Subject: Re: large array
Posted by btt on Tue, 12 Oct 2004 17:35:39 GMT
View Forum Message <> Reply to Message

Wolf Schweitzer wrote:
> Update.
>
> I believe that IDL can STILL only work with arrays of 2 GB size due to
> internal representation even on systems where more space is available.
> Positioning along any huge file using point_lun and the Long64-function
> and then reading a <2GB-part is no problem.
>
> http://www.rsinc.com/services/techtip.asp?ttid=2597
>
> But try:
>
> a = intarr (2000,2000,400)
> a [1999,1999,299] = 12
> b = where (a eq 12)
> help, b ; it is a LONG ... and that is a problem in my eyes;
> ; it shows they represent arrays with scalar counters and
> ; as long as these are LONG there is no way an array can be
> ; larger
>
>
> a = intarr (2000,2000,800)
> a [1999,1999,799] = 12
> b = where (a eq 12) ; (this really should crash your IDL)
>
> help, b ; i would be shocked if you get that far, but, please
> ; tell me what data type you obtain here
> print, b ; let me see this if you can, too
>
> Does anyone know when RSI will fix this serious limit? Until then, it'll
> probably be the chunk trick they wrote about.
>
>
>
>
> Wolf Schweitzer wrote:
>
>> I need to read a file that is ca. 7 GB large (the file size is
>> defineda s 2048 x 2048 x 900 are the dimensions, x 2 (integers) + 512
>> bytes header).
>>
>> The format is known (.ISQ) and I have a routine that deals with the
>> header information efficiently; I have some similarly formatted files
>> of the same scan process around 800-900 MB that I can read without

>> problems.
>>
>> Generally, I had no problems reading files up to 1.4 GB in size
>> directly into one variable under IDL.
>>
>> The data is read into an intarr (2048,2048,900) like this:
>>
>> imagearray = assoc (.., intarr(...), headersize)
>> imagearray = temporary (imagearray [0])
>>
>> Now, the reading takes a little while but it seems to work alright.
>> The machine - most of the time - does not crash (12 GB RAM, 64-bit AIX
>> version of IDL 6.1, IBM-Workstation).
>>
>> However, the visualisation of slice subscripts to this array later
>> does not display any interesting information; instead, the images of
>> these large data files look different each time and they do not
>> reflect the content of the data.
>>
>>
>> Question:
>>
>> What do I need to know in order to set up an array for very large
>> data? Is there a basic difference between arrays < 2 GB and arrays >
>> GB of size? Are the subscript variables all multiplied before the
>> array is really looked at, so do all of the subscript variables need
>> to be Long64? Such as intarr (long64(x),long64(y),long64(z))?
>>

Hello,

I'm the last guy who will ever understand all of this - but, here's a couple of things I have picked-up on the way.

(1) The big-file thing is releated to the memory_bits and file_offset_bits IDL operates under. It seems those are fixed by your platform and/or operating system. You can determine what you have to play with for any given verion of IDL with ...

```
IDL> help, !Version,/str
** Structure !VERSION, 8 tags, length=76, data length=76:
   ARCH            STRING    'ppc'
   OS              STRING    'darwin'
   OS_FAMILY       STRING    'unix'
   OS_NAME         STRING    'Mac OS X'
   RELEASE         STRING    '6.1'
   BUILD_DATE      STRING    'Jul 14 2004'
   MEMORY_BITS     INT             32
```

```
FILE_OFFSET_BITS
        INT         32
```

(2) If you are fortunate enough to have 64-bit access then you can use the /L64 keyword to WHERE, HISTOGRAM, etc. so that 64-bit integers are returned.

(3) I don't follow what you mean here...

>> However, the visualisation of slice subscripts to this array later
>> does not display any interesting information; instead, the images of
>> these large data files look different each time and they do not
>> reflect the content of the data.
>>

Could you explain that again?

Cheers,
Ben
