Overview of the Current HAFS Workflow and Configuration System

Bin Liu, Henry Winterbottom, Avichal Mehra
and The EMC Hurricane Project Team
In collaboration with the UFS hurricane application team

CROW Community Review Meeting
04/28/2020

Objectives of the HAFS Workflow Development

(Hurricane Analysis and Forecast System)

- HAFS is the UFS hurricane application, providing forecast for hurricane track, intensity, and related effects out to about one week.
- Developing and advancing HAFS is one of the key strategies of the new HFIP plan to address its science and R2O challenges, in response to the Weather Act of 2017.
- The HAFS development targets an operational data assimilation and modeling system, capable of providing reliable, robust and skillful model guidances for hurricane forecasting.
- HAFS also aims to be a community-based coupled earth modeling system, promoting cutting-edge research on TC dynamics and physics, advanced data assimilation techniques, and air-sea interaction processes.
- The HAFS workflow development intends to build a common workflow to support both operational and research hurricane applications, promoting easier Research to Operation to Research (R2O2R) transitions.

HAFS Code Repository and Management

The authoritative HAFS repository:

- https://github.com/NOAA-EMC/HAFS
- Supports the main development activities and operational implementations.
- Mainly hosts the **develop** and **master** branches, plus some implementation branches/tags.

The personal HAFS forks:

- Developer's forks for individual feature (or capability) development.
- New developments/features can be integrated back into the authoritative repository or the trusted forks through GitHub Pull Requests.

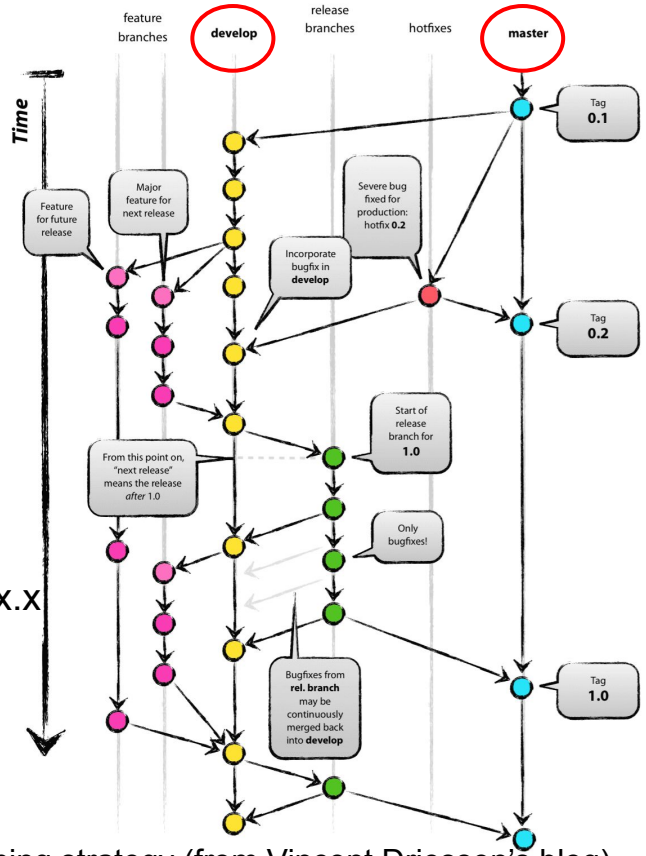
The community/organizational HAFS forks:

- e.g., https://github.com/hafs-community/HAFS
- Mainly provides community support and promotes organizational level collaborations.
- Host HAFS related developments for submodule repositories/forks.

- HAFS GitFlow Rational
- EMC Github repository code management

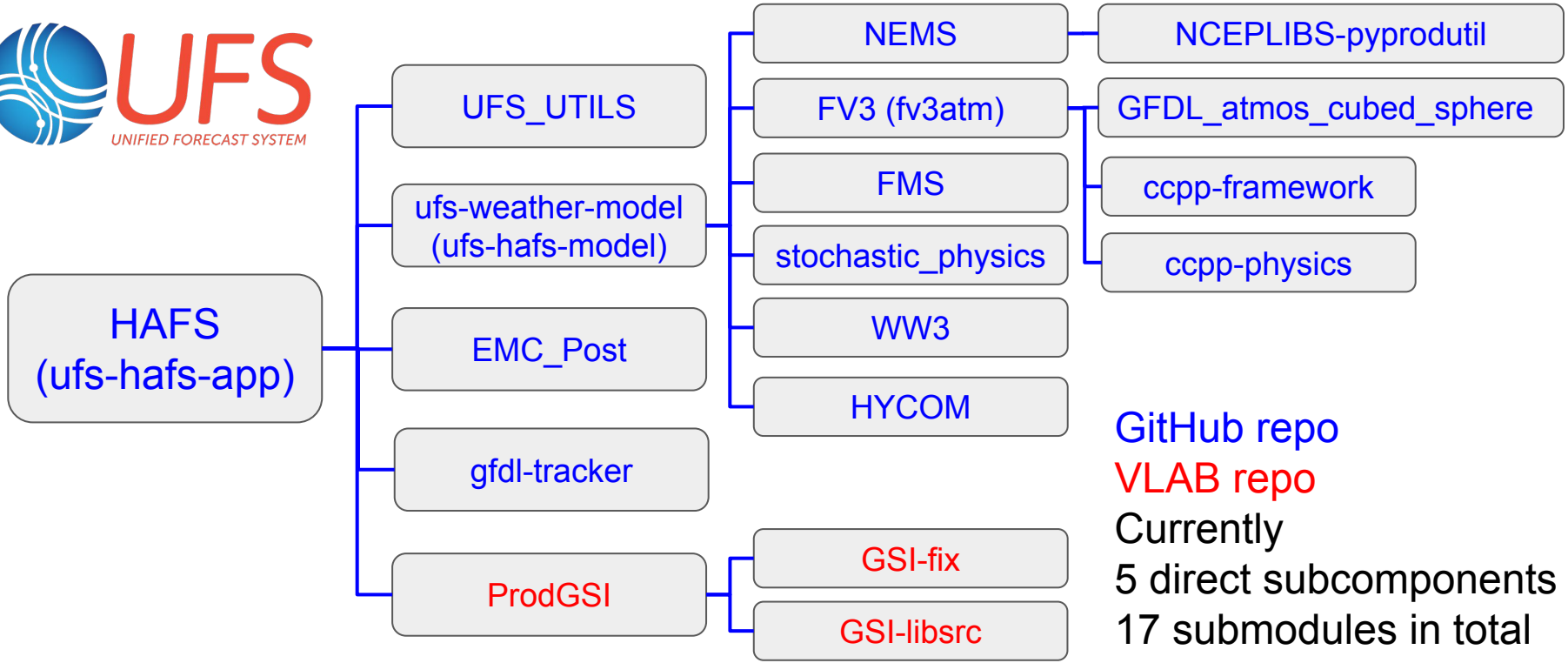
Branch naming:

- **develop**
- **master**
- **support**/[name]
- **product**/hafs.vx.x.x
- **release**/vx.x.x
- **feature/**[name]
- **hotfix**/[name]



A GitFlow branching strategy (from Vincent Driessen's blog)

HAFS Subcomponents/Submodules



GitHub repo

VLAB repo

Currently

5 direct subcomponents

17 submodules in total

HAFS Workflow Development Requirements

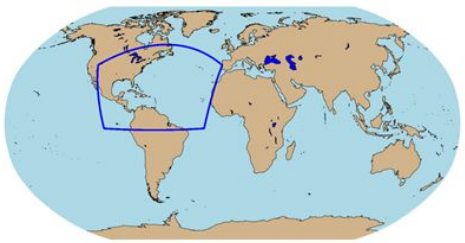
A common workflow to support both research and operational hurricane applications

- Compliant with operational standards for easy operational transition and implementation
  - NCO Implementation Standards (Latest version 10.2)
  - Recommendation for Adoption of NCEP EE2 Process Document
- Easy to learn and use to help promote community modeling and development
- Specialized for hurricane specific applications
  - Support event-triggered configuration for forecasting active storms as well as continuously-cycled configuration for TC genesis forecasting
  - Support regional, global-nesting, and global (uniform/stretched) configurations, as well as the storm-following moving nesting capability
  - Include sophisticated vortex initialization for warm-starting and cycling of the storm and advanced data assimilation techniques for high-resolution innercore DA
  - Optimized for hurricane dynamics and physics, as well as for air-sea interaction and coupling (eventually, earth system coupling)
  - Generate TC specific products and diagnostics

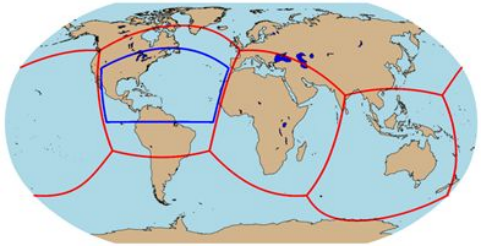
Current HAFS Workflow Development

- The current HAFS workflow is developed based on
  - HWRF/HMON workflows
  - FV3CAM regional workflow
  - FV3GFS global workflow
- Benefits from using subcomponent code repository with other UFS applications
- Supports both event-triggered and continuously-cycled configurations
- Supports both standalone regional and global-nesting configurations
- Supports both basin-focused and storm-focused configurations

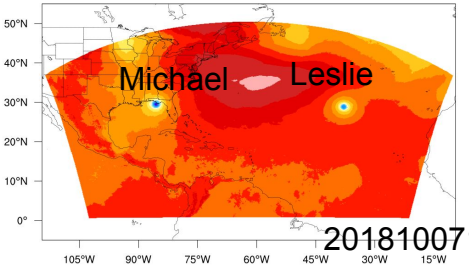
regional



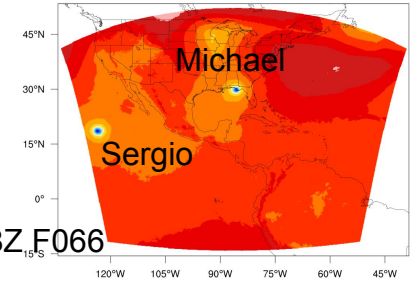
global-nesting



basin-focused



storm-focused



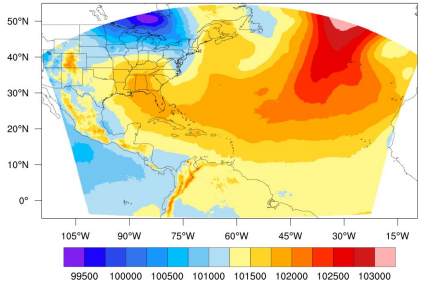
2018100718Z F066

HAFS A: basin-focused regional
HAFS B: basin-focused global-nesting
HAFS C: storm-focused regional
HAFS D: storm-focused global-nesting

Current HAFS Workflow Development

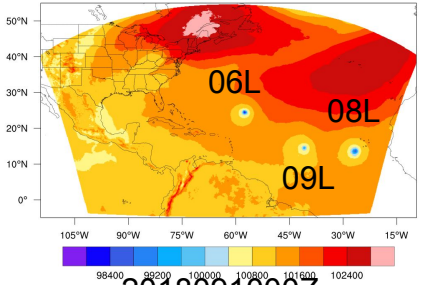
- Supports zero-storm (TC genesis, or any other regional/global-nesting applications), one-storm, and multiple-storm scenarios
- Supports C768 as well as other resolutions (C96, C192, C384, C1152, C1536, etc) for both regional and global-nesting configurations
- Includes TC specific pre-processing and post-processing workflow elements
- Supports WCOSS (Cray and Dell) as well as other NOAA RDHPCS (Jet and Hera)

zero-storm

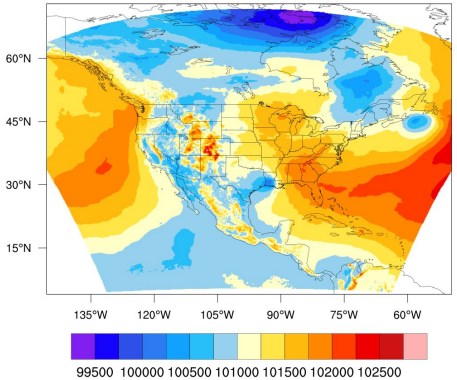


2019062512Z

multiple-storm



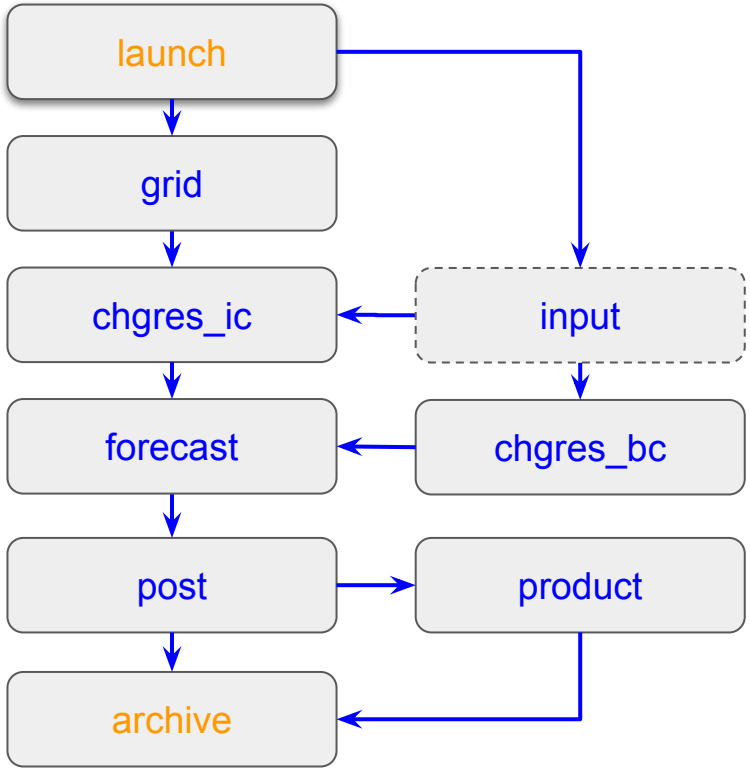
2018091000Z
Florence, Helene, Isaac



2019071400Z
for a CONUS domain

The Current HAFS Workflow

- Currently, the HAFS workflow is a Rocoto based workflow. It can be easily converted/ported to support EcFlow for future production implementation.
- Currently both Python and Shell based scripts are utilized in the workflow components. And some of the Shell script based jobs/tasks are very similar to those used by other UFS applications.
- The HAFS configuration/launch system is adapted/improved from the HWRF/HMON system, which is critical to support all different capabilities needed for hurricane applications.



A Quick Start for HAFS Users

A. Clone the repository

git clone --recursive https://github.com/NOAA-EMC/HAFS.git

B. Build and install

```
cd HAFS/sorc
./build_all.sh
./install_all.sh
./link_fix.sh
```

C. Configure and run HAFS

```
cd ../parm
cp system.conf.hera system.conf
cd ../rocoto
vi cronjob_hafs.sh
./cronjob_hafs.sh
```

Repeat running this driver script or add it as a cron task to advance the workflow.

Cheers! You are now running the HAFS workflow!

```
#!/bin/sh
cd /scratch1/NCEPDEV/hwrf/save/${USER}/HAFS/rocoto
./run_hafs.py -f -s sites/hera.ent 2019 05L HISTORY # Dorian
```

```
# Run one cycle of a storm (Humberto 09L2019)
./run_hafs.py -f -s sites/hera.ent 2019091600 09L HISTORY config.EXPT=HAFS

# Run some specified cycles of a storm (Florence 06L2018)
./run_hafs.py -f -s sites/hera.ent 2018083018-2018083100 06L HISTORY \
   config.EXPT=HAFS config.SUBEXPT=HAFS_try1
```

Note: a more detailed HAFS developers guide is available here.

HAFS Workflow Regression Test Examples

(HAFS/rocoto/cronjob_hafs_rt.sh)

```
#!/bin/sh
set -x
date
# NOAA RDHPCS Hera
HOMEhafs=/scratch1/NCEPDEV/hwrf/save/${USER}/save/HAFS
dev="-s sites/hera.ent -f"
PYTHON3=/apps/intel/intelpython3/bin/python3
cd ${HOMEhafs}/rocoto
EXPT=$(basename ${HOMEhafs}) #HAFS
scrubopt="config.scrub_work=no config.scrub_com=no"

# Regional static NATL basin-focused configuration with GFS nemsio format IC and grib2ab format BC
 ${PYTHON3} ./run_hafs.py -t ${dev} 2019091600 09L HISTORY \
     config.EXPT=${EXPT} config.SUBEXPT=${EXPT}_rt_regional_static_grib2ab_lbc \
     config.ictype=gfsnemsio config.bctype=gfsgrib2ab_0p25 \
     config.NHRS=12 ${scrubopt} ../parm/hafs_regional_static.conf

# Regional storm-focused configuration with GFS nemsio format IC and grib2 format BC
 ${PYTHON3} ./run_hafs.py -t ${dev} 2019091600 09L HISTORY \
     config.EXPT=${EXPT} config.SUBEXPT=${EXPT}_rt_regional_grib2_lbc \
     config.ictype=gfsnemsio config.bctype=gfsgrib2_0p25 config.NHRS=12 ${scrubopt}
```

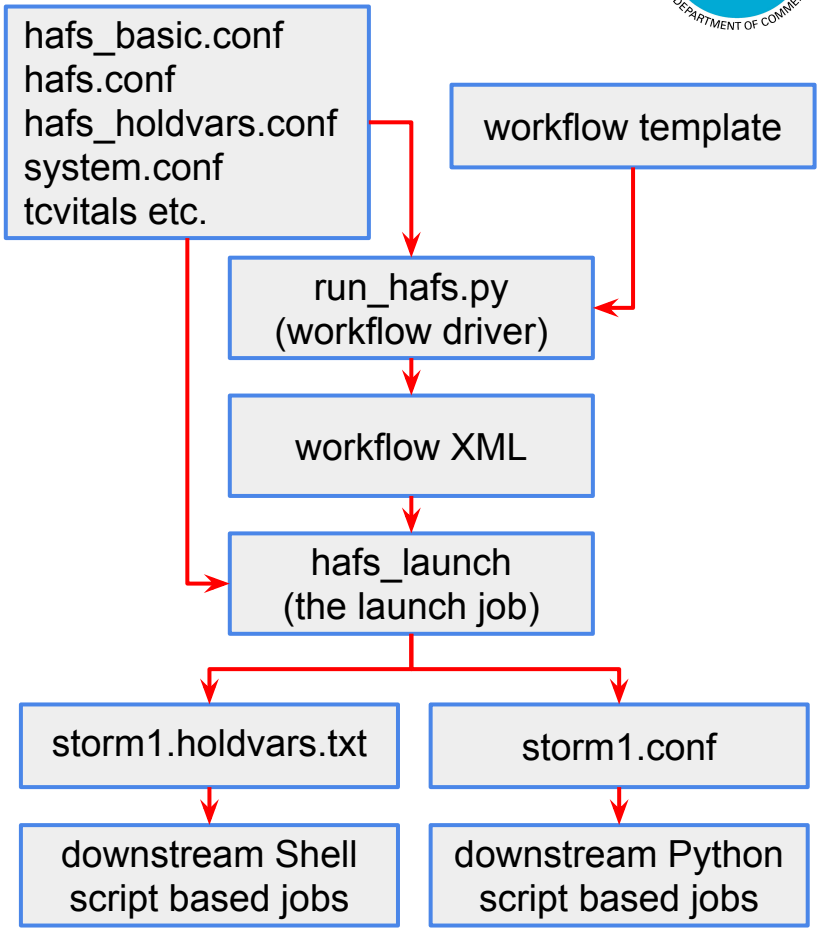
HAFS Workflow Regression Test Examples

(HAFS/rocoto/cronjob_hafs_rt.sh)

```
# Global-nesting NATL basin-focused configuration with GFS nemsio format IC
 ${PYTHON3} ./run_hafs.py -t ${dev} 2019091600 09L HISTORY \
     config.EXPT=${EXPT} config.SUBEXPT=${EXPT}_rt_globnest_static \
     config.NHRS=12 ${scrubopt} ../parm/hafs_globnest_static.conf

# Global-nesting storm-focused configuration with GFS nemsio format IC
 ${PYTHON3} ./run_hafs.py -t ${dev} 2019091600 09L HISTORY \
     config.EXPT=${EXPT} config.SUBEXPT=${EXPT}_rt_globnest \
     config.NHRS=12 ${scrubopt} ../parm/hafs_globnest.conf

# Fakestorm regional_C96s1n4_180x180 configuration
 ${PYTHON3} ./run_hafs.py -t ${dev} 2019091600 00L HISTORY \
     config.EXPT=${EXPT} config.SUBEXPT=${EXPT}_rt_regional_C96s1n4_180x180 \
     config.NHRS=12 ${scrubopt} \
     ../parm/examples/hafs_regional_C96s1n4_180x180.conf ../parm/hafs_fakestorm.conf

# Fakestorm globnest_C96s1n4_180x180 configuration
 ${PYTHON3} ./run_hafs.py -t ${dev} 2019091600 00L HISTORY \
     config.EXPT=${EXPT} config.SUBEXPT=${EXPT}_rt_globnest_C96s1n4_180x180 \
     config.NHRS=12 ${scrubopt} \
     ../parm/examples/hafs_globnest_C96s1n4_180x180.conf ../parm/hafs_fakestorm.conf
```

The HAFS Configuration and Launch System

- The current HAFS configuration system is based on the ConfigParser (INI) format configuration files (parm/hafs*.conf, etc.)
- The workflow driver (run_hafs.py) uses the config files to generate the workflow definition file and control its elements, resources, etc.
- The launch job parses the config files and tcvitals messages to set up the experiment, creates the working directory structure, and write out the domain information and the storm tcvital related files
  - storm1.vitals.*: storm tcvital history files
  - tmpvit/oldvit: the tcvital records
- It generates the storm1.conf file to configure all the downstream Python script based jobs/tasks, and the storm1.holdvars.txt file to configure all the downstream Shell script based jobs/tasks

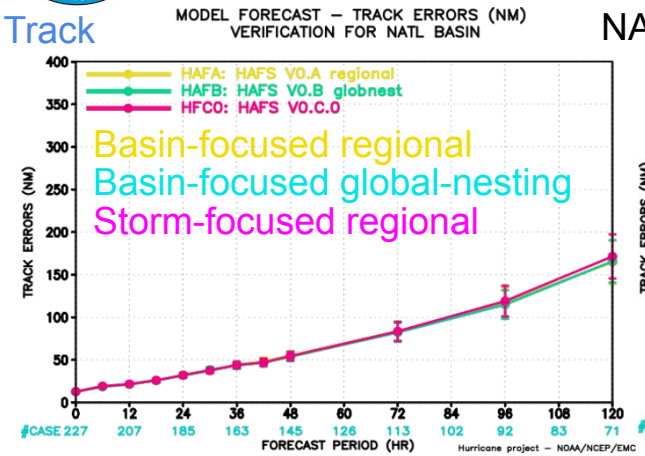


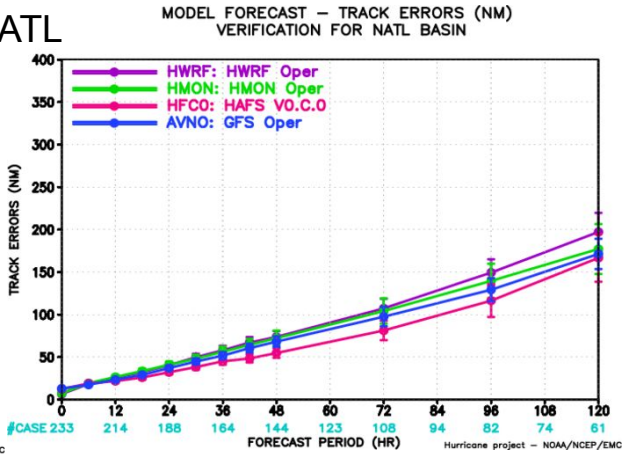
Support 2019 HFIP Real-Time Experiments

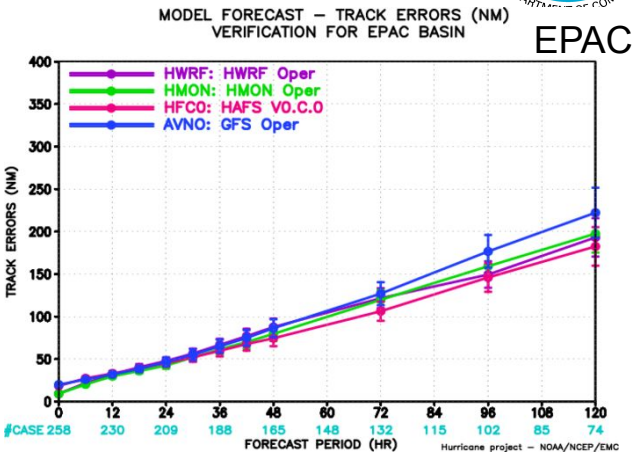
Track

NATL

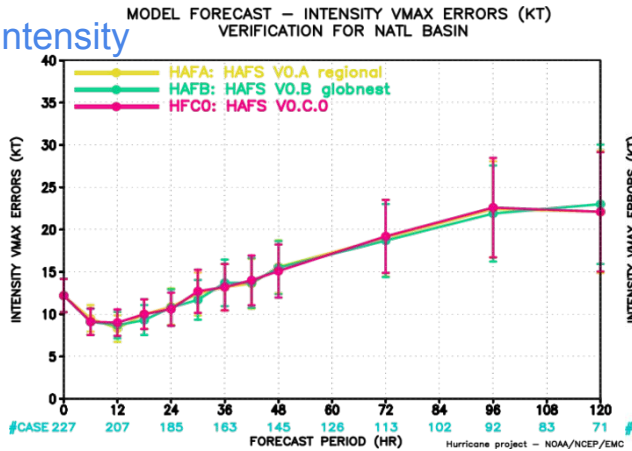
EPAC

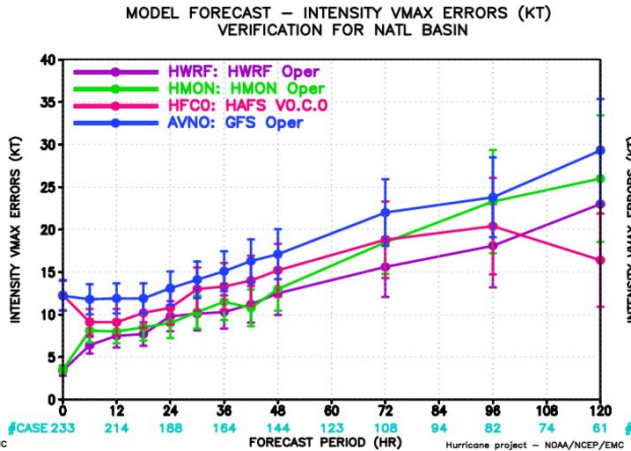


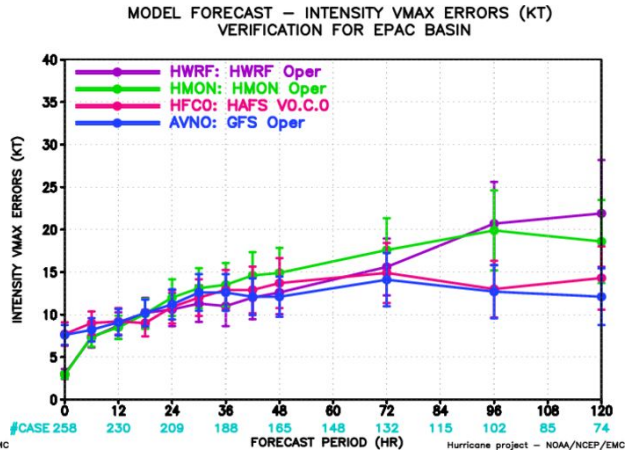




Intensity

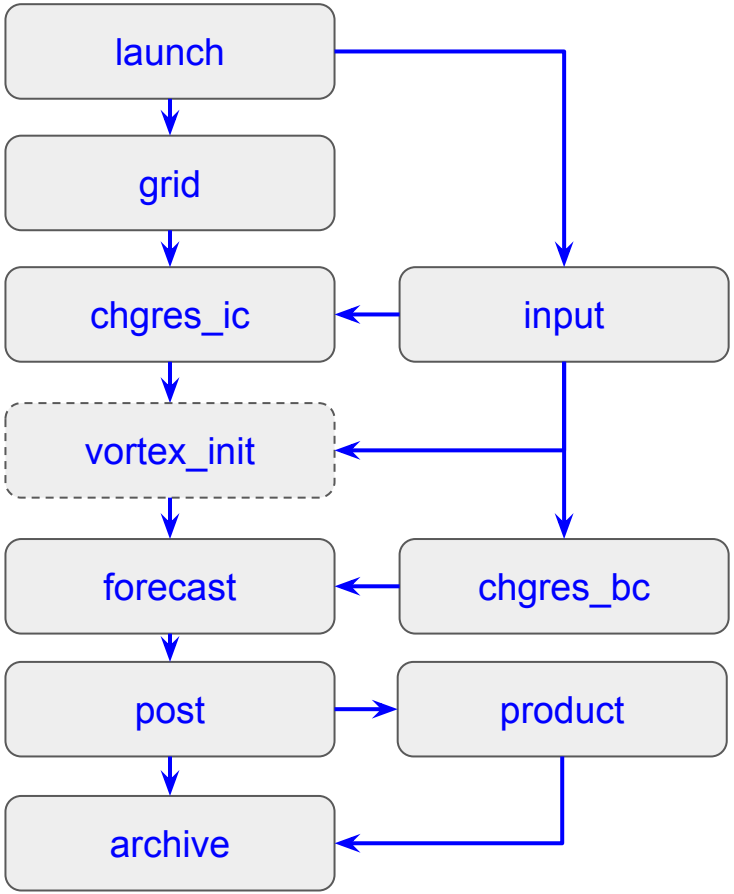






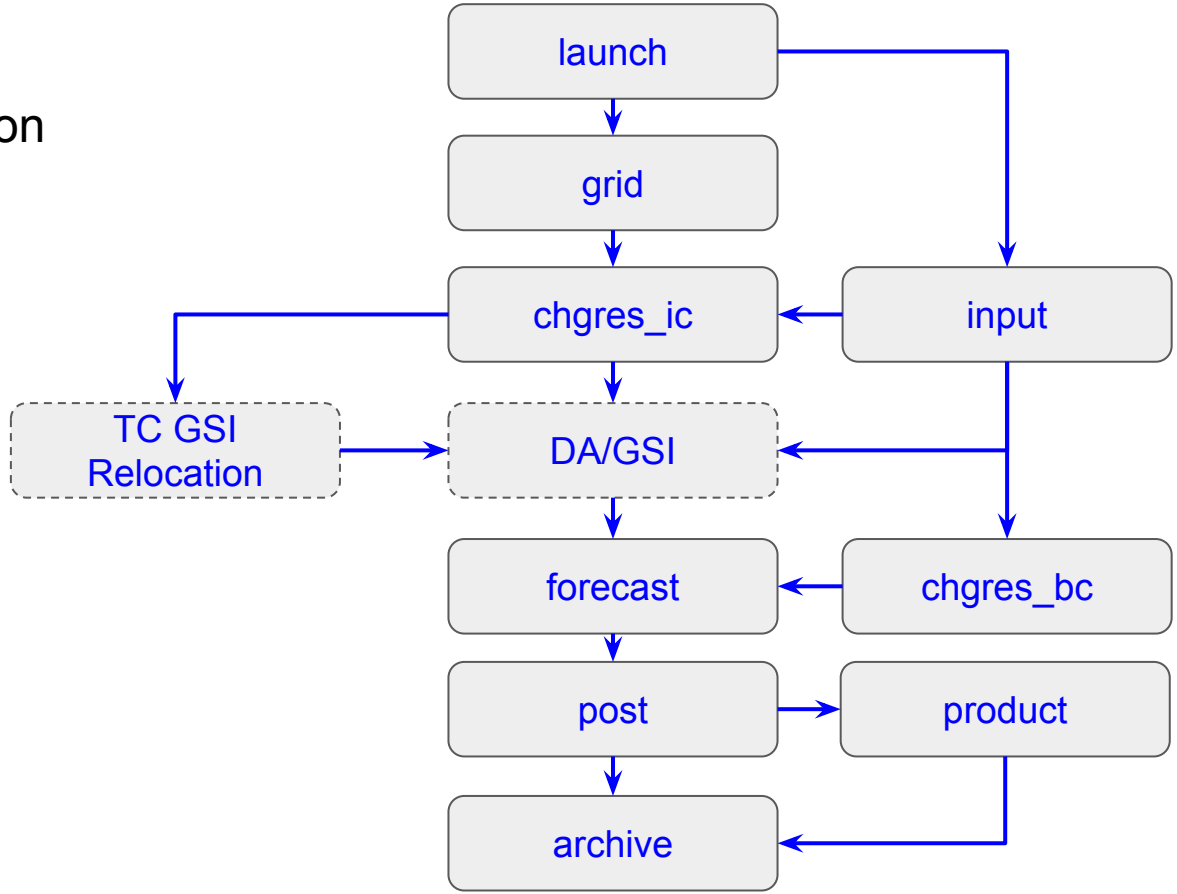
Ongoing HAFS Workflow Development with VI

Workflow with Vortex Initialization (VI)



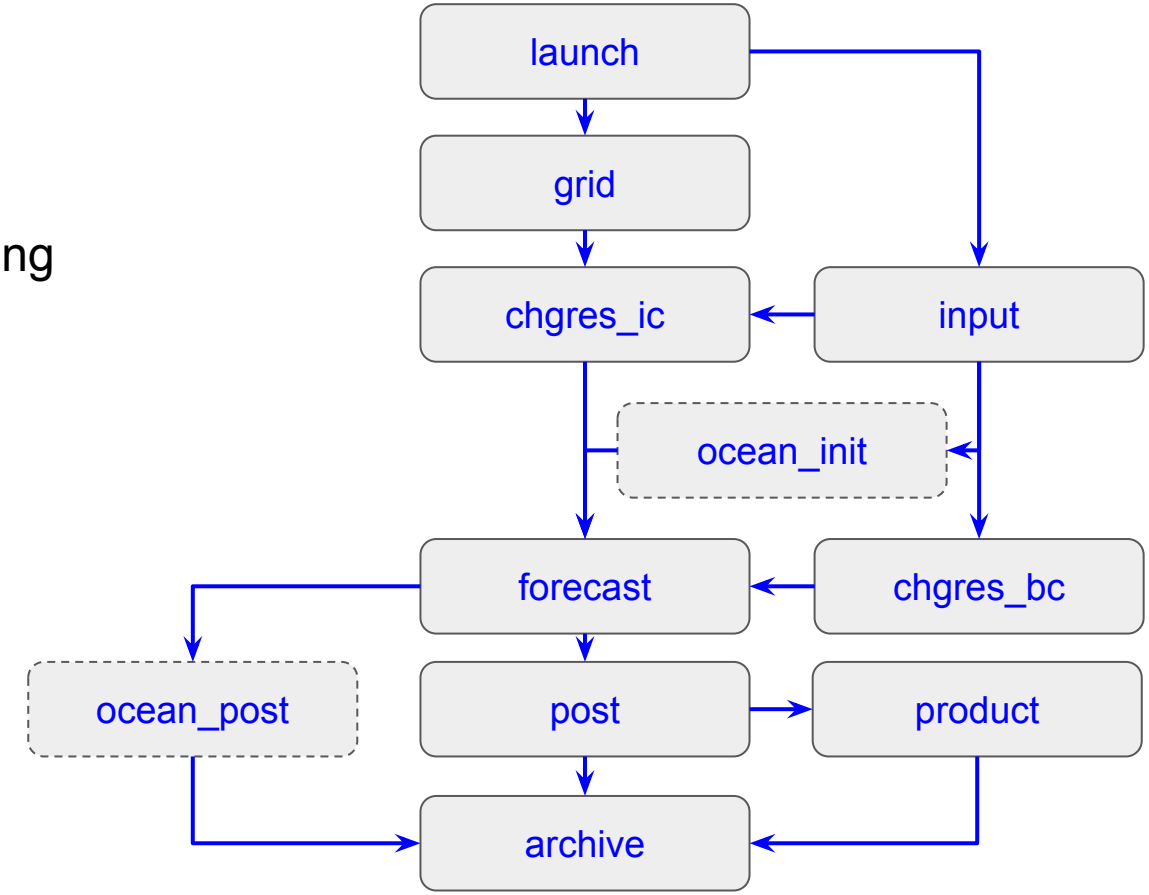
Ongoing HAFS Workflow Development with DA

TC GSI relocation
DA/GSI 3DVAR



Ongoing HAFS Workflow Development with Coupling

HYCOM coupling



Ongoing and Future HAFS Workflow Developments

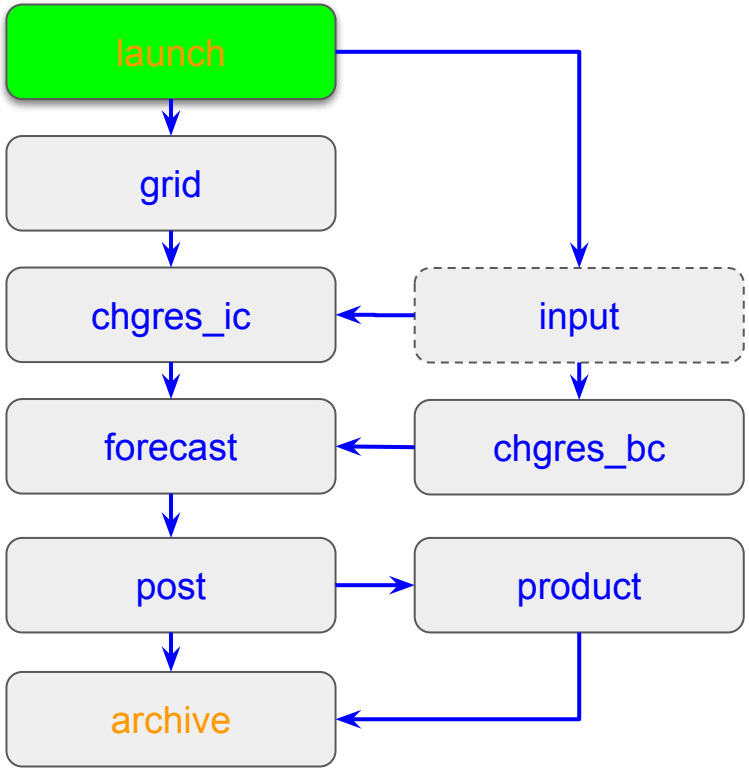
- Explore connecting HAFS with CROW and CIME (with DTC, NESII, NCAR/CGD)
- Add support for multiple, telescope, and moving nesting in the workflow once the nesting capabilities are developed (AOML/HRD, GFDL)
- Further advance the DA capability (with a self-cycled ENSDA system) for HAFS for high-resolution innercore DA
- Enable running HAFS as a coupled atmosphere-wave-ocean modeling system, eventually, as a fully coupled earth system model (with NESII, NCAR/CGD, DTC)
- Add more TC specific products, diagnostics and graphics
- Generalize the HAFS workflow to support global uniform/stretched applications
- Expand to include more workflow level regression tests on multiple platforms
- Explore adding HAFS ensemble forecast capability

Thanks!

Supplemental Materials

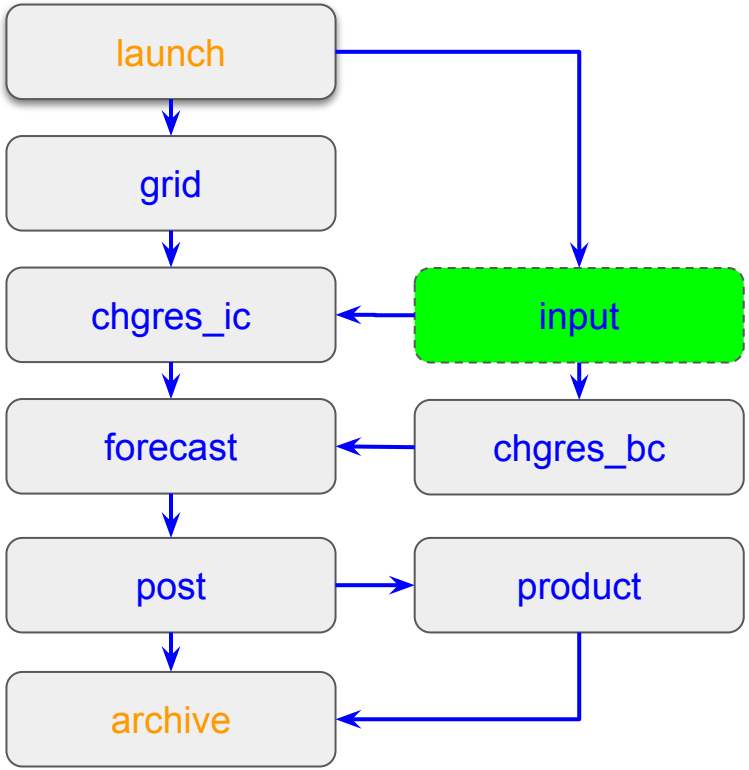
HAFS Launch Job

- Parse the config files and tcvital messages to set up the experiment
  - hafs_basic.conf, hafs.conf, hafs_holdvars.conf, system.conf, and other user specified config files
- Create the work and com directory structure
- Write out the domain center information
- Write out the storm tcvital related files
  - storm1.vitals.*: storm tcvital history files
  - tmpvit/oldvit: the tcvital records for the current/previous cycle
- Generate the storm1.conf file to configure all the downstream Python script based jobs/tasks
- Generate the storm1.holdvars.txt file to pass workflow options to all the downstream Shell script based jobs/tasks.



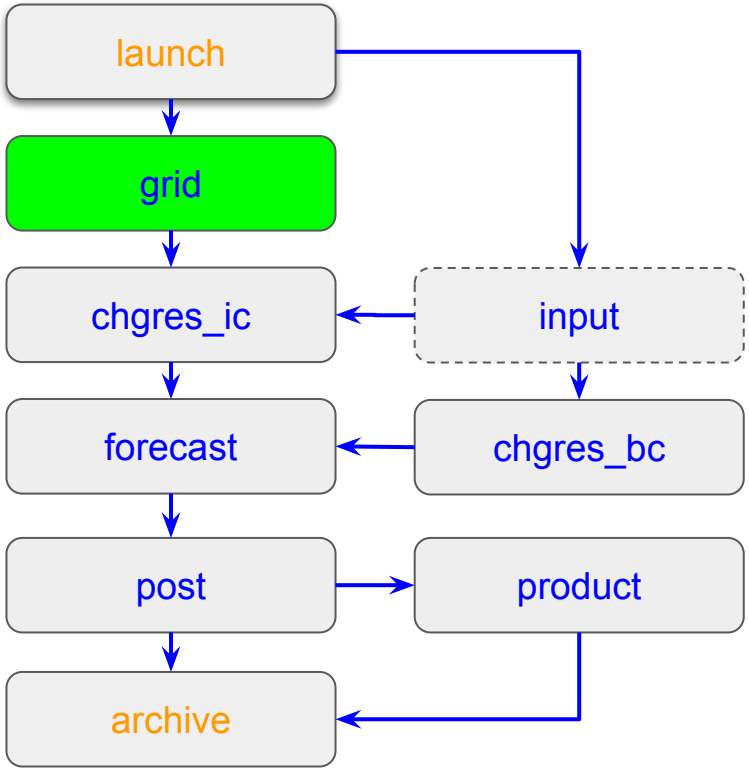
HAFS Input Job

- Prepare/stage the input files needed by the HAFS system
- So far, the HAFS prototype workflow system only needs the NEMSIO and/or grib2 format GFS input files to generate initial conditions and/or lateral boundary conditions (for the regional configuration)
- Currently the input job has not been connected into the HAFS workflow
- In the future, HAFS will need other input files from GFS, GDAS, RTOFS, WW3/multi_1, as well as observational data for data assimilation.



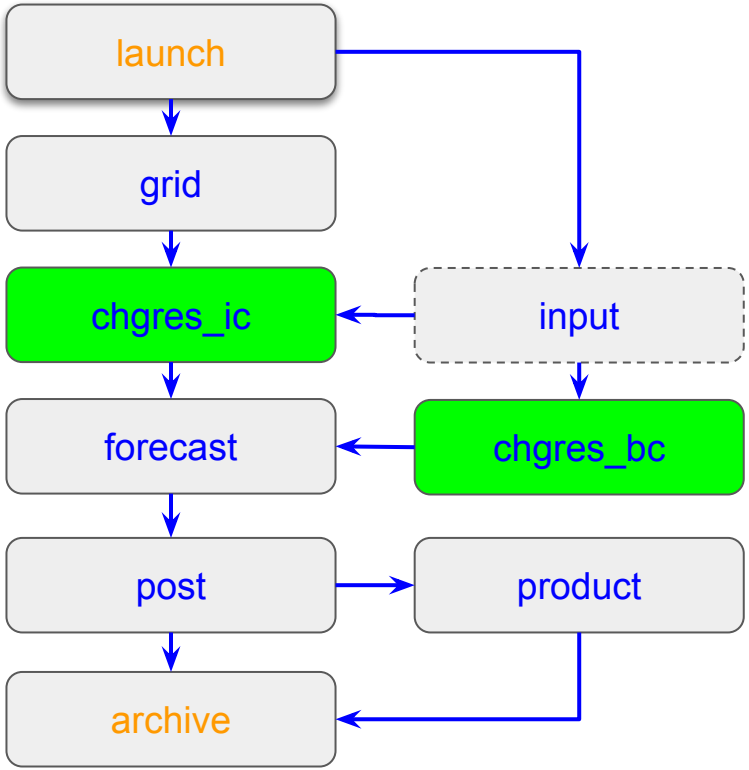
HAFS Grid Job

- Generate the computation grid and mosaic files through (hafs_make_hgrid.x and hafs_make_solo_mosaic.x)
- Prepare the orography files by hafs_orog.x
- For regional configuration,
  - run hafs_filter_topo.x to smooth orography
  - run hafs_shave.x to reduce grid and orography files to required compute size
- Run hafs_sfc_climo_gen.x to generate the surface climatology files
- For the basin-focused configurations, if the pre-generated grid and orography files already exist, the grid job will just copy over those files



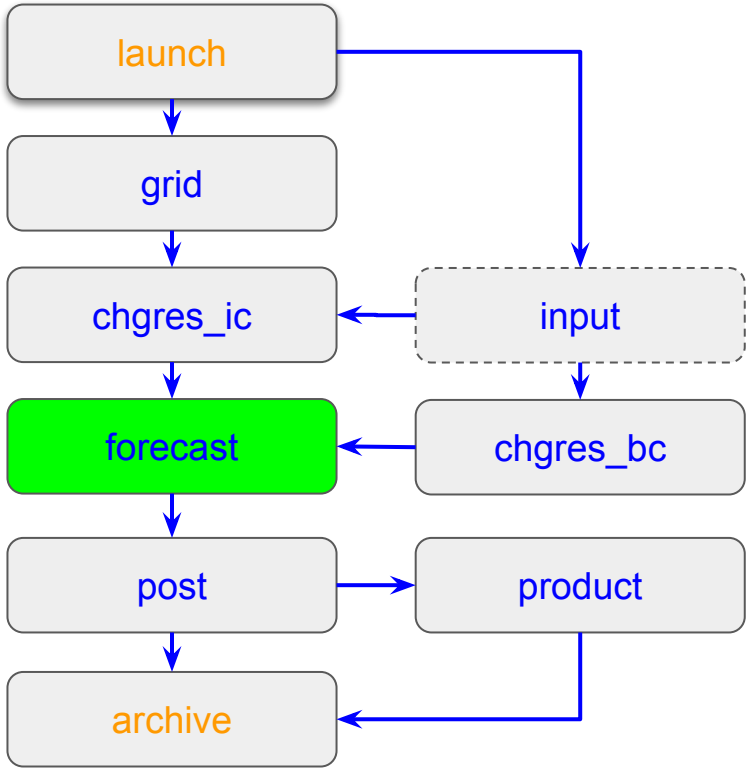
HAFS Chgres Jobs

- For the global-nesting configuration, the chgres_ic job generates initial conditions for the FV3 forecast job.
- For the regional configuration, the chgres_ic/bc jobs generate initial conditions and lateral boundary conditions for the FV3 forecast job.
- The chgres jobs can ingest
  - GFS nemsio format input files
  - GFS grib2 format input files
  - Previous GFS sigio format files
  - FV3 restart or history output netcdf files on native FV3 grid/tiles



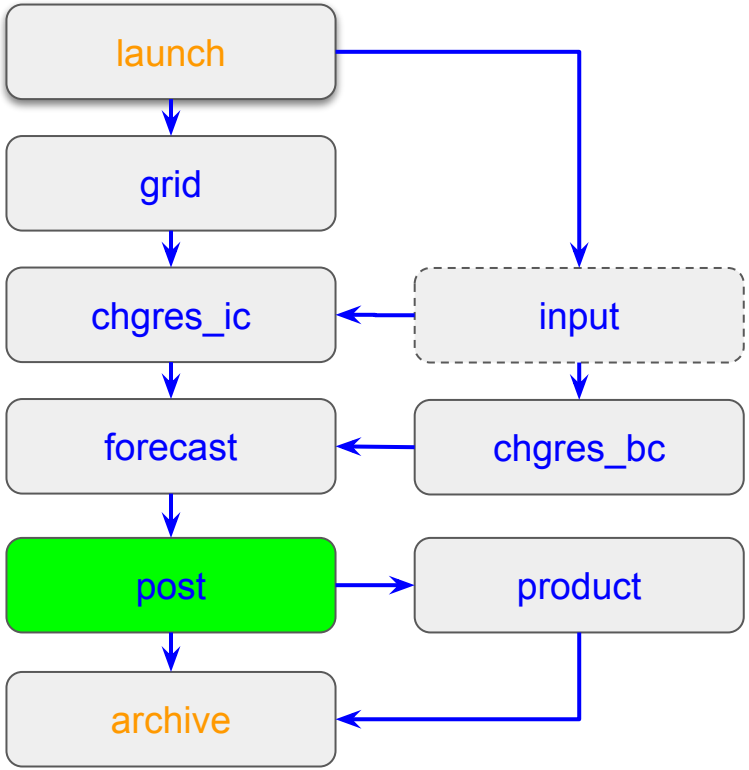
HAFS Forecast Job

- The forecast job takes the IC/BC files prepared by the chgres jobs, and runs the FV3 forecast model.
- The input namelist file(s) and the model configure file configure and control the dynamics, physics and other namelist options used by the forecast model. And the diag_table file controls the output fields from the forecast model.
- Although the FV3 forecast model supports both the FMS IO output method and the write_grid_component output method, the second option is adopted in the HAFS workflow.
- Recent developments for the forecast model's write_grid_component enable outputting a regular/rotated lat-lon/Lambert grid/domain covering the whole computational grid, as well as outputting the native computation grid.



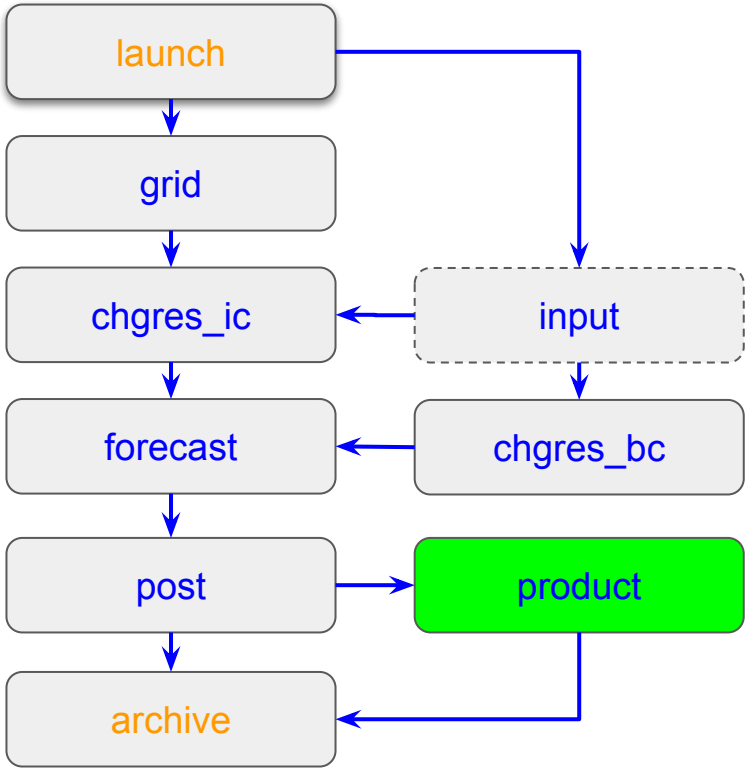
HAFS Post Job

- The post job runs UPP (EMC_post) on the FV3 history output files to produce grib2 format products and deliver them to the com dir.
- Besides, it will also extract the fields needed by the product/tracker job and deliver those files to the intercom dir.
- If the forecast model outputs on a rotated lat-lon grid, wgrib2 (a newer version) will be utilized for the interpolation from the rotated lat-lon grid to a regular lat-lon grid for the final grib2 products and the tracker.
- UPP can now also properly deal with the FV3 output regular lat-lon or rotated lat-lon netcdf files with missing values outside the computation grid.



HAFS Product Job

- The product job runs the GFDL tracker on the generated and extracted fields by the post job, creating the ATCF track files.
- Currently, the product/tracker job can track a single storm as well as multiple storms simultaneously.
- The track file(s) will be delivered to the com directory as well as to a noscrub directory.
- The latest version of the GFDL tracker (obtained from Tim Marchok in September, 2019) was used in the HAFS workflow. And the developer also helped fixed the memory usage issue for the tracker when tracking grib2 format files.



HAFS Archive/Scrub Jobs

- The archive jobs can support archiving the HAFS com directory on the local disk as well as on the HPSS archive.
- If desired, there is also a configure option to enable archiving the FV3 forecast history netcdf format output files.
- The scrub jobs (scrub_work and scrub_com) will delete the corresponding work and com directory to release disk spaces.