Build the UFS Short-Range Weather App v1.0

Linlin Pan[1,2,3], Jeff Beck[1,3,,4], Gerard Ketefian[1,2,3], Julie Schramm[3,5], Laurie Carson[3,5], Mike Kavulichi[3,5], and Jamie Wolff[3,5]

[1]NOAA/GSL, [2] University of Colorado/CIRES, [3] DTC, [4]Colorado State University/CIRA, [5]NCAR

UFS SRW App Training

21 September 2021

Developmental Testbed Center

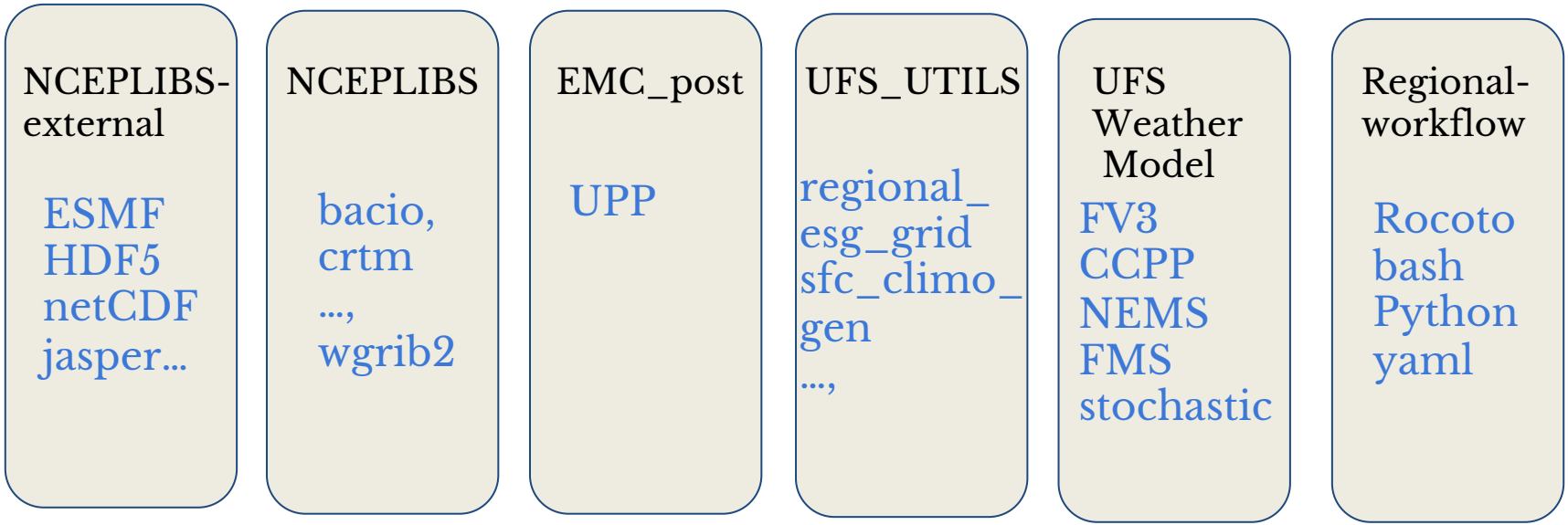
Build Overview

- Documentation for the UFS SRW App can be found at: https://ufs-srweather-app.readthedocs.io/en/ufs-v1.0.1/Introduction.html
- These slides follow the build steps in the Workflow Quick Start section: https://ufs-srweather-app.readthedocs.io/en/ufs-v1.0.1/Quickstart.html

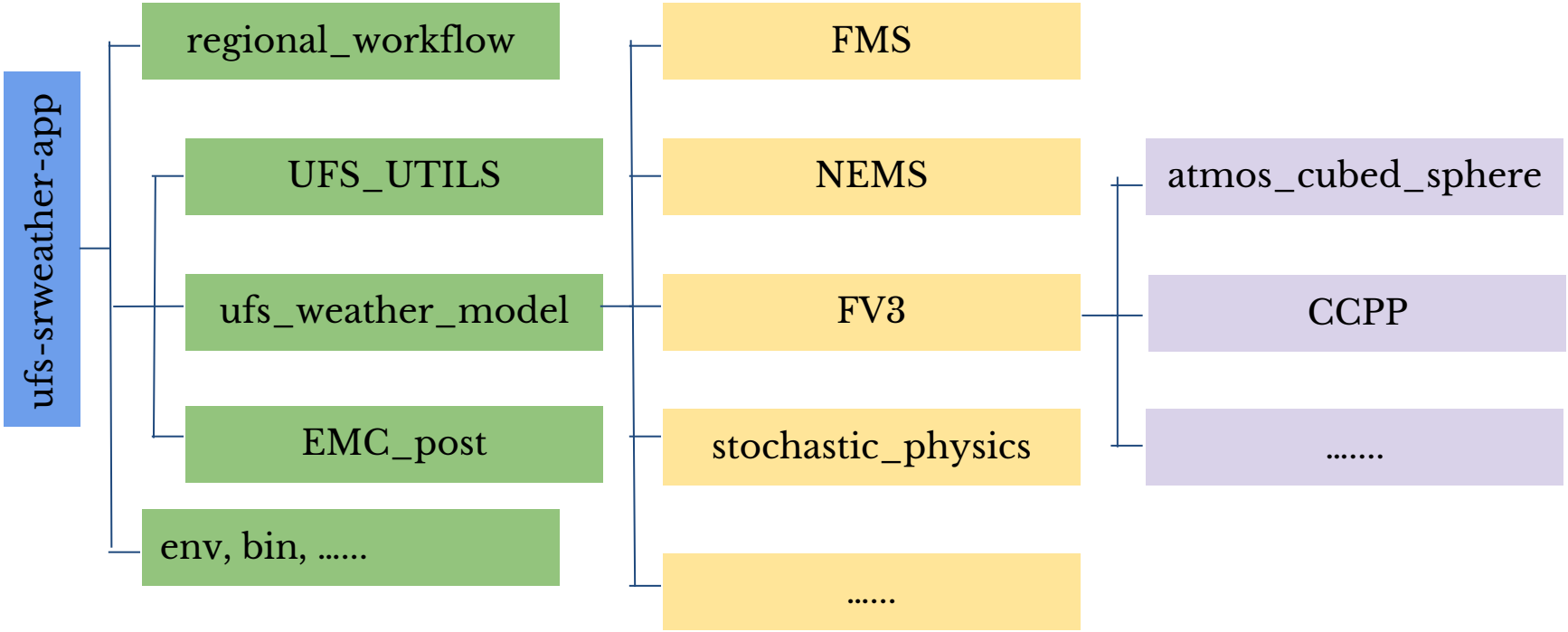
Components of UFS SRW App

- The major components, some with their own subcomponents:
  - **NCEPLIBS-external**
  - **NCEPLIBS**
  - **UFS Weather Model**
  - **UFS Utils (UFS_UTILS)**
  - **UPP (EMC_post)**
  - **Regional Workflow**
- All of these components are necessary to build the SRW App.

Overview of UFS SRW App Components



Code hierarchy



Steps to build the App

The steps to build the regional workflow covered here are:

1. Clone the code
2. Set up the environment
3. Build the executables

Step 1: Clone the code

```
git clone -b ufs-v1.0.1 https://github.com/ufs-community/ufs-srweather-app.git

cd ufs-srweather-app

./manage_externals/checkout_externals
```

This will clone the **ufs-v1.0.1** branch of the **ufs-srweather-app** repository and check out all of the externals and sub-modules:

Processing externals description file : Externals.cfg
Checking status of externals: regional_workflow, ufs_utils, ufs_weather_model, emc_post

Step 2: Set up the environment (1)

Prior to building the model, load the modules and set the proper environment variables

```
ls -1 env/build*
env/build_cheyenne_gnu.env
env/build_cheyenne_intel.env
env/build_gaea_intel.env
env/build_hera_intel.env
env/build_jet_intel.env
env/build_orion_intel.env
env/build_wcoss_cray_intel.env
env/build_wcoss_dell_p3_intel.env
```

Step 2: Set up the environment (2)

Prior to building the model, load the modules and set the proper environment variables (source build_cheyenne_intel.env)

```
module purge
module load ncarenv/1.3
module load intel/19.1.1
module load mpt/2.19
module load ncarcompilers/0.5.0
module load cmake/3.16.4

module use /glade/p/ral/jntp/GMTB/tools/NCEPLIBS-ufs-v2.0.0/intel-19.1.1/mpt-2.19/modules
module load NCEPLIBS/2.0.0

export CMAKE_C_COMPILER=mpicc
export CMAKE_CXX_COMPILER=mpicxx
export CMAKE_Fortran_COMPILER=mpif90
export CMAKE_Platform=cheyenne.intel
```

Step 2: Set up the environment (3)

Prior to building the model, load the modules and set the proper environment variables (source build_cheyenne_intel_tcsh.env)

cp build_cheyenne_intel.env build_cheyenne_intel_tcsh.env

change

```
export CMAKE_C_COMPILER=mpicc
export CMAKE_CXX_COMPILER=mpicxx
export CMAKE_Fortran_COMPILER=mpif90
export CMAKE_Platform=cheyenne.intel
```

to

```
setenv CMAKE_C_COMPILER mpicc
setenv CMAKE_CXX_COMPILER mpicxx
setenv CMAKE_Fortran_COMPILER mpif90
setev CMAKE_Platform cheyenne.intel
```

Step 3: Build the SRW App executables

Under the directory of ufs-srweather-app

```
mkdir build
cd build
```

```
cmake .. -DCMAKE_INSTALL_PREFIX=..
make -j 4  >& build.out &
```

ls bin

chgres_cube global_equiv_resol make_solo_mosaic NEMS.exe
regional_esg_grid shave filter_topo make_hgrid ncep_post
orog sfc_climo_gen vcoord_gen

Executable Name Description

- **chgres_cube:** Reads in raw external model (global or regional) and surface climatology data to create initial and lateral boundary conditions
- **filter_topo**: Filters topography based on resolution
- **global_equiv_resol:** Calculates a global, uniform, cubed-sphere equivalent resolution for the regional Extended Schmidt Gnomonic (ESG) grid
- **make_solo_mosaic:** Creates mosaic files with halos
- **ncep_post:** Post-processor for the model output
- **NEMS.exe:** UFS Weather Model executable
- **orog:** Generates orography, land mask, and gravity wave drag files from fixed files
- **regional_esg_grid:** Generates an ESG regional grid based on a user-defined namelist
- **sfc_climo_gen:** Creates surface climatology fields from fixed files for use in `chgres_cube`
- **shave:** Shaves the excess halo rows down to what is required for the LBCs in the orography and grid files
- **vcoord_gen:** Generates hybrid coordinate interface profiles