NRT-OPS at UAF-GINA

Near real time Satellite processing:
DevOps and web app tooling

Will Fisher - UAF-GINA
Jay Cable - UAF-GINA
UAF-GINA & Alaska National Weather Service

University of Alaska Fairbanks - Geographic Information Network of Alaska

- GINA was formed in 2001 and focused on direct broadcast satellite data reception for Alaska (initially, MODIS and AVHRR). Closely partnered with NOAA-NESDIS-FCDAS in 2004 to improve satellite wildfire monitoring and service to the Alaska Region of the National Weather Service.
- GINA is a team of scientists, remote sensing researchers, developers, systems administrators, geospatial specialists, mappers and a couple former NWS forecasters.
- GINA promotes collaboration at the local, state, and federal levels in the discovery and use of satellite remote sensing and geospatial data.

Alaska National Weather Service & GINA

- NWS in Alaska is an enthusiastic partner and consumer of data from the GINA & FCDAS partnership.
- NWS is a primary customer of low latency, full resolution GINA’s Alaska Direct Broadcast imagery and products.
- NWS and GINA have a close partnership through the High Latitude Proving Ground which supports integration of new GEO & LEO products into Alaska and Arctic operations.
- Near-real-time access for the Alaska NWS to S-NPP/JPSS data is a Key Performance Parameter (KPP) for the NOAA-NESDIS JPSS program; GINA Direct Broadcast helps meet this KPP.
UAF & NOAA NESDIS/FCDAS Processing Systems

UAF GINA Big Dog dish at UAF campus

Sandy Dog dish being installed at FCDAS

NESDIS/FCDAS in Gilmore Creek with hill of antennas

Not pictured: FCDAS Barrow Alaska antenna
Out with the old, in with the new
The old system

- Organically grown over time
- Wide variety of systems involved
- Everything used custom scripts
- Screen sessions
- Different methods of logging results
The new processing systems
End-user NRT direct broadcast needs

● Support generating products using:
  ○ CSPP
  ○ IMAPP/Polar2Grid
  ○ Terrascan
  ○ AAPP
  ○ GDAL
  ○ SeaDAS
  ○ etc....

● Mitigate risk of data gaps, availability, and delivery of data to end users
  ○ Receive data from multiple location, currently 5+ antennas
  ○ Redundant processing systems
  ○ Move from best effort downlink, processing and distribution to 24/7 operations.

● Make it easier for end-user to see system status
GINA NRT direct broadcast goals

Primary goal to replace our organically grown processing systems with a centralized system that was documented and includes a clear upgrade plan.

- **Visible** - easy to access, centralized monitoring of job status and statistics.
- **Available** - redundancy, failover and apply updates with minimal downtime
- **Scalable** - ability to scale up and down processing and storage resource as needed
- **Tooling** - needs to be able to run a wide variety of processing tools from different groups/vendors
How do we accomplish this?

- Infrastructure
- Configuration Management
- Testing
- GINA NRT DB Processing Dashboard
Infrastructure

- Physical hardware
  - Large Generic Compute Nodes - CentOS 6/7
  - Scalable network storage system - Gluster
  - 10GB/40GB Network

- Virtual machines
  - Each worker is a Virtual Machine
  - Currently 16 VMs in Production
  - Additional test environment that mirrors the production processing environment
  - Ability to deploy similar processing systems to AWS or other cloud providers
Configuration Management

- Infrastructure as Code
  - Install packages, configuration files, mounts, permissions
  - We use Chef Software - cookbooks, recipes, knife, kitchen, etc...
    - Other options: Puppet, Ansible, SaltStack, etc...
  - Version Control - Git & Github

- Idempotent
  - Always move forward
  - Converge system into the desired state

- Automation
  - Let the computer do what it’s good at so I can drink more coffee
  - Verify changes before they go into production
  - 5-10 minutes to go from Basic OS Install to Production processing but only takes 30 seconds from my perspective
Chef Overview

- Works with most Linux distros, Windows, Mac, etc...
- 86+ built in resources (packages, templates, execute, etc...)
- Ability to write custom resources
- Many more cookbooks written by community members available at supermarket.chef.io
  - Firewall - iptables, ufw, windows firewall, etc...
  - Web servers - NGINX, Apache, etc...
  - Monitoring tools - NAGIOS, Sensu, etc..
  - Many many more
CSPP SDR Recipe

1. Setups any needed user accounts
2. Downloads and installs 'CSPP SDR' package with the specified version and applied any specified patch.
3. Downloads and installs 'CSPP Ancillary data' files.
4. Sets attributes to handle the processing queues that system will listen to for jobs.
5. Handle any "common" setup tasks like formatting and mounting SSD scratch space, install and configure sidekiq to listen for jobs, etc...

```ruby
include_recipe "sandy_app:::_user"

cspp_package 'SDR' do
  source 'http://XXXXXX/SSEC/CSPP'
  version "2.2"
  patch "2.2.3"
  user 'processing'
  group 'processing'
end

cspp_ancillary_package 'SDR' do
  source 'http://XXXXXX/SSEC/CSPP'
  version "2.2"
  ancillary ['CACHE', 'STATIC']
  user 'processing'
  group 'processing'
end

node.default['sandy']['sidekiq']['queues'] = [
  { name: 'default', priority: 1 },
  { name: 'cspp_sdr', priority: 10 }
]

include_recipe 'sandy_workers:::_common'
```
"If you don’t experiment before you put things into production, production is always an experiment."

- Andrew Clay Shafer
Testing

- Test Kitchen
  - Create and Destroy VMs for testing on the fly
  - Supports - VirtualBox, AWS, Docker, VSphere, etc...
  - Define multiple build configurations (suites)

- Inspec - http://inspec.io
  - Validation
  - Security/Compliance - CIS profiles

- Automation - Continuous Integration (CI) Pipelines
  - Jenkins, Travis CI, Concourse CI
  - Have more confidence in the outcome
Bringing it all together
Web app - NRT Ops dashboard - high level overview of health of system

Satellite pass reception
All systems operational

Processing systems
12 unacknowledged errors in the last 12 hours

Satellites

- **AQUA ➤ MODIS**
  - Last pass: 9 minutes ago

- **METOP-B ➤ AVHRR**
  - Last pass: about 7 hours ago

- **NOAA15 ➤ AVHRR**
  - Last pass: about 1 hour ago

- **NOAA18 ➤ AVHRR**
  - Last pass: about 7 hours ago

- **NOAA19 ➤ AVHRR**
  - Last pass: about 1 hour ago

- **SNPP ➤ VIIRS, CRIS, ATMS, OMPS**
  - Last pass: 26 minutes ago

- **TERRA ➤ MODIS**
  - Last pass: about 7 hours ago
Web app - NRT Ops dashboard - high level overview of health of system

Lots of short passes from GLC
<table>
<thead>
<tr>
<th>Satellite pass reception</th>
<th>Processing systems</th>
</tr>
</thead>
<tbody>
<tr>
<td>All systems operational</td>
<td>12 unacknowledged errors in the last 12 hours</td>
</tr>
</tbody>
</table>

**Satellites**

Times are approximate, not all passes will be captured

- **AQUA** ➤ **MODIS**
  Last pass 9 minutes ago
- **METOP-B** ➤ **AVHRR**
  Last pass about 7 hours ago
- **NOAA15** ➤ **AVHRR**
  Last pass about 1 hour ago
- **NOAA18** ➤ **AVHRR**
  Last pass about 7 hours ago
- **NOAA19** ➤ **AVHRR**
  Last pass about 1 hour ago
- **SNPP** ➤ **VIIRS, CRIS, ATMS, OMPS**
  Last pass 26 minutes ago
- **TERRA** ➤ **MODIS**
  Last pass about 7 hours ago

Network storage issue
Web app - NRT Ops dashboard - high level overview of health of system

Problem fetching ancillary data from upstream
Web app - NRT Ops Dashboard - high level overview of health of system
Drill down into individual data passes received

Activity Logs
Outcomes from processing jobs - STDOUT, STDERR, and more
SNPP/VIIRS Processing Tree
SNPP/VIIRS Processing Branch

The ViirsSDRJob takes input from the previous step (SnppRtstpsJob) and generates SDR products. The green indicates that this step completed successfully.

Following the successful generation of SDR products three additional jobs are run, ViirsEdrJob, ViirsAwipsJob, ViirsGeoTiff. These jobs are not dependent on each other, so can run in parallel on the compute cluster. This allows for more timely product delivery.

The red on ViirsLdmInject indicates a failure. If any jobs had been dependant on the LDM Injection, they would have been automatically canceled.

The ViirsAwipsJob generates products suitable for display in an AWIPS system. When it has successfully completed, the ViirsLdmInject job is executed.

This job takes the AWIPS-ready files from the cache and places them into LDM for consumption by the NWS.
SNPP Pass Information

- Source pass name
- Facility that received the pass
- Acquisition start time
- Average Processing Time for the previous 100 jobs of this type
- APT from time of acquisition start
VIIRS LDM Inject job output

about 5 hours ago

Started: 2017-06-22 16:18:39 UTC
Finished: 2017-06-22 16:18:40 UTC
PT: less than a minute
PT from acquisition: 33 minutes
Duration: 0 seconds
Status: Error (256)

01 20170622T161840.6027722_pginsert[17850] ERROR pginsert.c:454:main() Product already in queue:
   b19e3f1c8b0e8b67753a867da7acc90e   8966192 20170622161840.602614 EXP 000
   UAF_AWIPS_npp_viirs_adaptive_dmb_203_20170622_154810.nc.gz

02 WARNING: the command "pginsert -p UAF_AWIPS_npp_viirs_adaptive_dmb_203_20170622_154810.nc.gz -f EX
/gluster/cache/gilmore/snpp/awips/viirs/2017/06/NPP.20170622.154626.dat.gz/UAF_AWIPS_npp_viirs_adaptive_dmb_203_20170622_154810.nc.gz"
returned an error
Metrics related to job
Number of passes received for NOAA 15, 18, and 19
### Products Query

**Facilities**
- uaf/gina_gilmore
- Leave blank for all

**Satellites**
- snpp
- Leave blank for all

**Sensors**
- Nothing selected
- Leave blank for all

**Levels**
- geotiff_i2
- Leave blank for all

**Start date**
- 06/06/2017

**End date**
- 06/06/2017

----

<table>
<thead>
<tr>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>...</th>
<th>Next</th>
<th>Last</th>
</tr>
</thead>
<tbody>
<tr>
<td>npp.20170610.1749_true_color.tif</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>npp.20170610.1749_true_color.small.png</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>npp.20170610.1749_DNB.tif</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>npp.20170610.1749_DNB.small.png</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>npp.20170610.1749_I05.tif</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>npp.20170610.1749_I05.small.png</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>npp.20170610.1749_I03_I02_I01.tif</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>npp.20170610.1749_I03_I02_I01.small.png</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>npp.20170611.1413_true_color.tif</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>npp.20170611.1413_true_color.small.png</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>npp.20170611.1413_DNB.tif</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>npp.20170611.1413_DNB.small.png</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>npp.20170611.1413_I05.tif</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>npp.20170611.1413_I05.small.png</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Future goals

- **Automate**: Allow the system to spin up new capacity as needed
- **De-duplication**: Merging passes from multiple receiving stations
- **Containerize**: switch from VMs to containers
- **Extend**: Integrate cloud infrastructure (AWS) and balance between existing hardware and cloud.
- **Statistics**: Improve analytics and statistics, more graphs and visualizations.
Questions?

Will Fisher

support@gina.alaska.edu
twitter.com/uafgina
gina.alaska.edu
The Geographic Information Network of Alaska (GINA) has integrated CSPP, GDAL, Terrascan, AAPP and other data processing packages using DevOps principles to build the Alaska Direct Broadcast Near Real Time (ADB-NRT) processing stack. Using a web-based application developed at GINA, we are able to manage, monitor and diagnose the processing of data from reception to distribution for user specific end-products in both production and test environments. The presentation will provide a high level overview of the DevOps principles and tools; including Chef, Habitat, Git, Test-Kitchen and other in-house web apps, used to manage and deploy the processing infrastructure.

GINA has deployed multiple instances of the processing stack, test and production, in order to validate software changes, updated data products without affecting operational users. Using DevOps principles GINA is able to reduce overhead of people and hardware costs, when applying the changes from test to production.

Users that consume data products from the ADB-NRT stack include the National Weather Service offices in Alaska, NWS Alaska Sea Ice Desk, Alaska Fire Service, State of Alaska, Alaska Volcano Observatory, and the general public.
Images/Logos Credits

Robot by Creative Mania from the Noun Project

GitHub OctoCat from & by GitHub

Chef Logo from Chef

Mad Scientist from Wikimedia Commons

Reception Dish & LEO-AK Globe by Dayne at GINA