November 2017 Hackathon

Dates: November 6-17

Place: NCAR Mesa Lab, Fleischmann Board Room (https://staff.ucar.edu/browse/locations/fb)

Participants: Anna Shlyaeva, mhu, Xin Zhang, user-cf056, Jing Guo, Ricardo Todling, gvernier, Benjamin Johnson, Bryan Karpowicz, John Michalakes, Yannick Tremolet, Gael Descombes, BJ Jung

(List to be completed, I'm having trouble with the "@ user" mentions. Support says there is a bug in the wiki software, they are looking into it. YT)

Goal: Two (or more) observation operators working in the JEDI framework

Scope:

- Implement one satellite and one conventional observation operator in the JEDI framework
    - Priority will be given to clear-sky radiance (AMSU-A first) and radiosondes (T, Q and wind)
    - GPSRO, other conventional observations and all-sky radiance can be added if time and resources allow
- Observation operators should include quality control
- Bias correction is not included in the scope of this hackathon
- Interpolations to observations locations are not included in the scope of this hackathon (a by-pass might be required if interpolations are not available by November 6)

Required before Nov 6:

- Sample observation data files (with only a few observations for quick testing and with many observations)
- Interpolation routines from grid to observations locations (preferred) or saved interpolated fields from GSI
- JEDI-OOPS source code
- Environment to compile and run tests (docker)
- Access to latest GSI and CRTM source code (read-only)
- Working UFO repository for developments (where we can all write)
- GSI H(x) output for test cases (NetCDF diag files preferred)
- Background or interpolated background sample files (low and high resolution)

Pre-requirements for participants:

- Working knowledge of git:
    - clone, commit, pull, push, branch, merge
    - understanding of git-flow branching model
    - introductions are available here, here and here.
- Basic understanding of cmake and ctest (any good tutorials?)
- Understanding of Fortran 2003 polymorphism and inheritance
- Basic knowledge of issue tracking in JIRA (tutorial?)
- Information slides from Yannick:

2017-11-06-Hackathon.pdf

Tasks for this hackathon:

- Define data structures for
    - Observations locations
    - Interpolated fields (to obs locations)
    - Observation vectors
    - Simplified observation space
- For selected "plain" observation operators:
    - Determine required inputs (fields and metadata)

        - Identify outputs
        - Encapsulate inputs and outputs
        - Interface inputs and outputs with framework
        - Write tests
        - Run tests and validate code
- Add quality control
    - Identify required inputs
    - Evaluate scope for generic QC operators
    - Write tests
    - Run tests and validate code

Repositories and branches for code sprint:

To check which repository you have cloned from: git remote -v

OOPS:

https://github.com/UCAR/oops.git

git pull (if you have already cloned but not recently)

git checkout --track origin/feature/ufo

UFO:

https://github.com/UCAR/ufo.git

git checkout --track origin/develop

git flow init -d

When you are ready to start your new development:

git flow feature start mystuff

GSI (still on bitbucket):

https://username@bitbucket.org/jcsda/gsi-vlab.git

branch that was used to write out netcdf files: csd-ioda_nc4_amsu_conv

NetCDF file with AMSU-A data (and GeoVaLs):

in GSI repo in util/read_ncobs/amsua_n19_wprofiles.nc4

File: amsua_n19_wprofiles.nc4

NetCDF T conventional file:

ftp://ftp.ucar.edu/pub/mmm/xinzhang/JEDI

To compile UFO (in container):

cd /jedi/build

```
mkdir ufo
cd ufo

ecbuild -DOOPS_PATH=${BUILD}/oops -DIODA_PATH=${BUILD}/ioda \
        -DCRTM_LIBRARIES=${BUILD}/crtm-v2.2.3/libsrc/libcrtm.a -DCRTM_INCLUDE=${BUILD}/crtm-v2.2.3/libsrc \
        ${SRC}/ufo

make
```