Mid-Course Review: NetCDF in the Current Proposal Period

Russ Rew

2011-05-23

General proposal objectives related to netCDF

- Provide high-level interfaces to geoscience data as well as simple mechanisms for locating, accessing, and distributing real-time and thematic data, creating and publishing metadata
  - netCDF/libcf, netCDF-Java, CDM, TDS, NcML, CF conventions
- Adopt, develop, and promote open standards, conventions and protocols for data formats, access, and metadata to enhance interoperability of data services
  - TDS, netCDF, CF conventions, OPeNDAP
- Facilitate seamless integration and analysis of data from diverse sources, including GIS information
  - TDS/ncWMS, netCDF, CF conventions, Windows ports, work with ESRI
- Enable users and partner institutions with data holdings to contribute and share their data easily
  - TDS, RAMADDA, netCDF, CF conventions
- Provide extensible frameworks for creating next generation, dynamic case study datasets
  - RAMADDA, TDS, netCDF-Java, CDM, CF conventions

Specific netCDF objectives, mostly completed

- Contribute to agreement on suitable metadata conventions
- Facilitate governance needed to sustain metadata conventions
- Facilitate research and education in climate as a new priority area
- Develop metadata conventions for observational data
- Improve compatibility between netCDF APIs and OPeNDAP protocols
- Improve netCDF performance, with support for stream access
- Develop library support for structured grids
- Improve netCDF APIs to support netCDF-4 and to exploit language advances
- Enhance netCDF utility programs with performance-related attribute representations
- Support CF conventions with Java and C-based reference implementations

Specific netCDF objectives, partly completed

- Develop guidance for use of netCDF-4 compression, chunking, and data structuring facilities
- Provide more benchmarks to improve performance
- Enhance library support for CF conventions
  - new observational data conventions in libCF
  - climate modeling calendars
  - interoperability with OGC and ISO data models

Specific netCDF objectives not started

- Enhance netCDF utility programs with comprehensive NcML support
- Modify C-based libraries to improve thread safety
- Provide interfaces for CF standard names that offer query functionality
- Provide generic examples (such as nccopy) for each language interface

Unanticipated tasks that were not in proposal

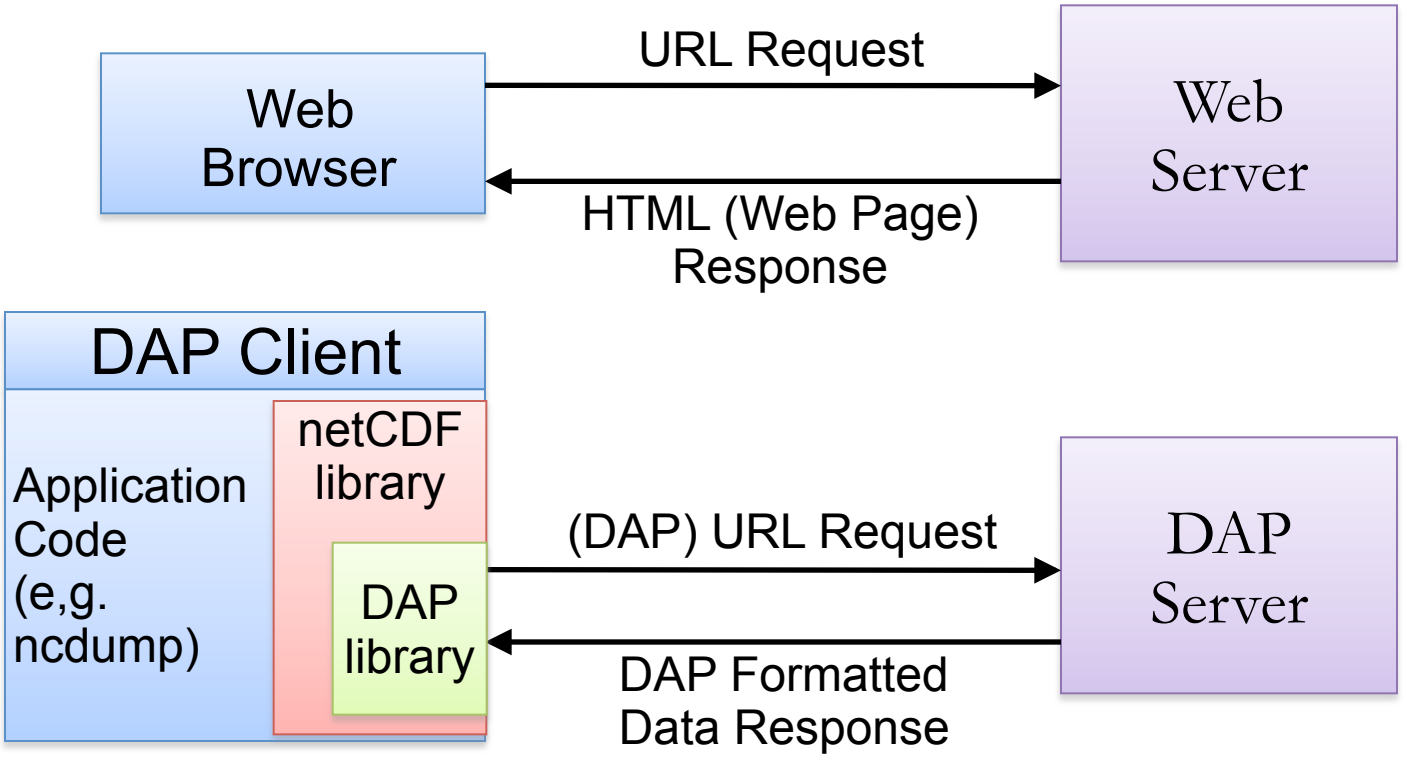
- Refactor netCDF C library architecture to allow plug-in extensions for other formats, as supported in netCDF-Java
- Implement Gridspec as part of libCF
- Add ability to read HDF4 and HDF5 datasets (e.g. Aura satellite data) through netCDF-4 API
- Add more functions to netCDF-4 for writing generic tools
- Provide nc-config utility, to make compiling and linking netCDF applications easier.
- Enhance nccopy utility to compress and to chunk by dimension
- Merge library for units handling into C-based netCDF
- Integrate Windows support into Linux build and test framework

Highlights of C-based netCDF status

- OPeNDAP client integration
- Refactored library architecture
- Improved performance
- Generic tools
- NetCDF-4 adoption
- Standards endorsements
- Gridspec API for structured grids
- Transparency in development process

DAP client-server architecture

- DAP data access is analogous to accessing a web page through a web browser



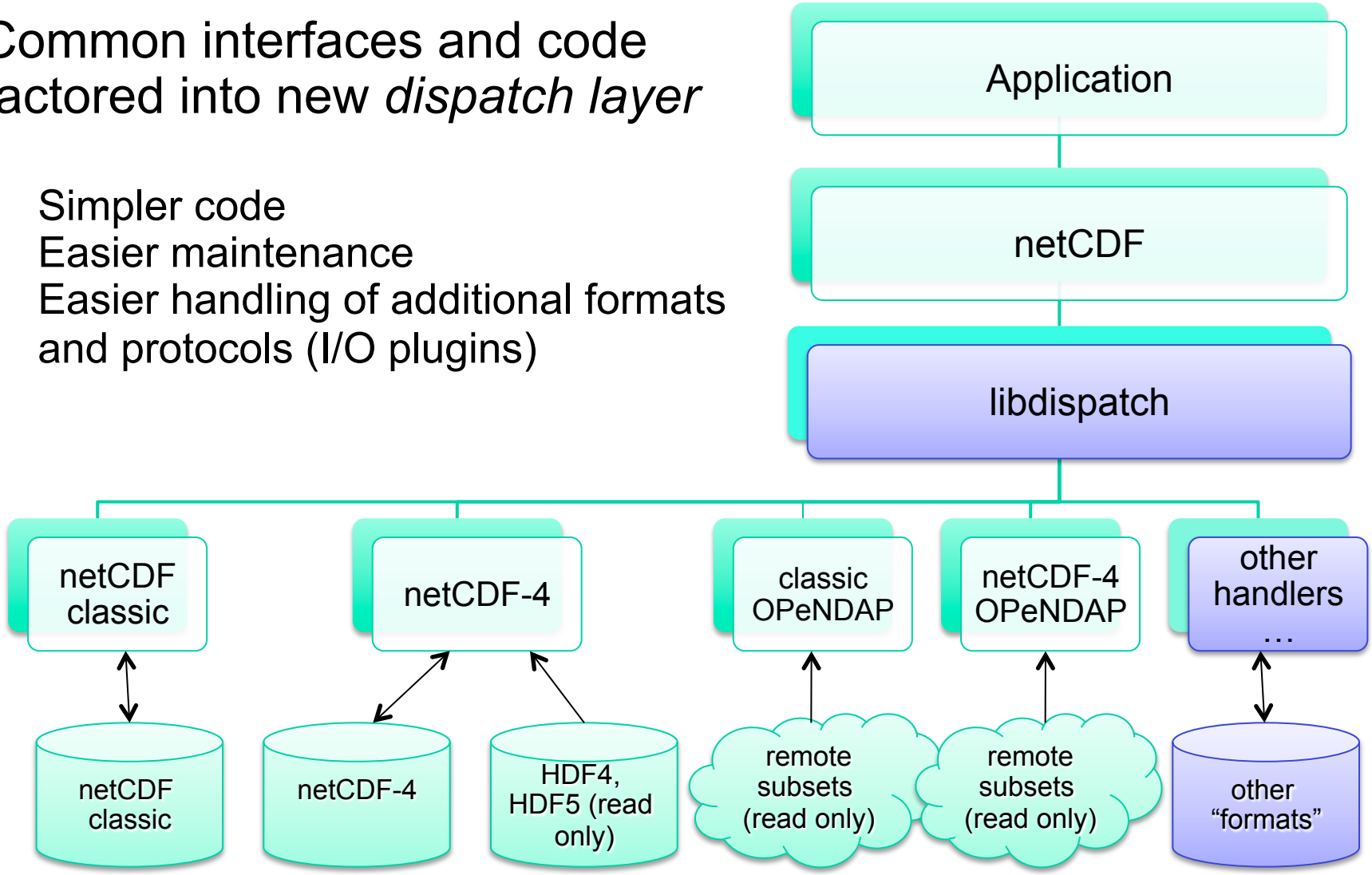
NetCDF/OPeNDAP integration a success

- OPeNDAP access from C-based clients
  - Supports subset access of selected variables, subarrays
  - Subset access using DAP often much faster than whole file access with FTP
  - Example: NCDC saves over 80% of bandwidth using OPeNDAP instead of FTP for NOMADS data access
- Benefits from remote access
  - Accessing data from other formats
  - NcML virtual datasets and aggregation
  - Subset access by coordinates

C library refactored for interoperability

Common interfaces and code factored into new *dispatch layer*

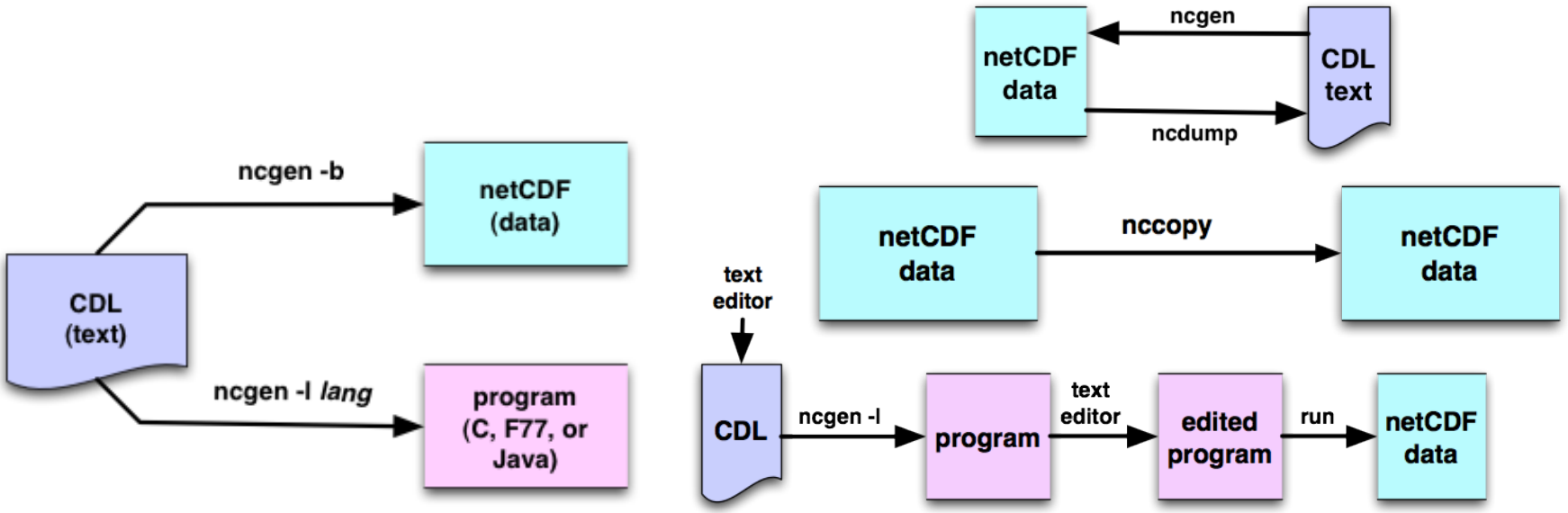
- Simpler code
- Easier maintenance
- Easier handling of additional formats and protocols (I/O plugins)



Performance improvements

- Refactored read code for large speedup on opening netCDF-4 files with compressed or chunked variables
- Sped up variable and dimension lookup by name
- Reduced memory footprint
- Improved handling large number of netCDF-4 attributes and variables
- Enhanced caching to remote access for OPeNDAP client

Generic tools



Adapted generic tools to netCDF-4 data model

ncdump: converts netCDF data to CDL text form

ncgen: converts CDL text to netCDF data or generates program

nccopy: copies, converts, compresses, chunks netCDF data

Status of netCDF-4 adoption

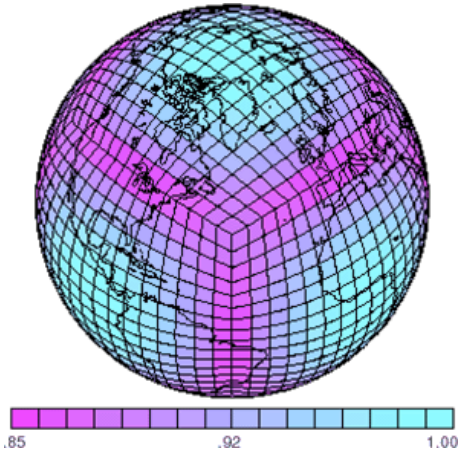
- **NetCDF-4 classic model** support in analysis and visualization apps: IDL, GrADS, CDAT, MATLAB, IDV, NCO, NCL, CDO, PyNGL, ncview, Panoply, Ferret, OGC WMS and WCS clients
- Data providers using **netCDF-4 classic model format** for transparent compression and chunking: within NASA, NOAA, GFDL, COLA, …
- **NetCDF-4 enhanced model** support in language APIs: C, Java (read only), C++ (beta), Fortran
- Partial support for netCDF-4 enhanced model also in NCO, NCL, Panoply, Python API, …

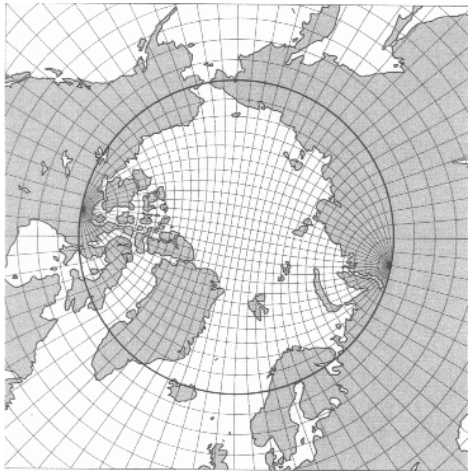
Standards endorsements

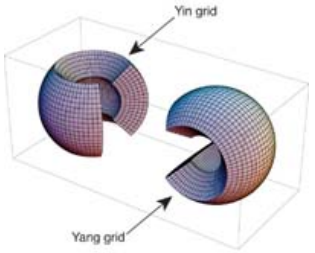
- **2009**: NASA ESDS endorses netCDF-3 formats for earth science data
- **2010**: US FGDC endorses netCDF-3 and netCDF-4 as “Common Encoding Standards”
- **2010**: NASA ESDS endorses CF Metadata Conventions as a standard for earth science data
- **2011**: Unidata TDS includes ncISO services providing ISO-19115 metadata (from NOAA/NGDC)
- **2011**: OGC approves netCDF as a core binary encoding standard
- Additional standards underway for netCDF-4, NcML, CF conventions

Libcf: Gridspec additions

- Proposed CF extension for complex grids, such as cubed-sphere grid
- Specifies multi-file implementation of host file, grid files, data files, and mosaic files
- Supports regridding







Transparency in development process

- Open source code repository
  - svn.unidata.ucar.edu/
- Open bug issue tracking, development plans
  - www.unidata.ucar.edu/jira/
- Unidata developers blog
  - www.unidata.ucar.edu/blogs/developer/
- Goals
  - Make participation in open-source development practical
  - Leverage community efforts

Conclusions

- Most of what was proposed has been accomplished
- Proposal didn't anticipate all needed development
  - Refactoring C library
  - Gridspec API
  - New portability approaches
- Important challenges remain
  - Supporting large user community, advancing data services and useful tools, evolving standards and conventions
- Unidata's leadership in infrastructure development and support continue to be important to Earth science modeling and observational communities