Subject: Re: Cannot allocate memory!
Posted by David Fanning on Mon, 16 Jan 2006 20:25:17 GMT

View Forum Message <> Reply to Message

Sheldon writes:

- > I am working with very large arrays of varying quantities, i.e.,
- > sometimes I have 89 arrays and other times I have 200 arrays. Now I
- > would like to average these arrays and reduce them to one array.
- > I use the MAKE_ARRAY() function but IDL screams that it cannot allocate
- > the memory.
- > Now these arrays are guite large (1215,1215) and I only need to hold
- > them within a function and then they are averaged using MEAN().
- > Is there away to do this so that IDL can allocate the memory?
- > IDI complains about this:
- > tmp = make_array([119,1215,1215], /float). I was thinking about
- > changing to /integer but I think is not such a good fix.
- > Any takers?

Uh, have you been reading the recent discussion of MEAN() in this newsgroup? There might be an idea or two in there. :-)

Cheers.

David

--

David Fanning, Ph.D. Fanning Software Consulting, Inc.

Coyote's Guide to IDL Programming: http://www.dfanning.com/

Subject: Re: Cannot allocate memory!
Posted by Craig Markwardt on Mon, 16 Jan 2006 20:26:00 GMT
View Forum Message <> Reply to Message

"Sheldon" <shejo284@gmail.com> writes:

> Hi everyone,

>

- > I am working with very large arrays of varying quantities, i.e.,
- > sometimes I have 89 arrays and other times I have 200 arrays. Now I
- > would like to average these arrays and reduce them to one array.
- > I use the MAKE_ARRAY() function but IDL screams that it cannot allocate
- > the memory.
- > Now these arrays are quite large (1215,1215) and I only need to hold
- > them within a function and then they are averaged using MEAN().
- > Is there away to do this so that IDL can allocate the memory?

- > IDI complains about this:
- > tmp = make_array([119,1215,1215], /float). I was thinking about
- > changing to /integer but I think is not such a good fix.

If all you are doing is straight averaging, then you can average each 1215x1215 slice individually using a FOR loop, and then average those averages. That allows you to keep only one image in memory at a time.

Good luck, Craig

Craig B. Markwardt, Ph.D. EMAIL: craigmnet@REMOVEcow.physics.wisc.edu Astrophysics, IDL, Finance, Derivatives | Remove "net" for better response

Subject: Re: Cannot allocate memory! Posted by Liberum on Mon, 16 Jan 2006 20:47:19 GMT

View Forum Message <> Reply to Message

Thanks David.

I am looking.....

Subject: Re: Cannot allocate memory! Posted by Liberum on Mon, 16 Jan 2006 20:57:59 GMT

View Forum Message <> Reply to Message

Hi Craig,

I cannot average each array because I want the average at each pixel for the 119 or so arrays.

What I am toying with now is this creating 2 1215*1215 arrays and use one to carry the new arrays and one to keep the running total:

t = make arrav(col.row)

sum = make_array(col,row)

for i=0, n-1 do begin

t = read hdf(f,str)

sum = temporary(sum) + temporary(t)

endfor

avg = sum/float(n)

Basicly I need to keep IDL from making many of these arrarys. I have not tested this yet.

Now you will really laugh at this but it is best that I can come up

with with what little experience I have using IDL. I am reading some older problems sent to the group but I am not done yet. Any comments?

Sheldon

Subject: Re: Cannot allocate memory!
Posted by David Fanning on Mon, 16 Jan 2006 21:09:59 GMT
View Forum Message <> Reply to Message

Sheldon writes:

- > I cannot average each array because I want the average at each pixel
- > for the 119 or so arrays.
- > What I am toying with now is this creating 2 1215*1215 arrays and use
- > one to carry the new arrays and one to keep the running total:
- > t = make_array(col,row)
- > sum = make_array(col,row)
- > for i=0, n-1 do begin
- > t = read hdf(f,str)
- > sum = temporary(sum) + temporary(t)
- > endfor
- > avg = sum/float(n)
- > Basicly I need to keep IDL from making many of these arrarys. I have
- > not tested this yet.
- > Now you will really laugh at this but it is best that I can come up
- > with with what little experience I have using IDL. I am reading some
- > older problems sent to the group but I am not done yet. Any comments?

I think this is more or less what Craig had in mind. At least it is tending in the right direction. :-)

Cheers,

David

--

David Fanning, Ph.D.

Fanning Software Consulting, Inc.

Coyote's Guide to IDL Programming: http://www.dfanning.com/

Subject: Re: Cannot allocate memory!

Posted by Liberum on Tue, 17 Jan 2006 08:56:06 GMT

View Forum Message <> Reply to Message

Thanks David,

IDL does have a very good manual.

I have a side question that I hope is short enough for you to answer. When looking at the other solutions for averaging large 3D arrays, I noticed something, namely that some synthax for depicting a 3D array took the form: array[50,50,3] where the third dimension is placed at the end and not the beginning: array[3,50,50].

Coming over from Python, this seems a bit strange. What is the difference between these two?

Doesn't IDL interpret former as a 50 dimensional array of 50 col x 3 rows and the latter as a 3 dimensional array of 50 col x 50 rows?

Cheers, Sheldon

Subject: Re: Cannot allocate memory!
Posted by Craig Markwardt on Tue, 17 Jan 2006 10:28:38 GMT
View Forum Message <> Reply to Message

"Sheldon" <shejo284@gmail.com> writes:

- > Thanks David,
- >
- > IDL does have a very good manual.
- > I have a side question that I hope is short enough for you to answer.
- > When looking at the other solutions for averaging large 3D arrays, I
- > noticed something, namely that some synthax for depicting a 3D array
- > took the form: array[50,50,3] where the third dimension is placed at
- > the end and not the beginning: array[3,50,50].
- > Coming over from Python, this seems a bit strange. What is the
- > difference between these two?
- > Doesn't IDL interpret former as a 50 dimensional array of 50 col x 3
- > rows and the latter as a 3 dimensional array of 50 col x 50 rows?

It doesn't really matter whether you call an array index a row or a column. There is more discussion here:

http://www.dfanning.com/misc_tips/colrow_major.html

I say it doesn't matter because most often, you must refer to the array indices explicitly, so you can define what you mean by a row and column yourself. The only time this is not true that I'm aware of is when doing matrix multiplication with the "#" and "##" operators. It's best just to verify which one is right for you.

For implementation efficiency, it may be important to know which array

| elements are adjacaent. | For an array, the elements ARRAY(*,J,K) are |
|-------------------------|---|
| contiguous in memory. | |
| | |

Craig

EMAIL: craigmnet@REMOVEcow.physics.wisc.edu Craig B. Markwardt, Ph.D. Astrophysics, IDL, Finance, Derivatives | Remove "net" for better response

Subject: Re: Cannot allocate memory! Posted by Liberum on Tue, 17 Jan 2006 17:43:18 GMT View Forum Message <> Reply to Message

Thanks for your help Craig,

I understand now.

The program worked with limiting the amount of arrays and without using MEAN().

Cheers, Sheldon