netCDF format in CORDEX

G. Giuliani International Centre for Theorethical Physics - Trieste Earth System Physics Section

ICTP - Earth System Physics Section

Advanced School on Regional Climate Modeling over South America
February 15-19, 2015

Data in Science

- A fourth paradigm after experiment, theory, and computation
- Involves collecting, exploring, visualizing, combining, subsetting, analyzing, and using huge data collections
- Challenges include
  - Deluge of observational data, exaflood of simulation model outputs
  - Need for collaboration among groups, disciplines, communities
  - Finding insights and discoveries in a Sea of Data
- Data-intensive science requires
- New tools, techniques, and infrastructure
- Standards for interoperability
- Institutional support for data stewardship, curation

Roles in Data Intensive Science

- Data users: access, understand, integrate, visualize, analyze, subset, and combine data
- Data scientists: develop infrastructure, standards, conventions, frameworks, data models, Web-based technologies
- Scientists/researchers: acquire, generate, analyze, check, organize, format, document, share, publish research data
- Software developers: develop tools, formats, interfaces, libraries, services
- Data curators: preserve data content and integrity of science data and metadata in archives
- Research funding agencies, professional societies, governments: encourage free and open access to research data, advocate elimination of most access restrictions

Growth in data from Sensors

According to Science article [2011-02-11, Baraniuk]:

- Majority of data generated each year now comes from sensor systems
- Amount generated passed storage capacity in 2007
- In 2010 the world generated 1250 billion gigabytes of data
- Generated data growing at 58% per year
- Storage capacity growing at 40% per year
- We generate more scientific sensor data than we can process, communicate, or store (e.g. LHC)

Data Model

- What is a data Model?

Data Model

- What is a data Model?
  - A collection of data objects

Data Model

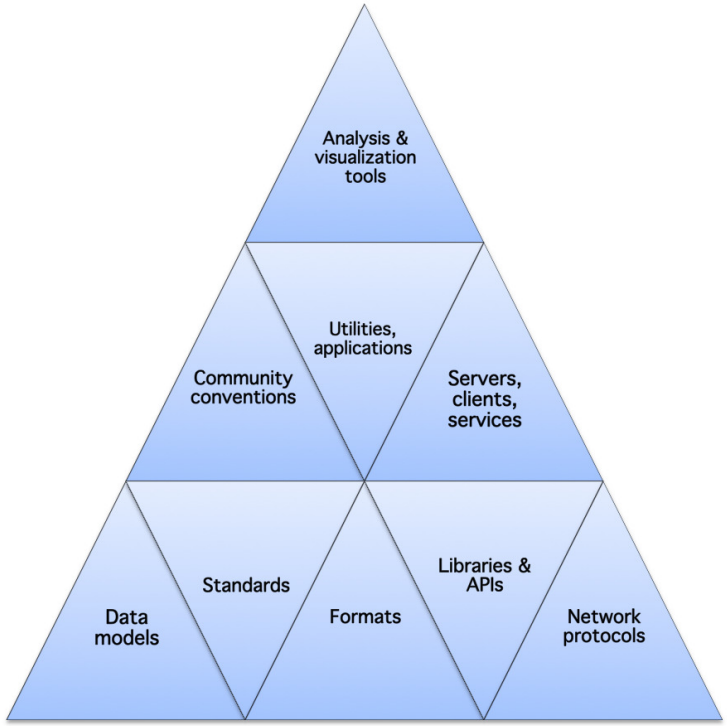
- What is a data Model?
  - A collection of data objects
  - A collection of operations to be applied on data objects such as retrieval, update, subsetting, averaging

Data Model

- What is a data Model?
  - A collection of data objects
  - A collection of operations to be applied on data objects such as retrieval, update, subsetting, averaging
  - A collection of integrity rules that define legal states or change of state

Data Infrastructure

- Applications depend on lower layers
- Sharing requires agreements
  - formats
  - protocols
  - conventions
- Data needs metadata
- Is all this infrastructure really necessary?



Data format

- ASCII

Data format

- ASCII
  - Easy to read if small

Data format

- ASCII
  - Easy to read if small
  - Can get large

Data format

- ASCII
  - Easy to read if small
  - Can get large
  - Have to know structure to make plots

Data format

- ASCII
  - Easy to read if small
  - Can get large
  - Have to know structure to make plots
  - Slow to read, write

Data format

- ASCII
  - Easy to read if small
  - Can get large
  - Have to know structure to make plots
  - Slow to read, write
- Binary

Data format

- ASCII
  - Easy to read if small
  - Can get large
  - Have to know structure to make plots
  - Slow to read, write
- Binary
  - Smaller, faster than ASCII

Data format

- ASCII
  - Easy to read if small
  - Can get large
  - Have to know structure to make plots
  - Slow to read, write
- Binary
  - Smaller, faster than ASCII
  - Have to know structure

Data format

- ASCII
  - Easy to read if small
  - Can get large
  - Have to know structure to make plots
  - Slow to read, write
- Binary
  - Smaller, faster than ASCII
  - Have to know structure
  - Not necessarily portable

Data format

- ASCII
  - Easy to read if small
  - Can get large
  - Have to know structure to make plots
  - Slow to read, write
- Binary
  - Smaller, faster than ASCII
  - Have to know structure
  - Not necessarily portable
  - Opaque from outside application

Data format

- ASCII
  - Easy to read if small
  - Can get large
  - Have to know structure to make plots
  - Slow to read, write
- Binary
  - Smaller, faster than ASCII
  - Have to know structure
  - Not necessarily portable
  - Opaque from outside application
- netCDF

Data format

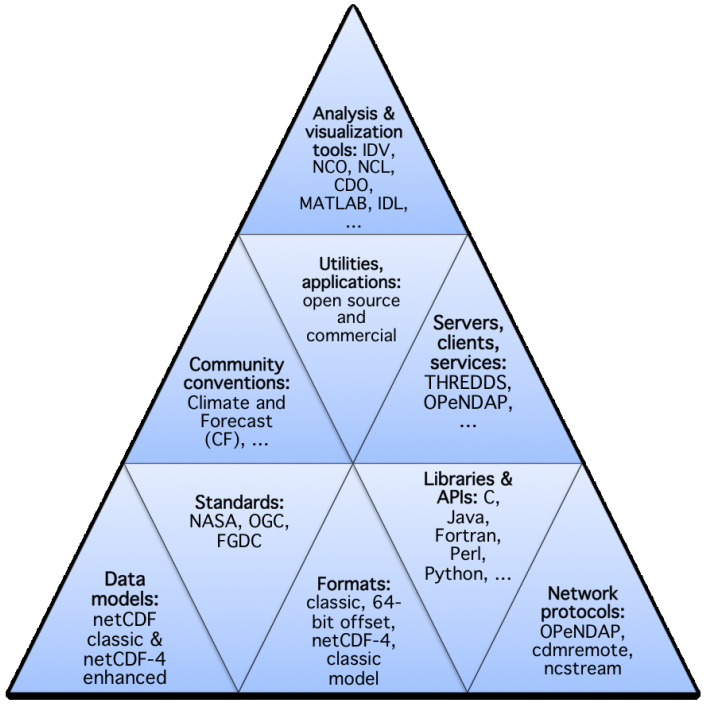
- ASCII
  - Easy to read if small
  - Can get large
  - Have to know structure to make plots
  - Slow to read, write
- Binary
  - Smaller, faster than ASCII
  - Have to know structure
  - Not necessarily portable
  - Opaque from outside application
- netCDF
  - Binary indexed portable format with standard access API for both data and metadata

netCDF Data format

- **Self-Describing**: A file includes metadata as well as data: description of variables, units of measure, etc.
- **Portable**: Data written on one platform can be read on other platforms.
- **Direct-access**: A small subset of a large dataset may be accessed efficiently, without first reading through all the preceding data.
- **Appendable**: Data may be efficiently added to a file without copying the dataset or redefining its structure.
- **Extensible**: Adding new dimensions, variables, or attributes to files does not require changes to existing programs that read the files.
- **Sharable**: One writer and multiple readers may simultaneously access the same file. With Parallel netCDF , multiple writers may efficiently and concurrently write into the same file.
- **Archivable**: Access to all earlier forms of netCDF data will be supported by current and future versions of the software.
- **Networkable**: Client access to remote servers through OPeNDAP.

netCDF Infrastructure

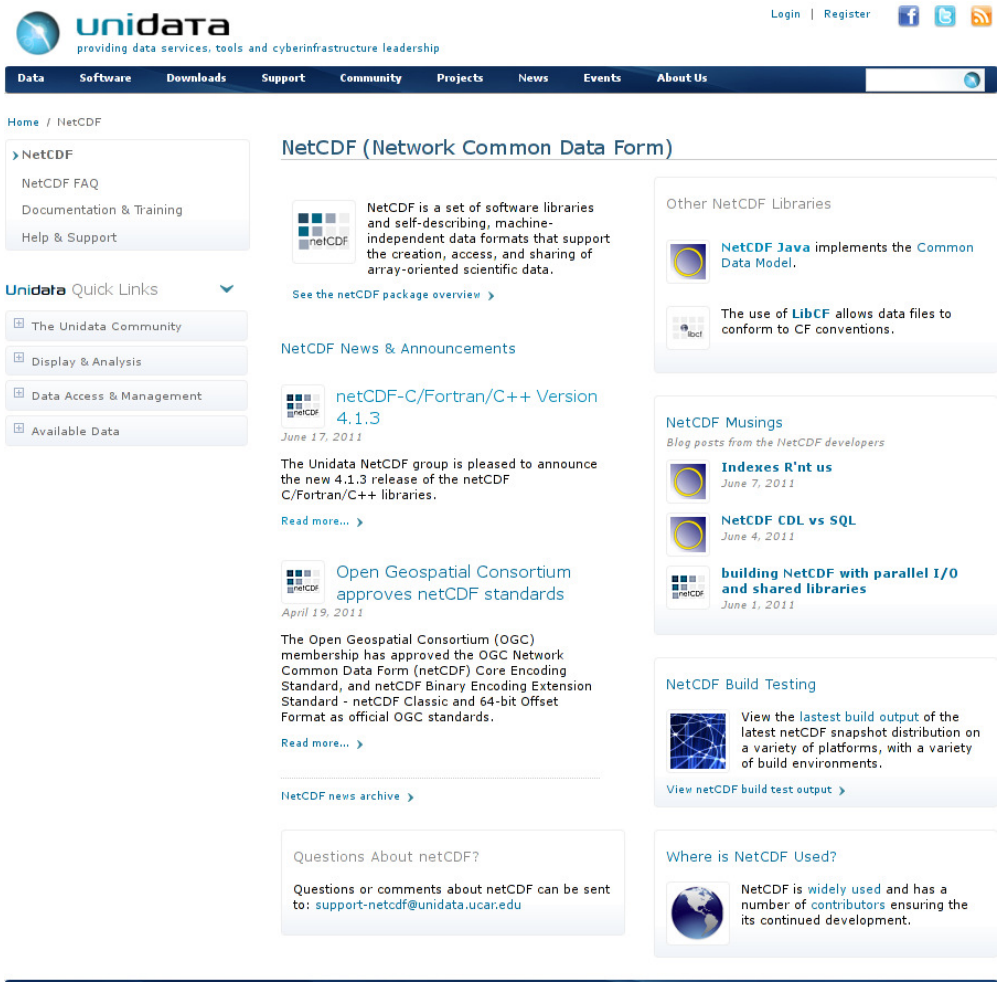
- Provides format and library for netCDF data model
- Endorsed by several standards bodies
- Active conventions communities
- OPeNDAP protocol
- Several servers for remote data access
- Many open source and commercial utilities and applications



netCDF Home

http://www.unidata.ucar.edu/software/netcdf

Unidata Site



netCDF Users

- Climate modelers
  - Program for Climate Model Diagnosis and Intercomparison (PCMDI)
  - Earth Systems Grid
- Ocean and atmospheric sciences
  - Forecast models
  - Atmospheric chemistry
- Neuroimaging
  - MINC - Medical Image NetCDF
  - NiBabel
- Fusion research
  - Culham Centre for Fusion Energy (C++ API for netCDF -4)
- Molecular dynamics simulations (e.g. AMBER)

netCDF Standard endorsement

- **2009-02-05**: NASA Earth Science Data Systems (ESDS) Standards Process Group endorsed netCDF classic and 64-bit offset formats as appropriate for NASA Earth Science data.
- **2010-03-1**: Integrated Ocean Observing System (IOOS) Data Management and Communications (DMAC) Subsystem endorsed netCDF with Climate and Forecast (CF) conventions as a preferred data format.
- **2010-09-27**: Steering Committee of the US Federal Geographic Data Committee (FGDC) officially endorsed netCDF as a Common Encoding Standard.
- **2011-03-07**: Open Geospatial Consortium (OGC) approved "OGC Network Common Data Form (NetCDF) Core Encoding Standard version 1.0" as a new OGC standard.

netCDF Classic Data Model

A netCDF "classic" file is composed of:

- Dimensions
- Variables
- Attributes
- Data

A file can have attributes, dimensions, and variables.
Dimensions are used to specify shapes of variables.
One dimension can be unlimited (record)
A variable can have dimensions and attributes.
Multiple variables can share dimensions (be on a grid).
Variables are of fixed primitive type (char, int, float)

Dimension

Dimensions are used to define shapes of variables.
Each dimension must have:

- Unique name in a file
- A length, i.e. an integer number

Attribute

An attribute is used to store metadata, either at file or variable level. Each attribute must have:

- Unique name in level (file or variable)
- A type
- A value

Metadata are used to establish conventions to share data.
For example, for the Climate and Forecast CF convention, a variable MUST have some attributes (for example units, standard name, etc.), and the convention name itself is a mandatory attribute at file level.

Variable

A variable is the shaped (by dimension) storage of data, defined by its metadata (attributes). Each variable must have:

- Unique name in a file
- A type
- Zero (scalar value) or more dimensions
- Zero or more attributes
- As many data values as specified by its shape

Actual scientific data are stored in variables.

netCDF Common Data Language

```
netcdf snow{   // example of CDL notation
dimensions:
  lon= 9 ;
  lat= 7 ;
  time = unlimited ;  // 3 currently
variables:
  float IR_flux(lon, lat) ;
    IR_flux:units = "W m-2" ;
    IR_flux:_Fill_value = -999 ;
    IR_flux:standard_name=  "downwelling_longwave_flux_in_air";
  float snow_cover(time, lon, lat) ;
    snow_cover:units = "kg m-2" ;
  // global attributes
    :title = "simple example, lacks some conventions" ;
data:
  IR_flux = 200, 201, ... ;
  snow_cover = 0.1, 0.2, 0.0, ... ;
}
```

netCDF V4

The netcdf data model is further extended with the new V4 format, built upon the HDF5 data format.

- Multiple unlimited dimensions
- User defined types and opaque types
- Data can be grouped together
- Compression and chunking
- Native Parallel and HPC oriented.

Data access or creation

- The BASE library is written in C. If performance needed, use C.
- The Fortran interface is a wrapper around the C library.

Programming Example

```
use netcdf
! netCDF file ID and variable ID
integer :: istat
integer :: ncid, varid
! array into which we will read values of 2D netCDF variable
real(8) , dimension(NLAT,NLON) :: tas_array
! Open file with read-only access
istat = nf90_open("foo.nc", NF90_NOWRITE, ncid)
if ( istat /= nf90_noerr ) then
  write(0,*) nf90_strerror(istat)
  stop
end if
! Get the id of the variable named "tas"
istat = nf90_inq_varid(ncid, 'tas', varid)
! Read variable "tas" as doubles, tas_array must be big enough!
istat = nf90_get_var(ncid, varid, tas_array)
! Close the file, freeing all resources.
istat = nf90_close(ncid)
```

CORDEX Data Format

- Data files are NetCDF format, version 4 compressed with zlib deflation, using the NetCDF 4 classic data model and the CF convention 1.4 or later
- Each file may contain only one output field (target variable) from a single simulation. It has to include attributes and coordinate variables respeting a standard format. The entire time series of a target variable has to be distributed over several files.
- All output fields must be of single precision (type NC_FLOAT), while all coordinate variables (time and space) have to be of double precision (type NC_DOUBLE) in accordance with the CMIP5 specifications.

CORDEX Grid

- A "domain" is a region for which the regional downscaling is taking place.
- Target resolution is 44 km (50 mostly used).
- In the Experiment, 13 domains have been identified (+2 high resolution).
- AFRica, Middle east North Africa, North AMerica, Central AMerica, South AMerica, EURope, West ASia, East ASia, Central ASia, AUStralasia, ANTarctica, ARCtic, MEDiterranean (MNA-22, MED-11).

CORDEX Variable Naming and Attributes

- The file MUST have a series of mandatory attributes
- The variables MUST have mandatory names and mandatory attributes
- Vertical defined variables MUST have vertical coordinate
- Time coordinate axis reference MUST be 1949-12-01 00:00:00
- Time statistics variable must have time_bnds coordinate to specify bounds.
- Projected grid should have a Coordinate Reference Systems WCS standard coordinate system description

CORDEX File Naming

- orog_EUR-44_ECMWF-ERAINT_evaluation_r0i0p0_ SMHI-RCA4_v1_fx.nc
- orog_EUR-44_ECMWF-ERAINT_evaluation_r1i1p1_ SMHI-RCA4_v1_fx.nc
- tas_EUR-44_ECMWF-ERAINT_evaluation_r1i1p1_SMHI-RCA4_v1_mon_198812-199011.nc
- tas_AFR-44_CMCC-CMCC-CM_historical_r1i1p1_CLMcom-CCLM4-8-17_v1_mon_194912-195011.nc
- tas_AFR-44_CMCC-CMCC-CM_historical_r1i1p1_CLMcom-CCLM4-8-17_v1_mon_195003-196011.nc
- tas_AFR-44_CMCC-CMCC-CM_historical_r1i1p1_CLMcom-CCLM4-8-17_v1_mon_200012-200511.nc
- tas_AFR-22_MPI-MPI-ESM-LR_historical_r1i1p1_CLMcom-CCLM4-8-17_v1_mon_200101-200512.nc
- tas_EUR-11_CNRM-CERFACS-CNRM-CM5_rcp45_r1i1p1_DMI-HIRHAM5_v1_day_20060101-20101231.nc
- mrro_EUR-11_MPI-MPI-ESM-LR_rcp26_r2i1p1_MPI-CSC-REMO2009_radv_6hr_2091010100-2092010100.nc
- snw_AFR-44_ICHEC-EC-EARTH_rcp85_r1i1p1_SMHI-RCAO-SN_v1_6hr_2091010100-2091123118.nc