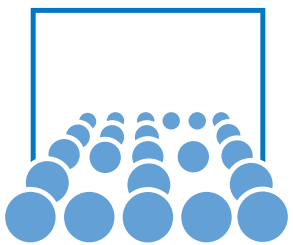
Lab: Scientific Computing – Tsunami-Simulation

Session 3: netCDF, Tsunamis

A. Breuer, M. Bader

11th November, 2014



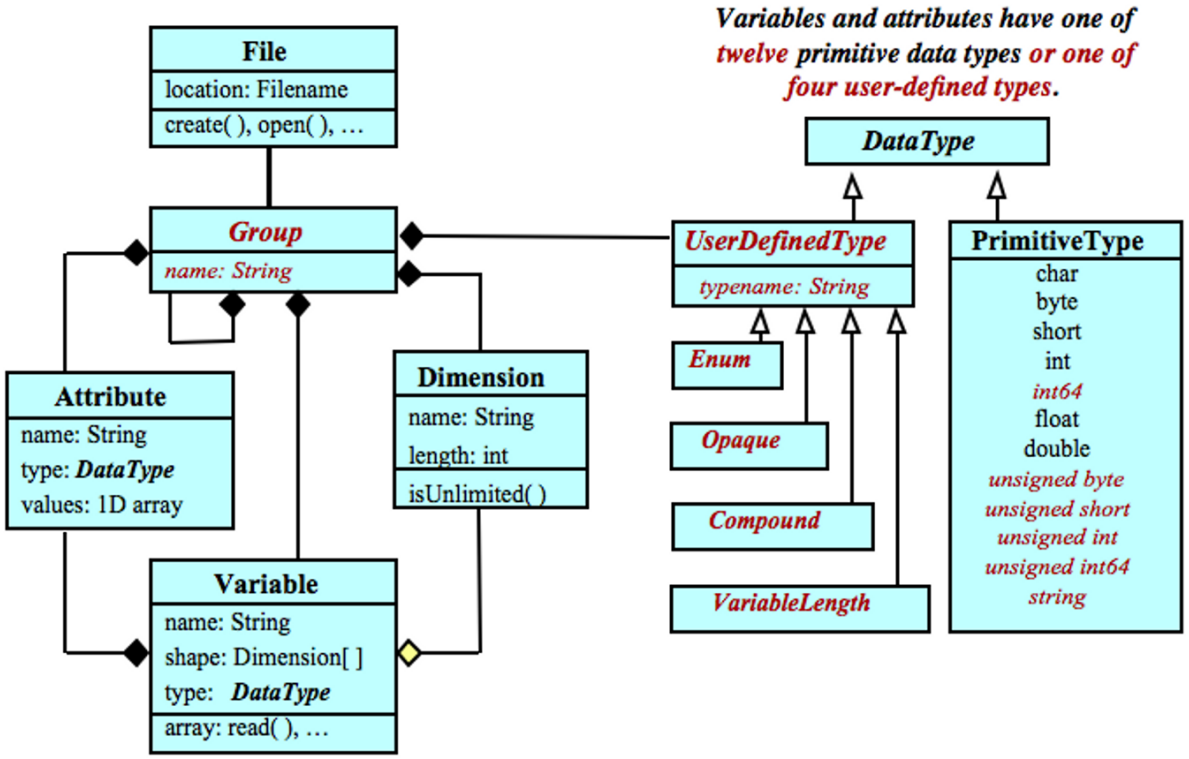
netCDF (Network Common Data Form)

- Interface to store and retrieve data in arrays
- Support for n-dimensional arrays
- Self-describing
- Portable
- Binary file format

netCDF (Network Common Data Form)

- Interface to store and retrieve data in arrays
- Support for n-dimensional arrays
- Self-describing
- Portable
- Binary file format
  $\rightarrow$ a factor of 4-5 smaller than text files

netCDF – Data Model

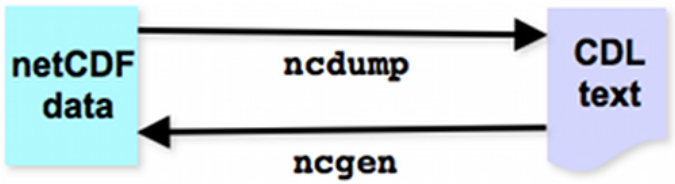


A file has a top-level unnamed group. Each group may contain one or more named subgroups, user-defined types, variables, dimensions, and attributes. Variables also have attributes. Variables may share dimensions, indicating a common grid. One or more dimensions may be of unlimited length.

Source: `http://www.unidata.ucar.edu/software/netcdf/docs/data_model.html`

CDL (network Common data form Description Language)

- Text representation of netCDF datasets
- Conversion between netCDF and CDL:



Source: `http://www.unidata.ucar.edu/software/netcdf/workshops/2011/utilities/NcgenNcdump.html`

CDL – Example

```
netcdf foo {    // example netCDF specification in CDL
dimensions:
  lat = 10, lon = 5, time = unlimited;
variables:
  int     lat(lat), lon(lon), time(time);
  float   z(time,lat,lon), t(lat,lon);

  lat:units = "degrees_north";
  lon:units = "degrees_east";
  time:units = "seconds";
  z:valid_range = 0., 5000.;

data:
  lat   = 0, 10, 20, 30, 40, 50, 60, 70, 80, 90;
}
```

Coordinate Variables

- Variables with the same name as a dimension
- Physical coordinate of the corresponding dimension
- Example:

```
  int lat(lat), lon(lon), level(level);
  short time(time);
...
data:
  level = 1000, 850, 700, 500;
  lat = 20, 30, 40, 50, 60;
  lon = -160,-140,-118,-96,-84,-52,-45,-35,-25,-15;
  time = 12;
```

Coordinate Variables

```
  int lat(lat), lon(lon), level(level);
  short time(time);
...
data:
  level = 1000, 850, 700, 500;
  lat = 20, 30, 40, 50, 60;
  lon = -160,-140,-118,-96,-84,-52,-45,-35,-25,-15;
  time = 12;
```

In our files:

- Only monotonically increasing coordinate variables
- Coordinate variables are uniformly spaced (except time)

Hardware failure

- Assume a computer fails with the probability of 0.1% during a simulation
  $\Rightarrow$ In 1 of 1000 cases, we simply restart the simulation

Source: http://www.pcmech.com/article/hp-support-problems-cheap-hardware-or-both/

Hardware failure

- Assume a computer fails with the probability of 0.1% during a simulation
  $\Rightarrow$ In 1 of 1000 cases, we simply restart the simulation
- We want to run the simulation on 100 computers in parallel
  $\Rightarrow$ Now, the probability that one of them fails is:
  $$1 - (99.9\%)^{100} \approx 9.5\%$$
  $\Rightarrow$ In 1 of 10 cases, we have to restart the simulation

Source: http://www.pcmech.com/article/hp-support-problems-cheap-hardware-or-both/

Checkpointing

- Save the current state of the simulation after defined time intervals
  $\Rightarrow$ Checkpoints
- In case of an error, load the last available checkpoint and continue the simulation
  $\Rightarrow$ Only the time steps between the last checkpoint and the error are lost