Subject: Re: "Unable to allocate memory" when using H5_parse iteratively
Posted by james.n.sweet on Tue, 13 May 2014 15:20:24 GMT
View Forum Message <> Reply to Message

On Monday, April 21, 2014 5:40:50 PM UTC-6, robinde...@gmail.com wrote:
> Hi David and Thank You for your reply.
>
> I added this line in my code but unfortunately the problem remains. Indeed the memory use increases with time, and the error occurs when it reaches about 350 Mb (on a total of 2000)
>
> I tried processing the data by smaller bunches, but it happens that after processing without problem a smaller bunch of 05 files, the program bugs when i run it for processing the next one, for the same error.
>
> Hi,
I've had a possibly similar problem. When using h5 parse, the memory usage would increase as more files were processed. I informed envi but the didn't follow up. My solution was to use the h5 open/read/close functions explicitly rather than using h5 parse. It's a pain but it is faster than parse (in my case anyway) and the memory problem is gone.

Hope this helps!

Jim
>
> Le lundi 21 avril 2014 17:09:24 UTC-5, David Fanning a écrit :
>
>> robindesvolcans@gmail.com writes:
>
>>
>
>>
>
>>
>
>>>> I'm Robin, a new member on this group, using IDL to process satellite images for volcano monitoring.
>
>>
>
>>>> I've written a code to process large number of data from the ozone monitoring instrument. These are HDF5 files weighing 28 Mb each.
>
>>
>
>>>> The code interactively calls the H5_parse function to open the files
>
>>
>

>> and extract the necessary variables, which undergo some basic processing
>
>>
>
>> (gridding, mapping and calculating the total SO2 mass) The variables are
>
>>
>
>> overwritten at each loop iteration. However after 30-40 iterations the
>
>>
>
>> H5_parse function returns me the unable to allocate memory error
>
>>
>
>> message. It looks as if the H5_parse function was not emptying its
>
>>
>
>> temporary memory use, but when i look at the task manager's display of
>
>>
>
>> the memory use, it stays constant and low (20%).
>
>>
>
>>> Does someone have an idea of how to solve the problem?
>
>>
>
>>
>
>>
>
>>
>
>> You may be overwriting the variables, but it is extremely unlikely you
>
>>
>
>> are cleaning up all the pointers and other heap variables returned by
>
>>
>
>> the H5_Parse function. I would try doing this at the end of each file
>
>>
>

>> processed:
>
>>
>
>>
>
>>
>
>>   for j=0,whatever do begin
>
>>
>
>>   h5data = HD_Parse(file[j])
>
>>
>
>>   ...
>
>>
>
>>   Heap_Free, h5data
>
>>
>
>>   endfor
>
>>
>
>>
>
>>
>
>> 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.")
