An All-Linux Access Grid Implementation

Chris Willing, Nicole Bordes, Bernard Pailthorpe
VisLab and School of Physics
The University of Sydney, NSW 2006, Australia
chris@vislab.usyd.edu.au

Abstract

Experience with construction of several Access Grid nodes in Australia has led to the first implementations of a Linux-only Access Grid node. It has also become apparent that, with judicious hardware selection, it is possible to deploy some high performance visualisation applications in an access grid. In our construction method the machines comprising the Access Grid node may be viewed as a subset of a larger (computational) cluster facilitated by the use of one operating system. The node may also be used as a PC-cluster driven Powerwall. Together these features lead to possibilities of new levels of collaborative activity between remote groups of users and suggest movement toward greater integration between the access and computational grids.

Keywords: access grid, grid computing, cluster.

1 Introduction

Recent advanced computing research has spawned the concept of the Grid [1]: the global set of networking, computing, data and software resources inter-connected by powerful networks. Argonne National Laboratory (ANL) has also developed the concept of the Access Grid, a collaborative environment that allows group-to-group working over the Internet using multicast [2, 3]. These nodes provide rich portals to grid resources and are more than just video conferencing facilities. The Access Grid has great potential for Australian researchers, as it breaks down the "tyranny of distance" between Australia and the rest of the world, as well as within Australia itself.

Starting from the ANL concept, we have modified the design to an all-Linux implementation, rather than a mixture of Linux and Microsoft Windows as used elsewhere. The single operating system enhances the capabilities of the Access Grid, to include computing and advanced graphics functions. Since the inception of the first Australian Access Grid in August 2001, we have built two Access Grid nodes and assisted in the construction of additional nodes throughout Australia, all of which have been implemented using our Linux system. In this paper we present the current state of the Australian Access Grid project and an overview of the Linux-only construction method.

2 Review of the ANL Design

The Access Grid, as designed by the ANL researchers, requires a large area display of sufficient capacity to display several video streams from different sites. This is in general achieved with three projectors. The conferencing side of the Access Grid requires several cameras, at least one showing the speaker and one the audience, several microphones and speakers.

A typical Access Grid node server consists of four PC class machines. The audio and video capture machines run the Linux operating system while the display machine and the control computers run Windows-NT or -2000. The software tools include a multicast beacon and viewer, distributed PowerPoint tools, a MUD client and the University College of London Mbone tools, *vic* and *rat* [4]. The Virtual Venue software developed at ANL coordinates and administers an Access Grid session with different geographically distributed participants. The network relies on multicast connectivity however it is possible to use a unicast bridge for sites with no multicast capability. The hardware cost of setting up an Access Grid varies depending on the quality of the hardware component but a typical cost was around US$45,000.

3 Australian Design

The motivation to alter the ANL design and develop our own version arose from our need to more efficiently utilise available bandwidth in Australia and to reduce costs. The first departure from the ANL model was the use of single operating system, Linux, in each of the display, video and audio machines versus a mix of Linux and Windows in the ANL design. Luckily the Access Grid applications running under Windows already had Linux versions so no porting was required; however two issues needed to be resolved. First was the problem of how to produce, in the display machine, a single desktop for the multiple VGA outputs driving the three display projectors and the operator's monitor. Second was the issue of communications between the three machines to synchronise their video, audio and MUD applications.

We used version 4 of Xfree86, the Linux X windowing system, which contained a stable implementation of a feature named Xinerama. Xinerama enables an arbitrary number of VGA devices to be combined into a single desktop. The arrangement of VGA devices within the desktop is specified in a configuration file. In any machine, the number of outputs arranged in this way is limited only by the number of VGA cards in the machine and the number of outputs available from each card. Our first implementation used a Matrox G200 PCI bus card with four outputs. While this worked quite well, we felt that responsiveness suffered when a large number of video streams were being displayed, presumably due to the limited bandwidth of the PCI bus itself. Improved results were obtained then we used a Matrox G450 (later

G550) AGP bus card together with a G450 PCI bus card, each of these cards having dual outputs to produce the required total of four VGA outputs.

The event space of the Access Grid consists of a number of virtual venues, each containing a number of virtual meeting rooms. An Access Grid node is navigated through a virtual venue to different virtual rooms using a web browser running on the node's display machine. At each of these moves, the hosting Virtual Venues server returns technical details about the new room back to the navigating client node. Most important of these details are the multicast group addresses for the audio and video streams in the room, which are sent in a Session Description Protocol (SDP) conformant message. This is used by the display machine to restart the video reception application with the correct multicast group address. A separate message from the server specifies room movement for a text based MUD application. The information contained in the SDP message must also be sent from the display machine to the audio and video machines so that the audio and video capture applications can be restarted and send data to the correct multicast group address.

In a traditional Access Grid node the information exchange between the node machines is mediated using CORBA. While this is convenient in a group of machines using different operating systems, it is unnecessarily complicated for a relatively simple message exchange in an homogenous group of machines. Furthermore, the particular CORBA implementation, which is distributed as part of ANL's Access Grid software release has possible copyright implications for non-academic use. Without a mechanism for message exchange between the node machines it was necessary to run the audio, video and mud applications by hand from the command line. While this can be adequate for an experienced operator, it can be quite cumbersome and, from a user perspective, it is quite different from the ANL model with which we want to remain compatible. We therefore developed our own virtual venues daemon, *vvd*, to process SDP message exchanges between node machines. It is a very lightweight daemon which runs on each machine in the node and whose behaviour is controlled in a simple text file. The configuration file specifies the machine(s) from which it will accept directions (MASTER machines), machine(s) to which it will send directions (SLAVE machines) and a count of the number of instances of any named application to run when a new SDP message is received. Typically, the display machine's configuration file will list the audio and video machines as its SLAVEs and will nominate one instance of *vic* (for video display) and one instance of *tkMOO-lite* (MUD client) to run; the video machine's configuration file will list the display machine as a MASTER and nominate three instances of *vic* (for 3 camera streams to capture); the audio machine will also list the display machine as a MASTER and nominate a single instance of *rat* (for audio).

The sequence of events for a node of machines configured in the manner described above is as follows: when a new virtual room is selected by the browser on the display machine, the mud and multicast session messages returned by the venues server are directed to *vvd*, which restarts vic with the SDP's multicast session information as parameters. Additionally, *tkMOO-lite* changes rooms in the MUD environment. The SDP message is then sent by *vvd* on the display machine to *vvd* on the nominated audio and video SLAVE machines. On the video machine, *vvd* receives the new session information and restarts 3 instances of *vic*, while on the audio machine, a single instance of *rat* is restarted. In all these cases, restarting an application is intended to mean starting it from scratch if it is not already running.

By this stage of development our Linux-only Access Grid nodes were fully operational. They operate in multipurpose labs where there is at least one other Windows-2000 machine available to run distributed PowerPoint sessions. The distributed PowerPoint application itself, *dppt*, is a Java application, which distributes page-turn events to cooperating hosts running local instances of Microsoft PowerPoint under Windows. Since PowerPoint can be run using a Linux implementation of the Windows API, such as WINE or CrossOver Office [5, 6], we hope to use this mechanism together with the Java-based *dppt* to run completely Linux based sessions.

In an effort to make their programs as widely useful as possible, writers of video streaming applications have tried to be as network-friendly as possible. Typically the streams have low image size and are heavily compressed. In the Access Grid, the default video used meets International Telecom Union (ITU) H261 standard, which supports a maximum image resolution of 352x288 pixels. While this is often adequate, the quality is relatively poor. In anticipation of both higher capacity networks and requirements for video with higher quality, we have commenced trials of a video capture/playback board which implements on-board Motion JPEG compression/decompression at full broadcast image size and frame rate (768x576x25 for PAL, 720x480x30 for NTSC). That board is an LML33 from Linux Media Labs [7] used in conjunction with Mat Delco's RTPtv software [8], which supports both unicast and multicast streaming; the video quality is astonishingly good compared with *vic*'s standard H261 stream. One of the standard camera capture boards in the single machine Access Grid node at our campus lab has been replaced with an LML33 board. It captures the video output of a digital video editing system and streams it to a multicast group address. At our ATP lab we have added an LML33 board to the audio machine where it outputs the video stream received from the multicast group onto a television monitor. Although the image quality is excellent, the network bandwidth required is much greater - we are fortunate to have a gigabit network connecting our two locations. The high quality video is currently being streamed at 20Mbits/sec., compared with 300-400Kbits for our *vic* streams, but as general network capacities are improved, it is expected that video streams of this quality will become common.

3.1 Single Machine Access Grid Node

We are aware of the need for low-cost, lightweight Access Grid nodes, possibly having reduced functionality, and have experimented with such systems. VisLab operates two laboratories: one on the University campus and one off site at a Technology Park (ATP), three kilometres from the campus – these two labs are connected by switched gigabit Ethernet network running over a dedicated single-mode optical fibre. The primary Access Grid node is located in the ATP lab off site. In order to follow some Access Grid sessions, we developed an Access Grid node implemented in a single machine operating Linux. This node is fitted with a low-cost echo cancellation unit and uses one or two projectors, one microphone and three video inputs.

4 Enhancements to the Access Grid

4.1 Graphics

Both nodes are fitted with VGA graphics cards, which can support the traditional Access Grid layout and which also possess strong hardware OpenGL features to support graphics clustering using Chromium, formerly known as WireGL [9,10,11,12]. Chromium allows OpenGL applications to work on a tiled display and we have chosen Performer, OpenInventor and OpenDX as the applications to test in the cluster environment, having found them to run successfully in both our access grid nodes using their standard display mode with the new VGA card set. Thus our Access Grid node can be used as a PowerWall for large-scale visualisation.

4.2 High Resolution Video

Matthew Delco has described his RTPtv project for production-quality internet television [8]. It uses hardware MJPEG compressed capture and network transmission of PAL or NTSC video streams. Output may be either hardware assisted to an external TV monitor or, at reduced quality, directly to the desktop using the OpenMash version of *vic*. After some experimentation, this system was incorporated into both our access grids.

4.3 New Media Streams

The desire to incorporate new media streams fully into the Access Grid environment meant that the Virtual Venues server software needed changes. Such changes were made so that the Asia Pacific Access Grid (APAG) server was able to return multicast group addresses for media streams relating to RTPtv and MLB. MLB is Media Lecture Board, a shared electronic whiteboard program developed by the Universities of Mannheim and Freiburg [13, 14]. The modifications were submitted to ANL in October 2002 for possible inclusion in their general release version of the Virtual Venues software.

5 Collaborative Events and Activities

5.1 SC Global

Our first connection to a live Access Grid session occurred on August 29th, 2001. It was driven by our planned participation in SC-Global, a part of the ACM-IEEE SC2001 conference based in Denver [15]. Figure 1 shows Australian participation at VisLab on November 14, 2001 and Figure 2 shows the Access Grid's projected images. In the run-up to this event, numerous tests called 'nanocruises' were attended to iron out technical wrinkles, especially those related to our multicast connectivity. The Australian Academic & Research Network (AARNet) dedicated network connection between the Access Grid node and the newly installed Southern Cross cable between Sydney and Seattle. Network and router tuning resulted in a Sydney-Manchester (UK) latency of ~250 ms, approaching the speed-of-light limit of ~100 ms. This participation in SC Global represented the official launch of the Access Grid in Australia. As well as attending several sessions suitable for our time zone, we also hosted several sessions.

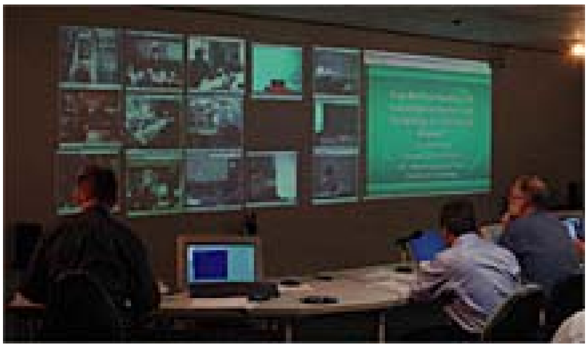

Figure 1: SC Global 2001 at VisLab

Figure 2: VisLab's Access Grid projected area (3840 x 1024 pixels)

5.2 Asia Pacific Access Grid

As the Access Grid environment is gaining momentum in the Asia Pacific, an Asia Pacific regional grouping was formed to promote Access Grid activities in the region. Kasetsart University in Thailand initiated an APAG mailing list and Sydney Vislab hosts the APAG web site, for exchange of access grid related news and ideas in this time zone [16]. In 2002 a Virtual Venues Server was brought on line [17]. Further testing, up to October 2002, indicates bandwidth availability and multicast issues as being the major obstacles to wider cooperation.

6 Conclusion: Status and future

The first of our Access Grid nodes used dual 1GHz processors in Tyan Tiger 230 motherboards with 1GB of RAM. We found these to be excellent until we started looking for VGA cards with good hardware OpenGL performance. Newer high performance graphics cards would not operate satisfactorily in the available AGP slot, so a newer motherboard also had to be found. The motherboard finally selected is a Soltek SL-85 ERV with 533MHz FSB, which includes an on-board sound chip and six PCI slots. It uses a single Intel P4 CPU operating at 2.26GHz. This enables a complete Access Grid node with sound, 3 cameras, gigabit network card and four VGA outputs (including high performance 3D) to be implemented in a single machine. The graphics cards used for the display machine are a single PNY Quadro4 550-XGL dual output AGP card and two PNY GeForce4 MX-420 single output PCI cards. This card set provides dramatically superior graphics performance at less cost than the Matrox G200 quad output PCI card of our first system. For the OpenGL applications we have targeted, the only limitation found so far is that hardware acceleration is only available in the two outputs from the AGP graphics card. This means that only two thirds of the projected portion of the desktop is hardware accelerated. To achieve hardware acceleration for all of the projected area our strategy is to use Chromium to tile the outputs from high performance graphics cards in each machine of a three-machine node. Figures 3 a-b show the transition from the standard Access Grid node display technique to a clustered technique. Since these AGP cards have dual output capability, there is an opportunity to run the entire hardware accelerated display in stereo.

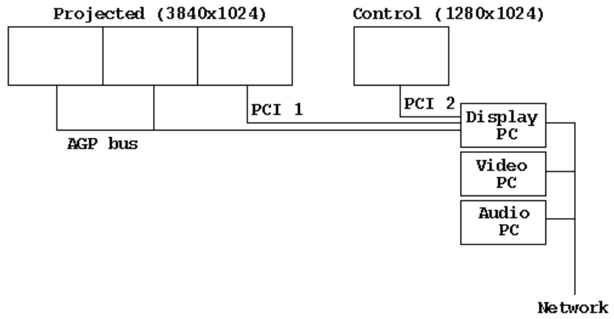

Figure 3a: Standard display. Of the three projected areas, two are supplied from AGP card and one from PCI card, and a second PCI card supplies the control monitor.

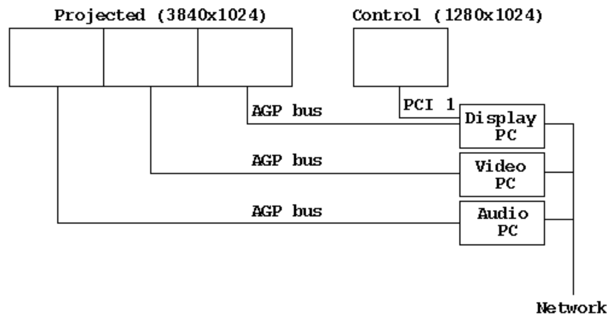

Figure 3b: Clustered display. Of the three projected areas, all are supplied from the AGP card of a different machine *i.e.* all three are display machines.

Tiled stereo displays have been previously implemented and it was BU's use of Performer, a 3D simulation package, in such an environment, which motivated our interest in this area. However we are not aware of any group, which has used such a display as the actual display for an Access Grid node. The AGAVE (Access Grid Augmented Virtual Environment) project at the University of Illinois Chicago [18] appears to use a separate stereo display, which is placed alongside an otherwise standard Access Grid node. Perhaps some of the AGAVE tools such as *Immersaview* could provide the basis for collaborative viewing and manipulation of common 3D model data, fully integrated into a tiled display Access Grid node. Other application specific tools of this type have been developed such as collaborative AVS [19, 20] and the Molecular Interactive Collaborative Environment (MICE) [21]. We intend to investigate the development of a more general event stream protocol to facilitate interaction between collaborating remote groups. Such a protocol should not be application specific. Instead, applications would invoke the event stream to carry events marked as relevant to that application. At remote sites, applications using the event stream would act on it in an application specific manner.

Although the number of Access Grid nodes in Australia is now beginning to increase, there have been very few other nodes in this time zone up till now. This means the machines have been sitting idle for long periods of time. It has occurred to us that this unused capacity could be applied to a small compute cluster. In fact the Access Grid machines could be combined with our original Access Grid machines (which were superseded when our new graphics card set required different motherboards). In such a situation, Access Grid node machines could be viewed as just machines within a larger cluster; machines that provide a video service or an audio service, just as other machines may provide a printing service, or a storage service. We intend to investigate the practicability of bringing Access Grid services within the umbrella of more generalized cluster and grid management environments.

7 Acknowledgements

This project was funded by the Australian Research Council.

8 References

[1] Forster,I & Kesselman,C (eds) “The Grid: Blueprint for a new Computing Infrastructure”, Morgan Kaufmann, 1999.

[2] L. Childers, T. L. Disz, M. Hereld, R. Hudson, I. Judson, R. Olson, M. E. Papka, J. Paris, and R. Stevens, "ActiveSpaces on the Grid: The Construction of Advanced Visualization and Interaction Environments," presented at Parallelldatorcentrum Kungl Tekniska Högskolan Seventh Annual Conference, Stockholm, Sweden, 1999.

[3] L. Childers, T. Disz, R. Olson, M. E. Papka, R. Stevens, and T. Udeshi, Access Grid: Immersive Group-to-Group Collaborative Visualization, In *Proceedings of the Fourth International Immersive Projection Technology Workshop*, June 19-20, 2000.

[4] http://www-mice.cs.ucl.ac.uk/multimedia/software/

[5] http://www.winehq.com/

[6] http://www.codeweavers.com/products/office/

[7] http://www.linuxmedialabs.com

[8] M.R. Delco, Production Quality Internet Television, In TR 2001-161, Berkeley Multimedia Research Center, August 2001. See http://bmrc.berkeley.edu/~delco/rtptv/

[9] G.Humphreys and P. Hanrahan, A Distributed Graphics System for Large Tiled Displays. *In IEEE Visualization 1999:* 215-223..

[10] G. Humphreys, I. Buck, M. Eldridge, and P. Hanrahan, Distributed Rendering for Scalable Displays. In *Supercomputing 2000 proceedings*, 2000.

[11] G. Humphreys, M. Eldridge, I. Buck, G. Stoll, M. Everett,and P. Hanrahan, WireGL: A Scalable Graphics System for Clusters. In *SIGGRAPH 2001*: 129-140.

[12] G. Humphreys, M. Houston, R. Ng, R. Frank, S. Ahern, P. Kirchner, and J. T. Klosowski, Chromium: A Stream Processing Framework for Interactive Rendering on Clusters of Workstations, In *ACM SIGGRAPH - Computer Graphics*, 21(3): 693-702 (2002).

[13] W. Geyer, J. Vogel, M. Mauve., An Efficient and Flexible Late Join Algorithm for Shared Whiteboards. In *Proc. of the Fifth IEEE International Symposium on Computers and Communication*s, ISCC'2000, pp. 404-409, Antibes, France, 2000.

[14] R. Weis, J. Vogel, W. Effelsberg, W. Geyer, S. Lucks, How to Make a Digital Whiteboard Secure - Using JAVA-Cards for Multimedia Applications. In *Proc. of 7th Workshop on Interactive Distributed Multimedia Systems and Telecommunication Services*, IDMS 2000, pp. 185-198, Enschede, Netherlands, October 2000.

[15] SC-Global, http://www.sc2001.org/shtml , Nov. 2001.

[16] http://www.ap-accessgrid.org (2001).

[17] http://venues.ap-accessgrid.org (2002).

[18] http://www.evl.uic.edu/cavern/agave/

[19] G. Johnson, Collaborative Visualization 101. In *ACM SIGGRAPH - Computer Graphics*, 32(2):8-11, May 1998.

[20] C. Natoli, http://www.vislab.usyd.edu.au/users/manuals/vis/avs5/collab/intro.html.

[21] P.E. Bourne, M. Gribskov, G. Johnson, J. Moreland, and H. Weissig, A Prototype Molecular Interactive Collaborative Environment (MICE). In *Pacific Symposium on Biocomputing* (1998) Ed. R Altman, K. Dunker, L.Hunter, and T. Klein pp.118-129.