Running HWRF

Christina Holt

Overview

- Configuring HWRF
  - Configuration files
  - Available configurations
- Submitting jobs
- Running HWRF with wrappers

HWRF Configuration

- Most configurable options are controlled by variables that live within the parm/ directory in conf files
- Each conf files contains sections in square brackets, i.e. [config]
- Four primary conf files control all options and required for each run and are called in the following order
  - hwrf_input.conf
  - hwrf.conf
  - hwrf_holdvars.conf
  - hwrf_basic.conf

HWRF Configuration

- Sections may appear in multiple conf files with different contents
- The last configuration option to be set overrides any previous option
- Configuration options can be combined in a group as a new configure file, or in the command line one at a time

hwrf_input.conf

- Contains sections that describe the default locations, naming, and pulling priority order of data on NOAA machines on disk and HPSS

Location of files

```
[jet_hist_PROD2014]
inputroot2014=/lfs3/projects/hwrf-data/hwrf-input   ;; Input root location
inputroot=/lfs3/projects/hwrf-data/hwrf-input       ;; Input root location
…
@inc=gfs2014_naming,para_loop_naming,jet_gefs_naming
```

Sources and priority of each source

```
[jet_sources_PROD2014]
@inc=gfs2014hpss
## Jet history area
jet_hist_PROD2014%location  = file:///
…
```

Names of files

```
[gfs2014_naming]
gfs_sf          = gfs.t{aHH}z.sf{fahr:02d}          ;; GFS spectral forec
gfs_sfcanl      = gfs.t{aHH}z.sfcanl                ;; GFS surface analysis
…
```

hwrf.conf

- Contains all the namelist-type parameters for all components
- Notable sections

```
[dir]
statusfile={WORKhwrf}/{stormlabel}.{YMDH}  ;; cycle status file
intercom={WORKhwrf}/intercom  ;; dir for communicating data files between jobs
…
```

```
[exe]
wgrib={utilexec}/wgrib  ;; wgrib GRIB1 indexing and manipulation program
cnvgrib={utilexec}/cnvgrib  ;; cnvgrib GRIB1/2 conversion program
grbindex={utilexec}/grbindex ;; GRIB1 binary index generation program
mpiserial={utilexec}/mpiserial ;; Executes serial programs via MPI
…
```

To see a full list of sections included in this file, type

```
grep "^\[" hwrf.conf
```

hwrf_holdvar.conf

- Sets variables that are only used to create the holdvars file for ksh
- Nothing in this section is ever used by the Python code
- That vestigial file is just for compatibility with legacy external workflows
- It will be removed eventually

hwrf_basic.conf

- The configuration file responsible for setting directory paths to which the later conf files refer
- Assumes another file has set CDSCRUB, CDSAVE, syndat and CDNOSCRUB variables in the [dir] section
- Configures the workflow-related variables in [config] section

[config]	workflow-related variables
[rocotostr]	string vaariables needed for Rocoto workflow
[rocotobool]	bool variables needed for Rocoto workflow
[prelaunch]	configures overrides for default settings
[sanity]	configures sanity checks
[dir]	directory paths
[archive]	archiving locations and methods

basin_overrides

- basin_overrides is perhaps the most powerful configuration option, from a user standpoint
- When set to yes, it will automatically load an additional configuration script specifically for the basin chosen that corresponds with the operational configuration of the basin
- The values set by the basin-specific conf file will not be overridden by other arguments passed to the launcher

East Pac Storms:

```
[config]
conditional_gsid03=yes
conditional_gsid02=yes
```

Central Pac Storms:

```
[config]
run_gsi=no
run_ocean=no
run_ensemble_da=no
```

basin_overrides

All Other Basins

```
[config]
run_gsi=no
run_ocean=no
run_ensemble_da=no

[wrf]
ptop=5000   ;; Alternative vertical structure: 50 mbar top
ptsgm=20000 ;; This value used in 2013 in WP, IO basins.
dt = 33+3/4 ;; Smaller timestep to handle strong storms.

[namelist_outer]
physics.movemin = 8

[namelist_inner]
physics.movemin = 16

[moad_namelist]
physics.movemin = 8

[wrf_namelist]
domains.eta_levels=1.0, 0.9919699, 0.9827400, 0.9710800, 0.9600599,
0.9462600, 0.9306099, 0.9129300, 0.8930600, 0.8708600, 0.8462000, …
```

Configuration methods

- Directly editing original conf files is not recommended
- Command line arguments vs. passing a new conf file during submission of run_hwrf.py (contents of rocoto/runhwrf_wrapper)
- Some configurations load additional files by default, which contain settings that cannot be overridden by passing additional arguments

Command line arguments

```
./run_hwrf.py -w {XMLfile} -d {DBFILE} {DATE} -n -s sites/sjet.ent
    {STID} HISTORY config.EXPT={EXPT} config.SUBEXPT={anyname}
    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 by passing a list of storms
- `-M` for running multistorm by passing a list of basins

New configure file

- Pass the directory/name of one extra configure file

```
./run_hwrf.py -w {XMLfile} -d {DBFILE} {DATE} -n -s sites/sjet.ent
    {STID} HISTORY config.EXPT={EXPT} hwrf_christina.conf
```

hwrf_christina.conf

```
[config]
disk_project=dtc-hurr
archive=none
scrub_com=no

[relocate]
scrub=no

[gsi_d02]
scrub=no
…
```

hwrf_v3.7release.conf

- For every run, you will need to set some user-specific paths and configuration options based on the capabilities of the public release

```
[config]
disk_project=dtc-hurr
input_catalog=comm_hist
archive=none
publicrelease=yes
run_ensemble_da=no
scrub=no
```

hwrf_v3.7release.conf

```
[dir]
inputroot=PATH/TO/INPUT/DATA
## Syndat directory for finding which cycles to run
syndat={inputroot}/SYNDAT-PLUS
## Output root is the desired output location for HWRF runs
outputroot=PATH/TO/DESIRED/OUTPUT
## Non-scrubbed directory for track files, etc.
CDNOSCRUB={outputroot}/noscrub
## Scrubbed directory for large work files.
CDSCRUB={outputroot}/pytmp
## Save directory.  Must be the parent directory of the HWRF install
CDSAVE=/PATH/TO/HWRF/PARENT

[comm_hist]
inputroot=/PATH/TO/INPUT/DATA/
gfs={inputroot}/gfs.{aYMDH}/
gdas1={inputroot}/gdas1.{aYMDH}/
…
tdr={inputroot}/tdr.{aYYYY}/{aYMDH}/{vit[stnum]:02d}{vit[basin1lc]}/
@inc=gfs2012_naming,gfs2012_grib2,gfs2012_grib1,para_loop_naming,prod_gefs_na
ming,gfs2014_grib

[exe]
…
```

TCVitals

Output Data

Input Data

Paths to Executables

How to build your own configuration

- Example:
  - Run HWRF for an East Pac storm with data assimilation and no ocean coupling. The WRF forecast should use Thompson microphysics.

Should we worry about basin_overrides?
Maybe. Do we need to turn that off?
If so, check hwrf_basic.conf for default.
Submit in command line (launcher_wrapper)

Workflow option?
Yes. Default found in hwrf_basic.conf, maybe overridden in hwrf_EP.conf.

Component option?
Yes. Default found in hwrf.conf, maybe overridden by hwrf_EP.conf.

How to build your own configuration

- Example:
  - Run HWRF for an East Pac storm with data assimilation and no ocean coupling. The WRF forecast should use Thompson microphysics.

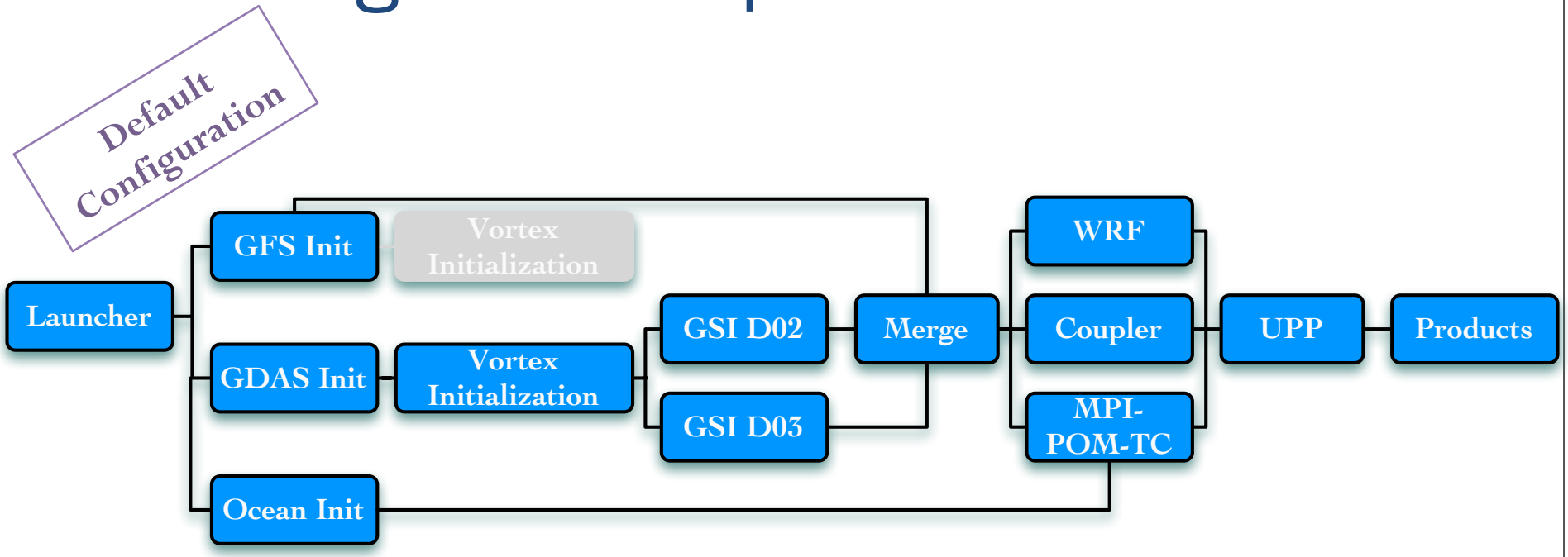
Should we worry about basin_overrides?
Maybe. Do we need to turn that off?
If so, check hwrf_basic.conf for default.
Submit in command line (launcher_wrapper)

Workflow option?
Yes. Default found in hwrf_basic.conf, maybe overridden in hwrf_EP.conf.

Component option?
Yes. Default found in hwrf.conf, maybe overridden by hwrf_EP.conf.

```
[config]
run_gsi=yes
run_ocean=no

[moad_namelist]
physics.mp_physics=8
```

Configuration Options: Workflow



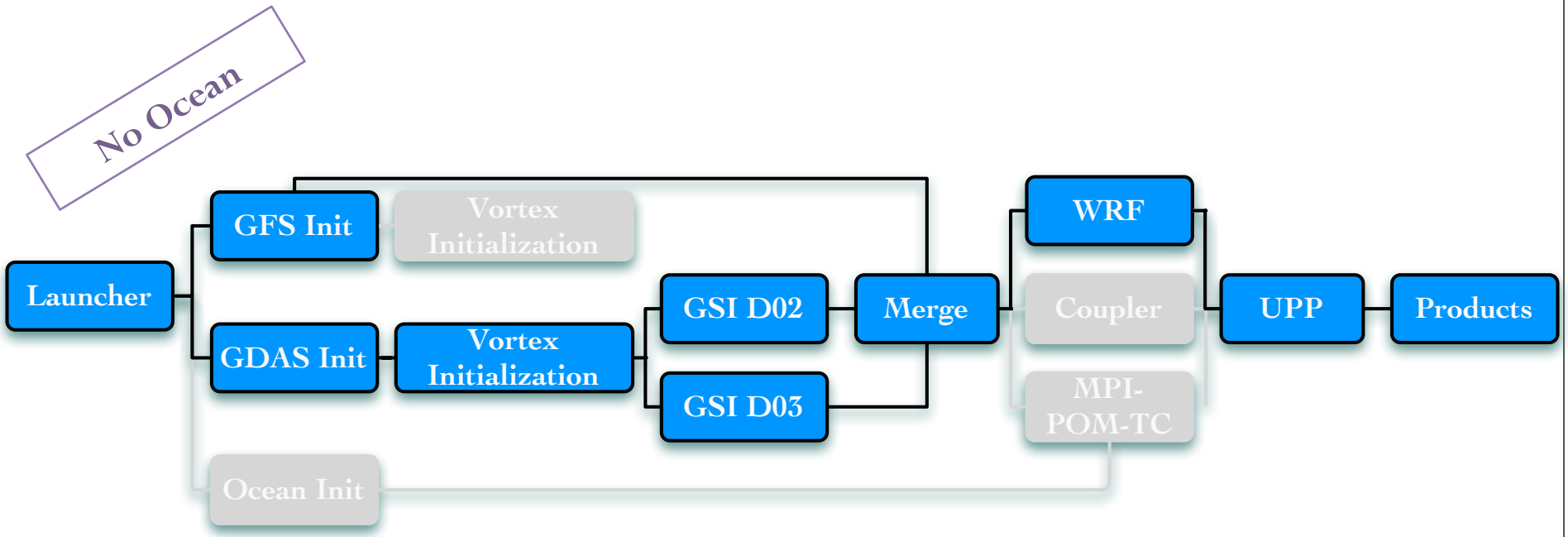
In runhwrf_wrapper,

```
$HOMEhwrf/scripts/exhwrf_launch.py 2014091306 \
    06L HISTORY config.EXPT={EXPT}
```

hwrf_basic.conf

run_gsi=yes
run_ocean=yes
run_relocation=yes
use_spectral=yes

Configuration Options: Workflow



In runhwrf_wrapper,

```
$HOMEhwrf/scripts/exhwrf_launch.py 2014091306 \
    06L HISTORY config.EXPT={EXPT}
    config.run_ocean=no
```

hwrf_basic.conf

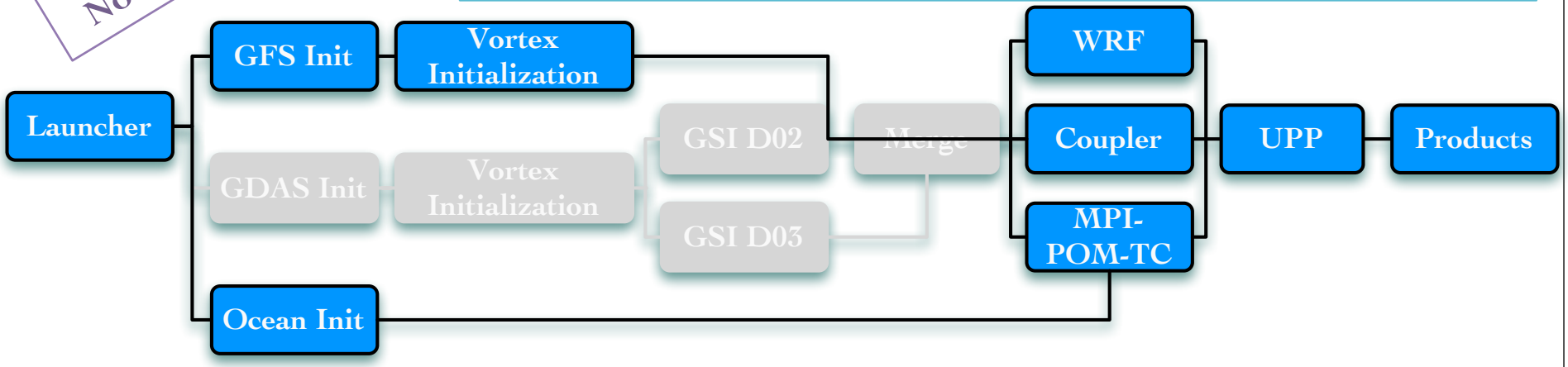
run_gsi=yes
run_ocean=no
run_relocation=yes
use_spectral=yes

Configuration Options: Workflow

In runhwrf_wrapper,

```
$HOMEhwrf/scripts/exhwrf_launch.py 2014091306 \ 06L
    HISTORY config.EXPT={EXPT} config.run_ocean=no
    config.use_spectral=no
```

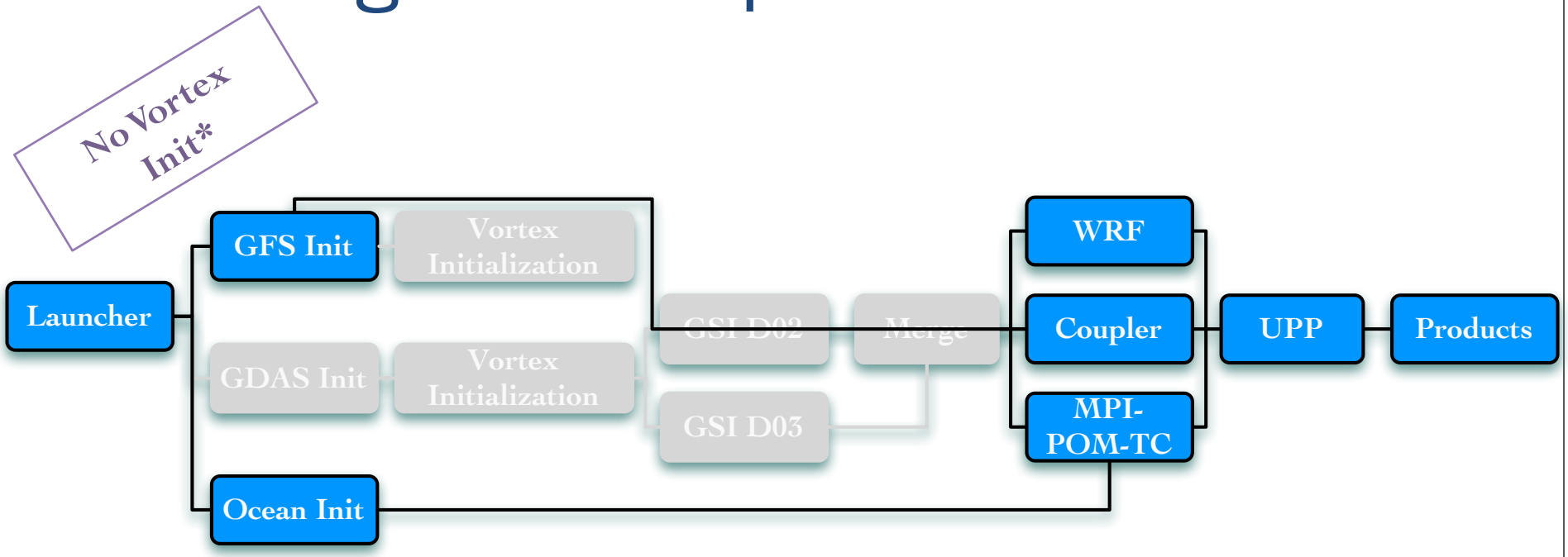
No GSI
No Spectral*



hwrf_basic.conf		hwrf_basic.conf
run_gsi=no run_ocean=yes run_relocation=yes use_spectral=yes	**OR**	run_gsi=no run_ocean=yes run_relocation=yes use_spectral=no

*No spectral requires No GSI to be explicitly set

Configuration Options: Workflow



In runhwrf_wrapper,

```
$HOMEhwrf/scripts/exhwrf_launch.py 2014091306 \
    06L HISTORY config.EXPT={EXPT}
    config.run_gsi=no config.run_ocean=no
```

hwrf_basic.conf

run_gsi=no
run_ocean=yes
run_relocation=no
use_spectral=yes

*No relocation requires No GSI to be explicitly set

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
```

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	

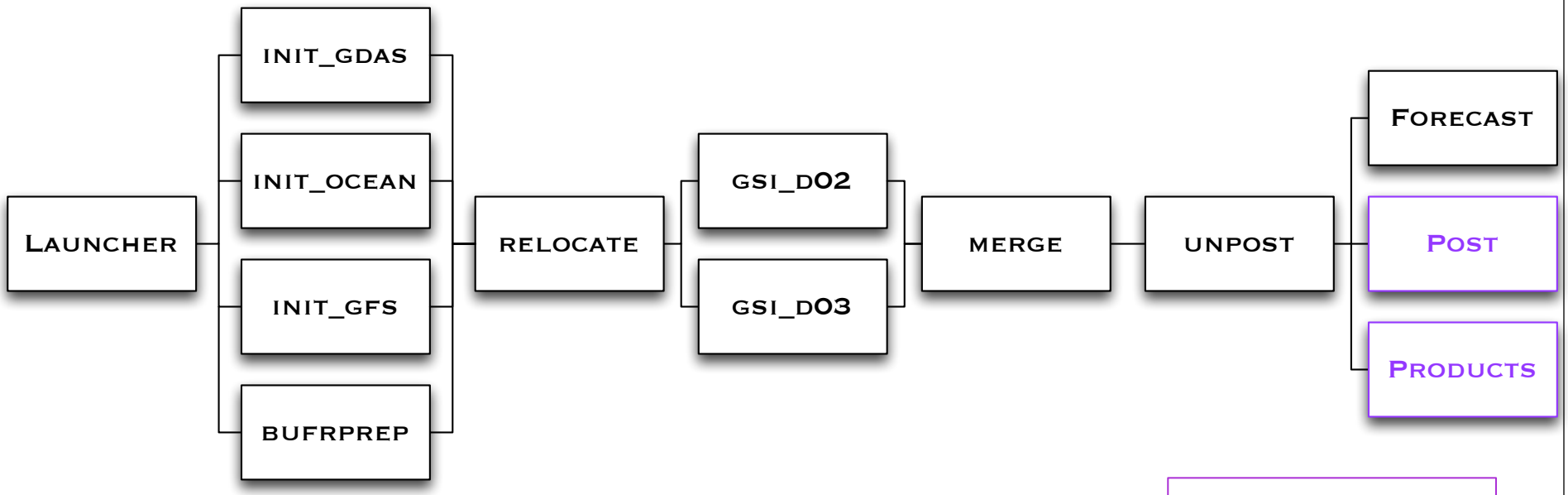
Wrappers: global_vars.ksh

- 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