
Subject: IDL to FORTRAN translator ?

Posted by [ebstein](#) on Thu, 30 Mar 1995 08:00:00 GMT

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

Does anyone know of tools that would be useful in porting some IDL code we have developed to FORTRAN ? TIA,

--

Steve Ebstein

ebstein@lexitek.com

Subject: Re: idl to fortran

Posted by [natha](#) on Wed, 10 Nov 2010 20:48:28 GMT

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

You can use parallel code in IDL using the IDL_IDLBridge object.

The IDL_IDLbridge object allows an IDL session to create and control other IDL sessions. If you want to share data between sessions you can use SHMMAP.

It is not very nice but I got some satisfactory results for large computations.

You can also take a look at <http://www.txcorp.com/products/FastDL/>

Cheers,
natha

Subject: Re: idl to fortran

Posted by [Michael Galloy](#) on Wed, 10 Nov 2010 21:08:46 GMT

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

On 11/10/10 1:48 PM, nata wrote:

> You can use parallel code in IDL using the IDL_IDLBridge object.
> The IDL_IDLbridge object allows an IDL session to create and control
> other IDL sessions. If you want to share data between sessions you can
> use SHMMAP.
> It is not very nice but I got some satisfactory results for large
> computations.
>
> You can also take a look at <http://www.txcorp.com/products/FastDL/>
>
> Cheers,
> nata

Disclaimer: I work for Tech-X Corporation and have worked on the products described below.

Yes, FastDL has both MPI and task farming options if you want to go that route. Another option to look into for increased performance is GPU computing (an option that can be combined with FastDL). We have a library of routines called GPULib, gpulib.txcorp.com, which can be called from IDL. Here's the API documentation to see what operations already exist:

http://www.txcorp.com/products/GPULib/idl_docs/index.html

We're always interested in hearing about what new routines would be useful, so if GPULib can't currently do the analysis you are hoping to do, contact me about what you need.

Mike

--

www.michaelgalloy.com
Research Mathematician
Tech-X Corporation

Subject: Re: idl to fortran

Posted by [penteado](#) on Wed, 10 Nov 2010 21:51:52 GMT

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

On Nov 10, 6:21 pm, a <oxfordenergyservi...@gmail.com> wrote:

> I've been writing some code in IDL. I've pushed it as fast as it can
> go but we may need it quicker. That means possibly many-cpu parallel
> code. That unfortunately means fortran or C with an OpenMP parallel
> compiling and running.

It may not hurt to point out that IDL can use multiple processors automatically (the routines that use the thread pool).

Subject: Re: idl to fortran

Posted by [penteado](#) on Wed, 10 Nov 2010 21:53:31 GMT

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

On Nov 10, 7:51 pm, Paulo Penteado <pp.pente...@gmail.com> wrote:

> On Nov 10, 6:21 pm, a <oxfordenergyservi...@gmail.com> wrote:

>

>> I've been writing some code in IDL. I've pushed it as fast as it can
>> go but we may need it quicker. That means possibly many-cpu parallel
>> code. That unfortunately means fortran or C with an OpenMP parallel
>> compiling and running.

>

> It may not hurt to point out that IDL can use multiple processors
> automatically (the routines that use the thread pool).

Also, if you do end up rewriting things, it will probably not be necessary to rewrite everything. You could do it just for the bottlenecks, and leave the rest in IDL, accessing those reimplemented parts through DLMS or call_external.

Subject: Re: idl to fortran

Posted by [oxfordenergyservices](#) on Fri, 12 Nov 2010 09:55:35 GMT

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

On Nov 10, 9:53 pm, Paulo Penteado <pp.pente...@gmail.com> wrote:

> On Nov 10, 7:51 pm, Paulo Penteado <pp.pente...@gmail.com> wrote:

>

>> On Nov 10, 6:21 pm, a <oxfordenergyservi...@googlemail.com> wrote:

>

>>> I've been writing some code in IDL. I've pushed it as fast as it can
>>> go but we may need it quicker. That means possibly many-cpu parallel
>>> code. That unfortunately means fortran or C with an OpenMP parallel
>>> compiling and running.

>

>> It may not hurt to point out that IDL can use multiple processors

>> automatically (the routines that use the thread pool).

>

> Also, if you do end up rewriting things, it will probably not be
> necessary to rewrite everything. You could do it just for the
> bottlenecks, and leave the rest in IDL, accessing those reimplemented
> parts through DLMS or call_external.

Thanks all the above for your comments, I'll look into them. In the mean time, I was considering your suggestion Paulo of the bottlenecks. I have not a huge amount of IDL code but the bottleneck is possibly 200 lines looped 5 million times. It would be pretty easy to code this in f90. I read somewhere else I could spawn the fortran code which does this bottleneck but how would I communicate between idl and fortran? I could write the data out from idl, read in fortran, write back out in fortran and read back in in idl. The data would be in RAM I assume so wouldn't be too slow but the data is rather large (a few 20 by 5 million arrays). Thanks for your suggestions

Russ

Subject: Re: idl to fortran

Posted by [oxfordenergyservices](#) on Fri, 12 Nov 2010 10:53:04 GMT

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

On Nov 12, 9:55 am, geoff <oxfordenergyservi...@googlemail.com> wrote:

> On Nov 10, 9:53 pm, Paulo Penteado <pp.pente...@gmail.com> wrote:

>

>

>

>

>

>

>

>

>

>> On Nov 10, 7:51 pm, Paulo Penteado <pp.pente...@gmail.com> wrote:

>

>>> On Nov 10, 6:21 pm, a <oxfordenergyservi...@googlemail.com> wrote:

>

>>>> I've been writing some code in IDL. I've pushed it as fast as it can

>>>> go but we may need it quicker. That means possibly many-cpu parallel

>>>> code. That unfortunately means fortran or C with an OpenMP parallel

>>>> compiling and running.

>

>>> It may not hurt to point out that IDL can use multiple processors

>>> automatically (the routines that use the thread pool).

>

>> Also, if you do end up rewriting things, it will probably not be

>> necessary to rewrite everything. You could do it just for the

>> bottlenecks, and leave the rest in IDL, accessing those reimplemented

>> parts through DLMS or call_external.

>

> Thanks all the above for your comments, I'll look into them. In the

> mean time, I was considering your suggestion Paulo of the

> bottlenecks. I have not a huge amount of IDL code but the

> bottleneck is possibly 200 lines looped 5 million times. It would be

> pretty easy to code this in fortran. I read somewhere else I could spawn

> the fortran code which does this bottleneck but how would I

> communicate between idl and fortran? I could write the data out from

> idl, read in fortran, write back out in fortran and read back in in

> idl. The data would be in RAM I assume so wouldn't be too slow but

> the data is rather large (a few 20 by 5 million arrays). Thanks for

> your suggestions

>

> Russ

sorry Paulo, bit keen on the keyboard there! I'll look into DLMS and
call_extrenal

Russ
