Subject: Re: Speed penalty using START and COUNT with HDF_SD_GETDATA
Posted by R.Bauer on Wed, 05 Sep 2001 15:44:46 GMT
View Forum Message <> Reply to Message

Bob Fugate wrote:
>
> Reimar,
> I don't have any control over how the data are written or stored. How can I
> do what you suggest? I am doing something like the following now (assumes
> there are 8000 frames in the SDS):
>
> hdf_sd_getdata,arrayid,data,start=[46,43,0],count=[32,32,800 0]
>
> where the first two numbers are the indices where I want to start extracting
> the data from the 128x128 array and 32 is the size of the extracted array.
> The above is much slower than
>
> hdf_sd_getdata,arrayid,data
>
> or even
>
> hdf_sd_getdata,arrayid,data,start=[0,0,0],count=[128,128,800 0]
>
> Can you make a specific suggestion as to how I can use 'limited dimension'
> in this context?
>
> Thanks

Ok,
I try to explain.

The first prcedure creates two datasets with two different dimensions.
The dimension of var1 is unlimited this is done by the [0] argument.
And var2 has the dimension of 10.

```
PRO create_data_dims

  sd_id = HDF_SD_START('test.hdf', /CREATE)
; Create an dataset that includes an unlimited dimension:
  sds_id = HDF_SD_CREATE(sd_id, 'var1', [0], /SHORT)
  sds_id = HDF_SD_CREATE(sd_id, 'var2', [10], /SHORT)
  HDF_SD_ENDACCESS, sds_id
  HDF_SD_END, SD_ID

END
```

The second procedure reads the dimensions of the data and
you get something like this back.

```
VAR1          0
VAR2         10
```

```
PRO read_data_dims
  sd_id = HDF_SD_START('test.hdf')

  index = HDF_SD_NAMETOINDEX(sd_id, 'var1')
  sds_id=HDF_SD_SELECT(sd_id,index)
  HDF_SD_GETINFO, SDS_ID,dims=dim
  PRINT,'VAR1',dim
  HDF_SD_ENDACCESS, sds_id

  index = HDF_SD_NAMETOINDEX(sd_id, 'var2')
  sds_id=HDF_SD_SELECT(sd_id,index)
  HDF_SD_GETINFO, SDS_ID,dims=dim
  PRINT,'VAR2',dim
  HDF_SD_ENDACCESS, sds_id

  HDF_SD_END, SD_ID
END
```

If you exchange test.hdf and the varnames to one of your files
you can examine if the last dimension is 0.
This means unlimited dimension.

If you found unlimited dimensions then one of the possibilities is
to read in the whole set and store it with limited dimensions.

Only by writing the decision between limited and unlimited could be
done.

If you don't have routines yourself for this I can share some of
our routines.

regards
Reimar

>
>> From: Reimar Bauer <r.bauer@fz-juelich.de>
>> Organization: Forschungszentrum Juelich GmbH

>> Newsgroups: comp.lang.idl-pvwave
>> Date: Wed, 05 Sep 2001 09:35:55 +0200
>> Subject: Re: Speed penalty using START and COUNT with HDF_SD_GETDATA
>>
>> Mark Hadfield wrote:
>>>
>>> "Bob Fugate" <rqfugate@mindspring.com> wrote in message
>>> news:B7BAF61A.2E03%rqfugate@mindspring.com...
>>>> I have a large number of 128x128 pixel arrays stored as SDS's in
>>>> HDF files. Since I am only interested in a 32x32 subset of each
>>>> array, I tried using the START and COUNT keywords to read
>>>> only that part of the array I need ---
>>>> thinking this would be faster and less taxing on memory.
>>>> However, I learned today that it is much faster to read
>>>> in the entire array.
>>>>
>>>> ...
>>>>
>>>> This is a so-so Windows NT machine; IDL 5.4. The data is on a
>>>> server. I have
>>>> a good connection to the server.
>>>>
>>>> Anyone had any similar experiences
>>>
>>> I have noticed something similar with IDL's netCDF interface: using the
>>> STRIDE keyword seems to be very inefficient. I got the impression that IDL
>>> is actually reading in the whole array then extracting a subset.
>>>
>>>> ...suggestions on how to speed up reading
>>>> only the part of the array I need?
>>>
>>> Have you tried copying the file to a local disk? The local disk's caching
>>> may suit the way IDL reads the data better.
>>>
>>
>>
>> I believe both of you are using unlimited dimension.
>> In the past we did a lot of tests with data which is stored in
>> limited and umlimited dimensions.
>>
>> During reading data in limited dimension is much much more faster,
>> I am not sure if I right remember but I believe about more than ten
>> times.
>>
>> We often use netCDF reading only one parameter or some parameters by
>> count
>> and offset and this is very fast. (Much more faster as reading the whole
>> file)

>>
>> I will explain what happens if you write with an unlimited dimension.
>>
>> e.g.
>>
>> DATA1 is 1 , 2, 3, 4, 5
>> DATA2 is 10,20,30,40,50
>>
>>
>> unlimited writes in this way
>>
>> 1,10,2,20,3,30,4,40,5,50
>>
>> Then exactly this happens you both described.
>> The whole file or much of the file must be read in to read only some
>> data.
>>
>>
>> if you write with limited dimensions the data is stored like
>>
>> 1,2,3,4,5,10,20,30,40,50
>>
>> In this case only parts of the data must be read in.
>>
>> We decided to write data with limited dimensions because normally they
>> are
>> once written but many times you like to read them as fast as possible.
>>
>>
>> hope this helps
>>
>>
>> regards
>> Reimar
>>
>>
>>
>> --
>> Reimar Bauer
>>
>> Institut fuer Stratosphaerische Chemie (ICG-1)
>> Forschungszentrum Juelich
>> email: R.Bauer@fz-juelich.de
>> http://www.fz-juelich.de/icg/icg1/
>> ========== ======
>> a IDL library at ForschungsZentrum Juelich
>> http://www.fz-juelich.de/icg/icg1/idl_icglib/idl_lib_intro.h tml
>>

>> http://www.fz-juelich.de/zb/text/publikation/juel3786.html
>> ========== ======
>>
>> read something about linux / windows
>> http://www.suse.de/de/news/hotnews/MS.html

--
Reimar Bauer

Institut fuer Stratosphaerische Chemie (ICG-1)
Forschungszentrum Juelich
email: R.Bauer@fz-juelich.de
http://www.fz-juelich.de/icg/icg1/
========== ======
a IDL library at ForschungsZentrum Juelich
http://www.fz-juelich.de/icg/icg1/idl_icglib/idl_lib_intro.h tml

http://www.fz-juelich.de/zb/text/publikation/juel3786.html
========== ======

read something about linux / windows
http://www.suse.de/de/news/hotnews/MS.html