NetCDF represented as a File System

Declan Valters, Blazej Krzeminski, Francesca Di Giuseppe, Kevin Marsh

Project solution and implementation

NetCDF as a FUSE filesystem

fuse-netcdf

- Written in Python (2.7/3.x)
- Runs in user-space – no root privelleges required
- FUSE – ‘filesystem in userspace’
- Single NetCDF file represented as a mounted directory
- NetCDF Dataset contents represented as files and subdirectories

A NetCDF filesystem overview

Usage

Single command line command to mount NetCDF file to mount-directory:

```
usage: ncfs [-h] [-v] NETCDF_FILE_PATH MOUNTDIR
```

Structure of Filesystem

mountpoint

└ — — Global Attributes -> *text files*

└ — — Variable -> *subdirectory*

└ — — Variable attributes -> *text files*

└ — — Variable data representation -> *text file representation*

Video Demonstration

Demo of Key Features

- Video – Basic Overview of Key Features
- (These videos are not present in the PDF/Slides on GitHub, but I will make them available through other means (YouTube)

Technical Overview

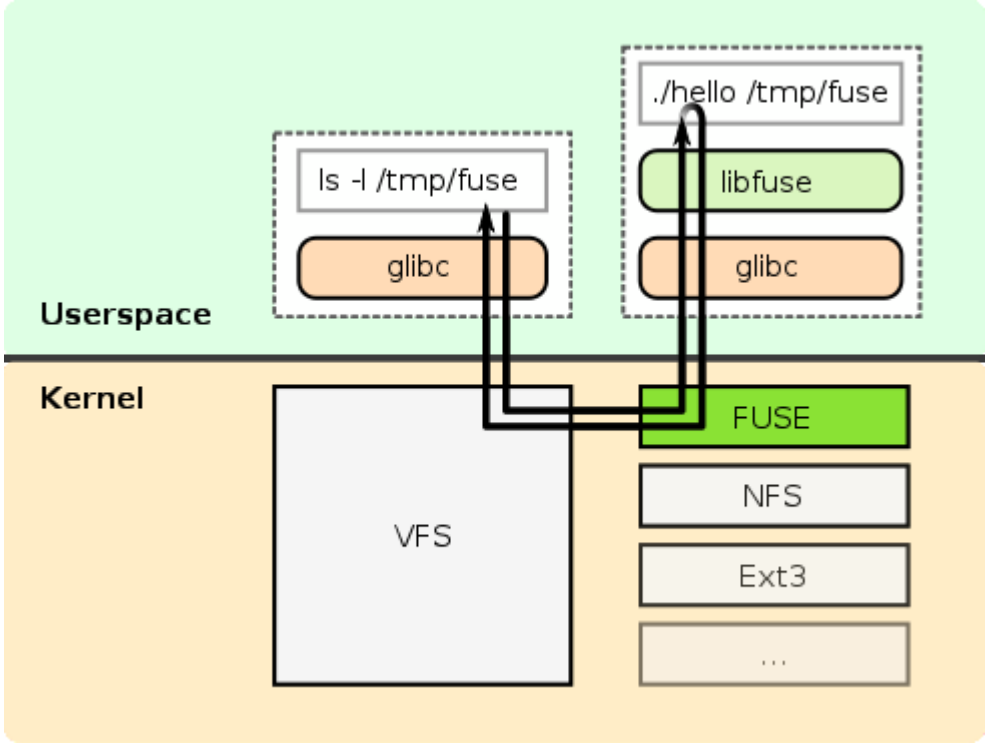
FUSE – *Filesystem in Userspace*

FUSE is a Unix system software library for allowing users to create custom filesystems:

1. without the need for root/administrator access

2. without the need for editing/knowledge of kernel code

https://github.com/libfuse/libfuse



Filesystem in User Space: This diagram was made by User:SvenTranslation. work based on a diagram at Fuse Sourceforge project page, CC BY-SA 3.0, https://commons.wikimedia.org/w/index.php?curid=3009564

Technical Overview

FUSE – *Filesystem in Userspace*

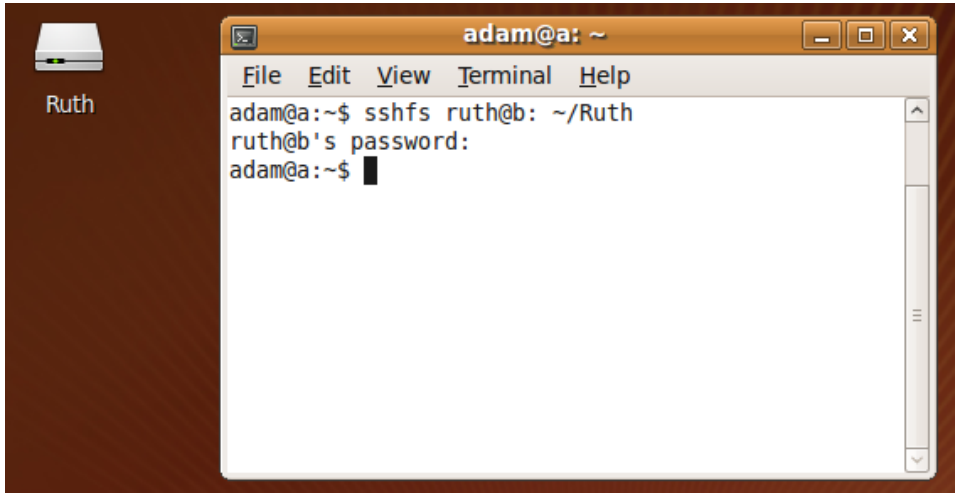
Examples:

archivemount (View tar/zip etc.)

SSHFS (View files over secure shell)

WikipediaFS (edit Wikipedia articles as if they were real files)

https://en.wikipedia.org/wiki/Filesystem_in_Userspace



SSHFS: By Behdad Esfahbod, Guilherme de S. Pastore, Havoc Pennington, Christian Persch & Mariano Suárez-Alvarez - my PC, GPL, https://commons.wikimedia.org/w/index.php?curid=7005170

Technical Overview

fusepy – high-level Python interface to the FUSE C-library

libfuse is written in C – ***fusepy*** provides a more user/developer friendly Python API.

https://github.com/fusepy/fusepy

(Documentation is a work-in-progress, some trial-and-error required!)

```
def open(self, path, flags):
    '''This method is called before the first call to the read or write methods,
       to open a file. May open a file at path, i.e. setting flags for permission or mode.
       The return value can be a file object, which will be passed as the fh parameter
       to subsequent calls to the methods read or write.
       This method does not need to be implemented; you can also open the file in the
       read/write methods.'''
return 0 # or return a file object
```

```
def read(self, path, size, offset, fh):
    '''This method is called when you read a file. I.e. by calling cat /my_filesystem/my_file
       It may be called multiple times, to read data sequentially or at random.
       But first the method getattr will be called once and then the method open.
       Returns data as a string.
       Should return *size* bytes of data associated with the file *path* at position *offset
return "Hello World" # note that getattr needs to set st_size to 11, so that cat reads 11 byt
# or better yet use string slice:
# return "Hello World!"[ offset : offset+size ]
```

Technical Overview

netCDF4-python
Python interface to the netCDF C library

A widely-used Python API for the NetCDF library.
Supports most common features for reading, editing, and writing to netCDF files.
Some features are not supported, like deleting variables from a dataset (potential workarounds exist…)

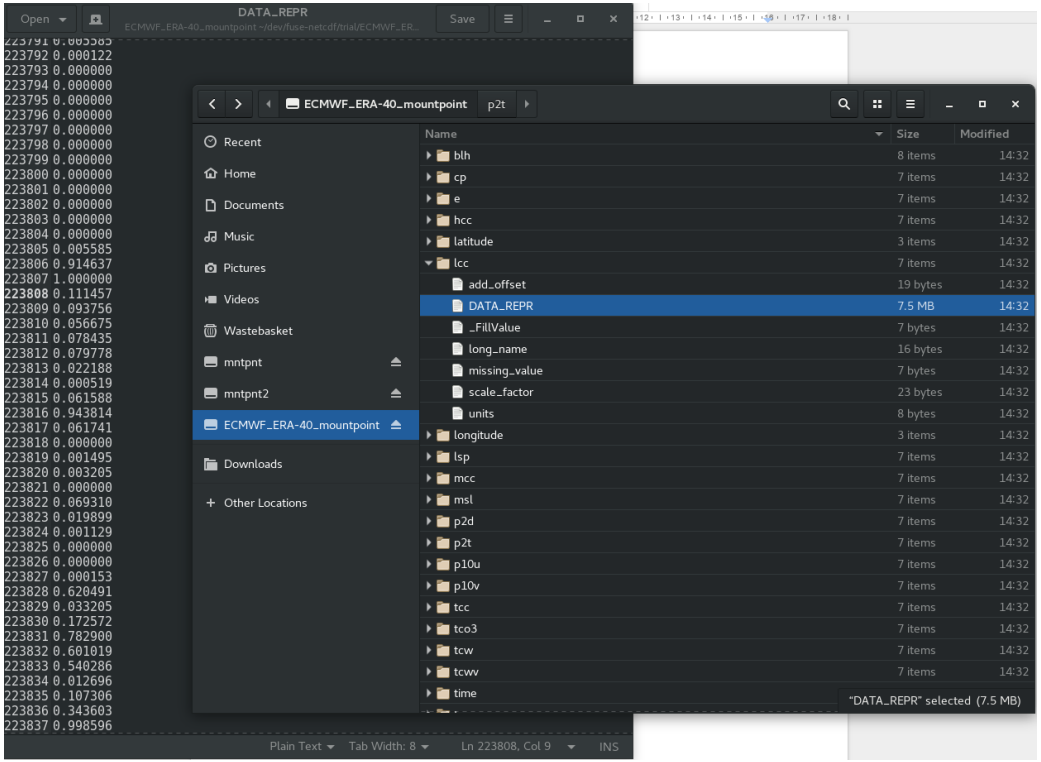
```
>>> from netCDF4 import Dataset
>>> rootgrp = Dataset("test.nc", "w",
format="NETCDF4")
>>> print rootgrp.data_model
NETCDF4
>>> rootgrp.close()
```

http://unidata.github.io/netcdf4-python/

Technical Overview

fuse-netcdf

This project, ***fuse-netcdf***, combines ***fusepy*** and ***netCDF4-python*** by mapping common Unix file system operations to the relevant NetcCDF library operations, presenting the contents of the dataset as a navigable filesystem.



Technical Overview

fuse-netcdf features:

Variables

- Variable represented as a directory
- Regular variable represented as a text file representation where appropriate. (NumPy array string representation)
- Dimension Variable's data represented as an editable text file containing a list of coordinate values
- Renaming a variable directory renames corresponding variable in NetCDF
- Copying a variable directory creates new variable in NetCDF

Attributes

- Deleting attribute text file removes corresponding attribute from NetCDF
- Renaming attribute text file renames corresponding attribute in NetCDF
- Creating new attribute text file creates new attribute in NetCDF
- Editing contents of attribute text file modifies value of attribute in NetCDF
- hanges do not result in inconsistent dataset: e.g. renaming Dimension Variable also renames the corresponding dimension and vice versa.

Technical

- Runs on Linux (Tested under SuSE, Fedora, Scientific Linux)
- Reads NetCDF3 Classic and NetCDF4 Classic files.

Challenges during development

- **FUSE and fusepy**

These packages had quite a steep-learning curve (if you are not a linux systems programmer…)
fusepy is quite sparsely documented
libfuse (the C library) is well documented, but…
Not always an exact mapping from fusepy to FUSE (libfuse) functions
Lots of experimentation required!

(This takes more time than envisaged…)

- **python-netcdf**

Limitations of python-netcdf library (deleting variables, for example)

Challenges during development

#ESoWC

Original aims that changed/postponed	Reason for change/postponement
• Mapping of multidimensional variables to subdirectories	• Requires substantial change to original code structure
• Deletion of Variables	• Not supported by python-netcdf module, but potential workaround exists: creating temporary new variable to hold changes, 'hide' old variable via FUSE. Copy to new dataset on unmount.
• Modify/define (new) Variable types	• Requires user to pass information about new variable's type though Unix filesystem command(?) – Could be done with ancillary text files.

(see: https://github.com/dvalters/fuse-netcdf/issues - we are open to contributions!)

Future development

Open to contributions!

- Extension of Variable representation/editing capabilities - Plugins
- Validity testing (e.g attribute values are correct type/valid) testing conformance to standards such as CF within the package)
- Multidimensional variable mapping to subdirectories
- Pip/Conda installation

Future development

Questions?

- Fusenetcdf is open source and open to contributions
- https://github.com/dvalters/fuse-netcdf
- Pull requests/suggestions welcomed through GitHub issue tracker