Subject: Re: reading/writing large files
Posted by Matthew on Fri, 01 Feb 2013 15:40:36 GMT
View Forum Message <> Reply to Message

You could index each chunk and keep track of how many rows each chunk contributes to the file. Whichever chunk you want to recall, sum of all the previous rows then use SKIP_LUN (or perhaps POINT_LUN) to skip to the desired line in the file.

Subject: Re: reading/writing large files
Posted by Carsten Lechte on Fri, 01 Feb 2013 15:46:01 GMT
View Forum Message <> Reply to Message

Hi, Russel

> Any ideas?

Have a look at the HDF5 scientific data format. You can write multiple-dimensional arrays of any data type, with the option of growing the array along one dimension, which should take care of the appending. Upon reading, you can specify a stride and maybe even a subarray. The libraries should be well optimised, so they could be faster than what you have so far.

However, I have not used these features of HDF5 myself, and they may not be accessible in IDL.

chl

Subject: Re: reading/writing large files
Posted by Matthew on Fri, 01 Feb 2013 15:52:00 GMT
View Forum Message <> Reply to Message

> Have a look at the HDF5 scientific data format.

Ahh. In this light, you could also use Common Data Format (cdf) files. I have never written a file myself, but they work similar to Carsten's description of HDF5. If you get a file written, I have a library that makes reading them (or a subsection of them) easy.

Subject: Re: reading/writing large files
Posted by Craig Markwardt on Fri, 01 Feb 2013 16:36:22 GMT
View Forum Message <> Reply to Message

On Friday, February 1, 2013 10:15:25 AM UTC-5, rr...@stsci.edu wrote:

> Okay gang I've been working on this for a few days and have given up.
>
>
>
> I've got this simulation that outputs an array of floating point numbers (roughly 5000 or so), which I want to put into a file. If the file exists, I want to append to it; if not, I want to create it. I want to do this of order a million times (at least append of order a million times). When the simulation finishes, I want to read these numbers and do some post-processing. I don't want to read the entire file at once because I'm afraid I'll run into memory problems (especially since I can envision doing the appending 10^7 or even 10^8 times). So, instead I'd like to read say all 10^6 (or 10^7 or 10^8) trials of the k-th element of the array and get a single floating-point array of 10^6 elements (or what have you). Basically, I'm envisioning a table with say 5000ish columns but the number of rows is variable, and I want to read the k-th column.
>

Suddenly all these questions about dealing with "large" files...

To me, a table with 10^6, 10^7 or 10^8 rows doesn't seem that large (although 5000 columns is pretty big).

If you are dealing in astronomy, consider using FITS files. There are lots of tools to deal with FITS files inside or outside of IDL.

Personally, I would have each simulation write one small table and then at the end, use a merging program such as 'ftmerge' (for FITS) to combine the small tables into one master table. This also allows you to have multiple simulations running on different processors, without fear of stomping on each other.

Craig

Subject: Re: reading/writing large files
Posted by Russell Ryan on Fri, 01 Feb 2013 16:49:05 GMT
View Forum Message <> Reply to Message

On Friday, February 1, 2013 11:36:22 AM UTC-5, Craig Markwardt wrote:
> On Friday, February 1, 2013 10:15:25 AM UTC-5, rr...@stsci.edu wrote:
>
>> Okay gang I've been working on this for a few days and have given up.
>
>>
>
>>
>
>>
>
>> I've got this simulation that outputs an array of floating point numbers (roughly 5000 or so), which I want to put into a file. If the file exists, I want to append to it; if not, I want to create it. I

want to do this of order a million times (at least append of order a million times). When the simulation finishes, I want to read these numbers and do some post-processing. I don't want to read the entire file at once because I'm afraid I'll run into memory problems (especially since I can envision doing the appending 10^7 or even 10^8 times). So, instead I'd like to read say all 10^6 (or 10^7 or 10^8) trials of the k-th element of the array and get a single floating-point array of 10^6 elements (or what have you). Basically, I'm envisioning a table with say 5000ish columns but the number of rows is variable, and I want to read the k-th column.
>
>>
>
>
>
> Suddenly all these questions about dealing with "large" files...
>
>
>
> To me, a table with 10^6, 10^7 or 10^8 rows doesn't seem that large (although 5000 columns is pretty big).
>
>
>
> If you are dealing in astronomy, consider using FITS files. There are lots of tools to deal with FITS files inside or outside of IDL.
>
>
>
> Personally, I would have each simulation write one small table and then at the end, use a merging program such as 'ftmerge' (for FITS) to combine the small tables into one master table. This also allows you to have multiple simulations running on different processors, without fear of stomping on each other.
>
>
>
> Craig

Hi Gang,

Thanks for the ideas. Yeah, I'm familiar with the HDF files and think I'm gonna look at the CDF files. Craig, Yeah I am an astronomer and have been using fits files. It would be awesome if I could use a binary table, but it seems that I can only have 999 columns (for fx*pro). I was looking at the ft*pro library from Landsman, but it's not clear to me that this will work for me either. DO you have any other advice on the fits I/O libraries?

R

Subject: Re: reading/writing large files
Posted by Craig Markwardt on Fri, 01 Feb 2013 18:44:46 GMT
View Forum Message <> Reply to Message

On Friday, February 1, 2013 11:49:05 AM UTC-5, rr...@stsci.edu wrote:
> On Friday, February 1, 2013 11:36:22 AM UTC-5, Craig Markwardt wrote:
> Thanks for the ideas. Yeah, I'm familiar with the HDF files and think I'm gonna look at the CDF files. Craig, Yeah I am an astronomer and have been using fits files. It would be awesome if I could use a binary table, but it seems that I can only have 999 columns (for fx*pro). I was looking at the ft*pro library from Landsman, but it's not clear to me that this will work for me either. DO you have any other advice on the fits I/O libraries?

Do you really have 5000 distinct name-worthy values? I bet you have vectors of data. FITS binary tables can easily accomodate vectors (or even arrays) in a single table cell. We routinely put whole images into a FITS binary table cell.

For real programming I usually use the fits_bintable library (FXB*.PRO), which allows one to be very explicit about the table structure. For quick-n-dirty you can use MRDFITS and MWRFITS. (for MWRFITS you need to be very careful about making sure your data has the same data structure and data type, so that you can merge it later without hassle.)

Craig

Subject: Re: reading/writing large files
Posted by Russell[1] on Fri, 01 Feb 2013 20:23:41 GMT
View Forum Message <> Reply to Message

On Friday, February 1, 2013 1:44:46 PM UTC-5, Craig Markwardt wrote:
> On Friday, February 1, 2013 11:49:05 AM UTC-5, rr...@stsci.edu wrote:
>
>> On Friday, February 1, 2013 11:36:22 AM UTC-5, Craig Markwardt wrote:
>
>> Thanks for the ideas. Yeah, I'm familiar with the HDF files and think I'm gonna look at the CDF files. Craig, Yeah I am an astronomer and have been using fits files. It would be awesome if I could use a binary table, but it seems that I can only have 999 columns (for fx*pro). I was looking at the ft*pro library from Landsman, but it's not clear to me that this will work for me either. DO you have any other advice on the fits I/O libraries?
>
>
>
> Do you really have 5000 distinct name-worthy values? I bet you have vectors of data. FITS binary tables can easily accomodate vectors (or even arrays) in a single table cell. We routinely put whole images into a FITS binary table cell.
>
>
>
> For real programming I usually use the fits_bintable library (FXB*.PRO), which allows one to be

very explicit about the table structure. For quick-n-dirty you can use MRDFITS and MWRFITS. (for MWRFITS you need to be very careful about making sure your data has the same data structure and data type, so that you can merge it later without hassle.)
>
>
>
> Craig

Hiya Craig,
Unfortunately I do have ~5000 unique variables. More concretely, I have an MCMC simulation which has ~5000 dimensions --- and I want ~10^6-8 random deviates from the posterior. But, only ~20 of these dimensions are truly interesting (yet don't want to throw away all that extra data).

It sounds like you're suggesting I write a FITS binary table, but each entry in the table is an array. This is possible, and I'm considering; especially since there is a natural way of doing this in my code. But it will have a minor draw back in recovering the file. In this scheme, each array (per cell) would be like a 1000 element array and if I have say 10^6-8 draws from it, I'm looking at reading 10^9-11 floating point numbers. This is becoming a problem, though I'm not sure if I'll ever *NEED* to do this. I just wanted to keep my options open.

Thank you all so much!
R