Subject: Re: Can i avoid the loop, help me speed up, thanks Posted by ben.bighair on Mon. 04 Aug 2008 20:44:18 GMT

View Forum Message <> Reply to Message

On Aug 4, 9:29 am, Rongchang Chen <chenrc1...@gmail.com> wrote:

- > I wrote a procedure to create sinograms from projections in
- > tomography, the main part of procedure please see below.
- > For large size and number projections, it's very very slow.
- > Can i avoid the loop(one is OK) to speed up, or another way to create
- > sinograms?
- > Thank you very much!!

```
******
```

- > n_sinogra:number of sinogram
- > n_projection:number of projection
- > files_projection:a string vector contain Directory and name of
- > projection
- files sino: a string vector contain Directory and name of sinogram

```
>
  for jj = 0,n_sinogram-1 do begin
     print, 'now creating', jj+1, 'th sinogram'
>
     sino = fltarr(sizepro[0],n projection)
>
>
     for ii=0, n_projection-1 do begin
>
        image = float(read_image(files_projection[ii]))
>
        some processing of image
>
        sino(*,ii) = image(*,jj)
>
     endfor
>
```

>

write_tiff,files_sino(jj),sino,/short,/float

endfor

Hi,

I don't think it is possible for anyone to penetrate where you are having trouble with the given information. I think you might try using the builtin PROFILER routine for a start. It should reveal to you where you are spending most of your time.

Unrelated to the speed issue, you seem to be specifying TIFF output simultaneously as a SHORT integer and a FLOAT. What type of image do you want to be saving?

Cheers, Ben

View Forum Message <> Reply to Message On Aug 4, 3:44 pm, "ben.bighair" <ben.bigh...@gmail.com> wrote: > On Aug 4, 9:29 am, Rongchang Chen <chenrc1...@gmail.com> wrote: > > > >> I wrote a procedure to create sinograms from projections in >> tomography, the main part of procedure please see below. >> For large size and number projections, it's very very slow. >> Can i avoid the loop(one is OK) to speed up,or another way to create >> sinograms? >> Thank you very much!! >> n_sinogra:number of sinogram >> n_projection:number of projection >> files_projection:a string vector contain Directory and name of >> projection >> files_sino:a string vector contain Directory and name of sinogram > >> for jj = 0,n sinogram-1 do begin print, 'now creating', jj+1, 'th sinogram' >> sino = fltarr(sizepro[0],n projection) >> > for ii=0, n_projection-1 do begin >> image = float(read_image(files_projection[ii])) >> some processing of image >> sino(*,ii) = image(*,jj)>> endfor > write_tiff,files_sino(jj),sino,/short,/float >> endfor ****** > Hi, > > I don't think it is possible for anyone to penetrate where you are > having trouble with the given information. I think you might try > using the builtin PROFILER routine for a start. It should reveal to > you where you are spending most of your time. > > Unrelated to the speed issue, you seem to be specifying TIFF output > simultaneously as a SHORT integer and a FLOAT. What type of image do you want to be saving?

> Cheers,

I agree with Ben - the question is: what is in the "some processing of image" step? If this can be "vectorized", then you might be able to avoid some looping.

Subject: Re: Can i avoid the loop,help me speed up,thanks Posted by Chris[6] on Tue, 05 Aug 2008 01:35:14 GMT View Forum Message <> Reply to Message

```
On Aug 4, 11:31 am, Vince Hradil <a href="mailto:hrad...@yahoo.com">hradil <a href="mailto:hrad...@yahoo.com">hrad...@yahoo.com</a>>
> On Aug 4, 3:44 pm, "ben.bighair" <ben.bigh...@gmail.com> wrote:
>
>
>> On Aug 4, 9:29 am, Rongchang Chen <chenrc1...@gmail.com> wrote:
>
>>> I wrote a procedure to create sinograms from projections in
>>> tomography, the main part of procedure please see below.
>>> For large size and number projections, it's very very slow.
>>> Can i avoid the loop(one is OK) to speed up,or another way to create
>>> sinograms?
>>> Thank you very much!!
>>> *********
>>> n_sinogra:number of sinogram
>>> n_projection:number of projection
>>> files_projection:a string vector contain Directory and name of
>>> projection
>>> files_sino:a string vector contain Directory and name of sinogram
>>> for jj = 0,n_sinogram-1 do begin
                print, 'now creating', jj+1, 'th sinogram'
>>>
                sino = fltarr(sizepro[0],n projection)
>>>
                for ii=0, n_projection-1 do begin
>>>
                     image = float(read_image(files_projection[ii]))
>>>
                     some processing of image
>>>
                     sino(*,ii) = image(*,jj)
>>>
                endfor
>>>
>
                write_tiff,files_sino(jj),sino,/short,/float
>>>
>>> endfor
>>>
>> Hi,
```

```
>> I don't think it is possible for anyone to penetrate where you are
>> having trouble with the given information. I think you might try
>> using the builtin PROFILER routine for a start. It should reveal to
>> you where you are spending most of your time.
>
>> Unrelated to the speed issue, you seem to be specifying TIFF output
>> simultaneously as a SHORT integer and a FLOAT. What type of image do
>> you want to be saving?
>
> Cheers,
>> Ben
>
> I agree with Ben - the question is: what is in the "some processing of
> image" step? If this can be "vectorized", then you might be able to
> avoid some looping.
```

I would also add that you are taking a huge penalty by opening every file in each outer loop iteration. IO is very, very slow. If you are looking for a speedup, see if you can reverse the order of the loops and read each file only once. The next step is to see if you can eliminate your (currently) outer loop through vectorization, but this requires a more detailed description of what kind of image processing you are doing.

Chris

Subject: Re: Can i avoid the loop,help me speed up,thanks Posted by Jeremy Bailin on Tue, 05 Aug 2008 11:48:21 GMT View Forum Message <> Reply to Message

```
>>> n_sinogra:number of sinogram
>>> n projection:number of projection
>>> files_projection:a string vector contain Directory and name of
>>> projection
>>>> files sino:a string vector contain Directory and name of sinogram
>>> for ii = 0,n sinogram-1 do begin
         print, 'now creating', jj+1, 'th sinogram'
>>>>
        sino = fltarr(sizepro[0],n projection)
>>>>
>
        for ii=0, n_projection-1 do begin
>>>>
           image = float(read image(files projection[ii]))
>>>>
           some processing of image
>>>>
           sino(*,ii) = image(*,jj)
>>>>
         endfor
>>>>
>
        write tiff, files sino(ji), sino, /short, /float
>>>>
>>>> endfor
>>>> *************
>>> Hi,
>>> I don't think it is possible for anyone to penetrate where you are
>>> having trouble with the given information. I think you might try
>>> using the builtin PROFILER routine for a start. It should reveal to
>>> you where you are spending most of your time.
>>> Unrelated to the speed issue, you seem to be specifying TIFF output
>>> simultaneously as a SHORT integer and a FLOAT. What type of image do
>>> you want to be saving?
>>> Cheers,
>>> Ben
>> I agree with Ben - the question is: what is in the "some processing of
>> image" step? If this can be "vectorized", then you might be able to
>> avoid some looping.
>
> I would also add that you are taking a huge penalty by opening every
> file in each outer loop iteration. IO is very, very slow. If you are
> looking for a speedup, see if you can reverse the order of the loops
> and read each file only once. The next step is to see if you can
> eliminate your (currently) outer loop through vectorization, but this
> requires a more detailed description of what kind of image processing
> you are doing.
> Chris
```

I think Chris has hit the nail on the head... without knowing the processing, I don't know exactly what the final answer will look like, but I expect it to be something like:

```
sino = fltarr(sizepro[0],n_projection)
for ii=0l,n_projections-1 do begin
   image = float(read_image(files_projection[ii]))
   some_processing_that_updates_sino
endfor
for jj=0l,n_sinogram-1 do write_tiff,files_sino(jj),sino,/short,/float
```

Hopefully "some_processing_that_updates_sino" can be vectorized, but even if it has to be a loop you should save lots of time.

-Jeremy.

Subject: Re: Can i avoid the loop,help me speed up,thanks Posted by Rongchang Chen on Mon, 11 Aug 2008 14:56:10 GMT View Forum Message <> Reply to Message

```
On Aug 5, 1:48 pm, Jeremy Bailin <astroco...@gmail.com> wrote:
> On Aug 4, 9:35 pm, Chris <beaum...@ifa.hawaii.edu> wrote:
>
>
>
>> On Aug 4, 11:31 am, Vince Hradil <hrad...@yahoo.com> wrote:
>>> On Aug 4, 3:44 pm, "ben.bighair" <ben.bigh...@gmail.com> wrote:
>>> On Aug 4, 9:29 am, Rongchang Chen <chenrc1...@gmail.com> wrote:
>>> > I wrote a procedure to create sinograms from projections in
>>>> > tomography, the main part of procedure please see below.
>>> > For large size and number projections, it's very very slow.
>>> > Can i avoid the loop(one is OK) to speed up,or another way to create
>>>> > sinograms?
>>>> > Thank you very much!!
>
>>>> > **********
>>>> > n sinogra:number of sinogram
>>>> > n projection:number of projection
>>> > files_projection:a string vector contain Directory and name of
>>>> > projection
>>>> > files_sino:a string vector contain Directory and name of sinogram
>>> > for jj = 0,n_sinogram-1 do begin
          print, 'now creating', jj+1, 'th sinogram'
>>>> >
```

```
sino = fltarr(sizepro[0],n_projection)
>>>> >
>
          for ii=0, n_projection-1 do begin
>>>> >
             image = float(read_image(files_projection[ii]))
>>>> >
             some processing of image
>>>> >
             sino(*,ii) = image(*,jj)
>>>> >
>>>> >
          endfor
          write tiff, files sino(jj), sino, /short, /float
>>>> >
>>>> > endfor
>>>> Hi,
>
>>>> I don't think it is possible for anyone to penetrate where you are
>>> having trouble with the given information. I think you might try
>>> using the builtin PROFILER routine for a start. It should reveal to
>>> you where you are spending most of your time.
>
>>>> Unrelated to the speed issue, you seem to be specifying TIFF output
>>> simultaneously as a SHORT integer and a FLOAT. What type of image do
>>>> you want to be saving?
>>>> Cheers,
>>>> Ben
>
>>> I agree with Ben - the question is: what is in the "some processing of
>>> image" step? If this can be "vectorized", then you might be able to
>>> avoid some looping.
>> I would also add that you are taking a huge penalty by opening every
>> file in each outer loop iteration. IO is very, very slow. If you are
>> looking for a speedup, see if you can reverse the order of the loops
>> and read each file only once. The next step is to see if you can
>> eliminate your (currently) outer loop through vectorization, but this
>> requires a more detailed description of what kind of image processing
>> you are doing.
>
>> Chris
> I think Chris has hit the nail on the head... without knowing the
> processing, I don't know exactly what the final answer will look like,
  but I expect it to be something like:
>
> sino = fltarr(sizepro[0],n_projection)
> for ii=0l,n_projections-1 do begin
     image = float(read image(files projection[ii]))
>
     some processing that updates sino
```

- > endfor
- > for jj=0l,n_sinogram-1 do write_tiff,files_sino(jj),sino,/short,/float

>

- > Hopefully "some_processing_that_updates_sino" can be vectorized, but
- > even if it has to be a loop you should save lots of time.

>

> -Jeremy.

Thank you for discussion!

I'm sorry didn't clarify the 'some processing of image', I update please check at the end.

For the output, because i processing the 16bit tiff image, so i choose tiff as output.

I use PROFILER to check and show the I/O take most of time, use Chris suggestion, I read all the projections (in 3D array) in advance and it speed up a lot, but use 3D array I can just read 100 projections, if more it complain' % Array has too many elements.'??? how to avoid this.

I'm thinking dose IDL has some type of file that I can write one row each time without read the file first?? Thank you very much.

```
The PROFILER, /REPORT processing 100 projections IDL> PROFILER, /REPORT
```

Module Type Count Only(s) Avg.(s) Time(s) Avg. (s)

ARG_PRESENT (S) 295614 0.832386 0.000003 0.832386 0.000003

DIALOG_PICKFILE (S) 2 2.387062 1.193531 2.387062 1.193531

FILE_SEARCH (S) 3 0.010046 0.003349 0.010046 0.003349

FINDGEN (S) 1 0.000010 0.000010 0.000010 0.000010

FLOAT (S) 32827 95.750752 0.002917 95.750752 0.002917

FLTARR (S) 327 0.612496 0.001873 0.612496 0.001873

KEYWORD_SET (S) 65652 0.208982 0.000003 0.208982 0.000003

MEAN (U) 34714 0.481618 0.000014 3.597480 0.000104

MOMENT (U) 34714 1.497712 0.000043 3.006887 0.000087

N_ELEMENTS (S) 197036 0.617928 0.000003 0.617928 0.000003

ON_ERROR (S) 131344 0.452073 0.000003 0.452073 0.000003

```
PRINT
           (S)
                326
                     0.163605 0.000502 0.163605
0.000502
PROFILER
             (S)
                    1
                       0.000029 0.000029
                                          0.000029
0.000029
QUERY IMAGE
                (U) 34794 5.392034 0.000155 25.506740
0.000733
QUERY TIFF
               (S) 32846 18.352950 0.000559 18.352950
0.000559
READ IMAGE
               (U) 34794 12.169956 0.000350 152.321188
0.004378
READ TIFF
              (S) 32846 114.154426 0.003475 114.154426
0.003475
          (S) 32827
                                         0.210118
SIZE
                      0.210118 0.000006
0.000006
                      0.000006 0.000006
STRARR
             (S)
                                         0.000006
0.000006
                 325
STRING
            (S)
                      0.000922 0.000003
                                         0.000922
0.000003
STRLEN
            (S) 32846
                        0.096697 0.000003
                                           0.096697
0.000003
STRMID
            (S) 32846
                        0.157079 0.000005
                                           0.157079
0.000005
STRPOS
             (S) 65692
                        0.334411 0.000005
                                           0.334411
0.000005
STRTRIM
             (S)
                       0.000949 0.000003
                  325
                                          0.000949
0.000003
STRUPCASE
               (S) 32846
                         0.147017 0.000004
                                             0.147017
0.000004
TOTAL
           (S) 32826
                      0.813293 0.000025 0.813293
0.000025
WIDGET PROCESS EVENTS
        (S)
              2 0.000057 0.000028
                                    0.000057
0.000028
WRITE_TIFF
              (S)
                   325
                        6.225517 0.019155
                                          6.225517
0.019155
*****
n_sinogra:number of sinogram
n projection:number of projection
```

```
n_sinogra:number of sinogram
n_projection:number of projection
files_projection:a string vector contain Directory and name of
projection
files_sino:a string vector contain Directory and name of sinogram
flat:image same size as projection(image =
float(read_image(files_projection[ii])))
for jj = 0,n_sinogram-1 do begin
    print,'now creating',jj+1,'th sinogram'
    sino = fltarr(sizepro[0],n_projection)
```