A Community Workflow for the Stand-Alone Regional (SAR) Configuration of the FV3

29 April 2020

Jeff Beck[1,2], Gerard Ketefian[1,3], Curtis Alexander[1], Larissa Reames[4], George Gayno[5,6], Dominikus Heinzeller[1,3], Linlin Pan[1,3], Tanya Smirnova[1,3], James Purser[5,6], Dusan Jovic[5,6], Thomas Black[6], James Abeles[5,6], Jamie Wolff[7], Laurie Carson[7], Julie Schramm[7], Michael Kavulich[7], Jacob Carley[6], Benjamin Blake[5,6], and Todd Arbetter[7]

[1]NOAA/ESRL/GSL, [2]CIRA, [3]CIRES, [4]CIMMS/NOAA/NSSL

[5]IM Systems Group, [6]NOAA/NCEP/EMC, [7]NCAR

Overview of the FV3-SAR Community Workflow

- With UFS community-oriented goals in mind, FV3-SAR workflow development began by incorporating internal requirements for use in GSL, EMC, and DTC with a user-friendly, modular, and expandable framework for the broader community (both research and operations)
- Need to provide maximum flexibility for users in many areas of research:
  - Domain/grid and write component (output) grid options
  - IC/LBC external model options (FV3GFS, GSMGFS, RAP, HRRR, NAM) with different file format options (grib2, nemsio)
  - Variable LBC interval and forecast length
  - Multiple computing platforms, including non-NOAA HPC
  - Mechanism to easily retrieve and build all necessary code from external repositories (model, pre-processing, and post-processing)
  - Support for different CCPP physics packages
  - Allow for both retrospective and real-time applications
  - Develop a library of unified bash-based utilities that are used across the experiment generation/workflow/tasks
- The goal was to create an automatically generated, experiment-specific Rocoto workflow file that calls each task shell script, while retaining the option to run scripts from the command line (e.g., without Rocoto)
- Therefore, the FV3-SAR community workflow represents an experiment generation system, the shell scripts that handle each component of the full end-to-end system, and the user-defined workflow management XML file

Setting Up and Generating the FV3-SAR Community Workflow

- A workflow configuration file (`config.sh`) is used to specify the experiment parameters, unspecified parameters are set to default values
- Sample `config.sh` files are provided for various configurations (e.g., for different domains/grids, physics suites, and/or external models); these serve as a starting point for users to customize their own experiments
- Once the configuration file is finalized, the user runs the `generate_FV3SAR_wflow.sh` script to create the experiment directory, Rocoto workflow XML file, and automatically generate a cron job for the experiment
- During the experiment generation step, a workflow variable definitions file (`var_defns.sh`) containing all user-defined and derived workflow variables is generated and copied to the experiment directory. This file is then sourced by each workflow task in the end-to-end system.
- When the experiment configuration is generated, error checks are run to ensure user-defined variables, FV3 MPI layout, and write component (output) options are valid
- Error messages during this step (and in each shell script of the end-to-end system) are designed to be verbose and specific to provide users with necessary information to pinpoint the source of most problems

FV3-SAR Community Workflow Tasks

Run once per expt. (can use pre-generatedfiles on disk)

1. **`make_grid`** – Generates grid files (GFDL or JP grid format; latter is preferred)
2. **`make_orog`** – Generates filtered orography files
3. **`make_sfc_climo`** – Generates surface climatology files (used if fields are not available in external model output)

Run once for each cycle

4. **`get_extrn_ics`** – Retrieves output files (staged on disk or NOAA HPSS) from the external model needed for generating ICs, surface fields, and the 0-th hour LBC)
5. **`get_extrn_lbcs`** – Retrieves output files (staged on disk or NOAA HPSS) from the external model needed for generating LBCs
6. **`make_ics`** – Creates ICs on the native FV3-SAR grid (including surface fields and the 0-th hour LBC)
7. **`make_lbcs`** – Creates LBCs on the FV3-SAR grid for each boundary condition interval (e.g., every 1, 3, 6 hours)
8. **`run_fcst`** – Runs a forecast (cycle) with the FV3-SAR
9. **`run_post`** – Processes write-component forecast output files and calculates derived fields through UPP, generates grib2 files

FV3-SAR Community Workflow Status and Details

- The workflow has been tested on multiple supercomputing platforms, including NOAA HPC (Hera, Jet) and is being tested in DC (WCOSS), at OU/NSSL (Stampede/Odin), and on NCAR's Cheyenne; MSU's Orion is next
- Supports FV3GFS, GSMGFS, RAP, HRRR, or NAM external model data for IC/LBCs and can read in grib2, nemsio, and netcdf data formats
- Compatible with the Common Community Physics Package (CCPP) and supports various versions of the GFS, RAP/HRRR, and RRFS-like physics suites
- Users can generate their own domains, or select from pre-defined domains, including coarse- and high-resolution versions of North American, CONUS, Alaska, Hawaii, Guam, Puerto Rico, and HAFSv0.A domains

```
.....

<workflow realtime="F" scheduler="&SCHED;" cyclethrottle="20">

<cycledef>00 &HH; &DD; &MM; &YYYY; *</cycledef>

 <log>
  <cyclestr>&LOG_DIR;/FV3_@Y@m@d@H.log</cyclestr>
 </log>

 <task name="make_grid" maxtries="3">

  <command>&USHDIR;/make_grid_orog.sh</command>
  <nodes>&PROC_MAKE_GRID_OROG;</nodes>
  <jobname>make_grid_orog</jobname>
  <join><cyclestr>&LOG_DIR;/make_grid_orog_@Y@m@d@H.log</cyclestr></join>

 </task>

 <task name="make_orog" maxtries="3">

  <command>&USHDIR;/get_GFS_files.sh</command>
  <nodes>&PROC_GET_GFS_FILES;</nodes>
  <jobname>get_GFS_files</jobname>
  <join><cyclestr>&LOG_DIR;/get_GFS_files_@Y@m@d@H.log</cyclestr></join>

  <dependency>
    <taskdep task="make_grid"/>
  </dependency>

 </task>

.....
```

FV3-SAR Community Workflow Merge with EMC

- Held code sprint last summer between GSL, EMC, NSSL, and DTC to formalize a plan toward a workflow merge between the operational EMC and community workflows

- Defined deliverables necessary to incorporate EMC/NCO requirements into the community workflow:
  - Split workflow tasks in j-job and ex-script layers
  - Move module loads into separate pre-jobs in workflow
  - Use NCO-compliant experiment directory structure
  - Use NCO-compliant variable names
  - Migrate to EMC build script

- Goal was to create a single, flexible, expandable experiment and workflow generation tool that can be used in both operational and research environments

- Final merged workflow currently being tested on WCOSS and Hera; can now run in either NCO or community mode

- Contacted NCO to inform them of merged FV3-SAR community workflow and to facilitate future testing related to implementation of the RRFS into operations

Accessing the FV3-SAR Community Workflow

- The community workflow code is available to the public in the `regional_workflow` repository on GitHub:

  https://github.com/NOAA-EMC/regional_workflow

- To clone the repository and checkout the `community_develop` branch, use the following command:

  git clone -b community_develop https://github.com/NOAA-EMC/regional_workflow.git

- Documentation currently exists as a Google Doc within NOAA, but will be provided within the repository as a users' guide with the official Short-Range Weather (SRW) application release this fall
  - For access now, please send an email to me (jeff.beck@noaa.gov) or Gerard Ketefian (gerard.ketefian@noaa.gov), and we will be happy to provide a copy of the workflow instructions

- Gitflow (forking, required testing, pull requests, and code review via GitHub) being used to provide the opportunity for contributions from anyone in the community
  - Currently provide 15 end-to-end workflow regression tests in the regional_workflow repository
  - Contributions have been made by users at GSL, PSL, DTC, EMC, UM, OU, and NSSL

Planning for the First UFS SRW App Release

- First UFS Short-Range Weather (SRW) application release currently planned for the beginning of November 2020
  - The community workflow will be included as the experiment generation and end-to-end management system for the SRW Application
  - Multiple focus teams have been set up (workflow, code, build, documentation, support, etc.)
  - A SRW app forum is now available on the UFS Users' Support website

Ongoing development work that will be incorporated into the community workflow prior to the release:

- Finalize new wrapper scripts for users without the Rocoto workflow manager
  - One script per workflow task
  - Options exist to run with or without a batch system
- Transition to a more robust and portable build system (cmake) for each component repository
- Expedite process to port and test the community workflow on Cheyenne, Orion, Odin, and WCOSS
- Provide an option to allow users to source grib2 external model data from online (e.g., NOMADS)
- Include option to blend external model fields near the edges of the integration domain (code now available from EMC) to minimize spurious standing waves
- Migrate Fortran namelist/XML generation and template management to Python-based libraries: yaml/jinja/f90nml

Future Development Work

- Incorporate DA into workflow (leverage previous work at EMC/GSL):
  - Include new workflow task for GSI
  - Introduce capability to start model from restart files

- Introduce user-configurable options for the vertical coordinate:
  - Model top pressure
  - Modify the transition level for transition from sigma to isobaric coordinates
  - Vertical distribution of coordinate surface

- Provide an ensemble configuration for the workflow, including the option to run both ad-hoc and SPP versions of stochastic physics

- Support ecFlow as a workflow manager option in addition to Rocoto

- Incorporate verification into the workflow; introduce METplus use cases for point and grid-based verification

Thank you!

Questions?