Subject: IDL_IDLBridge and the virtual machine
Posted by Helder Marchetto on Wed, 13 Mar 2013 11:58:07 GMT
View Forum Message <> Reply to Message

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(RefImg=0, Img=1)
Res2 = ComputationFunction(RefImg=1, Img=2)
Res3 = ComputationFunction(RefImg=2, Img=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

Subject: Re: IDL_IDLBridge and the virtual machine
Posted by markb77 on Thu, 14 Mar 2013 10:09:43 GMT
View Forum Message <> Reply to Message

I am also interested in doing this, and as far as I know it's not possible.

I think it's ridiculous that in today's world of multi-core CPUs, IDL still has no means of running multithreaded analysis from the VM. This is a major handicap to its use in cpu-intensive applications.

Subject: Re: IDL_IDLBridge and the virtual machine
Posted by Helder Marchetto on Thu, 14 Mar 2013 11:05:24 GMT
View Forum Message <> Reply to Message

On Thursday, March 14, 2013 11:09:43 AM UTC+1, superchromix wrote:
> I am also interested in doing this, and as far as I know it's not
>
> possible.

>
>
>
> I think it's ridiculous that in today's world of multi-core CPUs, IDL
>
> still has no means of running multithreaded analysis from the VM.
>
> This is a major handicap to its use in cpu-intensive applications.

Hi,
thanks for your input.
I'll send an email to Exelis and see what they say about this. I'll keep you up to date when and if I get a response on this.

Cheers,
Helder

Subject: Re: IDL_IDLBridge and the virtual machine
Posted by markb77 on Thu, 14 Mar 2013 12:55:48 GMT
View Forum Message <> Reply to Message

I did some searching and came across the "process_manager" written by
Allard de Wit, which uses "spawn" to spawn multiple IDL processes
which run on different cores.

http://www.exelisvis.com/Default.aspx?tabid=1540&id=1179

Does anyone know if this approach would work via the virtual machine?

best
Mark

Subject: Re: IDL_IDLBridge and the virtual machine
Posted by markb77 on Thu, 14 Mar 2013 13:58:15 GMT
View Forum Message <> Reply to Message

No.. after more digging, it looks like this would not work. You can't
run batch files with the idl virtual machine, which is required with
this approach.

On Mar 14, 1:55 pm, superchromix <mark...@gmail.com> wrote:
> I did some searching and came across the "process_manager" written by
> Allard de Wit, which uses "spawn" to spawn multiple IDL processes
> which run on different cores.
>

> http://www.exelisvis.com/Default.aspx?tabid=1540&id=1179
>
> Does anyone know if this approach would work via the virtual machine?
>
> best
> Mark

Subject: Re: IDL_IDLBridge and the virtual machine
Posted by Russell Ryan on Thu, 14 Mar 2013 14:30:04 GMT
View Forum Message <> Reply to Message

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(RefImg=0, Img=1)
>
> Res2 = ComputationFunction(RefImg=1, Img=2)
>
> Res3 = ComputationFunction(RefImg=2, Img=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

Subject: Re: IDL_IDLBridge and the virtual machine
Posted by Russell Ryan on Thu, 14 Mar 2013 14:30:14 GMT
View Forum Message <> Reply to Message

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(RefImg=0, Img=1)
>
> Res2 = ComputationFunction(RefImg=1, Img=2)

>
> Res3 = ComputationFunction(RefImg=2, Img=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

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(RefImg=0, Img=1)
>
>>
>
>> Res2 = ComputationFunction(RefImg=1, Img=2)
>
>>
>
>> Res3 = ComputationFunction(RefImg=2, Img=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

Subject: Re: IDL_IDLBridge and the virtual machine
Posted by Michael Galloy on Thu, 14 Mar 2013 16:16:57 GMT
View Forum Message <> Reply to Message

On 3/14/13 4:09 AM, superchromix wrote:
>
> I am also interested in doing this, and as far as I know it's not
> possible.
>
> I think it's ridiculous that in today's world of multi-core CPUs, IDL
> still has no means of running multithreaded analysis from the VM.
> This is a major handicap to its use in cpu-intensive applications.
>

Well, the VM is free, so it might be a lot to expect to be able to compile code from it as well (although the thread pool is available in the VM, so you automatically use multiple cores for supported operations). Also, remember that runtime licenses are cheaper than development licenses and can perform EXECUTE.

Mike
--
Michael Galloy
www.michaelgalloy.com
Modern IDL: A Guide to IDL Programming (http://modernidl.idldev.com)
Research Mathematician
Tech-X Corporation

Subject: Re: IDL_IDLBridge and the virtual machine
Posted by Helder Marchetto on Thu, 14 Mar 2013 16:40:49 GMT
View Forum Message <> Reply to Message

On Thursday, March 14, 2013 5:16:57 PM UTC+1, Mike Galloy wrote:
> On 3/14/13 4:09 AM, superchromix wrote:
>
>>
>
>> I am also interested in doing this, and as far as I know it's not
>
>> possible.
>
>>
>
>> I think it's ridiculous that in today's world of multi-core CPUs, IDL
>
>> still has no means of running multithreaded analysis from the VM.
>
>> This is a major handicap to its use in cpu-intensive applications.
>
>>
>
>
>
> Well, the VM is free, so it might be a lot to expect to be able to
>
> compile code from it as well (although the thread pool is available in
>
> the VM, so you automatically use multiple cores for supported
>
> operations). Also, remember that runtime licenses are cheaper than

> development licenses and can perform EXECUTE.
>
>
>
> Mike
>
> --
>
> Michael Galloy
>
> www.michaelgalloy.com
>
> Modern IDL: A Guide to IDL Programming (http://modernidl.idldev.com)
>
> Research Mathematician
>
> Tech-X Corporation

Hi Mike,
yes, you're right, I don't expect the VM to execute commands. I would simply like to run code in parallel. This is not my field, so I don't know if there is an *alternative* at all to using an execute command (therefor compiling) to run parallel processes. If an alternative is existing, I don't know about it and I'm digging for this sort of info. For the moment, I'm quite convinced that it is not possible with the current status.
Then the next question would be if something like this is implementable without the execute command. A sort of feature request...

Cheers,
Helder

Subject: Re: IDL_IDLBridge and the virtual machine
Posted by Russell[1] on Thu, 14 Mar 2013 17:26:43 GMT
View Forum Message <> Reply to Message

Hi Helder,

Not exactly. Suppose in your main program you want to initialize 5 bridges. You will need to set (and maybe get) some data to (from) each of them at startup (at completion). You *could* use the SetVar (GetVar) methods to the IDL_IDLBridge, but as you said it can't be used with VM but also those methods are slow as they duplicate the memory usage. Meaning, if you have a data cube that is 1Gb in size that you pass to five bridges, then you have to duplicate that data five times, now ballooning your total RAM usage to 5 Gb. Perhaps that's not an issue for you, but the VM usage seems to be.

The alternative to GetVar and SetVar is to use the shared memory maps. Here any data you want to pass between children and parent processes you will store in a shared memory map.

Then, to use the data from within the bridge, you simply initialize the shared memory map. Here's a simple example:

```
pro test
cd,cur=cwd & cwd=cwd[0]+'/'     ;get the current working directory

data2share=[1.,2.,3]

;create shared memory of the data
dim=size(data2share,/dim)
type=size(data2share,/type)
shmmap,'shareddata',dim=dim,type=type

;we'll need this command to set the data to the bridges, but
;only want to do it one time...
cmd='shmmap,"shareddata",dim='+string(dim,f='(I1)')+',type='+string(type,f='(I2)')



;now populate that shared memory
d=shmvar('shareddata')
d[0]=data2share

n=2                         ;number of bridges


bridges=objarr(n)               ;save the objects
for i=0,n-1 do begin

   ;create the bridge
    bridges[i]=obj_new('IDL_IDLBridge',out=cwd+string(i,f='(I1)' )+'.log')

   ;set the data to the bridge
   bridges[i]->Execute,cmd

   ;restore the data within a bridge
   bridges[i]->Execute,"thisdata = shmvar('shareddata')"

   ;print the data within the bridge
   bridges[i]->Execute,"print,thisdata"
endfor


;destroy the objects
for i=0,n-1 do obj_destroy,bridges[i]
```

end

Now, look in the files 0.log, 1.log, etc. Then you could get the data out via a similar prescription.

This has (one big advantage) for me, is that you do not duplicate any data in the children processes. *BUT* it's also considerably faster than SetVar and GetVar (as described by Fanning). I'm not sure if it'll work with the VM stuff or not, because you still have to use the Execute method (which smells an awful lot like the execute procedure).

-Russell

On Thursday, March 14, 2013 11:21:24 AM UTC-4, Helder wrote:
> 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(RefImg=0, Img=1)
>
>>
>
>>>
>
>>
>
>>> Res2 = ComputationFunction(RefImg=1, Img=2)
>
>>
>
>>>
>
>>
>
>>> Res3 = ComputationFunction(RefImg=2, Img=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

Subject: Re: IDL_IDLBridge and the virtual machine
Posted by Michael Galloy on Thu, 14 Mar 2013 18:56:43 GMT
View Forum Message <> Reply to Message

On 3/14/13 10:40 AM, Helder wrote:
> On Thursday, March 14, 2013 5:16:57 PM UTC+1, Mike Galloy wrote:
>> On 3/14/13 4:09 AM, superchromix wrote:
>>
>>>
>>
>>> I am also interested in doing this, and as far as I know it's
>>> not
>>
>>> possible.
>>
>>>
>>
>>> I think it's ridiculous that in today's world of multi-core CPUs,
>>> IDL
>>
>>> still has no means of running multithreaded analysis from the
>>> VM.
>>
>>> This is a major handicap to its use in cpu-intensive
>>> applications.
>>
>>>
>>
>>
>>
>> Well, the VM is free, so it might be a lot to expect to be able to
>>
>> compile code from it as well (although the thread pool is available
>> in

>>
>> the VM, so you automatically use multiple cores for supported
>>
>> operations). Also, remember that runtime licenses are cheaper than
>>
>> development licenses and can perform EXECUTE.
>>
>>
>>
>> Mike
>>
>> --
>>
>> Michael Galloy
>>
>> www.michaelgalloy.com
>>
>> Modern IDL: A Guide to IDL Programming
>> (http://modernidl.idldev.com)
>>
>> Research Mathematician
>>
>> Tech-X Corporation
>
> Hi Mike, yes, you're right, I don't expect the VM to execute
> commands. I would simply like to run code in parallel. This is not my
> field, so I don't know if there is an *alternative* at all to using
> an execute command (therefor compiling) to run parallel processes. If
> an alternative is existing, I don't know about it and I'm digging for
> this sort of info. For the moment, I'm quite convinced that it is not
> possible with the current status. Then the next question would be if
> something like this is implementable without the execute command. A
> sort of feature request...

Tech-X (disclaimer: I work for Tech-X) has a product, TaskDL, that
allows this type of task farming scenario. Of course, you still need
runtime licenses to run your tasks, but it does allow you to use
multiple cores, or even multiple machines, to runs your tasks.

Mike
--
Michael Galloy
www.michaelgalloy.com
Modern IDL: A Guide to IDL Programming (http://modernidl.idldev.com)
Research Mathematician
Tech-X Corporation

Subject: Re: IDL_IDLBridge and the virtual machine
Posted by markb77 on Thu, 14 Mar 2013 23:59:10 GMT
View Forum Message <> Reply to Message

On Mar 14, 5:16 pm, Michael Galloy <mgal...@gmail.com> wrote:

>
> Well, the VM is free, so it might be a lot to expect to be able to
> compile code from it as well (although the thread pool is available in
> the VM, so you automatically use multiple cores for supported
> operations). Also, remember that runtime licenses are cheaper than
> development licenses and can perform EXECUTE.
>
> Mike

hi Mike,

I think you misread my point. I'm not concerned with the VM being able to compile code. What I want to do is use IDL to run multi-threaded analysis on a multi-core CPU. I am running cpu-intensive analysis algorithms within stand-alone IDL applications which run in the VM, and I need a way to make full use of the processing power of the CPU.

I don't think this is a lot to expect from a modern programming language. I purchase IDL licenses for development work, using the full IDE, etc, and on-the-fly data analysis. The stand-alone applications I write are freely distributed to my scientific collaborators, and there is no possibility of purchasing a license for each and every copy of the distributed stand-alone code. If the free distribution of stand-alone applications is no longer part of the IDL business model, then I guess I should be switching to matlab or python.

Mark

Subject: Re: IDL_IDLBridge and the virtual machine
Posted by Michael Galloy on Fri, 15 Mar 2013 18:38:44 GMT
View Forum Message <> Reply to Message

On 3/14/13 5:59 PM, superchromix wrote:
> On Mar 14, 5:16 pm, Michael Galloy <mgal...@gmail.com> wrote:
>
>>
>> Well, the VM is free, so it might be a lot to expect to be able to
>> compile code from it as well (although the thread pool is available in
>> the VM, so you automatically use multiple cores for supported

>> operations). Also, remember that runtime licenses are cheaper than
>> development licenses and can perform EXECUTE.
>>
>> Mike
>
> hi Mike,
>
> I think you misread my point. I'm not concerned with the VM being
> able to compile code. What I want to do is use IDL to run multi-
> threaded analysis on a multi-core CPU. I am running cpu-intensive
> analysis algorithms within stand-alone IDL applications which run in
> the VM, and I need a way to make full use of the processing power of
> the CPU.
>
> I don't think this is a lot to expect from a modern programming
> language. I purchase IDL licenses for development work, using the
> full IDE, etc, and on-the-fly data analysis. The stand-alone
> applications I write are freely distributed to my scientific
> collaborators, and there is no possibility of purchasing a license for
> each and every copy of the distributed stand-alone code. If the free
> distribution of stand-alone applications is no longer part of the IDL
> business model, then I guess I should be switching to matlab or
> python.

I agree that technologies for parallelization on multi-core and GPU are going to be increasingly important for IDL (single cores aren't getting any faster). But with their current model, multi-core solutions (besides the thread pool) bump up against their licensing model for distributing code. I think the VM shows they want to support free distribution of applications, but it's unclear to me on how to change the VM limitations/licensing exactly that would accomplish this. Also, the average IDL user is probably benefited more by the thread pool since they don't have to even know it exists to get benefits from multi-core.

Mike
--
Michael Galloy
www.michaelgalloy.com
Modern IDL: A Guide to IDL Programming (http://modernidl.idldev.com)
Research Mathematician
Tech-X Corporation

Subject: Re: IDL_IDLBridge and the virtual machine
Posted by Mark Piper on Fri, 15 Mar 2013 21:36:21 GMT
View Forum Message <> Reply to Message

On Thursday, March 14, 2013 5:59:10 PM UTC-6, superchromix wrote:

>
> ... What I want to do is use IDL to run multi-
> threaded analysis on a multi-core CPU. I am running cpu-intensive
> analysis algorithms within stand-alone IDL applications which run in
> the VM, and I need a way to make full use of the processing power of
> the CPU.
>

Just to be clear, IDL's math operations are multithreaded:

http://www.exelisvis.com/docs/Routines_that_Use_the_Th.html

but I agree, we need a generalized technique for multithreading IDL pro code. I should be careful because this is somewhat distant, but we do have a parallel processing API under development that we'd like to introduce in IDL 8.4. (Note that 8.3 is scheduled for this fall.) Email me if you'd like to be a beta tester; same goes for anyone reading this.

We're also looking at better ways to distribute IDL applications. The VM has been OK, but it's time for something better.

mp

Subject: Re: IDL_IDLBridge and the virtual machine
Posted by Russell Ryan on Sun, 17 Mar 2013 05:34:18 GMT
View Forum Message <> Reply to Message

On Friday, March 15, 2013 5:36:21 PM UTC-4, Mark Piper wrote:
> On Thursday, March 14, 2013 5:59:10 PM UTC-6, superchromix wrote:
>
>>
>
>> ... What I want to do is use IDL to run multi-
>
>> threaded analysis on a multi-core CPU. I am running cpu-intensive
>
>> analysis algorithms within stand-alone IDL applications which run in
>
>> the VM, and I need a way to make full use of the processing power of
>
>> the CPU.
>
>>
>
>
>
> Just to be clear, IDL's math operations are multithreaded:
>

>
>
> http://www.exelisvis.com/docs/Routines_that_Use_the_Th.html
>
>
>
> but I agree, we need a generalized technique for multithreading IDL pro code. I should be careful because this is somewhat distant, but we do have a parallel processing API under development that we'd like to introduce in IDL 8.4. (Note that 8.3 is scheduled for this fall.) Email me if you'd like to be a beta tester; same goes for anyone reading this.
>
>
>
> We're also looking at better ways to distribute IDL applications. The VM has been OK, but it's time for something better.
>
>
>
> mp

Hi Mark,
I'd love to beta test such a code. The memory leak bug in the IDL_IDLBridge using the /nowait feature is killing me, and love to learn that there is a new/better way to do this explicitly...
-Russell

Subject: Re: IDL_IDLBridge and the virtual machine
Posted by markb77 on Sun, 17 Mar 2013 14:28:20 GMT
View Forum Message <> Reply to Message

On Mar 15, 10:36 pm, Mark Piper <mpi...@ittvis.com> wrote:
> On Thursday, March 14, 2013 5:59:10 PM UTC-6, superchromix wrote:
>
>> ... What I want to do is use IDL to run multi-
>> threaded analysis on a multi-core CPU. I am running cpu-intensive
>> analysis algorithms within stand-alone IDL applications which run in
>> the VM, and I need a way to make full use of the processing power of
>> the CPU.
>
> Just to be clear, IDL's math operations are multithreaded:
>
> http://www.exelisvis.com/docs/Routines_that_Use_the_Th.html
>
> but I agree, we need a generalized technique for multithreading IDL pro code. I should be careful because this is somewhat distant, but we do have a parallel processing API under development that we'd like to introduce in IDL 8.4. (Note that 8.3 is scheduled for this fall.) Email me if you'd like to be a beta tester; same goes for anyone reading this.

>
> We're also looking at better ways to distribute IDL applications. The VM has been OK, but it's time for something better.
>
> mp

hi Mark,

I would also be interested in beta-testing the parallel api, and
anything related to application distribution.
thanks,
Mark

Subject: Re: IDL_IDLBridge and the virtual machine
Posted by Yngvar Larsen on Mon, 18 Mar 2013 10:17:29 GMT
View Forum Message <> Reply to Message

On Friday, 15 March 2013 22:36:21 UTC+1, Mark Piper wrote:

> but I agree, we need a generalized technique for multithreading IDL pro code. I should be careful
> because this is somewhat distant, but we do have a parallel processing API under development

> that we'd like to introduce in IDL 8.4. (Note that 8.3 is scheduled for this fall.)
> Email me if you'd like to be a beta tester; same goes for anyone reading this.

I'll also volunteer as a beta tester.

Regarding bugs in IDL_IDLBRIDGE:

A year ago, I reported an error in this newsgroup ("idl_idlbridge weirdness on unix systems", 2012-02-28): Using more than 15 bridges simultaneously on a linux 64-bit system causes IDL to hang if the bridges are not destroyed in the exact opposite order to the order they were created. You replied that this is a known bug (CR64611), and that Tech Support had identified a possible workaround. However, the bug is still there in IDL 8.2.2 (linux 64-bit), a year later.

--
Yngvar

Subject: Re: IDL_IDLBridge and the virtual machine
Posted by gombgg on Fri, 08 Nov 2013 19:36:44 GMT
View Forum Message <> Reply to Message

I just ran into this problem myself on a 24-core system running IDL 8.2.3 on 64 bit Linux. I can create 24 bridge objects and use them for processing, but if I destroy them in the same order they

were created then IDL hangs after destroying the 8th one. If I destroy them in reverse order, everything works fine.

On Monday, March 18, 2013 4:17:29 AM UTC-6, Yngvar Larsen wrote:
> On Friday, 15 March 2013 22:36:21 UTC+1, Mark Piper wrote:
>
>
>
>> but I agree, we need a generalized technique for multithreading IDL pro code. I should be careful
>
>> because this is somewhat distant, but we do have a parallel processing API under development
>
>> that we'd like to introduce in IDL 8.4. (Note that 8.3 is scheduled for this fall.)
>
>> Email me if you'd like to be a beta tester; same goes for anyone reading this.
>
>
>
> I'll also volunteer as a beta tester.
>
>
>
> Regarding bugs in IDL_IDLBRIDGE:
>
>
>
> A year ago, I reported an error in this newsgroup ("idl_idlbridge weirdness on unix systems", 2012-02-28): Using more than 15 bridges simultaneously on a linux 64-bit system causes IDL to hang if the bridges are not destroyed in the exact opposite order to the order they were created. You replied that this is a known bug (CR64611), and that Tech Support had identified a possible workaround. However, the bug is still there in IDL 8.2.2 (linux 64-bit), a year later.
>
>
>
> --
>
> Yngvar

Subject: Re: IDL_IDLBridge and the virtual machine
Posted by markb77 on Mon, 21 Apr 2014 13:49:28 GMT
View Forum Message <> Reply to Message

On Friday, March 15, 2013 10:36:21 PM UTC+1, Mark Piper wrote:

> but I agree, we need a generalized technique for multithreading IDL pro code. I should be careful because this is somewhat distant, but we do have a parallel processing API under development that we'd like to introduce in IDL 8.4. (Note that 8.3 is scheduled for this fall.) Email me if you'd like to be a beta tester; same goes for anyone reading this.
>
>
>
> We're also looking at better ways to distribute IDL applications. The VM has been OK, but it's time for something better.
>
>
>
> mp

hmm.. This thread is about a year old now.. but I'm wondering if there has been any progress with multithreading in IDL? I'm about to start developing some new code which will need to be multithreaded..

Is anything in development for IDL 8.4?

best,
Mark

Subject: Re: IDL_IDLBridge and the virtual machine
Posted by natha on Tue, 22 Apr 2014 12:30:53 GMT
View Forum Message <> Reply to Message

Hi Mark,

I've developed a library for multitasking based on the IDL_IDLBridge. I have a beta version that I can send it to you with an example code.
I spent a lot of time developing it and, at some point, I will make it public.
Let me know if you are interested...

Subject: Re: IDL_IDLBridge and the virtual machine
Posted by markb77 on Tue, 22 Apr 2014 17:48:40 GMT
View Forum Message <> Reply to Message

On Tuesday, April 22, 2014 2:30:53 PM UTC+2, nata wrote:
> Hi Mark,
>
>
>

> I've developed a library for multitasking based on the IDL_IDLBridge. I have a beta version that I can send it to you with an example code.
>
> I spent a lot of time developing it and, at some point, I will make it public.
>
> Let me know if you are interested...

Hi Nata,

Thanks for your offer - I'd like to check it out and possibly use it. I also have a public IDL code library at http://www.github.com/superchromix . Can you send your library by email or post it as a repo on github?

thanks, (and sorry if this is a duplicate message - not sure if my last one went through)

Mark

Subject: Re: IDL_IDLBridge and the virtual machine
Posted by kramers541 on Thu, 24 Apr 2014 09:36:57 GMT
View Forum Message <> Reply to Message

It would be great if we could get some feedback from Exelis on this.. Perhaps there are some simple changes to the bridge which would make it more useable? e.g. efficient passing/sharing of more complex datatypes (objects etc) between processes? Or is there another approach to multithreading in development?

Mark

Subject: Re: IDL_IDLBridge and the virtual machine
Posted by natha on Thu, 24 Apr 2014 15:05:52 GMT
View Forum Message <> Reply to Message

It would be great to pass/share objects between processes.

Subject: Re: IDL_IDLBridge and the virtual machine
Posted by Jie Zhou on Mon, 28 Jul 2014 11:36:03 GMT
View Forum Message <> Reply to Message

On Wednesday, March 13, 2013 12:58:07 PM UTC+1, 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(RefImg=0, Img=1)
>
> Res2 = ComputationFunction(RefImg=1, Img=2)
>
> Res3 = ComputationFunction(RefImg=2, Img=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

Just after copy the license folder under the IDLDIR directory to the distribution directory, the multiprocessing CUP works. It seems now the save file works in runtime mode. Good luck.
Jie
