A real-time visualization system on the Grid

Hiroyuki Matsumoto (410112)

School of Information Science,
Japan Advanced Institute of Science and Technology

February 9, 2006

Keywords: Real-time visualization system, Jini, UNICORE, 3D CFD.

Recently, the computer improved the performance, and CFD (Computational Fluid Dynamics) simulation has developed with the performance improvement of the computer. However, it is insufficient to solve the complex and large-scale problem, and the performance of a computer has limit. This is surmountable by the grid computing technology.

The grid computing technology builds a efficient calculation resources by dispersed calculation resources are connected in a network. A user does not need to be conscious about calculation resources are distributing, communication routing, encrypted communication. There is UNICORE (UNiform Interface to COmputing REsorces) [1] as middleware which constitutes the grid computing technology. The Jini technology [2] can build JavaSpaces which can share the Java object on the network and can flexibly operates even if calculation resource is changed the number and the processing performance.

If the problem becomes complex and large-scale, it takes a lot of time until the calculation result until obtaining it. Therefore, even if the failure of the calculation happens, the user needs a lot of time until it knows the failure. Writing the calculation result in the file, and visualization are the problems, because there are storage capacity and a load when the file is output. This is surmountable by the real-time visualization system made visible together with execution of simulation without file output.

[1] UNICORE,"http://www.unicore.org"

[2] Jini technology,"http://www.sun.com/software/jini"

In this research, a real-time visualization system for 3D (Three Dimension) CFD simulation was constructed on the grid which used the UNICORE and the Jini technology, and the effectiveness of the system was verified. The volume of calculation data of 3D CFD simulation is more than that of 2D CFD simulation. This system use the JavaSpaces that tha Jini technology offers as a shared memory. The Jini technology is composed of the Java language which supports the multi-platform. Because UNICORE was composed by the Java language and the Perl language, this system used it for the grid middleware.

This system mounted real-time visualization system which forwards calculation result to system every time result of simulation is obtained and makes to visible. This system can inform user of progress situation of calculation at once and simulation result. If calculation data of 3D is treated as information of visualization, the load from communication and drawing processing is caused. This system decreases calculation data necessary for visualization by using the section vertically cut in an arbitrary coordinate axis for visualization. In addition, this system mounts a change and an additional function of the section of visualization. User can confirm behavior in the direction od the depth by confirming an arbitrary section of visualization.

The analysis about flow around 3D rectangular cylinder was done by using this system. The achievement of real-time visualization by section of an arbitrary coordinate axis, mounting the function to add the section of visualization, and mounting the steering function was confirmed respectively.

The influence of the simulation by visualization on the exection time was measured, and the delay was seen respectively. The delay of 2.22% was in the visualization of the pressure distribution chart, and the delay of 4.65% was in the visualization of the speed vector chart.

The measurement of the speed-up rate was done. The speed-up rate 1.91 was obtained for the measurement of four in the homogeneous distributed calculation environment, and speed-up rate 2.02 was obutained for the measurement of eight in the heterogeneous distributed calculation environment.

This system applied a dynamic load-balancing which made JavaSpaces

a task bag, and load-balancing was verified. When the number of partitions of the task and the number of computer are the same, the execution frequency of task did not depend on the computer performance, and load-balancing was not done. However, when the number of partitions of task was more than the number of computer, the exection frequency of the task had changed in proportion to the performance, and load-balancing was done.