NetCDF

A python point of view

R. David
david@unistra.fr
Direction Informatique
18/10/2011

UNIVERSITÉ DE STRASBOURG

Outline

- What is NetCDF ?
- NetCDF : python API
- NetCDF4, HDF5, pytables

What is NetCDF ?

- You probably use NetCDF much more than I do !
- This talk is about NetCDF :
  - from the python point of view
  - in its environment, how it interacts with other libraries
- Anyway, what is NetCDF ?

  (From the website) *NetCDF is a set of software libraries and self-describing, machine-**independent data formats** that support the creation, access, and sharing of **array-oriented scientific data**.*

What is NetCDF ?

- What can be stored in a NetCDF file ?
  - Variables : the data
  - Dimensions of the variables
  - Attributes : small meta-data
  - Groups : hierarchical organization of data

New in Common Data Format

- At low-level, two file formats are used :
  - the classic format : binary specific, aka NetCDF3
  - the HDF5 format : widely used library for storage of scientific data
  - The newest NetCDF API can read both files
  - NetCDF 4.x API is a software layer above HDF5

What is NetCDF ? API

- A NetCDF file contains dimensions, and data
- Dimensions are retrived via the `nc_inc_dim` call. Each dimension has a name
- Variables are queried by `nc_inq_var` et retrieved with `nc_get_var`
- In NetCDF4, groups are handled by `nc_abc_grpxyz` functions

Outline

- What is NetCDF ?
- NetCDF : python API
- NetCDF4, HDF5, pytables

NetCDF python API

- In Python, the recommended API is NetCDF4, inspired by Scientific.IO.NetCDF
  - Scientific is a package from Konrad Hinsen (CNRS) providing utilities for Scientific Computing
  - http://dirac.cnrs-orleans.fr/plone/software/scientificpython/
- Is compatible with both NetCDF file formats
- Is built on top of NetCDF4 and HDF5 library
- Data can easily be converted into numpy arrays

NetCDF python API : Workflow

- First, open the file and get a Dataset :

```
import NetCDF4
ds=netCDF4.Dataset(file1,'r')
print ds.variables # Holds the list of variables
OrderedDict([(u'votemper', <netCDF4.Variable
  object at 0x2b7d71af87d0>),...
```

- This variable is a dictionnary, where keys are strings
- Then, let's get the `votemper` variable

```
votemper=ds.variables['votemper']
```

NetCDF python API : Workflow

- The variable has a shape (4D array with only 1 element in each of the 2 dimensions)

```
print votemper.shape, type(votemper.type)
(1, 1, 511, 722),<type 'netCDF4.Variable'>
```

- Then the conversion to numpy is simple :

```
votemper=numpy.asarray(votemper)
```

- Extracting only the 2D array :

```
vt2d=votemper[0,0,:,:]
```

NetCDF python API : further on

- When reading a file, NetCDF knows about its underlying type :

```
ds=netCDF4.Dataset(file1,'r')
print ds.file_format
NETCDF3_CLASSIC
```

- When creating a file, you have to specify which underlying file format is to be used :

```
ds=netCDF4.Dataset(file1,'w',format='NETCDF4')
```

- To put Data in a file :
  - Create dimensions

```
ds.createDimension('x', None)
ds.createDimension('y', None)
```

NetCDF python API : further on

- Create a variable

```
myvar=ds.createVariable("myvar2",'f8',d
  imensions=('x','y'))
```

- Get data from numpy

```
data=something_in_numpy_format
#slice_copy_iy
myvar[:,:]=data[:,:]
```

- Put it in the file

```
ds.sync()
ds.close()
```

Outline

- What is NetCDF ?
- NetCDF : python API
- NetCDF4, HDF5, pytables

NetCDF4 : What's up ?

- The NetCDF library can deal with files created with previous versions of the library (commitment)

```
ds=netCDF4.Dataset(file1,'r')
print ds.file_format
NETCDF3_CLASSIC
```

- When creating a file, you have to specify which underlying file format is to be used :

```
ds=netCDF4.Dataset(file1,'w',format='NETCDF4')
```

- Groups are hierarchical objects available only with the `NETCDF4` format

NetCDF4 : HDF5

- HDF5 is a library for scientific i/o working with :
  - self-descriptive datasets
  - groups
  - user-defined datatypes
- The datasets have a name which is a character string. Datasets are organized in groups
- The names of the datasets use the "/" as a separator, as in filesystems : "**/group1/mydset**"
- Large and well-organized API in C,C++, Fortran 90
- Wrapped in python by h5py and pytables

h5py

- http://code.google.com/p/h5py/
- h5py is a python API close to the C API of Hdf5
- It is built on top of numpy library
- If you know Hdf5, you know h5py

pytables

- http://www.pytables.org/moin
- python API for hdf5 not so close to the C API of Hdf5
- It is built on top of numpy library
- Datasets path names separator is a "." : group.dataset.attribute1...
- hdf5 + querying facilities (`where()` iterator) ⇒ uses efficient indexes (Optimized Partially Sorted Indexes)
- Pytables is a relational framework built above hdf5