

---

Subject: Re: G4-related problem  
Posted by [DPJ](#) on Sat, 04 Oct 2003 13:01:15 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

Hi all,

I am a very new IDL user (novice) and I think I am running into this G4 problem that people have been discussing. I am very disappointed since the very first thing I tried to do with IDL and it seemed to result in utter nonsense. I am working on a dual processor G4 running OSX 10.2.(6?)

Here is what I noticed.

I have an ascii file with two columns of tab delimited floating point data (x and y). I have no trouble reading in this data and making a plot of it. This seems to be true whether there is 5000 lines of data or 50,000 lines of data (I have not tried to rigorously test out if it would fail at some point).

However, if I try to read a file with 10 columns of data (x1 y1 x2 y2...), the following happens. The first two columns read in fine but the following columns do not. This seems to be true no matter how many lines of data there are.

They all read in fine up to some point and then there is what appears to be random integer data for a while, and then good data again. It appears as though columns 3-9 always have good data up to item 256. That is, data.x3[255] is good while data.x3[256] is bad. Column 10 appears to "go bad" at item 128.

I am really feeling screwed. Going back to OSX 10.1 is not an option. I don't think my program is doing any graphics before/while this data input is going on (although it does plot the data after it has been read). I will try the workaround that Karl suggested to turn off the timer and hopefully that will work. If anyone has any additional suggestions, please let me know. (Please remember that I am a novice if you respond.) This is important because I have a ton of these data files to analyze.

David

"Karl Schultz" <[kschultz\\_no\\_spam@rsinc.com](mailto:kschultz_no_spam@rsinc.com)> wrote in message [news:vlc5b17e31l6f8@corp.supernews.com](mailto:news:vlc5b17e31l6f8@corp.supernews.com)...

> Update:

>

> We heard back from Apple and they were able to find the problem via a bug report I filed.

>

> They said that the problem does exist on 10.2 and also in 10.3. In the

### 10.3

> case, another issue hid the actual bug, which led us to believe that the  
> problem didn't exist in 10.3. In any case, the problem will not appear in  
> future OS X releases. The fix has been rolled into the current OS X 10.3  
> builds. It is unclear if there is going to be another 10.2.x release and  
if  
> this fix will be in it. (If there is another 10.2.x (10.2.7 ?), look for  
a  
> mention of this in the "fixed issues" list when you install the update).  
>  
> Adding together what I observed and what they told us, I believe that the  
> cause of the problem was Altivec state not being handled correctly when  
the  
> OS X kernel interrupts the main thread to run the timer signal handler  
> thread. OS X uses the Altivec to implement the bcopy function, and so any  
> changes to Altivec state that may occur in the signal handler would cause  
an  
> interrupted bcopy call to malfunction.  
>  
> Anyway, the problem is understood and the workarounds I suggested below  
are  
> still valid.  
>  
> Karl  
>  
> "Karl Schultz" <kschultz\_no\_spam@rsinc.com> wrote in message  
> news:vjq75m7imf90a2@corp.supernews.com...  
>>

>> news:bh8393\$2as\$1@surz18.uni-marburg.de...  
>>> Hi.  
>>>  
>>> I recently discovered a very nasty problem on my G4 (2\*867, Mac OS X  
>>> 10.2.6) running IDL 5.6.  
>>>  
>>> The following code should not produce any output, if everything works.  
>>> Unfortunately, this is not the case on my G4, where the variable  
>>> assignment "data1=data0" produces messy data in a randomly fashion. So  
>>> it works most of the time, but once in a while, the data in "data1"  
will  
>>> be corrupted. Typically, there are some zeros showing up in the array  
>>> and some other garbage.  
>>>  
>>> Here is the code in question:  
>>>  
>>> data0=fltarr(10000)+1.  
>>> for ii=0L, 200000 do begin & \$  
>>> data1=data0 & \$

```

>>> if min(where(data1 eq 0)) gt 0 then begin & $
>>>   print, ii & ii=200001L & endif & $
>>> endfor
>>>
>>>
>>> This problem did not show up on other machines I have tested (Compaq
>>> Alpha, a Windows PC or my iMac G3 running the same OS and IDL
>>> versions).
>>>
>>> So this can either be a G4-specific bug in IDL 5.6, a G4-specific bug
>>> in
>>> Mac OS X 10.2.6, or a hardware problem on my G4.
>>>
>> Aside from the Altivec code, there is no G4-specific code in IDL. Since
> the
>> Altivec is not used for simple copy operations and you disabled it, that
>> can't be the problem. (But good thinking on disabling it)
>>
>>> Since I do not have a second G4 available, I would like to ask anyone
>>> with a G4 to test the code snippet given above. You can simply paste
>>> the
>>> code to the command line in IDL.
>>>
>>>
>>> PS: What I already tested:
>>> - setting TPOOL_NTHREADS=1 and VECTOR_ENABLE=0
>>> - using different RAM
>>> - replacing the RAM modules
>>> - testing the RAM with the hardware test CD
>>>
>> I've spent a bunch of time on this and I think I can help a little here.
>>
>> - The problem seems to only occur on OS X 10.2. I don't know if the
>> story
>> changes depending on the version of 10.2 (e.g., 10.2.3). The problem
>> does
>> not seem to occur on 10.1 and the developer's preview of 10.3. This
>> last
>> > it
>> em is good news.
>> - I've only seen or heard the problem reported on G4.
>> - The flavor of the X11 server involved does not matter. The IDL
>> program
>> needs to be connected to *any* X server to demonstrate the problem.
>> That
>> is, the issue is all on the client side.
>>
>>

```

>> Given those things, the problem is on OS X 10.2 G4's while IDL has an X  
>> connection open.  
>>  
>> On some systems (OS X is one), IDL uses a 1-second interval timer proc  
> that  
>> runs whenever there is an X connection. Its job is to keep your  
graphics  
>> windows reasonably up to date while IDL is performing some long  
operation,  
>> between interpreter instructions. Disabling this timer operation keeps  
> the  
>> problem from occurring, but at the (minor?) expense of unresponsive  
windows  
>> during these operations.  
>>  
>> So, possible solutions are:  
>> 1) Use 10.1 or wait for 10.3.  
>> 2) Issue (the undocumented) DEVICE, /NOTIMER to turn off the timer proc.  
>> You'll have to judge if this is ok for your app or not.  
>> 3) Make sure your program does not do any graphics. Note that if you  
have  
>> some DEVICE command in your startup file to configure your windows, this  
>> still makes a connection, even if you don't open any graphic or UI  
> windows.  
>>  
>> If you aren't sure if your machine suffers from the problem, I'd suggest  
>> running the program above.  
>>  
>> Hope this helps,  
>> Karl  
>>  
>>  
>>  
>  
>

---