OAK RIDGE National Laboratory

Workflow manager: installation and maintenance

Mathieu Doucet

Oak Ridge National Laboratory

ORNL is managed by UT-Battelle, LLC for the US Department of Energy

U.S. DEPARTMENT OF ENERGY

Plan for the next few weeks

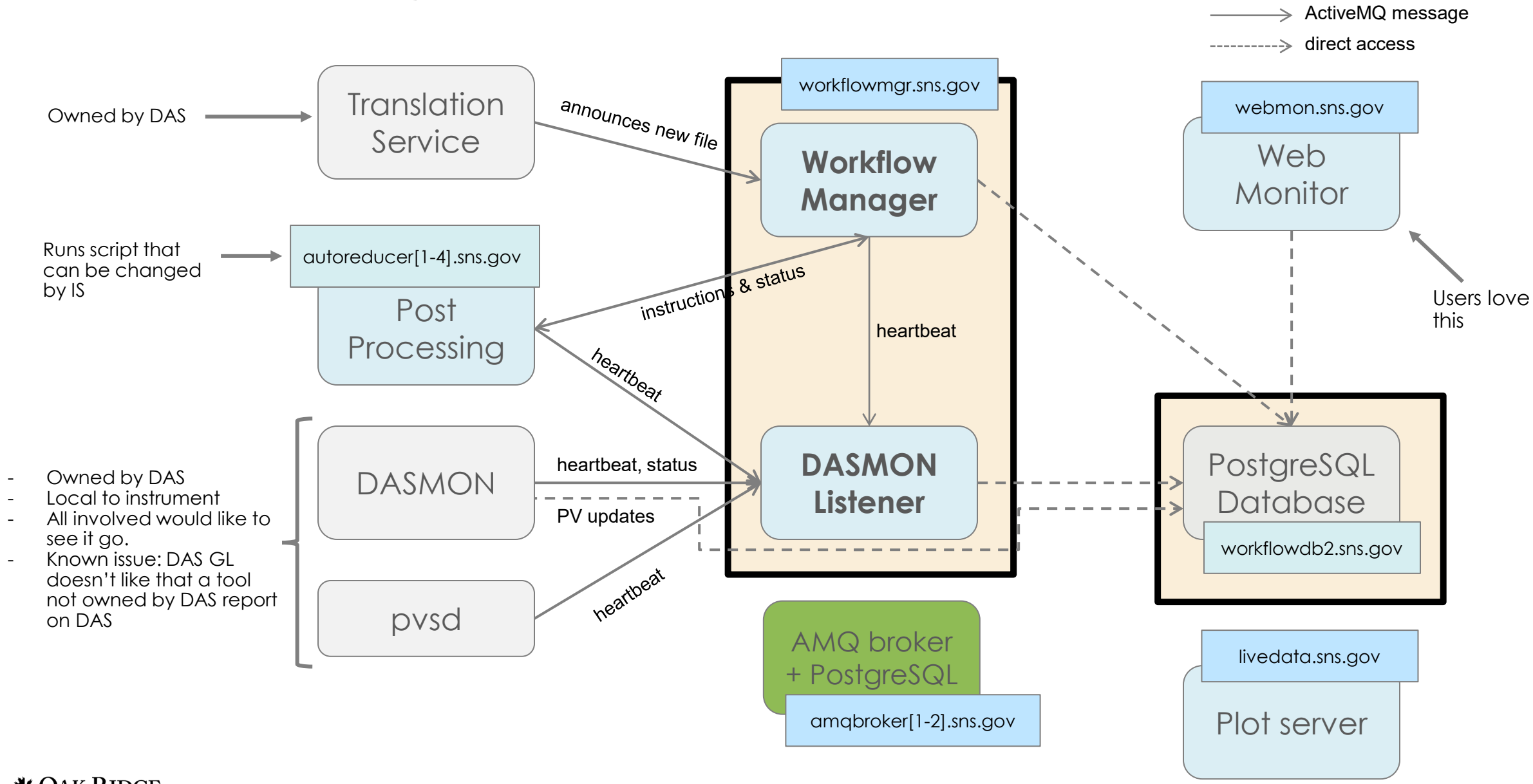
Test environment:

New RHEL8 machines are being set up so we can install them together

Topics to cover:

1. General overview
2. **Workflow manager and DASMON listener – Installation & maintenance [this presentation]**
3. Web monitor – Installation and maintenance
4. Autoreduction service – Installation and maintenance
5. Autoreduction setup through webmon – how-to and future vision
6. The IHC call – when things go wrong & recovery strategies
7. Vision for the future – what I would do differently

Post-Processing Architecture



Installation

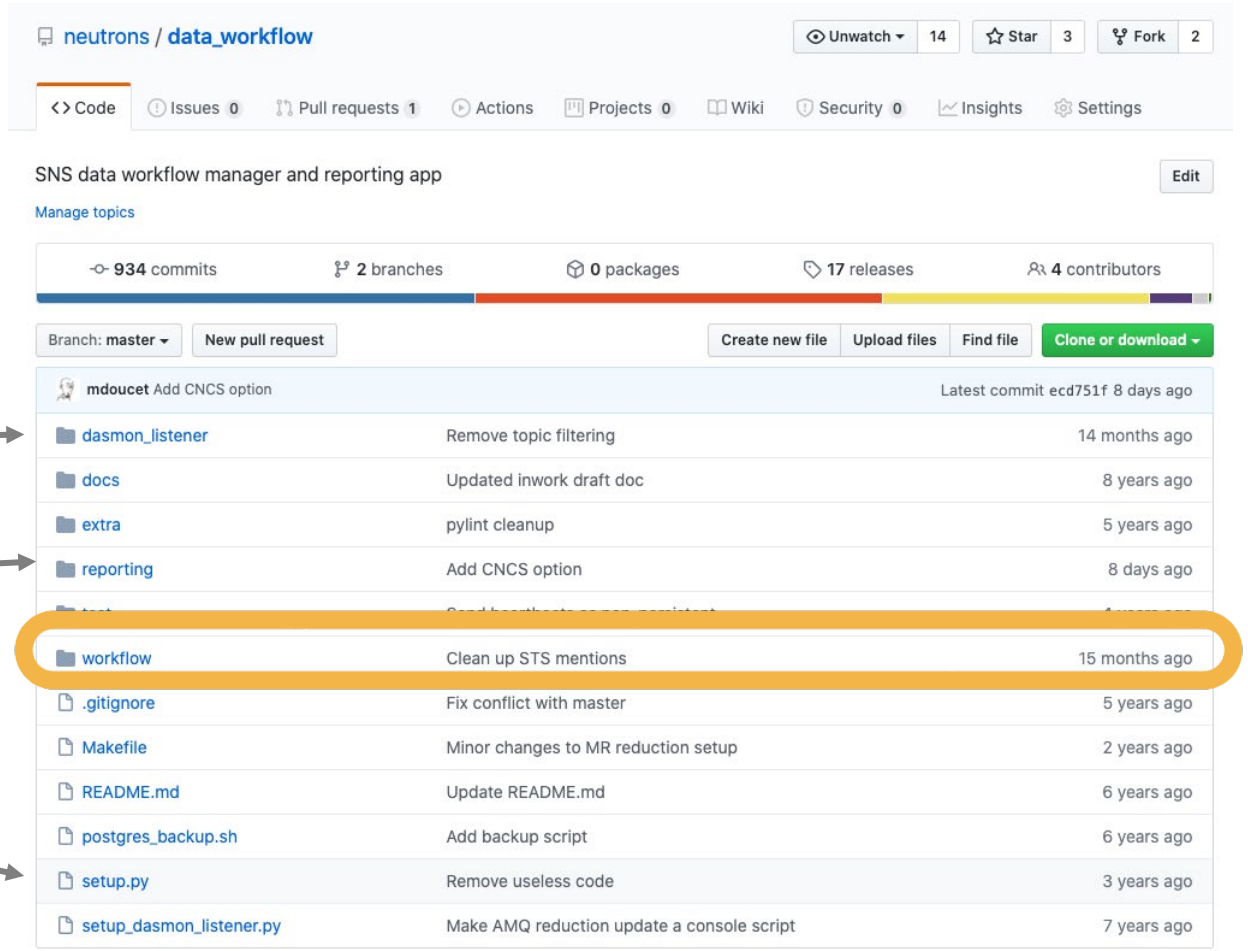
https://github.com/neutrons/data_workflow

- The `workflow` service runs on `workflowmgr.sns.gov`
- The `dasmon_listener` service also runs on `workflowmgr.sns.gov`
- The database is hosted on `workflowdb2.sns.gov`
- The service is an AMQ client, but it uses Django to abstract out DB calls.

Dasmon listener code

Web monitor application code

This installs the workflow package… but use the Makefile



Setting up the database

- Currently runs on `workflowdb2.sns.gov`
- A new test node is available:

  `workflowdbdev.sns.gov`
- Linux Support configures it and maintains it.
- pg_hba.conf changes are made through Linux Support.
- The stored procs are in the workflow repo

  `data_workflow/reporting/report/sql/stored_procs.sql`

Installation for a bare machine:

```
- postgresql installation
  > sudo yum install postgresql postgresql-devel postgresql-server
  > sudo yum install postgresql-libs postgresql-contrib
  > sudo yum install pgadmin3

- postgres setup, performed as postgres user
  > initdb
  > pg_ctl -D /var/lib/pgsql/latest/data -l /var/lib/pgsql/server.log start
  > createuser workflow -W [password will need to be chosen]
  > createdb -O workflow reporting_db

- Update pg_hba.conf and add the webmon and workflowmgr nodes.

- Install stored proc
```

PV entries come in at a high rate. These are only used for instrument monitoring and are deleted after 2 hours.

The postgres IDs will eventually run out!

```
ALTER SEQUENCE pvmon_pv_id_seq RESTART WITH 1;
```

workflow manager configuration

local_settings.py

```
DATABASES = {
    'default': {
        'ENGINE': 'django.db.backends.postgresql_psycopg2',
        'NAME': 'reporting_db',
        'USER': 'workflow',
        'PASSWORD': 'XXXXXX',
        'HOST': 'workflowdb.sns.gov',
        'PORT': '5432',
        }
}

SECRET_KEY = '-0zoc$fl2fa&amp;rmzeo#uh-qz-k+4^1)_9p1qwby1djzybqtl_nn'

TIME_ZONE = 'America/New_York'

USE_TZ = True

INSTALLED_APPS = (
    'report',
    )

"""
    ActiveMQ settings
"""
# List of brokers
brokers = [("amqbroker1.sns.gov", 61613),
           ("amqbroker2.sns.gov", 61613)]

# The is the user that listens only (damson_listener)
icat_user = "wfclient"
icat_passcode = "XXXXXX"

# This is the user for the workflow
wkflow_user = "wkflowmgr"
wkflow_passcode = "XXXXXX"
```

The DB settings are shared between the workflow app and the monitor app.

A number of places are available to write configs.

You could just write a local_settings.py file.

`data_workflow/workflow/settings.py`

imports

`data_workflow/workflow/database/settings.py`

imports (optional, not in repo)

`data_workflow/workflow/local_settings.py`

workflow manager messaging configuration

Tasks as defined in the DB:

Home › Report › Tasks

Select task to change

Search 4 results (73 total)

Action: --------- Go 0 of 4 selected

ID	Instrument id	Input queue id	Task class	Task queues	Success queues
108	seq	REDUCTION.COMPLETE		REDUCTION_CATALOG.COMPLETE;	
33	seq	REDUCTION.REQUEST		REDUCTION.DATA_READY;	REDUCTION.COMPLETE;
32	seq	POSTPROCESS.DATA_READY		REDUCTION_CATALOG.COMPLETE; REDUCTION.DATA_READY; CATALOG.COMPLETE; CATALOG.ONCAT.DATA_READY;	REDUCTION.COMPLETE; CATALOG.COMPLETE;
18	seq	CATALOG.REQUEST		REDUCTION_CATALOG.COMPLETE; CATALOG.ONCAT.DATA_READY;	REDUCTION_CATALOG.COMPLETE; CATALOG.COMPLETE;

Message queues to listen to are read from the DB.

Django administration Welcome, **m2d**. Log out

Home › Report › Status queues

Select status queue to change Add status queue

Action: --------- Go 0 of 33 selected

ID	Name	Is workflow input
33	CATALOG.ONCAT.COMPLETE	✓
32	CATALOG.ONCAT.ERROR	✓
31	CATALOG.ONCAT.STARTED	✓
30	CATALOG.ONCAT.DATA_READY	⛔
29	SMS	⛔
28	REDUCTION.AR4.DATA_READY	⛔

Filter

By is workflow input

All

Yes

No

dasmon_listener configuration

local_settings.py

```
DATABASES = {
    'default': {
        'ENGINE': 'django.db.backends.postgresql_psycopg2',
        'NAME': 'reporting_db',
        'USER': 'workflow',
        'PASSWORD': 'XXXXXX',
        'HOST': 'workflowdb.sns.gov',
        'PORT': '5432',
        }
}

INSTALLATION_DIR = "/var/www/workflow/app"

PURGE_TIMEOUT = 0.125
FROM_EMAIL = "xxxx@ornl.gov"
ALERT_EMAIL = []
IMAGE_PURGE_TIMEOUT = 30

brokers = [("amqbroker1.sns.gov", 61613),
           ("amqbroker2.sns.gov", 61613)]
amq_user = "wfclient"
amq_pwd = "XXX"

queues = ["/topic/SNS.COMMON.STATUS.WORKFLOW.0",
          "/topic/SNS.COMMON.STATUS.AUTOREDUCE.0",
          "/topic/SNS.*.APP.DASMON",
          "/topic/SNS.*.STATUS.DASMON",
          "/topic/SNS.*.SIGNAL.DASMON",
          "/topic/SNS.*.APP.SMS",
          "/topic/SNS.*.STATUS.SMS",
          "/topic/SNS.*.STATUS.POSTPROCESS",
          "/topic/SNS.COMMON.STATUS.ACK",
          "/topic/SNS.*.STATUS.PVSD",
          "/topic/HFIR.*.APP.DASMON",
          "/topic/HFIR.*.STATUS.DASMON",
          "/topic/HFIR.*.SIGNAL.DASMON",
          "/topic/HFIR.*.APP.SMS",
          "/topic/HFIR.*.STATUS.SMS",
          "/topic/HFIR.*.STATUS.POSTPROCESS",
          "/topic/HFIR.COMMON.STATUS.ACK",
          "/topic/HFIR.*.STATUS.PVSD",
]
```

Same story with the config hierarchy.

You could just write a local_settings.py file.

`data_workflow/dasmon_listener/settings.py`

imports

`data_workflow/workflow/database/settings.py`

and imports (optional, not in repo)

`data_workflow/dasmon_listener/local_settings.py`

Setting up the workflow manager

- Runs on `workflowmgr.sns.gov`. A new test node is available on `workflowmgrdev.sns.gov`
- `sudo /usr/sbin/deploy-workflow [path to code]`
- `sudo /usr/sbin/deploy-dasmonlistener [path to code]`
- **`sudo /sbin/service workflowmanager [start|stop|restart]`**
- **`sudo /sbin/service dasmonlistener [start|stop|restart]`**
- Linux support has `workflowmgr` and `dasmonlistener` in `/etc/rc.d/init.d`
- The services tend to leave dead processes behind… it's always good to check

Installation for a bare machine:

```
> sudo useradd workflowmgr
> wget https://bitbucket.org/pypa/setuptools/raw/bootstrap/ez_setup.py -O - | sudo python
> sudo easy_install django==1.6   [We need 1.6]
> sudo yum install python-devel

> sudo yum install postgresql postgresql-devel postgresql-server
> sudo yum install postgresql-libs postgresql-contrib
> sudo easy_install psycopg2

- run deploy-workflow and deploy-dasmonlistener scripts
- Copy service files (workflowmgr and dasmonlistener) in /etc/rc.d/init.d
```

How does it break

- It doesn’t really break
- You might want to restart it after restarting the AMQ brokers as a safety measure
- The workflow and dasmon_listener status are on the web monitor
- The main problem is that the way the service was set up is not optimal and dead processes can be left behind after a restart.

ActiveMQ Communication Flow