Report from the STSM to SMHI Norrköping

- Name: Matúš Tejiščák
- Date: 9 January - 13 January 2017
- Host: Günther Haase, SMHI Norrköping, Sweden

Purpose of the mission

The pipeline producing bird profiles from raw data did not produce satisfying outputs. Hence the purpose of this STSM was:

1. Upgrade `vol2bird`, the core algorithm extracting bird profiles from raw data, with new updates and improvements from the author.
2. Update/write a script that automates processing raw radar data into bird profiles.
3. Check the processing pipeline and ensure that the results are available via the Internet.

The STSM was done with Liesbeth Verlinden (UvA Netherlands) and our host Günther Haase (SMHI Sweden). This reports covers the work done by Matúš Tejiščák.

Work done

Tuesday, 10 January 2017

- Set up development environment, read documentation.
- Go over the files found to be incorrectly merged, determine the causes.
- Implement various scripts clarifying the issues.
- Create an overview of which files were processed incorrectly and why.

Wednesday, 11 January 2017

- Implement a fast TAR repacking script to obtain archived data faster.
- Examine further issues with merging polar volumes and scans.
- Patch `polar_merger.py` to ignore comments in data source strings.
- Implement a resampling merge script for differently sampled data.

Thursday, 12 January 2017

- Upgrade `vol2bird` to the current `master` version.
- Patch `vol2bird` for correct compilation on older systems.
- Assist with upgrade of the environment of `vol2bird`.
- Assist with examination of merging pipeline on the Baltrad server.

Friday, 13 January 2017

- Implement a new merging script, based on the procedure that was verified.
- Assist with examination of merging pipeline on the Baltrad server.
- Assist with copying of data.

Post-trip

- Finish the merging script, make it run on newer systems.
- Create a Docker image to host the merging script.
- Transfer raw data files over the internet from Baltrad to Amsterdam.
- Write documentation.

Results

This section describes what has been achieved for each aim mentioned in the introduction.

1. upgrade `vol2bird`

The program has been upgraded on the Baltrad server and patched for compatibility with the older system.

2. update/write a script processing raw data into bird profiles

I created a Docker image that transforms raw data into bird profiles. This container is based on existing software provided by Baltrad and on `vol2bird`.

The Docker image is available in the Docker Hub repository at the address https://hub.docker.com/r/ziman/baltrad-merge/.

The source code is available in the GitHub repository at the address https://github.com/ziman/baltrad-merge/.

3. check the processing pipeline

- [GOOD] Data from individual radars flows in correctly into the Baltrad server.
- [GOOD] Data is backed up from the Baltrad server to external storage correctly.
- [BAD] Real-time processing on the Baltrad server is incorrect. The problem seems to be that the Baltrad server is running a strange version of the processing code, as the outputs of real-time processing differ from data processed manually using the Baltrad-provided software. Due to our limited access to the Baltrad server and the lack of time, we were unable to figure out the exact cause and the live data is therefore still processed incorrectly.

However, the Docker container can be used for correct back-processing of archived data.

- [GOOD] Live data + the incorrectly processed data is now available again via FTP for 3 days.

Doc notes

This memo aims to sum up the information that would be useful to new programmers who come to work with the Baltrad data.

Terminology

Scan (data from) a 360-degree sweep with a radar beam pointed at a certain elevation angle from the horizontal plane. This angle can be negative (e.g. for radars placed on high peaks).

Polar volume a collection of scans at various elevation angles.

Quantity a physical quantity measured in a scan, such as reflectivity, radial velocity (from Doppler shift), etc. A scan may measure one or more quantities. Likewise, a polar volume can contain data on one or more quantities.

Merging the process of combining data stored in separate files into a single output file. Usually used to merge a collection of files containing separate scans into one file containing a polar volume, or merging data of different quantities stored in separate files.

Bird profile dataset giving average density (scalar) and speed (vector) of birds for each horizontal slice of space within a 25km radius from the radar, where each slice is 200m tall.

Number of rays number of different azimuths included in a scan.

Number of range bins number of different (radial) distances, at which quantities are sampled for each ray.

HDF5 file format used for data files.

Rave, HLHDF, RSL, vol2bird libraries/software used to process the data.

Interesting quantities

More details can be found in the OPERA standard[1].

DBZH, DBZV corrected reflectivity (horizontal/vertical polarisation)
VRAD radial velocity (from Doppler shift)
RHOHV correlation between DBZH and DBZV

[1] D Michelson, R Lewandowski, M Szewczykowski, H Beekhuis: EUMETNET OPERA weather radar information model for implementation with the HDF5 file format, 2014

The pipeline

Theory

1. Raw data is obtained from individual radars every 15 minutes as a batch of one or more HDF5 files.
    - This data conforms to the OPERA standard[2].
2. Raw data is merged every 15 minutes into 1 polar volume per radar containing all scans and all quantities.
    - Some countries send 1 separate file per scan per quantity but, at the other extreme, some countries send only one file containing everything (all scans, all quantities). We need to run merging ourselves for the countries that do not provide merged data.
3. The polar volume for each radar is fed into `vol2bird`, which computes bird profiles from it.
    - This procedure needs quantities `DBZ*` and `VRAD` in the same scan to work. Scans containing only one or the other (or neither) are disregarded entirely.
    - For dual-pol radars, it needs `RHOHV`
4. Bird profiles in the HDF5 format are the desired output from the pipeline and they are available on the Baltrad FTP server for about 3 days.
5. People in Amsterdam copy the data from FTP to Beehub, where the data is available at https://beehub.nl/ENRAM/.

Reality

1. Raw data files are sent from 128 individual radars to the Baltrad server.
    - Data is expected every 15 minutes but some countries send data more frequently. In such cases, the extra files get a special name at the Baltrad server (see file naming scheme below).
    - As mentioned, some countries send each scan in a separate file, other countries send everything merged into one big file every 15 minutes.
2. On the Baltrad server, there is a directory (let us call it `realtime`) that stores the incoming data for about a day.
3. Every 15 minutes, `cron` launches processing of data in `realtime`.
    - There are cron jobs for meteorologists that seem to merge data (but they do it incorrectly / not usefully for us). These jobs are meant to be independent but they do interfere. This has not been resolved.

[2]D Michelson, R Lewandowski, M Szewczykowski, H Beekhuis: EUMETNET OPERA weather radar information model for implementation with the HDF5 file format, 2014

- There is no other merging done afterwards and whatever data is available is fed as input into `vol2bird`.
- This cron job is a Python script but it dies from `SIGABRT` frequently. The reason is probably something in the `Rave`-related C code that is imported as a Python library – this script is not just a wrapper for simpler scripts but it imports `_rave` and uses it for HDF5 I/O.
- As a result, this procedure generally outputs bird profiles for only about 6 radars out of the total of 128 radars, either because merging fails, `vol2bird` rejects the merged data, or the whole script crashes on `SIGABRT`.

4. Computed bird profiles are placed in a separate directory accessible via FTP, where it stays for about 3 days.
5. The raw data is copied to another directory accessible via FTP, where it stays for about 3 days.
6. The raw data is regularly moved from `realtime` into long-term storage in Linköping.

Issues with real-world data

- Some radars provide both `DBZ*` and `VRAD` but never (or rarely) at the same elevation angle. This data is currently unusable and `vol2bird` fails on it.
- Some radars provide `DBZ*` and `VRAD` with different numbers of rays and range bins *and* different scales. Differences in resolution and scale generally do not cancel out – this means that scans of different quantities at the same angle may cover different physical area and both resampling *and* cropping is needed. We perform neither resampling nor cropping, although a proof-of-concept script is available in a private repository.
- Sometimes one radar produces different source strings, which trips the (unpatched) `rave` code. Since the only difference is the `CMT` field (comment), there is a patched version of `rave` that ignores comments when comparing data source strings for equality: https://github.com/ziman/rave/tree/stsm An (oral) pull request has been issued, since the authoritative `rave` repository is not on GitHub.

File naming scheme

Names of raw files match the following regular expression. Files that do not match are not raw data but temporary files left behind (probably) by the merging procedure.

```
(.....)_(scan|pvol)(?:_(-?[0-9.]+))?_(\d{8}T\d{4}Z)(?:_(0x[0-9a-f]+))?(?:_(\d+))?\.h5
```

Match groups:

1. 5-letter radar code (2 letters for country code, 3 letters for radar name)
2. Is the file a scan or a polar volume?
3. Scans may include their elevation angle in the file name.
4. UTC timestamp.
5. Hex-encoded bitmap of quantities, as defined by OPERA[3].
6. If multiple batches arrive in the same 15-minute period, each extra batch (not the first one) gets a unix timestamp appended at the end of all filenames. This timestamp is multiplied by a factor of 100.

Baltrad server at SMHI

- user `baltrad`
- (probably) no root access
- python 2.6.6
- `pip` available
- command `python2` available, it is unknown whether `pip2` is available

Directory structure

- install prefixes are generally somewhere in `/opt/baltrad` but each library is usually installed into a different subdirectory (e.g. `/opt/baltrad/rave`)
- there may be several installed copies of various libraries in various prefixes; this is certainly the case for `Rave`
- FTP-accessible directories are somewhere in `/jail`

Code and repositories

Outputs of this STSM

Processing data on the Baltrad server does not work. Processing data manually does work (see report). Hence the purpose of the following Docker image is to exactly reproduce the way we processed the data manually and package it as a script that one can install and run on various platforms.

- https://github.com/ziman/baltrad-merge
  - contains `Dockerfile` and wrapper scripts for running the containers
  - `src/` contains Baltrad-provided merging scripts
  - relies on merging functionality included in `rave`, with a small patch added in https://github.com/ziman/rave/commit/053d0157a6c3929256f3178ba97b4516bdf36163
- https://hub.docker.com/r/ziman/baltrad-merge/

[3] D Michelson, R Lewandowski, M Szewczykowski, H Beekhuis: EUMETNET OPERA weather radar information model for implementation with the HDF5 file format, 2014

  - Docker image, based on `adokter/vol2bird`, but:
    * uses a patched `rave`
    * uses an up-to-date build of `vol2bird`
  - Built from https://github.com/ziman/baltrad-merge

Based on

- https://github.com/adokter/vol2bird
  - contains `docker/Dockerfile` and also a list of dependencies in `README.md`
- https://hub.docker.com/r/adokter/vol2bird/
  - built from https://github.com/adokter/vol2bird
  - as of now, out of date, lagging behind the git `master`

Other observations

- The merging code in the current implementation performs the following:
  1. first merge different quantities together, either scans or polar volumes
  2. then, if necessary, merge (multi-quantity) scans into (multi-quantity) polar volumes