Subject: Re: Speeding up multiple file reading
Posted by Paul Van Delst[1] on Thu, 02 Feb 2006 17:53:22 GMT

View Forum Message <> Reply to Message

clivecook59@gmail.com wrote:

```
>  The code for reading the binary files was not written by me and reads a
>  proprietary data format.  I don't think that there is much i can do
>  with that code. However i do perform some operations on the data within
>  the loop and have been trying to relocate them outside of the loop. An
>  example of this is,
>
>  interp_height = 0 + INDGEN(int)*(16 - 0)/FLOAT(int - 1)
>
>  for i =0,count1 -1 do begin
>
>  data = read_binary_function(binary_file(i))
>
>  ch1x = data.ch1x ;1-D
>  ch2x = data.ch1x ;1-D
>  ch3x = data.ch1x ;1-D
>
>  correction = ......
>
>  sigheight(i,*) = height * correction
>
>  ch1(i,*)  = interpol(ch1x,sig_height(i,*),interp_height)
>  ch2(i,*)  = interpol(ch2x,sig_height(i,*),interp_height)
>  ch3(i,*)  = interpol(ch3x,sig_height(i,*),interp_height)
```

It's probably a small gain, but you might think about changing the order of your arrays from

```
   (i,*)
to
   (*,i)
```

IIRC, IDL accesses array elements like Fortran and the latter form is contigous memory.

Also, it can be a pain in the rear, but if you can replace the INTERPOL function with the INTERPOLATE function, things can go a lot faster. It's really only of benefit if the interpolation indices are all the same, and it looks like they are in your case. You reduce your

```
  ch1(i,*)  = interpol(ch1x,sig_height(i,*),interp_height)
  ch2(i,*)  = interpol(ch2x,sig_height(i,*),interp_height)
  ch3(i,*)  = interpol(ch3x,sig_height(i,*),interp_height)
```

to

```
  ...compute interpolation_index array once....
  ch1(i,*)  = interpolate(ch1x,interpolation_index)
  ch2(i,*)  = interpolate(ch2x,interpolation_index)
  ch3(i,*)  = interpolate(ch3x,interpolation_index)
```

Computing the interpolation index array is /very/ easy if sig_height(i,*) is regularly
spaced. If it's not, it's a bit more complicated, but using INTERPOLATE still means you
only have to do it once.

And, you may even be able to use INTERPOLATE outside the loop since you are doing linear
interp and you can supply both X and Y interpolation indices (where the latter would just
be the 0,1,2,3,4,... of your "i" dimension). Not sure about that though.

This is all of the top of my head, so of course check the manual and test.

paulv

>
> endfor
>
> So (not sure if this is explained very well) i am using the interpol
> function to grid the data to a regaular grid governed by the
> interp_height. I have tried to remove these interpol steps from the
> loop but with no luck,
>
> ch1 = interpol(ch1x,sig_height,interp_height)
>
> In this case ch1x,sig_height and interp_height have the same dimensions
> but does not produce the same results as in the loop. I use rebin to
> produce a new 2-D array with the same dimension for interp_height as
> ch1x and sig_height, (transpose(rebin(interp_height,400,100))).
>
> I hope this is clear.
>
> thanks
>
> Clive
>

--
Paul van Delst
CIMSS @ NOAA/NCEP/EMC