Current HAFS Workflow Repository and Developments to Support Research and Real-time Experiments


with other collaborators from

EMC, AOML/HRD, GFDL, DTC, ESRL, NCAR and NSSL
Objectives of the HAFS Workflow Development

- 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.
- HAFS is the UFS hurricane application, providing forecast for hurricane track, intensity, and related effects out to about one week.
- 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, providing support for easier Research to Operation (R2O) and Operation to Research (O2R) transitions.
Current HAFS Workflow Repository and Structure

The authoritative HAFS repository:
- [https://github.com/NOAA-EMC/HAFS](https://github.com/NOAA-EMC/HAFS)
- Mainly hosts the develop, master, release and prod branches
- Support the main development activities, public releases and operational implementations

The community/organizational HAFS forks:
- e.g., [https://github.com/hafs-community/HAFS](https://github.com/hafs-community/HAFS)
- Support community/organizational level collaborations
- Host HAFS related developments for submodule repositories/forks

The personal HAFS forks:
- Developer’s forks for individual feature (or capability) development
- New developments/features can be integrated back into the authoritative repository through GitHub Pull Requests
Current HAFS Subcomponents/Submodules

HAFS (ufs-hafs-app)
- UFS_UTILS
- ufs-weather-model (ufs-hafs-model)
- EMC_Post
- gfdl-tracker
- ProdGSI

- NEMS
- FV3 (fv3atm)
- FMS
- ccpp-framework
- ccpp-physics
- stochastic_physics
- WW3
- GSI-fix
- GSI-libs

- NCEPLIBS-pyprodutil
- GFDL_atmos_cubed_sphere

GitHub repo
VLAB repo
Currently
5 direct subcomponents
16 submodules in total
The GitFlow Rational

- **Code Management of Repositories under EMC management using GitFlow**
- **HAFS GitFlow Rational**
- **A successful Git branching model** By Vincent Driessen

Branch naming convention:

- **develop**: main development trunk
- **master**: mature part of the development
- **release**: for public releases
- **feature**: for individual feature/capability development
- **hotfix**: for bug fixes
- **support**: for supporting real-time experiments and retrospective tests
- **prod**: for supporting production implementation
HAFS Workflow Development Requirements

- **NCO Implementation Standards** and NCEP EE2 compliant for easy operational transition and implementation
- Easy to learn and use to promote community modeling and development
- Specialized for hurricane specific applications
  - Support regional, global-nesting, and global (uniform/stretched) configurations, as well as the storm-following moving nesting capability
  - Support event-triggered configurations for forecasting active storms and continuously-cycled configuration for TC genesis forecasting
  - Include VI for warm-starting and cycling of the storm and data assimilation techniques for high-resolution innercore DA
  - Support hurricane dynamics and physics optimizations
  - Support air-sea coupling (eventually, earth system coupling)
  - Generate TC specific diagnoses and products
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 storm-triggered and continuously-cycled configurations

- Supports both standalone regional and global-nesting configurations

- Supports both basin-focused (static) and storm-focused (relocatable) configurations

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) for a CONUS domain
Currently, the HAFS workflow is a Rocoto based workflow. It can be easily converted/ported to support EcFlow for future production implementation.

The workflow driver script (rocoto/run_hafs.py) automatically generates the workflow definition files and submits the HAFS workflow jobs/tasks.

The HAFS configuration/launch system is adapted/improved from the HWRF/HMON system, and it supports the various different HAFS configurations.

Currently both Python and Shell based scripts are utilized in the workflow components.
A Quick Start for HAFS Users

A. Clone and checkout

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 HAFS/parm

# Florence

Note: a detailed HAFS developer guide can be found here.
Ongoing and Future HAFS Workflow Developments

- The HAFS workflow was used in the 2019 HFIP HAFS real-time experiments
- Add the vortex initialization (VI) capability in the HAFS workflow for storm cycling
- Establish/advance the DA capability for HAFS for high-resolution innercore DA
- Enable outputting multiple model domains/grids from the forecast model
- Explore the capability of in-line post (UPP) with the forecast model
- Test/explore the capability of TC genesis tracking and native grid storm tracking (GDFL)
- Add more TC specific products, diagnoses and graphics
- Generalize the HAFS workflow to support global uniform/stretched applications
- Explore connecting HAFS with CROW and CIME (with DTC, NESII, NCAR/CGD)
- Establish regression tests on multiple platforms
- Add support for multiple, telescope, and moving nesting in the workflow once the nesting capabilities are developed (AOML/HRD, GFDL)
- Enable running HAFS as a coupled atmosphere-wave-ocean modeling system, eventually, as a fully coupled earth system model (with NESII, NCAR/CGD, DTC)
- Explore adding HAFS ensemble forecast capability