Supporting distributed visualization services for high performance science and engineering applications – A service provider perspective

Lakshmi Sastry*, Ronald Fowler, Srikanth Nagella and Jonathan Churchill
e-Science Centre,
Science & Technology Facilities Council,

Introduction

The Science & Technology Facilities Council is home to international Facilities such as the ISIS Neutron Spallation Source, Central Laser Facility and Diamond Light Source, the National Grid Service including national super computers, Tier1 data service for CERN particle physics experiment, the British Atmospheric data Centre and the British Oceanographic Data Centre at the Space Science and Technology department. Together these Facilities generate several Terabytes of data per month which needs to be handled, catalogued and provided access to. In addition, the scientists within STFC departments also develop complex simulations and undertake data analysis for their own experiments. Facilities also have strong ongoing collaborations with UK academic and commercial users through their involvement with Collaborative Computational Programme, generating very large simulation datasets. There is thus the need to support high resolution data analysis using distributed compute, data and visualization resources. At the same time, these requirements offer the computational and visualization scientists within STFC unique opportunities to advocate the take up of advanced visualization techniques and toolkits in distributed high performance, high resolution hardware environment. It gives an opportunity to understand the requirements and usefulness of distributed visualization. Given this seemingly advantageous position, the STFC vizNET partners have been actively pursuing visualization awareness activities and services aimed at application holders of various scientific disciplines. These activities include holding workshops, hands-on tutorials, show case demonstrations and the setting up of hardware based visualization services with technical support. This report provides details of these activities, the outcomes, the status and some suggestions as to the way forward.

Workshops and tutorials

The take up of advanced visualization techniques within STFC scientists and their colleagues from the wider academia is quite limited despite decades of holding seminars and surgeries to create awareness of the state of the art. Visualization events generally tend to attract practitioners in the field and an occasional application domain expert. This is a serious issue limiting the more widespread use of advanced visualization tools. In order to address this deficit, more recently, we have begun an escalation of such events by holding show and tell “Other Peoples Business” to introduce exemplars from specific domains and then the tools behind the exemplars, advertising these events exclusively to scientists of various fields. In addition we have also begun holding one day workshops focussed on a single technology theme aimed at specific application domains, with practitioners and users sharing their knowledge and experience with the users, with significant amount of time set aside for extensive question and answer and discussion sessions. Two workshops sponsored under the vizNET banner are typical examples of these, one aimed primarily at computational fluid dynamists and those studying mechanics of movement and the other at Particle Physicists. These were aimed at STFC scientists and user communities at STFC Daresbury Laboratory and at the Harwell Science & Innovation Campuses respectively. The first was on advanced visualization tools and techniques for high performance applications in science and engineering. The workshop focussed on current best practice from a number of related subdomains, on the role of VizNET in support of application developers, and also future

978-0-7695-3622-4/09 $25.00 © 2009 IEEE
DOI 10.1109/CCGRID.2009.94

requirements for the visualization of data from large-scale simulations that generate datasets heading towards 10s of TB. Presentations included exemplars, algorithms for parallel visualization, rendering, high resolution displays and ways to combine these to achieve near real time data exploration for extremely large datasets.

The second was on Visualization and Data Mining for very large datasets (> 100MB). The predominantly particle physics research audience were introduced to state of the art visualization toolkits and techniques, hardware and distributed visualization using commodity graphics clusters to support real time analysis of LHCb, ATLAS and CMS experimental datasets from CERN. There exemplars focus was to demonstrate data mining and parallel coordinates based visualization by international speakers.

A third strand of activity was to hold a series of hands on two days tutorial on computational steering, introducing the users the use of gViz and RealityGrid steering libraries. The users were introduced to the use of compute resources of the UK National Grid Service, digital certificate based authorization and authentication in using distributed resources as a precursor to using computational steering for complex application.

The considerable effort spent in getting application scientists to attend the events have generated a slightly better response than before and a small set of users who have never explored advanced visualization tools have now joined the Experimental National Visualization Service to explore what the technology can offer their scientific productivity. In the UK, this gives us an idea of effort required to bring in new user groups into the visualization community. On the other hand, despite considerable publicity, free registration and offer of follow-on support as well as holding the computational steering schools at STFC site and at Manchester University which are close or local to large pool of computational scientists brought little dividend with participants being mainly new members of the groups that used such software already. In addition, the feedback and the experience also demonstrated that these new technologies and software are far from offering ease of use, installation, documentation for the layers of client packages that are required all factors being formidable obstacles to take up.

In the next Section, the hardware, the software and support environment of the National Visualization Service associated with the National Grid Service and the use that is being made of by the initial set of new users.

Experimental National Visualization Service (ENVS)

STFC is attempting to estimate the requirements for scalable visualization within the UK academic research community through collaborations with other networks such as the EPSRC funded e-Research South to set up and reach out to new groups. We have set up a dedicated commodity visualization cluster (eVis) as an Affiliate member of NGS and reach across to new users.

eVis is a 17 node, 64-bit commodity cluster with dual processors on each node with Linux operating system. The cluster is built using commodity components and open source software stack. It comprises a 17Node/34 processor infiniband connected cluster, each node equipped with an NVIDIA FX4500 high end graphics card. These nodes can work together to render very large images in real time using the Tungsten Graphics Chromium software. The cluster provides multi user remote access to these resources in batch mode via globus or interactively via gsissh/virtualGL/X11 stack. The cluster provides access to serial and parallel rendering software such as NAG Iris Explorer, ParaView and VTK.

NGS Affiliation status was formally achieved on 22nd January 2009. This requires that the system passes the rigorous ‘Inca’ acceptance grid testing suite for a week without error, which was achieved in September 2008. Affiliation also requires an entry in the Grid Operations Centre Database (GOCDB https://goc.gridops.org/) to be maintained with planned downtimes etc and a contact on the NGS Helpdesk so that user support queries can be formally tracked. In addition the cluster is integrated into the eScience monitoring framework using GANGLIA software to monitor usage and NAGIOS for alerts. The service has its own web site and user registration page. Hardware maintenance was recently extended and software licenses updated, including the acquisition of NAG mathematical and statistical and data mining libraries. A custom 'visbsub' user scripts hides some of the complexities of visualization data port forwarding over the encrypted ssh/gsissh links from the cluster to the client PC.

Visualization software stack

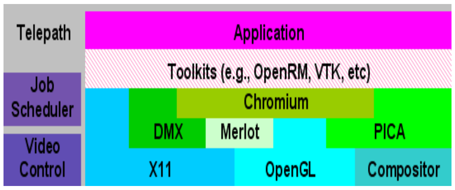

Figure 1 Architectural overview of Chromium with DMX

Chromium and VirtualGL are the two software packages that are used for this purpose. Chromium is an open source library for interactive parallel rendering of a single image using several rendering machines in parallel. There are several techniques in parallel rendering that will enable rendering large amounts of data that cannot be rendered in real time using a single machine. Chromium is a completely extensible architecture and enables the user to implement parallel rendering algorithms with ease. Most of the OpenGL applications will run under chromium without any modification to the application or recompiling it. To enable some of the features in chromium the user needs to modify the application using Chromium API and recompile the application. The API is simple to use and minimal changes are required in the application code.

Chromium supports parallel rendering in Distributed Multi-Threaded X Server mode [Figure 1]. This enables rendering on high resolution tiled displays to visualize extremely large datasets in high resolution which is very useful for medical and engineering applications.

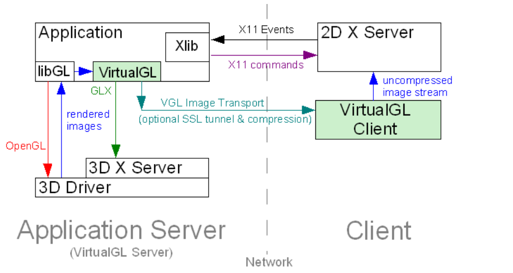

Figure 2 Overview of VGL

VirtualGL is an open source package which gives any UNIX or Linux remote display software the ability to run 3D applications with full hardware acceleration [Figure 2]. VirtualGL can also be used in conjunction with remote display software such as VNC to provide 3D hardware accelerated rendering for OpenGL applications. VirtualGL is very useful in providing remote display to thin clients which lack 3D hardware acceleration. VirtualGL can be used in two modes, namely, VGL Image Transport and X11 Image Transport. In the VGL Image Transport mode, the user desktop X Server sends the user inputs to the X Server at the remote server for interactive rendering of the image and the rendered image is transported to the user desktop via a network channel other than X Server link. In X11 Image Transport mode the VirtualGL talks to a VNC like remote display software and renders the 3D applications and sends the entire remote desktop to the user desktop. VGL Image Transport mode is the chosen default mode with the visualization cluster as it gives the flexibility of multiple users sharing the cluster at the same time.

ParaView was selected as the main visualization toolkit as it is an open source product for general purpose visualization tool which is under current maintenance and development and has been so since 1994. It supports parallel architectures such as that of the Vis Cluster.

ParaView has large number of active developers, 59 in the past 12 months and has good user ratings on several web sites. It builds on the widely used VTK software. VisIT is another parallel visualization toolkit that performs similar functions to ParaView on distributed architectures which may also be installed on the Vis Cluster in the future. The HPCx consortium chose ParaView over VisIT due to its greater flexibility in parallel configurations and its ease of reading raw binary data files (http://www.hpcx.ac.uk/research/hpc/technical_reports/HPCxTR0803.pdf). Binary versions of ParaView are available but to build a parallel version requires downloading and building from source with the appropriate MPI library. This was a little time consuming as it requires the use of CMake and careful selection of options according to the version of MPI that is used. Building the latest version of ParaView (3.4), requires the latest QT libraries which were not available for the installed cluster operating system (RHEL4) and also had to be built from source. At the moment the parallel build of ParaView 3.4 on the Vis Cluster using ScaliMPI gives run time errors. ParaView 2.6.1 does build and run successfully with our MPI and experience reported here is for this version of the parallel software.

ParaView 2.6 can be used without VirtualGL, to clients including Windows XP with the Exceed Hummingbird (V11) X11 server, though with lower performance. In one tested case over a very slow network (2 Mbits) the non-VirtualGL performance was about half that of the VGL connection. Using ParaView 3.4.0 (single processor version) to the same client without VirtualGL did not work at all due to Exceed (V11) not accepting certain GLX requests. With VGL the Windows/Exceed client worked correctly. Hence the use of VGL is not just about performance and is required in some cases. The performance of ParaView over a remote connection will of course be dominated by the available bandwidth. VirtualGL requires a connection of at least 100Mbit/sec to give good frame rates (>20fps) but can offer fair performance down to 10 or even 5Mbit/sec connections. However ParaView cannot generate frames of very complex geometry this fast, except when it is used to build an animation that is then displayed.

If we compare the performance of a local copy of ParaView 2.6.1 on a high performance laptop machine (2GB memory, Nvidia Graphics) to that of a remote copy of ParaView 2.6.1 on the Vis Cluster over a very slow (2Mbit broadband) connection, using a small isosurface of some 55000 elements, it is not surprising that the local ParaView performs much better. The local version reports of the order of 3E6 polygons per second while the remote system manages only a few thousand polygons per second and takes about 20 seconds to render a frame. This performance is particularly poor even allowing for the low connectivity. Using the single processor version of ParaView 3.4.0 on the remote system gives significantly better performance with images rendered within one second. While still much slower than local performance on a high powered system, there are significant advantages in moving to the latest release for performance over very slow networks. The main advantage of using ParaView remotely, as opposed to using it on the local laptop, is the ability to deal with much larger data sets. The above laptop will perform very well viewing an isosurface of about 9.2E6 polygons. However adding one more isosurface of similar size, or trying to do further operations on the existing surface, causes ParaView to run out of memory and die. Each node of the Vis Cluster has 8Gbytes of memory so even using just one node allows dealing with significantly more data than is possible with a 32bit version of Windows. Using 16 nodes of the cluster gives access to over 120Gbytes of main memory. In addition ParaView is able to utilize the high performance graphics cards that are available on each node of the cluster. This allows much better performance than would be possible with off-screen rendering.

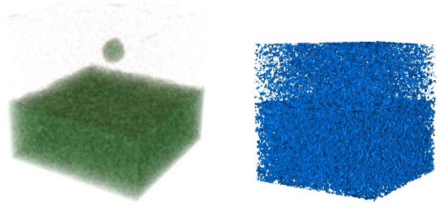

Figure 3 user data real time visualization

ParaView can also be used to drive a display for high resolution images. The Vis Cluster supports a six panel tiled display and ParaView can be used to drive this using the distributed X11-server DMX running on one of the display nodes. The ParaView client is then run on the same node as DMX and gathers data from the server nodes. Because ParaView exploits the graphics processors of server nodes, it is not possible to use the nodes attached to the display as server nodes since the display image will be corrupted by the server nodes. This can be overcome by using the slower off-screen rendering option.

One problem seen with ParaView 2.6.1 in using the display wall is that images will be rendered at lower than expected resolution when using a full screen window across all the display panels. This problem is not seen is the (slower) off screen rendering is used. It may be related to the default settings of the individual graphics cards.

The remote performance of ParaView can be improved slightly by adjusting the Level-Of-Detail (LOD) threshold and the LOD resolution (under View->3D image properties). The best values depend on the network connection and the size of the data being viewed. For example, over a very slow network with PV2.6.1, the rendering of a reduced resolution image when rotating a mesh does not help at all and it is best to only show the full image. The user needs to perform trials as to the best settings for their connection and data

eVis also offer Iris Explorer, a commercial highend visualization builder. It is able to benefit from the large address space available on a single node of the Vis Cluster. However modules are not in general able to exploit the parallel architecture by default. A user can write parallel modules which exploit the Vis Cluster architecture for computationally intensive parts of the visualization pipeline and return the results to the Explorer render module (e.g. http://www.nag.co.uk/visual/IE/iecbb/Posters/articles/ASCI.html). Efforts have been made to run Iris Explorer using VirtualGL to speed up the performance over a network. These have not been successful to date and it may be necessary to recompile Iris Explorer source to make the use of OpenGL compatible with that expected for VGL. VGL works by intercepting the applications normal calls to the GLX library and makes certain assumptions about the way in which an application will do this. While this works in many cases, it does not do so for Iris Explorer.

Figures 3 above are of volume density and isosurface visualization of atomistic molecular dynamic simulation in fluid flow in *real time* from one of the ENVS user. Simulation runs on NGS with each dataset is about 86MB. Data provided by Dan Lussier and Dr. Yiannis Ventikos of Engineering Science Department, University of Oxford.

Initial feedback and conclusion

The eVis system is being used with users undertaking atmospheric and oceanography based climate model analysis, typically using five dimensional data sets with a range of parameters with each data set being about 200MB. It is used for studying molecular dynamics, vortex shredding around aircraft wings and similar. Initial experience show that the users require intense one to one support to navigate through the first step of acquiring a digital certificate and getting to register with the system to all the way through the compute clusters to the visualization node, display and render services, the use of the visualization toolkits with VirtualGL. In addition, the users also require technical help to read in their output data from simulations into the visualization toolkits, introduction to new techniques available through these systems. The end results so far have been positive but the support required per user is about a person over a period of three months, more if the user is a complete novice. Setting up and running these systems are equally labour intensive. A full production service should ensure that not just hardware and administrative staff but a pool of visualization support service personnel are also in place.

Acknowledgement

The authors wish to thank STFC, EPSRC and JISC for support