Subject: Re: Best form factor for a NetCDF package of a dataset
Posted by Kenneth P. Bowman on Sat, 18 Jul 2009 13:01:15 GMT
View Forum Message <> Reply to Message

In article
<d2c0f0b6-df06-4d44-b74d-2f9d3127041c@z4g2000prh.googlegroups.com>,
Ed Hyer <ejhyer@gmail.com> wrote:

> Hello data wizards,
>
> I have a largish dataset that I have been tossing around in ASCII, but
> now have been asked to package as NetCDF. It consists of hourly model
> output (8 variables) at 1-degree resolution. The catch is that for any
> given hour, there is no data for ~90% of the globe. Thus, to package
> the data as grids would waste a lot of space on zeroes. However, if I
> simply package it in a "list" form, the amount of data in each file
> will vary, and that may create headaches for people using it.
>
> The "list" form would be analogous to HDF-EOS "Swath" products, which
> I have used successfully. The "grid" form would be, well, a grid.
>
> As data users, how strongly do you prefer data in simple grids? Would
> you be willing to accept 8x the data volume (compressed) in exchange
> for a simpler format?
>
> Looking to collect $0.02,
>
> --Edward H.

Hide the details from the user by gridding the data at the time
it is read.

It is not difficult to store the packed data in a netCDF file. Your unlimited
dimension is the number of grid cells that have data. That will vary from
file to file.

Store the lon and lat indices i and j of the grid cells that have data along
with the values of the dependent variable(s) (a, b, etc.) for those cells as
1-D variables in the netCDF file.

When you read i, j, and a from the netCDF file, create an empty grid
array (a_grid), and then just set

```
   a_grid[i,j] = a
```

I usually write this kind of thing as a function that returns a structure

```
RETURN, {name   : 'name of data', $
```

```
        x : x, $         ;Longitudes
        y : y, $        ;Latitudes
        units : units, $   ;Physical units
        values : a_grid}
```

To use it, you just call

```
data = DATATYPE_READ_NCDF(input_file_name)
```

The structure contains everything you need to know (data plus metadata).

Remember, file IO is very slow compared to memory IO, so this is likely to
be faster than storing the whole grid *and* make much better use of disk space.

Ken Bowman