Army Research Laboratory

Developing a High-Resolution Tile Display Based on Commodity Components

by Richard C. Angelini

ARL-TR-3640 **September 2005**

NOTICES

Disclaimers

Army Research Laboratory

Aberdeen Proving Ground, MD 21005-5067

ARL-TR-3640 **September 2005**

Developing a High-Resolution Tile Display Based on Commodity Components

Richard C. Angelini

Computational and Information Sciences Directorate, ARL

REPORT DOCUMENTATION PAGE	*Form Approved* *OMB No. 0704-0188*

1. REPORT DATE *(DD-MM-YYYY)* September 2005	**2. REPORT TYPE** Final	**3. DATES COVERED (From - To)** June 2004 – August 2004
4. TITLE AND SUBTITLE Developing a High-Resolution Tile Display Based on Commodity Components		**5a. CONTRACT NUMBER**
		5b. GRANT NUMBER
		5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S) Richard C. Angelini		**5d. PROJECT NUMBER** 4603CL
		5e. TASK NUMBER
		5f. WORK UNIT NUMBER
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) U.S. Army Research Laboratory ATTN: AMSRD-ARL-CI-HC Aberdeen Proving Ground, MD 21005-5067		**8. PERFORMING ORGANIZATION REPORT NUMBER** ARL-TR-3640
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)		**10. SPONSOR/MONITOR'S ACRONYM(S)**
		11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT

Approved for public release; distribution is unlimited.

13. SUPPLEMENTARY NOTES

14. ABSTRACT

The advent of powerful, low-cost desktop computers is changing the way computational scientists visualize the results of their calculations. Traditionally, only the least complex data sets could be analyzed at the local desktop, while large, time-dependent data sets required the use of high-end graphics servers to visualize the results. Today, we can analyze the results of very large computations that were previous unimaginable on the desktop, and in some cases, the graphics subsystem in new desktop workstations can outperform dedicated visualization servers. However, in many cases, scientists are now computing more than they can practically represent on their traditional desktop display monitor. The challenge presented is to effectively represent a high-performance computing-sized computational domain in a typical 1280 × 1024 display, or develop a high resolution, cost-effective display technology.

15. SUBJECT TERMS

commodity cluster, scientific visualization, tile display

16. SECURITY CLASSIFICATION OF:			**17. LIMITATION OF ABSTRACT**	**18. NUMBER OF PAGES**	**19a. NAME OF RESPONSIBLE PERSON** Richard C. Angelini
a. REPORT UNCLASSIFIED	**b. ABSTRACT** UNCLASSIFIED	**c. THIS PAGE** UNCLASSIFIED	UL	36	**19b. TELEPHONE NUMBER** (*Include area code*) 410-278-6266

Standard Form 298 (Rev. 8/98)
Prescribed by ANSI Std. Z39.18

Contents

List of Figures

List of Tables

Acknowledgments

The author thanks Dr. James Kohl and Jamison Daniel of Oak Ridge National Laboratory for their consultation on the design and engineering of a tile display system; Mr. Dennis Hash (ARL WMRD Materials Application Branch) for his assistance in the fabrication of components for the tile display wall; Mr. Kelly Kirk (ARL CISD High Performance Computing Division) for his assistance with Linux systems configuration and administration; and Mr. Chris Fahey and Mr. Kai Rassmussen (ARL MSRC Programming, Environment, and Training Summer Intern Program) for their assistance in assembly of the cluster and tile display.

INTENTIONALLY LEFT BLANK.

1. Introduction

Alternate display technologies are essential to the implementation of scientific visualization in the laboratory setting. High-performance computing (HPC) facilities, such as the Major Shared Resource Center (MSRC) located at the Army Research Laboratory (ARL), Aberdeen Proving Ground, MD, support computational projects that generate enormous amounts of data. These calculations typically generate tens of gigabytes to terabytes of data, and it is difficult for a researcher to interpret this vast amount of information without the extensive use of visualization techniques and technologies. There are numerous software tools available to analyze data, however, very large, time-dependent data sets require specialized scientific visualization software and hardware not typically found at the desktop. Visualization laboratories, collaboratoriums and virtual reality centers have been established at ARL and other top government and commercial laboratories for the sole purpose of providing an environment for researchers to view and interact with their data.

Until very recently, these visualization laboratories would have been very similarly equipped. Each would have had some sort of large, dedicated visualization server, such as a multiprocessor SGI Onyx2, with large memory, multiple graphics pipes, and large amounts of disk storage. The output from this system would then be displayed to a large screen via a stereoscopic projection system, such as single-screen Immersadesk (figure 1) or an immersive, multiprojection Cave or RAVE-II (figure 2).* At the time these displays were conceived and installed, this was a superior solution and state of the art. However, the cost associated with installing and maintaining such a system is quite substantial, and far beyond the reach of a departmental research staff. Such a display needed to be funded and supported at corporate levels, and required a significant amount of square footage to implement.

Today, however, the amount of computer and graphics power that can be placed on the desktop is staggering. Commodity computers and very low-cost graphics cards have demonstrated enormous growth in performance over the past few years, and the cost-performance ratio cannot be ignored when compared to the high-end visualization servers. Computational scientists can now work on very large computations on their desktop system that were previous unimaginable, and in some case, the graphics subsystem in new desktop workstations can outperform the dedicated visualization server. However, despite the availability of these resources, there will always be certain classes of computation problems that challenge both the hardware and software resources that are available and problems that ultimately require the unique capabilities of dedicated visualization hardware and the talents of the visualization team staff.

* Immersadesk, Cave and RAVE-II are all trademarks of Fakespace, Inc.

Figure 1. Immersadesk display.

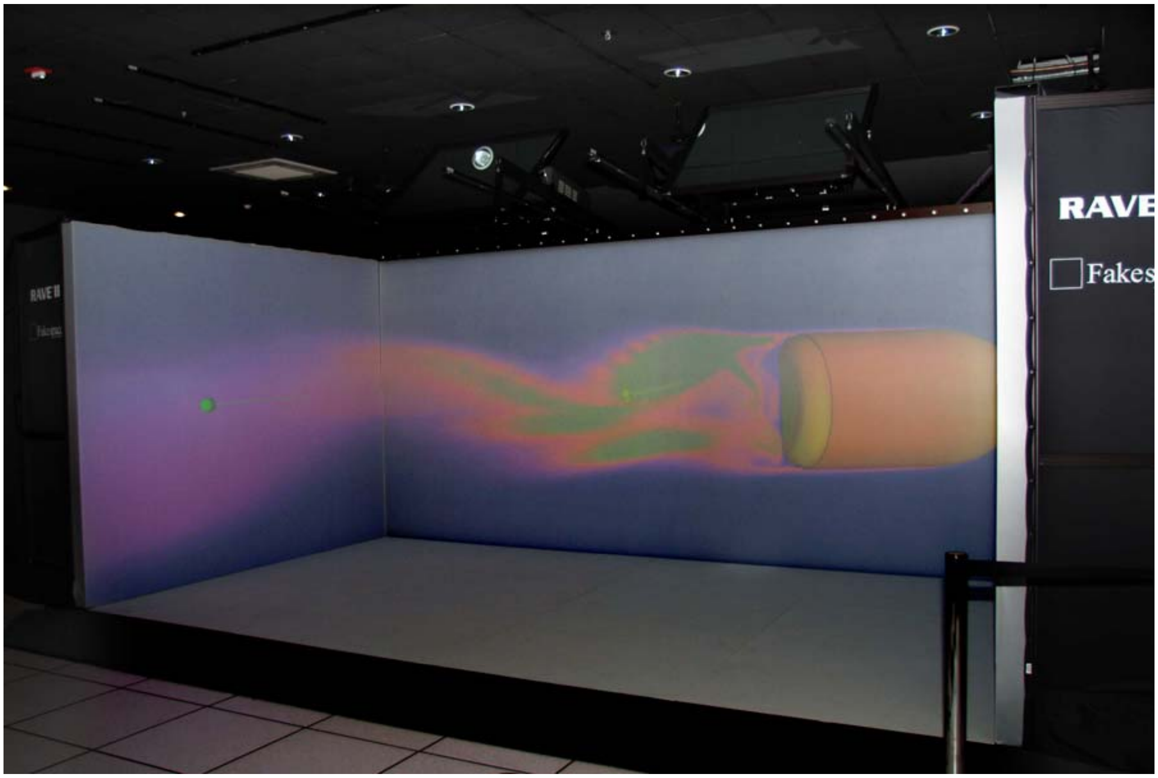


Figure 2. Rave-II display.

Projects related to the use of high-resolution displays and commodity computers have been established at a number of government laboratories and universities. The Department of Energy has several initiatives (*1–3*) under the Advanced Simulation and Computing Initiative (ASCI) to develop an infrastructure of Linux-based workstations with commodity graphics cards to provide the means to visualize their largest data sets. California Institute of Technology, Stanford, and Brown are among the list of universities who are partnering with government laboratories to develop and extend the capabilities of commodity cluster environments.

Scientists are most comfortable working in their own environment, and now in many instances have no need to go to a visualization laboratory to perform their routine data analysis and visualization. However, scientists are now computing problems larger than they can practically represent on their traditional desktop display monitor. The challenge presented is to effectively represent an HPC-sized computational domain in a typical 1280 × 1024 display, or develop a high resolution, cost-effective display technology. The risk of losing critical computed information exists if gigabytes of information are compressed to 1-million pixels of a traditional display. Therefore, a cost effective solution to display more pixels of data is required, and this solution needs to be easy to maintain, not take a large amount of floor space, and be cost effective to implement.

To address these needs, this report will focus on the methodology used to develop a very high-resolution tile display wall using cost effective commodity computers, standard flat-panel displays, and open source software. If successful, this display-wall technology can be implemented in critical work areas, allowing scientists to see more of their computation than they can at their desktop without going far from their desk, but at a cost which is practical.

2. Background

The Ballistic Research Laboratory (BRL), a predecessor organization to the current ARL, formed its' first Scientific Visualization team in 1991. Shortly thereafter, the first general-purpose visualization laboratory was established in Building 390, in conjunction with the Interior Ballistics Division of BRL.

In 1998, shortly after ARL was selected as an MSRC, a newer, larger visualization laboratory opened in Building 394 (replacing the Building 390 laboratory), creating a facility that was co-located with the visualization team staff. These initial facilities provided state-of-the-art visualization servers and graphics capabilities. At that time, most large data sets could not be viewed at the researchers' desktop, and researchers would routinely work in the visualization laboratory, taking advantage of the capabilities of the high-end graphics systems to view the results of their computations.

In the spring of 2002, ARL completed construction on its' first large scale visualization theater, known as the Collaboratorium (figure 3). Located at the Adelphi Laboratory Center (ALC), this facility houses a very high-end SGI visualization server that provides a stereoscopic video display on a 24-ft curved screen. The facility was designed to support the requirements of computation scientists wishing interact with their data, but it also serves as a large, shared resource where an audience of 30 or 40 people can view the same stereoscopic visualization and feel equally immersed in the experience. On the large curved screen, up to 2-million pixels can be displayed in stereo mode, and approximately 3-million pixels in standard display mode.

Figure 3. Collaboratorium at the Adelphi Laboratory Center.

In January of 2003, construction was completed on a new visualization center in Aberdeen to replace the facility that was developed in 1998. The Aberdeen Visualization Center (figure 4) houses a reconfigurable display system known as a Fakespace RAVE-II display wall. This display can be configured into a single wall 40 ft wide × 8 ft tall to support a presentation to an audience of 20 or 30 people. However, it was designed with two hinged "wing" displays, each of which is 10 ft wide × 8 ft tall. These wings can be closed to create a "cave"-like experience, displaying visualization on three side walls, plus the floor, supporting a single-person fully immersive experience. This reconfigurable display provides the ultimate in display technology, including head tracking and wand input interfaces, and a fully immersive stereoscopic "virtual reality" interface to scientific data. The RAVE-II can display approximately 7-million pixels in stereo or standard display mode.

Figure 4. Aberdeen Visualization Center.

Concurrent to the construction of the Aberdeen visualization center, powerful commodity desktop workstations based on standard PC technology became prevalent. Low-cost commodity graphics cards were showing signs of being able to outperforming the previously superior visualization server graphics pipes. In the years leading up to this time, it was clear that commodity computers were becoming faster, however, the user environment was unstable at best. The availability of open source, public domain operating systems such as Linux provided a desktop environment that researchers were familiar with, but there were concerns about network security, operational reliability was questionable, and corporate support was not available. In 2004, the company Red Hat announced the availability of a Linux operating system that was based on previously developed open-source standards, but was professionally supported. ARL adopted this operating system as a standard, and at this point, powerful Linux-based commodity desktop systems became a viable alternative to the proprietary Unix-based workstations such as those from SGI, SUN, and Hewlett-Packard.

In 2004, the scientific visualization team at ARL built a cluster of commodity computers to evaluate this technology as a practical alternative to traditional, shared-memory servers. We chose to implement the cluster using Opteron[*] processors from Advanced Micro Devices, Inc (AMD). This commodity processor was supported by Red Hat and still placed us on the leading edge of the technology curve. Our initial configuration placed five of these Linux-based, 64-bit

[*] Opteron is a trademark of Advanced Mico Devices, Inc.

commodity workstations on staff members desktops, connected to a dedicated gigabit ethernet network. Once it was clear that there was substantial software support for the implementation of a "virtual" cluster by tying these desktop systems together, we proceeded to the next step of building a small, dedicated cluster to serve as a transitional system as we moved toward more robust commodity computing technology. We chose to continue with Opteron-based processors for this transitional system to maintain compatibility with the desktop systems previously purchased.

Once the commodity cluster technology was selected, attention was focused on the details for developing the high-resolution display. There are two technical approaches to building a high-resolution display. The first alternative uses high-resolution projector technology (*4*) arranged in either a rear-projection or front-projection configuration. Projection displays (such as those already deployed in the APG and ALC visualization laboratories) can be prohibitively costly, and therefore, were not seriously considered as part of the commodity display solution. Tile displays use typical desktop monitor technology, and the delivery cost is significantly less than the projection-display solution. Of course, the tile display creates a visual grid due to the bezel edge around the flat-panel display which can be a distraction to the viewer. However, it is thought that once the end user has some experience with viewing data on the tile display, the visual grid will become less of a distraction.

The tile display itself would be made up of standard desktop flat-panel monitors, precisely wall-mounted in a 3 × 3 tile grid. Care was taken in selecting the flat-panel monitors in order to achieve the best possible results. Monitors with the smallest bezel (trim, also referred to as a mullion) would be used to minimize the distance between display frames. Open-source packages such as Distributed Multihead X (DMX), Blockbuster, and Chromium would also be used as a basis for demonstrating the viability of using a commodity cluster to visualize ARL MSRC data on a tile display.

3. Relative Performance of Desktop Workstations

It is generally recognized that modern commodity graphics cards provide exceptional performance, and may possibly outperform the traditional workhorse graphics systems offered by specialized vendors. Published benchmarks are often times measured under ideal conditions on complementary data sets and can therefore be misleading, so we decided to measure performance based on an actual application in use at ARL. The results presented in the following charts represent an effort to compare existing graphics workstations using a standard benchmark. The platforms tested included a wide variety of existing SGI desktop workstations and visualization servers, along with a number of different commodity-based workstations, both Linux and Windows. The benchmark run was a standard-sized data set using performance-monitoring tools available in Ensight.* Ensight is a commercial, multipurpose graphics package used on a variety of HPC-related applications, primarily in the area of computational structural mechanics and computational fluid dynamics.

The purpose of performing these benchmarks was to give a relative performance value to existing, common workstations used by computational scientists at ARL. This is not an absolute performance value as there are many different types of graphics applications and each has unique characteristics which determine overall performance. However, we hoped to find a rend in performance which would indicate the relative value of commodity machines versus the long-used SGI workstations.

The benchmark consists of running five tests from an Ensight command file and is designed to stress the system's graphics, memory and CPU.† The initial geometry contains 5,324 quads and 128,500 triangles which are duplicated six times, giving a total number of polygons in the test of 37,944 quads and 771,204 triangles. The test images are displayed in a 600 × 500 pixel window. These sizes are typical of data sets routinely visualized by our users.

The individual tests consist of the following:

1. Line-drawing test –360° rotation in 12° increments giving 30 screen refreshes. Generates a total of 73-million lines.

2. Shaded test – 360° rotation in 12° increments giving 30 screen refreshes. Generates 24-million polygons with each part being a single color.

3. Shaded test – same as test no. 2, but with parts colored on a per-vertex basis.

4. Shaded test – same as test no. 3, but rendered in immediate mode (as opposed to display-list mode in the previous tests).

5. Transparent test – same as test no. 4, but opacity is applied to the parts. This test stresses both the CPU and memory of the system in addition to the graphics. The scene is rotated 72° and generates five frames.

A composite score for each platform was computed by using the frame rate and weighing the importance of each of the tests. The composite score is calculated as follows:

$$C = 0.25*(30/T1) + 0.2*(30/T2) + 0.2*(30/T3) + 0.2*(30/T4) + 0.15*(5/T5). \qquad (1)$$

* Ensight and Ensight/Gold are registered trademarks of Computational Engineering, Inc.

† Description of the benchmark tests is paraphrased from the README file associated with the benchmark command files provided by Computational Engineering, Inc. (CEI), the manufacturer of Ensight.

The results of the benchmark study are presented in table 1. The SGI-based graphics technology was demonstrated to be significantly slower than the newer commodity-based desktop workstations in a real-world application that is used daily at the ARL MSRC. The graphs included below represent the results obtained. Figure 5 represents a comparison of all platforms for all five tests. Figure 6 compares the results of the best performing SGI vs. all of the tested commodity machines. Figure 7 graphs the results of the weighted composite score.

Table 1. Results of each of the five tests of the benchmark. Value is time (seconds).

Platform	1	2	3	4	5	Composite
SGI - O2	104	105	273	262	407	0.17598
SGI - Indigo2 High Impact	116	71	92	95	143	0.282782
SGI - Indigo2 Max Impact	78	46	85	56	147	0.409422
SGI - Onyx 2 - IR2 (Crab)	36	26	44	30	43	0.792908
SGI - Onyx 2 - IR2 (Clam)	33	16	20	29	65	1.120708
SGI - Onyx 3 - IR3 (Sponge)	18	13	16	18	43	1.60398
SGI - Octane V8	22.45	8.45	13.62	37.09	328.37	1.648716
SGI - Fuel V12	18.18	7.23	8.12	13.89	112.88	2.419943
Nvidia Quadro4 700 (Windows2K, 2.2 GHz)	4.59	5.96	7.25	11.32	37.17	4.018497
Fire-GL X1 (Windows 2K, 2.8 GHz)	3.23	5.625	7.968	8.625	29.15	4.863041
Radeon 9600 (Mac G5, 2 GHz)	3.36	4.76	4.9	15.53	50.53	5.118329
Nvidia FX3000 (64-bit Linux, 2.1 GHz)	3.33	4.74	6.34	6.29	21.53	5.453177
Nvidia FX1300 PCI-Express (64bit, 2.4 GHz)	3.3	3.56	4.63	5.69	26.09	6.337245
Fire-GL X2 (32-bit Linux, 1.4 GHz)	2.8	2.9	2.9	11.4	65.5	7.354269
Nvidia FX3400 PCI-Express (64bit, 2.4 GHz)	2.63	2.92	2.95	6.04	16.98	7.977951

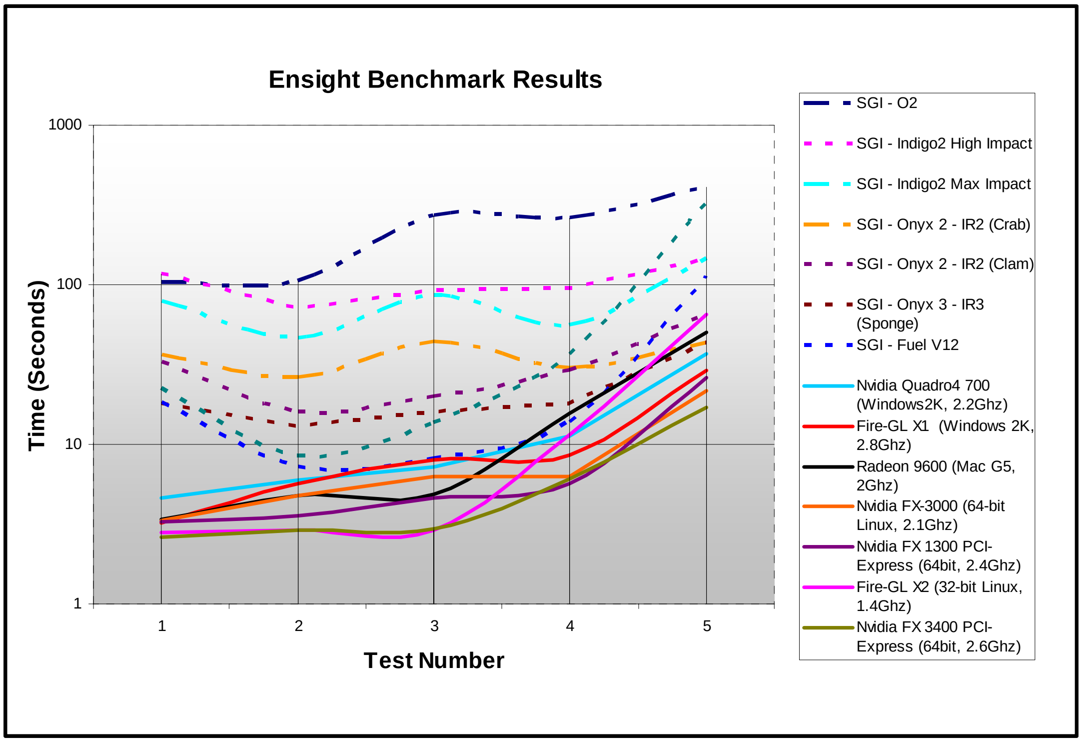


Figure 5. Benchmark results.

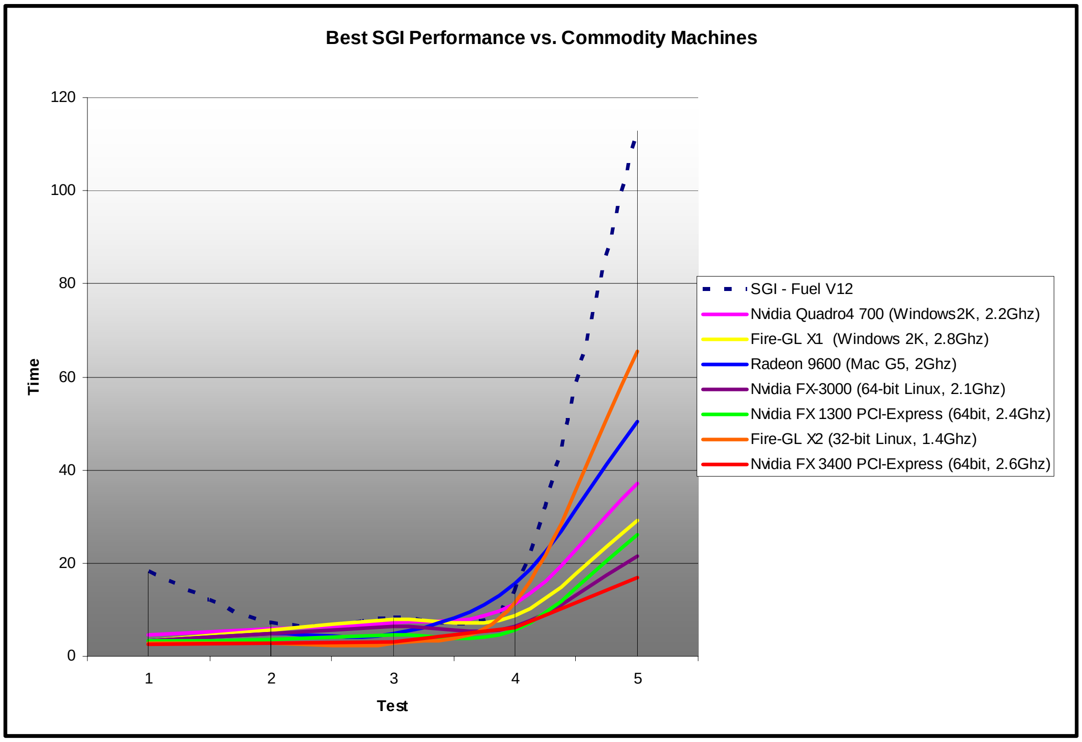


Figure 6. Best SGI performance vs. commodity workstations.

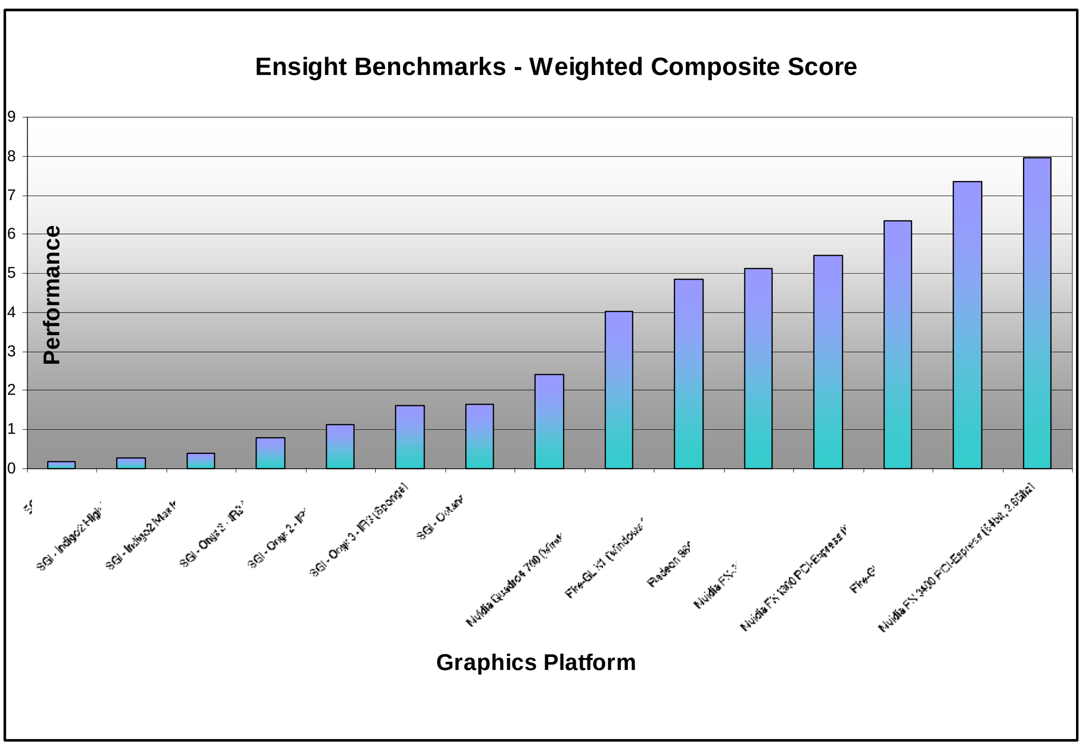


Figure 7. Weighted composite score.

4. Configuration

We believe that these results demonstrate that commodity-based workstations perform quite favorably compared to the traditional scientific workstation, thus justifying our initiative to base the high-resolution tile display on these workstations. Therefore, we determined that the commodity tile display would be driven by a 9-node Linux cluster. Each node consists of dual 64-bit Opteron processors, 4 gigabytes of system memory, a pair of 140-gigabyte disk drives, and an nVidia Quadro* FX-3000G graphics card. We had hoped to minimize the amount of rack space required to house the cluster by using ultra-slim “blade” systems. However, due to restrictions with the configuration of the graphics card with the accelerated graphics port, it was necessary to build each node of the cluster using traditional computer towers (figure 8) that are typically associated with desktop computer systems. The FX3000G graphics card offers dual digital video interface (DVI) output that was put to use in the configuration of the system. DVI-D cables are used to connect the output from the graphics card to the flat panel monitors. Also, a keyboard-video-mouse (KVM) switch was implemented, allowing all nine of the Linux workstations to share a single keyboard, mouse, and secondary monitor, providing convenient access by the end users to all of the systems (figure 9). The graphics driver associated with the FX3000G provides extensive capabilities to tune and configure performance. In this scenario, we configured to ports so that they were cloned, that is, the exact output was sent to each DVI interface, therefore displaying the same image on the flat panel monitor and through the KVM switch. (See appendix A for an example of this configuration file.)

The most complex issue was the design and construction of the tile display itself. It was important to identify a high-resolution, flat-panel monitor that had the small mullion to reduce the physical space between the displays on each monitor. The team was also concerned about the ability to mount the displays in such a way as to allow for minor adjustments so that the monitors lined up properly, minimizing any gaps and alignment issues. After reviewing design and mounting methodologies utilized by other groups who had built tiled displays, it was determined that a framework of extruded aluminum would be deployed. This solution was low cost, easily obtained and installed, and provided a sturdy mounting mechanism. This framework would be flush mounted against the wall (figure 10), and the monitor mounts would be secured to the aluminum frame. (See appendix B for a diagram of the monitor mounts.)

* Quadro is a registered trademark of nVidia Corporation.

Figure 8. Opteron-based commodity cluster used to drive tile display.

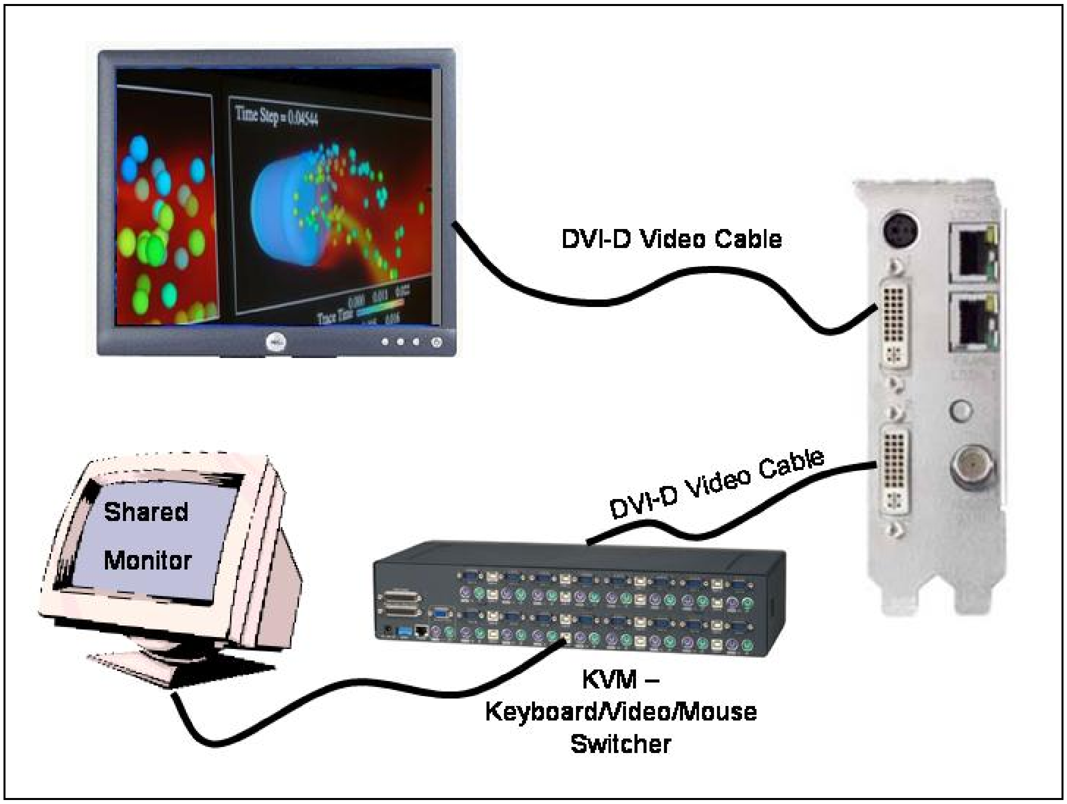


Figure 9. FX3000G Dual-DVI Interface configuration.

Figure 10. Extruded aluminum frame flush mounted to wall with monitors mounted.

It turned out to be very challenging to properly align the monitors. Despite the extensive efforts made to design a mounting mechanism that provided maximum adjustability, it was still very difficult to get a perfect alignment. Members of the ARL Weapons Material Branch, which includes a team of skilled carpenters and machinists, were brought in to brainstorm on ideas to resolve the alignment problem. A customized turnbuckle (figure 11) was designed and manufactured that would be mounted behind each monitor at the corners to provide a fine-tuning adjustment. The completed monitor layout can be seen in figure 12, and the final dressed configuration can be seen in figure 13.

Figure 11. Custom turnbuckle used to align monitors.

The final configuration provides a display of more than 20-million pixels, addressable using a number of applications. The commercial package Ensight/Gold has extensive capabilities to drive numerous types of advance displays, including tile displays and virtual reality environments, complete with head tracking and 6-degree-of-freedom input devices. Open sources packages, such as Paraview and Blockbuster, provide interfaces design specifically for support of a distributed environment required to compute and display in a tile-display environment. Each of these packages has been used to visualize data on the tile display. Paraview and Ensight provide a real-time interactive environment that allows the end users to manipulate the displayed geometry through rotation and translation, step through time dependent data, and create items of interest such as isosurfaces, particle traces, and clip planes. Blockbuster is more of a video playback mechanism; however, it uses parallel computing technology to allow for very high resolution animation playback, along with image-manipulation tools such as panning and zooming of the running animation.

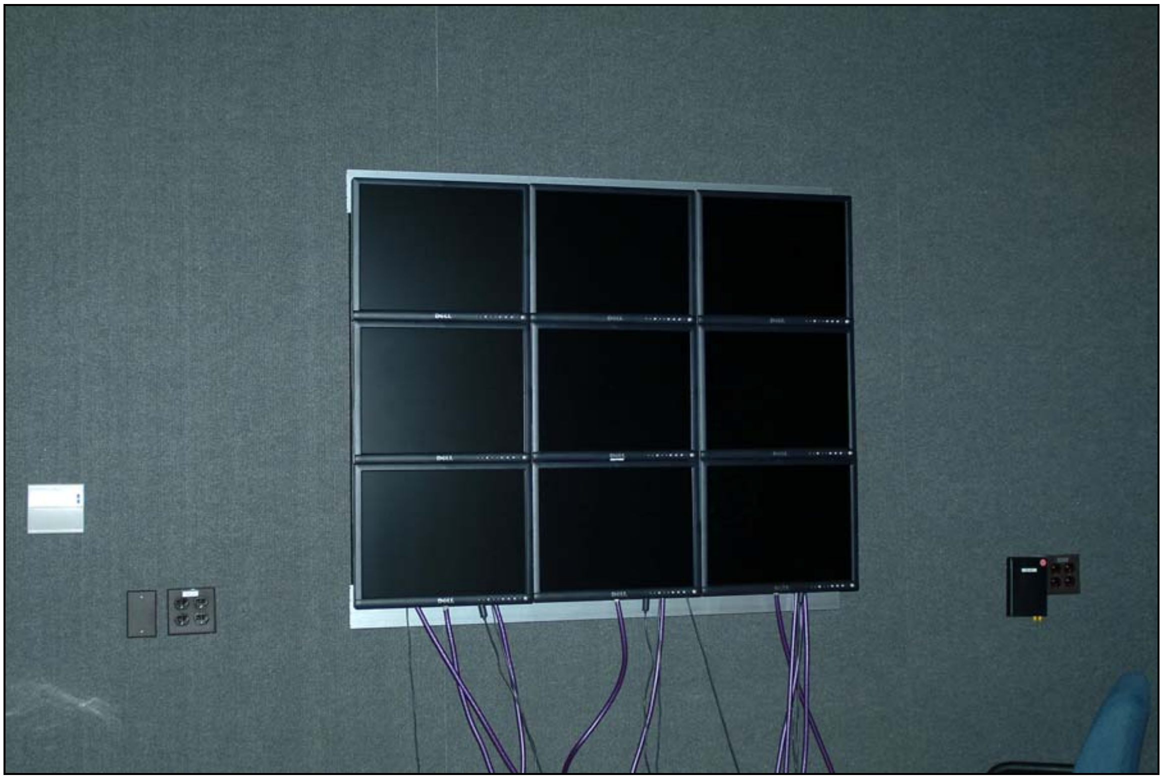


Figure 12. Monitor layout with exposed cabling.

There are issues associated with commodity cluster visualization that require continued attention, such as data locality and input/output (I/O) performance. To get optimal performance, the data needs to be located on the same node that is processing the data; however, moving data around to each node is often very cumbersome. Using a network shared file partition reduces the requirement for moving data to the local compute node, however, provides significantly reduced performance as each node is competing for read access to the same disk partition. We are currently evaluating the use of open source software Parallel Virtual Filesystem (PVFS2) as a methodology that combines the best of both worlds — a global file system that provides local I/O performance. We continue to pursue alternatives that will provide high-bandwidth performance to all nodes in the cluster.

Operationally, there are also issues that need to be addressed. The heat load associated with rack-mounted commodity computers is significant. If the cluster rack is located in an area that does not have a cooling system specifically designed for a computer facility, there may not be adequate cooling provided by normal building air conditioning. The commodity computers do have a high tolerance for heat, but there is concern about the long-term affect of heat on the longevity of these systems. Also, the noise generated by this cluster system may not be acceptable for location in an office work area.

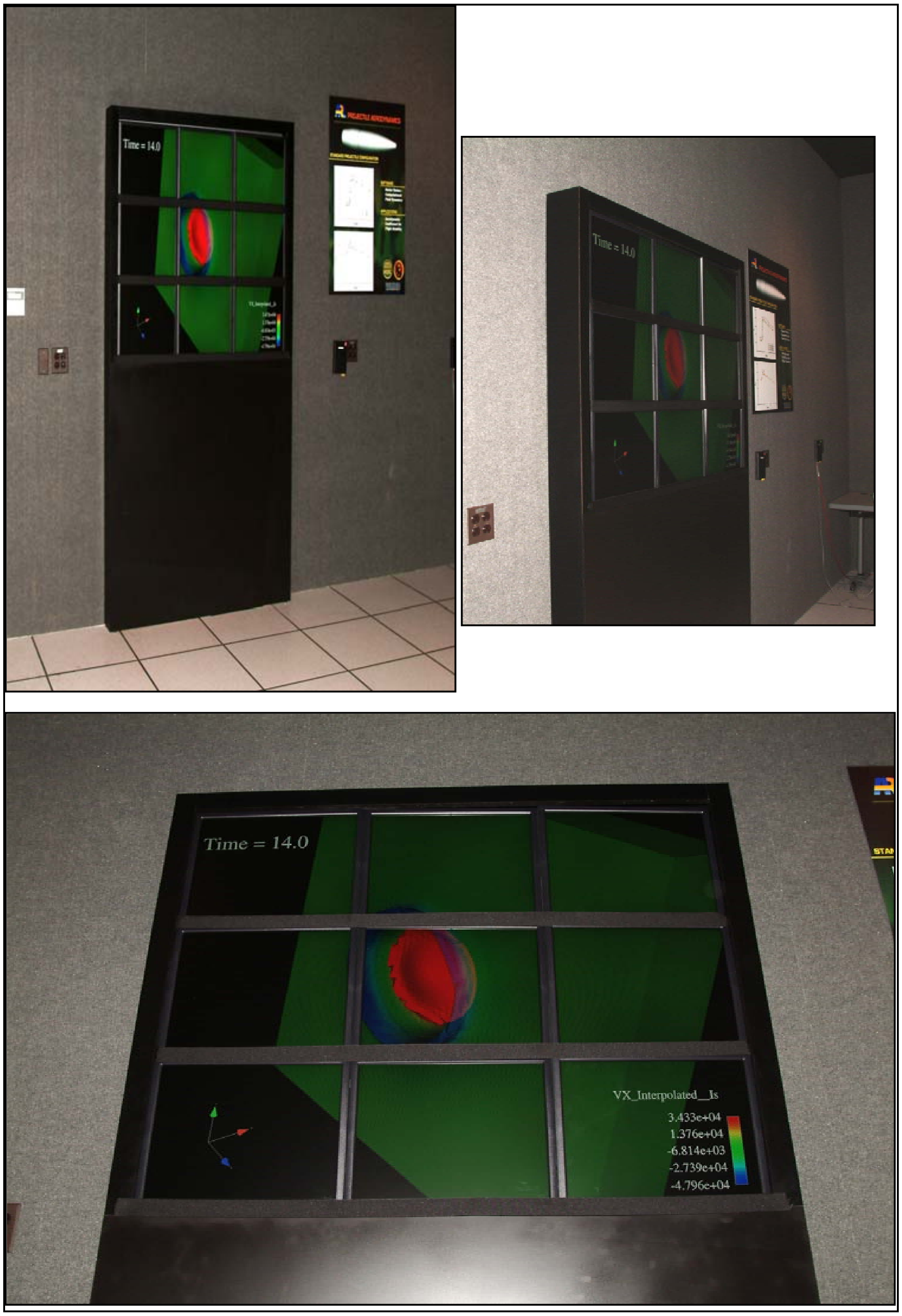


Figure 13. Completed tile display.

5. Performance of the Tile Display Cluster

We have demonstrated that the tile display and commodity workstation cluster can be assembled and enabled in such a way as to make a cohesive computational system. But, does the system provide capabilities that extend the value of scientific visualization? The evaluation of the overall performance of the cluster display wall is somewhat subjective. One way to measure the performance is to execute some simple timing tests on an application that is used on the tile display. For this measurement, an open-source utility from Lawrence Livermore National Laboratory called Blockbuster was used. Blockbuster reads image information from single media file and plays back the animation, and supports playback to both a single workstation display and to the tile display using distributed rendering applications under DMX.

The source file was a 600-frame animation rendered at a resolution of 2400 × 1200 pixels. In order to display this animation on a single, large CRT display, Blockbuster reduced the size of the images on the fly by one-third in order to fit it on the display screen, with a resulting resolution of approximately 1600 × 800 pixels. Running on a single workstation to a single display, blockbuster averaged a frame rate of ~6 frames per second, or ~7.6-million pixels per second. This frame rate is obtained simply taking the total amount of pixels display divided by the amount of time to process all 600 frames. From the observer's standpoint, the performance did not provide a visually smooth, consistent playback.

Using nine Linux workstations displaying to a 3 × 3 tile display, the performance is significantly increased. The resolution of the displayed image was 5097 × 3939 pixels, requiring Blockbuster to increase the resolution of the image by a factor of three in order to fill the display area. Blockbuster, in conjunction with the companion application DMX, runs in a client/server mode, with a master instance of Blockbuster running on a front-end workstation, and instances of a Blockbuster server and DMX running on each compute node. If we look strictly at the amount of time to start up Blockbuster and DMX, and display 600 frames of the animation, the resulting performance of 6 frames per second is equal to the overall performance of the single-node, single-display instance. But, in the distributed case, we are pushing an average of 120-million pixels per second to support the higher-resolution display.

It should be noted that there is an initially significant overhead associated with starting up both DMX and Blockbuster in this configuration. However, once the distributed application has been established on each node, there is a considerable increase in performance as viewed by the observer. Once running in the distributed environment, Blockbuster actually performs at about 12 frames per second, which is twice as fast as the single workstation performance, averaging about 240-million pixels of information per second. From the observer's standpoint, the animation is significantly larger, and runs quite smoothly when compared to the jittery results obtained from the single workstation playback.

6. Conclusion

This report presents a methodology for developing a high-resolution tile display based on commodity computing components. We successfully built a high-resolution tile display utilizing commodity display components, standard, off-the-shelf computer technology, and a variety of open sources and proprietary software packages.

Computational scientists are generating enormous data sets on the HPC resources at the ARL MSRC, and it is impractical to attempt to display these data sets in a compressed manner on a traditional computer monitor that will only show 1-million pixels of data. The graphics subsystems on many of today's desktop computers can easily outperform the visual supercomputers purchased only a few years ago, allowing scientists to perform complex visualizations at their desktop, visualizations that until now could only be performed in a state-of-the-art centralized visualization laboratory. The use of tile-display technology, along with a commodity computer cluster, will allow departmental groups to provide a facility to view high-fidelity representations of data, perform real-time analysis on very large data sets, and display the results of their computations to their peers.

7. References

1. Frank, R. J. *On the Use of Commodity Clusters for Large-Scale Scientific Application;* Presentation, Commodity Cluster Symposium, U.S. Department of Energy, Lawrence Livermore National Laboratory: Livermore, CA, July 2003.
2. Moreland, K. *Large Scale Visualization with Cluster Computing;* presentation, Linux Cluster Institute Workshop, U.S. Department of Energy, Sandia National Laboratory: Albuquerque, NM, October 2004.
3. Shalf, J.; Bethel, W. E.; Siegerist, C. *Personal Display Wall*. LBNL-PUB-894; Lawrence Berkeley National Laboratory: Berkeley, CA, 2003.
4. Clarke, J.; Vines, J.; Mark, E. *An Augmented Virtual Reality Scientific Data Center*; ARL-TN-229; U.S. Army Research Laboratory: Aberdeen Proving Ground, MD, September 2004.

Appendix A. XF86Config File

This appendix appears in its original form, without editorial change.

```
# This config file works for using both the DELL and Samsung monitors.
# The Dell is connected in the left-side DVI connector on the back of the computer.
# The Samsung is connected to the right-side HD-15 converter/connector.
# Created by Rick Angelini - July 26, 2004.
#
# XFree86 4 configuration created by redhat-config-xfree86

Section "ServerLayout"
	Identifier     "Default Layout"
	Screen      0  "Screen0" 0 0
	InputDevice    "Mouse0" "CorePointer"
	InputDevice    "Keyboard0" "CoreKeyboard"
	InputDevice    "DevInputMice" "AlwaysCore"
EndSection

Section "Files"

# RgbPath is the location of the RGB database.  Note, this is the name of the
# file minus the extension (like ".txt" or ".db").  There is normally
# no need to change the default.
# Multiple FontPath entries are allowed (they are concatenated together)
# By default, Red Hat 6.0 and later now use a font server independent of
# the X server to render fonts.
	RgbPath      "/usr/X11R6/lib/X11/rgb"
	FontPath     "unix/:7100"
EndSection

Section "Module"
	Load  "dbe"
	Load  "extmod"
	Load  "fbdevhw"
	Load  "glx"
#	Load  "record"
	Load  "freetype"
	Load  "type1"
#	Load  "dri"
EndSection

Section "InputDevice"

# Specify which keyboard LEDs can be user-controlled (eg, with xset(1))
#	Option "Xleds"			"1 2 3"
# To disable the XKEYBOARD extension, uncomment XkbDisable.
#	Option "XkbDisable"
# To customise the XKB settings to suit your keyboard, modify the
# lines below (which are the defaults).  For example, for a non-U.S.
```

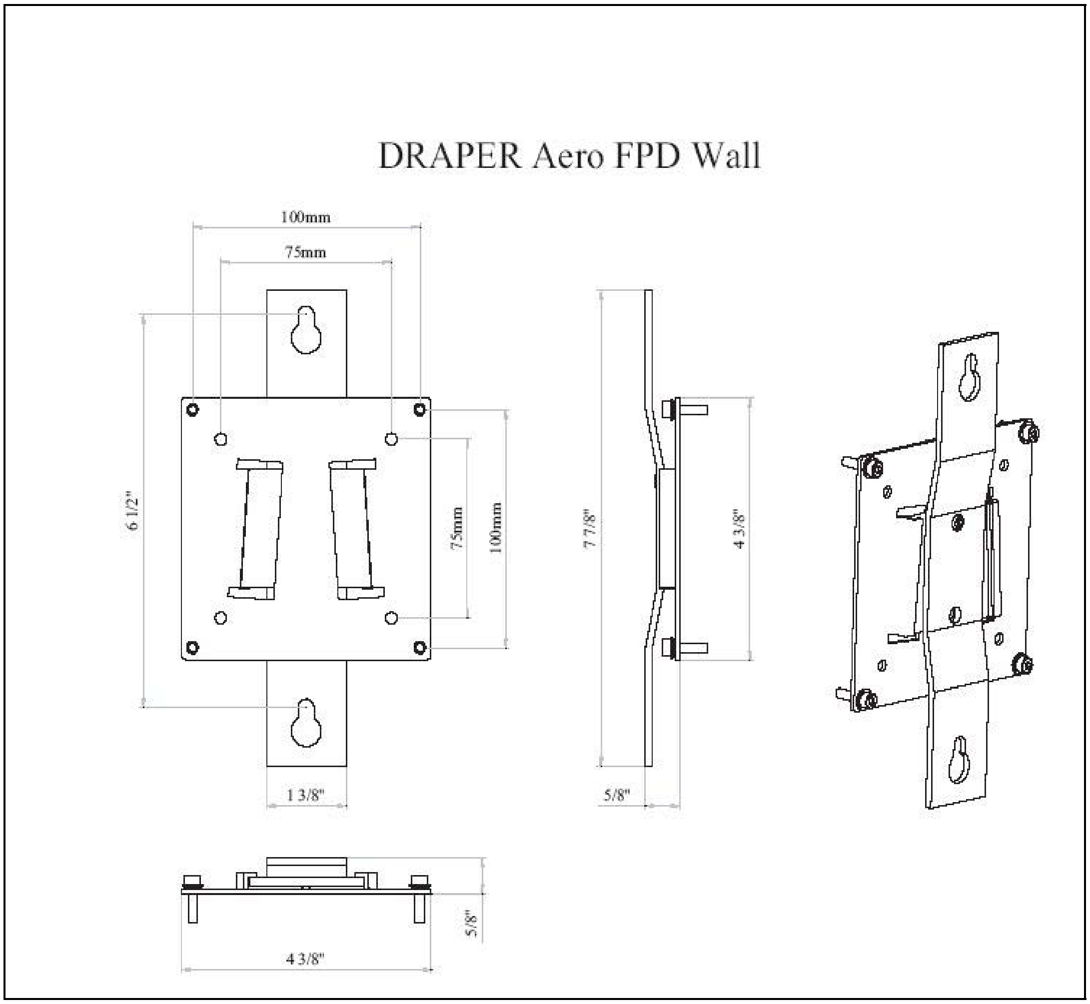
```
# keyboard, you will probably want to use:
#        Option "XkbModel"    "pc102"
# If you have a US Microsoft Natural keyboard, you can use:
#        Option "XkbModel"    "microsoft"
#
# Then to change the language, change the Layout setting.
# For example, a german layout can be obtained with:
#        Option "XkbLayout"   "de"
# or:
#        Option "XkbLayout"   "de"
#        Option "XkbVariant"  "nodeadkeys"
#
# If you'd like to switch the positions of your capslock and
# control keys, use:
#        Option "XkbOptions" "ctrl:swapcaps"
# Or if you just want both to be control, use:
#        Option "XkbOptions" "ctrl:nocaps"
#
         Identifier  "Keyboard0"
         Driver      "keyboard"
         Option      "XkbRules" "xfree86"
         Option      "XkbModel" "pc105"
         Option      "XkbLayout" "us"
EndSection

Section "InputDevice"
         Identifier  "Mouse0"
         Driver      "mouse"
         Option      "Protocol" "IMPS/2"
         Option      "Device" "/dev/psaux"
         Option      "ZAxisMapping" "4 5"
         Option      "Emulate3Buttons" "yes"
EndSection

Section "InputDevice"

# If the normal CorePointer mouse is not a USB mouse then
# this input device can be used in AlwaysCore mode to let you
# also use USB mice at the same time.
         Identifier  "DevInputMice"
         Driver      "mouse"
         Option      "Protocol" "IMPS/2"
         Option      "Device" "/dev/input/mice"
         Option      "ZAxisMapping" "4 5"
         Option      "Emulate3Buttons" "no"
EndSection
```

```
Section "Monitor"
        Identifier   "Monitor0"
        VendorName   "Monitor Vendor"
        ModelName    "Dell 2001FP"
        HorizSync    31.0 - 80.0
        VertRefresh  56.0 - 76.0
        Option     "dpms"
EndSection

Section "Monitor"
        Identifier   "Monitor1"
      VendorName   "Monitor Vendor"
      ModelName    "LCD Panel 1920x1200'
      HorizSync    63.8
      VertRefresh  59.8
      Option      "dpms"
EndSection

Section "Device"
        Identifier  "Videocard0"
        Driver      "nvidia"
        VendorName  "Videocard vendor"
        BoardName   "NVIDIA Quadro FX (generic)"
        VideoRam    131072
        Option     "NvAGP" "3"
        Option     "TwinView"
        Option     "TwinViewOrientation" "clone"
        Option     "SecondMonitorVertRefresh" "56.0 - 85.0"
        Option     "SecondMonitorHorizSync" "30.0 - 81.0"
#       Option      "HorizSync" "CRT-1: 30.0 - 96.0 , DFP-0: 31.5 - 90.0"
#       Option      "VertRefresh" "CRT-1: 48.0 - 160.0 , DFP-0: 60.0 - 60.0"
        Option     "Metamodes" "CRT-1: 1600x1200, DFP-0: 1600x1200"
EndSection

Section "Screen"
        Identifier "Screen0"
        Device     "Videocard0"
        Monitor    "Monitor0"
        DefaultDepth     24
        SubSection "Display"
                Depth     24
                Modes    "1600x1200" "1400x1050" "1280x1024" "1280x960" "1024x768"
"800x600" "640x480" "1920x1440"
        EndSubSection
EndSection
```

```
Section "Screen"
        Identifier "Screen1"
        Device     "Videocard0"
        Monitor    "Monitor1"
        DefaultDepth     24
        SubSection "Display"
                Depth     24
                Modes    "1600x1200" "1280x1024" "1280x960" "1152x864" "1024x768" "800x600!
        EndSubSection
EndSection


Section "DRI"
            Group        0
            Mode         0666
EndSection
```

INTENTIONALLY LEFT BLANK.

Appendix B. Draper Flat Panel Mounting Bracket



NO. OF COPIES	ORGANIZATION
1 (PDF ONLY)	DEFENSE TECHNICAL INFORMATION CTR DTIC OCA 8725 JOHN J KINGMAN RD STE 0944 FORT BELVOIR VA 22060-6218
1	US ARMY RSRCH DEV & ENGRG CMD SYSTEMS OF SYSTEMS INTEGRATION AMSRD SS T 6000 6TH ST STE 100 FORT BELVOIR VA 22060-5608
1	INST FOR ADVNCD TCHNLGY THE UNIV OF TEXAS AT AUSTIN 3925 W BRAKER LN AUSTIN TX 78759-5316
1	DIRECTOR US ARMY RESEARCH LAB IMNE ALC IMS 2800 POWDER MILL RD ADELPHI MD 20783-1197
3	DIRECTOR US ARMY RESEARCH LAB AMSRD ARL CI OK TL 2800 POWDER MILL RD ADELPHI MD 20783-1197
3	DIRECTOR US ARMY RESEARCH LAB AMSRD ARL CS IS T 2800 POWDER MILL RD ADELPHI MD 20783-1197
	ABERDEEN PROVING GROUND
1	DIR USARL AMSRD ARL CI OK TP (BLDG 4600)

NO. OF COPIES	ORGANIZATION
2	RAYTHEON SYS M BOLSTAD J RENTERIA 939 I BEARDS HILL RD PMB 191 ABERDEEN MD 21001
1	MISSISSIPPI STATE UNIV R MOORHEAD PO BOX 9627 MISSISSIPPI STATE MS 39762-9627
1	AERONAUTICAL SYS CTR R VICKERY HIGH PERFORMANCE COMPUTING BR ASC HP BLDG 676 2435 FIFTH ST WPAFB OH 45433-7802
	ABERDEEN PROVING GROUND
13	DIR USARL AMSRD ARL CI C NIETUBICZ AMSRD ARL CI HC R ANGELINI J CLARKE K KIRK E MARK R NAMBURU R SCHUMEYER J VINES W TAUBER AMSRD ARL WM TA B DONEY P KINGMAN AMSRD ARL WM TC K KIMSEY S SCHRAML

INTENTIONALLY LEFT BLANK.