Accessing climate data via OPeNDAP

Using IDL with remote data

VISualize 2012

Michael D. Galloy

Tech-X Corporation

19 June 2012
Outline

1. Data Access Protocol (DAP)
2. Accessing DAP served data from IDL
3. Convenient set of routines for other SDFs
4. GPULib update
Data Access Protocol (DAP)

1. open standard
2. simple HTTP-based protocol
3. remote access data via an URL
4. access individual variables as well as subsets
5. client libraries for most languages, including web browsers
6. multiple DAP server implementations
DAP data sources for climate data

1. 20th Century Reanalysis at NERSC
2. NASA DAACs
3. NOAA, UCAR, USGS, JPL, COLA, etc.

See more at:

docs.opendap.org/index.php/Dataset_List
20th Century Reanalysis

- 18 variables
- 180 x 91 lat/lon
- 56 ensemble members
- 1460 reading per year (every 6 hrs)
- 140 years, 1871 to 2010

Over 12 TB of finished product, plus raw data
Options for using DAP from within IDL:

1. OPeNDAP IDL client (DLM)
2. IDL’s netCDF 4.0 bindings (sort of)
3. Remote Data Toolkit
netCDF bindings

- netCDF 4.0 able to access DAP, but...
- curl library problem on some platforms, works for:
  - Linux: IDL 8.0+
  - Mac: IDL 8.2+
  - Windows: not yet
```python
filename = 'ps_1871.nc'
file_id = ncdf_open(filename, /nowrite)
var_id = ncdf_varid(file_id, 'ps')
ncdf_varget, file_id, var_id, value, $
    count=[180, 91, 1, 1], $
    offset=[0, 0, 25, 700], $
    stride=lonarr(3) + 1
ncdf_close, file_id
```
netCDF bindings for a DAP URL

```python
url = 'http://portal.nersc.gov/pydap/'
    + '20C_Reanalysis_ensemble/'
    + 'analysis/ps/ps_1871.nc'
file_id = ncdf_open(url, /nowrite)
var_id = ncdf_varid(file_id, 'ps')
ncdf_varget, file_id, var_id, value, 
    count=[180, 91, 1, 1], 
    offset=[0, 0, 25, 700], 
    stride=lonarr(3) + 1
ncdf_close, file_id
```
netCDF: TX_NC_GETDATA

url = 'http://portal.nersc.gov/pydap/' $
    + '20C_Reanalysis_ensemble/' $
    + 'analysis/ps/ps_1871.nc'
ps = tx_nc_getdata(url, 'ps[*, *, 25, 700]')
Special purpose routine

```idl
ps = mg_20c_getdata('ps', 1871, $
    ensemble_member=25, $
    time=700)
```
Attributes and **TX_NC_DUMP**

IDL> print, tx_nc_getdata(url, 'ps.standard_name')
surface_air_pressure
IDL> tx_nc_dump, url
+ FILE <http://portal.nersc.gov/pydap/20C_...>
   - VARIABLE fltarr(91) lat
     . ATTRIBUTE units = 'degrees_north'
     . ATTRIBUTE standard_name = 'latitude'
   - VARIABLE lonarr(56) ensemble_member
     . ATTRIBUTE long_name = 'ensemble member...'
...

HDF5: TX_H5_GETDATA

```python
f = filepath('hdf5_test.h5', $
    subdir=['examples', 'data'])
res = mg_h5_getdata(f, $
    '/arrays/3D int array[3, 5:*:2, 0:49:3]')
class = mg_h5_getdata(f, '/images/Eskimo.CLASS')
mg_h5_dump, f
```
HDF4: TX_HDF_GETDATA

```python
filename = 'MOD021KM.A2010019.' + '1235.005.2010259102219.hdf'
mg_hdf_dump, filename
sen_az = mg_hdf_getdata(filename, $ 'SensorAzimuth')
```
Save files: TX_SAVE_GETDATA

IDL> cow_filename = file_which('cow10.sav')
IDL> mg_save_dump, cow_filename
Variables: 4

Variables
---------
POLYLIST = lonarr(2321)
...
IDL> polylist = mg_save_getdata(cow_filename, 'polylist')
GPULib 1.6 update

- curve fitting project for next two years (NASA SAGE III mission)
- release soon with:
  - CUDA update,
  - bug fixes,
  - 8-dimensional arrays,
  - optimized scalar/array operations,
  - added a few simple routines like GPUCONJ
Future features

1. MAGMA for GPU accelerated LAPACK routines
2. Levenberg–Marquardt curve fitting
3. ability to create kernels “on the fly” from strings
Conclusion

Questions!

- mgalloy@txcorp.com
- www.txcorp.com/products/GPULib
- michaelgalloy.com