Subject: plot function performance
Posted by woodrask on Thu, 20 Aug 2015 00:01:11 GMT
View Forum Message <> Reply to Message

Has anyone else noticed a discrepancy in the time it takes to call a plot function in the various different OS's? I'm noticing that a fire-breathing linux server is slower than my old mac desktop.

For background, I have an old linux server that I need to replace. I have access to a really fast linux server, too and tried to figure out how much faster it would be. Unfortunately, when using function graphics they seem to be extremely similar in performance. This makes me think IDL has a problem with linux in general.

Here is an example that shows a substantial difference in creating one plot. The first time takes longer than successive calls. Why?

```
IDL Version 8.2.2, Mac OS X (darwin x86_64 m64). (c) 2012, Exelis Visual Information Solutions, Inc.
Installation number:
Licensed for use by:

IDL> x=systime(1) & p=plot(findgen(10)) & print,systime(1)-x
% Loaded DLM: PNG.
       1.9941020
IDL> x=systime(1) & p=plot(findgen(10)) & print,systime(1)-x
      0.19571400
IDL> x=systime(1) & p=plot(findgen(10)) & print,systime(1)-x
      0.20434308
```

Here is the same thing done on a significantly faster linux server.

```
IDL Version 8.4 (linux x86_64 m64). (c) 2014, Exelis Visual Information Solutions, Inc.
Installation number:
Licensed for use by:

IDL> x=systime(1) & p=plot(findgen(10)) & print,systime(1)-x
% Loaded DLM: XML.
% Loaded DLM: PNG.
      6.0436580
IDL> x=systime(1) & p=plot(findgen(10)) & print,systime(1)-x
      2.2157071
IDL> x=systime(1) & p=plot(findgen(10)) & print,systime(1)-x
      2.3254170
```

Here is the same thing on my old, infirmed, linux server:

```
IDL Version 8.2 (linux x86_64 m64). (c) 2012, Exelis Visual Information Solutions, Inc.
Installation number:
Licensed for use by:
```

```
IDL> x=systime(1) & p=plot(findgen(10)) & print,systime(1)-x
% Loaded DLM: XML.
      6.2218139
IDL> x=systime(1) & p=plot(findgen(10)) & print,systime(1)-x
      2.1858058
IDL> x=systime(1) & p=plot(findgen(10)) & print,systime(1)-x
      1.9238040
```

The first plot took about 6 seconds to create under linux (both new and old servers) but only 2 seconds for my old mac. Notice that the linux IDL 8.4 is compiling a DLM that mac OS is not (notice that linux with IDL 8.2 is also different).

After that first call the successive plots show a factor of 10 difference with Mac winning. What gives? Why is IDL new graphics on linux so slow? I've done the same test on Windows and it's comparable to the Mac. Also it does not seem to matter if it's the workbench or the command line, I get the same behavior.

If I want to create lots of graphics in linux what should I buy? A faster processor doesn't seem to make any difference, and new and old linux machines are the same speed.

Thanks in advance,
Don

Subject: Re: plot function performance
Posted by chris_torrence@NOSPAM on Thu, 20 Aug 2015 17:54:53 GMT

View Forum Message <> Reply to Message

On Wednesday, August 19, 2015 at 6:01:14 PM UTC-6, wood...@colorado.edu wrote:
> Has anyone else noticed a discrepancy in the time it takes to call a plot function in the various different OS's? I'm noticing that a fire-breathing linux server is slower than my old mac desktop.
>
> For background, I have an old linux server that I need to replace. I have access to a really fast linux server, too and tried to figure out how much faster it would be. Unfortunately, when using function graphics they seem to be extremely similar in performance. This makes me think IDL has a problem with linux in general.
>
> Here is an example that shows a substantial difference in creating one plot. The first time takes longer than successive calls. Why?
>
> IDL Version 8.2.2, Mac OS X (darwin x86_64 m64). (c) 2012, Exelis Visual Information Solutions, Inc.
> Installation number:
> Licensed for use by:
>
> IDL> x=systime(1) & p=plot(findgen(10)) & print,systime(1)-x

```
>  % Loaded DLM: PNG.
>        1.9941020
>  IDL> x=systime(1) & p=plot(findgen(10)) & print,systime(1)-x
>       0.19571400
>  IDL> x=systime(1) & p=plot(findgen(10)) & print,systime(1)-x
>       0.20434308
>
```

> Here is the same thing done on a significantly faster linux server.
>

```
>  IDL Version 8.4 (linux x86_64 m64). (c) 2014, Exelis Visual Information Solutions, Inc.
>  Installation number:
>  Licensed for use by:
>
>  IDL> x=systime(1) & p=plot(findgen(10)) & print,systime(1)-x
>  % Loaded DLM: XML.
>  % Loaded DLM: PNG.
>       6.0436580
>  IDL> x=systime(1) & p=plot(findgen(10)) & print,systime(1)-x
>       2.2157071
>  IDL> x=systime(1) & p=plot(findgen(10)) & print,systime(1)-x
>       2.3254170
>
```

> Here is the same thing on my old, infirmed, linux server:
>

```
>  IDL Version 8.2 (linux x86_64 m64). (c) 2012, Exelis Visual Information Solutions, Inc.
>  Installation number:
>  Licensed for use by:
>
>  IDL> x=systime(1) & p=plot(findgen(10)) & print,systime(1)-x
>  % Loaded DLM: XML.
>       6.2218139
>  IDL> x=systime(1) & p=plot(findgen(10)) & print,systime(1)-x
>       2.1858058
>  IDL> x=systime(1) & p=plot(findgen(10)) & print,systime(1)-x
>       1.9238040
>
>
```

> The first plot took about 6 seconds to create under linux (both new and old servers) but only 2 seconds for my old mac. Notice that the linux IDL 8.4 is compiling a DLM that mac OS is not (notice that linux with IDL 8.2 is also different).
>
> After that first call the successive plots show a factor of 10 difference with Mac winning. What gives? Why is IDL new graphics on linux so slow? I've done the same test on Windows and it's comparable to the Mac. Also it does not seem to matter if it's the workbench or the command line, I get the same behavior.
>
> If I want to create lots of graphics in linux what should I buy? A faster processor doesn't seem to make any difference, and new and old linux machines are the same speed.

>
> Thanks in advance,
> Don

Hi Don,

For the initial calls, the bottleneck is compiling the pro code. So it really depends upon where your files are stored, how fast your network is (if they are on a network drive), or how fast your hard drive is (if they are local files).

For subsequent calls, part of the time is taken in creating the IDL objects, and the rest will be for OpenGL calls (or Mesa for software rendering). So you might need to look at your graphics card in all of the machines.

Try setting renderer=1 when making those calls, to force software rendering. That should eliminate the difference between graphics cards.

Hope this helps.

-Chris