CCPP Training
College Park, MD, March 12-13, 2019

CCPP Code Management

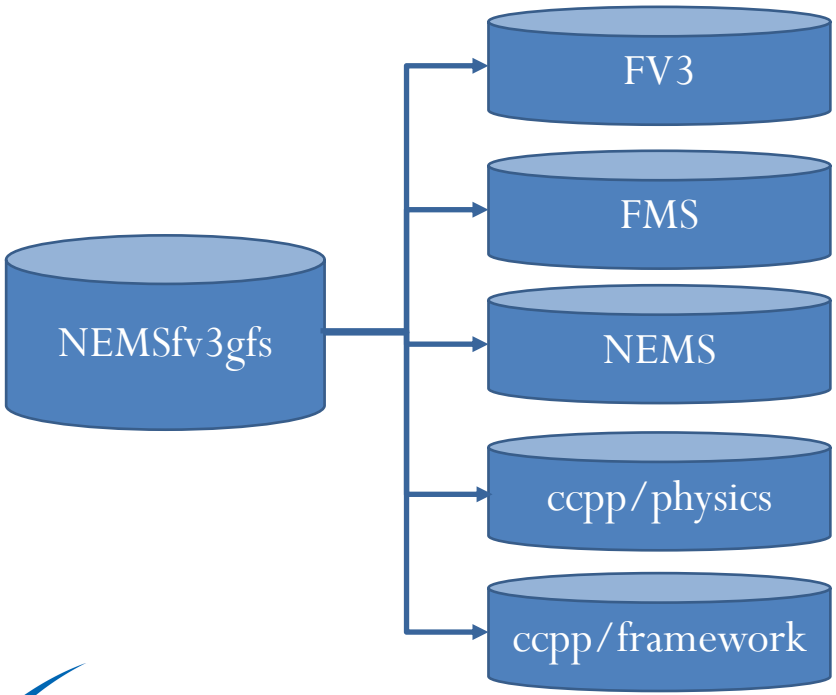
Laurie Carson
Global Model Test Bed

CCPP code management

- Repository Structure, Submodules
- Development workflow
- Regression tests, code review

Code repositories

- The repository structure for CCPP development in NEMSfv3gfs mirrors the Vlab repository structure, with the addition of the CCPP repositories



Repository (GMTB development version)	Branch name
https://github.com/NCAR/NEMSfv3gfs	**gmtb/ccpp**
https://github.com/NCAR/FV3	gmtb/ccpp
https://github.com/NCAR/ccpp-physics	master
https://github.com/NCAR/ccpp-framework	master
https://github.com/NCAR/NEMS	gmtb/ccpp
https://github.com/NCAR/FMS	GFS-FMS

Repository Structure, Submodules

- Users have read-only access to these authoritative repositories.
- Some of these repositories are public (no GitHub account required) and some are private.
- The public repositories are ccpp-framework, ccpp-physics, FMS
- The private repositories (NEMSfv3gfs, FV3 and NEMS) require access - please send a request and your GitHub username to gmtb-help@ucar.edu
- At this time, the content of the NCAR repositories NEMSfv3gfs, FV3 and NEMS are synchronized regularly with CCPP branches of the corresponding repositories in VLab

Development workflow

- Step-by-step instructions yesterday
- A configuration script is available to automate some of these steps!

https://github.com/climbfuji/domtools, master branch, NEMSfv3gfs/checkout/checkout_nemsfv3gfs.py

- Edit the config file, and run this script to ***clone*** the NCAR repositories and set up your ***development forks***, i.e.

```
[ccpp-framework]
add_own_fork = True
fork = NCAR
branch = master
```

Github forking workflow

Basic steps

1. Clone the authoritative repository locally
2. Create a local branch, add development, complete testing
3. Push local branch to your personal fork
4. Open a PR (pull request) to request a code review and merge with the authoritative repository

Regression Tests

Use the NEMSfv3gfs regression test script, rt.sh

1. Run standard, non-CCPP regression tests using Intel 15 in PROD mode against official baseline maintained by EMC.

   ./rt.sh -f 2>&1 | tee rt_full.log

2. Create your own baseline using the non-CCPP code using Intel 18 in REPRO mode.

   ./rt.sh -l rt_ccpp_ref.conf -c fv3 2>&1 | tee rt_ccpp_ref_create.log

3. Verify the various CCPP builds against own baseline using Intel 18 in REPRO mode.

   ./rt.sh -l rt_ccpp_hybrid.conf -m 2>&1 | tee rt_ccpp_hybrid.log

   ./rt.sh -l rt_ccpp_standalone.conf -m 2>&1 | tee rt_ccpp_standalone.log

   ./rt.sh -l rt_ccpp_static.conf -m 2>&1 | tee rt_ccpp_static.log

4. Create own baseline using the non-CCPP code using Intel 15 in PROD mode with selective lowering of the PROD compiler optimization flags.

./rt.sh -l rt_ccpp_ref_for_acceptance.conf -c fv3 2>&1 | tee rt_ccpp_ref_for_acceptance create.log

5. Verify the static CCPP build against own baseline using Intel 15 in PROD mode with selective lowering of the PROD compiler optimization flags.

./rt.sh -l rt_ccpp_static_for_acceptance.conf -m 2>&1 | tee rt_ccpp_static_for_acceptance create.log

NOTE: As the transition of CCPP into the VLab master proceeds, these procedures will change.

Code review, governance

- Once a PR is open and marked as ready-to-merge:
  - The code review must be completed and the changes approved by at least one of the CODEOWNERS.
  - These are added automatically as reviewers when a pull request is made.
  - Additional reviewers can be added as needed.
  - Typically, the code will be merged by one of the CODEOWNERS after the review is completed and the changes are approved.
- The code review process is designed to make sure that the code submitted complies with the CCPP requirements (coding standards, …), that the changes are understandable and that the regression tests pass as described.
- At this time, the governance lies with GMTB for all repositories except ccpp-framework.
- ccpp-framework governance is shared between GMTB and NCAR/CGD, and code review and approval is required by GMTB and NCAR/CGD.

Wrap up

- Use github.com to develop, debug, test new capabilities
- Code review, approval and regression testing is required
- Integration with EMC/Vlab will modify these steps in the coming weeks and months