
Subject: Re: IDL_IDLBridge and the virtual machine
Posted by [Helder Marchetto](#) on Thu, 14 Mar 2013 15:21:24 GMT
[View Forum Message](#) <> [Reply to Message](#)

On Thursday, March 14, 2013 3:30:04 PM UTC+1, rr...@stsci.edu wrote:

> On Wednesday, March 13, 2013 7:58:07 AM UTC-4, Helder wrote:

>

>> Hi,

>

>>

>

>> I would like to use the IDL_IDLBridge, but I just read in the documentation that it is not possible to use execute, getvar and setvar methods in VM mode.

>

>>

>

>> This is quite annoying, but is there another functional way to parallelize computations in IDL?

>

>>

>

>>

>

>>

>

>> I basically have a list of n images (n~100, maybe more) and I perform computations on images k with k-1 for the whole list. I wanted to start computations separately:

>

>>

>

>>

>

>>

>

>> Res1 = ComputationFunction(Reflmg=0, lmg=1)

>

>>

>

>> Res2 = ComputationFunction(Reflmg=1, lmg=2)

>

>>

>

>> Res3 = ComputationFunction(Reflmg=2, lmg=3)

>

>>

>

>> ...

>

>>

```
>
>> and put all the results in an array.
>
>>
>
>>
>
>>
>
>> Any suggestions?
>
>>
>
>>
>
>>
>
>> I could try writing a batch process and producing a .sav that puts the result in a file and so
on... but this seems a bit "dirty".
>
>>
>
>>
>
>>
>
>> Thanks,
>
>>
>
>> Helder
>
>
>
> Hi Helder,
>
>
>
> Two things.
>
>
>
> (1) Look at the shmmap and shmvar routines for passing data. You essentially create a shared
memory space and load the data there. Then all the nodes use a single string variable to initialize
that memory space within their scope. Fanning has a good tutorial on this:
>
> http://www.idlcoyote.com/code\_tips/bridge.html
>
```

>
>
> (2) I'm not sure how you're planning on using the IDL_IDLBridge. But if you use the /nowait feature (which, frankly is the main reason for using the Bridge in the first place), then there is a major memory leak which affects (it affects Linux and Mac as far as I can tell). If you only call the bridges a few times (say <100), it'll probably be fine --- the leaked memory will be smaller than your RAM so you'll never notice. But, if you call it many times >1000 --- then the run time *PER NODE* will increase very rapidly. I've emailed Exelis about this and I pinged them about it yesterday (I have some simulation code that would really benefit from the Bridges). I'll post more if I hear anything.
>
>
>
> Good luck,
>
> Russell

Hi Russell,
thanks for the tip.

Let me see if I got your point or not...

What you're trying to tell me is to use shared memory on which different processes share their status and results and the main code launches and waits for these processes to finish running? This way I would avoid using IDL_IDLBridge, and simply Spawn .sav files that read the info from shared memory.

I can see that this could work, but every time I spawn a new .sav file, the VM will up with the splash screen and each process has to be "clicked on".

Thanks for the info on the memory allocation sizes (from the Coyote) and the memory leak problems. However, the latter hopefully does not influence me too much because I have to deal with something like hundreds of processes.

Oh yeah... the idea is that I'm looking for image drift in a stack of n images (up to a few hundreds at a time). Each may take a few seconds to compute and have 4+ processors would speed up things (I hope).

Cheers,
Helder
