
Subject: Re: Speeding up data crunching using IDL_IDLBridge with asynchronous execution

Posted by [Russell Ryan](#) on Wed, 10 Jul 2013 17:26:09 GMT

[View Forum Message](#) <> [Reply to Message](#)

On Wednesday, July 10, 2013 12:55:12 PM UTC-4, nata wrote:

> Hi Russell,

>

>

>

> The memory leak you are talking about is it related to the variables defined at the end of the execution?

>

>

>

> If you execute something, after getting the result, the variables are not destroyed and they keep using memory. To prevent that I always execute the following commands:

>

>

>

> all_var='tmp_var'

>

> IDLBridge->Execute, all_var+='ROUTINE_INFO("\$MAIN\$"/VARIABLES)'

>

> all_var=self->IDL_IDLBridge::GetVar(all_var)

>

>

>

> command='DELVAR, '+STRJOIN(all_var,', ')

>

> IDLBridge->Execute, command

>

>

>

> Are you talking about something different?

>

> I guess that using "help, /memory" I would be able to track the usage of memory and see if the bridge definitely has a bug.

>

>

>

> nata

@Chris. Yes, by "triggering", I mean a call to the EXECUTE method.

@Nata. No, that doesn't fix the problem. Here is a simple script to try. You can see where I added your suggestion to the test code that Exelis asked me to try. You can see that the memory usage continues to increase with time. Again, it's a small increase --- but the point is it should

read 0. And if you run this long enough, that small bit *WILL* cause problems, trust me. I lost a very long time "debugging" my "bridged" code before I just emailed Exelis. They confirmed it was a problem, and I "unbridged" my code. I thought to run the bridged version for a long time (but not long enough for the leak to be a problem), then save the results, end IDL, restart IDL, start bridged code with old outputs. Heck, you could put this in a csh script or something and so it's seamless to me. But, I hated the idea of this, because gosh darn it, it should just work! This "feature" was added in IDL 6, they're on 8, and it's expensive software that shouldn't need a shell-script to "patch" something. I have since rewritten that bit of code in Fortran, which is easily multithreaded and (if necessary) call it from IDL using call_external.

```
pro b43494
  mbegin=(memory())[0]
  for i=0,100 do begin
    m0 = (memory())[0]
    o = obj_new('idl_idlbridge')
    o->Execute, 'x = 1'      ;, /nowait

    ;Nata's additions-----
    all_var='tmp_var'
    o->Execute, all_var+='ROUTINE_INFO('$MAIN$',/VARIABLES)'
    all_var=o->GetVar(all_var)

    command='DELVAR, '+STRJOIN(all_var,', ')
    o->Execute, command
    ;End of Nata's additions-----

    obj_destroy, o
    m1 = (memory())[0]
    if (i gt 0) then begin
      print
      print,'iteration: ',i
      print,'Memory lost in this step thru loop: ', m1 - m0
      print,'Total memory lost since beginning: ',m1-mbegin
      print
    endif
  endfor

end
```

-Russell

PS: This is without using the nowait flag to the EXECUTE method, which frankly is stupid. I mean, why would you do this effort of IDL_IDLBridge if not to *USE* the nowait feature. So, if you uncomment that bit, you will find the leaks are *WORSE*.

PPS: If I've done something wrong, please let me know. But if it's all okay, then I strongly discourage anyone from doing anything with IDL_IDLBridge
