Environment and organisms - Task #416

Scope out workflow for calculating monthly LST climatologies

05/17/2012 03:05 PM - Jim Regetz

Status:	In Progress	**Start date:**	05/15/2012
Priority:	Normal	**Due date:**	
Assignee:	Jim Regetz	**% Done:**	70%
Category:	Climate	**Estimated time:**	20.00 hours
Target version:			
Activity type:			

Description

Assess feasibility of running LST climatologies ourselves by implementing a basic scripted workflow and estimating overall time/storage requirements.

Key considerations:

- Initial focus on tiles for selected test regions, but with eventual expectation of running globally
- Initial focus on monthly, but ideally with flexibility to use arbitrary date intervals
- Include ability to incorporate information from QC layers
- Use daily (11A1) as input if practical, but substitute 8-day (11A2) if deemed necessary or sufficient

See related comments under issue #375.

Related issues:

Related to Task #375: Assemble monthly mean MODIS LST values for the complete...	**New**	**03/07/2012**

History

#1 - 05/17/2012 11:38 PM - Jim Regetz

- % Done changed from 0 to 70

I've now written some Python+GRASS code to do a good bit of what I think we want. For a user-specified *tile*, *year*, and either *month* or arbitrary *start and end days*, it provides functions to automate the following:

- Download the corresponding 11A1 tiles from LP DAAC ftp server
- ...or identify the corresponding tiles stored in a local directory
- Load the daily LSTs and filter out undesirable values based on QC flags (in GRASS)
- Calculate new rasters for (1) the mean and (2) the number of daily values contributing to that mean (in GRASS)

As currently coded, only high quality LST values are retained (i.e., if the first QC bit pair is 00). It would be easy to modify the code to inspect other QC components too, though it would also add to the run time. I listed this and several other notable TODOs in the script comments. The code is still halfway between a procedural script and a reusable module. It's currently on a new jr/lst branch of the repo (source:climate/extra/aggregate-daily-lst.py@01b3830e), but I'll merge into master and also move it from extra/ to lib/ if appropriate.

Although there are likely still some steps to add to the processing, here is an initial assessment of run times and storage needs:

Time to download data (v005)

- Downloading 31 HDFs for Jan 2005 for tile h21v09 (East Africa) took 118s.
- Downloading 31 HDFs for Jan 2005 for tile h09v04 (Oregon) took 111s.
- **If these are typical, it suggests ~4.5 hours to download a full 12-year set of daily LSTs for a single tile.**

Time to run calculations

- Generating QA-adjusted means and counts for Jan 2005 for tile h21v09 (East Africa) took 48s.
- Generating QA-adjusted means and counts for Jan 2005 for tile h09v04 (Oregon) took 53s.
- Generating QA-adjusted means and counts for all 12 months of 2005 for tile h09v04 (Oregon) took 800s.
- **If these are typical, it suggests ~2-3 hours to generate monthly climatologies for a full 12-year set of daily LSTs for a single tile.**

Storage

- The 31 daily 11A1 HDFs for Jan 2005 for h21v09 (East Africa) total 114MB.
- Consistent with the above, our ~10 years of previously downloaded daily HDFs for h09v04 (Oregon) total 14GB.
- **If these are typical, it suggests ~16GB for a full 12-year set of daily LSTs for a single tile.**

Additional notes

- We could process tiles and/or months in parallel across multiple cores (if using isolated, temporary GRASS data locations).
- Adding more calculations (e.g., standard deviation) and/or aggregation over additional variables (e.g., night LST) will obviously increase the processing time.
- Downloading the daily HDFs is clearly a bottleneck -- worth checking some alternative approaches.
- Claiming "typical" is probably a stretch: Download time, processing time, and storage requirements can all vary by tile and even over dates within a tile, largely correlated with the number of non-null cell values in the dataset. My guess is that extrapolating the above numbers across all 317 global tiles would overestimate things considerably, because many of those tiles have a much lower proportion of land area.
- Following up on the last point, the average file size for the tiles above is >3.5MB, whereas MODIS documentation lists the average size as only 2.1MB for the v005 product as a whole.