THE VALUE OF PERFORMANCE.

NORTHROP GRUMMAN

Generating Large-Scale Imagery from Satellite Data with Python

American Meteorological Society 94th Annual Meeting
Fourth Symposium on Modeling and Analysis Using Python

Feb. 3, 2014

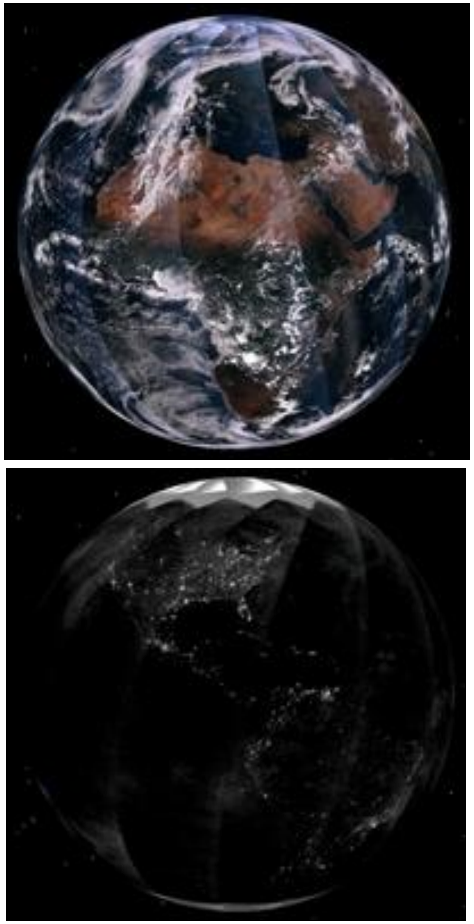
Albert Danial
al.danial@ngc.com

Overview

- Suomi NPP satellite.
- What is “large scale imagery”?
- This talk:
  1. Python and Linux challenges of producing
     - gigapixel imagery without PIL
     - high resolution animations from >10^5 frames.
  2. A technique for debugging parallel Python code

Suomi NPP

Suomi NPP artist rendering
http://www.ssec.wisc.edu/media/spotlight/npp-suomi.html



NG-generated full globe images from VIIRS bands M5/M4/M3 (above) and DNB (below)

What is Large Scale Imagery?

- Large = anything that causes size-related problems on a high-end computer (16 core, 128 GB of memory). Examples:
  - Takes too long to generate
  - Insufficient memory to generate using conventional techniques
  - Can’t view results at full-scale

- For today’s presentation I’ll narrow “large” down to
  - PNG files with >1.4 billion pixels (26000 x 52000)
  - Animation made from >86,000 HD frames (1900 x 1100)

- Why make large images?
  - Gigapixel images can be viewed in stunning detail on video walls—high scientific value to locate pixels with specific conditions (cloud-free sun glint near fresh water coast)
  - Animations are cool. Visit the Northrop Grumman booth on the exhibits floor.
    All Suomi NPP imagery created with Python, ImageMagick and ffmpeg.

PIL has memory issues with large images

- Try this:

```
#!/usr/bin/env python
import numpy as np
from   PIL import Image
size     = 32000, 32000   # pixel columns, pixel rows
n_pixels = np.prod(size)
rgba = zip((255*np.random.rand(n_pixels)).astype(np.uint8),  # red
           (255*np.random.rand(n_pixels)).astype(np.uint8),  # green
           (255*np.random.rand(n_pixels)).astype(np.uint8),  # blue
           (255*np.random.rand(n_pixels)).astype(np.uint8),) # alpha
im = Image.new('RGBA', size)
im.putdata(rgba)
im.save('random_gigapixel.png')
```

- Runs out of memory on machine with 128 GB RAM (should only need 18 GB).

- Work-around
  - write R,G,B,A bytes to binary NetPBM PAM format P7 [http://netpbm.sourceforge.net/doc/pam.html] (~ 4 GB) using header made with conventional `write()` statements and Numpy `.tofile()` method
  - Use ImageMagick `convert` to make JPEG or PNG (~1.3 GB) from PAM P7; 3x size reduction from PAM/P7 to PNG for typical earth images.

Animation with many frames

- Goal: make a movie showing a full day of VIIRS data as seen via a nadir view from the satellite

- basemap makes it easy to overlay data onto a globe.

- Problems:
  - basemap can’t make movies
  - Each orbit is half daytime, half nighttime—need to show different products.
  - Transition at ‘terminator’ can be jarring, need to blend.
  - Want a smooth movie, 30 frames/second!
  - Want high resolution!

- Solution:
  - Make 86,400 high-res frames, one frame per second
  - Render each frame as a three PNG’s with basemap
    - daytime R,G,B from VIIRS reflectance
    - nighttime “near constant contrast” from VIIRS DNB
    - small inset showing day/night terminator
  - Use ImageMagick `convert` command with `–composite` switch to blend day/night frames across the terminator
  - Use `ffmpeg` to make an MP4 out of the blended PNG frames

Challenges

- Large data volume (~ 1.4 TB for desired VIIRS products)
- Large file count (5 PNG’s per frame = > 400,000 image files)
- Computationally prohibitive without a cluster.

- Python tricks:
  - Sped up I/O by running a preprocessing step that extracts VIIRS data from HDF5 files, saves desired portion to Numpy npz files.
  - File “fan out” to multiple directories is essential.

    ```
     bash> for N in `seq -w 0 99`; do mkdir $N; done
    ```

    Then, in Python,

    ```
     file_path = '%02d/%s' % ( hash(filename) % 100, filename )
    ```

- Can make an MP4 (20 GB) movie in about four hours using 2,000 cores.

Debugging Parallel Python Code

- Parallel job division is simple; 86,400 frames across 2,000 cores; same executable, each invocation works on a different set of frames. But:
  - Some invocations produce tracebacks
  - Some invocations seg fault (bugs in basemap)

- Python/Unix trick:
  - Replace exception handler with call to pudb post-mortem debugger (based on http://code.activestate.com/recipes/65287-automatically-start-the-debugger-on-an-exception/ which does the same thing with pdb).
    1. First install `pudb`.
    2. Add to top of your Python code the line
       ```
       import pudb_pm_on_exception
       ```
    3. Run your code as usual.
    4. Tracebacks put you in the debugger! Magic!
  - Submit each program invocation as a Unix ‘screen’ session; successful jobs do their work and close the screen. Tracebacks live on in the debugger.
    - Submit job 332 of 2000:
      ```
      ssh node7743 screen –S job_332 –d –m make_frames.py 332
      ```
    - Then, to investigate the traceback
      ```
      ssh node7743
      screen –DR job_332
      ```

pudb_pm_on_exception.py

```
import pudb
import sys
def info(type, value, tb):
    if hasattr(sys, 'ps1') or not sys.stderr.isatty():
        sys.__excepthook__(type, value, tb)
    else:
        import traceback
        traceback.print_exception(type, value, tb)
        print
        pudb.pm()
pudb.set_interrupt_handler()
sys.excepthook = info
```

Based on
http://code.activestate.com/recipes/65287-automatically-start-the-debugger-on-an-exception/

Summary

- Simple Python tricks (and a cluster!) make the production of large scale imagery easy.
  - Write gigapixel images to PAM files directly, without existing Python modules, then use ImageMagick to convert to desired formats.
  - Manage large file counts by distributing files across directories, where a file's parent directory is a modulus of `hash()` on the file names.
  - IDE's are not well suited for debugging parallel code on clusters. Instead use `pudb` and `screen`.

- Northrop Grumman has the expertise to help you solve your big data problems.

THE VALUE OF PERFORMANCE.

NORTHROP GRUMMAN