

Administering Remote Graphics

By Micah Altman

Last Revision: Nov. 17, 1997
Copyright 1992-7

Administering Remote Graphics

- Types of Remote Graphics
 - "vanilla" X-windows
 - "remote GL" Distributed GL (*obsolete*)
 - GLX (Open GL, X extension)
- Issues
 - analyzing connections
 - debugging remote graphics display
 - improving performance
 - maintaining security

Overview of Remote Graphics Connections

- Application is executed on graphics client, display appears on graphics server
- Setting up a connection
 - choose a target display
 - establish a connection
 - display graphics

Analyzing Remote Graphics Connections

- Choosing a target display
 - all use X display format, machine:server#
 - local: localhost:0
 - remote: distant.net:0
 - display location specified by X resource, command line argument, or the environment variable DISPLAY

— the `.login` files SGI distributes with its default accounts automatically set `DISPLAY` to point back to remote machine when you login from a remote machine

- Establishing a connection
 - `GL`, `X`, and `GLX` use standard X connection mechanism to establish a window
 - All use X access control mechanisms
 - `/etc/services` lists the port for the X-windows service
 - Uses the TCP protocol

Displaying Graphics Remotely

- Open GL requires the `GLX` extension on the X server to display graphics
- (*OBSOLETE*) Distributed/remote GL uses the `dgl` mechanism to display information
 - `sgi-dgl` service must be listed in `/etc/services` on both server and client
 - `sgi-dgl` service must be listed in `/etc/inetd.conf` on server
 - `dgl` must run on server
 - display only supported on SGI machines

Examining X Windows

- List clients running with `xlswins`
 - `xlswins` - lists active X clients
 - this syntax extracts name and id number of each window:
`xlswins -indent 0 |grep -v '()'`

Get information about window

- `xprop` - shows information for each client
- use `xprop -id` with the id number obtained above
- can obtain name of application `WM_NAME` field, and name of machine running application `WM_CLIENT_MACHINE (STRING)`
- can kill window with `xkill`

More detailed information about connections

- if a client is failing to connect for an unknown reason use `xscope` to get more information
- on the server run `xscope on display 1`
- set your client to display on display 1,
`setenv DISPLAY remote.machine.net:1`

- all details of connection will be traced
- remember, use `gldebug` for local gl program debugging
- `xmonui` - graphic interface and filter for `xscope`

Troubleshooting Remote Graphics Connections

- Use techniques for diagnosing basic tcp services, but remember
 - client must be authorized to connect to server
 - errors are reported in `/usr/lib/X11/xdm-errors` as well as in `SYSLOG`
- Some symptoms of connection problems
 - client can't open/connect to display
 - client not authorized to connect to display

• Common Problems with remote Display

- Remote host specified incorrectly
- X application fails to use NIS correctly, if not linked with `-lsun`
- X client not authorized to connect to server
- X server not running
- X window configuration file problems.
 - remember to check all configuration files, normally: `xsession`, `Xresources`, `xsession-remote`, `.sgisession`
 - check system defaults in `/usr/lib/x11/xdm`, user defaults in user's home directory
- Open GL: GLX extension not supported by X server
- Remote GL: `dgld` service not running on server, or not listed on client

Special Troubleshooting Issues

- Decnet
 - SGI 4DDN (Decnet) implementation does not support X windows
- X terminals

- x terminal software must be installed
- most X terminals use tftp, rarpd, and or bootp to start up and load necessary data. Make sure these services are installed and configured
- most X terminals will not currently support GLX extension
- when running windowing system set the X resource `SG_manageSession: true`, and the shell environment variable `_SGI_NO_REMOTE_GL` to avoid use of GL in the window manager
- or set `_NO_DESKTOPIMAGES`, to avoid even more graphics in the window manager

Tuning X Clients

- X clients can degrade server performers
 - X servers grow in response to resource requests by clients
 - large X server can degrade performance of host
 - large color images, bitmapped root windows, display postscript operations, are particularly expensive
- Choose clients carefully
 - there are light weight alternatives to many common client programs (e.g. xterm, xclock, etc)
 - avoid mixing clients which use different X toolkits (e.g. Xlib vs. Motif)
- Reduce network activity
 - avoid running clients over the network which transmit bitmaps
 - avoid active screen-savers on X-terminals, blank the screen instead
 - avoid network traffic when not needed. Run local programs with `DISPLAY` set to `"localhost:0"`

Tuning the X Server

- Control resources in `/usr/lib/X11/xdm/xdm-config`
- Reduce X server size
 - X server may grow over time because of client memory requests
 - to avoid bloating, X server can be restarted at the end of each session using the resource:
`DisplayManager.DISPLAY.terminate_server: true`
- If network congestion is causing X connections to time out

— adjust timeout values for opening servers:

```
DisplayManager.DISPLAY.openDelay  
DisplayManager.DISPLAY.openRepeat  
DisplayManager.DISPLAY.openTimeout  
DisplayManager.DISPLAY.startAttempts
```

— adjust timeout values for active windows:

```
DisplayManager.DISPLAY.pingInterval  
DisplayManager.DISPLAY.openTimeout
```

- xperf can be used to benchmark various server operations

Security and Remote Graphics

- What access to the X server allows
 - creation of windows can be used to spoof logins
 - keyboard input can be monitored, for snooping attacks
 - xkey : a publically available snooper
- Authorization
 - anyone logged in has access to X server
 - xhosts is used to list other hosts that are allowed to connect to server
 - xauth can be used to allow particular users on other machines to connect to the server
 - in general, these mechanisms are very weak
- Server grabbing
 - grabbing the server input makes snooping via X more difficult
 - the XDM login can be set to grab server input during the login process:

```
DisplayManager.DISPLAY.grabServer: true  
DisplayManager.DISPLAY.grabTimeout: value
```
 - sophisticated programmers can still read keypresses

Security Features, Pt. 1 - xhost

- xhost controls access to display by different client hosts
 - xhost + / xhost - , enables/disables all host access
 - xhost +/- hostname, enables/disables access by particular host
 - NOTE: in earlier OS versions default SGI /usr/lib/xdm/Xsession and /usr/lib/xdm/Xsession-remote automatically issue "xhost +", to disable access control

— NOTE: users connecting remotely via `telnet/rlogin` and then issuing x connections from processes running on host are seen as local to host, not as remote. Under some OS you can use `xhost - `hostname` localhost` to disable use of display by anyone except physical console login.

Security Features, Pt. 2 - xauth

- xauth allows access to individual users not on the xhost client access list
 - code values, called "magic cookies", are transmitted to check authorization
 - SGI supports only plaintext (unencrypted) "magic cookies"
 - to extract a code value for one display and send it to another use:
`xauth extract - $DISPLAY | rsh other_host xauth merge`
 - magic cookies are stored in `~/.Xauthority` and can be shared through NFS mounting
 - to set up server set the `server` resource
`DisplayManager.*.authorize: true`
 - NOTE: only the current version of remote GL understands xauth mechanism

Security Features, Pt. 3 - xdm

- XDM configuration
 - the resource `DisplayManager.accessFile` points to a access control file for XDM
 - this file contains a database of hostnames
 - allow/deny direct access to this machine
 - forward queries
- Advanced Issues
 - x through firewalls: need a proxy in order to determine that foreign access request was initiated by user inside firewall (consider SSH - below)
 - encrypted keys: normal magic cookies are plaintext
 - `mxconns`: Free X11 connection monitor. Creates alternate X server where you choose each connection individually.
 - `ssh` - secure shell, can also handle x proxy and *session encryption*
 - + X11 clients should use Xauth (MIT-MAGIC-COOKIE protocol)
 - + Activate X forwarding in ssh config (on by default)
 - + Log on via SSH
 - + SSH sets up pseudo display on host and `DISPLAY` variable on client

Going Further

- Some Useful books

Young, Doug 1989. The X Window System: Applications and Programming with Xt (Motif Version),
Prentice Hall: U.S.A.

Mui, Linda and Eric Pearce.1993., X Window System Administrator's Guide for X11 R4
O'reilly and Associates: U.S.A.

Neider, Jackie, Tom Davis, and Mason Woo 1993 OpenGL Programming Guide,
Addison-Wesley: Massachusetts

- Newsgroups

- `comp.windows.x`

- `comp.graphics.api.open-gl`