
Subject: Re: Graphics problem on Sun Ultra2 workstation
Posted by [Karl Schultz](#) on Thu, 14 Aug 2003 22:42:55 GMT
[View Forum Message](#) <> [Reply to Message](#)

"Alex Schuster" <Wonko@wonkology.org> wrote in message
news:8rp\$HkNud8B@wonkology.org...
> timm.weitkamp@nowhere.edu (Timm Weitkamp) writes:
>
>> Alex and Jim,
>
>> I have little knowledge about the hardware internals of computer
>> graphics, but I have seen the phenomenon described by Alex lots of times
>> on different kinds workstations (HP, Sun...). It occurs not only with
>> IDL.
>
>> I am pretty sure that this is simply the way that their 8-bit visuals
>> deal with any application's attempt to allocate a larger number of
colors
>> than is left free by the other applications (including the window
>> manager). In this case, the application gets a "private colormap", and
>> every time the focus is put in the application's window, the entire
>> screen switches to this colormap, inevitably messing up the rest of the
>> screen. In other words: it's a feature, not a bug.
>
>> I'd be surprised if it had anything to do with the refreshing rate.
>
>> Alex, your Sun probably has Netscape installed, right? Try start it by
>
> Sort of, but the times when I used SUNs to view Web sites are over, I
> prefer PCs for that now. Netscape is painfully slow, and crashes way
> too often. Mozilla is okay, but even slower.
> I used to do the -install trick with Netscape some years ago, but my
> problem happens without any color-using applications. There is just
> OpenWindows running. I also tried CDE, even in black-white mode, but
> this does not change the color flashing problem.
>
> When I start Netscape, I can see true-color images. Using xcolor to
> display the colormap, I see no change here. I can also open another
> application, using ~200 colors, without color flickering. But I can see
> the color map changing then.

How many colors do you think your IDL application is trying to obtain?

It is possible that this application asked for 200 colors, but may have
asked for read-only cells, which can be shared with other clients. IDL
would need private read/write cells, since you can change IDL's color table.

> There are a couple of SUNs here I am working with. Sparc5 machines,

- > sparc10, and some Ultras. But only this Ultra-2 machines gives the
- > problems. With all other machines I can prevent the flickering (if it
- > happens at all) by doing a DEVICE, PSEUDO=8 or DEVICE, DECOMPOSED=0
- > before opening any graphics window.

You may not be seeing flickering on the other machines because they might have multiple hardware color tables. The "number of colormaps" line from xdpinfo tells you how many you have. Most X servers say:

number of colormaps: minimum 1, maximum 1

Higher-end graphics cards and X servers may have more than one, which allows the server to install more than one colormap at a time. The display hardware switches between color tables as it scans out the raster image to the display. This delays the color table flashing problem up to the point that the hardware color tables are exhausted.

- > But looking at the output of xdpinfo, I just noted that the default
- > visual ID is 0x20, and this means PseudoColor. Although there are a
- > couple of TrueColor visuals. But this may be normal, I see a similar
- > output on another Sun.

Even if there are TrueColor visuals on the server, your application requires PseudoColor, presumably because the application changes colors with the color table without redrawing the scene. So, the TrueColor visual won't help here.

I still think that you are having problems with something on your machine using up colormap cells and your application is requesting more colors than are available.

One thing you can try is

```
WINDOW, COLORS = -1
```

which tells IDL to allocate all the unallocated colors (except one). You'll get whatever is left over, but that may not be enough for the application if the app is coded to require a certain number of colors. You can query the IDL window to see how many colors you ended up with.

See <http://www.rsinc.com/services/techtip.asp?ttid=1688> for more info.

One possibility regarding the number of cells that are taken up before running any color-hungry applications is the existence of standard colormaps. It is possible to configure the X server, window manager, or desktop to fill part or all of a colormap with a standard colormap. A standard colormap puts a range of popular or common colors in the colormap as read-only cells with the hope of X clients sharing these read-only cells

instead of allocating their own. (Ref man stdcmap) This only works if the app can tolerate a cell that is "close-enough" and does not need to modify that cell. Sometimes you can spot this with a client like xcolor or xcmmap by looking for a "ramp" pattern of colors in the colormap. You can probably also use xprop on the root window to look for std cmmaps.

Another last resort is to figure out how to start your X server in a minimal way without a desktop and some simple window manager like twm. That might help you isolate the component that is using up color cells.

Karl
