Alternative Methods of Running

HWRF Python Scripts Training

Miami, FL

November 20, 2015

Overview

- Usefulness of alternative methods
- Interactive batch jobs and wrappers
- Run ex-scripts from the shell
- Manually run HWRF Python functions

Ways to Run

- Automation system
  - ecFlow
  - Rocoto
- Wrappers
- Interactive batch jobs
- Manually submitting scripts and functions

Usefulness

- When running more than a few cycles of HWRF, it is recommended that some automation capability is used.
- When implementing new capabilities and debugging, the other forms of job submission may be more effective/ efficient in the testing process
- Wrappers can be used to submit the jobs that are supported in the HWRF v3.7a public release. Others would need to be developed as needed.
  - Quickly run one component at a time. (cannot start from the middle)
- Manual execution is ideal for quick turnaround on debugging

Running HWRF with Wrappers

Wrappers

- Each wrapper submits a single component of the system

bufrprep_wrapper	launcher_wrapper
forecast_wrapper	merge_wrapper
gsi_d02_wrapper	post_wrapper
gsi_d03_wrapper	products_wrapper
init_gdas_wrapper	relocate_wrapper
init_gfs_wrapper	unpost_wrapper
init_ocean_wrapper	

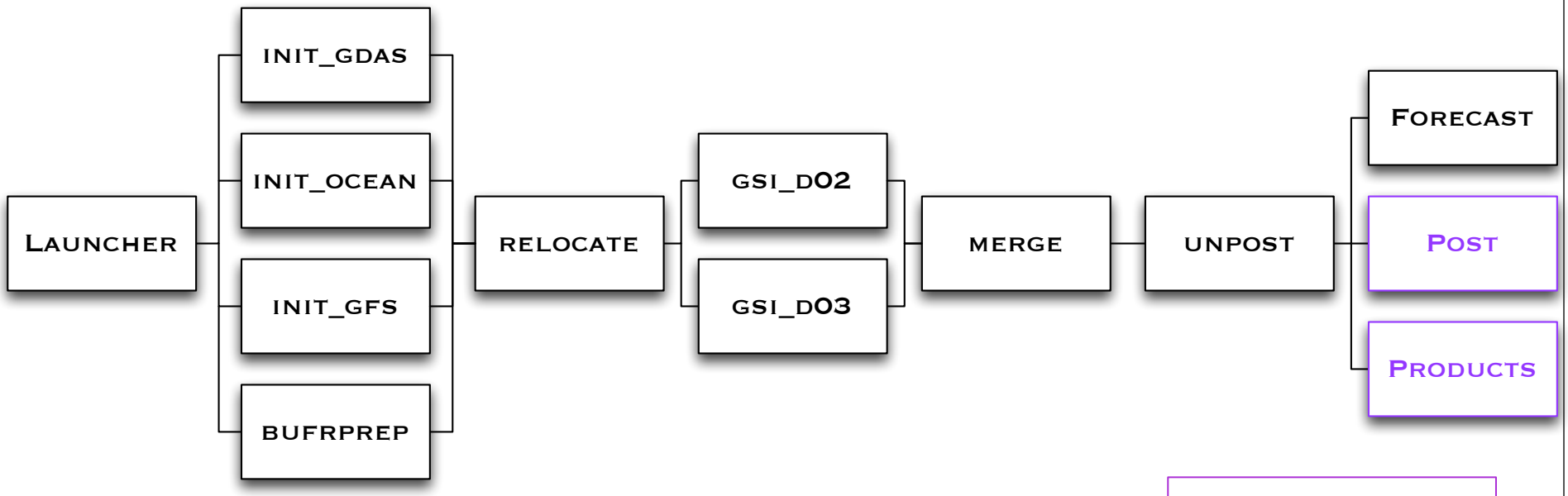
Wrappers: global_vars.ksh

- Each wrapper sources the global_vars.ksh file, which sets a few variables required by each component

```
##### Definition of the Storm ######
export START_TIME=2014101412    # Initial start date
export SID=08L                  # Storm ID
export CASE=HISTORY             # HISTORY OR FORECAST

##### Location of HWRF installation #####
export HOMEhwrf=/PATH/TO/HWRF/INSTALLATION

export EXPT=`echo ${HOMEhwrf} | rev | cut -d/ -f1 | rev`

##### File containing the case-specific variables defined in launcher
#####
export startfile=${HOMEhwrf}/wrappers/$EXPT-${START_TIME}-$SID.start
```

Wrappers

- Wrappers must be submitted in sequence
- Some wrappers may be submitted simultaneously, while others require completion of previous task before submission



Don't submit until forecast job is running

Submitting Jobs

- Each batch system has its own set of requirements for submitting a job
- The following is an example of the resources needed for the forecast job on

```
#!/bin/csh

#BSUB -R "span[ptile=8]"  # how many tasks per node (up to 8)
#BSUB -n $NPROCS          # number of total tasks
#BSUB -o init_gfs.out     # output filename (%J to add job id)
#BSUB -e init_gfs.err     # error filename
#BSUB -J init_gfs         # job name
#BSUB -q regular          # queue
#BSUB -W 1:40             # wallclock time
#BSUB -P PXXXXXXXX        # Account number

$WRAPPER_NAME
```

Run ex-scripts Manually

Go to Sam's documentation