Subject: Scope_Varfetch "IDL workbench" train wreck
Posted by wlandsman on Wed, 18 Nov 2009 18:19:18 GMT
View Forum Message <> Reply to Message

I recently received a note from a user saying that my procedure
readcol.pro
(http://idlastro.gsfc.nasa.gov/ftp/pro/misc/readcol.pro) - one of
many such procedures to read an ASCII file into IDL variables -- was
taking ~100 times longer to run from the IDL workbench than from the
IDL command prompt. I of course replied that this was complete
nonsense, and that the interface used to call the procedure shouldn't
affect its speed.

Of course, when I tested this, a run that took under 1 second from the
command prompt completely hung my Mac (x86_64 darwin unix Mac OS X
7.1 Apr 21 2009 64 64) when run from the Workbench. My Linux box
was not quite as bad but also eventually hung with messages such as

java.lang.OutOFMemoryError: unable to create new native thread
java.lang.ArrayIndexOutOfBoundsException: 2601

The culprit apparently is my use of SCOPE_VARFETCH. readcol.pro
calls SCOPE_VARFETCH thousands of time for a large file, since it
writes data directly into the output variables, and uses a
SCOPE_VARFETCH call for each item written.

Below is a little test program to be called with
IDL> test, x1,x2,x3,x4,x5,x6,x7,x8,x9

I find it runs instantly from the IDL command line but takes very long
or hangs the IDL workbench. --Wayne

```
pro test,a,b,c,d,e,f,g,h,j

vv = ['a','b','c','d','e','f','g','h','j']
for k=0,8 do begin                      ;Create output variables
   res = execute(vv[k] + '=fltarr(5000)' )
endfor
for jj=0,4999 do for k=0,8 do (scope_varfetch(vv[k],Level=0))[jj]=
float(jj)
return
end
```