DMX for Wall Displays?
(on UNIX/Linux platforms)

In July of 2001, Kevin E. Martin, David H. Dawes, and Rickard E. Faith presented a document that "...covers the motivation, background, design, and implementation of the distributed multihead X (DMX) system. It is a living document and describes the current design and implementation details of the DMX system. As the project progresses, this document will be continually updated to reflect the changes in the code and/or design. Copyright 2001 by VA Linux Systems, Inc."[1] Since that time, considerable progress has been made by the efforts of many, to the extent that now wall displays using the DMX concept are now in use.[2]

One appealing feature of DMX is the ability to create a large wall display consisting of many monitors showing an OpenGL image in a "single logical screen" (stretched desktop). Another feature allows the single xscreen to be "driven" by a number of computers, thus "distributing" the rendering load. Each of these rendering machines can be "controlled" by a remote computer running an OpenGL client that is directing the local rendering machines by parceling out portions of the work to each of them.

Another "feature" is that DMX does not require the use of Xinerama, since Xorg's X server implementation of Xinerama does not (yet) provide acceleration for all monitors, and apparently still has some stability problems.[2] The example at the University of Fairbanks, Alaska ("UAF") in footnote 2 uses graphics cards that can drive two monitors from a single graphics chip. This does not require Xinerama, and will provide acceleration on both monitors.

Requiring half the number of computers as there are monitors in the wall display array is one price to pay for DMX systems that use graphics cards that "drive" only two monitors from a single graphics chip. There are some cards such as the FireMV 2400 that can drive four monitors, but two graphics chips are required. Xinerama would be required to stretch an xscreen across the two chips, so they are not so useful for this application. The UAF example uses twenty (20) monitors, with ten (10) computers, each with an Nvidia card that has one chip driving two monitors with a single xscreen.

The ten rendering X servers of the UAF example are coordinated by an up-front proxy X server which communicates with the clients and apportions the X Protocol Packets across the ten rendering X servers to create the single logical screen display. Depending upon the communications and rendering loads, it might be possible to use one of the rendering machines as the proxy machine. Otherwise an eleventh machine will be required for the example at UAF.

_____

1. http://www.x.org/archive/X11R6.8.2/doc/dmx.html
2. http://www.cs.uaf.edu/2007/powerwall/
3. One would think that after five or six years, the Open Source Community with its "hundreds of contributors" could have fixed both the stability problems and the inability to provide hardware accelerated views on all monitor views.

The UAF system is depicted in Figure 1 below. (The ship we are showing is using Accelerated-X powered systems described later). Here we have assumed an eleventh machine running the proxy X server. There are over 35 Megapixels displayed with the 20 monitors. This setup assumes that the ten rendering machines can be used by one to ten remote clients, and the communications bandwith in the system is designed to support this. From the description of the setup provided by the UAF team, the rendering load seems not to require high performance, since the displays appear reasonably static or slow moving. However, the entire setup did require 7.2 KW of power. In Alaska, maybe that is a bonus.

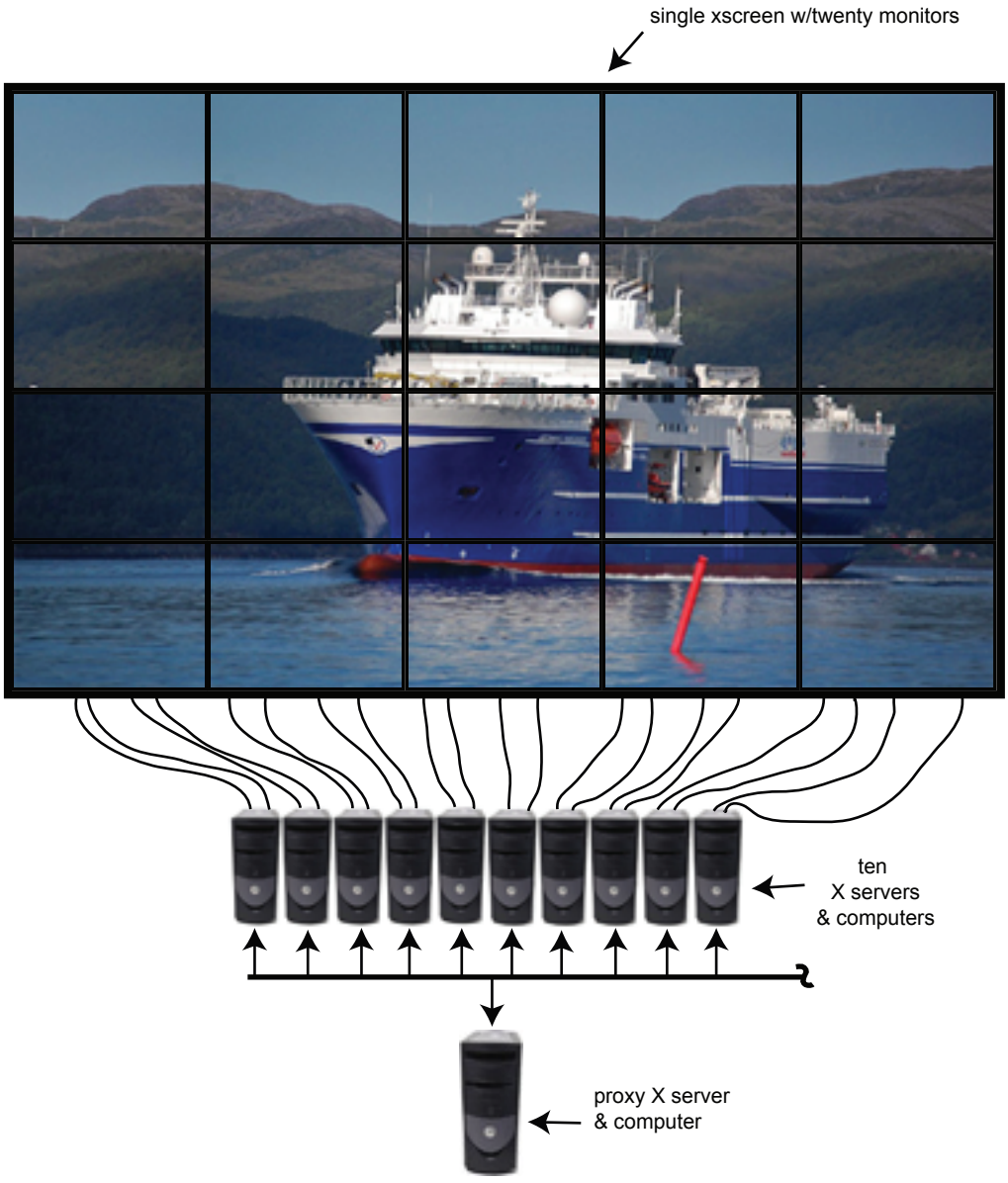

Figure 1. DMX example of Display Wall with 20 monitors

It is not clear from the UAF information whether the overall system is stable, nor what performance limitations, if any, have been encountered that limited the systems usefullness over time. It was clear, however, that the system used a lot of computers and graphics cards.

Alternative Approach

Using a single computer and three Colorgraphic Xentera GT8 graphics cards, or three Matrox's 8-monitor PPX graphics cards, it is possible to build a UNIX/Linux system to drive up to twenty-four monitors from a single computer and have hardware 2D acceleration for all monitor views if Xi Graphics' Wall Display HX Series software were used. Currently only 2D is available with the HX Series, since no one has expressed a (real) interest in OpenGL for such large displays using a commercial (non-open source) X server. On the other hand, there has been interest in using OpenGL on some rather large (in terms pixel real estate) wall display systems supported by other Xi Graphics' Summit Series.

One new Xi Graphics customer builds systems for viewing and analyzing 3D geophysical data in a conference room. In this case, the computers are located remotely in an equipment room where the fan noise cannot be heard by the conferees. This is accomplished by using Matrox's Remote Graphics Units that contain fanless graphics hardware, and has sound, keyboard, and mouse connections. The RG Units are located close to the monitors, and are connected to the computer via fibre optic cables. Oh, by the way, the conference room is on-board the ship shown in Figure 1.

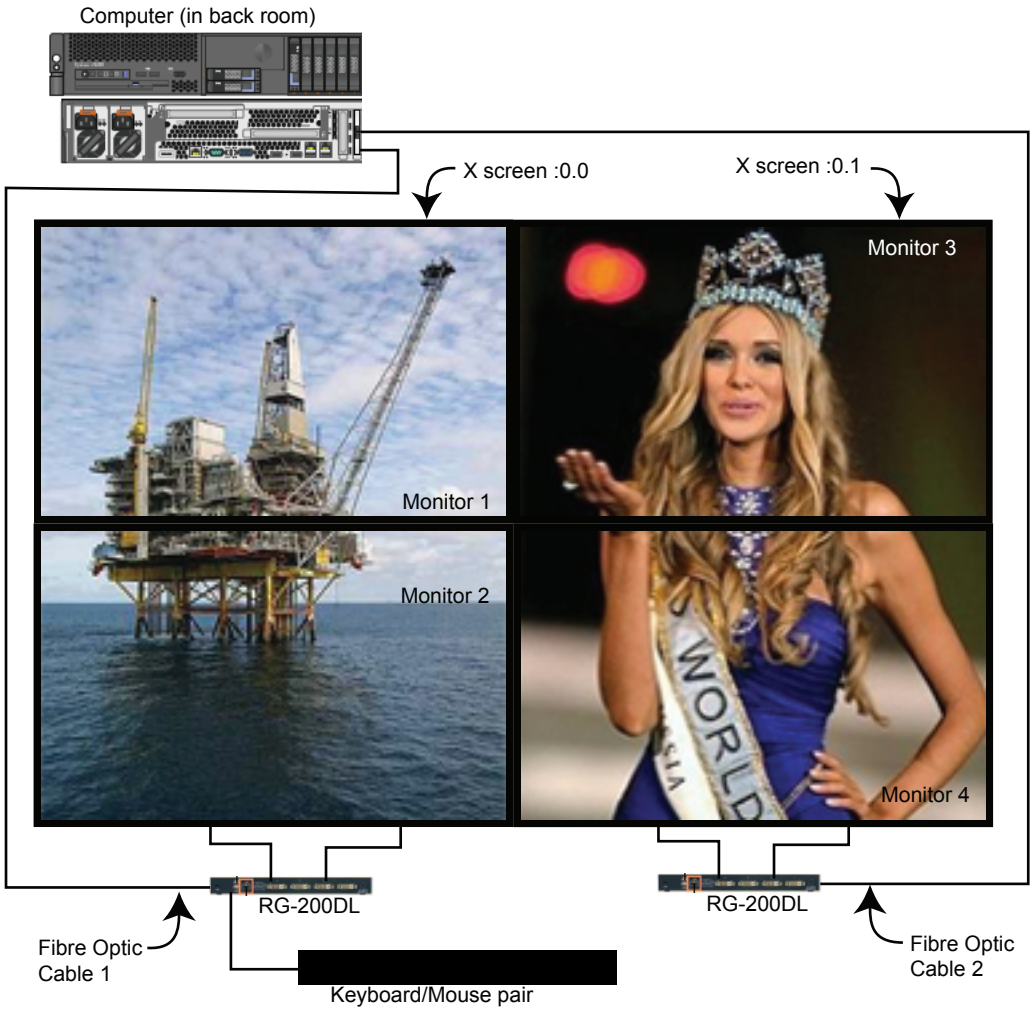

Figure 2. Dual RG-200DL System-Configuration

One system configuration for this customer is shown in Figure 2 on the previous page. It utilizes two Matrox RG–200DL units to drive two pairs of DualLink 2560x1600 resolution monitors. There are two xscreens as indicated, one for each stacked pair of monitors, each monitor comprising over 4 million pixels, and all using 24+8 colors (TrueColor main image plus PseudoColor image overlays). All views are fully hardware accelerated. Both RG Units are supported by one Accelerated–X™ server, using one mouse/keyboard pair.

This OpenGL-capable setup is very impressive, quite powerful, and very quiet. It is impressive because of the 16 MegaPix real estate, quite powerful (and fast) because of Accelerated-X, and very quiet since the computers are relegated to the equipment room, and there are no graphics card fans. And another small feature - the Accelerated-X graphics software is very stable.

Another configuration this customer uses, also with OpenGL, is a bit different. It utilizes two of the Matrox RG-400 Units, each supporting one xscreen as shown in Figure 3. The monitors are smaller, but there are eight of them, with all views hardware accelerated. Again, the displays are 24+8 and use hardware-accelerated image overlays. The real estate involved is a bit less than in the first system, being only 15 MegaPix. On the other hand, the monitors are considerably less expensive.

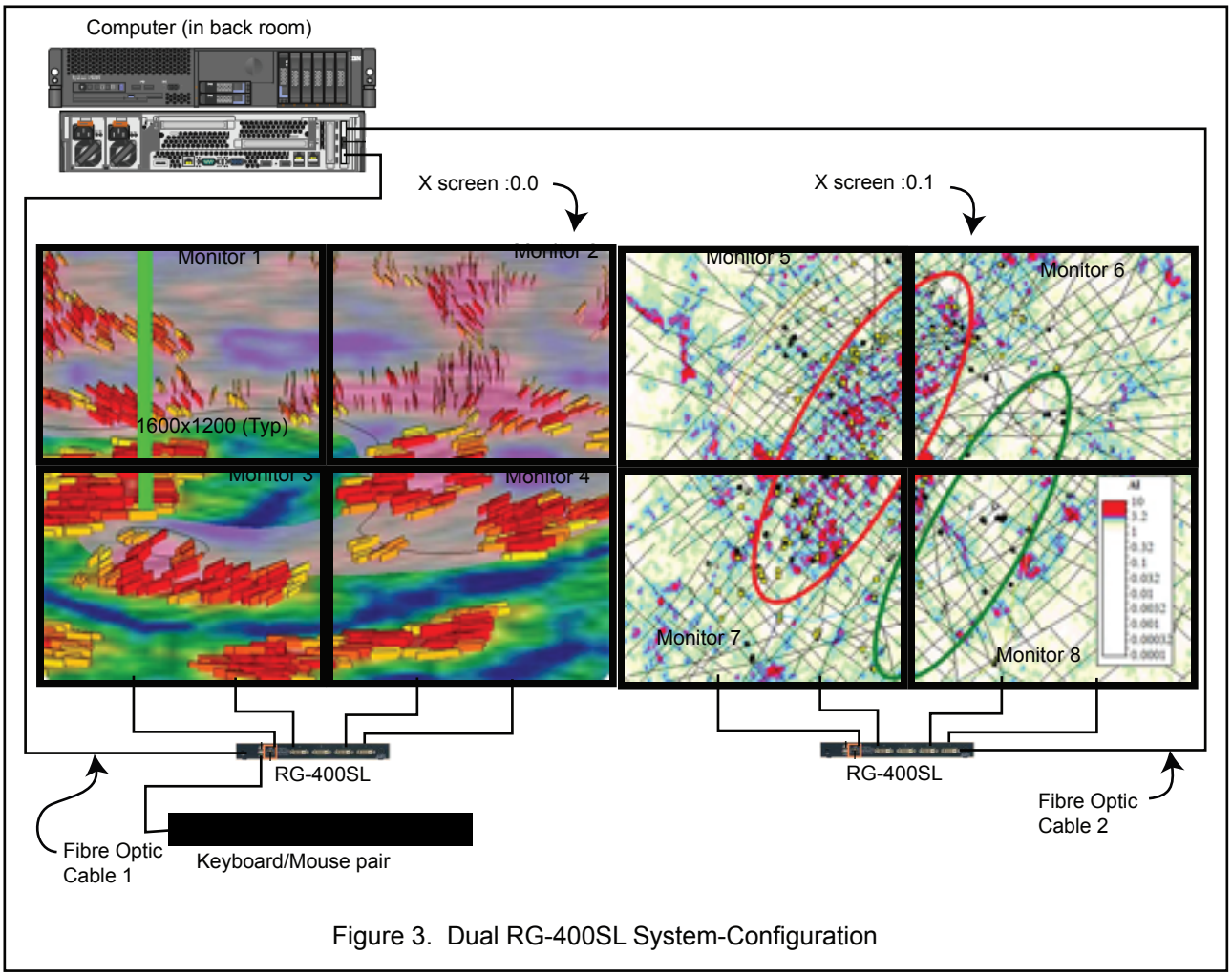

Figure 3. Dual RG-400SL System-Configuration

Xorg Open Source Freeware vs Accelerated-X

Each situation is different, of course, and given enough time and effort, the Xorg open source approach can usually be made to work, at least in some fashion. However, the larger, more complex systems are sometimes very difficult to bring up to a level of performance and stability for commercial applications. In a University Environment, performance and time may not be heavily-weighted factors. In the case of the customer we have used as an example, the firm was unable to get satisfactory performance or stability using the open source approach, and "found" Xi Graphics when searching for help to meet a fast approaching deadline on a major project - sending the new ship on its maiden assignment.

In this situation, the cost of commercial quality graphics software was trivial in comparison to the cost of missing a major deadline. There have been many, many such interesting situations that Xi Graphics has had the pleasure of working on over the years. And they just keep coming.

---

dhm revised Feb 2009