

## Remote Visualization on Medical Datasets over Thin Clients with Support of Grid Environment

Aboamama .A. Ahmed, Muhammad .S Abd Latiff, Kamalrulnizam Abu Bakar

*Department of Computer Systems and communications*

*Faculty of Computer Science and Information Technology*

*Universiti Teknologi Malaysia, 81310 UTM Skudai Malaysia*

[Atahar@siswa.utm.my](mailto:Atahar@siswa.utm.my) , [shafie@fksm.utm.my](mailto:shafie@fksm.utm.my) , [kamarul@fksm.utm.my](mailto:kamarul@fksm.utm.my)

**Abstract** - Existing grid enabled data visualization and remote distributed visualization are often take the direction of remote viewing and zooming techniques of remotely processed datasets. However the continuous increase in the size of datasets and visualization operation causes insufficient performance with traditional desktop computers. Additionally, the visualization techniques such as Isosurface is depend on the available resources of the running machine and the size of datasets. Moreover, the continuous demand for powerful computing powers and continuous increase in the size of datasets results an argent need for a grid computing infrastructure. However, some issues raises with current grid such as resources availability at the client machines which are not sufficient enough to process large datasets. Additionally, different output devices and different network bandwidths between the visualization pipeline components often results output suitable for one machine and not suitable for another. This paper focuses and investigates the visualization of medical datasets on thin client devices with support of grid computing.

**Keyword:** Visualization, Grid computing, Medical datasets, visualization techniques, thin client devices.

### 1- Introduction

In the past decade the field of scientific data visualization becomes an emerging research area. It concerns the transformations of numerical datasets into pictorial format that can be observed easily by human. The available visualization systems today are well suited to work as stand alone applications on the desktop. The common requirements of existing visualization systems are to be based on supercomputing machines at the back-end with additional graphics hardware requirement. However, the developed access methods in which these systems can be used remotely are vary and in continuous rapid development. Some of the implementations use extensive utilization of remote client server architecture. Others suggest the development of open services approach in which the service is used anonymously by users such as web service. Lately the development of open grid services which promises to solve the issues of data flow and data steering problems of large datasets. The clear challenge for visualization and grid

computing communities is to agree on standard architecture in which the visualization will be supported in the grid environment. On the other hand, most of the existing grid visualization applications are agreed to use similar components in the visualization architecture, but they didn't have a clear agreement on the sequence to follow in the visualization process. This results the development of several approaches specifically for remote visualization of large datasets. Some methods implemented as client sever architecture which is not suitable for large number of users in the distributed environment. Others follow the methods of distributing the components of visualization pipeline on the grid and implement it as problem solving environment. The result is that they mostly end-up with limited time solutions for specific problems. In addition to that, the different network bandwidths and different display devices results output suitable for one device such desktop and not suitable for others such as thin clients. This paper investigates the challenges of transferring large datasets between the visualization pipeline components and the issue of displaying large and complex datasets on thin clients. The paper begins with review of the existing visualization system. We pay particular attention to grid enabled visualization systems. Section 2 of this paper review some of the grid enabled visualization systems and projects. A review of grid computing concept is described in and its possible implementation in the proposed architecture is also described. Section 3 describes the new mirrored visualization architecture as an Architecture and solution on analyzing complex data using grid for remote devices and the internal components of the visualization pipeline. Section 4 is describing the proposed transmission Protocols Section 5 is the discussion of the methodology implementation and contribution. Finally the conclusion.

### 2- Background and related work

The scientific data visualization was sparked by landmark NSF report 'Visualization in Scientific Computing' by McCormick *et al* [13]. The introduced visualization concept was based on breaking down the dataflow of the visualization process to smaller distributed process. The smaller process can be placed on distributed location which interconnected by network to form a modular visualization.

Each part can contribute as an independent modular to form the over all the visualization process. However, the existing grid enabled visualization systems are in the direction of translating the existing dataflow concept presented by Haber and McNabb [11] as described in Figure [1].

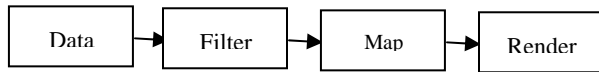


Figure [1] Haber-McNabb Visualization Pipeline

The existing visualization systems such as VTK William et.al, [22], IBM Data Explorer, AVS Express Upson et.al [18], OpenGL VizServer and IRIS Explorer Walton et.al [21] are generally available today and used to visualize a wide verity of large volume of data including medical data. For example a system such as VTK is making use of wide range of visualization algorithms and VTK supports parallelism through the use of threads. However these visualization systems generally designed to work in single high capabilities hardware machine. Despite the fact that VTK designed in object oriented, but during the design of these system there was no consideration to be supported in the grid environment. Many of currently on going projects are in a direction to extend the capabilities of these visualization systems. For instance gViz project is an on going project attempting to extend IRIS Explorer. The possible integration into the grid should be based on the design of internal components of these systems. The challenge now is in providing a flexible and effective mechanism to support remote access to the resources. Current implementations of grid enabled visualization are often tied to expensive hardware and powerful graphic support. In addition to that, the different network bandwidth and different output devices between the rendering location and presentation location produces output suitable for one device and not suitable for another. Additionally, the existing visualization systems are often making assumption on the available resources 'render local and render remote methodologies'. The following are some of Grid enabled visualization applications and projects.

RAVE Walker et.al [20] is a grid enabled visualization system that reacts and responds to available heterogeneous resources. RAVE implements techniques to make use of both remote and local resource according to the participating machines from High capabilities machines to Small PDA's. The gViz project Wood et.al [22] is another grid enabled visualization application. The idea was to incorporate the grid in the internal components of the IRIS Explorer Walton et.al [21].

The E-Demand Charters et.al [4] is another grid enabled visualization project focusing on the use of Grid services to support stereoscopic visualization in a distributed environment. The E-demand application considered as PSE "problem solving environment" on the grid .OGSA Foster et.al. [10] presents each model as an entity. Multi rendering

services can be deployed to form a collaborative environment.

The SuperVise Osborne et.al [15] is another grid implementation. In SuperVise Project the phases of visualization pipeline such as filtering and geometry transformation distributed across the grid. When the user select data SuperVise selects the appropriate resources and form the visualization pipeline.

The Distributed Visualization System Mahovsky et.al [14] is visualization application uses frameless buffer for rendering to distribute the pixel images between several machines. Each machine receives subset of the pixels to render it and submit the rendered part to create the full image, but each machine must have a copy of the full image.

Some Other visualization applications don't relay totally on software in their implementations for instance Visapult Bethel et.al [2] is a visualization framework with the ability to render a huge amount of datasets (of the order of 1-5 Tb). Visapult uses parallel rendering hardware to carryout the high speed rendering processes. Using Cactus Allen et.al [1] the data are distributed amongst many parallel nodes for volume rendering, the rendered subset 2D image sent to the client for local rendering.

Engel\_vis Engel et.al [8] is another application combines Local and Remote Visualization Techniques for Interactive volume rendering in medical applications. The application was implemented using java, java 2D and java 3D based on the client and communicating with server implemented using C++ and OpenInventor. The methodology followed is to load the datasets from the client side. Clients send the data sets to slicing tool. The slicing tool inspects slices in axial, coronal directions, and transfers the volume data to the server application. The server is stand-alone application utilizes 3D texture mapping hardware. Renders images off-screen and sends back compressed images to the clients. The methodology presented can not work well for clients with limited capabilities for local rendering and geometry transformation.

However, most of the mentioned grid enabled visualization applications are well structured and designed to solve specific problem. Some of them are following the same visualization techniques despites the difference in the internal architectures and specifications. Some of the applications provide the participating machines with no ability to do the rendering processes such as COVISE. Others applications assume that the participating machines support the rendering resources such as OpenGL VizServer 3.1. In addition, the other new implementation is trying to solve the rendering and resources support such as RAVE visualization application. RAVE is trying to develop a mechanism to figure out which machines support the rendering tasks and which machines have limited support to the rendering.

### 3.1 - Grid Visualization pipeline

### 3.2 - Visualization Pipeline Stages

Filter Service is responsible for taking the data from data service and convert the data if required. The filter service implements data compression techniques to reduce the size of the large files. Data calculation is required for registering the amount of the data to assign the suitable render service.

Render Service is responsible for taking the geometric data produced by mapping services and render them to produce images for transmission to the client. Render service is connecting to the visualization agent on the grid. The render service receives request from the visualization agent to block the data if necessary or the increase the amount of data. This process acts according to the network bandwidth.

Figure 4 Enabled Visualization Framework Design

### 3.2 - Visualization agent

```

graph LR
    subgraph Pipeline [Visualization Pipeline]
        MS[Mapping Service  
Isosurface Event Capture]
        RS1[Render Service  
CORBA / Java IDL Event Reader]
        RS2[Render Service  
Java 3D API Render Service]
        AG[Agent  
CORBA / Java IDL Event Generator  
XML Message Writer/Reader]
        MEM[Agent Memory  
Read / write]
        D[Display  
Application Parameters  
Menu Event  
Mouse Event]
    end
    MS --> RS1
    RS1 <--> AG
    AG --> RS2
    AG --> MEM
    MEM --> RS2
    RS2 --> D

```

Figure 5 Show the Interaction between the visualization Pipeline and the visualization Agent

### 3.4- The proposed mirrored visualization architecture

Our mirrored visualization concept is an attempt to solve the issue of limited resources problem at the thin clients. The clients with limited memory and desk storage are often excluded when we deal with complex datasets, despite the usefulness of common use in current technological advances. We noticed that in order for us to provide real-time interactive visualization for remote users. Specifically users with thin clients whom wish to have a clear insight at their data with the available machine. We should not think to transfer the entire data sets. We should find away to create life sample of the visualization at the client side as mirrored view. With the ability to transfer the changes made to the remote life sample.

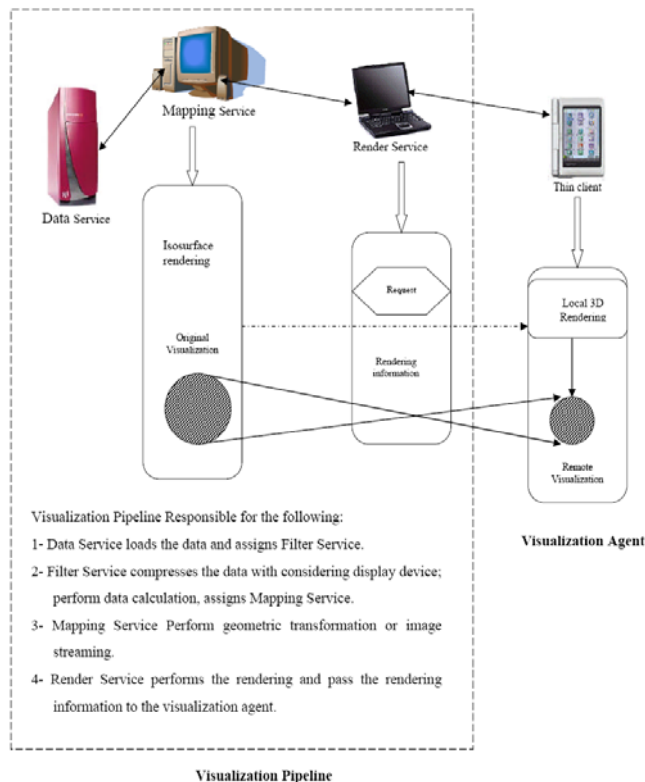


Figure 6 architecture of mirrored visualization

The mirrored visualization as in Figure [6] concept is the ability to transfer the parameters running on the visualization pipeline specifically on the mapping and render services. The mapping service sends data containing information of the isosurface of the object. At this point the visualization agents running on thin client will receive this information as XML messages to transform this information for simple local rendering using JAVA 3D API capabilities. The visualization pipeline will have to go through the entire visualization process. Except that this approach will have

two phases of rendering. However the visualization agent will receive information on how the data being rendered on the visualization pipeline side. The amount of time needed for this approach is limited to the network latency and the amount of information sent and received by the visualization agent. In addition to the time needed to have local rendering on thin client machine. The proposed approach will make use of special JAVA packages designed and developed for thin client machine.

### 4- Transmission Protocols

The proposed grid visualization application will not be restricted to one single approach to deal with visualization. The architecture will use two different methods to bring the visualization to the remote small size device. First method will use extensive xml messages to transfer the visualization information to the visualization agent. The exchanged information contains the isosurface values about the drawn surface of the object. The second method the visualization agent uses XML envelopes to inform the visualization pipeline on the amount of data required and according to the network bandwidth available.

For resource allocating and discovering the pipeline uses OGSA Services to find the available rendering services. The components of the visualization pipeline promoted using Web Service Definition Language (WSDL). WSDL can be registered with a UDDI server, in This case it is easy to integrate or replace any of the visualization pipeline components.

### 5- Discussion

The new grid enabled visualization architecture is based on two main contributions. First part in grid visualization pipeline .The second part is the technique of data steering mechanism. The steering in our case is the way of altering the isosurface and ordering it to redraw according to the passed parameters from the user. On the other hand, the automatic data blocking mechanism is based on the available resources in the client machine. At the Mapping service stage the mirrored visualization is performed. The automatic data blocking mechanism will monitor the process of data transformation from the source to geometric format, then sends the data resulted from the monitoring process to the requester to form life sample of the visualization. However, the grid enabled visualization system associated with the issue of transferring the entire data sets between the components of the pipeline. Since the XML will be used extensively as transporting mechanism the architecture will take full advantages of xml characteristics .One of the important XML features is XML tells you what type of data you have, not how to display the data, because of the tag identify information and break up the data into parts. the other important advantages of using xml is it's ability to

describe the data at several levels and it's ability to process large XML documents without consuming large memory .

## 6- Conclusion

The remote visualization of medical datasets with support of grid environment is presented as an architecture solution. The main contribution of the described architecture is to allow flexible integration of thin clients into the grid. The proposed visualization pipeline will be supported with several mechanisms such as automatic data blocking to allow the users to have accurate control of large data sets in real-time with modest resources available at the thin client machine. The presented architecture will allow the scientist to have additional access methods to their datasets. In addition to that, the scientist should be able to use any device as long as it's connected to the network. On the other hand, the presented visualization pipeline will work well not only for medical data sets, but with additional optimization rendering algorithms to support other type of datasets.

## 7- References

- [1] Allen .G, Benger. W, Goodale. T, Hege H.-C, Lanfermann . G , Merzky . A, Radke. T , Seidel .E, Shalf .J (2000): The Cactus Code: A Problem Solving Environment for the Grid. In Proceedings of the Ninth International Symposium on High Performance Distributed Computing (HPDC'00), pages 253–262. IEEE, August 2000
- [2] Bethel .W, Tierney. Brian, Lee . J, Gunter .D, Lau S (2000): Visapult Using High-Speed WANs and Network Data Caches to Enable Remote and Distributed Visualization, 2000 IEEE.
- [3] Brodlie, K.W., Duce, D.A., Gallop, J.R., Walton, J.P.R.B. and Wood, J.D. 2004. Distributed and Collaborative Visualization, Computer Graphics Forum
- [4] Charters, S., Holliman, N.S. and Munro, M. 2003. Visualization in e-Demand: Grid Service Architecture for Stereoscopic Visualization, Proceedings of UK e-Science Second All Hands Meeting.
- [5] Cristiano .K. D'Agostino. D, (2002): Using VTK for Isosurface Extraction in a Distributed Environment", Technical Report 8 - 2002, IMATI - CNR, 2002.
- [6] Daniela G S, (2003): Interactive Visualization of Biomedical Data. Master thesis Universiteit van Amsterdam, 2003
- [7] Dare Obasanjo 2003: Understanding XML. Microsoft Corporation July 2003 <http://msdn.microsoft.com/webservices/webservices/understanding/xmlfundamentals>
- [8] Engel K Sommer .O, Ernst C, Ertl T. (2000): Remote 3D Visualization using Image- Streaming Techniques. 2000
- [9] Engel K. H. B. P. T. K. E.T. E. (2000): Combining Local and Remote Visualization Techniques for Interactive Volume Rendering in Medical Applications. 2000
- [10] Foster, C. Kesselman, Nick .K. M., Tuecke .S (2002): The Physiology of the Grid: An Open Grid Services Architecture for Distributed Systems Integration. Technical report, Globus, February 2002.
- [11] Haber, R.B. and McNabb, D.A. 1990. Visualization Idioms: A Conceptual Model for Scientific Visualization Systems. In: Visualization in Scientific Computing, Shriver, B., Neilson, G.M., and Rosenblum, L.J., Eds., IEEE Computer Society Press, 74-93.
- [12] Giraldi .G, Porto . G, Fontes V, Dutra . M, Schulze . B (2005): Grid Computing and Scientific Visualization. 2005
- [13] McCormick B. H., DeFanti T. A., Brown M. D. (1987), "Visualization in Scientific Computing", Computer Graphics 21 1-14.
- [14] Mahovsky .J, Benedicenti. L (2003): Architecture for Java-Based Real-Time Distributed Visualization. IEEE Transactions on Visualization and Computer Graphics, 9(4):570 – 579, October December 2003.
- [15] Osborne .J, Wright .H, (2003) SuperVise: Using Grid Tools to Support Visualization. In Proceedings of the Fifth International Conference on Parallel Processing and Applied Mathematics (PPAM 2003), [16] Stuart M. Charters, Nicolas S. Holliman and Malcolm Munro: Distributing Stereoscopic Scientific Visualization across the Grid. 2004
- [17] Stuttgart (2000). COVISE Features. <http://www.hlrs.de/organization/vis/covise/features/>, HLRS, September 2000.
- [18] VTK. VTK—the Visualization Toolkit. <http://www.vtk.org/>, 2005
- [19] Upson, C., Faulhaber, T., Kamins, D., Schlegel, D., Laidlaw, D., Vroom, J., Gurwitz, R. and van Dam, A. 1989. The Application Visualization System: a Computational Environment for Scientific Visualization, IEEE Computer Graphics and Applications 9, 4, 30- 42.
- [20] Walker D. W. , Grimstead .I (2004): Resource aware visualization environment. <http://www.wesc.ac.uk/projects/rave/2004>
- [21] Walton, J.P.R.B. (2004). NAG's IRIS Explorer. In: Visualization Handbook, Johnson, C.R. and Hansen, C.D., Eds., Academic Press (in press). Available at [http://www.nag.co.uk/doc/TechRep/Pdf/tr2\\_03.pdf](http://www.nag.co.uk/doc/TechRep/Pdf/tr2_03.pdf)
- [22] William J. Schroeder, Kenneth M. Martin, and William E. Lorensen (1996): The Design and Implementation of an Object-Oriented Toolkit for 3D Graphics and Visualization
- [23] Wood. J, Brodlie, K., J. Walton. (2003) gViz – visualization and steering for the grid. In Proceedings of the UK All Hands Meeting 2003, <http://www.nesc.ac.uk/events/ahm2003/AHMCD/pdf/030.pdf> , <http://www.visualization.leeds.ac.uk/gViz>.