NetCDF-4: Benefits and Advice for Data Providers

Edward Hartnett
and R. K. Rew

89th AMS Annual Meeting, 11-15 January 2009
Introduction to NetCDF

- NetCDF is a set of software libraries and machine-independent data formats that support the creation, access, and sharing of array-oriented scientific data.
- First released in 1989.
- NetCDF-4.0 (June, 2008) introduces many new features, while maintaining full code and data compatibility.
- In this talk we advise data producers about when and how to use new netCDF-4 features.
NetCDF Versions

- The C/Fortran software library has a version. 3.6.3 was the last release in the 3.x series.
- The v2 C/F77 API was released soon after netCDF was released. Code for the v2 API is still valid for the 4.0 release.
- The v3 C/F77 API was released in 1993 and represents a complete rewrite of the netCDF API.
- The v4 C/F77 API was released in 2008 and is a superset of the v3 API.
NetCDF Binary Formats

- Classic format - The original binary format of netCDF is still fully supported (and is the default format for created files).

- 64-bit offset format – Introduced in version 3.6.0 (2005) this format is much like the classic format, but with some relaxed size limits.

- netCDF-4/HDF5 format – introduced in 2008, this is a HDF5 file that can be read by netCDF (as well as HDF5) programs.
The netCDF data model, consisting of variables, dimensions, and attributes (the classic model), has been expanded in version 4.0.

The enhanced 4.0 model adds expandable dimensions, strings, 64-bit integers, unsigned integers, groups and user-defined types.

The 4.0 release also adds some features that need not use the enhanced model, like compression, chunking, endianness control, checksums, parallel I/O.
NetCDF Language APIs

- NetCDF-4 developed and maintained in C.
- Java API version 4.0 (out as a development release) can read, but not write, netCDF-4/HDF5 files.
- Fortran 77 API mirrors C API.
- Fortran 90 API has been extended to support netCDF-4.
- C++ can create and read netCDF-4/HDF5 files, but classic model only.
Upgrading without Converting to NetCDF-4

- Users should upgrade to the latest version of netCDF, to ensure that they are taking advantage of the latest enhancements, bug fixes, and performance improvements.

- The 4.0 release is a drop-in replacements for netCDF-3.x. Upgrading will not change the output of your programs.

- The default for these releases is to build the classic library without netCDF-4/HDF5 features. This must be explicitly turned on during install of netCDF-4.
Upgrading to NetCDF-4 with Classic Model Compatibility

- To create a netCDF-4/HDF5 file, supply the proper argument to the create mode parameter when creating a netCDF file.
- These files are transparently read by netCDF applications that have been relinked to the 4.0 version of the library.
- Without any other changes, size limits are removed.
- Writer can also use compression, endianness control, chunking, checksums, or parallel I/O, without any changes in reading programs.
New functions have been added to the API to set variables' compression, endianness, checksum, or chunking.

These must be set after a variable is defined, but before netCDF metadata is written to the file.

This can be easily added to existing netCDF code.

These are transparent to readers (except for performance).
Using Chunking

- Chunking controls the size and shape of data access blocks. If matched to your access patterns, setting the correct chunk sizes results in significant performance improvement.
- If you are I/O bound, examine chunking carefully. Otherwise, accept netCDF-4 defaults.
- Use the largest chunk that suits your unit of access.
Using Parallel I/O

- NetCDF-4 support parallel I/O for netCDF-4/HDF5 files only.
- Create or open file with special functions (nc_open_par, nc_create_par in C).
- When using parallel I/O to create a file, the resulting file is an ordinary netCDF-4/HDF5 file.
- Create/open file, and create metadata, collectively in every process. Then each process can read/write data independently.
Code conversion from sequential to parallel I/O is generally easy. File and metadata creation code needs to be run on all processes (collectively). Each process then writes/reads its own domain by changing the start/count parameters to address the global index space.

- Requires a parallel file system.
- Cannot write compressed files with parallel I/O.
Using NetCDF-4 Enhanced Model

- The enhanced model includes groups and user-defined types.
- Files containing these features will be unreadable to existing netCDF software, even after that software has upgraded to netCDF-4. New code must be added to the software to take advantage of the new netCDF-4 features.
- NetCDF utilities ncdump (as of 4.0) and ncgen (as of 4.0.1) handle new netCDF-4 features fully.
Suggested Users of Enhanced Data Model

- Significant performance improvement possible for compound type.
- Suggested use: observational datasets. Nested structures and variable-length or ragged arrays are well-suited to lots of kinds of observed data that cannot be represented very well with the classic model.
- Suggested use: model restart files. They are not widely distributed to users, but contain lots of data.
NASA GMAO – Converted GOES-5 assimilation system to netCDF-4 (using decade-old v2 API code). Immediate reason: variable size limits. Conversion to parallel I/O is underway.

WRF – Will allow users to specify that netCDF-4/HDF5 files are to be used. Existing netCDF code base conserved.
Future Plans for NetCDF

- Upcoming 4.0.1 release contains an experimental netCDF-4 capable ncgen, plus performance enhancements for reading/writing netCDF-4/HDF5 files.
- The 4.1 release in early 2009 will include C-based OPeNDAP client, allowing remote access to data files of many formats on OPeNDAP servers.
- In 4.2 release the OPeNDAP protocol is being expanded to handle enhanced netCDF-4 model.
Unidata's netCDF-4.0 release is a drop-in replacement for any of the netCDF-3.x releases.

Full backward code and data compatibility is assured.

With minor code changes data providers can take advantage of: larger datasets, compression, checksums, endianness, and chunk sizes.
With more significant changes, data providers can take advantage of parallel I/O to dramatically improve performance in high performance computing environments.

New netCDF-4 model features allow better data organization and I/O performance, but will not be compatible with any existing code.
Give Feedback

- For support or to suggest improvements: support-netcdf@unidata.ucar.edu
- Take the netCDF User Survey! https://survey.ucar.edu/opinio/s?s=4409
- Low traffic mailing list: netcdfgroup@unidata.ucar.edu
- Stop by Unidata booth(616).