Subject: Re: too many elements
Posted by David Fanning on Thu, 11 Jun 2009 22:19:51 GMT
View Forum Message <> Reply to Message

Hu writes:

> As I have read the themes in this group that refers to 'Array has too
> many elements'. I do not got the idea about how to solve my following
> problem.
>
> supposing that I have to define a matrix (like, A) that have
> 10000*90000*10000 elements. how can obtain this without change the 32-
> b operate system? sentence like
>
> A=fltarr(10000,90000,10000)

Please tell us how many bytes you expect to be
able to fit in that array. (Don't forget to multiply
by 4 as there are four bytes for each floating point
value.) Then study that number for a couple of minutes
before asking the question again. :-)

Cheers,

David

--
David Fanning, Ph.D.
Coyote's Guide to IDL Programming (www.dfanning.com)
Sepore ma de ni thui. ("Perhaps thou speakest truth.")

Subject: Re: too many elements
Posted by Hu on Thu, 11 Jun 2009 22:39:05 GMT
View Forum Message <> Reply to Message

On Jun 11, 6:19 pm, David Fanning <n...@dfanning.com> wrote:
> Hu writes:
>> As I have read the themes in this group that refers to 'Array has too
>> many elements'. I do not got the idea about how to solve my following
>> problem.
>
>> supposing that I have to define a matrix (like, A) that have
>> 10000*90000*10000 elements. how can obtain this without change the 32-
>> b operate system? sentence like
>
>> A=fltarr(10000,90000,10000)
>

> Please tell us how many bytes you expect to be
> able to fit in that array. (Don't forget to multiply
> by 4 as there are four bytes for each floating point
> value.) Then study that number for a couple of minutes
> before asking the question again. :-)
>
> Cheers,
>
> David
>
> --
> David Fanning, Ph.D.
> Coyote's Guide to IDL Programming (www.dfanning.com)
> Sepore ma de ni thui. ("Perhaps thou speakest truth.")

Ok, maybe you though the question is not meaningful, sorry for that.

the problem is: I want to deal with a time series images,each image is 2300*1500 pixels, there are 300 images in total. so I have to define a array 2300*1500*300 to store all pixels. there is no problems till now, the problem is : each pixel at each time point have its own ancillary data, these ancillary information are essential for process the pixel values through time and space axises.

and, for my limited knowledge about IDL, I have to define another four array to store these ancillary data. the error appears when I define an extra array to store the values after all processing steps....

you could image my embarrassing situation.

all i want to know is whether there is an way to store these information without any memory allocation, does Pointer help?

Thanks again.

Subject: Re: too many elements
Posted by David Fanning on Thu, 11 Jun 2009 22:47:52 GMT
View Forum Message <> Reply to Message

Hu writes:

> the problem is: I want to deal with a time series images,each image is
> 2300*1500 pixels, there are 300 images in total. so I have to define a
> array 2300*1500*300 to store all pixels. there is no problems till
> now, the problem is : each pixel at each time point have its own
> ancillary data, these ancillary information are essential for process
> the pixel values through time and space axises.

>
> and, for my limited knowledge about IDL, I have to define another four
> array to store these ancillary data. the error appears when I define
> an extra array to store the values after all processing steps....
>
> you could image my embarrassing situation.
>
> all i want to know is whether there is an way to store these
> information without any memory allocation, does Pointer help?

OK, no, there is no way to store the data without
"memory allocation". And, no, a pointer doesn't help.

What you *might* be able to do is store these data
on your disk and access them via an "associated
variable" method. (This is how ENVI works with large
files.) With an associated variable you associate
some kind of data structure with a file. You can
pull pieces of the data out of the file without having
to read the entire file into a variable all at once.
This is, I think, your only hope with a 32-bit OS.

Cheers,

David

--
David Fanning, Ph.D.
Coyote's Guide to IDL Programming (www.dfanning.com)
Sepore ma de ni thui. ("Perhaps thou speakest truth.")

Subject: Re: too many elements
Posted by Chris[6] on Fri, 12 Jun 2009 09:13:55 GMT
View Forum Message <> Reply to Message

On Jun 11, 12:47 pm, David Fanning <n...@dfanning.com> wrote:
> Hu writes:
>> the problem is: I want to deal with a time series images,each image is
>> 2300*1500 pixels, there are 300 images in total. so I have to define a
>> array 2300*1500*300 to store all pixels. there is no problems till
>> now, the problem is : each pixel at each time point have its own
>> ancillary data, these ancillary information are essential for process
>> the pixel values through time and space axises.
>
>> and, for my limited knowledge about IDL, I have to define another four
>> array to store these ancillary data. the error appears when I define
>> an extra array to store the values after all processing steps....

>
>> you could image my embarrassing situation.
>
>> all i want to know is whether there is an way to store these
>> information without any memory allocation, does Pointer help?
>
> OK, no, there is no way to store the data without
> "memory allocation". And, no, a pointer doesn't help.
>
> What you *might* be able to do is store these data
> on your disk and access them via an "associated
> variable" method. (This is how ENVI works with large
> files.) With an associated variable you associate
> some kind of data structure with a file. You can
> pull pieces of the data out of the file without having
> to read the entire file into a variable all at once.
> This is, I think, your only hope with a 32-bit OS.
>
> Cheers,
>
> David
>
> --
> David Fanning, Ph.D.
> Coyote's Guide to IDL Programming (www.dfanning.com)
> Sepore ma de ni thui. ("Perhaps thou speakest truth.")

This is neither an IDL problem, nor a 32 bit cpu problem. A 10k by 90k
by 10k data cube (of floats, say) is 36 * 10^12 bytes = 32 Terabytes.
The problem is that finding 32 Tb of RAM is...difficult

chris

Subject: Re: too many elements
Posted by Hu on Fri, 12 Jun 2009 14:35:26 GMT
View Forum Message <> Reply to Message

On Jun 11, 6:47 pm, David Fanning <n...@dfanning.com> wrote:
> Hu writes:
>> the problem is: I want to deal with a time series images,each image is
>> 2300*1500 pixels, there are 300 images in total. so I have to define a
>> array 2300*1500*300 to store all pixels. there is no problems till
>> now, the problem is : each pixel at each time point have its own
>> ancillary data, these ancillary information are essential for process
>> the pixel values through time and space axises.
>
>> and, for my limited knowledge about IDL, I have to define another four

>> array to store these ancillary data. the error appears when I define
>> an extra array to store the values after all processing steps....
>
>> you could image my embarrassing situation.
>
>> all i want to know is whether there is an way to store these
>> information without any memory allocation, does Pointer help?
>
> OK, no, there is no way to store the data without
> "memory allocation". And, no, a pointer doesn't help.
>
> What you *might* be able to do is store these data
> on your disk and access them via an "associated
> variable" method. (This is how ENVI works with large
> files.) With an associated variable you associate
> some kind of data structure with a file. You can
> pull pieces of the data out of the file without having
> to read the entire file into a variable all at once.
> This is, I think, your only hope with a 32-bit OS.
>
> Cheers,
>
> David
>
> --
> David Fanning, Ph.D.
> Coyote's Guide to IDL Programming (www.dfanning.com)
> Sepore ma de ni thui. ("Perhaps thou speakest truth.")

Yes, your method is works.
I first store these dataset as an file in hard disk, and then
associate them with a variable when necessary.

Thanks, David.
