Subject: Re: Read large ascii file quickly
Posted by penteado on Thu, 06 Jan 2011 21:53:17 GMT
View Forum Message <> Reply to Message

On Jan 6, 7:39 pm, Paul Magdon <paulmag...@yahoo.de> wrote:
> Dear all,
> I have to read a large ascii file (2.5e+07 lines) into a 5000x5000 intarr. The code below works fine but it takes roughly 15-20min on my machine(not the oldest :)). Any suggestions how to accelerate this?
>
> CODE:
>
> OPENR, unit, file, /GET_LUN
>
> result = INTARR(5000,5000, /NOZERO)
> count=0d
> WHILE (NOT EOF(unit)) DO BEGIN
> READF, unit, a,b,c,d,e,f
> print, count
> result(count) =FIX(f)
> count++
> ENDWHILE
>
> CLOSE, unit & FREE_LUN, unit

Just from that it is hard to say. How is the file organized (can you show a few lines of it)? What are the types of a,b,c,d,e,f?

Just as a side note, if you use free_lun, the call to close is unnecessary. free_lun closes the unit if it is open.

Subject: Re: Read large ascii file quickly
Posted by penteado on Thu, 06 Jan 2011 21:57:30 GMT
View Forum Message <> Reply to Message

On Jan 6, 7:53 pm, Paulo Penteado <pp.pente...@gmail.com> wrote:
>> I have to read a large ascii file (2.5e+07 lines) into a 5000x5000 intarr. The code below works fine but it takes roughly 15-20min on my machine(not the oldest :)). Any suggestions how to accelerate this?
>
>> CODE:
>
>> OPENR, unit, file, /GET_LUN
>
>> result = INTARR(5000,5000, /NOZERO)
>> count=0d

Also, it would be better to count with integers. With a long
(initialized 0L) you can count up to 2^31-1, with ulong (0UL) up to
2^32-1, exactly (without the loss of precision that would eventually
happen with doubles). Those are long enough that the time to count to
that will be an issue before the count overflows, but there are also
the much larger long64 and unsigned long64.

Subject: Re: Read large ascii file quickly
Posted by jeanh on Thu, 06 Jan 2011 22:07:16 GMT
View Forum Message <> Reply to Message

on a side note, don't spend time printing the counter!

Jean

Subject: Re: Read large ascii file quickly
Posted by penteado on Thu, 06 Jan 2011 22:20:45 GMT
View Forum Message <> Reply to Message

On Jan 6, 8:07 pm, jeanh
<jghasb...@DELETETHIS.environmentalmodelers.ANDTHIS.com> wrote:
> on a side note, don't spend time printing the counter!
>
> Jean

Well noted. I imagine nobody is going to read the 25 million numbers
printed on the screen. The most efficient way to read the file will
probably not use a loop anyway.

Subject: Re: Read large ascii file quickly
Posted by Kenneth P. Bowman on Thu, 06 Jan 2011 22:52:12 GMT
View Forum Message <> Reply to Message

In article
<d68424af-61e3-45f0-a7f9-09f05ec6d55e@glegroupsg2000goo.googlegroups.com
> ,
Paul Magdon <paulmagdon@yahoo.de> wrote:

> I have to read a large ascii file (2.5e+07 lines) into a 5000x5000 intarr.
> The code below works fine but it takes roughly 15-20min on my machine(not the oldest :)).
> Any suggestions how to accelerate this?

Read it one time and then write it out in some reasonable binary
format (e.g., netCDF or plain binary).

Ken Bowman

Subject: Re: Read large ascii file quickly
Posted by Paul Magdon on Thu, 06 Jan 2011 22:54:24 GMT
View Forum Message <> Reply to Message

Dear all,
I print the counter just to make sure something is working.

Here are a couple of lines of the ascii:

0.050249,0.058117,0.029660,0.097091,0.358593,1
0.051483,0.061990,0.030229,0.095512,0.371539,1
0.055431,0.062700,0.031565,0.090342,0.382935,1

>
> Well noted. I imagine nobody is going to read the 25 million numbers
> printed on the screen. The most efficient way to read the file will
> probably not use a loop anyway.

Yes, I would love to see another option without a loop but I don't know any
Cheers Paul

Subject: Re: Read large ascii file quickly
Posted by David Fanning on Thu, 06 Jan 2011 23:06:24 GMT
View Forum Message <> Reply to Message

Paul Magdon writes:

> I print the counter just to make sure something is working.
>
> Here are a couple of lines of the ascii:
>
> 0.050249,0.058117,0.029660,0.097091,0.358593,1
> 0.051483,0.061990,0.030229,0.095512,0.371539,1
> 0.055431,0.062700,0.031565,0.090342,0.382935,1

Well, then I would do it like this:

```
OpenR, lun, 'file.dat', /Get_Lun
data = Intarr(5000*5000)
ReadF, lun, data, Format='(44x,I0)'
Free_Lun, lun
```

data = Reform(data, 5000, 5000)

Cheers,

David

--
David Fanning, Ph.D.
Fanning Software Consulting, Inc.
Coyote's Guide to IDL Programming: http://www.dfanning.com/
Sepore ma de ni thui. ("Perhaps thou speakest truth.")

Subject: Re: Read large ascii file quickly
Posted by penteado on Thu, 06 Jan 2011 23:29:06 GMT
View Forum Message <> Reply to Message

On Jan 6, 9:06 pm, David Fanning <n...@dfanning.com> wrote:
> Paul Magdon writes:
>> I print the counter just to make sure something is working.
>
>> Here are a couple of lines of the ascii:
>
>> 0.050249,0.058117,0.029660,0.097091,0.358593,1
>> 0.051483,0.061990,0.030229,0.095512,0.371539,1
>> 0.055431,0.062700,0.031565,0.090342,0.382935,1
>
> Well, then I would do it like this:
>
> OpenR, lun, 'file.dat', /Get_Lun
> data = Intarr(5000*5000)
> ReadF, lun, data, Format='(44x,I0)'
> Free_Lun, lun
> data = Reform(data, 5000, 5000)

It is what I would do for a large file, except that I do not see why
make data 1D then reform it, instead of making it 2D from the start.

Since the file is large and he only wants one column out of 6, I guess
the above would be better than the one-liner

data=reform((read_csv('file.dat')).(5),5000,5000)

Which would use more memory, and I expect would be slower.

Anyway, if the same file is going to be read several times, it may be
useful to convert it to some binary format, like Ken suggested (might
even be a savefile).

Subject: Re: Read large ascii file quickly
Posted by David Fanning on Fri, 07 Jan 2011 00:06:08 GMT
View Forum Message <> Reply to Message

Paulo Penteado writes:

> It is what I would do for a large file, except that I do not see why
> make data 1D then reform it, instead of making it 2D from the start.

I don't know. I'm writing a book. Pedantry, I guess. :-)

> Since the file is large and he only wants one column out of 6, I guess
> the above would be better than the one-liner
>
> data=reform((read_csv('file.dat')).(5),5000,5000)
>
> Which would use more memory, and I expect would be slower.

I would expect it to be about as slow as the loop, but
I've never tried it.

> Anyway, if the same file is going to be read several times, it may be
> useful to convert it to some binary format, like Ken suggested (might
> even be a savefile).

Yeah, people who put big data sets in columns should
be shot, but sometimes you just have to play the hand
you are dealt. :-)

Cheers,

David

--
David Fanning, Ph.D.
Fanning Software Consulting, Inc.
Coyote's Guide to IDL Programming: http://www.dfanning.com/
Sepore ma de ni thui. ("Perhaps thou speakest truth.")

Subject: Re: Read large ascii file quickly
Posted by MC on Fri, 07 Jan 2011 10:29:03 GMT
View Forum Message <> Reply to Message

On Jan 7, 11:54 am, Paul Magdon <paulmag...@yahoo.de> wrote:
> Dear all,
> I print the counter just to make sure something is working.

>
> Here are a couple of lines of the ascii:
>
> 0.050249,0.058117,0.029660,0.097091,0.358593,1
> 0.051483,0.061990,0.030229,0.095512,0.371539,1
> 0.055431,0.062700,0.031565,0.090342,0.382935,1
>
>
>
>> Well noted. I imagine nobody is going to read the 25 million numbers
>> printed on the screen. The most efficient way to read the file will
>> probably not use a loop anyway.
>
> Yes, I would love to see another option without a loop but I don't know any
> Cheers Paul

Printing the counter may be the slowest step Comment that out. Also, I
think you can read the data in one go into a single big array -so no
loop needed as EOF stops the read with an error message -you can then
catch the EOF error generated. Sort of an ugly way of doing it but it
might be faster than looping

Cheers MC

Subject: Re: Read large ascii file quickly
Posted by Paul Magdon on Fri, 07 Jan 2011 12:01:04 GMT
View Forum Message <> Reply to Message

Dear All,
I tried some of your ideas:

1.) @ pp data=reform((read_csv('file.dat')).(5),5000,5000)

works fine but takes the same time as with the loop. (David was expecting that :))

2.)@ David ReadF, lun, data, Format='(44x,I0)'

does not work as the dataset also includes negative values where 44x is not correct any longer

3.) @ MC Also, I think you can read the data in one go into a single big array -so no loop needed as EOF stops the read with an error message -you can then
catch the EOF error generated

How to do that?

So, still I didn't have a faster solution.

X.) Yeah, people who put big data sets in columns should be shot
I totally agree but since I need to process so called arff files there is no other way to do it. Our
does anyone knows an implementation of RandomForest classifiers in IDL?

Subject: Re: Read large ascii file quickly
Posted by penteado on Fri, 07 Jan 2011 12:27:16 GMT
View Forum Message <> Reply to Message

On Jan 7, 10:01 am, Paul Magdon <paulmag...@yahoo.de> wrote:
> 2.)@ David ReadF, lun, data, Format='(44x,I0)'
>
> does not work as the dataset also includes negative values where 44x is not correct any longer

```
OpenR, lun, 'file.dat', /Get_Lun
data = replicate({a:fltarr(5),b:0},[5000,5000])
ReadF, lun, data
Free_Lun, lun
data = data.b
```

Subject: Re: Read large ascii file quickly
Posted by edward.s.meinel@aero. on Fri, 07 Jan 2011 14:39:28 GMT
View Forum Message <> Reply to Message

On Jan 6, 4:39 pm, Paul Magdon <paulmag...@yahoo.de> wrote:
> Dear all,
> I have to read a large ascii file (2.5e+07 lines) into a 5000x5000 intarr. The code below works fine but it takes roughly 15-20min on my machine(not the oldest :)). Any suggestions how to accelerate this?
>
> CODE:
>
> OPENR, unit, file, /GET_LUN
>
> result = INTARR(5000,5000, /NOZERO)
> count=0d
> WHILE (NOT EOF(unit)) DO BEGIN
> READF, unit, a,b,c,d,e,f
> print, count
> result(count) =FIX(f)
> count++
> ENDWHILE
>
> CLOSE, unit & FREE_LUN, unit

You could try all of the above suggestions; you could even use

READ_ASCII. But, there is no two ways about it -- reading ascii files
is slow. Always has been, always will be.

Subject: Re: Read large ascii file quickly
Posted by David Fanning on Fri, 07 Jan 2011 15:15:20 GMT
View Forum Message <> Reply to Message

edward.s.meinel@aero.org writes:

> You could try all of the above suggestions; you could even use
> READ_ASCII. But, there is no two ways about it -- reading ascii files
> is slow. Always has been, always will be.

Well, I don't know about that. Reading ill-formed ASCII
files is slow, *because* you have to do it in a loop,
as READ_ASCII does. But, if you have well-formed
ASCII input (usually this means written with some kind
of format statement), I've never found IDL to be slow reading it.

Cheers,

David

--
David Fanning, Ph.D.
Fanning Software Consulting, Inc.
Coyote's Guide to IDL Programming: http://www.dfanning.com/
Sepore ma de ni thui. ("Perhaps thou speakest truth.")

Subject: Re: Read large ascii file quickly
Posted by jeanh on Fri, 07 Jan 2011 17:11:16 GMT
View Forum Message <> Reply to Message

> Printing the counter may be the slowest step Comment that out.
>
> Cheers MC

for 25 000 000 iterations, it takes about 4 minutes on my oldish
computer.... so 20-30% of the time of the OP loop!

Jean

Subject: Re: Read large ascii file quickly
Posted by Kenneth P. Bowman on Fri, 07 Jan 2011 22:40:16 GMT
View Forum Message <> Reply to Message

In article <MPG.2790dbf37472c29b989913@news.giganews.com>,
 David Fanning <news@dfanning.com> wrote:

> Well, I don't know about that. Reading ill-formed ASCII
> files is slow, *because* you have to do it in a loop,
> as READ_ASCII does. But, if you have well-formed
> ASCII input (usually this means written with some kind
> of format statement), I've never found IDL to be slow reading it.

But reading ASCII will always be noticeably slower than
reading the equivalent binary because of the overhead involved
in converting from ASCII to the internal binary representation.
Binary formats, on the other hand, basically just move bytes
from the file to memory.

ASCII files are quite portable, but they are usually inefficient
in terms of storage size. They also rarely carry adequate metadata
within the file itself, but that is a failing of the file's
author rather than the format, per se.

Ah, remember the good old days when you had to try to read
a binary file written on a 60-bit CDC machine on a byte-
oriented machine. :-)

Ken Bowman

Subject: Re: Read large ascii file quickly
Posted by David Fanning on Fri, 07 Jan 2011 23:19:47 GMT
View Forum Message <> Reply to Message

Kenneth P. Bowman writes:

> But reading ASCII will always be noticeably slower than
> reading the equivalent binary because of the overhead involved
> in converting from ASCII to the internal binary representation.
> Binary formats, on the other hand, basically just move bytes
> from the file to memory.

No argument from me here. But "slower" doesn't mean
minutes slower for well-formatted ASCII files. Possibly
some number of seconds slower. But, as someone pointed
out, I don't have a lot of experience with extremely
large ASCII files, as people with big data sets generally

use binary formats.

> ASCII files are quite portable, but they are usually inefficient
> in terms of storage size. They also rarely carry adequate metadata
> within the file itself, but that is a failing of the file's
> author rather than the format, per se.

Well, binary files force good documentation practices because
without it you are utterly and totally without hope. :-)

> Ah, remember the good old days when you had to try to read
> a binary file written on a 60-bit CDC machine on a byte-
> oriented machine. :-)

Before my time, Ken. ;-)

Cheers,

David

--
David Fanning, Ph.D.
Fanning Software Consulting, Inc.
Coyote's Guide to IDL Programming: http://www.dfanning.com/
Sepore ma de ni thui. ("Perhaps thou speakest truth.")

Subject: Re: Read large ascii file quickly
Posted by Kenneth P. Bowman on Sun, 09 Jan 2011 20:08:24 GMT
View Forum Message <> Reply to Message

In article <MPG.27914d7ad0fe57dd989914@news.giganews.com>,
David Fanning <news@dfanning.com> wrote:

> No argument from me here. But "slower" doesn't mean
> minutes slower for well-formatted ASCII files. Possibly
> some number of seconds slower. But, as someone pointed
> out, I don't have a lot of experience with extremely
> large ASCII files, as people with big data sets generally
> use binary formats.

In my experience ASCII files larger than a few 10's of MB
quickly become tedious for interactive work (and for batch
work when you have many of them). Gigabyte sized files
... forget it.

We find it to be worthwhile to convert virtually all ASCII files

to netCDF.

> Well, binary files force good documentation practices because
> without it you are utterly and totally without hope. :-)

If only *that* were true! Fortunately I very rarely see a
plain binary file anymore.

Ken
