Subject: Re: Object graphics under Linux: are they supposed to be that slow?
Posted by marc schellens[1] on Mon, 28 Jan 2002 19:33:35 GMT
View Forum Message <> Reply to Message

Could you post the program?
I would like to compare the performance on my
machine (PIII 700Mhz, Matrox G400).
This way it could be sorted out if its the graphic driver.

Subject: Re: Object graphics under Linux: are they supposed to be that slow?
Posted by nobody@nowhere.com (S on Mon, 28 Jan 2002 20:38:07 GMT
View Forum Message <> Reply to Message

Hi Mark:
I'm not sure I've the answer for you, but I suspect that the big difference is in the graphics subsystem, the windows more than likely supporting the hardware acceleration and Linux (XFree I'm assuming) not. You didn't mention what version of X you were using. Here's from the XFree site about hardware acceleration support of the latest XFree release:

4.2.0:

Support (accelerated) for the Intel i740 is provided by the "i740" driver, and support for the Intel i810 (including i810-dc100 and i810e), i815, and i830 is provided by the "i810" driver. The "i810" driver is currently supported only on Linux and FreeBSD (4.1 and later), and requires AGP GART kernel support.

So you might check your kernel and see if it has this compiled in by doing ksyms -m, you should see something about agpgart appear in the output, if you don't you probably need to compile a kernel with support for this. If you do, you might check the output of XFree to verify that it is using the acceleration, if you're kernel supports it. On my system, this is spewed to /var/log/xdm-error.log (I use xdm, you probably do too). My Linux box uses an Nvidia chip and the accelerated-driver is proprietary and buggy, so I've just forgone any acceleration on my system.

I don't know what your reasons for moving to Linux were, but if you want accelerated graphics (Open GL / DRI), that's on the bleeding edge of Linux. I've been a Linux user since 1995 and I think it has many advantages, but this is an admitted weakness. I'd be interested to hear what other users say about this, and how well the software accelerated Open GL provided by RSI with IDL for Linux works on other systems.

On Mon, 28 Jan 2002 12:23:58 +1300,
Mark Hadfield <m.hadfield@niwa.co.nz> wrote:
> Hi guys
>

> I have recently been considering a switch to from Windows to Linux for
> various reasons that I won't go into here. I have set up a dual-boot system
> on my PC and, as of today I have IDL running on both OSes. I'm afraid it's
> been a disappointment. I mean, I've used IDLDE on another Unix system so I
> wasn't expecting too much of it. (I planned to use IDLWAVE in any case). But
> object graphics rendering on the Linux side is unusably slow! For example I
> have an object-graphics animation example program that presents a series of
> 25 x 25 IDLgrSurface objects. It runs along at a tolerable 15 frames per
> second on Windows but barely manages 2 frames per second under Linux. Using
> software rendering on Linux seems to speed things up slightly, but not much.
>
> The PC has a Pentium 3 800 Mhz processor with an Intel 815 built-in graphics
> controller. I run 16 colours, 1280 x 1024 on both OSes. The Windows OS is
> Windows 2000 and the Linux one is Redhat 7.2 (kernel 2.4.7-10). The system
> has oodles of RAM and disk space.
>
> Is there anything I can do to improve OG performance under Linux?
>
> ---
> Mark Hadfield
> m.hadfield@niwa.co.nz  http://katipo.niwa.co.nz/~hadfield
> National Institute for Water and Atmospheric Research
>
>
>


--
Steve S.

steve@NOSPAMmailaps.org
remove NOSPAM before replying

---

Subject: Re: Object graphics under Linux: are they supposed to be that slow?
Posted by Mark Hadfield on Tue, 29 Jan 2002 01:55:49 GMT
View Forum Message <> Reply to Message

"Steve Smith<steven_smith>" <nobody@nowhere.com> wrote in message
news:slrna5be3l.qih.nobody@pooh.nrel.gov...
>  Hi Mark:
>  I'm not sure I've the answer for you, but I suspect that the big
difference
>  is in the graphics subsystem, the windows more than likely supporting the
>  hardware acceleration and Linux (XFree I'm assuming) not....

I have had email responses from a few people. They tell me that IDL 5.5 on
Linux does not use the system's OpenGL drivers even if they are available.

This is because in RSI's judgement the hardware support on Linux is (as you note) rather flaky. This is likely to change in future versions.

I suspect that on my system the graphics driver is so crippled that hardware rendering would not help IDL much. It doesn't make much difference under Windows. Windows describes the graphics controller as an "Intel 810 Graphics Controller Hub" with 4MB memory. That doesn't sound like much video RAM, does it? However I believe the Intel 810 can access system RAM. Anyway, it's quite possible that performance would improve significantly if I reduced the screen resolution. I will look into this and report back to the group.

> I don't know what your reasons for moving to Linux were...

I'm not sure myself! Better integration with other Unix systems mostly. I haven't really moved to Linux yet, just dipped my toe in the water and run back shivering. I haven't yet got my Linux setup to the stage where I can actually do productive work with it.

---
Mark Hadfield
m.hadfield@niwa.co.nz http://katipo.niwa.co.nz/~hadfield
National Institute for Water and Atmospheric Research

---

Subject: Re: Object graphics under Linux: are they supposed to be that slow?
Posted by nobody@nowhere.com (S on Tue, 29 Jan 2002 16:25:52 GMT
View Forum Message <> Reply to Message

On Tue, 29 Jan 2002 14:55:49 +1300, Mark Hadfield <m.hadfield@niwa.co.nz> wrote:
> "Steve Smith<steven_smith>" <nobody@nowhere.com> wrote in message
> news:slrna5be3l.qih.nobody@pooh.nrel.gov...
>> Hi Mark:
>> I'm not sure I've the answer for you, but I suspect that the big
> difference
>> is in the graphics subsystem, the windows more than likely supporting the
>> hardware acceleration and Linux (XFree I'm assuming) not....
>
> I have had email responses from a few people. They tell me that IDL 5.5 on
> Linux does not use the system's OpenGL drivers even if they are available.
> This is because in RSI's judgement the hardware support on Linux is (as you
> note) rather flaky. This is likely to change in future versions.
>
At least in previous versions of IDL (up to 5.2), I've heard that IDL uses the hw-accel if it's available, but maybe that's changed. I always found the idl demo Open GL stuff to be pretty snappy, but I've no 3-D apps. If you're using XFree, I guess you should say XFree doesn't support hw-accel graphics Open GL or DRI with the same scope as Windows, since XFree is not Linux (and you don't have to use it, there are commercial Xservers that will support most

cards).
> I suspect that on my system the graphics driver is so crippled that hardware
> rendering would not help IDL much. It doesn't make much difference under
> Windows. Windows describes the graphics controller as an "Intel 810 Graphics
> Controller Hub" with 4MB memory. That doesn't sound like much video RAM,
> does it? However I believe the Intel 810 can access system RAM. Anyway, it's
> quite possible that performance would improve significantly if I reduced the
> screen resolution. I will look into this and report back to the group.
>

I would be suprised if it didn't improve at lower screen resolutions. I would
also look at the rest of the system and see that all your memory is being used,
that you don't have a lot of crap running that doesn't need to (RedHat installs
a lot of things you DO NOT need and usually don't want). The strong points of
Linux and other open source OS's is that you can build your system from the
ground up and you have control over everything. I didn't mean to say that any
thing about Linux per say was 'flaky'. As a windows user, I'm sure the word is
not foreign to you either (plenty of 'flaky' windows apps, subsystems). It is
just that graphics is not a traditional strong point of Unix, except in the
case of SGI, where they had complete control over the hardware and the
software.With PC's, only in the last few years have they had access to high
performance graphics systems, and MS-Windows has _conspired_ to have a similar
control over both hardware and software ends of the graphics subsystem, because
of the potential market for 3D gaming, for instance. Meanwhile, the open source
mostly volunteer XFree project has endeavored to try to incorporate much of the
rapidly changing hardware into X, sometimes by reverse engineering, a daunting
task do doubt. In the end, if you can get good hw-accelerated graphics on
a Linux box, it's by good fortune. You would have access to open source Open GL
toolkits and could probably do some pretty neat stuff with it, but for the
casual user this isn't useful.

>>  I don't know what your reasons for moving to Linux were...
>

For me, it's in what they say: "... use the source, Luke!"

> I'm not sure myself! Better integration with other Unix systems mostly. I
> haven't really moved to Linux yet, just dipped my toe in the water and run
> back shivering. I haven't yet got my Linux setup to the stage where I can
> actually do productive work with it.
>
> ---
> Mark Hadfield
> m.hadfield@niwa.co.nz  http://katipo.niwa.co.nz/~hadfield
> National Institute for Water and Atmospheric Research
>
>
>

--
Steve S.

steve@NOSPAMmailaps.org
remove NOSPAM before replying

---

Subject: Re: Object graphics under Linux: are they supposed to be that slow?
Posted by Mark Hadfield on Wed, 30 Jan 2002 03:00:02 GMT
View Forum Message <> Reply to Message

Several people have asked me for a copy of the program on which I based my assessment that IDL object graphics is considerably slower in Linux than in Windows. I haven't made it available yet, because it is entangled deeply in my IDL libraries; it doesn't make a very good benchmark anyway because it is event-driven and it's hard to include the timing functions. What I do have is an object-graphics benchmark program called TIME_TEST_GR2. This was originally written by Randall Frank and copyright is held by RSI. I am making it available by kind permission of Randall Frank and Karl Schultz:

http://katipo.niwa.co.nz/~hadfield/gust/software/misc/time_t est_gr2.pro

This version is functionally identical to the original but prints out a little more information about the window and device properties.

Perhaps TIME_TEST_GR2 could be part of JD's revised benchmark suite? However it does have the drawback that it takes quite a while to run (~ 10 minutes) and you can't do much else on the system while it's running.

I have also produced 3 files of TIME_TEST_GR2 output generated on my system:

http://katipo.niwa.co.nz/~hadfield/gust/software/misc/ttgr2_ Windows.txt

http://katipo.niwa.co.nz/~hadfield/gust/software/misc/ttgr2_ Windows_Renderer _1.txt
http://katipo.niwa.co.nz/~hadfield/gust/software/misc/ttgr2_ Linux.txt

The configurations are:

* IDL 5.5 on Windows 2000 using RENDERER=0 (hardware)
* IDL 5.5 on Windows 2000 using RENDERER=1 (software)
* IDL 5.5 on Linux. This uses RENDERER=0 but, as is obvious from the DeviceInfo string, the rendering is carried out by the Mesa

software library and does not access any hardware acceleration

The geometric-mean elapsed time figure provides a rough ranking of the configurations:

| | |
|---|---|
| Windows RENDERER=0 | 4.58 s |
| Windows RENDERER=1 | 3.11 s |
| Linux | 5.55 s |

The surprise here is that Windows is slower with RENDERER=0. So much for hardware acceleration! This suggests my graphics controller is no ball of fire. Linux lags behind both the Windows configurations, though not by as much as my original assessment suggested.

I have checked that the Windows and Linux configurations are as similar as possible. The screen dimensions (1280 x 1024) and colour depth (16 bits) are the same in both. I think I may have already mentioned that the PC's graphics controller (Intel 810) has only 4 MB on-board memory (at least that's what the Windows Display properties applet tells me).

I intend to investigate the effect of reducing the number of pixels on the screen. I also intend to investigate the effect of grovelling to our IT people for some better graphics hardware.

The detailed results to TIME_TEST_GR2 are interesting. Windows with RENDERER=0 is notably slow in the instancing test. Linux is notably slow in the polygon test.

---
Mark Hadfield
m.hadfield@niwa.co.nz  http://katipo.niwa.co.nz/~hadfield
National Institute for Water and Atmospheric Research

---

Subject: Re: Object graphics under Linux: are they supposed to be that slow?
Posted by David Fanning on Wed, 30 Jan 2002 04:55:23 GMT
View Forum Message <> Reply to Message

Mark Hadfield (m.hadfield@niwa.co.nz) writes:

> The configurations are:
>
>  * IDL 5.5 on Windows 2000 using RENDERER=0 (hardware)
>  * IDL 5.5 on Windows 2000 using RENDERER=1 (software)
>  * IDL 5.5 on Linux. This uses RENDERER=0 but, as is obvious from
>  the DeviceInfo string, the rendering is carried out by the Mesa
>  software library and does not access any hardware acceleration

>
> The geometric-mean elapsed time figure provides a rough ranking of the
> configurations:
>
>  Windows RENDERER=0  4.58 s
>  Windows RENDERER=1  3.11 s
>  Linux  5.55 s

Just to give you something to chew over, Mark. Here are
my results with IDL 5.5 on Windows 2000, with a 32MB
NVIDIA GeForce 2GTS graphics card. Screen resolution
is 1280 by 1024 at 32 bits True-Color.

  Windows RENDERER=0  0.71 s
  Windows RENDERER=1  1.34 s

That graphics card was a couple of hundred bucks, as
I recall. :-)

Cheers,

David

--
David W. Fanning, Ph.D.
Fanning Software Consulting
Phone: 970-221-0438, E-mail: david@dfanning.com
Coyote's Guide to IDL Programming: http://www.dfanning.com/
Toll-Free IDL Book Orders: 1-888-461-0155

---

Subject: Re: Object graphics under Linux: are they supposed to be that slow?
Posted by karl_schultz on Wed, 30 Jan 2002 18:23:09 GMT
View Forum Message <> Reply to Message

David Fanning <david@dfanning.com> wrote in message
news:<MPG.16c12aa8f85252b59897e1@news.frii.com>...
> Mark Hadfield (m.hadfield@niwa.co.nz) writes:
>
>> The configurations are:
>>
>>  * IDL 5.5 on Windows 2000 using RENDERER=0 (hardware)
>>  * IDL 5.5 on Windows 2000 using RENDERER=1 (software)
>>  * IDL 5.5 on Linux. This uses RENDERER=0 but, as is obvious from
>>  the DeviceInfo string, the rendering is carried out by the Mesa
>>  software library and does not access any hardware acceleration
>>
>> The geometric-mean elapsed time figure provides a rough ranking of the

>> configurations:
>>
>> Windows RENDERER=0 4.58 s
>> Windows RENDERER=1 3.11 s
>> Linux 5.55 s
>
> Just to give you something to chew over, Mark. Here are
> my results with IDL 5.5 on Windows 2000, with a 32MB
> NVIDIA GeForce 2GTS graphics card. Screen resolution
> is 1280 by 1024 at 32 bits True-Color.
>
> Windows RENDERER=0 0.71 s
> Windows RENDERER=1 1.34 s
>
> That graphics card was a couple of hundred bucks, as
> I recall. :-)
>
> Cheers,
>
> David

Here's another data point:

PIII 750Mhz Windows NT 4.
nVidia RIVA TNT2 AGP SSE. 32 bits/pixel

I'd characterize this graphics card as a medium-low range card today.
The driver uses the AGP port, which is good, and apparently leverages
the Intel SSE instructions, which is also good.

Windows RENDERER=0 9.41 s
Windows RENDERER=1 2.72 s

The hardware was MUCH slower at image operations. (GL is generally
not a good image processor) In fact, I had to disable the image
STRETCH test because it was taking way too long. The hardware really
only beat out the software at texture-mapped polygons. I suppose that
this hardware/driver package was tuned for the Quake-like games :-).
(I should check for a driver update)

While this is a great benchmark, it may not be representative of
typical IDL application usage of graphics. For example, I might think
that this test is a little heavy on images. This program was written
to monitor the object graphics performance during development and
later modification and so hits most aspects of object graphics.
Therefore this program probably isn't the best means to select a card
or even in deciding between hardware and software rendering. Looking
at the individual test results can help a bit more if you know what

sort of things you are drawing a lot.

For example, in this case, removing the image tests would probably bring the hardware and software numbers closer together. And that would be important to me if my programs didn't use IDLgrImage very much.

Karl
RSI

---