Subject: Re: Faster way to "shift" array?
Posted by Fabzi on Tue, 11 Jun 2013 16:33:07 GMT
View Forum Message <> Reply to Message

On 06/11/2013 03:44 PM, rjp23@le.ac.uk wrote:> On Tuesday, June 11, 2013 2:28:50 PM UTC+1, David Fanning wrote:
>> I'm probably doing it wrong.
> I suspect not :p
>
> I think my issue might be how I then subset a geographic
> region if it crosses the date line.
>
> Thanks for pointing me in the right direction :-)

I also agree that sometimes it is impossible to overcome a subset problem without having to shift the data. Let's take the example of netcdf files, which have the nice capability to be subset _without_ reading the full databox in the memory. If your subset goes something like: [-40, 40] in longitude and your data is in [0, 360] then you can't efficiently use the netcdf COUNT and OFFSET keywords.
Data organized in [0, 360] has the bad property of cutting Europe and Africa in two, while [-180, 180] mostly cuts oceans. It depends on how often you use the data...

I tried this on my laptop:

```
pro test
  t_lon = findgen(7200) / 20.
  data =  (LONARR(3601)+1) ## t_lon
  ; Convert longs
  lon = ((t_lon + 180) MOD 360) - 180
  nl = n_elements(lon)/2.
  print, 'Shift'
  tic
  new_data1 = shift(data, nl)
  toc
  print, 'Concatenate'
  tic
  new_data2 = [data[nl:*, *], data[0:nl-1,*]]
  toc
end
```

```
Shift
% Time elapsed: 0.098950863 seconds.
Concatenate
% Time elapsed: 0.20306301 seconds.
```

It seems that shift is twice as fast as the "array concatenation" solution.