Subject: Re: IDL and supercomputers?

Posted by Ben Panter on Thu, 18 Dec 2003 14:09:22 GMT

View Forum Message <> Reply to Message

Jonathan Greenberg wrote:

- > Has anyone worked with ENVI/IDL on any supercomputers? Any suggestions on
- > optimizing code for use with them? I'm currently trying to get time on the
- > San Diego Supercomputer, and was wondering if its worth the time -- one
- > question I had, in this case, is there such thing as too MANY processors
- > (the SDSC has 1100!) -- do I want to limit the number of threads when
- > working on an array? Thanks! Any other stories related to this would be
- > great!

Jonathan,

The nearest I get is trivially parallel jobs running on 30 machines over a hetrogeneous cluster (in fact the more powerful observatory machines at night). If you have any luck with this project I'd love to hear about it - I think one of your prime worries might be getting enough licenses to run this stuff...

Ben

--

Ben Panter, Royal Observatory, Edinburgh My name (no spaces)@bigfoot which is a com.

Subject: Re: IDL and supercomputers?

Posted by Jamie on Thu, 18 Dec 2003 21:49:32 GMT

View Forum Message <> Reply to Message

There are DLMs written for using PVM and MPI using IDL. I sincerely hope you have read and understood the thread white paper, see bottom of http://www.rsinc.com/services/techtip.asp?ttid=3252. I doubt that using 1100 CPUs will prove particularly useful unless you are doing simple, independent calculations and comparing the results later (Monte Carlo runs of a model that is capable of running without inter-process communication would be a example). IDL has hard limits that you will undoubtedly run into if you were to try and solve a large memory SMP problem.

You must certainly will need to be careful with any version of IDL > 5.5 as you must synchronize the number of CPUs you allocate with the number of threads. IDL 5.6 has been banned from one cluster I know of because it is impossible for the admin to globally restrict the number of threads it uses. The number of threads using IDL-RT or IDL-VM can only be set by the

user. In short, when IDL starts, it detects the number of CPUs present and then sets ncpu == nthreads. This is very bad form if you didn't allocate ncpu processors.

I suspect that you will have a very hard time convincing people that running IDL code on a big-iron supercomputer will provide benefit. Is there any particular reason you aren't writing this code in HPF/HPC?

Jamie

>

On Thu, 18 Dec 2003, Ben Panter wrote:

> > Jonathan Greenberg wrote:

>> Has anyone worked with ENVI/IDL on any supercomputers? Any suggestions on

>> optimizing code for use with them? I'm currently trying to get time on the

>> San Diego Supercomputer, and was wondering if its worth the time -- one

>> question I had, in this case, is there such thing as too MANY processors

>> (the SDSC has 1100!) -- do I want to limit the number of threads when

>> working on an array? Thanks! Any other stories related to this would be

>> great!

> Jonathan,

> The nearest I get is trivially parallel jobs running on 30 machines > over a hetrogeneous cluster (in fact the more powerful observatory

> machines at night). If you have any luck with this project I'd love to

> hear about it - I think one of your prime worries might be getting

> enough licenses to run this stuff...

> Ben

Subject: Re: IDL and supercomputers?
Posted by Jonathan Greenberg on Thu, 18 Dec 2003 22:32:14 GMT
View Forum Message <> Reply to Message

Good advice -- I wasn't aware that IDL just took over the entire system (does IDK 6.0 have better protection from this?) I do a lot of array calls, that IDL claims will be pretty close to linearly related to the number of processors (although I'm not sure anyone's done multithreading with a machine like this before using IDL). I'm guessing my process will be a blip in the greater scheme of things -- it would take several days on a PC, but I'm hoping it just takes a few minutes on one of these big bastards.

I'm an ecologist, which is why I'm not developing more parallel optimized code -- i really don't have time to learn a brand new language at this

point -- i'm relying on IDL to have done a reasonable job parallizing their array calls (I'm matricizing my code as much as possible). I'll probably do some tests to see how well mp systems work with a large number of processors.

--j

"Jamie" <jamiedotwheeleratoxacuk@dummy.com> wrote in message news:Pine.LNX.4.44.0312182119310.16262-100000@moriarty.atm.o x.ac.uk... > There are DLMs written for using PVM and MPI using IDL. I sincerely hope > you have read and understood the thread white paper, see bottom of > http://www.rsinc.com/services/techtip.asp?ttid=3252. I doubt that using > 1100 CPUs will prove particularly useful unless you are doing simple, > independent calculations and comparing the results later (Monte Carlo runs > of a model that is capable of running without inter-process communication > would be a example). IDL has hard limits that you will undoubtedly run > into if you were to try and solve a large memory SMP problem. > > You must certainly will need to be careful with any version of IDL > 5.5 > as you must synchronize the number of CPUs you allocate with the number of > threads. IDL 5.6 has been banned from one cluster I know of because it is > impossible for the admin to globally restrict the number of threads it > uses. The number of threads using IDL-RT or IDL-VM can only be set by the > user. In short, when IDL starts, it detects the number of CPUs present > and then sets ncpu == nthreads. This is very bad form if you didn't > allocate ncpu processors. > > I suspect that you will have a very hard time convincing people that > running IDL code on a big-iron supercomputer will provide benefit. Is > there any particular reason you aren't writing this code in HPF/HPC? > **Jamie** > > On Thu, 18 Dec 2003, Ben Panter wrote: > >> >> Jonathan Greenberg wrote: >>> Has anyone worked with ENVI/IDL on any supercomputers? Any suggestions on >>> optimizing code for use with them? I'm currently trying to get time on the >>> San Diego Supercomputer, and was wondering if its worth the time --

processors

>>> guestion I had, in this case, is there such thing as too MANY

```
>>> (the SDSC has 1100!) -- do I want to limit the number of threads when
>>> working on an array? Thanks! Any other stories related to this would
be
>>> great!
>>
>> Jonathan,
>>
     The nearest I get is trivially parallel jobs running on 30 machines
>>
>> over a hetrogeneous cluster (in fact the more powerful observatory
>> machines at night). If you have any luck with this project I'd love to
>> hear about it - I think one of your prime worries might be getting
>> enough licenses to run this stuff...
>>
      Ben
>>
>
```