15. Advanced Visualization Research Team

15.1. Team members

Kenji Ono	(Team Leader)
Chongke Bi	(Postdoctoral Researcher)
Tomohiro Kawanabe	(Visiting Researcher)
Masahiro Fujita	(Visiting Researcher)
Kentaro Oku	(Visiting Researcher)
Hiroki Mizuno	(Visiting Researcher)
Yukiko Hayakawa	(Assistant)

15.2. Research Activities

The following is the objectives of our research in FY2012.

1) Installation of a cluster system for visualization and data processing research.
2) Development of a visualization software system for large-scale dataset.
3) Development of elemental techniques for visualization of large-scale dataset.

Researchers or engineers have their own scenarios for visualization and post processing. In addition, the well-known knowledge in each research field is utilized to comprehend phenomena and/or to lead scientific discovery. Many applications will help in the post processing process and it is efficient to apply the already existing software. However, all applications are not always operated on the K computer due to porting issue and so on. Our team has therefore introduced a widely used PC cluster system that has Intel CPU to offer the opportunity of running such applications. This cluster system will be directly connected to the global file system of the K computer in the next year. Moreover, the common visualization tools like VisIt, ParaView, and VMD will be installed on the visualization cluster.

On the other hand, application users strongly demand a visualization system that is able to operate on the K computer to avoid the extra copy of the dataset or to enhance the scalability of their applications. For this purpose, a visualization system named LSV has been now developing. The LSV system has been developed in the grand challenge program in RIKEN during 2007-2012, and its development is carried on into our team. The LSV system is originally designed for the commodity platforms such as Linux machine or clusters. This year, we investigated how to port the LSV system onto the K computer and found that the GLES/GLSL rendering API can be used instead of the conventional OpenGL API. We confirmed that a written test program worked well and could generate images on the K computer directly. The demonstration was conducted for molecular dynamics results.

Another point of view, it is of great significance to apply data compression method to the large-scale dataset generated from simulations. We have been investigating an efficient compression method using proper orthogonal decomposition (POD) that enables us to reduce the data size small with high quality. Due to the limitation of memory size and computational cost, a parallel POD method has been proposed. A preliminarily test cases showed us that the proposed method have good potential but there is some room to improve the parallel efficiency.

Through the whole processes of computer simulation, it is important not only to manage the distributed files for high performance computation but also usability. To efficiently handle and manage the distributed file of Cartesian data structure, a lightweight file I/O library CIOlib was developed. This library provides functions; parallel distributed file I/O, management of distributed files, restart from different number of files N on M processes, restart with refinement, and restart with staging process on the K computer. This library will be applied for the simulation in the field of fluid, astrophysics and so on.

In SC12 conference, one of the AICS team gave a demonstration using Intel Phi processors, which showed a real-time climate CFD simulation including visualization. Our team took in charge of the visualization part. In this demonstration, the visualization was carried out by pre-computing and streaming technique, and showed reasonably fair frame rate.

So far, research activities in FY 2012 were summarized in above mentioned with three categories.

15.3. Research Results and Achievements

15.3.1. Installation of Visualization Cluster

Table 1 shows the specification of installed visualization cluster system and Fig. 1 depicts the structure of the system. Since this system is connected to the global file system of the K computer in original plan, no extra disk storage is supplied for this system now. There are two main purposes to operate this cluster. One is to run applications that are not able to operate on the K computer for some technical reasons, and the other is to offer an interactive visualization environment for users because the interactive environment for visualization is not offered yet. This cluster system will be operated for general post processing platform not only for visualization.

Table 1: Specification of visualization cluster.

Number of Node	Head node x 1 (Fujitsu PRIMERGY RX350 S7) Visualization node x 32 (Fujitsu PRIMERGY RX350 S7)
CPU	Intel Xeon E5-2670 x 2 (2.6GHz/8core)
Memory	64GB(8GBx8) >> 2TB(aggregate)
HDD	SAS 600GBx2(RAID1)
Network	InfiniBand QDRx1, LANx2
Operating System	Red Hat Enterprise Linux 6.1

In the next year, several extensions will be planned, such as the connection between this cluster and the global file system of the K computer, additional storage, and installation of GPUs.

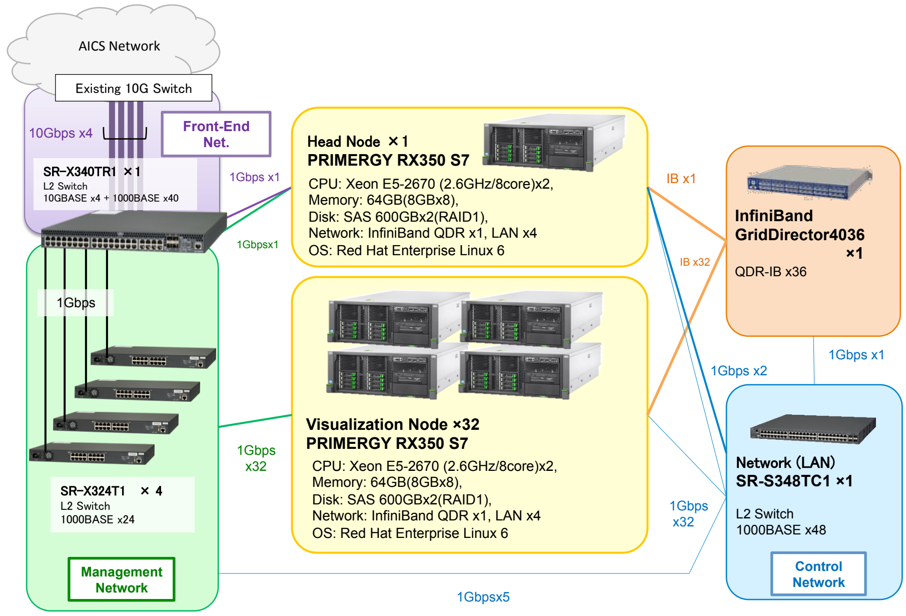


Figure 1: Configuration of an installed visualization cluster system.

15.3.2 Large-Scale Data Visualization System (LSV)

Fig.2 shows an internal logical design of the LSV system. The LSV system has a module structure and is designed so that the system can offer several useful visualization scenarios illustrated in Fig. 3. The LSV had been developed to operate on common PC clusters for these a couple of years. We had been discussed to make the LSV operate on the K computer. Usually, OpenGL API is used to render images but we found that OpenGL API is not supported on the K computer. We have decided to use OpenGL shading language (GLSL) for rendering API and OpenGL ES API (GLES) instead of the conventional OpenGL. A parallel ray-tracing program, which is a prototype of LSV's rendering kernel, was written by using GLES/GLSL and was compiled on the K computer. The parallel ray-tracer generated images as shown in Fig. 4 for the results of a molecular dynamics simulation and an extracted polygon data. Both images are rendered with the global illumination effect for better visibility. Fig. 5 also shows high quality and high resolution rendering image for the example polygon model. The global illumination and the ambient occlusion technique are used to increase visual quality.

15.3.3 Data Compression using Snapshot POD

Visualizing and analyzing large-scale dataset is an important task for scientific research in various fields. However, the visualization process is time-consuming, which is quite inconvenient for researchers and engineers to analyze the time-varying dataset. In order to resolve these problems, we proposed an approach to generate a small-scale dataset from the original large-scale one.

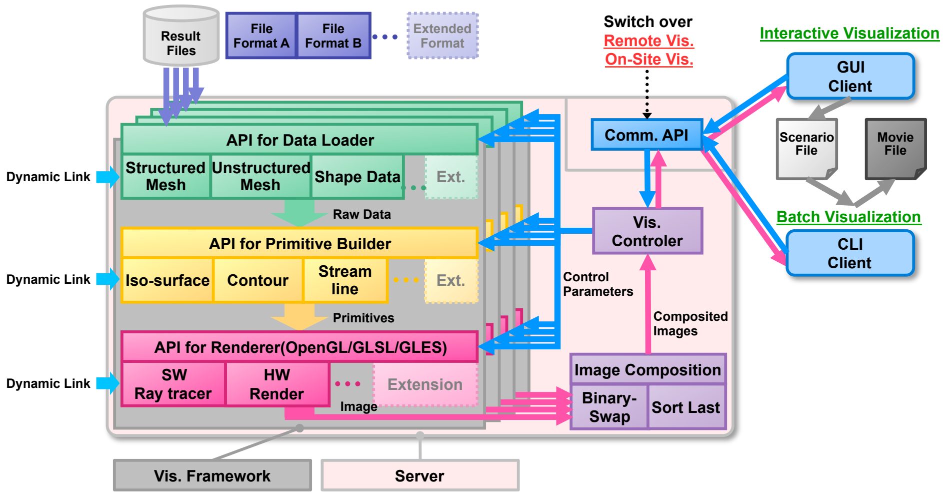


Figure 2: Logical design of a LSV system. Switching over the connection of modules, the LSV provides various operating scenarios for users.

Mode	Client	Vis. process
On-site	User PC	User PC
	Remote server	Server
Remote	User PC	Server

- **User interaction**
  - **Interactive** : **Images are generated by user's operation**
  - **Batch** : **Image/movie is obtained by predetermined scenario**
- **Timing**
  - **Real-time** : **InSitu**
  - **Post processing** : **File based method**
- **Renderer**
  - **Hardware** : **GPU rendering, high speed**
  - **Software** : **CPU rendering, high quality**

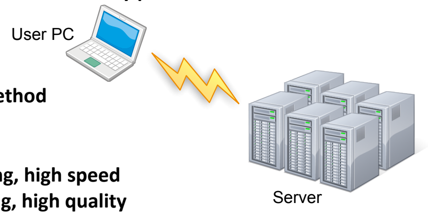
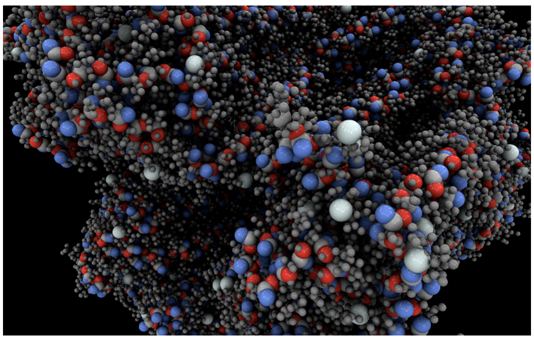
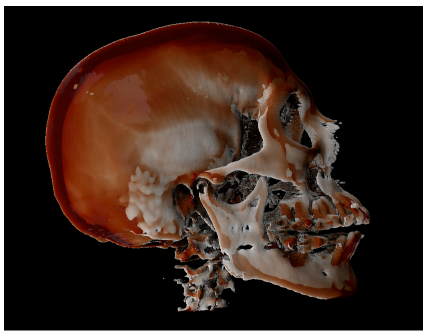


Figure 3: User's operating scenario of visualization.



(a) PDB data



(b) Sub-surface scattering effect is applied for extracted iso-surface. Data is taken from Stanford CT Head data.

Figure 4: Parallel rendering results using GLSL API on the K computer.

(a) Full image. (b) 2,014 resolution. (c) 4,096 resolution.

Figure 5: A high-resolution image shows detail precisely. Figure (b) and (c) are closeup view of a yellow box in figure (a).

Figure 6 shows the detail of the proposed approach. The original dataset in the layer 0 was divided into 3 small groups firstly. POD algorithm is used to compress the datasets of the three small groups in parallel. However, if the compression process is stopped here, the compression ratio is 4/11, which is still high. Therefore the POD-basements in the layer 1 are compressed again as shown in the second line of Fig. 6. This process will be repeated until the POD-basements cannot be compressed any more. Restoring the compressed datasets efficiently is as important as compression process. In Fig. 6, the red arrows show the process of restoring the dataset in the 6th time step t_{0_6}. In the restoring process, it is unnecessary to calculate the POD-basements in the middle layers. The original datasets can be linearly restored using the POD-basements in the deepest layer and the related coefficients.

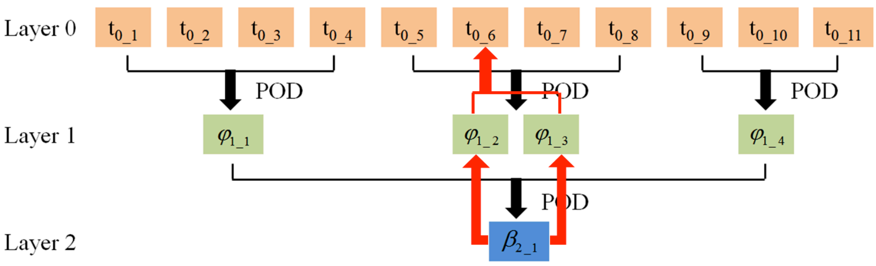
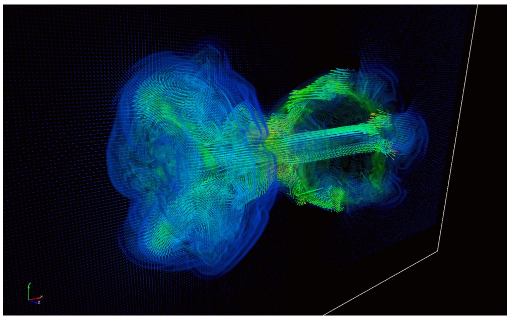
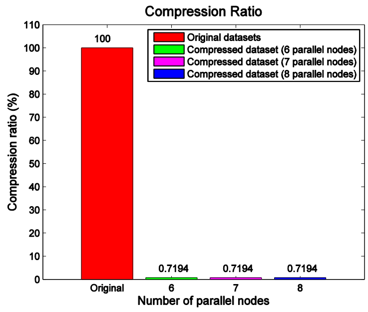


Figure 6: The flow chart of the proposed approach.

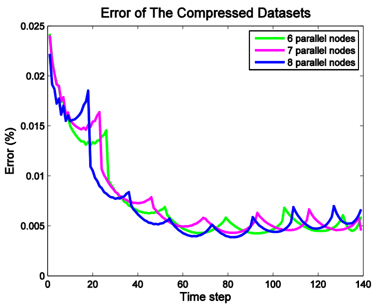
Figure 7 shows the result to compress a large-scale time-varying dataset (500×200×125×139 [x×y×z×timesteps]) obtained from a flow simulation in the air jet mixer of a machinery, which is rendered by the magnitude of velocity (Fig. 7(a)). The colors of red, green, magenta, and blue represent original dataset, the compression results with 6 nodes, 7 nodes, and 8 nodes in parallel, respectively. Fig. 7(b) is the compression ratio calculated. The result shows that our approach can successfully compress such kind of large-scale dataset in a quite low compression ratio. Meanwhile, the precision of the compressed dataset can also be preserved in such low compression ratio, as shown in Fig. 7(c). Here, the number of parabola shape is the same with the numbers of nodes. These parabola shapes can be used to prove the correctness of our algorithm. This is because the mean velocity for the neighbor time steps is nearly equal to that of the middle time step in a small size group. Furthermore, the computational cost is also investigated in temporal space, as shown in Fig. 7(d). It becomes smaller as the numbers of parallel nodes are increased. This also described one of the merits of our parallel algorithm.



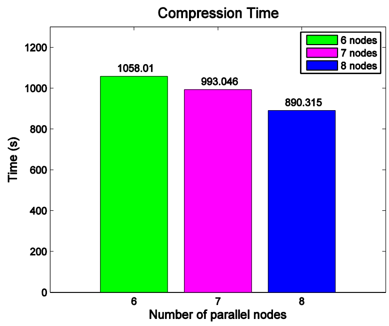
(a) Flow field of sample data. Color means the magnitude of velocity.



(b) Compression ratio for different parallel cases.



(c) Error distribution.



(d) Comparison of compression time for different parallel cases.

Figure 7: Result of compressed a large-scale dataset of a flow simulation in the air jet mixer.

15.4. Schedule and Future Plan

In addition to the three topics listed in section 2, several elemental technologies, e.g., knowledge extraction by agent-based method, in-situ visualization, tiled-display technology and parallel particle tracking, will be investigated in the next year.

15.5. Publication, Presentation and Deliverables

(1) Journal Papers

1. Bi, C., Sakurai, D., Takahashi, S. and Ono, K.: Interactive Control of Mesh Topology in

Quadrilateral Mesh Generation Based on 2D Tensor Fields, Lecture Notes in Computer Science, Advances in Visual Computing, Vol. 7432, pp. 726-735, (2012).
2. Ono, K., Advanced Visualization Technology on Large-Scale Numerical Simulation, Journal of the Visualization Society of Japan, Vol, 32, No.125, pp.1-6, (2012).

(2) Conference Papers

1. Ono, K., Kawanabe, T. and Hatada, T.: HPC/PF - High Performance Computing Platform: An Environment that Accelerates Large-Scale Simulations, Vecpar 2012, (2012).
2. Mao, X., Watanabe, D. and Ono, K.: Gaze-Directed Flow Visualization, Proceedings of the Conference on Computational Engineering and Science, Vol.17, (2012).
3. Suzuki, S. and Ono, K.: An Efficient Data Structure of Building-Cube Method for Large-Scale Computation, Proceedings of the Conference on Computational Engineering and Science, Vol.17, (2012).
4. Ono, K., Mizuno, H., Mukai, Y. and Oku, K.: Development of technology for very-large-scale voxel generation and its interface for simulators, Proceedings of the Conference on Computational Engineering and Science, Vol.17, (2012).
5. Suzuki, S., Ono, K., Ogawa, T. and Onishi, J.: A Software Framework of the Building-Cube Method for a Large-scale Computation, MASCOT and ISGG 2012, (2012).
6. Onishi, J. and Ono, K.: Development of a parallel Poisson equation solver for two-phase flow simulations, MASCOT and ISGG 2012, (2012).
7. Ono, K.: Technology that joins product design and simulation‐ Construction of HPC/PF and its application scenario, JSME 25th Computational Mechanics Division Conference, (2012).
8. Onishi, J. and Ono, K.: Convergence properties of the Poisson equation solvers for two-phase fluid flows (2nd report: Applicability of the MultiGrid Preconditioned Conjugate Gradient Method), The 26th CFD symposium, B09-4, (2012).
9. Onishi, J. and Ono, K.: Verification of a sharp interface method for two-phase flow simulations, Annual meeting of The Japanese Society for Multiphase Flow 2012, (2012).

(3) Invited Talks

1. Ono, K.: Strategy of Visualization toward Exa-Scale Computing, PC cluster consortium workshop in Kyoto, Jan., Kyoto, Japan, (2012).
2. Ono, K.: Automatic Grid Generation over 10 Billion Scale and Visualization, VINAS user group meeting, Oct., Tokyo, Japan, (2012).
3. Ono, K., Onishi, J. and Kawanabe, T.: Issues and approach for large-scale CFD on K computer, International Workshop on Future of CFD and Aerospace Sciences, Dec., Pusan, Korea, (2012).

(4) Posters and presentations

1. Ono, K.: Large-scale CFD for Industrial Application and Post Processing, 3rd International WS of large-scale visualization, Oct., Kobe, Japan, (2012).
2. Ono, K.: Research Activities of Visualization on K-computer, The 3rd AICS International Symposium: Computer and Computational Science for Exascale Computing, Feb., Kobe, Japan, (2013).
3. Ono, K.: Role of numerical library on road map for exa-flops computer, Automatic Tuning Research Group meeting, Jul., Koganei, Tokyo, (2012).
4. Bi, C. and Ono, K.: POD-Based Parallel Compression for Visualizing Large-Scale Dataset, High Performance Computing Symposium 2013, Jan., Tokyo, Japan, (2013).

(5) Patents and Deliverables

-None