Subject: Speeding up multiple file reading
Posted by clivecook59 on Thu, 02 Feb 2006 15:21:14 GMT
View Forum Message <> Reply to Message

Hi,

I have a program where i need to read in multiple files. Currently i read in 6000 binary files using a function i have written. This reads three columns of data out of each file. To do this i use a loop that calls the function to read a file who's data is then added to an array. At the moment it takes around 90 seconds to go through the 6000 files. Is there any way that i can read this data not using a loop? Or at least are there any tips for speeding this up?

thanks

Clive Cook

Subject: Re: Speeding up multiple file reading
Posted by clivecook59 on Thu, 02 Feb 2006 18:44:07 GMT
View Forum Message <> Reply to Message

Thanks for all the suggestions.

How exactly do i calculate the interpolation_index. The sig_height array and the interp_height arrays are both regularly spaced and the same size.

thanks

Clive

Subject: Re: Speeding up multiple file reading
Posted by Paul Van Delst[1] on Thu, 02 Feb 2006 20:32:36 GMT
View Forum Message <> Reply to Message

clivecook59@gmail.com wrote:
> Thanks for all the suggestions.
>
> How exactly do i calculate the interpolation_index. The sig_height
> array and the interp_height arrays are both regularly spaced and the
> same size.
>
> thanks
>

> Clive
>

Well, if you know what your input interval (call it dsig_height) is, then you can simply do:

```
interpolation_index=(interp_height-MIN(sig_height(i,*)))/dsi g_height
```

A complication is when the the interpolation points *may* fall outside the range of your input points. In that case I would do the following:

```
; -- The array of differences between your interpolation
; -- heights and the start of your input heights
dheight = interp_height-MIN(sig_height(i,*))

; -- Only want to work with the interpolated points
; -- *within* the input height range
overlap_index=WHERE( dheight GT 0.0d0 AND $
                     ( MAX( sig_height(i,*) ) - interp_height ) GT 0.0d0, $
                     overlap_count )
if ( overlap_count eq 0 ) then $
  MESSAGE, 'No points to interpolate!'

; -- Compute the useful interpolation indices.
interpolation_index=dheight[overlap_index]/dsig_height
```

This ensures that you are only interpolating the data, not extrapolating past the edges.

It also assumes that:
- the input arguments are MONOTONIC and SORTED in ascending order.
- sig_height consists of values with a REGULAR interval.

You then call the INTERPOLATE function as in my previous post.

And, as always, check the above.

paulv

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