

Running HWRF with Rocoto

Christina Holt and Ligia Bernardet

HWRF Developers Meeting

October 2014

christina.holt@noaa.gov

ligia.bernardet@noaa.gov

With many thanks to Chris Harrop and Sam Trahan



Why we need Rocoto

- Rocoto acts like a server at a restaurant
- A few things can be done right away when we arrive at the table
 - Menus, water glasses, silverware, etc.
- The waiter checks in with us often to ask questions:
 - Are you ready for drinks?
 - Ready for food?
- When the answer is yes, he starts processing our requests.
 - Puts in drink/food orders, brings it to the table when it's ready.

Hi! My name is
Rocoto.
How may I serve you
today?



Rocoto's Mission



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)
- Rocoto is installed in user space and does not require an administrator to set up
- Rocoto is dependable
 - If machine goes down, Rocoto will pick right back up when the machine re-starts

Rocoto

- 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

Rocoto XML – Required Pieces

- List of cycles
- Workflow log location
- Workflow steps
 - Tasks
 - Script to submit
 - Dependencies (time, data, task, metatask)
 - Runtime/queueing requirements
 - Metatasks
 - Collections of tasks

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 HOMEhwrp "/pan2/projects/dtc-hurr/Christina.Holt/CC_rel_branch/&EXPT;">
  <!ENTITY WORKhwrp "/pan2/projects/dtc-hurr/Christina.Holt/CC_rel_branch/pytmp/&SUBEXPT;/@Y@m@dH/&SID;">
  <!ENTITY COMhwrp "/pan2/projects/dtc-hurr/Christina.Holt/CC_rel_branch/pytmp/&SUBEXPT;/com/@Y@m@dH/&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

Variables for include files (rocoto/*)

HWRF Config Variables

Variables for include files

parm/*.conf

rocoto/*

Rocoto XML - Workflow

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

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

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

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

Cycles to run

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

Log of submit statuses

```
<!-- Initialization tasks -->
```

```
<metatask name="meta_init" mode="parallel">
```

```
<var name="ENS">&ENSIDS;</var>
```

List of Rocoto tasks to run

```
&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>
```


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>&LOGhwrff;/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>&EXhwrff;/exhwrff_merge.py</command>
  <jobname>hwrff_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>&WORKhwrff;/hwrff_merge.out</cyclestr></stdout>
  <stderr><cyclestr>&WORKhwrff;/hwrff_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

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	hwrp_cpl_forecas	8586	1	228	--	02:59:00	R	01:08:40
30352898.jetbqs3	Christina.H	batch	hwrp_post_18L_20	15369	1	12	--	02:59:00	R	00:59:16
30352899.jetbqs3	Christina.H	batch	hwrp_post_helper	15833	1	12	--	02:59:00	R	00:59:16
30353062.jetbqs3	Christina.H	batch	hwrp_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	19.0
201210280600	bdy_E99	30348043	SUCCEEDED		0	4784.0
201210280600	init_GFS_0_E99	30347301	SUCCEEDED		0	974.0
201210280600	init_GDAS1_3_E99	30347302	SUCCEEDED		0	1206.0
201210280600	init_GDAS1_6_E99	30347303	SUCCEEDED		0	1194.0
201210280600	init_GDAS1_9_E99	30347304	SUCCEEDED		0	1204.0
201210280600	ocean_init_E99	30347305	SUCCEEDED		0	1938.0
201210280600	relocate_GFS_0_E99	-	-		-	-
201210280600	relocate_GDAS1_3_E99	30348196	SUCCEEDED		0	488.0
201210280600	relocate_GDAS1_6_E99	30348198	SUCCEEDED		0	475.0
201210280600	relocate_GDAS1_9_E99	30348199	SUCCEEDED		0	493.0
201210280600	gsi_d02_E99	30348505	SUCCEEDED		0	1157.0
201210280600	gsi_d03_E99	30348509	SUCCEEDED		0	474.0
201210280600	merge_E99	30349722	SUCCEEDED		0	104.0
201210280600	check_init_E99	30352258	SUCCEEDED		0	10.0
201210280600	coupled_forecast_E99	30352530	RUNNING		-	0.0
201210280600	uncoupled_forecast_E99	-	-		-	-
201210280600	unpost_E99	druby://fe3:37405	SUBMITTING		-	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

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
- 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

Configure HWRF

Namelist options, executables, etc.

```
[exe]
tar=tar
htar=htar
hsi=hsi
```

parm/hwrf.conf

```
gsi={EXEChwrf}/hwrf_gsi
post={EXEChwrf}/hwrf_post
wgrib={utilexec}/wgrib
copygb={EXEChwrf}/hwrf_egrid2latlon
cnvgrib={utilexec}/cnvgrib
tave={EXEChwrf}/hwrf_tave
vint={EXEChwrf}/hwrf_vint
grbindex={utilexec}/grbindex
:
[wrf_namelist]
```

```
time_control.debug_level = 1
physics.var_ric = 1.0
physics.coef_ric_l = 0.16
physics.coef_ric_s = 0.25
physics.co2tf = 1
physics.num_soil_layers = 4
dynamics.euler_adv = .False.
bdy_control.spec_bdy_width = 1
bdy_control.specified = .true.
domains.feedback = 1
domains.num_moves = -99
:
```

Components'
internal options

HWRF
configuration
options

```
:
:
[config]
```

parm/hwrf_basic.conf

```
out_prefix={vit[stormnamelc]}{vit[stnum]:02d}{vit[basin1lc]}.
{vit[YMDH]}

RUNhwrf={SUBEXPT}

# Enable or disable parts of the workflow:
run_gsi=yes           ; GSI and FGAT initialization
run_ocean=yes         ; POM coupling
run_relocation=yes    ; vortex relocation
run_satpost=yes       ; make synthetic satellite products
use_spectral=yes       ; spectral vs. GRIB input for initial & bdy
allow_fallbacks=no    ; if gsi fails, run off of GFS reloc vortex
extra_trackers=no     ; turn on 9km and 27km trackers (extra jobs)
:
```

Configure HWRF

Paths

```
[config]
disk_project=dtc-hurr
input_catalog=jet_{fcsthist}
archive=hpss:{SUBEXPT}/{out_prefix}.tar
```

parm/system.conf

Paths, input catalog, archive

```
[dir]
CDNOSCRUB=/pan2/projects/{disk_project}/{ENV[USER]}/noscrub
CDSCRUB=/pan2/projects/{disk_project}/{ENV[USER]}/pytmp
CDSAVE=/pan2/projects/{disk_project}/{ENV[USER]}
syndat=/lfs1/projects/hwrf-vd/hwrf-input/SYNDAT-PLUS
:
```

Location of input data

```
[jet_hist]
# jet datasets that contain historical data:
inputroot=/lfs1/projects/hwrf-vd/hwrf-input
gfs={inputroot}/HISTORY/GFS.{aYYYY}/{aYMDH}/
gdas1={inputroot}/GDAS1/{aYYYY}/{aYMDH}/
gdasr={inputroot}/GDAS/{aYYYY}/{aYMDH}/
enkf={inputroot}/ENKF/{aYYYY}/{aYMDH}/
syndatdir={inputroot}/SYNDAT-PLUS/
loopdata={inputroot}/LOOP-CURRENT/
hd_obs={inputroot}/RECON/gdas.{aYMD}/
tdr={inputroot}/TDR/{aYYYY}/{aYMDH}/{vit[stnum]:02d}{vit[basin1lc]}/
@inc=gfs_naming,gfs_grib2,gfs_grib1,para_loop_naming
```

parm/hwrf_input.conf

hwrp/rocoto/

- **run_hwrp.py**
 - Get environment variables from confs
 - Check for a TC Vital record
 - Generate xml from template (or use existing)
 - Source the include file that loads modules
 - Issue rocotorun command
- **hwrp_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 HWRP
- **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)

Users may want to skip scrubbing/ archiving

- Edit the workflow template `hwrp_workflow.xml.in`
 - Comment out the archiving tasks and scrubbing tasks, and dependencies

```
<!-- Data delivery tasks -->
<!-- <metatask name="meta_archive" mode="parallel">
  <var name="ENS">&ENSIDS;</var>
  &disk_archive_task;
  &tape_archive_task;
</metatask> -->
:
:
<!-- Scrub disk areas -->
<!-- <metatask name="meta_scrub" mode="parallel">
  <var name="ENS">&ENSIDS;</var>
  &scrub_work_task;
@** if SCRUB_COM==YES
  &scrub_com_task;
@** endif
</metatask>
-->
:
:
<!--           <metataskdep metatask="meta_archive"/>
           <metataskdep metatask="meta_scrub"/>
-->
```

Environment and automation

- The modules and path to Rocoto are set in `ush/hwrf_pre_job.ksh.inc`
 - For Jet, the following modules are loaded
- For now Rocoto path needs to be set to a copy compiled on disk

```
module load intel
module load grads
module load netcdf
module load hsms
module load mvapich2
module use /pan2/projects/hwrf-vd/soft/modulefiles/
module unload emc-utils
module load nco
export PATH=/lfs2/projects/hwrfv3/Samuel.Trahan/rocoto/bin:$PATH
```

Before submitting any jobs...

- You may want to edit the workflow template to add/rm any tasks.
 - Refer to `README.rocoto` for instructions on adding tasks and their dependencies
 - Add a task file (comp. requirements, dependencies, etc.)
 - Edit `tasks/all.ent` to add task
 - Edit `hwr_f_workflow.xml.in` to add the new task in appropriate order
 - Edit other tasks' dependencies accordingly
 - Edit the account found in the `sites/`
- Run the `psychoanalyst.py` sanity check script to ensure that you have met all of the requirements to submit a job

```
ush/psycoanalyst.py {STID} HISTORY config.expt={EXPT}
```

{STID} is the storm ID, i.e. 18L for Sandy

{EXPT} is the name of parent directory of `rocoto/`

Running Rocoto for HWRF

- Arguments for `run_hwrf.py` are 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

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)

Using the cron

- UNIX utility that can run a command at a defined frequency
- Run `run_hwr f.py` every few minutes easily by submitting to cron
- Edit your cron to run the script every 5 minutes, or so
- On Jet, add the following line to receive output from the cron entry

```
MAILTO=email.address
```

More information on cron can be found here:

https://sites.google.com/a/noaa.gov/oar-jetdocs/home/getting-things-done/starting-recurring-processes-with-cron#Best_Practices

Output Directories

- \$WORKhwrf
 - fgat.DATE
 - gfsinit
 - pom
 - gsi_d0[2 | 3]
 - gdas.DATE
 - runwrf
 - regribber
 - intercom

Output Directories

- \$WORKhwrf
 - fgat.DATE
 - gfsinit
 - pom
 - gsi_d0[2 | 3]
 - gdas.DATE
 - runwrf
 - regribber
 - intercom

- fgat.DATE and gfsinit
 - wps
 - prep_hybrid
 - realinit
 - wrfanl
 - ghost
 - relocate
 - regribber
 - tracker
 - realfcst (GFS only)

- intercom
 - “essential” files used by subsequent components

Logs

- `$CDSCRUB/$EXPT/log/jlogfile`
 - Important messages from all cycles of all storms
 - Messages include INFO, WARNING, and CRITICAL
- `$CDSCRUB/$EXPT/log/rocoto_SID_DATE.log`
 - Information concerning submission/completion of all tasks
- `$WORKhwr f/*.out` and `*.err`
 - Standard out and standard error for each task

Customizing the Workflow

- The workflow in the trunk will work for many purposes
 - Physics experiments
 - Source code changes
 - Etc.
- For projects that require running different components, changes in Python scripts, XML tasks, etc. would be necessary
- Advanced changes can be supported on an individual basis

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