

A Remote Parallel Computation System for Biomedical Data Visualization on the Grid

Chih-Han Lin Lai-Yi Chu Chun-Cheng Wei Ming-Shi Wang*

Department of Engineering Science, National Cheng Kung University, Tainan 701, Taiwan

*E_mail : mswang@mail.ncku.edu.tw

Abstract

In this paper, a collaborative system for visualizing biomedical dataset on a Grid environment was proposed. Globus Toolkit was used for the integration of computational resources in the Grid environment, and the Visualization Toolkit (VTK) is used as the tool for processing the biomedical dataset. The results can be viewed by the users who are located at different places and connected together via internet. To allow the users to discuss the results cooperatively, Access Grid (AG) is used as the basic operation platform. Several shared applications were designed to allow users at remote sites to share user interface and visualizations. The system allows for the manipulating of the viewing objects by rotation, translation, and scaling.

Keywords: Grid Computing, Globus Toolkit, Access Grid, Collaborative visualization, Parallel computing

1. Introduction

In the recent years, the improvements in imaging techniques have led to the development of higher resolution devices, and resulting in ever increasing amounts of data being produced. For example, biomedical data is data collected by applying image techniques such as computed tomography (CT), magnetic resonance imaging (MRI) and positron emission tomography (PET), to an object of biological or medical origin.

In order for this dataset to be analyzed easily, it has to be converted into visual representations such as 3-D model or 2-D slices, this process is called the scientific visualization. Physicians rely on these visualizations to better understanding the information inside this dataset and to diagnose patients rapidly.

The visualization of biomedical data is becoming more popular with the years, information sources of this data may be widely distributed, and the volume of dataset to be processed is usually extremely large, and the visualization process is time-consuming.

To reduce the processing time, the technique of parallel computation via multiple computation nodes is usually used solution. In this report, Globus Toolkit [6] was used as the backbone to create a parallel computation environment. The commercial package, Visualization Toolkit (VTK) [1], is used as the tool for processing the visualization data. In order for the real time interaction of the 3D model, the designed system provides a sub-sampling mechanism for the users. The 3D model and 2D slices are generated by the computing nodes on the Grid. A simple Resource Broker was designed for submitting the jobs for the Grid.

In order to realize the collaboration environment for the users, Access Grid (AG) [7] over the Internet is used in this report. We also extended the shared application of the Access Grid, by which the users who join on the discussion room can interactively operate the viewing object.

2. Background Review

2.1. Globus Toolkit

The Globus Toolkit is an open source software which includes a suite of applications and libraries that work together to provide a suite of components for supporting grid services. It is used by many organizations to build Computational Grid that supports their applications. There are four core components of the Globus Toolkit needed for creating a Grid computing environment. These are: Grid Security Infrastructure (GSI), Grid Resource Allocation Management (GRAM), Grid Information Services (GIS) and Grid File Transfer Protocol (GridFTP).

2.2. The Access Grid

The Access Grid is a collaborative environment, which is used for group to group collaboration via high-speed networking over the Internet. It provides

high quality audio and real-time video, and allows participants to imitative human interaction. Another major component is application sharing tools between participants, such as shared browser, shared question tool and shared presentation. There are two approaches to extend the functionality of the Access Grid. One is the creation of shared applications. The other is to deploy the services using the node service infrastructure.

2.3. Visualization Toolkit

The Visualization Toolkit is an open source freely available software toolkit for 3D computer graphics, image processing, and visualization. It has recently been extended to process visualization tasks in parallel [2]. VTK offers an open source and multi-platform solutions.

In VTK, communication between processes uses a class vtkMPIController that is implemented on the top of MPI (Message Passing Interface) and the MPICH libraries [5]. VTK also has a communication class, vtkSocketController, based on sockets which can be transparently substituted for MPI based class. On the top of these communication classes, VTK has implemented remote-method invocations (RMIs), which can be registered with these communication classes.

VTK supports three distinct parallel models [2], task parallelism, pipeline parallelism, and data parallelism. In this report, data parallelism is adopted. In data parallelism model, the large dataset is evenly divided into independent subsets using the visualization toolkit's streaming data model [3]. Each subset of data is processed independently and called a process. Identical pipelines are created in every process.

3. Implementation

The Grid environment demonstrated in this report was made up of eight personal computers, each with CPU 2.8GHz, 512MB RAM, and Fedora Core 3 operating system. For the software, Globus 2.4, VTK 4.2.1 and AG 2.4 were installed in each node.

3.1. System Architecture

Figure 1 shows the overall system architecture. The client users can use the SharedAPJobSubmit service located in the VenueClient to invoke a job submission request. This request is sent to the SharedAPSJServer which is located in the Venue server of the AG system. Then the SharedAPSJServer transmits the request to the Resource Broker for real job submission. All the

computing results produced by computing nodes on Grid system will be collected by the rendering node.

The rendering node is connected with SharedApp3DServer service via VTK's Socket, and accepts all the VTK control messages that provided by the client users and send these messages to the remote VTK program located in the computing nodes by RMIs.

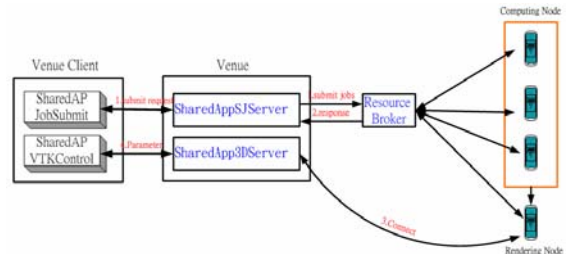


Fig. 1: System Architecture

3.2. Resource Broker

The use of the resource broker provides a uniform interface to access any of the available and appropriate resources using user's credentials. The Resource Broker over the Globus Toolkit is used to do the jobs of process dispatch, scheduling and submitting for the computation nodes of the Grid. The resource broker takes the responsibility to find the resources from the MDS service provided by Globus and to match the required resources that made by user's job requests. Once the resource broker found the required resources, it then used the GRAM service to dispatch the jobs to these resources. Figure 2 shows the relationship among the SharedAPJobSubmit service, SharedAPSJServer service, and the Resource Broker.

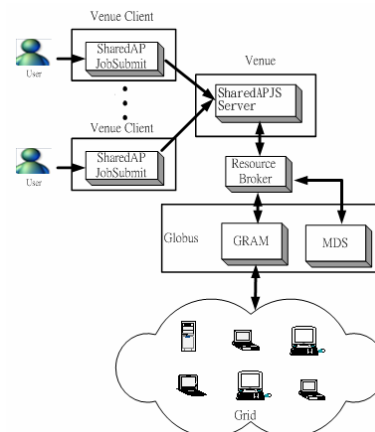


Fig. 2: The role of Resource Broker for job submitting

3.3. Data Parallelism in Our System

Example of data parallel pipeline used for the extraction of 3-D model and 2-D slices in computing nodes of the system is shown in Figure 3, this is an

example of 4 processes executed on different node of the Grid, and each computing node has the identical pipeline. Process 0 is the “master” process, and the others are the “slave” processes. Slave processes are responsible for the computation of extracting the 3-D model and 2-D slices. The master process is responsible for collecting the results, by using the `vtkAppendedPolyData` class which use VTK’s streaming data model for data partition, that produced by slave processes.

The communications among master and slave processes are handled by four VTK classes called `vtkSocketController`, `vtkSocketCommunicator`, `vtkOutputPort`, and `vtkInputPort`. The class `vtkInputPort` and `vtkOutputPort` are used to connect the pipeline, whereas `vtkSocketController` and `vtkSocketCommunicator` are used to pass data around.

In slave processes, several RMI functions were written and registered with the `vtkSocketController`. The master process can trigger these RMI functions to modify the parameter of the 3-D model (ex: transparency, color, rotate and isovalue, etc...), and the orientation or normal of the 2-D slices.

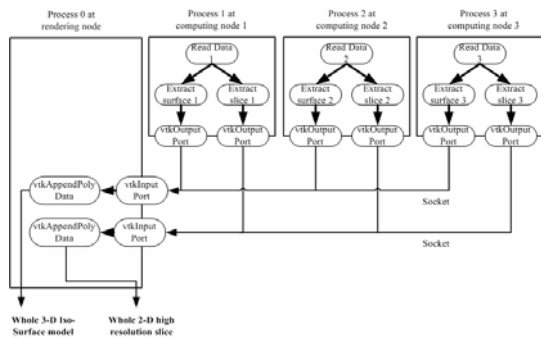


Fig. 3: The data parallel pipeline

3.4. Integrate Our System with Access Grid

To provide the capability for the users on the Access Grid to view and control the result images generated by the rendering node, it is needed to communicate between these users and the remote rendering node. Two classes, `vtkSocketController` and `vtkSocketCommunicator` are used to exchange the data over the VTK pipeline. A shared application called `SharedApp3DServer` service was designed in Access Grid and used to communicate with the above two classes to provide the information exchange between AG’s venue server and the rendering node. The `SharedApp3DServer` can also receive the control messages of the users in the event channel, and then send these control messages to the remote VTK programs by the means of RMIs.

A VTKControl shared application was also designed and installed in the venue clients for the users to operate the remote VTK program interactively. Figure 4 shows the whole integrated system.

The remote VTK program uses the FLXmitter [9] to convert these VTK’s result images from RGB signals to YUV signals. These signals will be encoded with H.261 format and then transferred to the clients of the AG system via Multicast Environment by RTP.

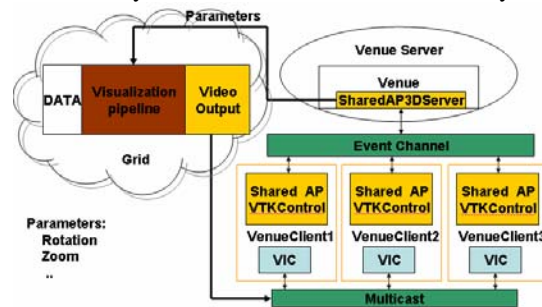


Fig. 4: Integration of the system and Access Grid

4. Results

The experimental dataset was taken from the ACM SIGGRAPH [4]. It is a 109-slice MRI data set of a human head. The complete slices are stored consecutively as a 256 x 256 array each with 256 gray levels, pixels consisting of 2 consecutive bytes making one binary integer.

Figure 5 is shown the basic three Access Grid user interface. When the Resource Broker is started, it will create a `SubmitJob` service in the application sessions of the Venue Client. Users can start the service for job submission requests by double click on the “SubmitJob” service in Venue Client’s application sessions region, and then a “Submit Job Panel” will be opened. As show in figure 6, users can choose the Data Set, request the number of machines (processors), select the executed program, and configure some parameters. Once the user finishes these configurations, he can click on the send button to send this job request to the Resource Broker for executing the job dispatch, scheduling and submission.

When the job had been done, the result can be visualized in 3-D model or 2-D slices in the video panel of the clients and then the Venue Client will create a shared application service called “VTKControl” service in the application sessions region to allow the users to operate the view. Of course, all the users logged in the same venue server should view the same result. Only one of the users’ operations will be adapted by the system. The users on the same venue can be communicated via the audio tool, RAT, of AG.



Venue Client

Video

Fig. 5: The screen layouts of Access Grid

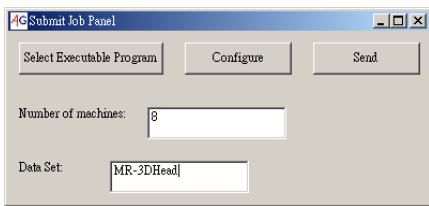
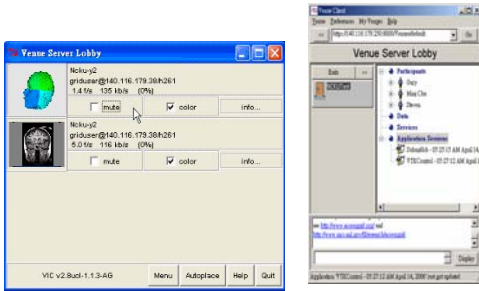


Fig. 6: The Submit Job Panel



Video

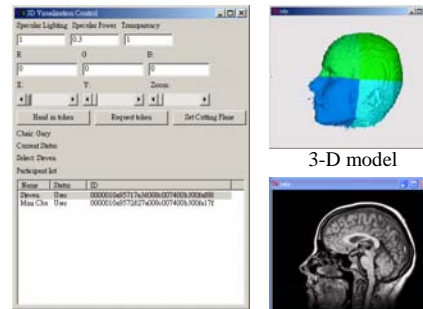
Venue Client

Fig. 7: The result images and the VTKControl service

Figure 7 is shown the result images and the VTKControl service created in Venue Client., As show in figure 8, with the VTKControl panel, users can set and change parameters for the 3-D model, respectively. The parameters are RGB color, specular lighting, specular power, transparency, rotate, and zoom. Also, users can allow extracting another 2-D slice by click on the button of "Set Cutting Plan".

5. Conclusions

The primary goal of this system is to provide an approach to effectively visualize and analyze the biomedical dataset stored remotely on the Grid. In this paper, a collaborative system that combined Grid environment and Access Grid was proposed. The Globus Toolkit was used for the integration of computational resources in the Grid environment, and the Visualization Toolkit (VTK) is used as the tool for processing the biomedical dataset. The Access Grid, a video conference system, is used as the basic operation



VTKControl panel

3-D model

2-D slice

Fig. 8: The VTKControl panel can re-extract the 3-D model and 2-D slice

platform to connect the users located at different places together to discuss the same subject on the internet.

The results generated by the Grid computing facilities can be viewed by the users on the Access Grid system. A Resource Broker was designed to do the jobs of process dispatch, scheduling and submitting for the computation nodes of the Grid. Several shared applications were designed to allow the users at remote sites to share user interface and visualizations.

6. References

- [1] W. Schroeder, K. Martin, B. Lorensen, "The Visualization Toolkit: An Object-Oriented Approach to 3D Graphics. Third Edition," Kitware, Inc., 2004.
- [2] J. Ahrens, C. C. Law, W. J. Schroeder, K. Martin, M. Papka, "A parallel approach for efficiently visualizing extremely large time-varying datasets," Tech. Rep. LAUR-00-1620, Los Alamos National Laboratory, 2000.
- [3] C. C. Law, W. J. Schroeder, K. M. Martin, J. Temkin, "A multi-threaded streaming pipeline architecture for large structured datasets," Proceedings of Visualization'99, IEEE Computer Society Press, 1999.
- [4] http://www.siggraph.org/education/materials/vol-viz/volume_visualization_data_sets.htm
- [5] The MPI, <http://www-unix.mcs.anl.gov/mpi/>
- [6] The Globus Alliance, <http://www.globus.org>.
- [7] The Access Grid, <http://www.accessgrid.org>
- [8] I. Rodero, J. Corbalan, R. M. Badia, and J. Labarta. "eNANOS Grid Resource Broker," In Peter M. A. Sloot et al., editors, Advances in Grid Computing - EGC 2005, 2005.
- [9] The FLXmitter SPU, <http://fl-cvs.mcs.anl.gov/viewcvs/viewcvs.cgi/fl/FLXmitter/>