Subject: Re: Object Graphics Windows in X Windows
Posted by Karl[1] on Fri, 07 Mar 2008 22:45:24 GMT
View Forum Message <> Reply to Message

On Mar 7, 2:59 pm, Vince Hradil <hrad...@yahoo.com> wrote:
> On Mar 7, 1:25 pm, "R.G. Stockwell" <notha...@noemail.com> wrote:
>
>> "R.G. Stockwell" <notha...@noemail.com> wrote in message
>
>> news:fqs4lj$fj3$1@aioe.org...
>
>>> "David Fanning" <n...@dfanning.com> wrote in message
>>> news:MPG.223a6f373da1440998a2b5@news.frii.com...
>> ...
>>> I use VNC (TightVNC) to access my linux desktop from my laptop,
>>> and it works very well. One nice feature is that you can have multiple
>>> people log on and share a desktop (i.e. to broadcast a meeting for
>>> instance).
>
>> PS I just verified that I-tools do indeed work under VNC remotely.
>
>> Cheers,
>> bob
>
> That's what I wanted to hear. Thanks.

I noticed that Xming has a file in their download section with 'mesa'
in its name. This may give you (Fanning) the 3d support you want.

Not all X servers support OpenGL. Xming strikes me as a sort of
lightweight product that has minimal frills. So, they probably made
the GL support, which is not a lightweight feature, an option.

I'd suggest that David download and try the 'mesa' option. I couldn't
find many docs on that Sourceforge site and I have not played with
this X server myself, so I cannot say much more about it.

The other poster who said they were using IDL with Xming just fine was
apparently only using Direct Graphics, which is pure X protocol.

X servers that support OpenGL over a remote connection do so via the
GLX extension to the X prototcol, which is basically a protocol
wrapper for OpenGL graphical elements.

As an X client, what IDL is *supposed* to do if it hooks up to a
remote X server that supports GLX is to send GLX protocol for Object

Graphics. If the server does not support GLX, IDL should render
OpenGL to a local software buffer, and then blit that to the X server
via X protocol.

On a remote connection, each of these methods have a different "wire
cost". The GLX connection performance will depend on the complexity
of the scene in terms of number of primitives, etc. The software
rendering path will be a consistent wire load, the cost of blitting a
window-sized bitmap.

If the user forces software rendering, IDL will render the scene using
software and send the buffer to the X server.

These same principles apply to a local environment, client and server
on the same machine, except that there is a "direct" connection
between the X client IDL, and the X server, where the 3D commands are
sent more directly to the graphics processor for performance reasons.

Now, as a server, the Xming product itself with the mesa option may or
may not hardware accelerate OpenGL. Since the name of their optional
module is 'mesa', that suggests software rendering, but that may not
be the case - it could easily be hardware accelerated. It depends on
how Xming was implemented. And Mesa supports hardware accel OpenGL on
many platforms and devices. (Mesa is the OpenGL lib used to implement
GL on Linux and other environments)

I've seen really good performance running a remote OpenGL app and
having it display on the Linux based X server with hardware
accelerated OpenGL in front of me. As long as the protocol traffic
didn't get too large, it worked quite well.

These technologies (X, GLX, and Xming) are quite different from the
"remote desktop" approach to remote computing. In the former, you
have an application, or set of applications, sending protocol to the
server. For a remote desktop situation like VNC, you're rendering the
entire desktop on the remote side and doing an efficient desktop
"copy" operation to the client viewing program, using special tiling
algorithms.

The X VNC server presents itself as just an basic X "framebuffer"
server to the attaching X client, and usually does not support GLX.
(If it did support GLX, it would likely be a software
implementation). If the attaching X client is IDL, IDL would see that
there is no GLX, and then revert back to software rendering and send a
bitmap with the result to the X VNC server. That's why IDL and I-
Tools work with VNC. IDL is probably using software rendering here.

Karl