
Subject: How to improve the efficiency of IDL on cluster
Posted by [Chang Liao](#) on Mon, 03 Jun 2013 15:09:00 GMT
[View Forum Message](#) <> [Reply to Message](#)

Hi!
I am currently running IDL on a Linux cluster.
Due to the massive mount of remote sensing dataset, it takes days to finish just one year's calculation.
I can specific the computer node, memory(48G/96G), and cores(24 cores) for my QBS job. So I am thinking shall I use multi-threading or other ways to speedup the calculation.
Does anyone have any suggestion or experience in this related topics, I couldn't find too much on Exelis or Google. I do remember IDL has some demo in earlier version, but it's gone in IDL 8.0 +.

Please help, thanks in advance!

Subject: Re: How to improve the efficiency of IDL on cluster
Posted by [Kenneth Bowman](#) on Tue, 04 Jun 2013 21:26:42 GMT
[View Forum Message](#) <> [Reply to Message](#)

On 2013-06-03 15:09:00 +0000, Chang Liao said:

> Hi!
> I am currently running IDL on a Linux cluster.
> Due to the massive mount of remote sensing dataset, it takes days to
> finish just one year's calculation.
> I can specific the computer node, memory(48G/96G), and cores(24 cores)
> for my QBS job. So I am thinking shall I use multi-threading or other
> ways to speedup the calculation.
> Does anyone have any suggestion or experience in this related topics, I
> couldn't find too much on Exelis or Google. I do remember IDL has some
> demo in earlier version, but it's gone in IDL 8.0 +.
>
> Please help, thanks in advance!

You can get some benefit from multi-threading, and you can go to fully parallel, as Mike Galloy suggests, but first, can you trivially parallelize the job by splitting it into independent parts. Such as, process January as one job and February as a separate job?

If so, and memory permitting, you will probably do much better than multi-threading.

And, of course, always vectorize first, as Craig suggested.

Ken Bowman

Subject: Re: How to improve the efficiency of IDL on cluster
Posted by [Matt Francis](#) on Wed, 05 Jun 2013 03:23:13 GMT
[View Forum Message](#) <> [Reply to Message](#)

On Tuesday, 4 June 2013 01:09:00 UTC+10, Chang Liao wrote:

> Hi!
>
> I am currently running IDL on a Linux cluster.
>
> Due to the massive mount of remote sensing dataset, it takes days to finish just one year's calculation.
>
> I can specific the computer node, memory(48G/96G), and cores(24 cores) for my QBS job. So I am thinking shall I use multi-threading or other ways to speedup the calculation.
>
> Does anyone have any suggestion or experience in this related topics, I couldn't find too much on Exelis or Google. I do remember IDL has some demo in earlier version, but it's gone in IDL 8.0 +.
>
>
>
> Please help, thanks in advance!

Maybe it's heresy around these here parts, but in my view if you're putting IDL and cluster in the same sentence you're doing it wrong. IDL is a great Interactive Data Language but it's simply the wrong tool for the job for large parallel batch processing. You can put lipstick on a pig, but there are much better alternatives out there for doing this kind of stuff, both easier to use (for this use case) and faster.

Subject: Re: How to improve the efficiency of IDL on cluster
Posted by [victor.cazcarra](#) on Wed, 05 Jun 2013 15:18:08 GMT
[View Forum Message](#) <> [Reply to Message](#)

El lunes, 3 de junio de 2013 17:09:00 UTC+2, Chang Liao escribió:

> Hi!
>
> I am currently running IDL on a Linux cluster.
>
> Due to the massive mount of remote sensing dataset, it takes days to finish just one year's calculation.
>
> I can specific the computer node, memory(48G/96G), and cores(24 cores) for my QBS job. So I am thinking shall I use multi-threading or other ways to speedup the calculation.
>
> Does anyone have any suggestion or experience in this related topics, I couldn't find too much on Exelis or Google. I do remember IDL has some demo in earlier version, but it's gone in IDL 8.0 +.

>
>
>
> Please help, thanks in advance!

Hi!

Did you try to use the object IDL_IDLBridge? (
<http://slugidl.pbworks.com/w/page/29199259/Child%20Processes>)

It is like a 'fork' in C to create childs to do a big work in parallel. I have done some work to parallelize big loops and it works very well.

regards,

Víctor
