Subject: Re: Efficient sub array extraction
Posted by Phillip M. Bitzer on Tue, 11 Dec 2012 22:34:09 GMT

View Forum Message <> Reply to Message

In short, array subscripting can take up a lot of memory. Take a peek at:

http://www.idlcoyote.com/misc_tips/submemory.html

There's some helpful information by none other than JD Smith.

On 12/11/12 4:26 PM, markjamie@gmail.com wrote:
> I'm using large 2D arrays e.g 30000 x 30000 and need to remove a 2D sub array for later
processing e.g. Rows 30-700 and columns 100-10000.
>
> Is there a more efficient way to do this than using subscript ranges?
>
> For example?
>
> Subarray = largearray[100:10000, 30:700]
>
>
> The exact code I'm using is as follows:
>
> A = [100,10000]
> B = [30,700]
>
> Subarray = largearray[A[0]:A[1],B[0]:B[1]]
>
> Mark
>

Subject: Re: Efficient sub array extraction
Posted by David Fanning on Tue, 11 Dec 2012 23:40:16 GMT

View Forum Message <> Reply to Message

Phillip M. Bitzer writes:

> In short, array subscripting can take up a lot of memory. Take a peek at:
>
> http://www.idlcoyote.com/misc_tips/submemory.html
>
> There's some helpful information by none other than JD Smith.

Subscripting *can* take a lot of memory, but unless
there is a specific reason you are worried about
this, I wouldn't bother changing a thing. Reading

your large array into memory is what is killing you,
not the subscripting. :-)

Cheers,

David

--
David Fanning, Ph.D.
Fanning Software Consulting, Inc.
Coyote's Guide to IDL Programming: http://www.idlcoyote.com/
Sepore ma de ni thue. ("Perhaps thou speakest truth.")

Subject: Re: Efficient sub array extraction
Posted by Heinz Stege on Tue, 11 Dec 2012 23:41:57 GMT
View Forum Message <> Reply to Message

On Tue, 11 Dec 2012 14:26:40 -0800 (PST), markjamie@gmail.com wrote:

> I'm using large 2D arrays e.g 30000 x 30000 and need to remove a 2D sub array for later
processing e.g. Rows 30-700 and columns 100-10000.
>
> Is there a more efficient way to do this than using subscript ranges?
>
> For example?
>
> Subarray = largearray[100:10000, 30:700]
>
>
> The exact code I'm using is as follows:
>
> A = [100,10000]
> B = [30,700]
>
> Subarray = largearray[A[0]:A[1],B[0]:B[1]]

Hi Mark,

from my point of view the use of subscript ranges is fine. (However do
not use arrays for subscripting.) Using subscript ranges is fast and
don't need much memory. In the following example roughly about 100
extra bytes are used at creating the subarray:

IDL> array=bindgen(30000,30000)
IDL> a=[100,10000]
IDL> b=[30,70]

```
IDL> help,/mem
heap memory used:  900729595, max:  900729989, gets:      1073, frees:
232
IDL> subarr=array[a[0]:a[1],b[0]:b[1]]
IDL> help,/mem
heap memory used:  901135568, max:  901135685, gets:      1085, frees:
243
IDL> print,!version
{ x86 Win32 Windows Microsoft Windows 8.0.1 Oct  5 2010      32
64}
```

Cheers, Heinz

Subject: Re: Efficient sub array extraction
Posted by markjamie on Wed, 12 Dec 2012 00:00:35 GMT
View Forum Message <> Reply to Message

Thanks for your responses. Glad to hear my subscripting was ok. You mention not to use arrays as subscripts - is there a reason for this?

Subject: Re: Efficient sub array extraction
Posted by Heinz Stege on Wed, 12 Dec 2012 16:01:29 GMT
View Forum Message <> Reply to Message

On Tue, 11 Dec 2012 16:00:35 -0800 (PST), markjamie@gmail.com wrote:

> Thanks for your responses. Glad to hear my subscripting was ok. You mention not to use arrays as subscripts - is there a reason for this?

Yes.

There is a big difference between using "subscript ranges", for
example
   range=[10,110]
   subarr=array[range[0]:range[1]]
or "array subscripts", for example
   subscripts=indgen[101]+10s
   subarr=array[subscripts]
The results are the same. But using array subscripts is typically
slower and needs more memory during the calculation of the subarray.
The difference may get very huge when subscribing multidimensional
arrays.

The reason probably is similar to the observation, discussed in the
article by David Fanning, cited by Phillip M. Bitzer in a previous

post within this thread. Please read it.

Heinz