Python Parsers

This page should apply to all geodashboard projects.

Parsers built for GLTG

Repositories

The legacy repository for GLTG parsers is located in https://opensource.ncsa.illinois.edu/bitbucket/projects/GEOD/repos/gltg-parsers-py/browse. Some of the parser sources have been updated recently and some of them have not been touched for years.

These should be migrated to https://opensource.ncsa.illinois.edu/bitbucket/projects/GEOD/repos/pygeotemporal-parsers/browse on update

Legacy Data Files

https://uofi.app.box.com/folder/131386931931

Overview of GLTG parsers

- nebula instance: gltg-parsers2.ncsa.illinois.edu
- user = parsers (no password)
- root directory = /home/parsers
- directory structure
    - 4 directories for 4 systems
        - 3 parsers for 3 sources for each system
- run parsers
    - each system has a shell script that runs all three sources sequentially
    - for each source all data is parsed first, then a subprocess runs the binning with a wait on subprocess until finished. When done the next source parser starts. No timeout between source parsers.
- cronjobs
    - get greon data from gltg.ncsa.illinois.edu:/var/opt/CampbellSci/loggernet_ordered
        - runs in as marcuss user (this can be changed but for now it is due to permissions on loggernet on gltg)
        - /home/marcuss/get_greon_data.sh
            - uses rsync to pull data to /home/marcuss/data/greon/ then copies to/home/parsers/greon-data/
    - parsers
        - 4 lines for 4 systems

General processes that take significant time

- updating sensor statistics - required at end of parsing to update start and end times https://opensource.ncsa.illinois.edu/bitbucket/projects/GEOD/repos/pygeotemporal/browse/pygeotemporal/sensors.py#239
    - maybe it does more but not sure
    - maybe the query can be simplified

System Parsing Times

Here the each source parses the data then waits for the subprocess that bins the data until finished. When the binning finishes, the the next source parser starts. No timeout between parsers.

- gltg-dev
    - resources
        - nebula, proxy 2cpu 4ram
        - nebula, postgres (4 CPU, 8G RAM)
    - times by source
        - greon .5h
        - iwqis 1h
        - usgs
- gltg
    - resources
        - sd stack, proxy 4cpu 6ram
        - nebula, postgres (4 CPU, 8G RAM)
    - times by source
        - greon 6m
        - iwqis 40m
        - usgs 3h 10m
- ilnlrs
    - setup
        - 10m sleep between greon and iwqis, 20m sleep between iwqis and usgs
        - binning
            - 2 workers (not 4)
            - USGS uses 60s sleep after a bin finishes
    - times

        - greon 25m (can be improved
        - iwqis 40m
        - usgs 54m

GLM Parsing Time

GLM Zooplankton/Phytoplankton ingestion timing for Production server (141.142.211.239):

- Update Statistics: About an hour for all sensors. (api/sensors/update/).
- Binning by season
    - Ran the following endpoint for all 3 parameters simultaneously:
        - /api/cache/season/parameter-name
    - Took about 9 hours for all to complete.
- Number of Datapoints:
    - Zooplankton-biomass: 3004
    - Zooplankton-biovolume: 3004
    - Phytoplankton-biovolume: 1930