Using open source and commercial visualization packages for analysis and visualization of large simulation dataset

Simon Su, Werner Benger, William Sherman, Eliot Feibush, Curtis Hillegas
Princeton University, Louisiana State University, Indiana University, Princeton Plasma Physics Laboratory, Princeton University
simonsu@princeton.edu, werner@cct.lsu.edu, shermanw@indiana.edu, efeibush@pppl.gov, curt@princeton.edu

Abstract

Scientists today are generating more data than ever. In any high performance computing center, analyzing and visualizing large amounts of simulation data are becoming more and more challenging. Therefore, high performance computing centers will not only need to manage their high performance computational resources, but also develop state of the art data analysis and visualization capabilities. Simulation data will also need to be generated and visualized on the same storage hardware because transferring the simulated data after the simulation for analysis and visualization is no longer feasible due to the size of the data generated.

The scientific visualization community has been developing visualization software with the capability to read, analyze, and generate visualization from large simulation data. Numerous open source and commercial visualization software packages are available for use by the scientists for data analysis and visualization purposes. However, there is no single visualization package that can satisfy all the visualization requirements. Most visualization centers will utilize a combination of open source and commercial visualization packages in their visualization projects. In addition, visualization centers will also develop their own customized visualization module for their project specific visualization requirements that are not supported by the standard visualization packages.

1. Introduction

Scientific data visualization is a crucial part of any small and large modeling and simulation research. The simulation data generated will need to be analyzed to verify the accuracy of the model. Especially in this era of petascale and exascale computing, simulation data are only going to get larger and larger[1]. Therefore, ultra-scale visualization is a crucial part of any high performance computing center.

The private sector and United States federal government have been developing and funding visualization software toolkits that have the capability to analyze and visualize large simulation datasets. The software toolkits provide the domain scientists the capability to visualize large dataset at an interactive rate and at different stages of the modeling and simulation research. When the visualization pipeline is established, domain scientists will be able to visualize their data within minutes of the completion of their simulation on the supercomputer. Some simulation algorithms will generate output in stages and the domain scientists will be able to use the established visualization pipeline to start analyzing and verifying the accuracy of the model before the completion of the simulation. They can stop the simulation if the results are not what they would expect to find instead of wasting their resource allocation on the supercomputer. The scientists can then fine tune their model and re-run the simulation. When the simulation is completed, domain scientists can also use the same visualization pipeline to create comprehensive visualization of their dataset that will enable them to present their research to their peers and to the general public.

In addition to the visualization of scientific data, the ability to interact in real-time and in three-dimension (3D) with the 3D simulation data during the visualization process is also crucial to the process of analyzing and understanding the dataset[2]. Many visualization packages like VisIT[3], VISH[4], Paraview[5], and AVS Express[6] have the ability to plot and create visualization from the simulation data in the data format that they can load. However, only a handful of visualization packages will support fully immersive display and 3D interactions with the dataset displayed. AVS Express provides an additional module to support immersive display and 3D interactions with the dataset. Paraview on the other hand, has a limited support in those capabilities. Another way to accomplish immersive display and 3D interactions in scientific data visualization is to develop data visualization applications using a visualization software toolkit like VTK[7] together with software toolkits that can support immersive display and 3D interactions like FreeVR[8] and VR Juggler[9].

In section 2, we will describe the visualization hardware setup of the Visualization Center. Sections 3 and 4 will describe the open source and commercial toolkits used in the Visualization Center respectively. Section 5 will describe a unique visualization toolkit that is freely available for research and non military usage. We will discuss the advantages of using both open source and commercial software in section 6 and we will conclude in the next section.

2. Hardware Environment

The Visualization Center has a single wall rear projection display using Sony SRX-S110 projector with 10,000 lumens and 4096 x 2160 native resolution which translates to over 8.8 million pixels on the screen. Figure 1 illustrates the Sony SRX-S110 projector. That resolution is more than 4 times the amount of pixels on an office monitor at 1920x1080 resolution.

Figure 1. Sony SRX-S110

The screen is 120" tall and 216" wide with OptiTrack[10] optical tracking to support 3D interaction within the visualization application. Figure 2 and Figure 3 illustrates the Elevation View and the Plan View of the design of the visualization system.

The projector is driven by an Intel-based graphics workstation with Nvidia Quadro Plex 2200 D2. The 4 DVI-D connections from Quadro Plex 2200 D2 are being connected to the 4 DVI-D inputs on the Sony SRX-S110 projector. Each DVI-D connection is configured at the 2048 x 1080 resolutions, which combines to provide the projector with 4096 x 2160 resolutions input. The graphics workstation has 8 CPU cores and 24 gigabytes of system memory. In addition, a fiber connection card is also used to provide dedicated data connection to the GPFS file system server which it is a common data storage for the research community. The graphics workstation will run both Microsoft Windows and PUIAS Linux operating systems.

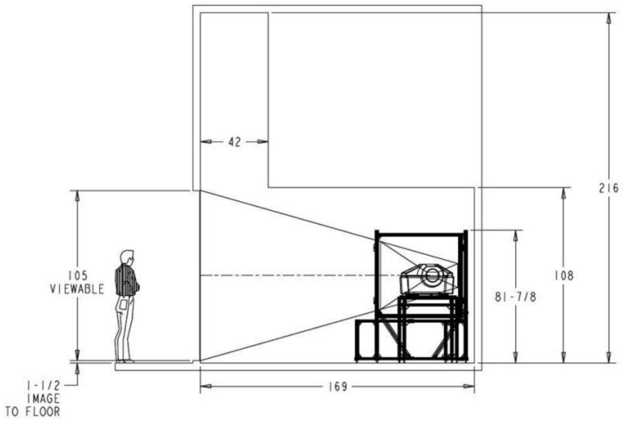

Figure 2. Elevation View

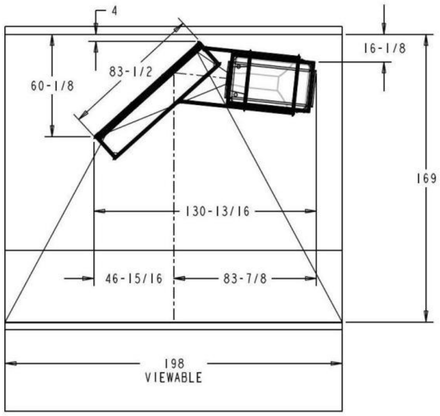

Figure 3. Plan View

3. Open Source Toolkit in Visualization and Virtual Reality

Data visualization is a highly customized process that depends largely on the type of data and visualization required. Therefore, the use of open source visualization software will allow the software to be freely customized to address individual data visualization needs. Although the access to the source code will allow the user to customize the visualization software, a good software design will also enable the user to add new visualization features and load new data formats with the software without getting too deep into the core of the visualization package. This will reduce the complexity of the understanding needed expand on the visualization software by a new user.

Although commercial software vendors offer paid support in using their commercial visualization software, government funded open source visualization software development projects can also provide excellent support through the user group mailing list. Many of the users of the open source visualization software fall under similar category of user and they are using the visualization software to visualize similar dataset. Many of the issues faced by a new user may have already been addressed by someone on the mailing list. Therefore, both the developers and the users of the visualization software in general will be able to help address the issue faced by a new user. This free support model works well where there is a large user base for the visualization software.

The following open source visualization software projects and 3D interactive and immersive software projects are good examples of software toolkits with good support from the developers and the user community.

3.1. VisIT

VisIt[3] is visualization software that can be used to visualize large scale simulation datasets. It enables the user to perform general post-processing and visualization of extremely massive datasets in parallel by utilizing the processing power of a supercomputer. It is using the processing power of a supercomputer in a way similar to the how the complex modeling and simulation process runs on the supercomputer to generate the massive datasets. Furthermore, VisIt also provides the user with a simplified access to the supercomputer's massive computational power by decoupling the graphics component and the computational component of the complex visualization software. The most powerful feature of VisIt is to allow the user to run the graphical component of VisIt on the graphics workstation in the user's office while running the computational processing component of VisIt on a remote supercomputer. The user will use the graphical user interface running on the local graphics workstation to start a visualization plot that will initiate the computational process on the remote supercomputer and gather all the processed data to combine them into the final data visualization plot on the local graphics workstation. This enables the user to process and visualize massive datasets that would otherwise be impossible on the local graphics workstation in the user's office. The targeted use cases of VisIt include data exploration, comparative analysis, visual debugging, quantitative analysis, and presentation graphics.

Like any large and complex software toolkit, VisIt's core software functionalities depend on several other software toolkits. VisIt depends on the Qt widget library for its user interface, the Python programming language for a command line interpreter, and the Visualization ToolKit (VTK) library for its data modeling and visualization algorithms. The use of existing toolkits allow the core development team of VisIt to focus their efforts on parallelization for large data sets, user interface design, implementation of custom data analysis routines, visualization of non-standard data models like adaptive refinement meshes and mixed materials zones, and creation of a robust overall product.

VisIt runs on various platforms of operating systems including Microsoft Windows, Apple Macintosh, and many UNIX variants, including AIX, IRIX, Solaris, Tru64, and, of course, Linux, including ports for SGI's Altix, Cray's XT4, and many commodity clusters.

VisIt's basic design is a client-server model, where the server is parallelized. The client-server aspect allows for effective visualization in a remote setting, while the parallelization of the server allows for the largest data sets to be processed at a reasonably interactively rate to support interactive visualization which is crucial in the visual debugging process. The client-server model also allows the user to mix VisIt processes running on various operating systems that allows for the use of a non-homogenous hardware platform. The user can run VisIt's graphical user interface on a Microsoft Windows machine and connect to a parallel computational processes running on supercomputer running on Linux operating system.

VisIt's user community has been using VisIt to visualize many large data sets, including a two hundred sixteen billion data point structured grid, a one billion point particle simulation, and curvilinear, unstructured, and AMR meshes with hundreds of millions to billions of elements. At the Visualization Center, we have been using visit to process 5 gigabytes of simulation data which would otherwise be impossible to visualize given the hardware resources we have on a single machine.

3.2. Paraview

ParaView[5] is an open source data analysis and visualization application that runs on various platforms of operating systems including Microsoft Windows, Apple Macintosh, and several Linux variants. ParaView users can use different filters provided and effortlessly build visualizations to analyze their data using qualitative and quantitative techniques. ParaView supports both 3D interactive and scripted processing of a large dataset. ParaView takes advantage of the distributed memory computing resources to analyze extremely large datasets.

ParaView is also a scalable visualization application that runs on a wide variety of hardware architecture. This ranges from a single-processor desktop machine or laptop machine, to a multi-processor shared-memory supercomputers or clusters of computers.

ParaView can process a wide variety of data formats including structured and unstructured, time varying and static datasets. ParaView takes advantage of the Visualization ToolKit (VTK) to provide a comprehensive suite of visualization algorithms. However, ParaView is also designed so its visualization algorithms can be easily extensible so that new algorithms can be easily added to enhance the visualization capability of the toolkit.

At the Visualization Center, ParaView is used to visualize engineering and atmospheric simulation data where the numerical data can be easily presented and analyzed by the domain scientists.

3.3. FreeVR

FreeVR is an open source virtual reality (VR) library that provides the developer with all the functionalities needed to create a complex VR application. It has been designed to work with a wide variety of input and output VR hardware that are commonly used in VR systems. VR applications developed with FreeVR library can easily be run on the existing virtual reality hardware facilities, as well as newly established VR hardware systems.

FreeVR uses a configuration file that provides an overall specification of the VR hardware system running the VR applications. The configuration file, which is unique for each different VR hardware system, provides a layer of abstraction that hides the complexity of accessing different complex VR devices from the application developer. The VR application developed will be able to run on any VR hardware system that has the correct FreeVR configuration file without modifying any part of the application source code.

FreeVR library has very little dependencies and compiles on any Linux platform with minimal effort. Like many other VR integration library, FreeVR only manages the rendering window and the developer will have to use VTK, OpenGL or other scene graph management toolkit to render graphics into the rendering window managed by FreeVR. The currently available FreeVR library runs on a multi-processor shared-memory system and a cluster capable version of FreeVR is under intense development and will be available soon as an open source software library. FreeVR library is used to develop architectural walkthrough applications at the Visualization Center.

3.5. VR Juggler

The name of the toolkit, VR Juggler [9], implies that the toolkit juggled various components together to provide a complete functionality of a typical VR library. It is a VR toolkit developed and supported by a team of researchers at Virtual Reality Application Center at Iowa State University.

VR Juggler toolkit consists of several different components including Gadgeteer, Juggler Configuration and Control Library (JCCL), Tweek, Sonix, CORBA, VRP, OpenAL/AudioWorks, and OpenGL that sit on top of the operating system layer. The Gadgeteer component provides device management functionality for VR Juggler that enables access to local or remote I/O devices. JCCL component provides the ability to configure and monitor application performance. Tweek is the distributed model-view-controller implementation within VR Juggler. The Sonix component provides the library with simple sound-abstraction support.

The latest VR Juggler library provides support for both multi-pipe and cluster-based rendering systems. VR Juggler also supports different VR system configurations by using XML based configuration files. This is transparent to the application layer by creating a uniform interface for the VR application developer to access the hardware layer of the different VR systems. The toolkit also provides a Java based VR Juggler configuration tool to assist the users in the creation of the complex XML based configuration file.

Similar to any complex software toolkit, VR Juggler depends on a number of software libraries, including CppDom, GMTL, Java Developer Kit, omniORB, OpenAL, OpenALUT, Audiere, VRPN, and Boost to provide different functionality in VR Juggler library. During compilation, the VR Juggler build process also relies on the Flagpoll software library for setting the correct compiler and linker flags. Compilation of VR application developed VR Juggler relies on Doozer software package for many of the compile flags settings.

VR Juggler used an object-based methodology where application developers overloaded virtual functions in the main class. Application developers have to write an application class that overloads pre-defined functions and passes the object to VR Juggler kernel, which is responsible for managing and controlling all the resources of the VR application across the system. VR Juggler does not provide rendering or scene graph functionality, and application developers use VTK library or OpenGL to render graphics into the rendering window managed by VR Juggler. VR Juggler library is used to develop architecture walkthrough and multimedia 3D applications at the Visualization Center.

4. Commercial Toolkit in Visualization and Virtual Reality

Research groups that have limited expertise in visualization are more likely to use commercially available visualization toolkit. One of the key advantages of using commercial visualization software is to have the option to purchase support that can be budgeted into the project proposal. In general, commercially supported visualization toolkits also have a more structured support with training classes and more expert users that can be hired to work on the project.

The Visualization Center has also made available a few commercial visualization software packages that are commonly used by existing user base of the center. The choices of visualization software provided by the center are determined mainly by the requests of the users of the center. Some users have preferred visualization software that they would like to be using in the hardware environment of the Visualization Center.

4.1. AVS/Express

AVS/Express provides a comprehensive and versatile data visualization tool that can be used by both application developers and end-users to visualize their datasets. It enables the user to perform rapid data analysis and rich visualization techniques via a complex set of graphical user interface.

AVS/Express provides powerful visualization for the users across multiple domains including science, business, engineering, medicine, telecommunications and environmental research. The Visualization Center is using AVS/Express for scientific data visualization.

4.2. ArcGIS

ArcGIS is a collection of geographic information systems (GIS) software products that provides the users with spatial analysis, data management, and mapping capabilities. The Visualization Center is using ArcGIS for maps and geographical features visualization.

5. VISH - Freely available visualization toolkit

VISH is a scientific visualization software that provides a platform for bridging the gap between algorithm development and accessibility of new visualization techniques to the application scientists. The framework enables development of new visualization algorithms that are not bound to their runtime environment which helps to facilitate the integration of the newly developed visualization algorithm into the domain scientist's daily work flow.

The developers of VISH designed the software framework to allow the development and implementation of visualization algorithms with independence from a specific application. This will provide the developer a framework to test their new visualization algorithms and also enable them to share their new visualization module among other applications and their peers. VISH framework also has minimal overhead on dependencies and external components which allow a new user to compile and use the framework without a steep learning curve. The minimal overhead and dependencies will also allow for easy deployment to end users. Many components of VISH can also be used independently of the framework which increases the usefulness of the components outside the framework.

VISH software framework acts like an "operating system for visualization algorithms" with a minimal set of abstract API serving as a framework for development and allows sharing of plugins throughout applications, which provide implementations of the same interfaces. VISH has a vision of becoming a "visualization shell" ("shell" in the sense of a "structural work" or "skin") and it is not an application by itself. It is an implementation of the infrastructure necessary for scientific visualization. It has a collection of interfaces in the form of libraries which application developers can use to implement their visualization algorithm. The algorithms implemented will just see the VISH API and remain application independent.

In the Visualization Center, VISH framework is used to support visualization projects that require some of the new and experimental visualization algorithms that have not yet been released in other open source or commercial visualization packages.

6. Up-front economic advantages and increased productivity achieved

The Visualization Center is utilizing a suite of freely and commercially available visualization packages to support multi-disciplinary visualization projects. The use of open source toolkits has really enhanced the visualization capability of the center in addition to the commercial visualization software. It is important to be able to leverage on the advantages of different toolkits to address our visualization challenges. Many of the visualization challenges are more easily addressed by open source toolkit compared to the commercial toolkit with the accessibility of the application source code and the support of the software's large user community base that worked on similar issues that are willing the share their experiences. It is without a doubt that the use of open source visualization software has increased the productivity of the visualization projects at the center.

In addition to the usability advantages, the center has also greatly benefited economically from the use of both freely and commercially available software toolkits by significantly reducing the initial setup and acquisition cost of the software suite that is needed to support the visualization projects of the center. The use of freely available toolkits shaved tens of thousands of dollars from the center's initial software budget and provided the users of the center a broader range of software toolkits to use in their visualization projects. Table 1 summarizes the software toolkits used in the Visualization Center.

Table 1. Software features summary

Software	Significant features
VisIT	Large data visualization
Paraview	Large data visualization
FreeVR	VR integration library
VR Juggler	VR integration library
AVS Express	Large data visualization
ArcGIS	Geographical data visualization
IDL/ENVI	Image processing, data visualization

The availability of different software toolkits at the center also encourages collaborations from other research institutions running the same toolkit. The visualization application developed by other visualization centers can easily be port into the software and hardware environment of the center if the application is developed on the same software framework.

7. Conclusion

The Visualization Center provides high-end visualization hardware and software that will help facilitate scientific and non-scientific discovery across multiple disciplines. The goal of the center is to provide unified visualization resources and support that would otherwise be unavailable or too expensive to maintain to the individual researcher at their respective research laboratories or department.

The visualization center will keep adding more open source and commercial visualization software toolkits into the list of software toolkits available for use at the Visualization Center in the future. The use of both free and commercial software packages has significantly reduced the economic requirement of running the center without sacrificing technical ability of the Visualization Center.

8. Acknowledgement

The authors would like to thank PICSciE and the university faculty members for their generous contributions to the funding of the initial setup cost of the Visualization Center.

9. References

[1] Kwan-Liu Ma, Chaoli Wang, Hongfeng Yu, Kenneth Moreland, Jian Huang, and Rob Ross Smith, “Next-Generation Visualization Technologies: Enabling Discoveries at Extreme Scale”, *SciDac Review*, IOP Publishing, http://www.scidacreview.org/, Spring 2009, Issue 12, pp. 12-21.

[2] Wen Qi, Jean-Bernard Martens, Robert van Liere, and Arjan Kok, “Reach the virtual environment: 3D tangible interaction with scientific data”, *Proceedings of the 17th Australia conference on Computer-Human Interaction*, Australia, November 23-25, 2005, pp. 1-10.

[3] Hank Childs, Eric S. Brugger, Kathleen S. Bonnell, Jeremy S Meredith, Mark Miller, Brad J Whitlock, and Nelson Max, “A Contract-Based System for Large Data Visualization”, *IEEE Visualization 2005*, Minneapolis, Minnesota, Proceedings of IEEE Visualization 2005, October 23-25, 2005, pp. 190-198.

[4] Werner Benger, Georg Ritter, and Ren\'e Heinz, “The Concepts of VISH”, *High-End Visualization Workshop*, Austria, Lehmanns Media-LOB.de, Berlin, June 18-21, 2007, pp. 26-39.

[5] Amy Squillacote, *The ParaView Guide*, Kitware Inc, February 2008.

[6] AVS Express, http://www.avs.com/, accessed June 27, 2009.

[7] Kitware Inc, “VTK Users’s Guide”, Kitware Inc, September 2006.

[8] William Sherman, http://www.freevr.org/, accessed June 29, 2009.

[9] A. Bierbaum, C. Just, P. Hartling, K. Meinert, A. Baker, and C. Cruz-Neira, “VR Juggler: A virtual platform for virtual reality application development”, *Proc. of Virtual Reality 2001*, Yokohama, IEEE Computer Society Press, Los Alamitos (2001), 2001, pp. 89-96.

[10] OptiTrack optical tracking system, http://www.naturalpoint.com/optitrack/products/tracking-tools/, accessed June 27, 2009