Hybrid Cloud Data Storage

Ken Tanaka, D. Neufeld
Ken.Tanaka@noaa.gov • https://orcid.org/0000-0001-9290-9761 • National Centers for Environmental Information/NOAA, CIRES/University of Colorado

Goal

Transfer data to the cloud with minimal disruption to an existing local data flow.

Additional Objectives

- Ease of installation
- Familiar language
- Efficient operation
- Configurable data filtering
- Notifications to downstream processing such as a Metadata Catalog

Specifics

For our use case, the data are Crowdsourced Bathymetry (CSB), which we process from a hierarchy of directories and files with a Python program.

```
UNIQUE_ID,FILE_UUID,LON,LAT,DEPTH,
ROSEP-2feada92-09d5-4ba8-a34e-742a645c36e4,7a...b7,-90.804212,34.278100,
    TIME,PLATFORM_NAME,PROVIDER
    27.10,2020-04-04T15:35:30.000Z,MELVIN R.TODD,Rose Point
```

The CSB data are collected under the International Hydrographic Organization project, with some countries withholding permission to share bathymetry collected within their territorial waters. For this reason, we use the geopandas library to apply a **geospatial filter** to the data we upload. A database of the spatial restrictions is consulted to supply geometric water boundaries, skipping files that cross these territories.

A manifest of files already processed is kept, allowing us to **identify new files** in the filesystem directories. The manifest is actually a series of files, indexed by a set of directories mimicking the data directory layout, and this proves adequate for performant and simple upload tracking.

We create temporary files in a **standardized data format**, just a CSV file with one header line at the start. Although the format is a bit redundant, it lends itself to some later processing steps such as using Athena to query the data with an SQL interface.

Cloud processing, for example, metadata collection, can utilize Amazon Web Services messaging, Simple Notification Service (SNS) and an SQS topic for OneStop, which then catalogs metadata.

Dependencies

- Python 3.8 - Chosen as a language that is easy to work with and generally well liked by scientists
- Boto3 - Provides access to Amazon Web Services for Python
- Geopandas - Facilitates working with geospatial data
  - Shapely - Provides spatial geometry processing (GEOS)
  - fiona - Geometry data transport and format conversion (GDAL)
  - rtree - Improves the performance of spatial indexes
    - libspatialindex - Native C++ library supporting rtree

Data Flow