Subject: Re: File sizes and the SAVE command
Posted by Paolo Grigis on Wed, 22 Mar 2006 11:08:10 GMT
View Forum Message <> Reply to Message

Carsten Pathe wrote:

```
>  Hi there,
>
>  I am wondering about the IDL save command and the disk space of the
>  created save files.
>  Just an simple example:
>
>  a=intarr(100000)
>  tmp = size(a)
>  print, string(format='(f10.3)',(tmp(1)*tmp(2))/(2.^10.))+' kbyte'
>  ;195.313 kbyte
>  save, a, filename='d:\temp\test\b.dat'
>
>  b=fltarr(100000)
>  tmp = size(b)
>  print, string(format='(f10.3)',(tmp(1)*tmp(2))/(2.^10.))+' kbyte'
>  ;390.625 kbyte
>  save, a, filename='d:\temp\test\a.dat'
>
>  c=dblarr(100000)
>  tmp = size(c)
>  print, string(format='(f10.3)',(tmp(1)*tmp(2))/(2.^10.))+' kbyte'
>  ;488.281 kbyte
>  save, c, filename='d:\temp\test\c.dat'
>
>  When you look at the created files and their sizes, you will see the
>  following:
>  a.dat 393 kb
>  b.dat 393 kb
>  c.dat 784 kb
>
>  If you compare the file sizes to the sizes, the arrays were allocating
>  in the memory before they were save to disk, you see differences which
>  will cost you a lot of disk space when saving arrays of several hundred
>  megabytes.
>  Does anybody know, why the save command is producing files larger than
>  they should be?
```

Because (size(a))[2] is the type code, which has nothing to do with the byte size of each type, which is:

TYPE	#BYTES
Byte	1

Integer	2
Unsigned Integer	2
Long	4
Unsigned Long	4
64-bit Long	8
64-bit Unsigned Long	8
Floating-point	4
Double-precision	8

Ciao,
Paolo

>
> PS: I know, that I can also use:
> openw, 10, 'd:\temp\test\a.dat'
> writeu, 10, a
> close, 10
> But when I want to restore the data, I have to know the structure of the
> data to restore - which is not always the case.
>
> Thanks a lot help

Subject: Re: File sizes and the SAVE command
Posted by Maarten[1] on Wed, 22 Mar 2006 11:10:02 GMT
View Forum Message <> Reply to Message

I don't think you calculated quite what you thought you did.

tmp = size(a) & tmp[1]*tmp[2]
for a single dimensional array a will be the length of the array times
the _type_ of the array, which has nothing to do with the actual
byte-size of the elements.

The save-sizes seem consistent though: 100000 * 4 bytes for float and
int (long), double that for double precision floating point data.

Maarten

Subject: Re: File sizes and the SAVE command
Posted by Klaus Scipal on Wed, 22 Mar 2006 11:58:29 GMT
View Forum Message <> Reply to Message

The problem is not related to the calculation of the filesize, but the
actual amount of memory required

Take two arrays
a=intarr(100000)
b=fltarr(100000)

and save them using the save comannd
the file for array a takes 402096 bytes diskspace
the file for array b takes 402096 bytes diskspace

save them using openw & writeu
the file for array a takes 200000 bytes diskspace
the file for array b takes 400000 bytes diskspace

So the save command seems to waste a lot of diskspace, but why? Does the IDL
save command convert an integer automatically into a longinteger?

Klaus

"Maarten" <maarten.sneep@knmi.nl> wrote in message
news:1143025802.678782.180020@i39g2000cwa.googlegroups.com.. .
> I don't think you calculated quite what you thought you did.
>
> tmp = size(a) & tmp[1]*tmp[2]
> for a single dimensional array a will be the length of the array times
> the _type_ of the array, which has nothing to do with the actual
> byte-size of the elements.
>
> The save-sizes seem consistent though: 100000 * 4 bytes for float and
> int (long), double that for double precision floating point data.
>
> Maarten
>

Subject: Re: File sizes and the SAVE command
Posted by R.Bauer on Wed, 22 Mar 2006 13:26:13 GMT

View Forum Message <> Reply to Message

Klaus Scipal wrote:
> The problem is not related to the calculation of the filesize, but the
> actual amount of memory required
>
> Take two arrays
> a=intarr(100000)
> b=fltarr(100000)

>
> and save them using the save comannd
> the file for array a takes 402096 bytes diskspace
> the file for array b takes 402096 bytes diskspace
>
> save them using openw & writeu
> the file for array a takes 200000 bytes diskspace
> the file for array b takes 400000 bytes diskspace
>
> So the save command seems to waste a lot of diskspace, but why? Does the IDL
> save command convert an integer automatically into a longinteger?
>
> Klaus
>

why not using /compress ?

An idl sav file is not only a binary copy of your value. It does use a
XDR exchange data format to create files which are platform independent.
Each value has always it's XDR description included.

In general one of the scientific data formats e.g. netCDF are much
better to store your data in a common structure which is exchangable to
a lot of platforms too

cheers

Reimar

--
Reimar Bauer

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

Subject: Re: File sizes and the SAVE command
Posted by Klaus Scipal on Wed, 22 Mar 2006 15:17:28 GMT
View Forum Message <> Reply to Message

Hi Reimar

re compress:
Compress helps but on the cost of time when reading/writing the data.

re xdr:
but why is the filesize then so different
In our case for a float array it will take 2096 bytes to store the overhead
and for the integer array 202096 bytes. This difference can not only be the
result of the XDR description.

But maybe the XDR format use 4 bytes instead of 2 bytes for integer
represenation?

Klaus

"Reimar Bauer" <R.Bauer@fz-juelich.de> wrote in message
news:dvrj9m$be87$1@zam602.zam.kfa-juelich.de...
> Klaus Scipal wrote:
>> The problem is not related to the calculation of the filesize, but the
>> actual amount of memory required
>>
>> Take two arrays
>> a=intarr(100000)
>> b=fltarr(100000)
>>
>> and save them using the save comannd
>> the file for array a takes 402096 bytes diskspace
>> the file for array b takes 402096 bytes diskspace
>>
>> save them using openw & writeu
>> the file for array a takes 200000 bytes diskspace
>> the file for array b takes 400000 bytes diskspace
>>
>> So the save command seems to waste a lot of diskspace, but why? Does the
>> IDL
>> save command convert an integer automatically into a longinteger?
>>
>> Klaus
>>
>
> why not using /compress ?
>
> An idl sav file is not only a binary copy of your value. It does use a
> XDR exchange data format to create files which are platform independent.
> Each value has always it's XDR description included.

>
> In general one of the scientific data formats e.g. netCDF are much
> better to store your data in a common structure which is exchangable to
> a lot of platforms too
>
>
> cheers
>
> Reimar
>
>
>
> --
> Reimar Bauer
>
> Institut fuer Stratosphaerische Chemie (ICG-I)
> Forschungszentrum Juelich
> email: R.Bauer@fz-juelich.de
> -- -------
> a IDL library at ForschungsZentrum Juelich
> http://www.fz-juelich.de/icg/icg-i/idl_icglib/idl_lib_intro. html
> == =======

Subject: Re: File sizes and the SAVE command

Posted by Maarten[1] on Wed, 22 Mar 2006 15:27:41 GMT

View Forum Message <> Reply to Message

Klaus Scipal wrote:

> But maybe the XDR format use 4 bytes instead of 2 bytes for integer
> represenation?

yes, see http://www.faqs.org/rfcs/rfc1014.html

> From that page:

"The representation of all items requires a multiple of four bytes (or
32 bits) of data. [...] An XDR signed integer is a 32-bit datum that
encodes an integer in the range [-2147483648,2147483647]. The integer
is represented in two's complement notation. "

Maarten