Simulation Monitoring System Using AVS

Tadashi Watanabe[1], Etsuo Kume[1], and Katsumi Kato[2]

[1] Center for Promotion of Computational Science and Engineering,
Japan Atomic Energy Research Institute, Tokai-mura, Ibaraki-ken,
319-1195, Japan
watanabe@sugar.tokai.jaeri.go.jp
kume@brian.tokai.jaeri.go.jp
[2] Research Organization for Information Science and Technology,
Tokai-mura, Ibaraki-ken, 319-1195, Japan
kato@sugar.tokai.jaeri.go.jp

Abstract. A simulation monitoring system has been developed to visualize ongoing numerical simulations on supercomputers or workstations. The output data for visualization are transferred from the calculation server to the visualization server and visualized automatically by the monitoring system. Visualization is performed by AVS on UNIX or WINDOWS environment. Modification of simulation program is not necessary, and the monitoring system is applied for both interactive and batch process of numerical simulations.

1 Introduction

Visualization or animation technique has been used in computational science as the post processing technology. Numerical simulations are performed on supercomputers in a computer center, and visualization is performed on graphics workstations or personal computers after the simulations. This is because supercomputers are composed of front and back end systems generally and the network queuing system is adopted. The start and end of batch jobs are not known. The numerical simulations in computational science need long computational time. The efficiency of research is not improved if the numerical results are visualized only after the simulations, even though the efficient simulations and visualization are performed. Visualization of ongoing simulations is desirable for large-scale simulations using supercomputers in a computer center.

The visualization systems, in which results of ongoing simulations are transferred from supercomputers or workstations to a visualization server, have been developed and commercialized by computer makers such as NEC[1] and Fujitsu[2]. Similar systems for research and development purposes have been proposed [3,4,5,6]. These visualization systems are adapted to specified computer systems, and it is necessary to modify simulation programs to make the display data or to use libraries for data transfer. Visualization and numerical simulations are closely linked together and applications of these systems are limited. These visualization systems might be useful for simulations using fixed computer codes and computer systems. It is, however, not

P.M.A. Sloot et al. (Eds.): ICCS 2002, LNCS 2329, pp. 990–999, 2002.

convenient to researchers who make and modify their own simulation programs frequently using several types of computer systems.

In this paper, a simulation monitoring system is described. This monitoring system is developed for researchers in a small research group where many kinds of simulation programs are developed and modified frequently, and several computer systems including supercomputers in a computer center and workstations in a laboratory are used for numerical simulations. The dependency of monitoring system on computer environments and the modification of simulation program for visualization are, thus, not preferable. UNIX-based workstations or WINDOWS PCs are used for visualization, and AVS5, AVS/Express, and MicroAVS are used as the visualization software.

2 Simulation Monitoring System

Numerical simulations, for which visualization is performed during calculations, are performed on the calculation server such as workstations in a research laboratory or supercomputer systems in a computer center. Simulation monitoring is performed on the visualization server such as graphics workstations or PCs in the laboratory. Several methods are possible in this situation to visualize results of ongoing simulations. The AVS series are used as the visualization software so that the simulation programs are not modified for visualization. The format of output data is the same for UNIX workstations and for WINDOWS PCs. Display data are not generated on the calculation server and simulation results are transferred from the calculation server to the visualization server. Time for data transfer may not be small according to the simulation data. It is, however, not significant since the computational time is generally much larger than the time for data transfer and visualization. AVS is used for visualization, and thus the change in displayed picture during monitoring is possible and easy.

The simulation monitoring system is composed of two parts: one on the calculation server and the other on the visualization server. The monitoring system on the calculation server is watching the execution and the output of numerical simulations. Once the output file of simulation results is detected by the monitoring system, the output file is transferred from the calculation server to the visualization server by FTP. On the visualization server, the arrival of output file is detected by the monitoring system and visualization is performed by AVS5, AVS/Express, or MicroAVS.

2.1 Detection of Output File on Calculation Server

The execution and the output of numerical simulations are watched by the monitoring system on the calculation server. Once the output file for visualization is generated from the simulations, the monitoring system detects the output file and send it to the visualization server. The generation of output file is easily detected on UNIX workstations, and thus the case with the computer system using batch process is described here.

The output of numerical simulations is sometimes related to the end of batch process on computer systems in a computer center, even if its OS is based on UNIX. In this case, an example of UNIX shell on the front-end of calculation server for detecting the end of batch job is shown below. This example is used in csh environment. The submission of numerical simulations is also included in this example. In this example, "qsub" in line #02 indicates a submission of calculation to the back-end computer system. Parameters for batch process are described in the file "sub.sh" in this case. The start and end of the batch process are not known since many batch jobs are submitted by many users. The output file from the back-end system is detected in line #03 in this example. In our computer system, the output file named "sub.sh.o*" is sent back from the back end to the front end, where "*" in the file name is the job number. The word "logout" in the output file indicates the normal end on the back-end system. After the normal end is detected, the job number is obtained in line #08, and the empty file "lock.*" is temporally made in line #10. The output file for visualization is transferred from the calculation server to the visualization server by FTP in line #11. After sending the data for visualization, the empty file "lock.*" is sent to the visualization server. The arrival of this file to the visualization server is corresponding to the end of data transfer, and the visualization process is started. The procedure from submission to FTP is performed again in the last line in this example. This example is used for a numerical simulation with restart calculations. In this case, the output file for visualization is made once in a batch job, and the normal end of the batch job is detected as the end of output file. In case of a job with multiple output files, it is not difficult to detect the file after the output of each file.

Example of UNIX Shell for Detecting the Output File of Numerical Simulations on Calculation Server

```
## this is submit-shell                                    #01
 qsub sub.sh                                               #02
 until grep logout sub.sh.o*                               #03
       do                                                  #04
       sleep 300                                           #05
       done                                                #06
 cat sub.sh.o* >> sub-out-file                             #07
   a1=`ls sub.sh.o* | awk '{print substr($1,9,5)}'`        #08
   /bin/rm sub.sh.o*                                       #09
   touch lock.$a1                                          #10
   ftp -n ip-address << eod                                #11
     user user-id password                                 #12
     cd ftp-directory                                      #13
     prompt                                                #14
     bin                                                   #15
     put avs-data avs-data.$a1                             #16
     put fld-data fld-data.$a1                             #17
     put lock.$a1                                          #18
   eod                                                     #19
   /bin/rm lock.$a1                                        #20
   nohup submit-shell &                                    #21
```

2.2 Detection of Transferred File on Visualization Server

On the visualization server, the simulation monitoring system is waiting for the arrival of the output file and the lock file. Transferred files are detected by the similar method as described above. The output file is given to AVS after the lock file is detected. If the output file itself is used for detection, AVS may start reading the data before the completion of data transfer, and thus, the lock file is used for detection. The lock file is empty and the time for data transfer is negligible.

The simulation monitoring system uses the AVS series for visualization on UNIX workstations or WINDOWS PCs. The lock file is detected by the AVS modules on UNIX workstations. The AVS modules are written in C language and registered in AVS environment after compilation. On WINDOWS PCs, the monitoring program, which detects the lock file and calls MicroAVS, is written in C language.

3 Monitoring

Examples of monitoring using AVS5, AVS/Express, and MicroAVS are shown in this section. These examples are used for numerical simulations of multi-phase flow phenomena. The monitoring of ongoing calculations is indispensable for transient analyses in many research fields such as fluid dynamics research.

3.1 AVS5

The control panel of the AVS5 module for file detection is shown in Fig.1. This module is connected to the module for reading a data file such as the read_field module in the AVS5 network. The parameters for monitoring are set by using this module. The directory for the output and lock files are selected in "search" and "select" columns. The names of the output and lock files are set in "prefix" and "lock" columns, respectively. The first step number and the increment are set in "first" and "delta" columns, respectively. The output and lock files have the same step number, which may be the simulation step number or job number. The time period for searching the lock file is set in "interval" column. Whether output files are deleted or not after visualization is selected by "delete" or "undelete" button. Monitoring is started by "run" button, and the current step number is indicated in "current" column.

An example of the AVS5 network is shown in Fig. 2. The name of the AVS5 module for file detection is "Check_Step" in this example. The Check_Step module gives the file name to the read_filed module after the lock file is detected. Scalar variables in field data are visualized in this example.

The displayed picture is shown in Fig. 3, where a snapshot of the collision process of two spherical droplets are shown. The numerical simulation is performed using the lattice Boltzmann method, which is one of the particle simulation methods for analyses of complex flow phenomena. The surface of droplets and the pressure field in a cross section are visualized using the isosurface and orthogonal_slicer modules, re-

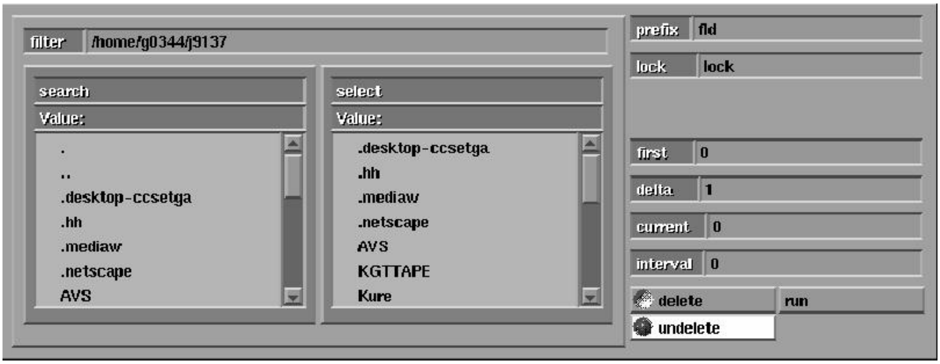

Fig. 1. Control panel of AVS5 module for file detection

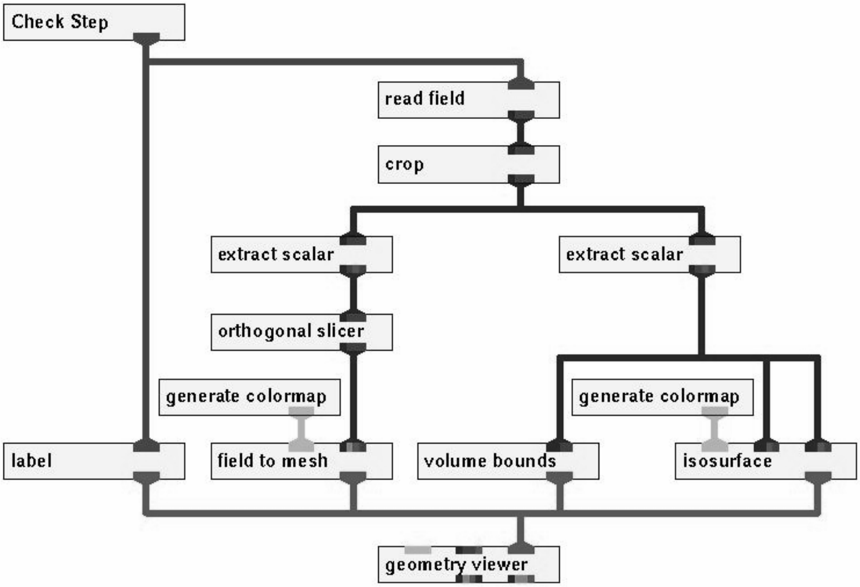

Fig. 2. AVS5 network for monitoring

spectively. This picture is displayed until the next file arrives. The displayed picture can be changed during monitoring since AVS functions are all available.

3.2 AVS/Express

The control panel of the AVS/Express module for file detection is shown in Fig.4. As is the case with AVS5, this module is connected to the module for reading a data file in the AVS/Express application. The parameters have the same meaning as before.

An example of the AVS/Express application is shown in Fig. 5. Scalar variables in field data are visualized in this example. The displayed picture is shown in Fig. 6, where a snapshot of the two-phase flow in rectangular pipe is shown. The numerical simulation is performed using the lattice Boltzmann method, and the two-phase interfaces and the pressure field are visualized.

3.3 MicroAVS

The control panel for file detection is shown in Fig.7 for MicroAVS on WINDOWS PC. This control panel is displayed at the left corner of PC's display. The parameters are the same as before. This control panel is developed in the visual C++ environment.

An example of monitoring is shown in Fig. 8, where a snapshot of the mixing process of two fluids is shown. The numerical simulation is performed using the lattice gas method, which is one of the most simple particle simulation method for calculations of various flows. The concentration of two fluids in a mixing chamber is visualized. As is the case with AVS5 and AVS/Express, all the functions of MicroAVS are available during monitoring.

4 Summary

Computer systems and computational techniques have been developed very rapidly, and large-scale numerical simulations are performed in many science and engineering fields. A huge amount of simulation results is obtained and the efficient visualization tools, which are easy to use for researchers, are desired.

The simulation monitoring system to visualize ongoing numerical simulations on supercomputers or workstations is described in this paper. The output data for visualization are transferred from the calculation server to the visualization server and visualized automatically. Visualization is performed by AVS on UNIX or WINDOWS environment, and thus many types of visualization methods are easily used. Modification of simulation program is not necessary, and the monitoring system is applied for both interactive and batch process of numerical simulations. The monitoring system is useful for a research group, where many kinds of simulation programs are developed and modified frequently, and many types of computer systems are used. In this paper, the examples of monitoring used for numerical simulations of multi-phase flow phenomena are shown, since the monitoring of transient calculation is of importance for

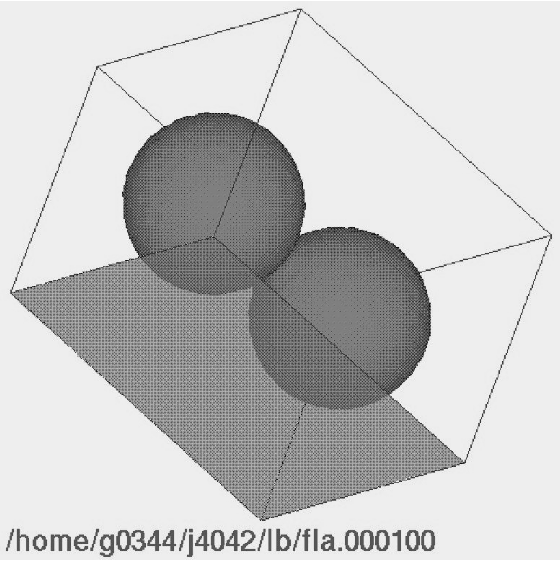

Fig. 3. Example of monitoring using AVS5 (collision process of two droplets)

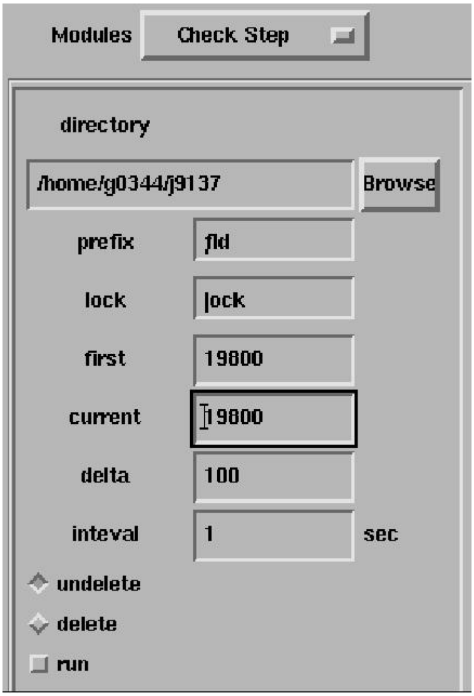

Fig. 4. Control panel of AVS/Express module for file detection

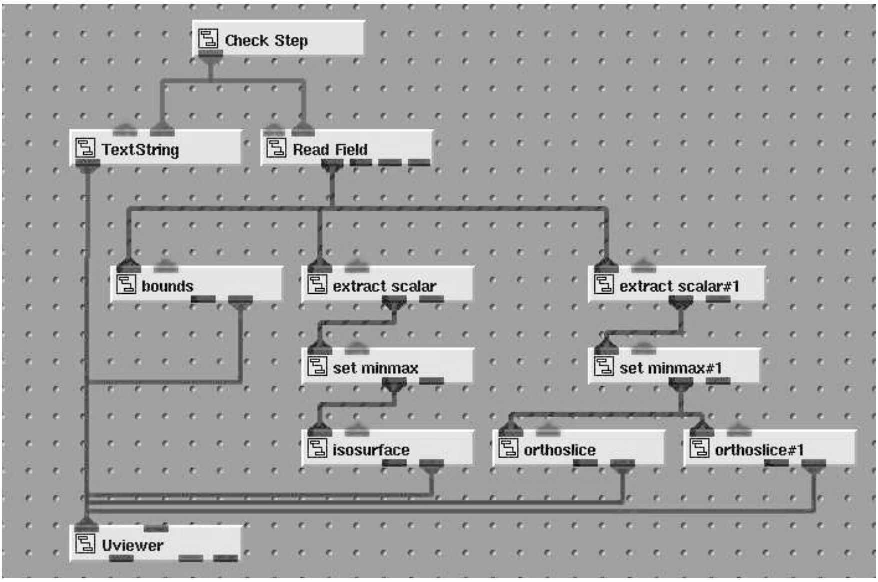

Fig. 5. AVS/Express application for monitoring

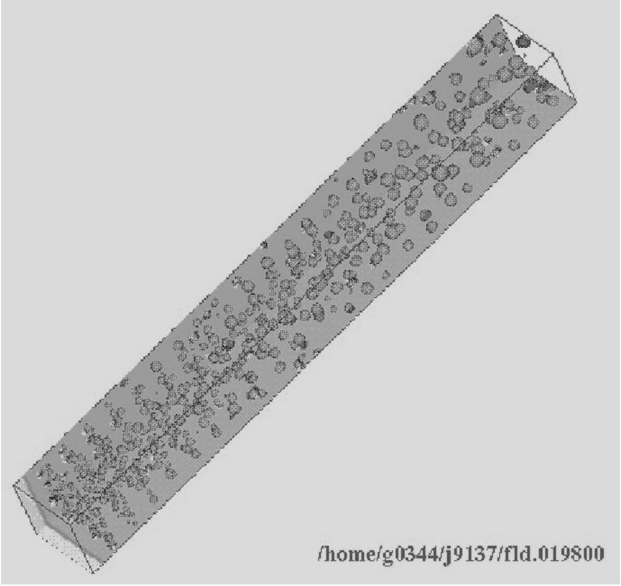

Fig. 6. Example of monitoring using AVS/Express (two-phase flow in a pipe)

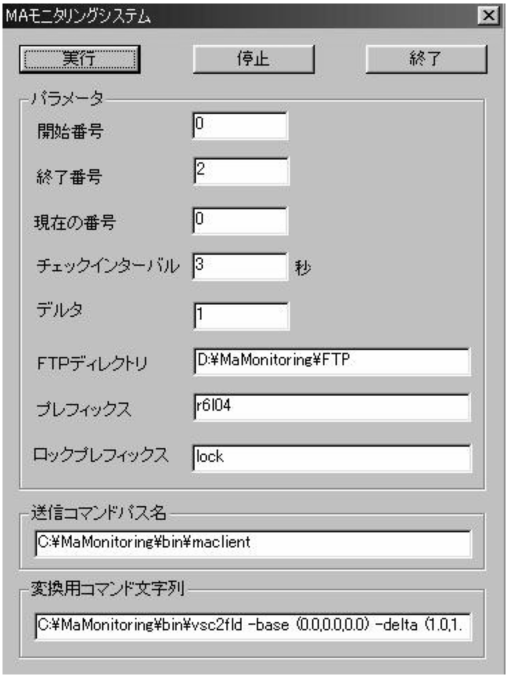

Fig. 7. Control panel for file detection for MicroAVS

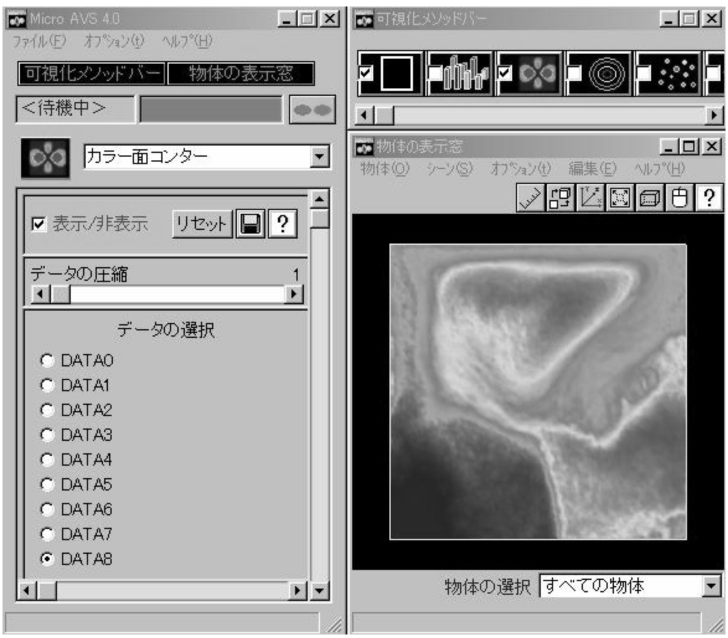

Fig. 8. Example of monitoring using MicroAVS (mixing process of two fluids)

transient analyses in fluid dynamics research. We have a plan to use the simulation monitoring system for 3D visualization on UNIX and WINDOWS environment in the near future.

References

1. http://www.sw.nec.co.jp/APSOFT/SX/rvslib/index.
2. http://www.nagano.fujitsu.com/avs/Products/Relation/
3. Muramatsu, et al.: Development of Real Time Visualization System for Fluid Analyses on Parallel Computers, JAERI-Data/Code 98-014(1998)
4. CRIEPI-Hitachi Joint Research Group on Parallel Applications: A Study on Software Infrastructures for Computational Science and Engineering, U99033(2000)
5. http://raphael.mit.edu/visual3/visual3.html
6. http://www.icase.edu