NetCDF: A Public-Domain-Software Solution to Data-Access Problems for Numerical Modelers

Harry L. Jenter[1] and Richard P. Signell[2]

Abstract

Unidata's network Common Data Form, netCDF, provides users with an efficient set of software for scientific-data storage, retrieval, and manipulation. The netCDF file format is machine-independent, direct-access, self-describing, and in the public domain, thereby alleviating many problems associated with accessing output from large hydrodynamic models. NetCDF has programming interfaces in both the Fortran and C computer language with an interface to C++ planned for release in the future. NetCDF also has an abstract data type that relieves users from understanding details of the binary file structure; data are written and retrieved by an intuitive, user-supplied name rather than by file position. Users are aided further by Unidata's inclusion of the Common Data Language, CDL, a printable text-equivalent of the contents of a netCDF file. Unidata provides numerous operators and utilities for processing netCDF files. In addition, a number of public-domain and proprietary netCDF utilities from other sources are available at this time or will be available later this year. The U.S. Geological Survey has produced and is producing a number of public-domain netCDF utilities.

Introduction

Hydrodynamic modeling is a data-intensive activity. This is particularly true with time-dependent, 3-dimensional circulation models. Modern methods of producing, analyzing, and visualizing simulation data place significant demands on model output formats. For example, because computer networks are commonplace, numerical modelers often use different platforms for different steps of their investigations. In order to be shared efficiently from step to step, data should be machine-independent, i.e., interpretable on a variety of computers without the burden of file-format translation. Another demand is that modelers usually

[1] U.S. Geological Survey, Reston, Virginia 22092

[2] U.S. Geological Survey, Woods Hole, Massachusetts 02543

view output on 2-dimensional media—e.g. a piece of paper or a computer screen. Therefore, 3- or 4-dimensional data must be stored in such a way that it can be reduced easily to 1- or 2-dimensional data before being analyzed. This, in turn, implies that stored data must be direct-access, i.e., organized such that small subsets may be extracted easily from large files. Lastly, numerical modeling is a repetitive process. It is not unusual for a modeler to run a circulation model many times while changing its input parameters only slightly. This results in a number of data files that may be very similar, yet have meaningful differences. Stored data, therefore, should be self-describing, i.e., should include ancillary information describing its origin in sufficient detail to distinguish it from different but similar files.

Each one of these demands, machine-independence, direct-access and self-description, can be demonstrated in concrete terms by examining a specific numerical modeling investigation. The authors are studying the hydrodynamic response of Massachusetts and Cape Cod Bays to winds, tides and freshwater inflows using a time-dependent, 3-dimensional finite-difference model. Typically, the circulation model is run on a supercomputer in Boston, Massachusetts. Afterward, output files are transferred by way of a computer network to graphical workstations in Woods Hole, Massachusetts, and Reston, Virginia, for analysis and visualization. Each of the three machines involved in this process has a different architecture, thus making a machine-independent and network-transportable data format crucial to efficient data processing.

The circulation model produces time series of 3-dimensional salinity, temperature and velocity-vector fields, as well as 2-dimensional, water-level fields. Because the model grid consists of approximately 50,000 cells (68, length; 68, width; 11, depth), one time step of the model produces slightly more than a megabyte of output. One day of hourly values consists of approximately 25 megabytes of data. The model output is analyzed typically by extracting time series of values at single points or by extracting vertical or horizontal planes of values. For efficiency, small subsets of data must be extracted from the output files without reading the entire file, thus making a direct-access data format necessary.

Because the forcing mechanisms of Massachusetts Bay are diverse and complicated, the authors need to make hundreds of model runs to separate the effects of tides, winds, and freshwater inflows. Simple file-organization techniques, such as a systematic file-naming scheme, are not sufficient to distinguish the numerous files. Descriptive information must be included in each output file, making a self-descriptive data format a modeling requirement.

After considering several public-domain file formats (See Treinish (1991) for a good review of file formats.), the authors have adopted netCDF for storing model output. NetCDF's machine-independent, direct-access, self-describing format addresses the constraints on model output organization described above, as well as other lesser constraints such as cost—netCDF is public-domain, and, therefore, available free of charge. The remainder of this paper contains discussions of the basic netCDF format and scientific software for processing netCDF files, as well as an illustrative example of a netCDF application.

The NetCDF data format

NetCDF, network Common Data Form, was developed at the Unidata Program Center as part of Unidata, a nationwide data-processing-software development program sponsored by the National Science Foundation's Division of Atmospheric Sciences and managed by the University Consortium for Atmospheric Research. Unidata maintains and distributes the netCDF software free of charge. At present, this consists of FORTRAN- and C-callable subroutines for reading and writing netCDF files, a user's manual (Unidata Program Center, 1991) and a number of stand-alone programs for manipulating netCDF files.

NetCDF files are encoded in XDR, a nonproprietary binary external data representation developed by Sun Microsystems, Incorporated[3]. Because of this, netCDF files can be exchanged among any machines on which XDR is implemented. Almost all computers provide access to XDR. Examples include Apple Macintoshes, IBM PCs, Sun workstations, DEC workstations, DEC VAXes, IBM mainframes, and Cray supercomputers. XDR also permits file transmittal over computer networks without data format modification.

Data addressing information stored in netCDF files permits access to subsets of data without requiring preceding data in the file to be read. This direct-access capability allows efficient storage and retrieval of data. Large amounts of model output can be stored in a single file without hindering a user's ability to extract small subsets of data.

NetCDF files contain ancillary data, or metadata, in addition to the model output. This metadata is stored in a header section of the file. Information, such as data units, model input parameters, or a user's comments, can be stored in the file. There are no practical constraints on the size or content of metadata. The user is free to determine the metadata to be included in the output file. The structure of a netCDF file allows the header section to expand or contract to fit the metadata exactly. There is no wasted space.

Fortunately, Unidata has insulated users from details of the XDR file structure by implementing an abstract data type. This means that the user does not need to understand the binary structure of the file, and that the computer code used to read or write a netCDF file on one machine looks and works exactly the same as it would on any other machine. NetCDF subroutines access the files using low-level C input/output functions, but these are invisible to the user. The user accesses data by named file components using either FORTRAN or C subroutines. The three types of netCDF file components that can be accessed are dimensions, variables, and attributes.

[3] Brand names are for identification purposes only, and do not constitute endorsement by the U.S. Geological Survey.

Standard Attributes	
units	scale_factor
long_name	add_offset
valid_range	C_format
valid_min	FORTRAN_format
valid_max	title
_FillValue	history

Table 1: Unidata's standard netCDF attributes

Dimensions are named integer parameters that describe the shape of data arrays stored in a netCDF file. There is no practical constraint on the size of a dimension. Unidata has implemented one special dimension, the unlimited or record dimension, which, if used, causes a data array to be organized in such a way that it is easy to append values to the array along that dimension. A situation where this is important is the storage of time-dependent data when it is unknown a priori how many time steps are to be stored.

Variables are named arrays of data. There can be multiple variables stored in a single netCDF file, and these variables can be of different data types. Unidata has implemented single-byte, 1-byte character, 2-byte integer, 4-byte integer, 4-byte real number, and 8-byte real number data types. The size of a variable is determined by its dimensions. Variables may have up to 32 dimensions with one of these being the unlimited dimension.

Attributes are ancillary, descriptive data associated either with a variable or with the entire file. The former are called variable attributes, and the latter are called global attributes. Attributes are single values or 1-dimensional arrays of values. They can have any of the data types that variables can have. The user is responsible for deciding what attributes to include to supplement the data in a netCDF file. Unidata has defined a small set of standard attributes that applications which read and write netCDF files are supposed to interpret in a consistent way. The standard attributes are shown in Table 1. A user is free to include these and (or) other attributes.

Even though the XDR encoding of a netCDF file prevents its contents from being viewed directly, it is often useful, or even necessary, to view the file contents at the abstract data level. To facilitate this, Unidata has developed CDL, the Common Data Language. CDL is a printable text-equivalent of the contents of a netCDF file. Figure 1 depicts an example CDL file containing fabricated water-level data.

Interpretation of a CDL file is very intuitive. The CDL in Figure 1 represents the contents of a netCDF file named example.cdf. There are two dimensions, `time` and `station`. `Time` is the optional unlimited dimension. There are three variables: `time`, `station` and `wl`. `Time` and `wl` have the float data type

```
netcdf example{
dimensions:
       time = unlimited;
       station=2;
variables:
        float time(time);
        time:units     = "seconds";
        short station(station);
        station:units  = "identification number";
        float wl(time,station);
        wl:units       = "centimeters";
        wl:long_name   = "water level";
        wl:gage_type   = "Joe's Real Good Gages, Model 1";
        :data_source   = "fabricated for demonstration";
        :comments      = "This file is for demonstrating the
                          basic form of CDL, Unidata's Common
                          Data Language.";
data:
        time = 0., 2.5, 5.0, 7.5, 10.0;
        station = 4321, 1234;
        wl = 1.32, 1.65, 1.78, 1.90, 2.39,
             1.35, 1.59, 1.63, 2.01, 2.65;
}
```

Figure 1: An example Common Data Language (CDL) file

(4-byte real number), and station has the short data type (2-byte integer). Notice that it is acceptable to have dimensions and variables with the same name. The netCDF software assigns each file component an integer identification number. With respect to file access, the dimension time and the variable time are treated as completely different components by the netCDF subroutines. However, other programs that read and write netCDF files are free to assign meaning to dimensions and variables with the same name. There are numerous attributes in the example CDL, including both variable and global attributes. The variable attributes are units, long_name and gage_type. Units and long_name are standard netCDF attributes, and gage_type is unique to the example file. The two global attributes are data_source and comments. Data is displayed at the end of the CDL file.

The NetCDF software

Unidata distinguishes three types of software: subroutines, operators and utilities. Subroutines are computer codes that read, write or modify netCDF files, but must be linked to a user-supplied program before being used. Operators are stand-alone programs that input and output netCDF files. Utilities are stand-alone programs that either input or output netCDF files, but not both.

Unidata distributes the netCDF subroutines. There are FORTRAN- and C-callable subroutine libraries (a C++ library is planned for release in the future.). They produce exactly the same file structure. A netCDF file written by a FORTRAN program can be read by a C program just as easily as one written by a C program. Availability of multiple language interfaces increases flexibility for programmers writing applications that use data stored in netCDF files.

Unidata plans to distribute a large set of netCDF operators later in 1992 (R. K. Rew and S. Emmerson, Unidata, unpublished). These operators will be capable of processing, creating and/or modifying netCDF files. Unidata's proposed operators can be divided into four categories: selectors, combiners, mathematical, and miscellaneous.

Selectors extract subsets of data from a netCDF file. One of the proposed selectors will extract entire variables or variable hyperslabs (array subsets having continuous indices). One will subsample variables at regular intervals, one at user-specified intervals, and one according to user-specified conditional statements. Lastly, one selector will reduce dimensionality by averaging or sampling.

Combiners merge data from more than one netCDF file. One of the proposed combiners will combine files with the same structure by adding to existing arrays an additional dimension which corresponds to the number of files being combined. One combiner will form the union of a group of netCDF files. Lastly, one combiner will concatenate netCDF files and rename variables with conflicting names by appending a numeric suffix to the names.

A mathematical operator will be available for performing element-by-element mathematics on netCDF variables using arbitrary, user-specified, C-language expressions. There will be a calculus operator that differentiates and one that integrates. Also, there will be a convolution operator for filtering data.

A number of miscellaneous operators are planned. These include a sorter, an interpolator, a renamer, a comparator, a packer, and an unpacker. The last two will perform the only type of data compression implemented in netCDF files. For example, 4-byte real numbers can be packed into 2-byte integers by scaling and offsetting the data. The packer and unpacker operators will be used to translate from one data types to the other. Precision is lost in the packing process. However, if it is not important to save numbers with 32-bit precision, the 50% savings in storage space can be helpful. Thirty-two-bit precision is rarely necessary in hydrodynamic modeling.

In addition to distributing the subroutine libraries and proposed operators, Unidata distributes two very useful netCDF utilities: NCGEN and NCDUMP. NCGEN is a utility that inputs a CDL file and outputs either the FORTRAN or C code to produce a netCDF file or outputs the netCDF file itself. This allows a user to design the content of a netCDF file in the intuitive CDL without worrying about coding details. NCGEN allows a user who is unfamiliar with netCDF programming to create a programming template from which to develop netCDF software. NCDUMP takes a netCDF file and produces the equivalent CDL. It has options to dump the entire file or just the metadata. This utility is useful for examining the contents of existing netCDF files.

There are three notable public-domain netCDF utilities in addition to those provided by Unidata. The first is a set of utilities called nctools. Nctools was developed at the U.S. Geological Survey in Woods Hole, Massachusetts. It consists of operating-system-level, command-line versions of each of the netCDF subroutines distributed by Unidata. These are particularly useful for performing single operations on existing netCDF files, such as adding an attribute or renaming a variable. The second public-domain utility is a library of graphics programs known as the Generic Mapping Tools (GMT) (Wessel and Smith, 1991). GMT was developed at the Lamont-Doherty Geological Observatory. GMT scripts can be used to read netCDF files and to produce sophisticated plots, including 3-dimensional surface plots, overlays and contour plots. The last utility is the Scientific, Interactive and Extensible Visualization Environment (SIEVE) (Granger and Johnson, 1992). SIEVE is an interactive X Window System-based graphics environment developed at the U.S. Geological Survey in Reston, Virginia. SIEVE is currently under development and is planned for distribution sometime in 1992.

A number of proprietary utilities operate on netCDF files. Although these tend to be machine-dependent and expensive at present, they serve as an indication that the mainstream computer industry is accepting netCDF as a viable format for storing scientific data.

NetCDF Software Availability

In general, all netCDF software and documentation (including some of the references in this paper) can be obtained by way of anonymous file transfer protocol (ftp) over the Internet computer network. Relevant Internet addresses, machine names, directories, filenames, and descriptive information are provided in Table 2. If you do not understand the meaning of this information, a computer-system administrator at your institution should be able to explain it. The basic procedure for obtaining files by way of anonymous ftp is as follows:

1. From your computer, invoke ftp using the Internet address or machine name from Table 2 — e.g., *ftp 128.117.140.3* or *ftp unidata.ucar.edu.*

2. Login using *anonymous* as the userid and your own userid and machine name as a password — e.g., *hjenter@stress.er.usgs.gov.*

3. Change directories to the directory specified in Table 2 using the *cd* command — e.g., *cd pub.*

4. Set the file type by issuing either the *ascii* or *binary* command. File names that end in the extension *.tar* or *.Z* are binary files. File names that end in the extension *.ps* are ascii files.

5. Retrieve each desired file by issuing the *get* command—e.g., *get netcdf.tar.Z*

6. Logout of the ftp session by issuing the *bye* command.

File names with the *.tar* extension are packed groups of files that must be extracted using the UNIX *tar* command. Files with the *.Z* extension are compressed files that must be uncompressed using the UNIX *uncompress* command before being used. If you are unfamiliar with the *tar* or *uncompress* command, your system administrator should be able to assist you. Files with *.ps* extensions are documents that can be printed on a Postscript-compatible printer.

In addition to acquiring software, a user can participate in a netCDF users' group. The group offers technical assistance by way of electronic mail. To subscribe to the mailing list, send an electronic mail message to the mailing list administrator (*netcdfgroup-adm@unidata.ucar.edu*). To ask for technical assistance, send a problem description to *netcdfgroup@unidata.ucar.edu.* It has been the authors' experience that this mailing group is an efficient and effective mechanism for technical support of the netCDF software.

A NetCDF Example

An example, taken from the authors' experience, is presented here in order to summarize the adaptability and utility of netCDF. Figure 2 depicts an example CDL file from the Massachusetts Bay model. It is essentially the output from

```
netcdf example2 {
dimensions:
     xpos = 68 ;
     ypos = 68 ;
     zpos = 11 ;
     time = UNLIMITED ; // (2 currently)
variables:
     float time(time) ;
          time:long_name = "time" ;
          time:units = "days" ;
     float sigma(zpos) ;
          sigma:long_name = "stretched vertical coordinate
                             levels" ;
          sigma:units = "fraction of depth" ;
     float x(ypos, xpos) ;
          x:long_name = "easting" ;
          x:units = "meters" ;
     float y(ypos, xpos) ;
          y:long_name = "northing" ;
          y:units = "meters" ;
     float depth(ypos, xpos) ;
          depth:long_name = "bathymetry" ;
          depth:units = "meters" ;
          depth:valid_range = 0.f, 270.f ;
     short elev(time, ypos, xpos) ;
          elev:valid_range = -4.f, 4.f ;
          elev:long_name = "elevation" ;
          elev:units = "meters" ;
          elev:scale_factor = 0.00012207404f ;
          elev:add_offset = 0.f ;
     short temp(time, zpos, ypos, xpos) ;
          temp:valid_range = -5.f, 30.f ;
          temp:long_name = "temperature" ;
          temp:units = "degrees Celsius" ;
          temp:scale_factor = 0.00053407392f ;
          temp:add_offset = 12.5f ;
}
```

Figure 2: Example CDL from a Massachusetts Bay Model run.

Machine name Internet address	Directory	File name	Description
unidata.ucar.edu 128.117.140.3	pub	netcdf.tar.Z guide.ps	netCDF subroutines netCDF users' manual
	pub/sdm	ncprogs.ps	Rew & Emmerson, netCDF operators description
crusty.er.usgs.gov 128.128.19.19	/pub/nctools and subdirectories	all files	nctools
sparky.er.usgs.gov 130.11.51.69	pub	sieve.ps	Granger & Johnson, SIEVE description
kiawe.soest.hawaii.edu 128.171.151.16	pub /gmt	all files	GMT software

Table 2: NetCDF software availability

Unidata's NCDUMP utility that was obtained using a netCDF model-output file as input to NCDUMP and specifying the NCDUMP option for printing only metadata. As can be seen from the CDL, netCDF is sufficiently flexible to allow the file to contain 1-, 2-, 3-, and 4-dimensional arrays. These include the model times at which output was written to the file (stored in the array `time`), a description of the model's orthogonal curvilinear grid (stored in the arrays `sigma`, `x`, `y`, and `depth`), a time series of calculated water-level elevations (stored in the array `elev`), and a time series of calculated temperatures (stored in the array `temp`). In order to conserve space in the example, other dependent variables calculated by the model were excluded from the netCDF file. A typical model-output file also includes arrays for salinity and three components of flow velocity.

Note, the dependent variables, `elev` and `temp`, are stored as arrays of 2-byte numbers and that each array has a `scale_factor` and an `add_offset`. This is done to halve the storage requirement for the large time-dependent data arrays produced by the model. Programs, such as the SIEVE visualization software mentioned above, that operate on the dependent variable arrays, convert the data to 4-byte numbers by first multiplying by the `scale_factor` and then adding the `add_offset`.

In summary, the netCDF file format has proven to be an efficient, flexible tool for the authors' hydrodynamic-modeling efforts. It is our experience that our scientific productivity has been enhanced greatly by the ability to use different computers for different modeling tasks, the ability to look at arbitrary subsets of data from model output files, and the ability to organize large numbers of output files for comparison. NetCDF provides the capability for each of these.

References

Granger, G. J. and M. F. Johnson, 1992. An Interactive Scientific Data Visualization Environment within the X Window System. *in press.*

Rew, R. K. and G. P. Davis, 1990. The Unidata netCDF: Software for Scientific Data Access. *Proceedings of the Sixth International Conference on Interactive Information and Processing Systems for Meteorology, Oceanography, and Hydrology*; Anaheim, California; p. 33–40.

Rew, R. K. and S. Emmerson, unpublished. NetCDF Operators and Utilities. Unidata Program Center, Boulder, Colorado. 13 pages.

Treinish, L. 1991. SIGGRAPH '90 Workshop Report: Data Structures and Access Software for Scientific Visualization. *Computer Graphics*, volume 25, number 2, pages 104–118.

Unidata Program Center, 1991. NetCDF User's Guide: An Interface for Data Access. Unidata Program Center, Boulder, Colorado. 150 pages.

Wessel, P. and W. H. F. Smith, 1991. Free Software Helps Map and Display Data, *Eos*, vol. 72, number 41, page 441.