2012 4th International Conference on Computer Modeling and Simulation (ICCMS 2012)
IPCSIT vol.22 (2012) © (2012) IACSIT Press, Singapore

Research and implementation of a web-based grid-enabled visualization system

HAN Wei-jie[1], Wang yu[1], Yan Hui[2], Zhang Xue-bo[1]

[1]Department of Information Equipment, The Academy of Equipment
[2]Department of Scientific Research, The Academy of Equipment

Abstract. Grid-based visualization system can aggregate and share geographically distributed data sources, analysis facilities and visualization systems, and also offer transparent data access capabilities. We develop a web-based visualization system on the grid. The system consists of three components: the WebSheet, the web server and the visualization server. The front-end of the system provides the users with the web-based access interface built on JSP, and the embedded VisPortal provides grid access portal for the users. The middle part of the system is the web server which aims to manipulate, dispatch, utilize grid resources and mediate the interaction between the user and the grid. The back-end of the system is the visualization server and we build one based on Vis5D as a grid, which delivers grid-based visualization services of 5-D gridded datasets generated from the numerical weather models.

Keywords :Web-based visualization, grid, VisPortal, visualization server

1. Introduction

Web-based visualization can utilize distributed remotely-located computing, visualization, rendering and storage resources and delivers advanced remote/distributed visualization services. But web-based visualization will be faced with the problem about how to utilize the heterogeneous resources distributed in the Internet[1,2]. The grid technology emerges as the times require being a distributed Internet-oriented technology.

Grid computing[3] aims to aggregate and integrate various kinds of heterogeneous high performance computers, data servers, large-scale storage systems and visualization systems geographically distributed in the Internet by high-speed network in order to solve the challenging applied research problems difficult to complete by a standalone researcher. Grid technology can satisfy the complex computing and storage requirements by dynamic resource management, expand the capability and scope of the researchers to visualize and understand the complex scientific datasets.

In order to utilize the computing, storage and visualization resources located in Computer Network Information Center (CNIC) of Chinese Academy of Sciences, we develop a web-based grid-enabled visualization system. The system consists of three components. The front-end of the system provides dynamic WebSheet for users as the web-based access interface built on the Tomcat/JSP web engine, in which the embedded VisPortal provides grid access portal for the users. The middle part aims to aggregate, manage and utilize the distributed grid resources and also mediates the interaction between VisPortal and the grid resources. The back-end of the system consists of a wide variety of grid resources. We build a visualization server based on Vis5D as a grid resource which runs on a high-performance computer, called Lenovo DeepComp 6800, located in CNIC and delivers remote visualization services to researchers.

2. Related Works

In the filed of grid-based visualization, researchers have carried out a wide range of research and developed a number of applications.

Martin Aeschlimann et al[4] designed a framework, called Dv, for building interactive distributed visualizations on computational grids. The system provides a flexible framework for building distributed applications that are performance-portable in the presence of heterogeneous resources and that adapted to dynamic changes in the status of system resources such as processor cycles and network bandwidth.

Knosp et al[5] designed a grid-based volume rendering framework which utilizes the services that Globus provided such as resources management, information scheduling and data transfer tools so as to realize volume rendering.

The Texas Advanced Computing Center located at the University of Texas at Austin[6] developed a campus grid, called UT Grid, which not only realizes visualization and rendering by utilizing grid resources, but also provides remote visualization services for users.

Some other researchers have explored to develop grid-oriented visualization portal which supplies visualization services for users through the grid portal.

The LBNL/NERSC Visportal[7] effort explored ways to deliver advanced remote/distributed visualization capabilities through a grid-enabled web-portal interface. The effort focuses on latency tolerant distributed visualization algorithms, GUI designed that are more appropriate for the capabilities of web interfaces, and refactoring parallel-distributed applications to work in a N-tired component deployment strategy.

The NPACI Grid Portal Toolkit (GridPort)[8] facilitated development and utilization of Grid technologies such as the Globus Toolkit and the Storage Resource Broker from within an integrated, unified API. GridPort supports a set of centralized services that allows multiple applications portals to share services and a single-login environment.

In addition, there were some other researchers who studied about extending the existing visualization systems to exploit new grid technologies. The gViz[9] project aims to provide e-Scientist with visualization software that works within modern Grid environments. A major part of gViz is the Grid-enabling of a widely used visualization system, IRIS Explorer from NAG. The extension allows this network to span a set of Grid resources, so that user interface modules execute on the scientist’s desktop, but computationally intensive modules are launched securely on remote servers using Globus middleware.

The above visualization systems explore various kinds of ways to deliver remote/distributed visualization capabilities by making use of a variety of grid resources. Their development experiences make it somewhat easier for us to design and implement a web-based visualization system on the grid. Our process of developing the visualization system is in accordance with the Open Grid Service Architecture, and the deployment and management of grid resources in accordance with the Global Grid Forum standard. The system currently is running on Lenovo DeepComp 6800 located in CNIC and delivers remote visualization services of 5-D gridded datasets generated from the numerical weather models.

3. Architecture

The architecture of the visualization system is show in Fig.1.

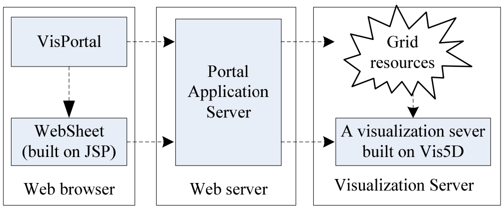


Fig.1 The architecture of the visualization system

3.1. Web-based Grid Portal

The grid portal provides access interface to grid resources for users. The users can utilize the grid resources and realize various kinds of visualization tasks remotely. The WebSheet is built on JSP and provides an accessible UI to the users for grid resources through the embedded grid portal—VisPortal. When a user accesses the URL of our visualization system, the WebSheet will be downloaded to the client-side desktop and the available grid resources accessible for users are listed and displayed in the WebSheet.

The VisPortal provides the single-point login mechanism for users to access the grid resources through a Web portal interface, and hides the under-lying complexity of resource selection and distributed application management on heterogeneous distributed computation and storage resources from the users. The user can access the grid resources without worrying about the complex access process.

3.2 . Web Server

The web server acts as the connection bridge between the front-end WebSheet and the back-end grid resources, which manages the grid resources on the one hand and also monitors the communication process between the users and the grid resources and assists the users to access and utilize grid resources to realize visualization tasks.

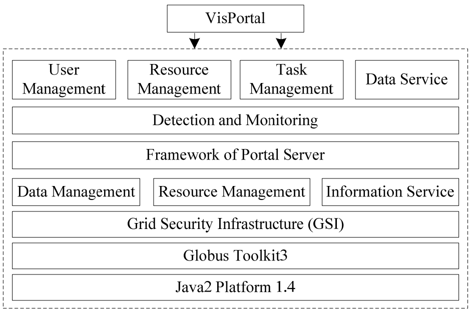


Fig.2 The architecture of the Portal application server

The Portal application server embedded in the Web server is the core of our visualization system whose architecture is show in Fig.2. It provides the five important services for VisPortal including user management, resource management, task management, data service and detection and monitoring service. These above services supplies opportunities for parallel, remote and collaborative visualization based on the following functions such as resource management, task management, user management, security mechanism and data transfer. These functions are implemented based on the services provided by Globus Toolkit3 including data management, resource management, information service and grid security infrastructure as shown in Fig.2.

The user management service provided by the Portal application server aims to support a multi-user collaboratory environment by storing the user's settings, recording the user's operations, maintaining the user's task list and taking statistics of the resources used by the user. When a user submits a visualization task to the VisPortal, the VisPortal will create a task description file based on XML and then deliver it to the user agent generated by the Portal application server. Finally, the user agent launches the visualization task based on the required resources specified by the task description file.

The resource management service provided by the Portal application server aims to aggregate the available grid resources such as computing resources, storage resources and rendering resources so on. The service monitors the grid resources constantly and dynamically and discovers the available grid nodes timely. Then, the information of the available grid resources is registered into the service registration center and the resource management center of the Portal application server. The registered information is described in the form of Resource Specification Language. In this manner, the resource management service will select available grid nodes in response to the request from the VisPortal according to the specification of the registered information. Consequently, the selected grid nodes register their provided services into the service registration center and refresh their status information stored in the resource management center. The implementation of the resource management service is based on the information service offered by Globus Toolkit3[10].

The task management service provided by the Portal application server is offered by the user agent aiming to support task launching, scheduling and collaboration. After receiving a visualization request, the user agent fetches the information of the available grid nodes from the service registration center and the resource management center according to the task description file, and then launches an appropriate number of processes or threads running on the grid nodes based on the task launching service of the grid node.

The data service provided by the Portal application server is realized based on GridFTP which is a high-performance and reliable data-transfer protocol. When the user agent needs to transfer data among the grid nodes, the service will be called. The data service guarantee the integrity and confidentiality of data transmission based on Grid Security Infrastructure.

The grid monitoring service provided by the Portal application server aims to monitor the real-time status of the grid nodes and display their status information for the server. The grid detection service provided by the Portal application server is implemented by adding specific detection procedures into the grid node based on Globus. Based on the grid monitoring and detection service, our visualization system can realize automatic error detection and fault handling so as to regulate the system performance and adjust the grid resources scheduling policy.

3.3. Building a Visualization Server based on Vis5D

Finally, we build a visualization server based on Vis5D as a grid resource. Vis5D[11] is an application software written by the Visualization Project at the University of Wisconsin-Madison Space Science and Engineering Center. Vis5D can realize interactive visualization of large 5-D gridded datasets such as those produced by numerical weather models. One can make isosurface, contour line slices, colored slices, volume renderings etc of data in a 3-D grid, then rotate and animate the images in real time. There's also a feature for wind trajectory tracing, a way to make text annotations for publications, support for interactive data analysis. In the process of building the visualization server based on Vis5D, the following issues need to be solved.

3.3.1 Adding Communication Module

In a web-based visualization system, the back-end visualization server receives the visualization requests from the front-end web-based UI and returns the visualization results to the client side correspondingly. In order to build a visualization server based on Vis5D, the communication function module should be added by modifying its standalone architecture. In order to improve the communication efficiency, we build a single-process multitasking server based on Vis5D by adopting the select() polling mechanism based on TCP/IP. In this manner, the overhead of deriving sub-processes from the server can be avoided and the communication latency can be decreased efficiently. The select() function needs three file description symbols. Once a file description symbol changes along with the user process, the function will be called and submit the changed file description symbol to the user process for processing correspondingly.

Because the visualization server is built based on C language and the WebSheet is built on JSP, the issue of data format conversion between the visualization server and the client side must be solved in order to implement the communication capability of receiving/sending visualization parameters and returning the visualization results. During the communication process, the visualization server receives the byte data sent from the WebSheet and stores it into a character array firstly. Secondly, the character array is converted to integer data or real data by data format conversion functions. In this manner, when the visualization server returns visualization data, the data is converted to a string firstly and then stored into a byte array and converted to other formats.

3.3.2 Multi-thread Parallel Processing Mechanism

The visualization server is running on the high-performance computer called Lenovo DeepComp 6800. In order to take full advantage of computing and rendering resources of Lenovo DeepComp 6800, the multi-thread parallel processing mechanism[12] is introduced to process visualization tasks parallel. The parallel process of handling visualization tasks is as follows:

(1) During the stage of initiation, the visualization server obtains the current number of available CPUs and determines how many tasks to process in parallel;

(2) During the stage of receiving visualization requests, the visualization server establishes a visualization request queue. After receiving a request, the server inserts it into the queue.

(3) During the stage of handling the visualization request, the visualization server fetches the same number of visualization tasks from the queue according to the number of threads created initially. Then the server assigns the visualization tasks for each thread and the threads start running in parallel.

(4) When a thread completes a visualization task, the resources occupied by the thread will be released and ready for the next visualization process.

In this way, parallel processing can be achieved and the efficiency of visualization can be improved significantly.

3.3.3 Offscreen Rendering

The process of computation and visualization will be both executed on the visualization server in order to take full advantage of the computing and rendering resources of the high-performance computer. However, because the high-performance computer is always not equipped with graphics and display devices, the offscreen rendering technology[13] should be introduced. During the process of offscreen rendering, the rendering image will be firstly drawn into the memory buffer in stead of the screen. Then the image will be taken out from the memory and saved as an image file.

Our visualization server realizes offscreen rendering based on X Windows mechanism offered by Linux. During the visualization process, the rendering image and its corresponding window are mapped to a display instead of being displayed directly by the function XMapWindows(display, window). At this time, the visualization graphics elements are shown in the window behind the scenes practically. Afterwards the graphics elements are taken out of the memory and saved as image files.

3.3.4 Responding to Interaction Requests from the Client Side

When the client user browses the visualization image in the WebSheet, he might rotate/translate/drag the image at random. Then the visualization server needs to respond to the client actions and redraws the image.

(1) Responding to rotating

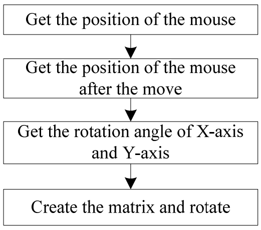

Get the position of the mouse
Get the position of the mouse after the move
Get the rotation angle of X-axis and Y-axis
Create the matrix and rotate

Fig.3 Rotating of visualization image

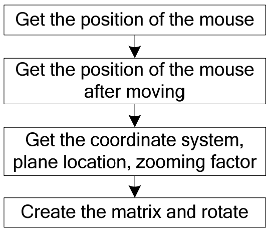

Get the position of the mouse
Get the position of the mouse after moving
Get the coordinate system, plane location, zooming factor
Create the matrix and rotate

Fig.4 Translating of visualization image

The visualization server realizes the rotation of an image based on the function vis5d_set_matrix(int index, float ctm[4][4]). The parameter index specifies the scene to rotate. The parameter ctm represents the matrix which specifies the image how to rotate. The rotation process is show in Fig.3.

(2) Responding to dragging

The process of responding to dragging an image is realized in a similar way as rotation.

(3) Responding to translating

The visualization server realizes the translation of an image based on the function vis5d_set_camera(int index, int perspec, float front, float zoom). The parameter index specifies the scene to rotate, perspec specifies the coordinates of the scene, front specifies the location of the front side, zoom specifies the zooming factor. The translating process is show in Fig.4.

In the above three transformation processes, it is necessary to create the transformation matrix by the function make_matrix(float rotx, float roty, float rotz, float scale, float transx, float transy, float transz, MATRIX mat).

4. A Practical Visualization CASE

The process of remote visualization based on our visualization system is as follows:

(1) When a user accesses the WebSheet, the VisPortal launches and establishes the connection with the Portal server.

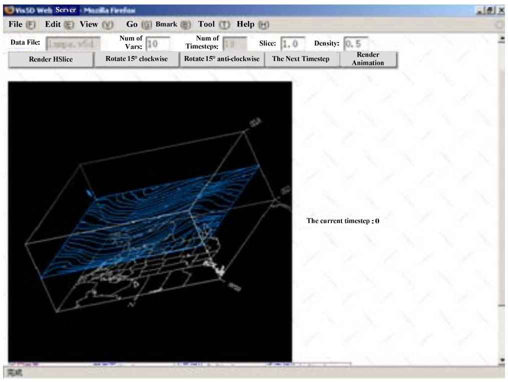
(2) The user management service of the Portal application server verifies the identity of the user. If succeed, the Portal application server will generate a user agent for the user. Then, the user agent will display the information of the available datasets and the login user in the VisPortal.

(3) After the user chooses one dataset and launches a visualization task, the VisPortal will generate a task description file and send it to the user agent. The user agent submits the resource request to the resource management service according with the task description file.

(4) After obtaining the available grid node, the user agent launches the visualization application by the task launching function. The client side will launch the corresponding visualization interface, and the address and port of the selected grid node will be sent to the web-based interface from the task management service. Then the connection between the client side and the grid node will be established consequently.

(5) After establishing the connection between the client side and the grid node, the visualization task can begin executing.

The rendering horizontal contour slices image of lamps.v5d is show in Fig.5. The file lamps.v5d consists of 190 grids, 10 variables and 19 time steps. The values of the visualization parameters are as follows: density: 0.5 and slice: 1.



Data File: lamps.v5d

Num of Vars: 10

Num of Timesteps: 19

Slice: 1.0

Density: 0.5

The current timestep : 0

Fig.5 Rendering horizontal contour slices image of lamps.v5d

5. Conclusions and Future Work

The grid visualization technology emerges based on parallel visualization, remote visualization, collaborative visualization and grid computation technology. It provides advanced capabilities for visualizing the scientific computing results and establishes an excellent platform for a wider range of researchers to realize remote visualization or collaborative visualization through the network.

The web-based visualization system we have developed on the grid aggregates various kinds of resources by the VisPortal in order to utilize the storage, computing and rendering resources distributed in the Internet and deliver remote/distributed visualization services. The system can visualize 5-D gridded datasets generated from numerical weather models. The system also has good scalability because it is built based on the grid technology standards. In the future, we will continue to improve the following functions such as the service management, detection and monitoring, and system security so on in order to make it more robust and adaptive.

6. References

[1] Brodlie K. Visualization over the world wide web [J]. Proceedings of Scientific visualization, 1997, PP:23-29.

[2] Jerome Burgaud, Stephane Dutilleul. Simulation steering and interactive visualization over the world wide web[J]. Visual Data Exploration and Analysis V, 1998(3298) , PP :10-18.

[3] Silvana Roncagliolo, Virginica Rusu, Cristian Rusu, et al. Grid Computing Usability Heuristics in Practice[J]. Proceedings of the 8th International Conference on Information Technology: New Generations, 2011, PP:145-150.

[4] M Aeschlimann, P Dinda, L Kallivokas, et al. Preliminary report on the design of a framework for distributed visualization. Proceedings of the 1st International Conference on Parallel and Distributed Processing Techniques and Applications, 1999, PP:1833-1839.

[5] B Knosp, S Wang, J Ni. Grid-based volume rendering. The 2002 ACM/IEEE Conference on Supercomputing, 2002.

[6] Boisseau, J.R. UT Grid: a comprehensive campus cyberinfrastructure. Proceedings of the 13th IEEE International Symposium on High Performance Distributed Computing, 2004, PP:274-275.

[7] Wes Bethel, John Shalf, Cristina Siegerist, et al. VisPortal: Increasing Scientific Productivity by Simplifying Access to and Use of Remote Computational Resources[R]. Lawrence Berkeley National Laboratory, 2004.

[8] Thomas M p, Mock S, Boisseau J. Development of Web toolkits for computational science portals: The NPACI HotPage[J]. Proceedings of the 9th IEEE International Symposium on High Performance Distributed Computing. 2003, PP:308-309.

[9] Jeremy Walton, Jason Wood, Ken Brodlie. Visualization aids steering of complex simulations on the grid[J]. Scientific Computing and Instrumentation, 2004, 21(4), PP:17-19.

[10] The Globus Toolkit 3 Programmer's Tutorial. http://www.casa-sotomaryor.net/gt3-turorial. 2003.

[11] Surachade, Komsan, Worraluk, et al. Meterological Simulation Portal on the Thaigrid System[J]. Proceedings of ECTI-CON 2008. 2008, PP:153-156.

[12] Micah Beck, Huadong Liu, Jian Huang, et al. Scalable Distributed Execution Environment for Large Data Visualization[J]. Proceedings of IEEE International Parallel and Distributed Processing Symposium. 2007. 2007, PP:1-6.

[13] Tobias Rick, Anette von Kapri, Torsten Kuhlen. GPU Implementation of 3D Object Selection by Conic Volume Techniques in Virtual Environments[J]. Proceedings of 2010 IEEE Virtual Reality. 2010, PP:243-246.