Rocoto

HWRF Python Scripts Training

Miami, FL

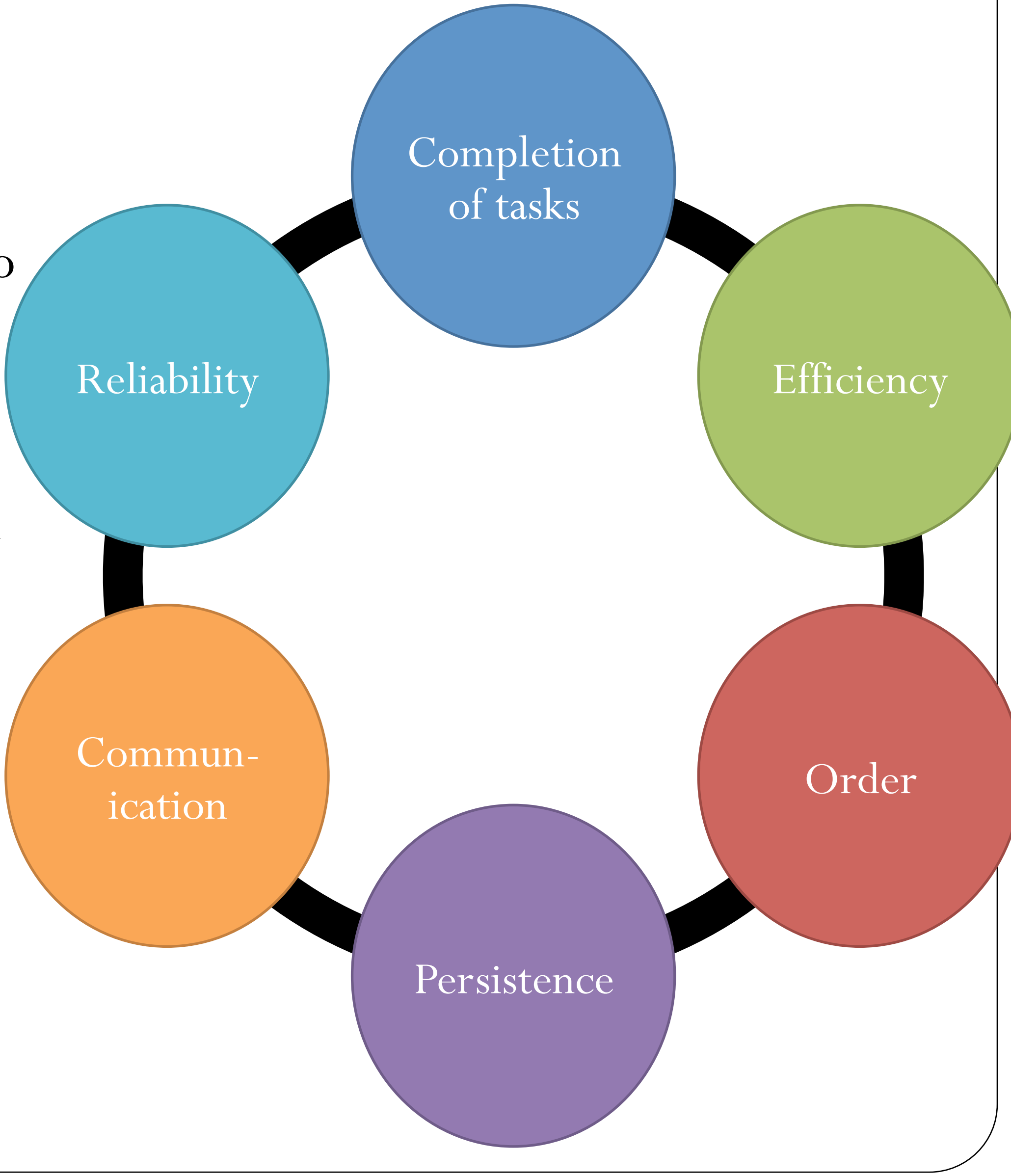
November 19, 2015

Outline

- Introduction to Rocoto
- How it works
  - Overview and description of XML
- Effectively using Rocoto (run, boot, stat, check, rewind, logs)
- Activities:
  - Check status of run (Two cycles: one dead, one hung)
  - Why did it hang?
  - To boot or not to boot?
  - How would you
    - Change the dependencies that make a certain task run (e.g., vortex relocate can only run between 2 and 3 pm, or something else)
    - Tinker with the number of processors used to run each job?

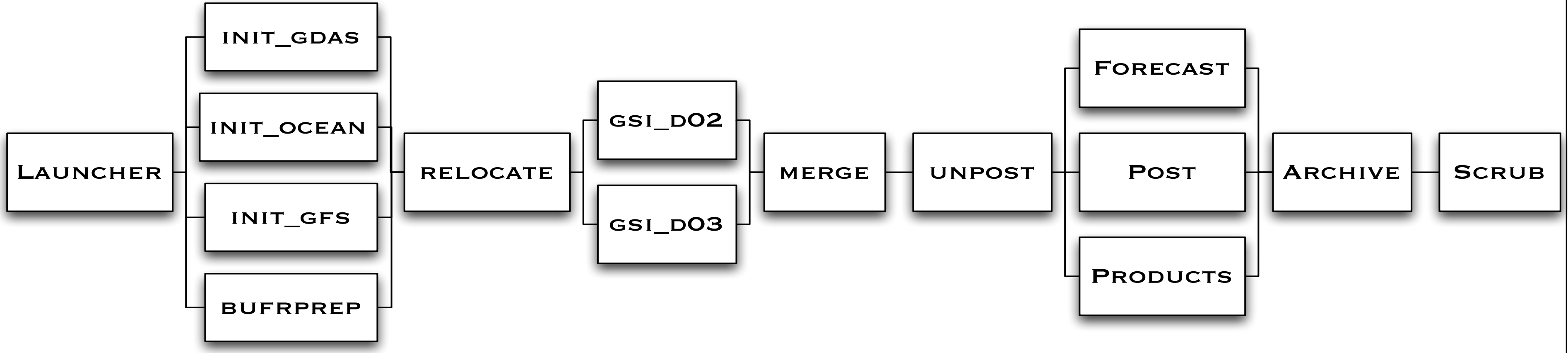
Rocoto's Job

- Workflow management
  - A workflow is a collection of interconnected steps employed to accomplish an overall goal
  - Rocoto is a workflow manager
    - A means of defining a workflow
    - Automation of workflow execution
- Rocoto is capable of
  - Tracking dependencies
  - Checking job status, including failures
  - Resubmitting failed jobs (to a maximum number of attempts)



How Rocoto operates

- Basic overview:
  - Submits a task if its dependencies have been met
  - Run again to check completion of jobs, and whether more jobs can be submitted
  - Continue submitting until all tasks have completed



- Rocoto uses a custom XML language to define the workflow
  - Tasks and interdependencies
  - Runtime requirements (queueing, environment variables)
  - Automation controls

How it works

Rocoto XML introduction

XML Components

- Header
- Entities
- Important tags
  - <workflow>
    - Everything lives inside here
  - <log>
    - Defines the location of the Rocoto log file
  - <cyclestr>
    - References the "current" cycle at runtime
  - <cycledef>
    - Defines the set of cycles to be run for the workflow
  - <task>
    - Job submission portion of workflow
  - <metatask>
    - Collection of tasks

Taken from https://github.com/christopherwharrop/rocoto/wiki/Documentation

Rocoto XML – Environment Variables

HWRF XML EXAMPLE

```
<?xml version="1.0"?>
<!DOCTYPE workflow
[
  <!-- Scrub Times -->
  <!ENTITY COM_SCRUB_TIME "14400">
  <!ENTITY WORK_SCRUB_TIME "1200">
  <!ENTITY CYCLE_THROTTLE "4">

  <!-- External parameter entities -->
  <!ENTITY % SITES    SYSTEM "sites/all.ent">
  <!ENTITY % TASKS    SYSTEM "tasks/all.ent">
  <!ENTITY % STORMS   SYSTEM "storms/H214.ent">
  %SITES;
  %TASKS;
  %STORMS;
:
:
  <!ENTITY EXPT "trunk">
  <!ENTITY SUBEXPT "trunk">
  <!ENTITY HOMEhwrf "/pan2/projects/dtc-hurr/Christina.Holt/CC_rel_branch/&EXPT;">
  <!ENTITY WORKhwrf "/pan2/projects/dtc-hurr/Christina.Holt/CC_rel_branch/pytmp/&SUBEXPT;/@Y@m@d@H/&SID;">
  <!ENTITY COMhwrf "/pan2/projects/dtc-hurr/Christina.Holt/CC_rel_branch/pytmp/&SUBEXPT;/com/@Y@m@d@H/&SID;">
:
:
<!-- Enabling or disabling  parts of the workflow: -->
  <!ENTITY RUN_GSI "YES">
  <!ENTITY RUN_OCEAN "YES">
  <!ENTITY RUN_RELOCATION "YES">
  <!ENTITY EXTRA_TRACKERS "NO">

  <!-- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -->

  <!-- External entities -->
  <!ENTITY ENV_VARS   SYSTEM "env_vars.ent">
  <!ENTITY cycling_condition SYSTEM "cycling_condition.ent">
:
:
<!-- Workflow below here -->
```

Header

HWRF System Variables

parm/*.conf

Variables for include files (rocoto/*)

HWRF Config Variables

rocoto/*

Variables for include files

Rocoto XML - Workflow

```
<!-- Workflow below here -->

<workflow realtime="F" cyclethrottle="&CYCLE_THROTTLE;"
          scheduler="&SCHEDULER;" taskthrottle="20">

  <cycledef>201210280600 201210280600 06:00:00</cycledef>


  <log><cyclestr>&LOGhwrf;/rocoto_&SID;_@Y@m@d@H.log</cyclestr></log>

  <!-- Initialization tasks -->
  <metatask name="meta_init" mode="parallel">
    <var name="ENS">&ENSIDS;</var>
    &launch_task;
    &bdy_task;
    &init_gfs_metatask;
    &init_gdas1_metatask;
    &ocean_init_task;
    &relocate_gfs_metatask;
    &relocate_gdas1_metatask;
    &gsi_metatask;
    &merge_task;
  </metatask>
:
:
</workflow>
```

cyclethrottle and taskthrottle limit the number of cycles or tasks that run at one time

Cycles to run

Log of submit statuses

List of Rocoto tasks to run

Rocoto XML - Workflow

```
<!-- Workflow below here -->

<workflow realtime="F" cyclethrottle="&CYCLE_THROTTLE;"
          scheduler="&SCHEDULER;" taskthrottle="20">

  <cycledef>201210280600 201210280600 06:00:00</cycledef>


  <log><cyclestr>&LOGhwrf;/rocoto_&SID;_@Y@m@d@H.log</cyclestr></log>

  <!-- Initialization tasks -->
  <metatask name="meta_init" mode="parallel">
    <var name="ENS">&ENSIDS;</var>
    &launch_task;
    &bdy_task;
    &init_gfs_metatask;
    &init_gdas1_metatask;
    &ocean_init_task;
    &relocate_gfs_metatask;
    &relocate_gdas1_metatask;
    &gsi_metatask;
    &merge_task;
  </metatask>
:
:
</workflow>
```

cyclethrottle and taskthrottle limit the number of cycles or tasks that run at one time

Cycles to run

Log of submit statuses

List of Rocoto tasks to run

Task

```
<task name="merge_E#ENS#" maxtries="3">

  <command>&EXhwrf;/exhwrf_merge.py</command>
  <jobname>hwrf_merge_&SID;_<cyclestr>@Y@m@d@H</cyclestr>_E#ENS#</
jobname>
  <account>&ACCOUNT;</account>
  <queue>&PE;</queue>
  <nodes>1:ppn=1:tpp=&THREADS;</nodes>
  <envar>
    <name>TOTAL_TASKS</name>
    <value>1</value>
  </envar>
  <walltime>00:39:00</walltime>
  <memory></memory>
  <stdout><cyclestr>&WORKhwrf;/hwrf_merge.out</cyclestr></stdout>
  <stderr><cyclestr>&WORKhwrf;/hwrf_merge.err</cyclestr></stderr>

  &ENV_VARS;
  &RESERVATION;
  &CORES_EXTRA;
  &REQUEST_THREADS;

  <dependency>
    <and>
      <metataskdep metatask="meta_gsi_E#ENS#"/>
      <taskdep task="init_GFS_0_E#ENS#"/>
      <streq><left>&RUN_GSI;</left><right>YES</right></streq>
    </and>
  </dependency>
</task>
```

Queue tags

Set environment variables

Dependencies

Types of Dependencies

- Task `<taskdep>`
  - cycle_offset: `<taskdep task="wrfpost_f006” cycle_offset="-6:00:00"/>`
  - state: `<taskdep state="succeeded" task="X"/>`
- Metatask `<metataskdep>` `tasks/gsi_post.ent`
- Data `<datadep>`
  - age & minsize: `deps/cycling_condition.ent`
- Time `<timedep>` `tasks/launch.ent`
- Cycle exists `<cycleexistdep>` `tasks/launch.ent`
- Grep `<sh> grep…` `tasks/forecast.ent`

Activity 1

Describe the dependencies for the following tasks in words:

1. Relocate GFS
2. Uncoupled Forecast
3. Post_helper

Activity 2

Change the following tasks to have the corresponding dependencies:

1. post and products cannot run until forecast is complete
2. com_scrub should never run
3. relocate must only be scheduled between 2 and 3 pm

Activity: Create a simple XML

- Create an XML script to run HWRF.sh
- Environment variables required
  - HWRF=1
  - BASIN=AL
  - SID=11L
- Cycles: 2015092712-2015093006

Effectively Using Rocoto

To run the Rocoto XML…

- Documentation available here: http://rdhpcs.noaa.gov/rocoto/

```
rocotorun -w XMLFILE -d DATABASEFILE
```

- Generates a database file the first time it’s run
- Must run several times to complete the entire workflow
  - Manually run while debugging
  - Use cron during production
- Performs the following steps each time:
  - Read the database file specified by –d flag
  - Query the batch system for current state of workflow
  - Take action based on state of workflow
    - Resubmit crashed jobs
    - Submit jobs for tasks whose dependencies are now satisfied
  - Save the current state of the workflow in the database file specified by –d flag
  - Quit

qstat

```
qstat -u USERNAME
```

```
Job ID                  Username    Queue    Jobname          SessID  NDS   TSK    Memory   Time     S   Time
----------------------- ----------- -------- ---------------- ------ ----- ------ ------ --------- - ---------
30352530.jetbqs3        Christina.H batch    hwrf_cpl_forecas   8586     1    228     --  02:59:00 R  01:08:40
30352898.jetbqs3        Christina.H batch    hwrf_post_18L_20  15369     1     12     --  02:59:00 R  00:59:16
30352899.jetbqs3        Christina.H batch    hwrf_post_helper  15833     1     12     --  02:59:00 R  00:59:16
30353062.jetbqs3        Christina.H batch    hwrf_products_18   1129     1      6     --  02:59:00 R  00:54:11
```

rocotostat

```
rocotostat -w XMLFILE -d DATABASEFILE -c YYYYMMDDHHMM
```

- Check the status of a set of cycles

```
 CYCLE                        TASK                       JOBID              STATE        EXIT STATUS     TRIES      DURATION
=============================================================================================================
201210280600              launch_E99                    30347274          SUCCEEDED                  0          1          19.0
201210280600                 bdy_E99                    30348043          SUCCEEDED                  0          1        4784.0
201210280600          init_GFS_0_E99                    30347301          SUCCEEDED                  0          1         974.0
201210280600        init_GDAS1_3_E99                    30347302          SUCCEEDED                  0          1        1206.0
201210280600        init_GDAS1_6_E99                    30347303          SUCCEEDED                  0          1        1194.0
201210280600        init_GDAS1_9_E99                    30347304          SUCCEEDED                  0          1        1204.0
201210280600          ocean_init_E99                    30347305          SUCCEEDED                  0          1        1938.0
201210280600      relocate_GFS_0_E99                           -                  -                  -          -             -
201210280600    relocate_GDAS1_3_E99                    30348196          SUCCEEDED                  0          1         488.0
201210280600    relocate_GDAS1_6_E99                    30348198          SUCCEEDED                  0          1         475.0
201210280600    relocate_GDAS1_9_E99                    30348199          SUCCEEDED                  0          1         493.0
201210280600             gsi_d02_E99                    30348505          SUCCEEDED                  0          1        1157.0
201210280600             gsi_d03_E99                    30348509          SUCCEEDED                  0          1         474.0
201210280600               merge_E99                    30349722          SUCCEEDED                  0          1         104.0
201210280600          check_init_E99                    30352258          SUCCEEDED                  0          1          10.0
201210280600    coupled_forecast_E99                    30352530            RUNNING                  -          0           0.0
201210280600    uncoupled_forecast_E99                         -                  -                  -          -             -
201210280600              unpost_E99         druby://fe3:37405         SUBMITTING                  -          0           0.0
201210280600                post_E99                           -                  -                  -          -             -
201210280600         post_helper_E99                           -                  -                  -          -             -
201210280600            products_E99                           -                  -                  -          -             -
201210280600          tracker_d1_E99                           -                  -                  -          -             -
201210280600         tracker_d12_E99                           -                  -                  -          -             -
201210280600              output_E99                           -                  -                  -          -             -
201210280600              completion                           -                  -                  -          -             -
```

SUCCEEDED
RUNNING
SUBMITTING
FAILED
DEAD
UNKNOWN

rocotocheck

```
rocotocheck -w XMLFILE -d DATABASEFILE -c YYYYMMDDHHMM -t TASK
```

- Detailed status info for a specific task in a specific cycle

```
Task: ocean_init_E99
  account: dtc-hurr
  command: /pan2/projects/dtc-hurr/Christina.Holt/CC_rel_branch/trunk/scripts/exhwrf_ocean_init.py
  cores: 9
  final: false
  jobname: hwrf_ocean_init_18L_2012102806_E99
  maxtries: 3
  memory:
  metatasks: meta_init
  name: ocean_init_E99
  native: -l partition=ujet:tjet:vjet:sjet
  queue: batch
  seqnum: 5
  stderr: /pan2/projects/dtc-hurr/Christina.Holt/CC_rel_branch/pytmp/trunk/2012102806/18L/hwrf_ocean_init.err
  stdout: /pan2/projects/dtc-hurr/Christina.Holt/CC_rel_branch/pytmp/trunk/2012102806/18L/hwrf_ocean_init.out
  throttle: 9999999
  walltime: 00:59:00
  environment
    CONFhwrf ==> /pan2/projects/dtc-hurr/Christina.Holt/CC_rel_branch/pytmp/trunk/com/2012102806/18L/storm1.conf
    HOMEhwrf ==> /pan2/projects/dtc-hurr/Christina.Holt/CC_rel_branch/trunk
    PARAFLAG ==> YES
    PYTHONPATH ==> /pan2/projects/dtc-hurr/Christina.Holt/CC_rel_branch/trunk/ush
    TOTAL_TASKS ==> 9
    WORKhwrf ==> /pan2/projects/dtc-hurr/Christina.Holt/CC_rel_branch/pytmp/trunk/2012102806/18L
    jlogfile ==> /pan2/projects/dtc-hurr/Christina.Holt/CC_rel_branch/pytmp/trunk/log/jlogfile
  dependencies
    AND is satisfied
      launch_E99 of cycle 201210280600 is SUCCEEDED
      'YES'=='YES' is true

Cycle: 201210280600
  State: done
  Activated: Fri Oct 10 15:14:35 UTC 2014
  Completed: Fri Oct 10 19:25:10 UTC 2014
  Expired: -

Job: 30347305
  State:  SUCCEEDED (C)
  Exit Status:  0
  Tries:  1
  Unknown count:  0
  Duration:  1938.0
```

rocotoboot

```
rocotoboot -w XMLFILE -d DATABASEFILE -c YYYYMMDDHHMM -t TASK
```

- Forces a task to run, regardless of dependencies

rocotorewind

```
rocotorewind -w XMLFILE -d DATABASEFILE -c YYYYMMDDHHMM -t
TASK1 -t TASK2 -t TASK3
```

- Clear the database of specified tasks
- Resubmit jobs that have dependencies met
- Kills jobs already running or in the queue
- Rewinding the launcher will delete com and work directories
- To rewind an entire cycle, use the –a option

Activity 3

Use Rocoto utilities to find more information about failures

HWRF Layer to Configure XML

- HWRF is a complex system that has many configurable options
  - Choice of configuration can change the steps and the dependencies of each step
  - Rocoto does not have branching capabilities…no logic structures
- Python layer on top of Rocoto layer
  - Populates an XML template that matches your configuration
  - Removes the burden of matching the workflow to the configuration from the user

hwrf/rocoto/

- run_hwrf.py
  - Get environment variables from confs
  - Check for a TCVital record
  - Generate xml from template (or use existing)
  - Source the include file that loads modules
  - Issue rocotorun command
- hwrf_workflow.xml.in
  - Template for xml workflow
- sites/
  - Files containing variables specific to known machines
  - Any machine can be added by copying and modifying one of the sites/ files
- storms/
  - Not currently used
- tasks/
  - Files defining Rocoto tasks specific to HWRF
- cycling_condition.ent
  - Lists of dependencies for cycled runs (rocoto XML file)
- env_vars.ent
  - List of environmental variables defining location of code, conf, output, etc. (rocoto XML file)

hwrf/rocoto/

- `run_hwrf.py`
  - Get environment variables from confs
  - Check for a TCVital record
  - Generate xml from template (or use existing)
  - Source the include file that loads modules
  - Issue rocotorun command
- `hwrf_workflow.xml.in & hwrf_multistorm_workflow.xml.in`
  - Template for xml workflow
- `runhwrf_wrapper`
- `sites/`
  - Files containing variables specific to known machines
  - Any machine can be added by copying and modifying one of the `sites/` files
- `storms/`
  - Not currently used
- `tasks/ & multistorm_tasks/`
  - Files defining Rocoto tasks specific to HWRF
- `deps/`
  - Complex dependencies
- `env_vars.ent, forecast_procs.ent, ms_vars.ent`
  - Variable definitions

Running Rocoto for HWRF

- Arguments for run_hwrf.py are nearly the same as for exhwrf_launch.py

```
./run_hwrf.py –w {XMLfile} –d {DBFILE} {DATE} –n –s sites/sjet.ent
       {STID} HISTORY config.EXPT={EXPT} config.run_gsi=no
```

- `{XMLfile}` is the XML file (optional)
- `{DBFILE}` is the database file (optional)
- `{DATE}`
  - `YYYYMMDDHH-YYYYMMDDHH` for a range of cycles
  - `YYYYMMDDHH` for a single cycle
  - `YYYYMMDDHH YYYYMMDDHH` for two specific cycles
- *`{STID}`* is the storm ID, i.e. 18L for Sandy
- *`{EXPT}`* is the name of parent directory of `rocoto/`
- Can set any conf parameter in this line without editing a conf file
  - e.g. add option: `config.run_gsi=no`

- `–n` turns of invest renumbering
- `–s` to specify site file (optional)
- `–f` for running subsequent instances
- `–m` for running multistorm with a particular storm
- `–M` for running multistorm for a list of basins

Running Rocoto for HWRF

- The first instance of the `run_hwrf.py`
  - Generates the xml code in `rocoto/`
  - Invokes rocotorun which generates database file in `rocoto/`
- Run every few minutes using the –f argument
  - Checks for the completion of tasks
  - Submits tasks when dependencies have been met
  - Does not overwrite db and xml files when –f option is used (asks otherwise)
- Run HWRF with a cron job (crontab –e to edit your jobs)

Questions?

Additional Resources:

Rocoto for HWRF: http://www.emc.ncep.noaa.gov/HWRF/weeklies/OCT14/OCT162014.html
Rocoto: http://rdhpcs.noaa.gov/rocoto/
Cron:
https://sites.google.com/a/noaa.gov/oar-jetdocs/home/getting-things-done/starting-recurring-processes-with-cron#Best_Practices
Rocoto Help: rdhpcs.rocoto.help@noaa.gov