizing the simulated results while steering the simulations by changing the parameters on the fly. When this is attempted, the transfer of an enormous amount of computed results creates a bottleneck, causing difficulty for interactive simulation and visualization. Interactivity, or real-time operation, is the most important requirement for intuitive understanding of simulated phenomena.

GRAPE is a well-known special-purpose computer for astronomical simulations that uses dedicated hardware for fast calculation[4)]. The basic idea is to carry out the calculation

[†] Mitsubishi Precision Co., Ltd.

of universal gravitation $\frac{m_1 \cdot m_2}{r^2}$ using dedicated hardware. GRAPE can perform fast calculations for some specific applications in a cost-effective manner.

In recent years, there has been increasing use of PC clusters in applications once dominated by supercomputers. Many papers relating to this scheme have been published. Although the final goal is simultaneous simulation and interactive visualization, only the the latter has been implemented for large-scale volumetric data, because of its relative simplicity. To realize true interactive visualization, dedicated hardware devices for composition have been proposed 5)~7). In particular, Ogata et al.7) evaluate the performance degradation caused by communications during visualization in a space-partitioning scheme. There are also reports describing the necessity of dedicated hardware for image composition in order to realize interactive visualization systems 5),6),8).

Some experimental systems attempting to realize simultaneous simulation and visualization have been presented 9)~11). The GPU Cluster uses graphics processing units(GPUs) to accelerate numerical calculations. It achieved simultaneous simulation and visualization for a 480×400×80 Lattice Boltzman model using 30 PCs at 0.32 second/step 9), and is approaching the ability to actually solve practical problems.

This might suggest a simple question: “If we increase the number of PCs, can we then get enough computational power for simultaneous simulation and visualization at a video rate?” Few previous papers have evaluated the communications bottleneck caused by rendering and simulation with a high degree of parallelism. For this paper, we carried out experiments involving bottlenecks caused by communications in both parallel simulation and visualization.

3. Communications for Parallel Processing

For real-time simulation with visualization, it is necessary and natural to use a parallel processing scheme. Large-scale PC cluster systems are coming into common use. It is necessary to communicate over a network because the data to be processed is separated into parts and stored in each PC. In such simulations, the communication cost due to network traffic is one of the important issues to be considered for system performance.

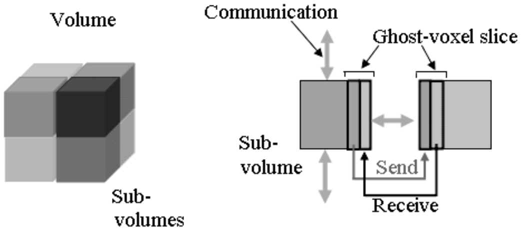


Fig. 1 Space partition for parallel simulation. The ghost-voxel slices are exchanged periodically between adjacent subvolumes.

3.1 Communications for Parallel Simulation

It is also natural to adopt a space-partition parallel scheme for large-scale real-time simulations, because the memory size of each PC is limited. In such a scheme, a large simulation space is separated into small subspaces. We refer to a simulation space as a volume and to subspaces as subvolumes.

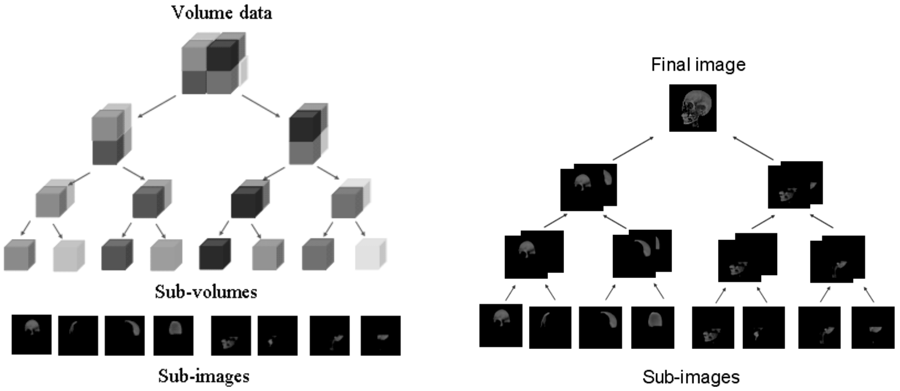
A space-partition scheme inevitably requires data communications. **Figure 1** depicts ghost-voxel slices, or overlapping slices, which hold copies of adjacent subvolumes. The ghost-voxel slices have to be exchanged periodically between adjacent subvolumes, i.e. between PCs, by means of communications. The number of overlapping slices for a ghost-voxel slice is usually one, but it depends on the simulation.

3.2 Communications for Parallel Visualization

The sort-last parallel-rendering approach 12),13) is the most suitable for a space-partition scheme. **Figure 2** presents an overview of the image composition procedure for sort-last parallel rendering. Each PC has a subvolume and generates a 2D image. To obtain a final image, the composition procedure is repeatedly applied to pairs of images while traversing a binary tree from bottom to top. It is therefore necessary to transfer large amounts of image data between PCs in order to obtain the composite final image. The amount of data to be transferred is proportional to the number of parallel PCs. The transfer of larger images increases the network load. This greatly decreases the processing speed for parallel visualization, and consequently interactive simulation and visualization become difficult.

4. Platform for Experiment

Figure 3 presents the VGCluster cluster system used for evaluating the communication cost



(a) A volume is divided into subvolumes, and each PC then generates a 2D image for this subvolume.

(b) The composition procedure is repeatedly applied to pairs of images while traversing a binary tree from bottom to top.

Fig. 2 Parallel rendering in the space-partition scheme.

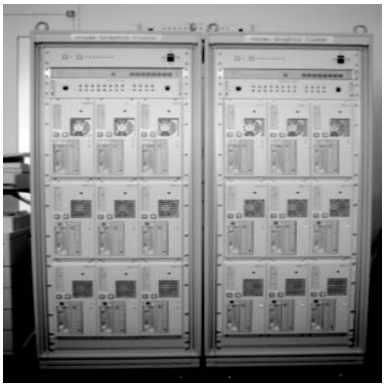


Fig. 3 VGCluster cluster.

Table 1 Specification of the VGCluster cluster.

No.	Item	Specifications
1	The number of PCs	16 nodes, 1 host
2	CPU	Xeon 2 GHz × 2
3	GPU	Geforce 4
4	Memory	2024 MB
5	Network	Myrinet: 2 Gbits/s
6	OS	Linux 7.2, Score 5.0.1
7	Image composition device	PCI 32/33 MHz

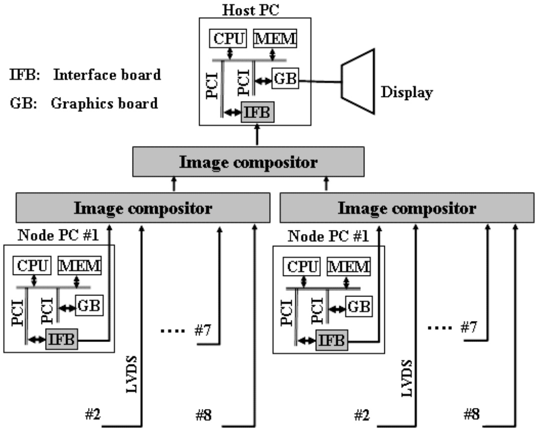


Fig. 4 Configuration of a VGCluster cluster.

for both parallel simulation and visualization.

4.1 VGCluster Cluster Configuration

The system is composed of two VGClusters with a fast network, Myrinet, and one extra dedicated image-composition device. The VG-Cluster is constructed from eight-node PCs with a graphics board on each PC, a single host PC, and a dedicated image-composition device. The specifications and configuration of the VG-Cluster cluster system are shown in **Table 1** and **Fig. 4**.

4.2 Parallel Rendering with an Image Composition Device

As the degree of parallelism increases, interprocessor communication quickly becomes a bottleneck. Video-rate visualization implemented in software also becomes more difficult with increased parallelism[7]. In order to solve these problems, we developed an image composition device that accepts eight inputs as a unit and performs pipelining composition[6),7)]. **Figure 5** illustrates the dedicated image-composition device, an important element for reducing the network traffic in a VGCluster cluster system. **Figure 6** shows a block diagram of the image composition device. As illustrated in Figs. 2 and 4, each node PC is responsible for generating a 2D partial image for the corresponding sub-volume. This partial image (a stream of pixels with RGBA val-

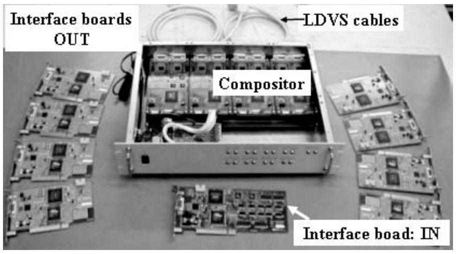


Fig. 5 Image composition device.

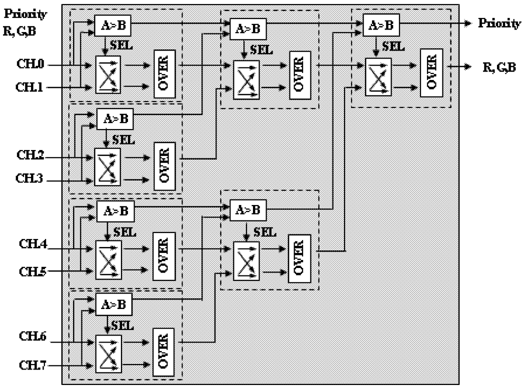


Fig. 6 Block diagram of the image composition device.

ues), along with priority information indicating closeness to the eye, is transferred to the input port of the composition hardware via an interface board inserted into the PCI bus.

Subimages transferred to the image composition device from each PC are subject to composition. In the block diagram in Fig. 6, the composition of two images is repeatedly applied with an OVER block using priority information. It is a direct implementation of a binary tree, like the composition in Fig. 2 (b).

An image composed by means of a composition device is loaded into the frame memory of the graphic board in the host PC via an interface board and is displayed as the final image. The composition device enables simultaneous processing of simulation and visualization without increasing the network load for parallel processing.

Although the interface boards of the composition device use a low-speed bus and do not use cutting-edge technology, this does not negatively affect the true purpose of the experiment.

5. Experiments

In this section, we explain in detail an experiment using the VGCluster cluster system to evaluate the communication cost of simulation with visualization. There are two experimental conditions, with and without the image composition device. For each condition, the number of node PCs is varied from 2 through 16 to verify the scalability. When the image-composition device is not used (without-C/D hereafter), i.e., when the conventional method is used, communication loads associated with the simulation and visualization are borne by Myrinet. When the image-composition device is used (with-C/D hereafter), the load associated with the visualization is borne by the image-composition device. In our experiment, we measure the update rate for both cases.

5.1 Simulation Model: Deformation

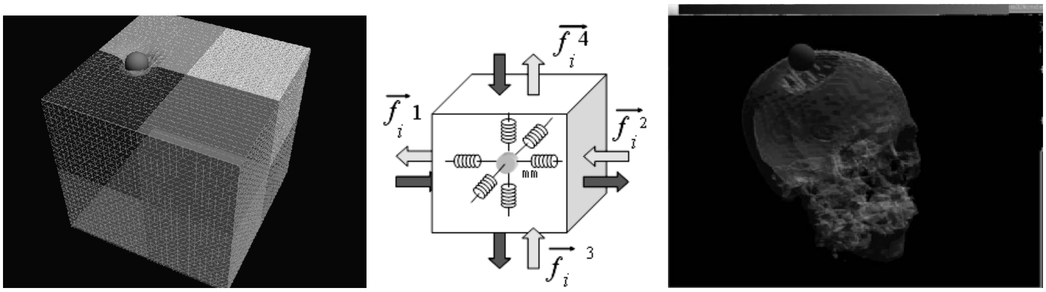
We used a human head deformation program to evaluate the communication cost, as shown in **Fig. 7**. This deformation requires communication of ghost-voxel slices for simulation and of images for image composition. The mechanical relationship among the cells is described by Eq. (1):

$$\begin{aligned} \boldsymbol{f}_i = & -\gamma_i\dot{\boldsymbol{x}}_i - m_i\boldsymbol{g} \\ & -\sum_j k_{ij}(|\boldsymbol{x}_{ij}| - L_{ij})\frac{\boldsymbol{x}_{ij}}{|\boldsymbol{x}_{ij}|}\ , \end{aligned} \qquad (1)$$

where
$\boldsymbol{f}_i$ is the external force for grid i,
$\boldsymbol{x}_i$ is the position vector of grid i,
m_i is the mass of grid i,
γ_i is the viscosity at grid i,
k_{ij} is the Fuch's constant between grids i and j,
$\boldsymbol{x}_{ij}$ is the vector from grid i to grid j,
L_{ij} is the initial distance between grids i and j.

The first, second, and third terms correspond to the viscosity, gravity, and elastic distortion for the spring model. The following is the pseudocode of the program:

```
/*******************************************
/*   Cellular automaton for space-partition scheme.
/*      Processing on node.
/*******************************************
For each node $n$  {
   ∀ Adjacent nodes{
         Send and Receive ghost-voxel-slice;
   }  /*— End of ∀ adjacent nodes —*/
   ∀ Effective grid $i$  {
         Calculate $f_i$ with equation (1);
   }  /*— End of ∀ grid. —*/
   Carry out isosurface generation using the
   Marching Cubes method; Draw image;
  if ( Without-C/D ) Send image to host;
  else       Send image to interface board;
}  /*— End of for each node $n$ —*/
```



(a) Example of the simulation space partition scheme. The space is divided into 8 subspaces with 2 divisions along each axis.

(b) Communications of a cellular automaton. Ghost-voxel slices are exchanged between adjacent subvolumes at each boundary surface of subvolumes.

(c) Simulated image: deformation of a head

Fig. 7 Spring model implemented using cellular automaton.

Table 2 Results of experiment.

Number	Average update rate: (Hz)	
of nodes	With C/D	Without C/D
1	4.04	4.04
2	7.35	7.12
4	12.94	9.12
6	17.35	9.14
8	20.62	8.29
10	23.13	7.10
12	25.46	6.47
14	27.41	5.88
16	28.75	5.26

```
/*********************************************
/*   Cellular automaton for space-partition scheme.
/*      Processing on host.
/*********************************************
{
  if ( Without-C/D ){
    ∀ Node n {  / * Node * /
          Receive 2D image data from node;
    }  /*—— End of ∀ node ——*/
    Compose received 2D images from nodes;
  }   else {
     Receive composite image from interface board ;
  }  /*—— End of if ——*/
   Draw 2D image;
}
```

The number of grids to be simulated is 32^3. Output images are 512×512 pixels with full color. Evaluation is performed by examining the update rate. The result under each condition was obtained from an average of 10 measurements.

5.2 Results of Experiment

Table 2 presents the results of the experiment. **Figure 8** shows the corresponding plots. We set the video rate to 30 Hz as a target value for real-time processing. For every number of nodes, "with-C/D > without-C/D" holds true, demonstrating that the scalability has been sustained by the image-composition device.

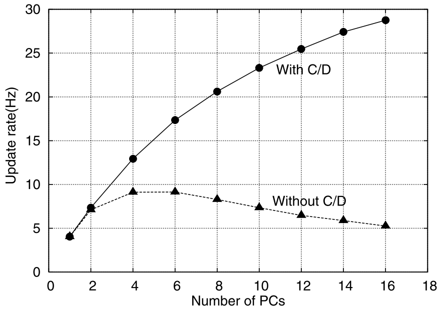


Fig. 8 Comparison of the performance of the composition device and a traditional network.

Comparison of the highest update rate under each condition yielded a value of 9.14 Hz for 6 nodes in the without-C/D condition, and 28.75 Hz for 16 nodes in the with-C/D condition, resulting in a difference of 1.9 times for 6 nodes, or of 5.47 times for 16 nodes. The experiment proves that combining both communications causes rapid deterioration in the scalability.

6. Discussion of Performance Model

6.1 Performance Model

Figure 9 depicts the well-known Amdahl's law regarding the speed-up ratio with respect to the number of parallel processors. Any serialization due to data dependencies or I/O bottlenecks limits concurrency and thus the speed-up ratio. The portion of a computation extended due to inherent serialization is termed the "Amdahl fraction" of a computation.

Amdahl's law states that the speed-up ra-

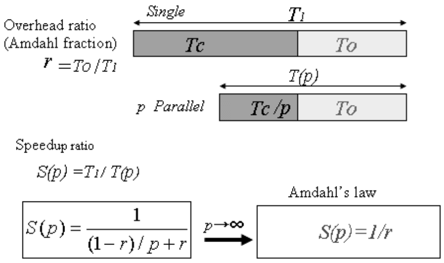


Fig. 9 Theoretical limit imposed by Amdahl's law.

tio is inherently limited to some upper bound by a characteristic of the process indicated by the Amdahl fraction, i.e., T_o/T_1. The speed-up does not increase above this upper bound with an increase in parallelism. However, actual experimental results indicate that the speed-up ratio converges to zero, as it does without a dedicated image composition device, as indicated in Fig. 8.

The main reason for this discrepancy between Amdahl's law and our experiment is the assumption regarding the overhead with respect to the number of parallel PCs. Amdahl's model assumes a simple communications model in which the overhead is constant with respect to the number of parallel PCs. This assumption is not relevant in our case. Although the model is appropriate for use as a first-order approximation of the system, it is not accurate enough for more precise estimation.

Our proposed performance model is expected by Eq. (2):

$$T(p)=T_1 r + T_1(1-r)/p + c(v^2)(p^{1/3}-1)+b(m^2)(p-1), \quad (2)$$

where

$T(p)$ is the parallel processing time with p PCs,
p is the number of PCs; $p \geq 2$,
T_1 is the processing time on the unit processor,
r is the overhead ratio T_o/T_1 or Amdahl fraction,
$c(v^2)$ is the communication cost for boundary voxels in the simulation. A volume consists of v^3 voxels,
v is the number of voxels for each axis,
$b(m^2)$ is the communication cost for subimage in rendering; A screen consists of m^2 pixels.

The first and second terms in Eq. (2) are the same as in Amdahl's model. We introduce the communication load for simulation and image composition in Eq. (2). The third term, $c(v^2)(p^{1/3}-1)$, expresses the communication cost for the exchange of ghost-voxel slices when the volume is divided into subvolumes. Similarly, the fourth term, $b(m^2)(p-1)$, expresses the communication cost for image composition. Each of the p nodes transfers images to the host to compose for visualization. The details are presented in Appendix A.1.

Table 3 Estimation of parameters in the performance model of Eq. (2).

No.	Estimated parameters	T_1 = 0.2473 s: Measured value $v^2=32^2$; $m^2=512^2$	
		With C/D	Without C/D
1	r	0.078	0.075
2	$c(v^2)$	0.00082	0.00082
3	$b(m^2)$	0.0	0.0106

6.2 Performance Metric

The parallel processing efficiency factor $E(p)$ is commonly used to evaluate parallel system performance. The processing efficiency is defined in Eq. (3):

$$E(p) = S(p)/p \ . \quad (3)$$

The speed-up ratio $S(p)$ is defined in Eq. (4):

$$S(p) = T_1/T(p) \ . \quad (4)$$

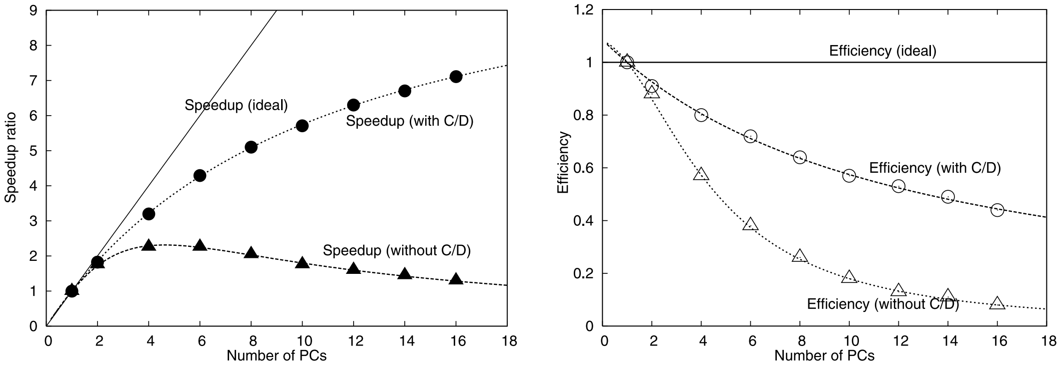
We estimated the parameters in Eq. (2) on the basis of the experimental results from Table 2, using the least-squares method. **Table 3** presents the estimated parameters for the performance model. In the estimation, we assumed only that the parameter $c(v^2)$ is identical in both cases. The mathematically estimated values of the Amdahl fraction r in both cases are almost identical, as shown in Table 3. This might be taken as strong evidence that the model is reasonable.

Figure 10 illustrates the speed-up ratio and efficiency factor of the model with respect to the number of parallel PCs p with and without the use of dedicated hardware for image composition. Both metrics are plotted by using Eq. (2) and the parameters in Table 3. The plotted curves and the points of the experimental results are almost identical.

6.3 Proposed Metric for Time-restricted Operation

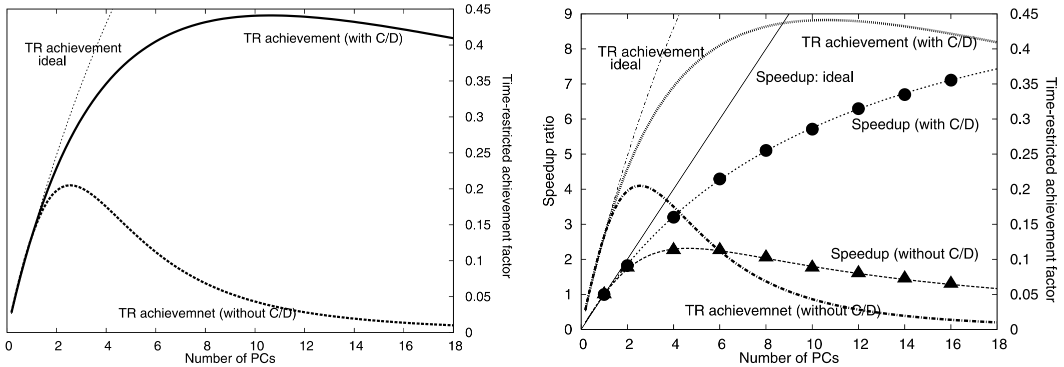
The parallel processing efficiency $E(p)$ and speed-up ratio $S(p)$ are commonly used to evaluate parallel system performance.

Unfortunately, these metrics are not adequate for evaluating our time-restricted system. To obtain a high value of $E(p)$, it is common to increase the scale of the problem, which decreases the ratio of the communication load relative to processing. This is a commonly accepted argument for improvements in



(a) Speedup ratio (b) Efficiency factor

Fig. 10 Comparing performance between the composition device and a traditional network using a traditional metric. The curves are plotted in accordance with the performance model. The points are plotted using experimental results.



(a) Time-restricted achievement factor (TR). (b) TR with Speedup. For "speed-up", use left vertical axis.

Fig. 11 Parallel speed-up ratio and time-restricted achievement factor. The curves are plotted on the basis of the performance model. The points are plotted using experimental results.

efficiency. Unfortunately, scaling-up the problem size results in longer times for T_1 and $T(p)$. This results in a meaningless improvement in efficiency, since the processing time $T(p)$ becomes farther from the target processing time. This efficiency improvement seems irrelevant to time-restricted operations. Real-time systems must complete their processing within 33.3 ms or 16.7 ms. We need a time-restricted evaluation metric.

The following is the proposed metric. We call this metric a time-restricted achievement factor $A(p)$. The factor is defined as follows:

$$\begin{aligned} A(p) &= (T_{target}/T(p))E(p) \\ &= \mu S(p)E(p) \ , \end{aligned} \tag{5}$$

where T_{target} is the application-dependent target time and μ is a constant that satisfies $T_{target} = \mu T_1$. For example, in our application T_{target} is 33.3 ms or 16.7 ms. For other applications, T_{target} may be one day, such as in a car collision simulation.

Figure 11 shows evaluations using the proposed metric and a traditional metric. In the figure, we illustrate the performance for a deformation simulation that was intended to perform simultaneous simulation and visualization over two cases. In the first case, deformation was implemented with no dedicated hardware for image composition; in the second case, dedicated hardware was used for image composition.

The maximum value of the metric is 0.45 at around 10 nodes for the with-C/D case. In addition, this high ratio is maintained over a relatively wide range of 6 to 16 nodes. The number of processors corresponding to the maximum value of the time-restricted achievement factor indicates the critical number of processors. We can easily identify the critical number of nodes as 10. In the without-C/D case, the maximum ratio was 0.18 at around 3 nodes. This high ratio was maintained only over a very narrow width. The critical number of nodes is 3. Considering this, the dedicated image-composition device was very effective for sustaining the scalability, though a mechanism to reduce communications for parallel simulation was necessary for video-rate operation.

Although in the figure the efficiency ratio $E(p)$ indicates the tendency described above for the without-C/D case, it does not indicate this tendency clearly for the with-C/D case. In the with-C/D case, the critical number of nodes is not clear.

7. Conclusion

The higher the computational power, the larger the number of PCs becomes. The increasing number of PCs affects the communication cost inversely and creates a bottleneck for both parallel simulation and parallel visualization.

We conducted experiments to study the effects of both kinds of communications. On the basis of the results, we have discussed a performance model and proposed a new performance metric. This metric can be used to evaluate time-restricted operations such as in real-time systems. On the basis of the observed results, we demonstrated that communication channels separate from simulations are effective for realizing simultaneous simulation and visualization in an interactive manner. Although it is not a final solution, the image-composition device was effective in sustaining scalability for simultaneous simulation and visualization.

Acknowledgments We thank the late Dr. Shigeru Muraki of AIST for his contribution to this work. This work has been financially supported in part by the National Institute of Information and Communications Technology in order to develop a practical surgical simulator.

References

1) Matsuo, Y. and Tsuchiya, M.: Early Experience with Aerospace CFD at JAXA on the Fujitsu PRIMEPOWER HPC2500, *SuperComputing 2004*, ACM SIGARCH and IEEE Computer Society (2004).

2) Nakano, A., Kalia, R.K. and Vashishta, P.: Scalable Atomistic Simulation Algorithms for Materials Research.

3) Oliker, L., Carter, A.C.J. and Shalf, J.: Scientific Computations on Modern Parallel Vector Systems, *SuperComputing 2004*, ACM SIGARCH and IEEE Computer Society (2004).

4) Makino, J., Kokubo, E., Fukushige, T. and Daisaka, H.: A 29.5 Tflops Simulation of Planetesimals in Uranus-Neptune Region on GRAPE-6, *SuperComputing 2002*, ACM SIGARCH and IEEE Computer Society (2002).

5) Heirich, A. and Moll, L.: Scalable Distributed Visualization Using Off-the-Shelf Components, *IEEE Symposium on Parallel Visualization and Graphics*, IEEE Computer Society, pp.55–118 (1999).

6) Muraki, S., Ogata, M., Ma, K.-L., Koshizuka, K., Kajihara, K., Liu, X., Nagao, Y. and Shimokawa, K.: Net-Generation Visual Supercomputing Using PC Clusters with Volume Graphics Hardware Devices, *SuperComputing 2001*, ACM SIGARCH and IEEE Computer Society (2001).

7) Ogata, M., Muraki, S., Ma, K.-L. and Liu, X.: The Design and Evaluation of a Pipelined Image Composition Device for Massively Parallel Volume Rendering, *Volume Graphics 2003*, Eurographics Organization, Eurographics Organization, pp.61–68 (2003).

8) Lombeyda, S., Moll, L., Shand, M., Breen, D. and Heirich, A.: Scalable Interactive Volume Rendering Using Off-the-Shelf Components, *IEEE Symposium on Parallel and Large-Data Visualization and Graphics*, IEEE Computer Society, pp.115–121 (2001).

9) Fan, Z., Qiu, F., Kaufman, A. and Yoakum-Stover, S.: GPU Cluster for High Performance Computing, *SuperComputing 2004*, ACM SIGARCH and IEEE Computer Society (2004).

10) Kruger, J. and Westerman, R.: Linear Algebra Operators for GPU Implementation of Numerical Algorithms, *SIGGRAPH2003*, ACM SIGGRAPH (2003).

11) Muraki, S., Lum, B.E., Ma, K.-L., Ogata, M. and Liu, X.: A PC Cluster System for Simultaneous Interactive Volumetirc Modeling and Visualization, *IEEE Symposium on Parallel and Large-Data Visualization and Graphics*, IEEE Computer Society, pp.95–102 (2003).

12) Molnar, S., Cox, M., Ellsworth, D. and Fuchs, H.: A Sorting Classification of Parallel Rendering, *IEEE CG & Application*, Vol.14, No.4, pp.23–32 (1994).

13) Ma, K.-L., Schussman, G., Wilson, B., Ko,

K., Qiang, J. and Ryne, R.: Advanced Visualization Technology for Terascale Particle Accelerator Simulations, *SuperComputing 2002*, ACM SIGARCH and IEEE Computer Society (2002).

Appendix

A.1 Performance Model

We will derive the performance model for parallel processing in a space-partition scheme, i.e. Eq. (2).

If we assume that the volume is separated by planes, then Eq. (6) is a performance model for the space-partition scheme.

$$\begin{aligned} T(p) &= T_1 r + T_1(1-r)/p \\ &\quad + c^*(v^2)N_p + b(m^2)(p-1) \quad , \end{aligned} \tag{6}$$

where

$T(p)$ is the parallel processing time with p PCs,
p is the number of PCs; $p \geq 2$,
T_1 is the processing time on a unit processor,
r is the overhead ratio T_o/T_1 or the Amdahl fraction,
$c^*(v^2)$ is the communication cost for boundary voxels in simulation; a volume consists of v^3 voxels,
N_p is a number of separation planes for a volume,
v is the number of voxels for each axis,
$b(m^2)$ is the communication cost for a subimage; a screen consists of m^2 pixels.

The first and second terms are similar to Amdahl's model, the third term is the communication cost for simulation, and the fourth term is the communication cost for image composition. The third term $c^*(v^2)N_p$ is a direct consequence of the fact that the communications are performed between subvolumes bisected by planes.

We assume that the volume is separated by a plane, so that the following relation between the number of parallel processors and the number of separations for each axis will be satisfied:

$$p = (N_x + 1)(N_y + 1)(N_z + 1) \tag{7}$$

$$N_p = N_x + N_y + N_z \tag{8}$$

Here, N_x is the number of separation planes along the x axis, N_y is the number along the y axis, and N_z is the number along the z axis.

We further provide the following inequality between the arithmetical mean and the geometrical mean.

$$\begin{aligned} &\frac{(N_x+1)+(N_y+1)+(N_z+1)}{3} \\ &\quad \geq ((N_x+1)(N_y+1)(N_z+1))^{1/3} \end{aligned} \tag{9}$$

From Eq. (7), Inequality (9) becomes the following inequality.

$$\frac{(N_x+1)+(N_y+1)+(N_z+1)}{3} \geq p^{1/3} \tag{10}$$

From the above relation, the performance model becomes the following inequality:

$$\begin{aligned} T(p) &\geq T_1 r + T_1(1-r)/p \\ &\quad + c(v^2)(p^{1/3} - 1) + b(m^2)(p-1) \ , \end{aligned} \tag{11}$$

where $3c^*(v^2)$ is to substituted for the coefficient $c(v^2)$.

If the number of separations is identical for each axis, then the equality is satisfied in Inequality (10) and Inequality (11) becomes the following equation:

$$\begin{aligned} T(p) &= T_1 r + T_1(1-r)/p \\ &\quad + c(v^2)(p^{1/3} - 1) + b(m^2)(p-1) \ . \end{aligned} \tag{12}$$

$T(p)$ represents the greatest lower bound of the processing time with p processors.

(Received October 3, 2005)
(Accepted January 31, 2006)

Masato Ogata is a manager of the research and development department at Mitsubishi Precision Co., Ltd. in Japan. He graduated from Oita National College of Technology, received his Dr. E. in information science and technology from Yokohama National University. He is a certified professional engineer in information engineering. His research background has involved computer graphics and related topics. In 1982, he developed nation's first real-time visual system for flight simulators. In 2001, he received remarkable invention award by Ministry of Education, Culture, Sports, Science and Technology. Current his research interests are real-time simulations and large-scale visualization. He is a member of IEICE, ITE and IEEE Computer Society respectively.

Kagenori Kajihara received the B.E., and the M.E. in aeronautics from the University of Tokyo, in 1962, and 1964 respectively and the Dr. E. in information science and technology from Tokyo Institute of Technology in 2004. He is a certified professional engineer in information engineering. He joined Mitsubishi Electric Co. in 1967 and then he moved to Mitsubishi Precision Co., Ltd. where he studied and developed flight simulators and visual systems. His primary research interests are real-time simulations and large-scale visualization. In 1982, he developed nation's first real-time visual system for flight simulators. In 1987, he received remarkable invention award by Ministry of Education, Culture, Sports, Science and Technology.

Takaaki Kikukawa is a project engineer of the research and development department at Mitsubishi Precision Co., Ltd. in Japan. He received B.S. degrees in physics from Chuo University, in 1990. He has been working for developing a Surgical Training Simulator. His research interests include parallel computer architecture for real-time simulations and large-scale visualization. He is a member of The Institute of Electronics, Information and Communication Engineers.

Takafumi Terada is a project engineer of the research and development department at Mitsubishi Precision Co., Ltd. in Japan. He received B.S. degrees in electrical engineering and M.S. degrees in astrophysics from Kumamoto University in 1985 and 1987 respectively. He has been working for developing a Surgical Training Simulator and a Rehabilitation System. His research interests include Virtual Reality in particular Haptic Device. He is a member of The Society of Instrument and Control Engineers, The Virtual Reality Society of Japan, Japanese Society for Medical Virtual Reality, and Japanese Society of Medical Imaging Technology.