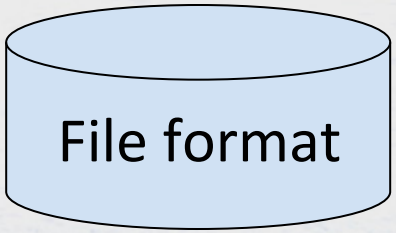
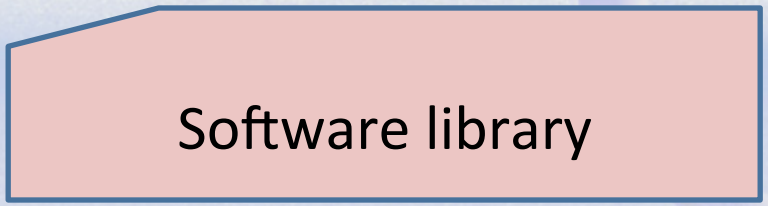

John Caron
Unidata
October
2012
what is netcdf?
And what are its plans for world domination?

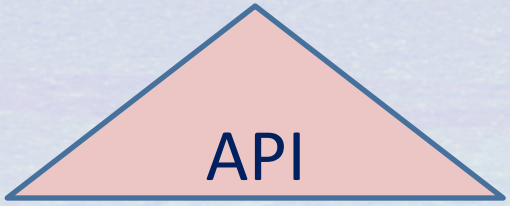
NetCDF is a…



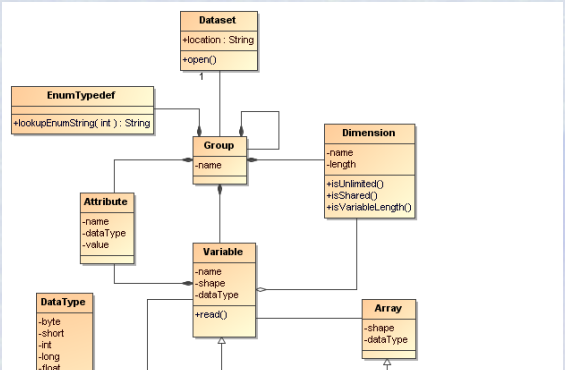
- Store data model objects
- Persistence layer
- NetCDF-3, netCDF-4



- Implements the API
- C, Java, others

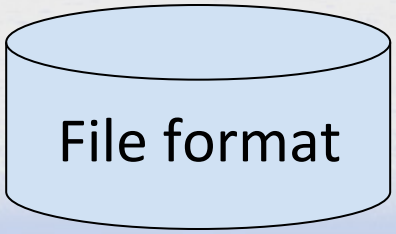


An ***API*** is the interface to the Data Model for a specific programming language



An **Abstract Data Model** describes data objects and what methods you can use on them

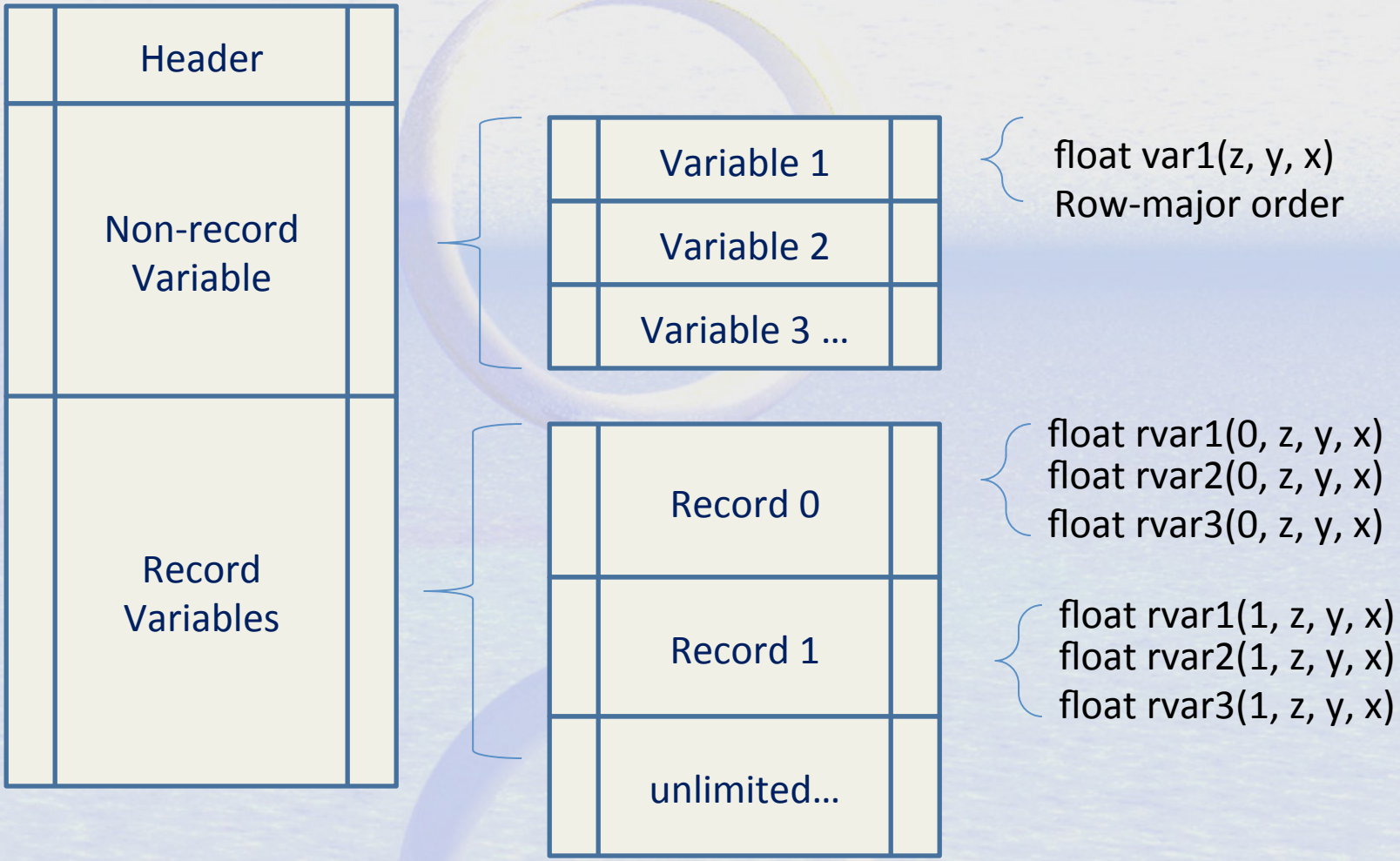
NetCDF is a…



- Stores scientific data
- Persistence layer
- NetCDF-3, netCDF-4

- Portable Format: *Machine, OS, application independent*
- Random Access: *fast subsetting*
- Simple: *self-describing, user accessible"flat file"*
- Documented: *NASA ESE standard, 1 page BNF grammer (netcdf-3)*

NetCDF-3 file format



NetCDF-4 file format

- Built on HDF-5
- Much more complicated than netCDF-3
- Storage efficiency
  - Compression
  - Chunking (can optimize for common I/O pattern)
  - Multiple unlimited dimensions
  - Variable length data

Row vs Column storage

- Traditional RDBMS is a row store
  - All fields for one row in a table are stored together
- Netcdf-3 is a column store
  - All data for one variable is stored together
- Netcdf-4 allows both row and column store
  - Row: compound type
  - Column: regular variable
- Recent commercial RDBMS with column oriented storage, better pergormance in some cases
- NetCDF-3 record variables are like a compound type

NetCDF is a…

Software library

- Reference library in C
  - Fortran, C++, Perl, Python, Matlab, …
- Independent implementation in Java
- others?

- Open Source
- Active community
- Supported

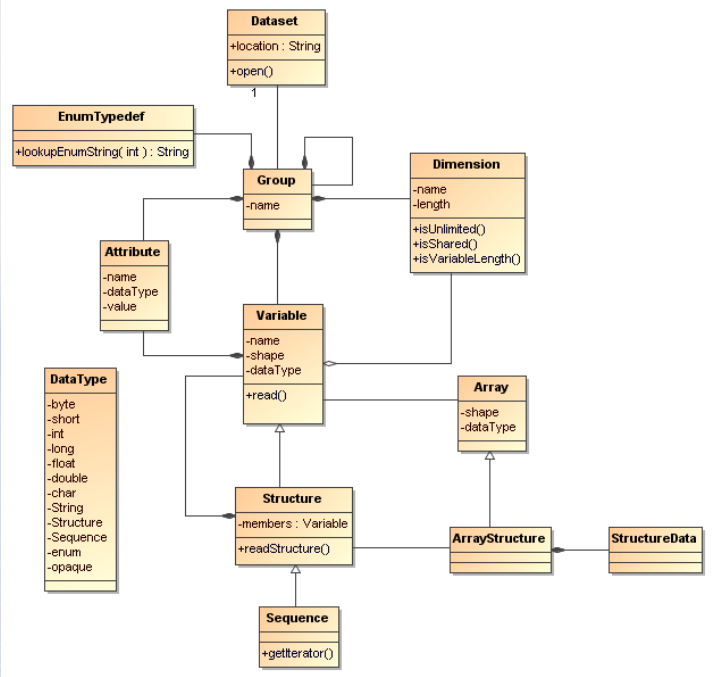
- No reference library, no user group for GRIB, BUFR
  - fragmented, not interoperable, difficult to use

NetCDF is a…

The Application Programming Interface (***API)*** is the interface to the Data Model for a specific programming language.

- Clean separation of concerns
- Information hiding – user never does a seek()
- Stable, backwards compatible
- Easily understood – no surprises
- Interface has small “surface area”

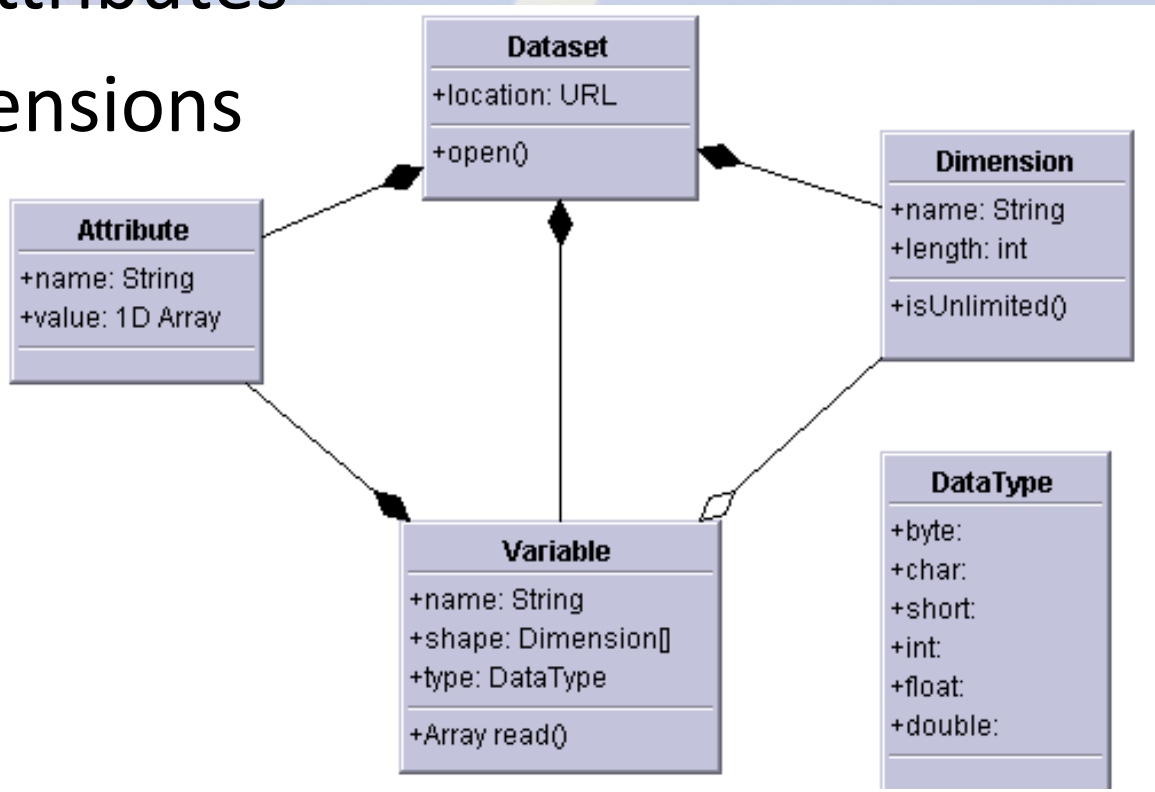
NetCDF is a…



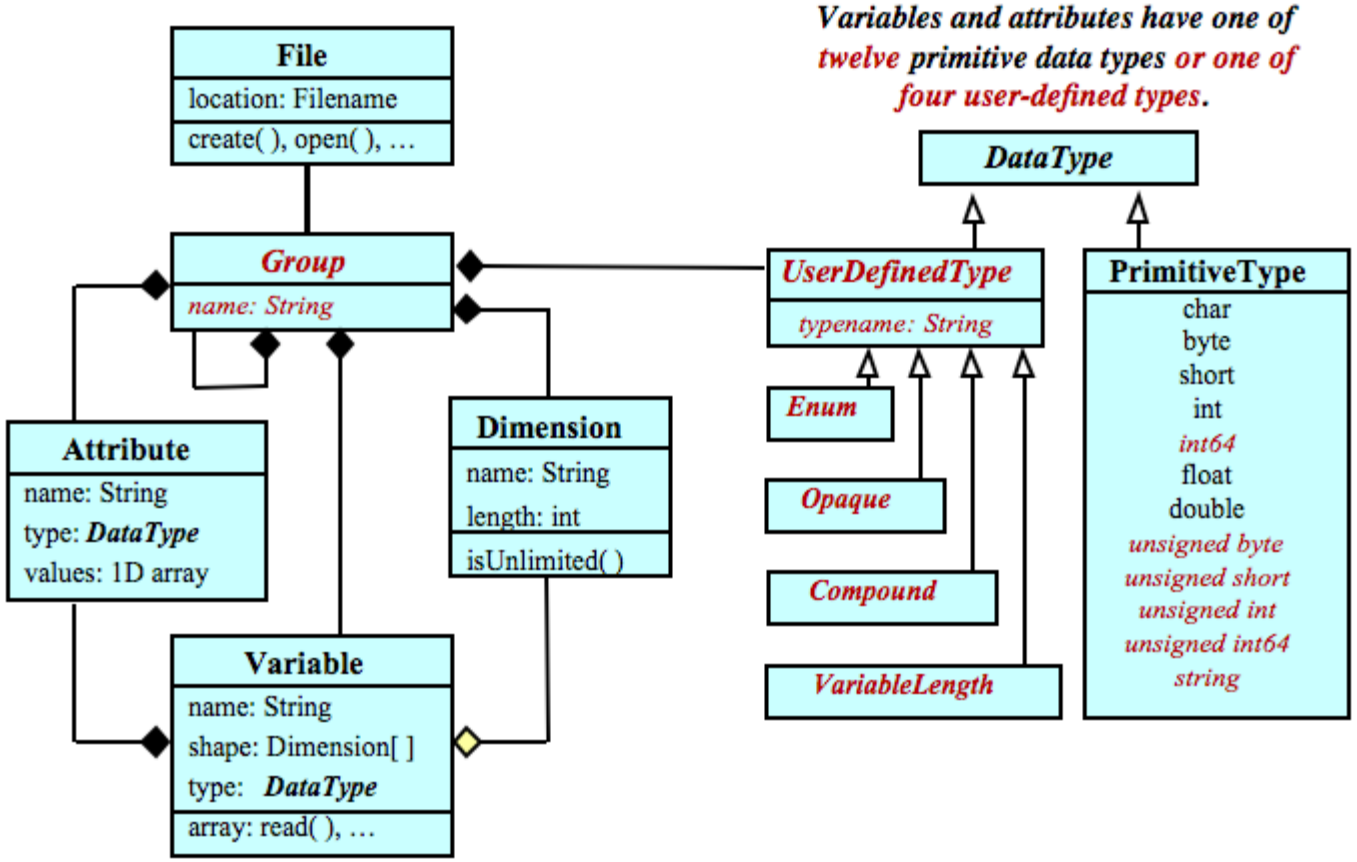
An **Abstract Data Model** describes data objects and what methods you can use on them.

NetCDF-3 data model

- Multidimensional arrays of primitive values
  - byte, char, short, int, float, double
- Key/value attributes
- Shared dimensions
- Fortran77

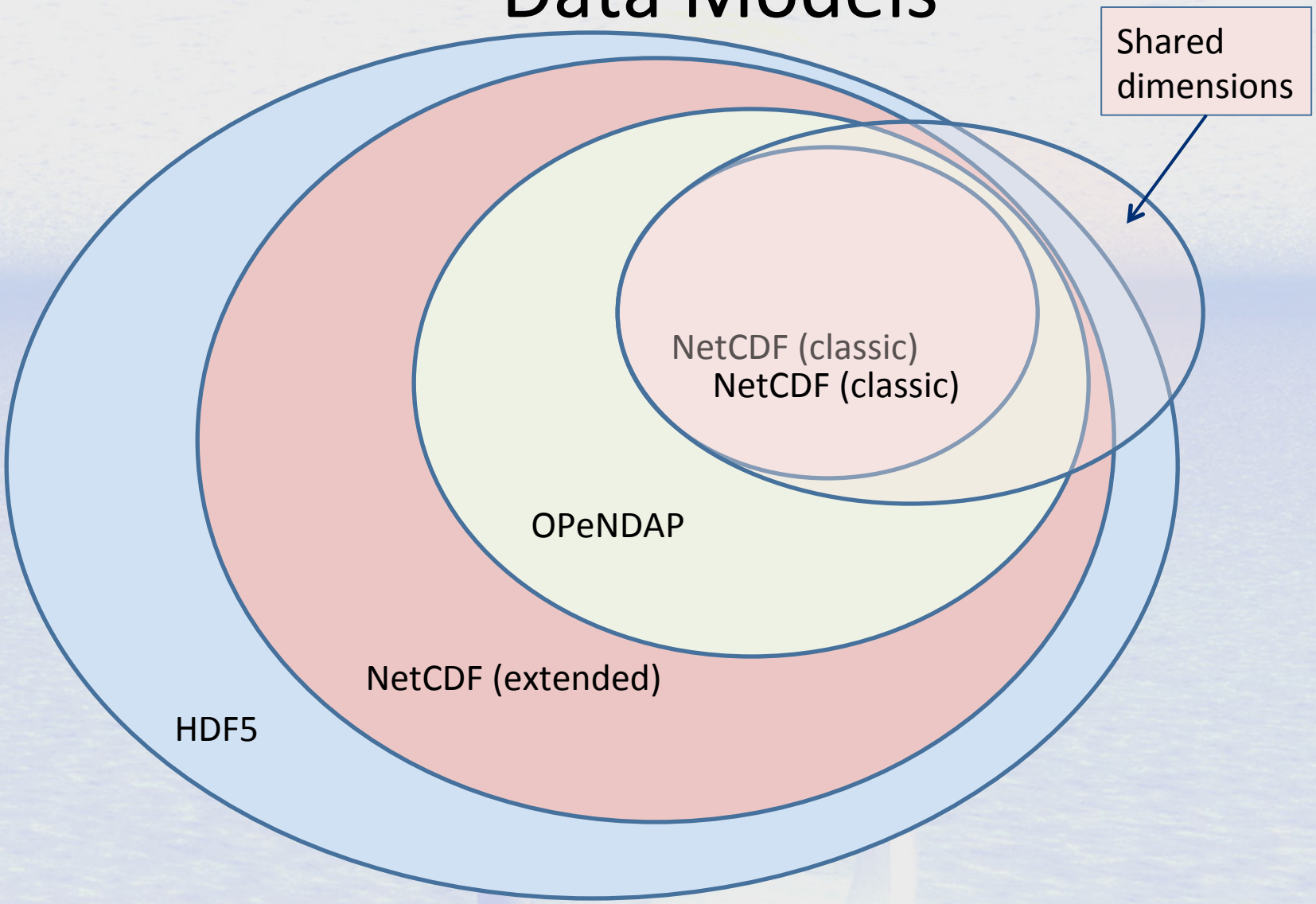


NetCDF-4 Data Model



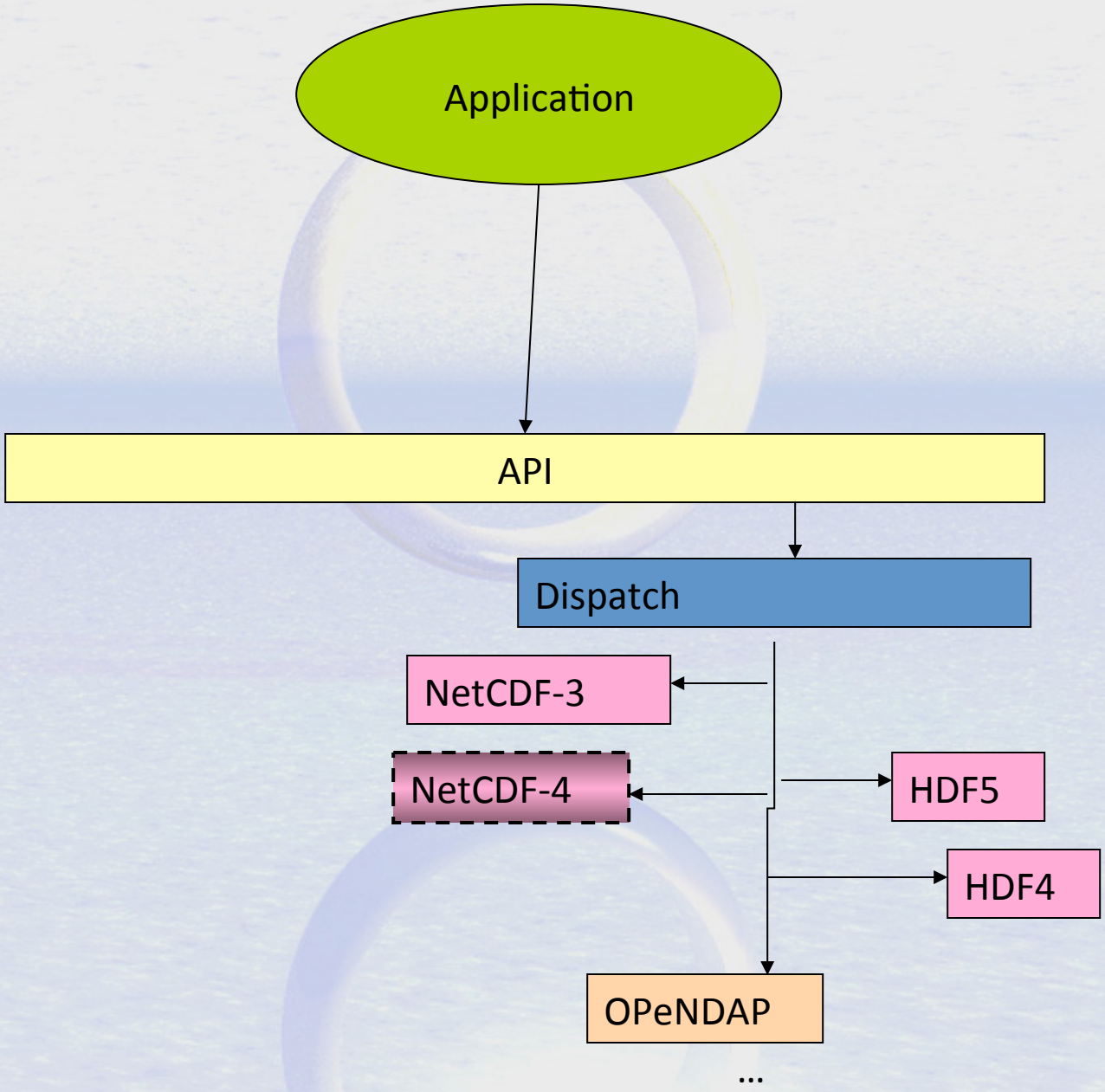
A file has a top-level unnamed group. Each group may contain one or more named subgroups, user-defined types, variables, dimensions, and attributes. Variables also have attributes. Variables may share dimensions, indicating a common grid. One or more dimensions may be of unlimited length.

NetCDF, HDF5, OPeNDAP Data Models

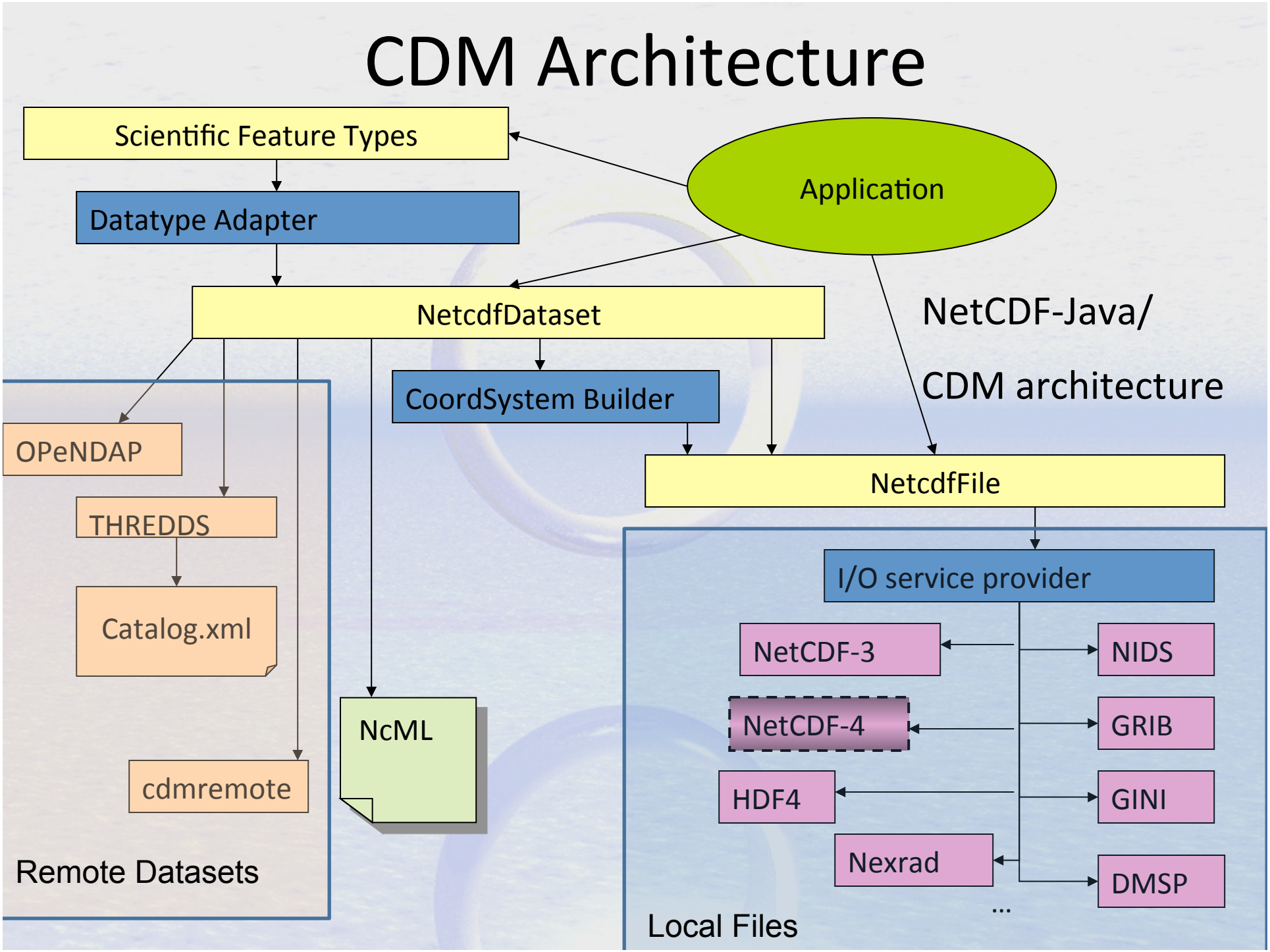


NetCDF-Java Library
(aka) Common Data Model
Status Update

C Library Architecture



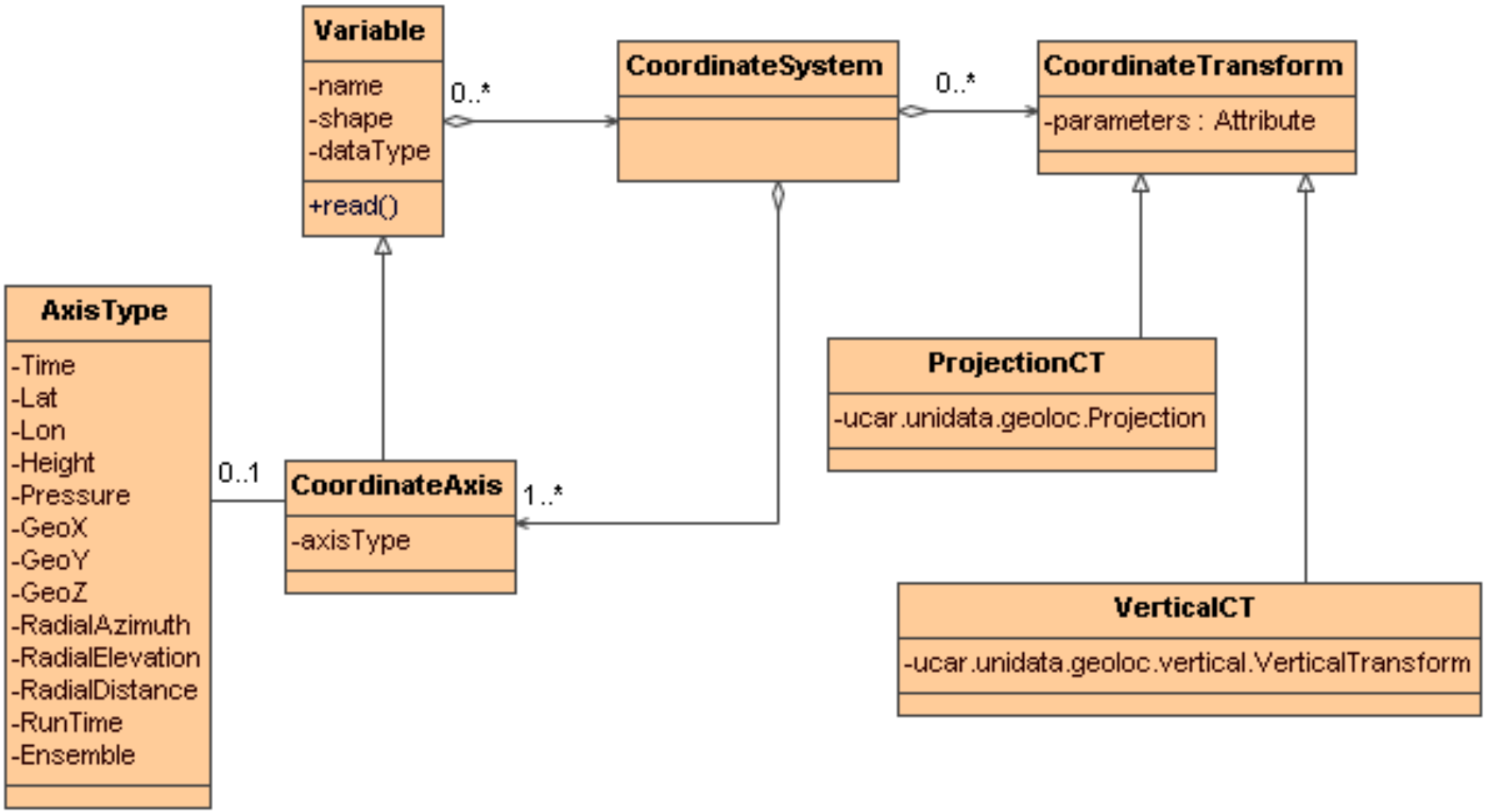
CDM Architecture



CDM file formats

Id	Description	Reference URL
BUFR	WMO Binary Universal Form	http://www.wmo.int/pages/prog/www/WMOCodes/OperationalCodes.html
CINRAD	Chinese Level-II Base Data	http://www.cinrad.com/
DMSP	Defense Meteorological Satellite Program	http://dmsp.ngdc.noaa.gov/
DORADE	DOppler RAdar Data Exchange Format	http://www.eol.ucar.edu/rsf/UserGuides/SABL/DoradeFormat/DoradeFormat.html
F-TDS	Ferret I/O Service Provider and Server-side Analysis	http://ferret.pmel.noaa.gov/LAS/documentation/the-ferret-thredds-data-server-f-tds
FYSAT	Chinese FY-2 satellite image data in AWX format	http://satellite.cma.gov.cn/
GempakGrid	GEMPAK Gridded Data	http://www.unidata.ucar.edu/software/gempak/
GempakSurface	GEMPAK Surface Obs Data	http://www.unidata.ucar.edu/software/gempak/
GINI	GOES Ingest and NOAAPORT Interface	http://weather.unisys.com/wxp/Appendices/Formats/GINI.html
GRIB1	WMO GRIB Edition 1	http://www.wmo.ch/pages/prog/www/WMOCodes/Guides/GRIB/GRIB1-Contents.html
GRIB2	WMO GRIB Edition 2	http://www.wmo.ch/pages/prog/www/WMOCodes/Guides/GRIB/GRIB2_062006.pdf
GTOPO	USGS GTOPO digital elevation model	http://edc.usgs.gov/products/elevation/gtopo30/gtopo30.html
HDF4	Hierarchical Data Format, version 4	http://www.hdfgroup.org/products/hdf4/
HDF5	Hierarchical Data Format, version 5	http://www.hdfgroup.org/HDF5/
McIDASArea	McIDAS area file	http://www.ssec.wisc.edu/mcidas/doc/misc_doc/area2.html
McIDASGrid	McIDAS grid file	http://www.ssec.wisc.edu/mcidas/doc/prog_man/2006 http://www.ssec.wisc.edu/mcidas/doc/prog_man/2006/formats-20.html#22077
netCDF	NetCDF classic format	http://www.unidata.ucar.edu/software/netcdf/index.html
netCDF-4	NetCDF-4 format on HDF-5	http://www.unidata.ucar.edu/software/netcdf/index.html
NEXRAD-2	NEXRAD Level-II Base Data	http://www.ncdc.noaa.gov/oa/radar/radarresources.html http://www.tsc.com/SETS/_3TDWR.htm
NEXRAD-3	NEXRAD Level-III Products	http://www.ncdc.noaa.gov/oa/radar/radarresources.html
NLDN	National Lightning Detection Network	http://www.vaisala.com/weather/products/aboutnldn.html
NMCon29	NMC Office Note 29	http://www.emc.ncep.noaa.gov/mmb/data_processing/on29.htm
OPeNDAP	Open-source Project for a Network Data Access Protocol	http://opendap.org/
SIGMET	SIGMET-IRIS	
UAMIV	CAMx UAM-IV formatted files	http://www.camx.com/
UniversalRadarFormat	Universal Radar Format	ftp://ftp.sigmet.com/outgoing/manuals/program/cuf.pdf

Coordinate System UML



Conventions

- CF Conventions (preferred)
  - dataVariable:coordinates = “lat lon alt time”;
- COARDS, NCAR-CSM, ATD-Radar, Zebra, GEIF, IRIDL, NUWG, AWIPS, WRF, M3IO, IFPS, ADAS/ARPS, MADIS, Epic, RAF-Nimbus, NSSL National Reflectivity Mosaic, FslWindProfiler, Modis Satellite, Avhrr Satellite, Cosmic, ….
- Write your own *CoordSysBuilder* Java class

Projections

- albers_conical_equal_area (sphere and ellipse)
- azimuthal_equidistant
- lambert_azimuthal_equal_area
- lambert_conformal_conic (sphere and ellipse)
- lambert_cylindrical_equal_area (sphere and ellipse)
- mcidas_area
- mercator
- METEOSAT 8 (ellipse)
- orthographic
- rotated_pole
- rotated_latlon_grib
- stereographic (including polar) (sphere and ellipse)
- transverse_mercator (sphere and ellipse)
- UTM (ellipse)
- vertical_perspective

Vertical Transforms (CF)

- atmosphere_sigma_coordinate
- atmosphere_hybrid_sigma_pressure_coordinate
- atmosphere_hybrid_height_coordinate
- atmosphere_ln_pressure_coordinate
- ocean_s_coordinate
- ocean_sigma_coordinate
- ocean_s_coordinate_g1, ocean_s_coordinate_g2
- existing3DField

NetCDF "Index Space" Data Access:

OPeNDAP URL:

http://motherlode.ucar.edu:8080/thredds/dodsC/NAM_CONUS_80km_20081028_1200.grib1.ascii?Precipitable_water[5][5:1:30][0:1:77]

"Coordinate Space" Data Access:

NCSS URL:

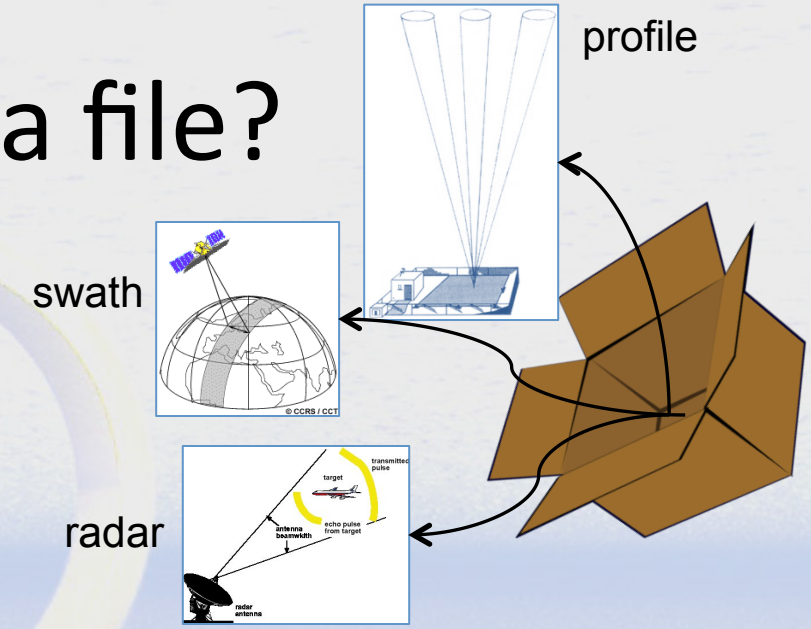
http://motherlode.ucar.edu:8080/thredds/ncss/grid/NAM_CONUS_80km_20081028_1200.grib1?**var=Precipitable_water&**time=2008-10-28T12:00:00Z&north=40&south=22&west=-110&east=-80

Scientific Feature Types

- Classification of earth science data into broad categories.
- Take advantage of the regularities that are found in the data for performance
- Scale to large, multifile collections
- Support subsetting in Space and Time

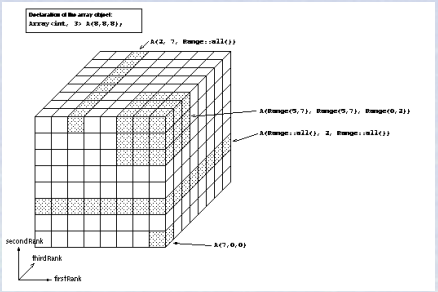
What's in a file?

1. Feature Types



2. NetCDF File

Multidimensional Arrays



3. OS File

Bag of Bytes

Gridded Data

- **Grid**: multidimensional grid, separable coordinates
- **Radial**: a connected set of *radials* using polar coordinates collected into *sweeps*
- **Swath:** a two dimensional grid, *track* and *cross-track* coordinates
- **Unstructured Grids**: finite element models, coastal modeling *(under development)*

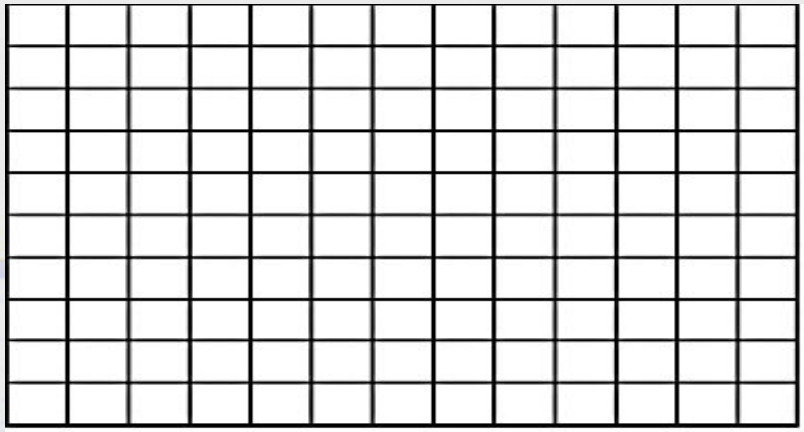
Point Data

- **point**: a single data point (having no implied coordinate relationship to other points)
- **timeSeries**: a series of data points at the same spatial location with monotonically increasing times
- **trajectory**: a series of data points along a path through space with monotonically increasing times
- **profile**: an ordered set of data points along a vertical line at a fixed horizontal position and fixed time
- **timeSeriesProfile**: a series of profile features at the same horizontal position with monotonically increasing times
- **trajectoryProfile**: a series of profile features located at points ordered along a trajectory

Discrete Sampling Convention CF 1.6

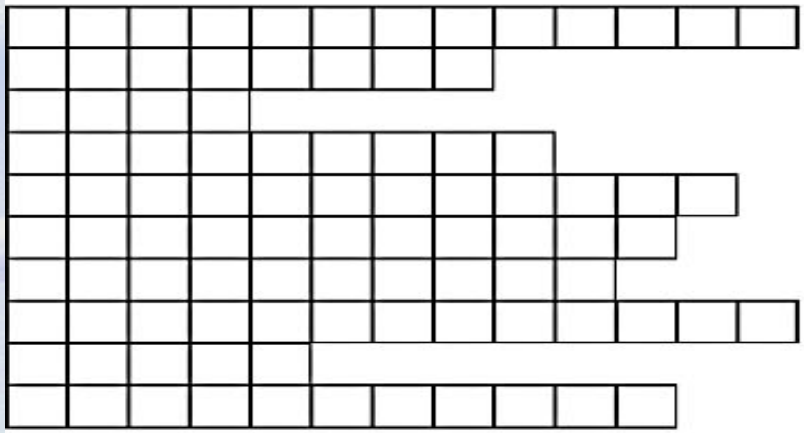
- Encoding standard for netCDF classic files
  - Challenge: represent ragged arrays efficiently
- Classifies data according to connectedness of time/ space coordinates
- Defines netCDF data structures that represent *features*
- Make it easy / efficient to
  - Store collections of features in one file
  - Read a Feature from a file
  - Subset the collection by space and time

Rectangular Array

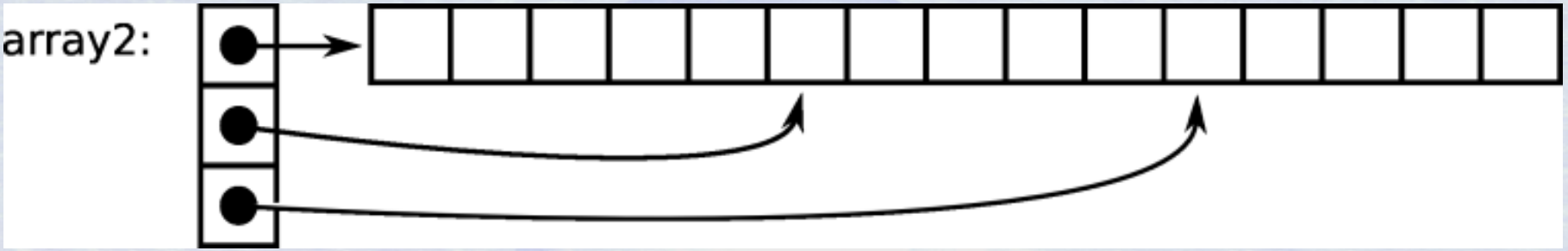


A standard two-dimensional array is a rectangle.

Ragged Array



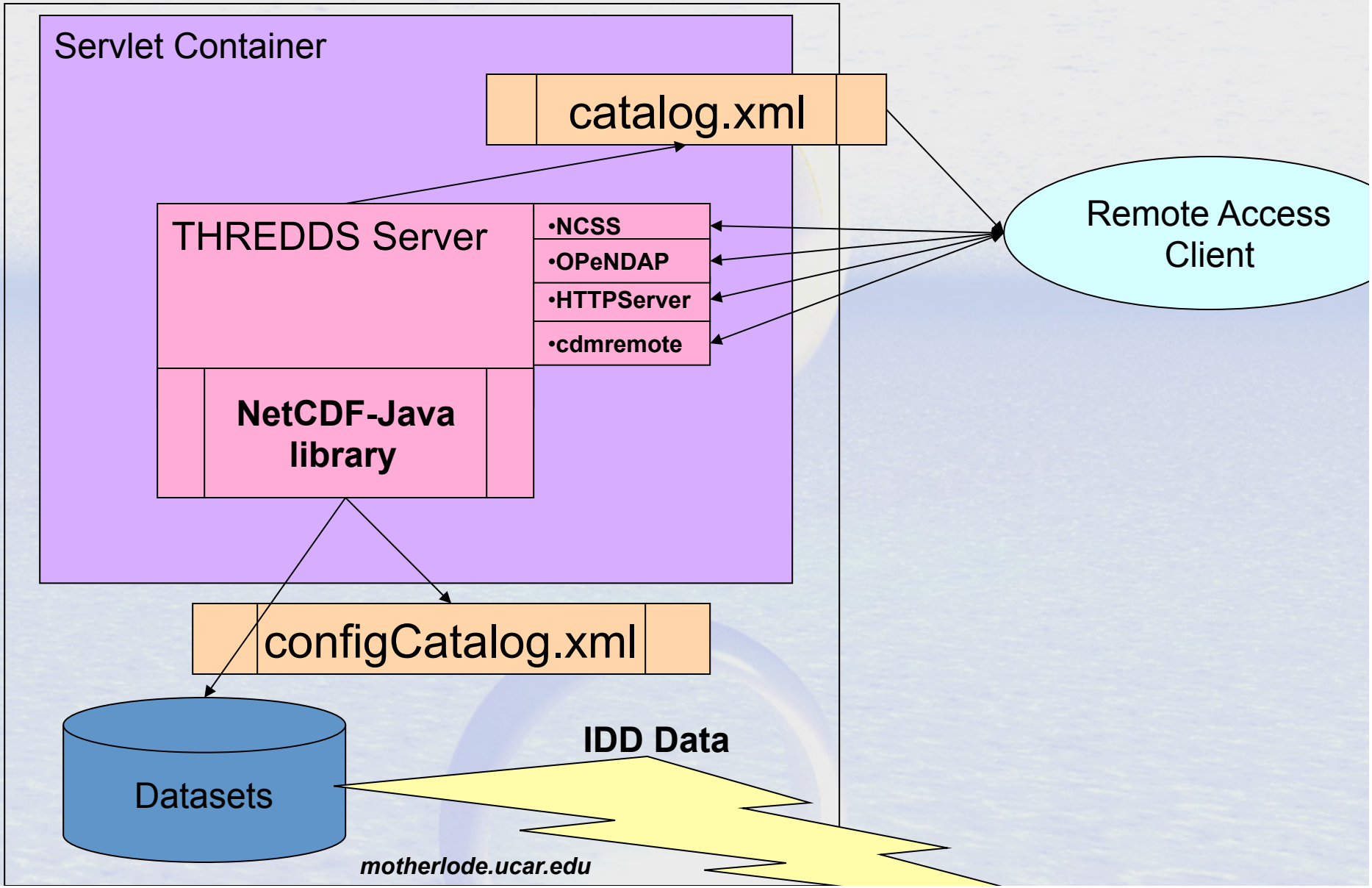
With Variant arrays, you need not waste space.



NetCDF Markup Language (NcML)

- XML representation of netCDF metadata (like ncdump -h)
- Create new netCDF files (like ncgen)
- Modify ("fix") existing datasets without rewriting them
- Create virtual datasets as aggregations of multiple existing files.
- Integrated with the TDS

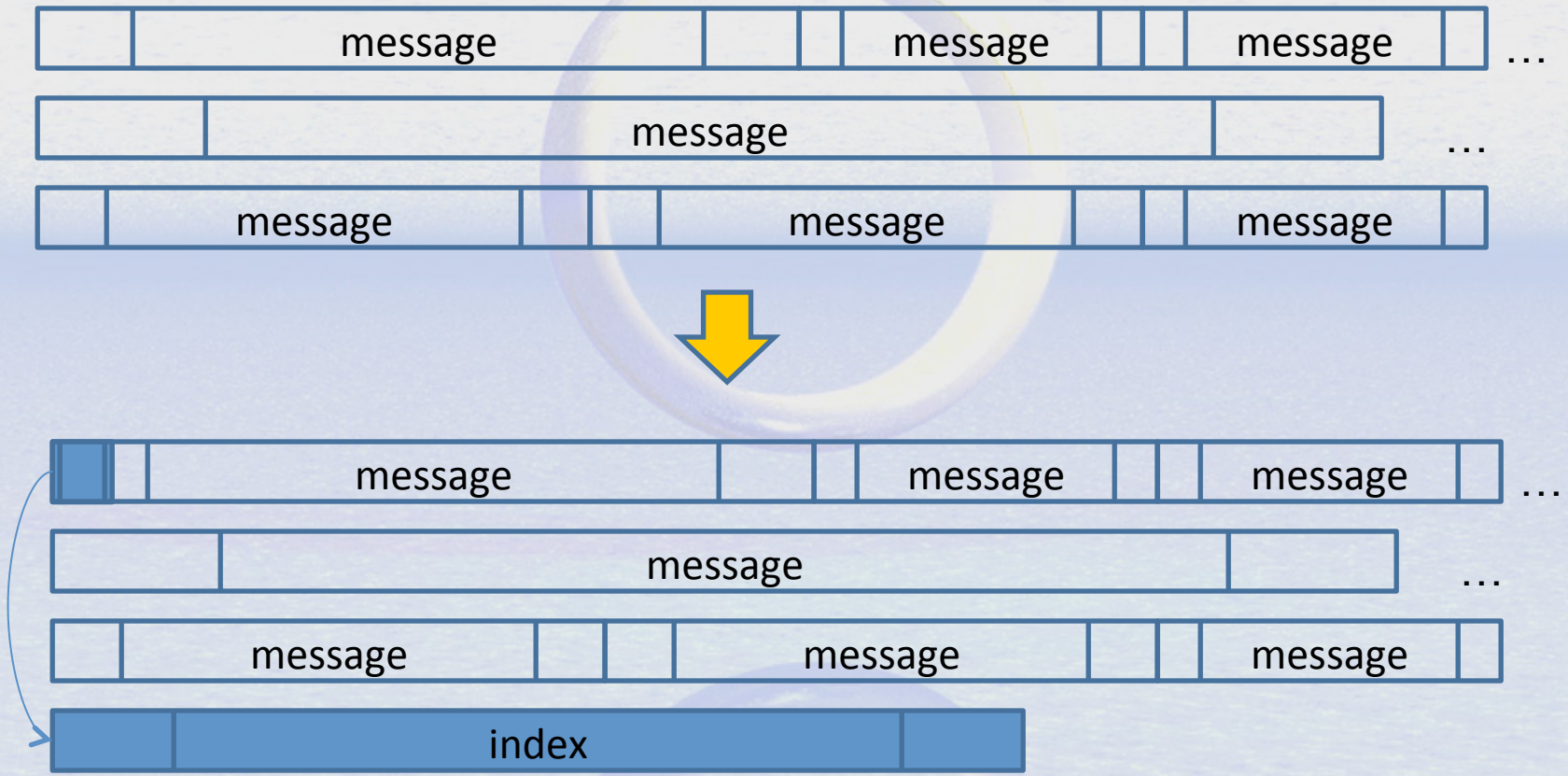
THREDDS Data Server



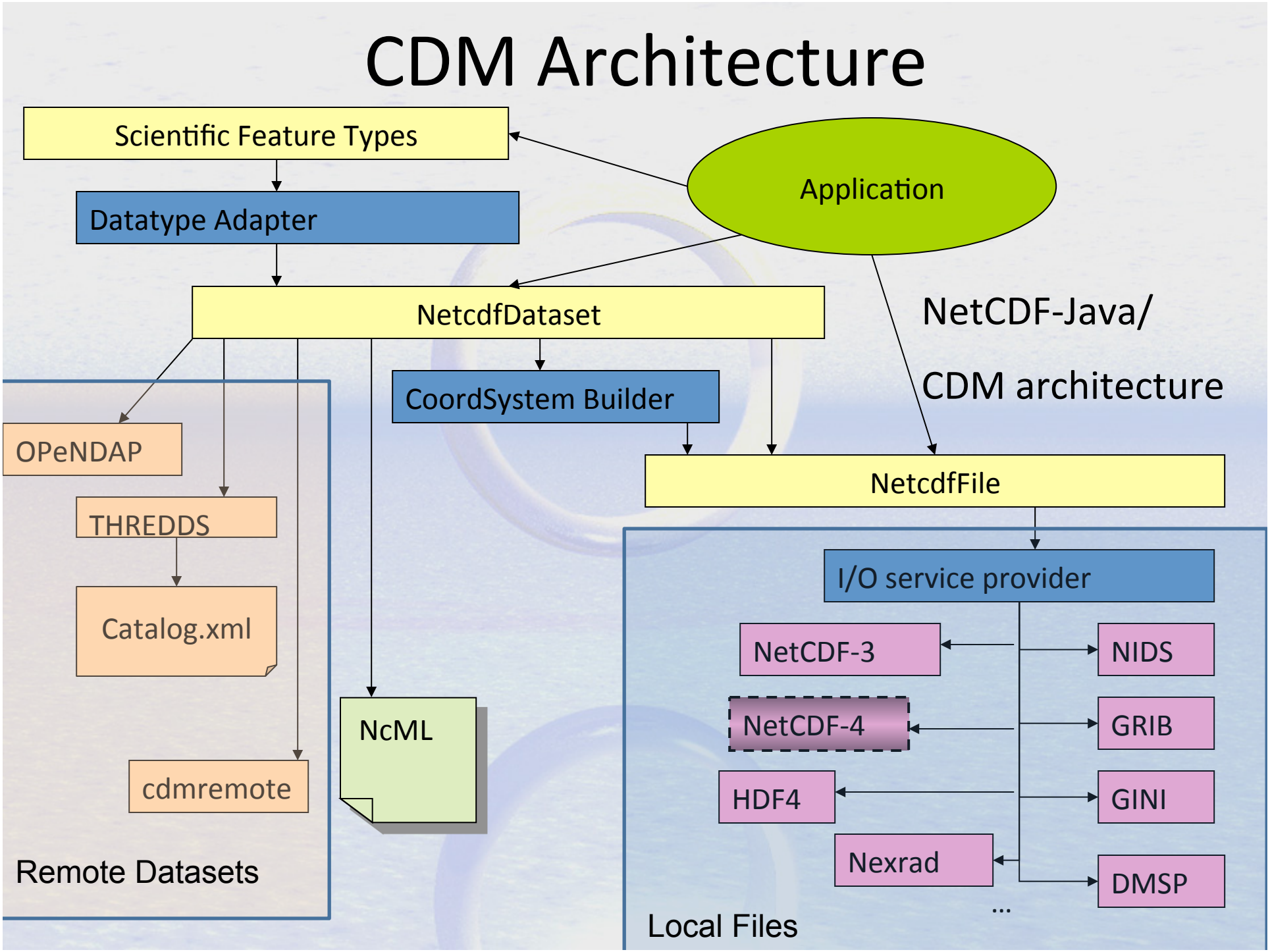
Remote Access

- OPeNDAP 2.0
  - index space access
  - Cant transport full netCDF extended data model
  - Will replace with DAP 4 next year
- cdmremote
  - Full data model, index space access
- Netcdf Subset Service
  - coordinate space access to gridded data
  - Delivers netCDF files (also csv, xml, maybe JSON)
  - Now writes netcdf-4 (alpha test), with C library / JNI
- cdmrFeature Service
  - coordinate space access to point data
  - Feature type API

ncstream serialization



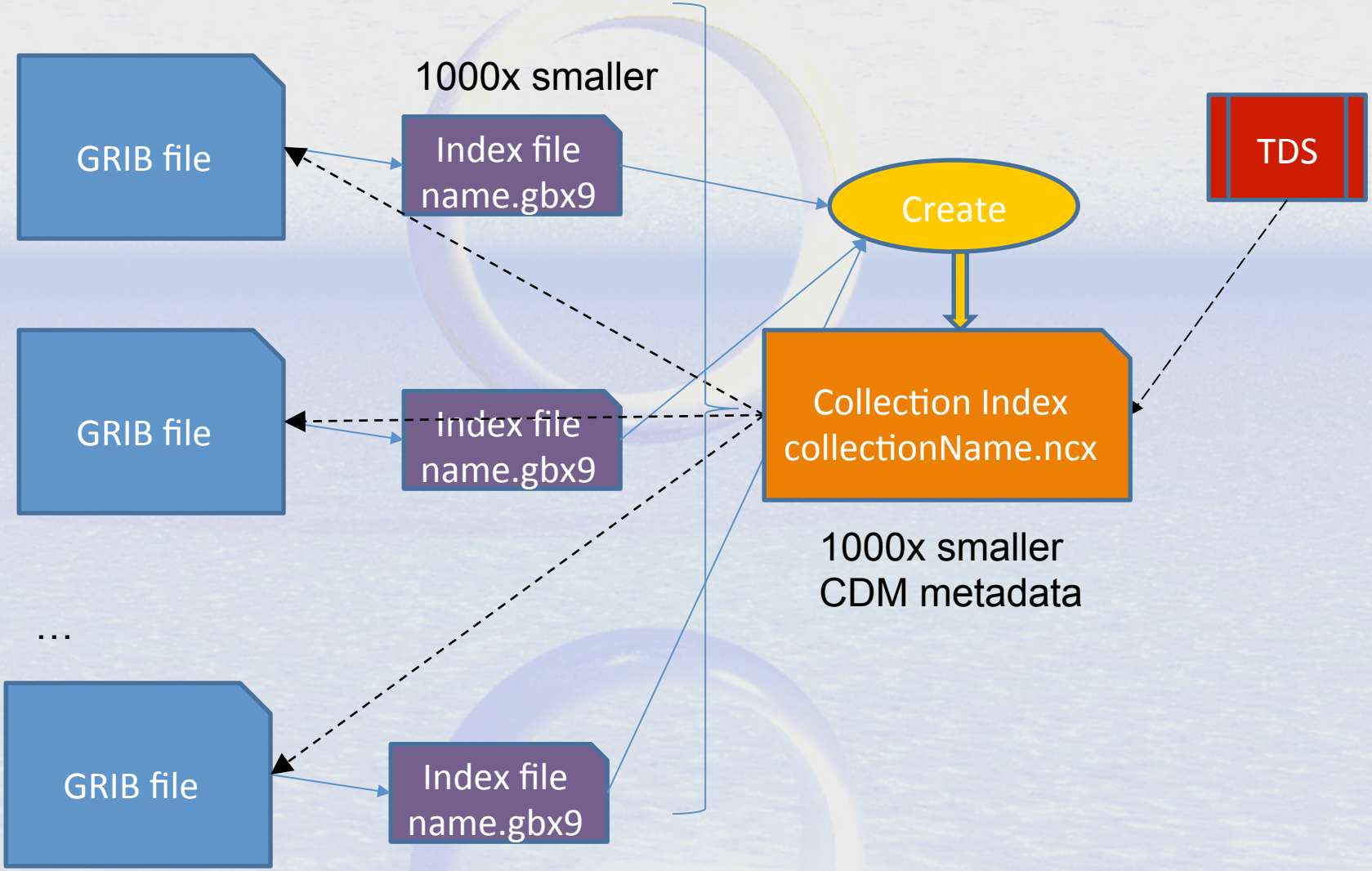
CDM Architecture



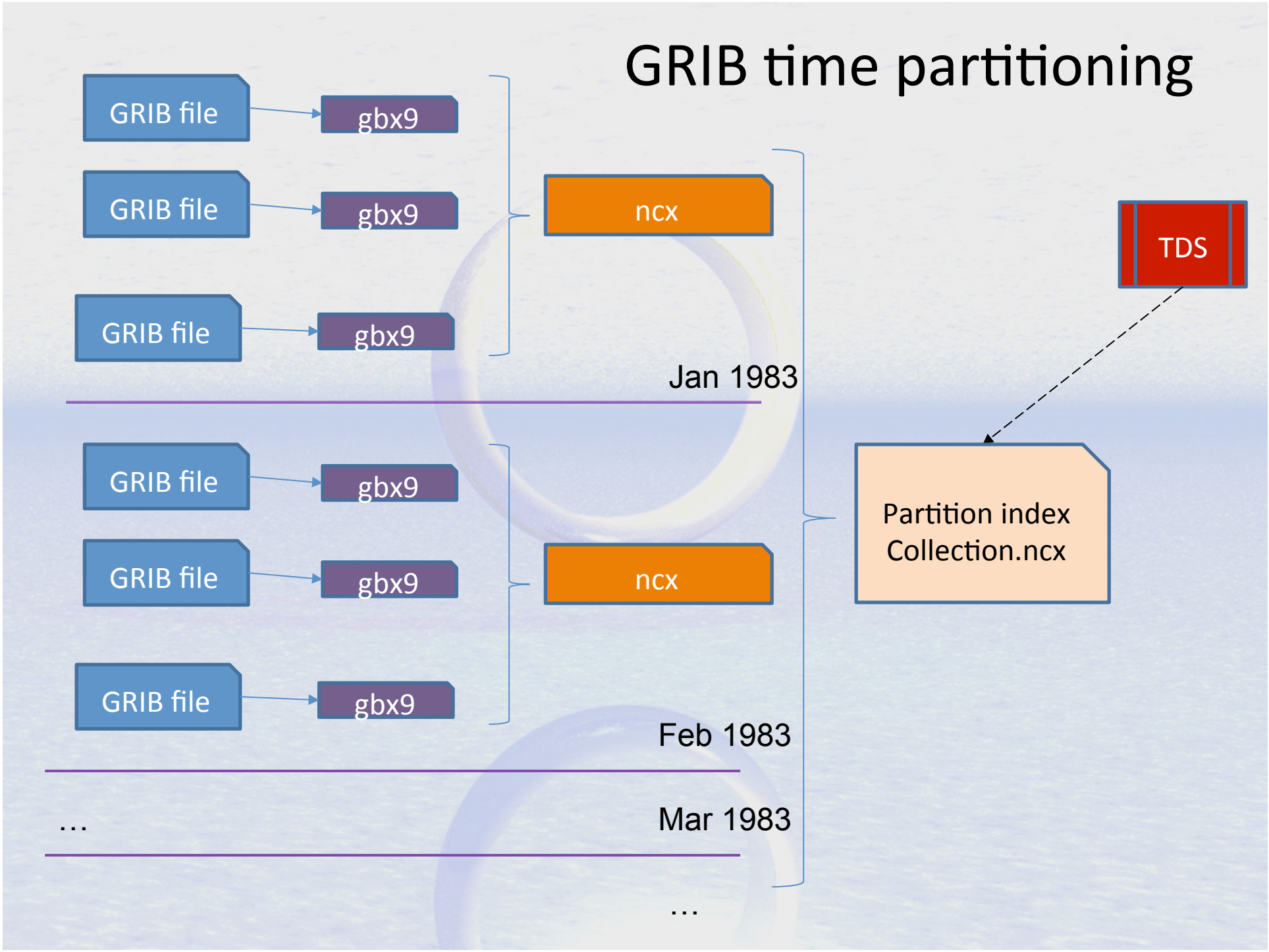
CFSR timeseries data at NCDC

- Climate Forecast System Reanalysis
- 1979 - 2009 (31 years, 372 months)
- Total 5.6 Tbytes, 56K files
- Grib2 data

GRIB collection indexing



GRIB time partitioning



What have we got ?

- Fast indexing allows you to find the subsets that you want in under a second
  - Time partitioning should scale up as long as your data is time partitioned
- No pixie dust: still have to read the data!
- GRIB2 stores compressed horizontal slices
  - decompress entire slice to get one value
- Experimenting with storing in netcdf-4
  - Chunk to get timeseries data at a single point

Big Data

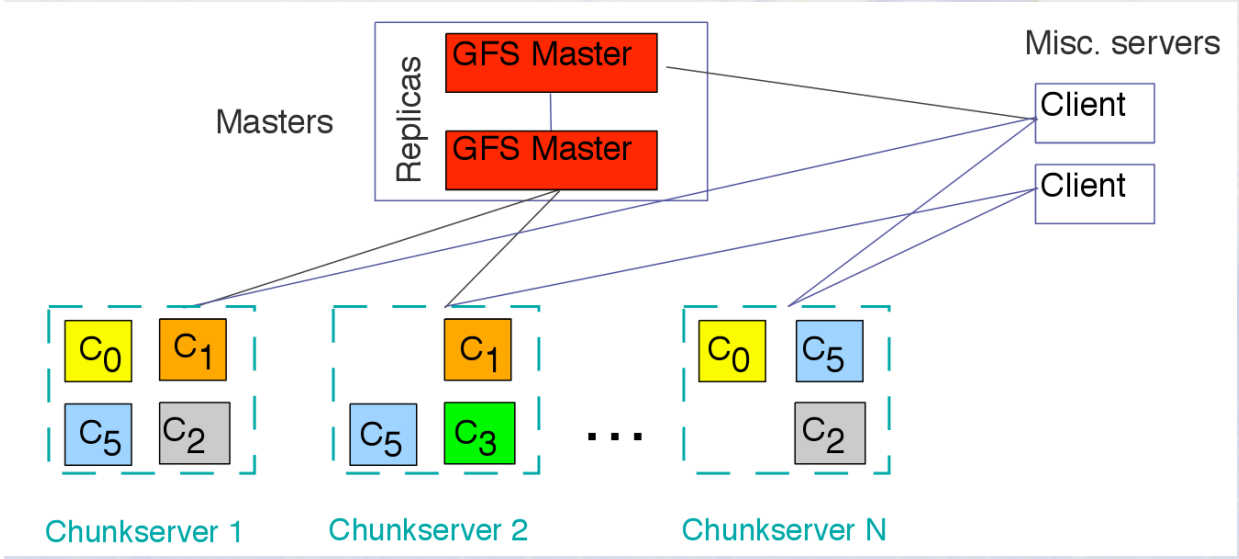
Bigger Data

- CMIP5 at IPSL Climate Modelling Centre
  - 300K netCDF files, 300 Tb.
- Sequential read of 300 Tb @ 100Mb/sec
  - 3 x 10^6 sec = 833 hours = 35 days
- How to get that down to 8 hours ?
  - Divide into 100 jobs, run in parallel

Required: Parallel I/O Systems

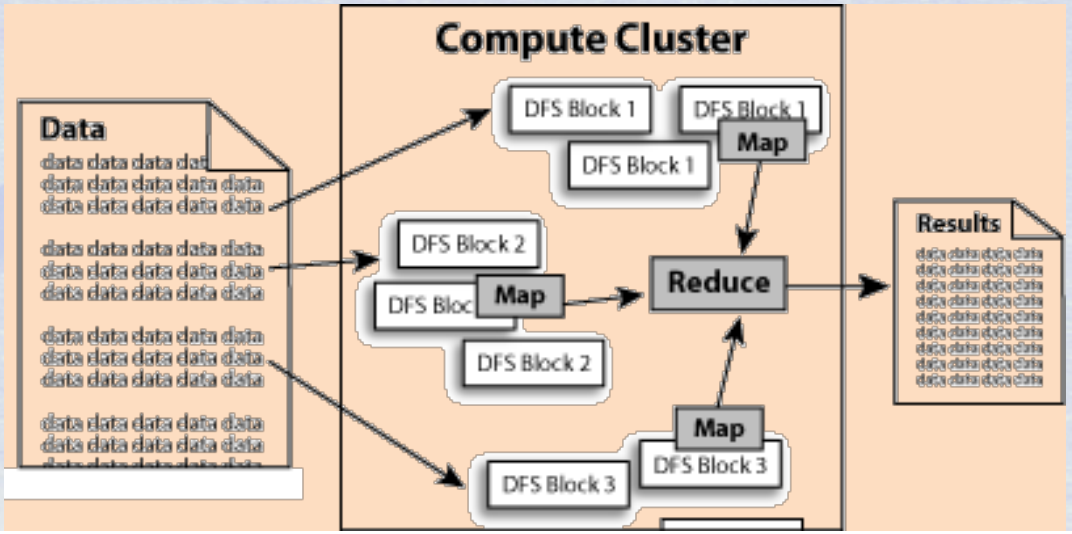
- Shared nothing, commodity disks
- Fault tolerant, replicated data (3x)
- Google File System using map/reduce
  - Hadoop is open source implementation
- Wide industry use
- Cost
  - $3000 per node TCO per year
  - $300K per year for 100 node cluster
  - Cost will continue to fall
  - Not sure if you should rent or buy

Parallel File I/O



Google File System

Hadoop



Required:
Send User programs to server

- Need a query / computation language
  - easily parallelized
  - scientists can/will use
  - Powerful enough to accomplish hard tasks
- What language?
  - Not going to retrofit existing Fortran code
    - Remember, this is post-processing, not model runs
  - Not Fortran, C, Java (too low level)
  - Some subset of Python ?

Send User programs to server

- Probably a Domain Specific Language (DSL)
  - Make it up for this specific purpose
  - But make it familiar!
  - So it could look like some subset of existing language

Existing Candidates

- SciDB just proposed ArrayQL:

> *"ArrayQL currently comprises two parts: an array algebra, meant to provide a precise semantics of operations on arrays; and a user-level language, for defining and querying arrays. The user-level language is modeled on SQL, but with extensions to support array dimensions."*

- Google Earth Engine is developing a DSL

Required: Parallelizable High Level Language

- Scientific Data Management in the Coming Decade, Jim Gray (2005)
  - Now: File-at-a-time processing in Fortran
  - Need: Set-at-a-time processing in HLQL
  - Declarative language like SQL (vs. procedural):
  - Define dataset subset to work against
  - Define computation
  - Let the system figure out how to do it

NetCDF "Index Space" Data Access:

OPeNDAP URL:

http://motherlode.ucar.edu:8080/thredds/dodsC/NAM_CONUS_80km_20081028_1200.grib1.ascii?Precipitable_water[5][5:1:30][0:1:77]

"Coordinate Space" Data Access:

NCSS URL:

http://motherlode.ucar.edu:8080/thredds/ncss/grid/NAM_CONUS_80km_20081028_1200.grib1?**var=Precipitable_water&**time=2008-10-28T12:00:00Z&north=40&south=22&west=-110&east=-80

“Coordinate Space” Data Access:

http://motherlode.ucar.edu:8080/thredds/ncss/grid/
NAM_CONUS_80km_20081028_1200.grib1?
var=Precipitable_water&
time=2008-10-28T12:00:00Z&
north=40&south=22&west=-110&east=-80

Fake SQL:

SELECT Precipitable_water
FROM NAM_CONUS_80km_20081028_1200.grib1
WHERE time=2008-10-28T12:00:00Z
AND space=[north=40,south=22,west=-110,east=-80]

More Elaborate

DATASET cfsr

FROM CFSR-HPR-TS9

WHERE *month=April* **AND** *year >=* 1990

AND *space*=[north=40,south=22,west=-110,east=-80]

AS *Grid*

SELECT precip=Precipitable_water, rh=Reletive_Humidity,
T=Temperature

FROM cfsr

CALC *DailyAvg(Correlation(* precip*,* rh*) / Avg(*T*))*

RETURN AS *Grid*

APL example

DATASET cfsr

FROM CFSR-HPR-TS9

WHERE *month=April* AND *year* >= 1990

AND *space*=[north=40,south=22,west=-110,east=-80]

AS *Grid*

CALCDEF *myCalc (X,Y,DATA) {*

X ← 3 3⍴÷⍳9 ◊ Y ← DATA[⍋DATA] *}*

SELECT precip=Precipitable_water, rh=Reletive_Humidity,
T=Temperature

FROM cfsr

CALC *myCalc(precip, rh, T)*

RETURN AS *Grid*

Summary: Big Data Post-Processing

- Need parallel I/O System
  - Shared nothing, commodity disks, replicated data
- Need parallel processing system
  - Hadoop based on GFS (Map/reduce)
- Need to send computation to the server
- Need a parallelizable query / computation language
  - Possibly declarative
  - Must be expressive and powerful
  - Probably a new “domain specific” language
  - Need to capture common queries