Introduction

The X Window System was started *via* Project Athena at MIT back in 1984. Good graphics terminals were expensive in those days, and powerful computers rare, so most centres used many cheap terminals connected to one large computer, with one or two good graphics displays.

Researchers logged in from a small and cheap text terminal to the large computer, often running UNIX, to access various programs that were only available on the large computer: some to run a C compiler, some to run engineering models, some to develop graphics programs. Thus UNIX was multi-tasking (running many different programs at the same time) and multi-user (having many different users logged on to the same computer) from the beginning. Its networking ability was designed to accomodate all these needs.

X11 System Design

The X11 system lets you login from one computer `elcheapo` to a second `powerful`; run a program on `powerful`; and display the results on a third computer `display`. The computer `powerful` can be in the same room, in the same city, in the same country or anywhere across the world. The computer `display` can be in the same room, the same city, the same country or across the world. The network can be a direct cable, a Local Area Network, an intranet or the internet. All that is necessary is that each computer has a known IP address at its interface to the network.

Typically, you access the program that you need on massive computing power elsewhere and display results on your own screen. Your computer `elcheapo`, however, has typically no hard disc or a small hard disc, perhaps just a floppy disc, and a bit-mapped VESA graphics card.

This magic is performed by a small program running on `elcheapo` called an `X-server` or just `X`. Here are the sizes of the X-Server and its libraries on `debian etch 2.6.18` at present:

```
   7672 /usr/bin/X
1241392 /lib/tls/i686/cmov/libc-2.3.6.so
  88164 /lib/ld-2.3.6.so
```

As you can see, the total size is less than 1.4 Mbytes, so these libraries and the X-server can fit on one floppy disc and can thus run on pretty well any computer.

The X-server receives requests from your program to display graphics on the terminal. There are many ways for this to work; here are three common ones:

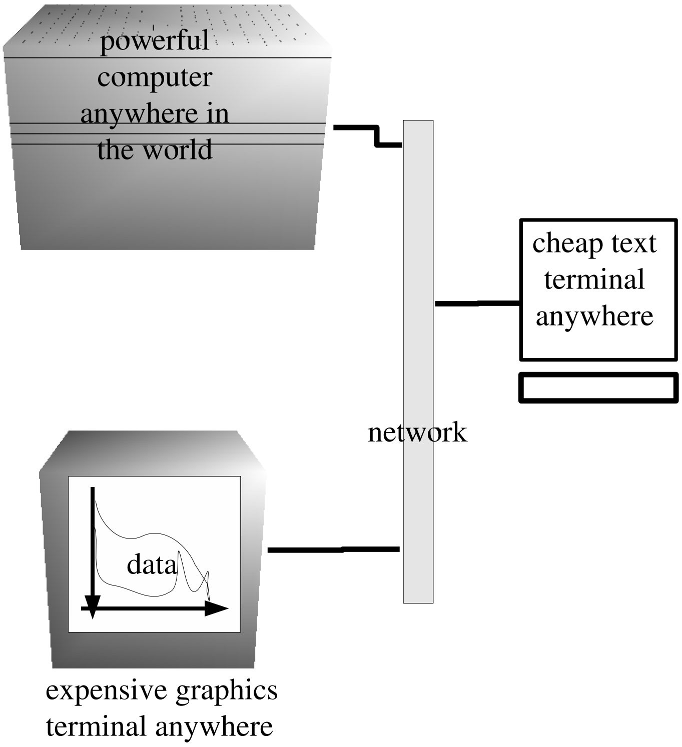
powerful computer anywhere in the world
cheap text terminal anywhere
network
data
expensive graphics terminal anywhere

Figure 1: X Window System in Action

- On a single workstation, the requests can paint graphics to any of a number of virtual displays, which are referred to as `:0, :1, ... :9`, etc.
- On a network, requests can come from trusted `xhosts`. Requests are displayed on the logical terminal defined in the environmental variable `DISPLAY`.
- Most X application programs have a command-line option `-display` so you can say where the results should be displayed when the program is run.

We shall see all of these methods in operation during this talk.

X on a single computer

Now try all this on your own `debian etch` laptop installation. Boot and log in. Bring up a terminal (press `CTRL-ALT-T` or click icon). Make sure you have all the software you need – run as root:

```
# apt-get install ssh xterm xserver-xorg-core xserver-xorg x11-common xfonts-base
```

The first virtual text terminal might still be used for your login session, so switch to the next available virtual text terminal by pressing `CTRL-ALT-F2`. You should see a text login prompt, so log in as yourself. Your login display manager might have already started X on the first logical display (:0). So let's start an X-server running on the second logical display by typing:

```
  elcheapo>  X :1 &
```

which means: run the X-server (X) on display number one (:1) and put it in the background (&) so you can go on to do other things. The screen goes blank while the X-server thinks and then displays its normal grey cross-hatched screen with a single × cursor on it – this means that the X-server is working (it understands the graphics card and the mouse, at least). The cursor can move, but there is no way to input commands yet. Lets fix that now.

Return to your virtual text terminal by pressing CTRL-ALT-F2 and then press ENTER. Put an X-terminal onto that display :1 by typing:

```
elcheapo> xterm -display :1.0 &
```

which means: run the xterm program (xterm) on the root window (.0) of display number one (:1) and put it in the background (&) so you can go on to do other things. You won't see much happen but a job number and the command prompt appear like this: [2] 1739. However, now switch to that X-terminal by pressing CTRL-ALT-F8 (CTRL-ALT-F7 is used already by xdm) and you see that you have started a smallish white X-terminal in the upper left corner of the grey X-server area, with a command prompt in it. Get the focus into that X-terminal by moving the crossed mouse cursor into it, until it changes to an I, and then type some command, like ls followed by ENTER – and it works!

So we have one X-terminal. Start another by typing xterm &. Whoops! What happened just now!? You do not see much. But in fact a new X-terminal has started *over the top* of the other one. This is because the default position for an xterm is up there in the top left corner. However, we cannot move the top one out of the way. So how do we know there are two of them?

Window Managers

Let's talk a bit about window managers. Window managers allow you to move and resize windows. You probably already have one running on your login session on the default virtual terminal accessed by pressing CTRL-ALT-F7 (have a look now!). We need to get one to run on this screen.

There are countless window managers for Linux. How do you know which are available on *your* system? To find out, type this:

```
elcheapo>  man -k window|grep manager|less
```

and you might see some window managers from amongst those in this list:

```
9wm aewm afterstep amaterus amiwm asclassic ctwm enlightenment evilwm fluxbox
gnome-wm icewm ion2 matchbox nawm olvwm openbox oroborus pwm qvwm ratpoison
sapphire sawfish twm uwm w9wm waimea windowmaker wmii xfwm4 x-window-manager
```

If you find none, install one by typing `sudo apt-get install [choose one from above]`. This list is not exhaustive. Try any of them; e.g., type `gnome-wm` into the xterminal then press `ENTER`.

What a surprise! Suddenly you have a full window-manager running, and all the bare windows now have title bars and they can be moved and resized! Try it. As you move one of the xterms out of the way, you can see that we had two xterms on top of one another in that top left corner.

Now just focus in the xterm where you typed `gnome-wm` and press `CTRL-C`; this terminates the window-manager and just as suddenly all your windows have lost their title bars and can no longer be moved!

Try this: start any other window manager from your list by typing its name.

Now lets look at the other X displays on your computer. Press CTRL-ALT-F9 to get to virtual display number 2. We have an xterm running there. Type into it, say, `twm` or any of your available window managers, and now this screen has the ability also to move and resize windows. If you want the window-manager to persist then activate it like this instead: `twm &`.

X on a network

Enough exploring possible window managers. Lets do something practical.

A typical situation is that you are on a small and simple server without much extra application software – for example, you have the system on a pretty-full hard drive, and you have no `OPENOFFICE` package installed, and no room left for it (a full install can take up about 160 Mb or so) but you want to create a spreadsheet for admin. purposes.

Log into your `elcheapo` text terminal (which has at least a basic VESA buffer controller for the video card and the X-server program). First start an X-server on `elcheapo` by the methods above. On the terminal exposed by pressing `CTRL-ALT-F2`, type:

```
elcheapo>  X :1 &
```

This starts a hatched grey X-server on the display :1. (Return to the virtual terminal by pressing `CTRL-ALT-F2`). Then you may allow another computer to send X-requests which display on this one. For example, if the `powerful` interface has the IP address 192.168.1.2 you would type:

```
elcheapo>  xhost +192.168.1.2
```

which obtains the response `"192.168.1.2 being added to the control access list"`.

First check the description of the display that you are on. At the prompt, type:

```
elcheapo>  printenv|grep DISPLAY
```

which results in something like:

```
DISPLAY=:1.0
```

Now get back to the command line on `elcheapo` by pressing `CTRL-ALT-F2` and log into `powerful`. For example, suppose that `powerful` is reached at IP address=192.168.1.2 and your account name on `powerful` is `lsg`. You would type:

```
elcheapo> ssh lsg@192.168.1.2
```

When logged in, first divert the display from `powerful` back to the `elcheapo` X-server screen that we created above. For example, if the `elcheapo` interface is at 192.168.1.70 and the display screen was described as `:1.0` like above, you would type:

```
powerful> export DISPLAY=192.168.1.70:1.0
```

From now on, you can run any program (e.g. `openoffice`) on `powerful` by typing into the xterm on `elcheapo` (which is logged in to `powerful`) like this:

```
powerful> openoffice
```

and the result will be displayed on `elcheapo` right in front of you, and all keystrokes and mouse clicks go back from here to `powerful`. You can then create your spreadsheet. Try it. Copy your file back by typing

```
powerful>  scp file.sxc lsg@192.168.1.1:/home/lsg/file.sxc
```

and zap the `X-server` when you have finished with it by pressing `CTRL-ALT-BACKSPACE`.

Using X to Share Desktops

Virtual Network Computing (VNC) is a general field that includes sharing desktops for instruction, debugging help, demonstrations, etc. X makes it fairly straightforward; let's see how.

Log in to `elcheapo`. You want to share the desktop with `friendly` which you know can be reached at (e.g.) IP address 192.168.1.100 on your LAN.

First, let's make sure you have all the software you need: the `X11` virtual network computing server and the `X11` virtual network computing viewer (client). Run the following command at your terminal: `sudo apt-get install x11vnc xvncserver`

`friendly` decides which display is to be shared by going to that display and typing on it:

```
friendly>  x11vnc &
```

This starts a program (`x11vnc`) in the background, which allows *anyone* to view that display, warning that you are running `vnc` without a password (OK for now) and *also* telling what the display is called; e.g., it might say that the VNC desktop is: `friendly:0`. You need to know this.

You now start a `vnc` viewer to view that display on your `elcheapo` by typing:

```
elcheapo>  xvncviewer 192.168.1.100:0
```

which specifies `friendly's` IP address (192.168.1.100) and the display description (`:0`) that was told to you (in person or over the phone, or whatever) by `friendly's` user.

Suddenly, you can see what each other types, see the result of that typing, and can start applications that display on both terminals *just as if you were sitting at the other terminal.* [In effect, the X server is now controlling two terminals, two keyboards and two mice over the network.] You may switch to another desktop and start a similar session with yet other computers on the network. You end the session by hitting key `F8` and then clicking on → `quit viewer` in the menu that appears.

Useful References

These are handy.

"X Power Tools" by Chris Tyler (publ. O'Reilly in 2008).

"X Window System Administrator's Guide for X11 Release 4 and 5" by Linda Mui and Eric Pearce (publ. O'Reilly in 1992).