Subject: maximum LUN
Posted by greg.addr on Mon, 24 Nov 2008 14:38:25 GMT
View Forum Message <> Reply to Message

I'm running out of IDL-allocated LUNs in my current program, the maximum being 28. I can write a new get_lun to make use of the the hundred lower values, but even 128 seems rather measly for 2008. Is there any reason why this number couldn't be larger?

cheers,
Greg

Subject: Re: maximum LUN
Posted by greg.addr on Mon, 24 Nov 2008 18:33:20 GMT
View Forum Message <> Reply to Message

> Have you tried it? I'd be surprised if that's true - but then, I've
> been surprised before. ;^)

I did do it that way at first: I later thought that by keeping the files open, I might benefit from the disk caching when reading from another close or overlapping area. I'm not sure if closing and reopening a file makes any difference to this, but my impression was that it became faster. I'd be surprised if there's not an overhead in reestablishing tens of file links for each drag of the mouse - but it might be smaller than I think. In any case, I still find it preferable in theory to keep the files open - otherwise one could argue that we never need more than one LUN...

For the moment I made a get_lun to handle 100 LUNs. If those run out, I'll start experimenting with an opening and closing marathon.

cheers,
Greg

Subject: Re: maximum LUN
Posted by Jeremy Bailin on Tue, 25 Nov 2008 03:46:36 GMT
View Forum Message <> Reply to Message

On Nov 24, 1:30 pm, Reimar Bauer <R.Ba...@fz-juelich.de> wrote:
> greg.a...@googlemail.com schrieb:
>
>
>
>>> What reason could there possibly be for having hundreds

>>> of files open simultaneously? The only thing that comes
>>> immediately to mind for me is a poor programming concept. :-)
>
>> I kind of expected that reply... :) I think I do have a valid
>> application, though. I have an archive of a few thousand satellite
>> images, each of which has 6 bands, stored in separate files. I've made
>> an object which can handle the six bands and return a subset image for
>> a selected region processed as I want it. It also returns the image's
>> coverage of the region, so that I can fill in any gaps with data from
>> other images. This I do by opening further image objects. So far, I
>> can use four of these to make an on-the-fly mosaic before I run out of
>> LUNs - and it works well up to that point. The program allows zooming
>> and panning, so I need repeat access to similar regions of the same
>> files. I think it would be too slow to close and reopen them every
>> time, so the objects hold the files open. The archive is several
>> terabytes, changing, and not in my control - so preprocessing is out.
>> With 128 LUNs instead of 28 I could mosaic about 20 image subsets,
>> which will be enough for the moment. Still, if the number 128 was just
>> an arbitrary choice long ago, I'd like to ask early for a bigger one!
>
>> cheers,
>> Greg
>
> close and open costs nothing. The code between both lines can be
> efficient or not.
>
> cheers
> Reimar

Not exactly "nothing", but about 0.2 ms for me:

```
IDL> s1=systime(/sec) & openr,1,'foo' & close,1 & s2=systime(/sec) &
print, s2-s1
   0.00021815300
IDL> s3=systime(/sec) & s4=systime(/sec) & print, s4-s3
   6.9141388e-06
```

-Jeremy.

Subject: Re: maximum LUN
Posted by R.Bauer on Tue, 25 Nov 2008 08:34:09 GMT
View Forum Message <> Reply to Message

Jeremy Bailin schrieb:
> On Nov 24, 1:30 pm, Reimar Bauer <R.Ba...@fz-juelich.de> wrote:
>> greg.a...@googlemail.com schrieb:
>>

>>
>>
>>>> What reason could there possibly be for having hundreds
>>>> of files open simultaneously? The only thing that comes
>>>> immediately to mind for me is a poor programming concept. :-)
>>> I kind of expected that reply... :) I think I do have a valid
>>> application, though. I have an archive of a few thousand satellite
>>> images, each of which has 6 bands, stored in separate files. I've made
>>> an object which can handle the six bands and return a subset image for
>>> a selected region processed as I want it. It also returns the image's
>>> coverage of the region, so that I can fill in any gaps with data from
>>> other images. This I do by opening further image objects. So far, I
>>> can use four of these to make an on-the-fly mosaic before I run out of
>>> LUNs - and it works well up to that point. The program allows zooming
>>> and panning, so I need repeat access to similar regions of the same
>>> files. I think it would be too slow to close and reopen them every
>>> time, so the objects hold the files open. The archive is several
>>> terabytes, changing, and not in my control - so preprocessing is out.
>>> With 128 LUNs instead of 28 I could mosaic about 20 image subsets,
>>> which will be enough for the moment. Still, if the number 128 was just
>>> an arbitrary choice long ago, I'd like to ask early for a bigger one!
>>> cheers,
>>> Greg
>> close and open costs nothing. The code between both lines can be
>> efficient or not.
>>
>> cheers
>> Reimar
>
> Not exactly "nothing", but about 0.2 ms for me:
>
> IDL> s1=systime(/sec) & openr,1,'foo' & close,1 & s2=systime(/sec) &
> print, s2-s1
>   0.00021815300
> IDL> s3=systime(/sec) & s4=systime(/sec) & print, s4-s3
>   6.9141388e-06
>
> -Jeremy.

Now we can discuss about your computers clock ;)

cheers
Reimar

Subject: Re: maximum LUN
Posted by greg.addr on Tue, 25 Nov 2008 10:55:41 GMT

I made some tests with data reads at different separations to see how opening/closing affects the caching. The effect's not large, but appears to favour files held open.

Still, if we agree the cost of opening a file is very low, I don't see why the number open needs to be rationed so strictly.

```
pro filetest
 f1="D:\mydocs\data\hrsc\lev4\2112\h2112_0000.nd4.51" ; ~1 GB each
 f2="D:\mydocs\data\hrsc\lev4\2123\h2123_0000.nd4.50" ;
 a=bytarr(2000,2000)
 for j=0,5 do begin
  distance=(long([0,1e3,5e5,1e6,1e7,1e8]))[j]

  s1=systime(/sec)
  for i=0,9 do begin
     openr,3,f1
     point_lun,3,i*distance
     readu,3,a
     close,3
     openr,3,f2
     point_lun,3,i*distance
     readu,3,a
     close,3
   endfor
   s2=systime(/sec)
   print, "repeat open:",s2-s1

   s1=systime(/sec)
   openr,1,f1
   openr,2,f2
   for i=0,9 do begin
     point_lun,1,i*distance
     readu,1,a
     point_lun,2,i*distance
     readu,2,a
   endfor
   close,1
   close,2
   s2=systime(/sec)
   print, "single open:",s2-s1

   print
 endfor
end
```

IDL> filetest
repeat open: 0.12500000
single open: 0.094000101

repeat open: 0.10899997
single open: 0.078000069

repeat open: 0.14100003
single open: 0.12500000

repeat open: 0.14100003
single open: 0.13999987

repeat open: 0.14100003
single open: 0.12500000

repeat open: 0.15599990
single open: 0.12500000

IDL> filetest
repeat open: 0.13999987
single open: 0.094000101

repeat open: 0.093999863
single open: 0.078000069

repeat open: 0.14100003
single open: 0.13999987

repeat open: 0.14100003
single open: 0.14000010

repeat open: 0.15700006
single open: 0.13999987

repeat open: 0.14100003
single open: 0.14100003

and with the blocks reversed:

IDL> filetest
single open: 0.10899997
repeat open: 0.11000013

single open: 0.078000069
repeat open: 0.092999935

single open: 0.12500000
repeat open: 0.15700006

single open: 0.12500000
repeat open: 0.15599990

single open: 0.12500000
repeat open: 0.15599990

single open: 0.12500000
repeat open: 0.14100003

Subject: Re: maximum LUN
Posted by R.Bauer on Tue, 25 Nov 2008 12:10:03 GMT
View Forum Message <> Reply to Message

greg.addr@googlemail.com schrieb:
> I made some tests with data reads at different separations to see how
> opening/closing affects the caching. The effect's not large, but
> appears to favour files held open.
>
> Still, if we agree the cost of opening a file is very low, I don't see
> why the number open needs to be rationed so strictly.
>

We don't maintain idl from ittvis. It's leisured to discuss with us why
they did something. Add a feature request to their software tracking
page and may be report here what happens (what the developers/product
manager answered).

The idl-pvwave newsgroup is from users for users it is not an official
site from ittvis.

cheers
Reimar

>
> pro filetest
> f1="D:\mydocs\data\hrsc\lev4\2112\h2112_0000.nd4.51" ; ~1 GB each
> f2="D:\mydocs\data\hrsc\lev4\2123\h2123_0000.nd4.50" ;
> a=bytarr(2000,2000)
> for j=0,5 do begin
>   distance=(long([0,1e3,5e5,1e6,1e7,1e8]))[j]
>
>   s1=systime(/sec)
>   for i=0,9 do begin
>   openr,3,f1

```
>       point_lun,3,i*distance
>       readu,3,a
>       close,3
>       openr,3,f2
>       point_lun,3,i*distance
>       readu,3,a
>       close,3
>     endfor
>     s2=systime(/sec)
>     print, "repeat open:",s2-s1
>
>     s1=systime(/sec)
>     openr,1,f1
>     openr,2,f2
>     for i=0,9 do begin
>       point_lun,1,i*distance
>       readu,1,a
>       point_lun,2,i*distance
>       readu,2,a
>     endfor
>     close,1
>     close,2
>     s2=systime(/sec)
>     print, "single open:",s2-s1
>
>     print
>   endfor
>  end
>
>
>  IDL> filetest
>  repeat open:      0.12500000
>  single open:     0.094000101
>
>  repeat open:      0.10899997
>  single open:     0.078000069
>
>  repeat open:      0.14100003
>  single open:      0.12500000
>
>  repeat open:      0.14100003
>  single open:      0.13999987
>
>  repeat open:      0.14100003
>  single open:      0.12500000
>
>  repeat open:      0.15599990
>  single open:      0.12500000
```

>
> IDL> filetest
> repeat open: 0.13999987
> single open: 0.094000101
>
> repeat open: 0.093999863
> single open: 0.078000069
>
> repeat open: 0.14100003
> single open: 0.13999987
>
> repeat open: 0.14100003
> single open: 0.14000010
>
> repeat open: 0.15700006
> single open: 0.13999987
>
> repeat open: 0.14100003
> single open: 0.14100003
>
> and with the blocks reversed:
>
> IDL> filetest
> single open: 0.10899997
> repeat open: 0.11000013
>
> single open: 0.078000069
> repeat open: 0.092999935
>
> single open: 0.12500000
> repeat open: 0.15700006
>
> single open: 0.12500000
> repeat open: 0.15599990
>
> single open: 0.12500000
> repeat open: 0.15599990
>
> single open: 0.12500000
> repeat open: 0.14100003
>

Subject: Re: maximum LUN
Posted by greg.addr on Tue, 25 Nov 2008 12:31:02 GMT
View Forum Message <> Reply to Message

> We don't maintain idl from ittvis. It's leisured to discuss with us why

> *they* did something. Add a feature request to their software tracking
> page and may be report here what happens (what the developers/product
> manager answered).
>
> The idl-pvwave newsgroup is from users for users it is not an official
> site from ittvis.
>
> cheers
> Reimar

Thanks, Reimar - I'm well aware of that.

cheers,
Greg
