Software Updates and Plans

Dave Gill

OpenMP Parallelism is Pretty Good

January 2000 Benchmark

Radiation Day

Core Count		SM Efficiency	DM Efficiency
1	74x61	100	100
2	74x31	97	100
4	37x31	93	97
8	37x16	91	96
16	19x16	65	85

Avg 5.76 s

Std 0.019 s

n = 17

Radiation Night

Core Count		SM Efficiency	DM Efficiency
1	74x61	100	100
2	74x31	97	100
4	37x31	93	95
8	37x16	88	92
16	19x16	59	83

Avg 2.16 s

Std 0.005 s

n = 24

Not Radiation Timestep

Core Count		SM Efficiency	DM Efficiency
1	74x61	100	100
2	74x31	94	97
4	37x31	84	80
8	37x16	68	71
16	19x16	43	52

Avg 0.39 s

Std 0.012 s

n = 432

OpenMP Parallelism is Pretty Good

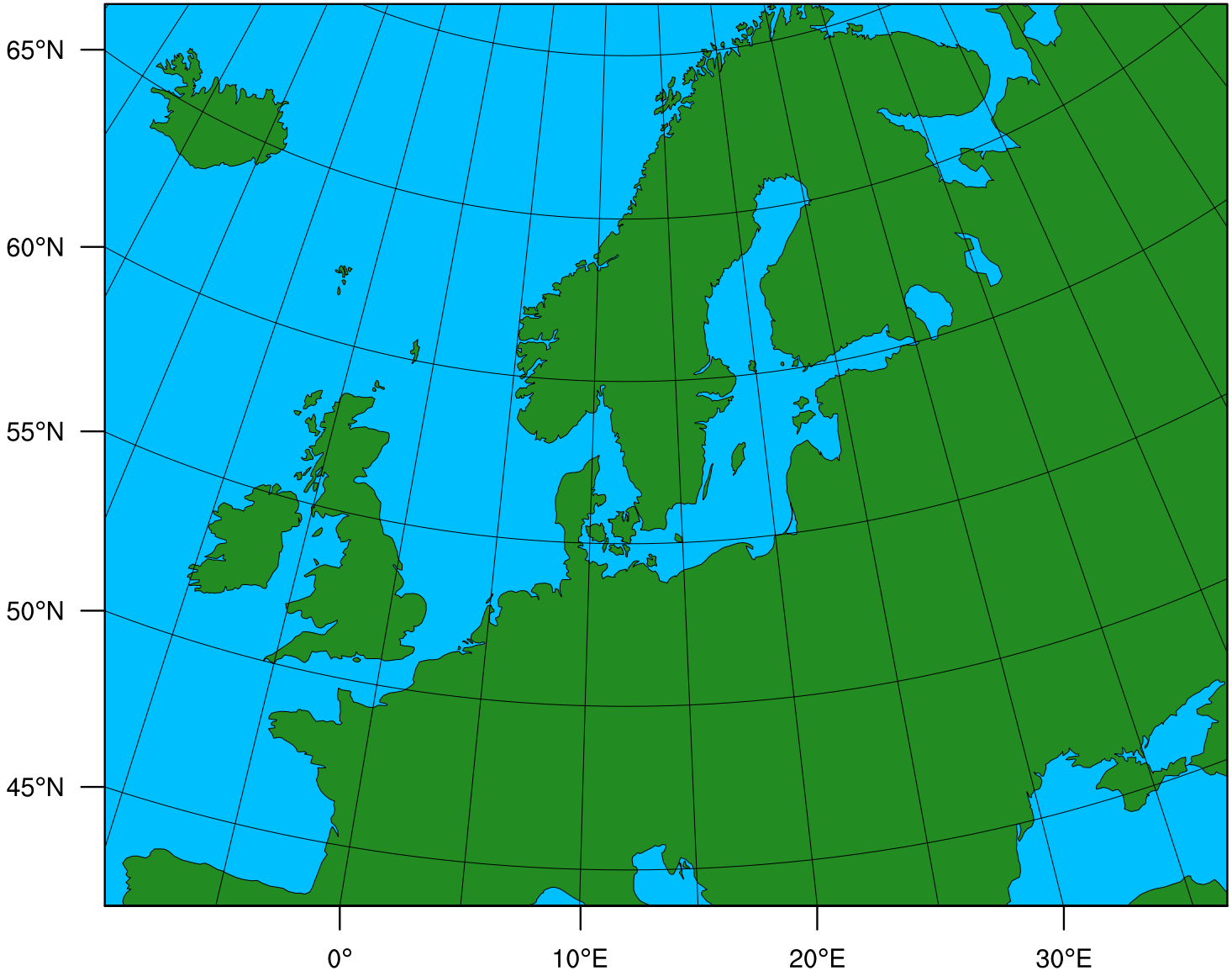
- Mods to v3.9 fix “oops” in tile computations
- On NCAR supercomputer cheyenne, use **`omplace`** to pin threads to the same core
- Google “NCAR cheyenne OpenMP” for example batch scripts

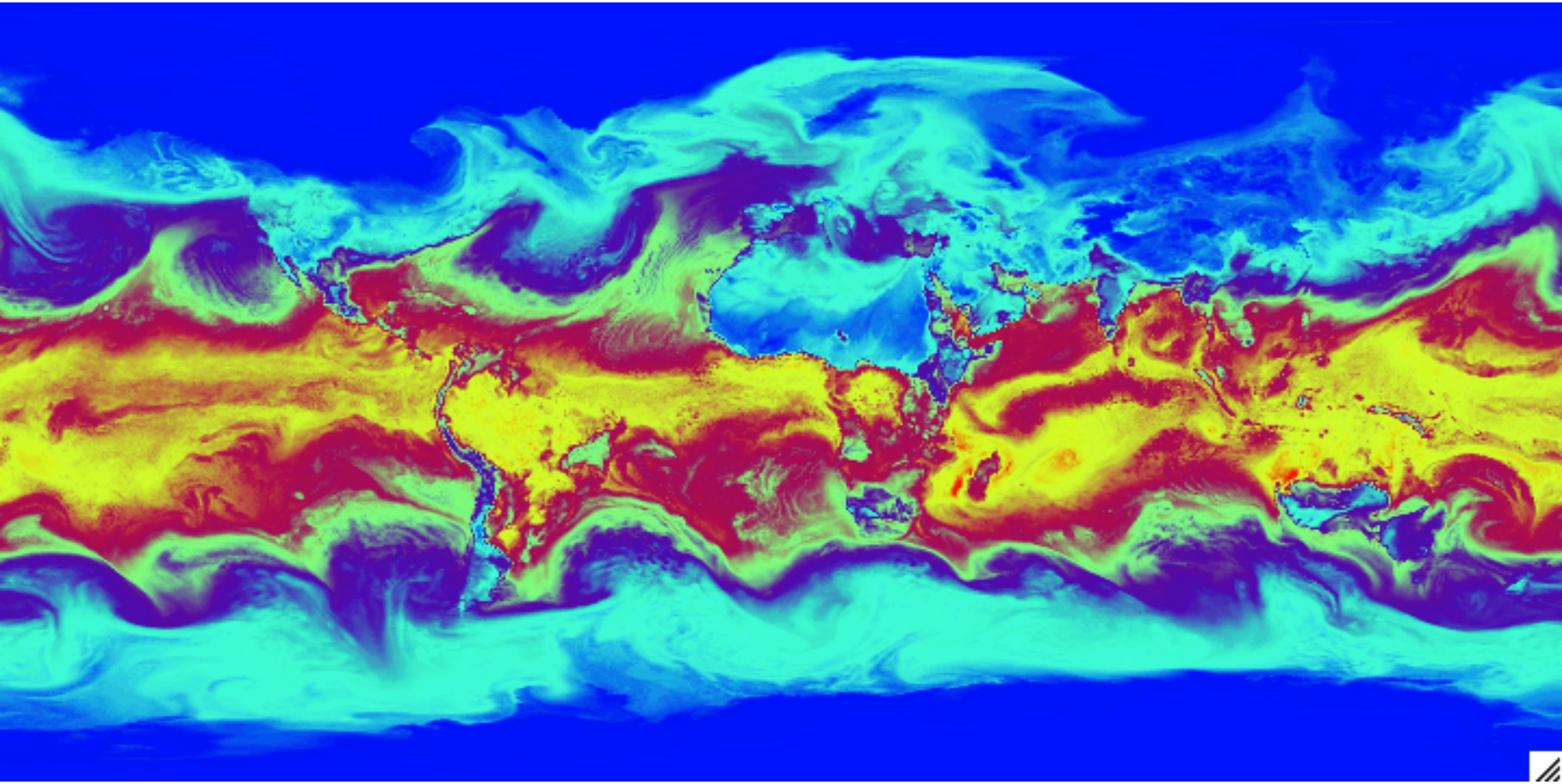
MPAS -> WRF (v3.9 WPS)

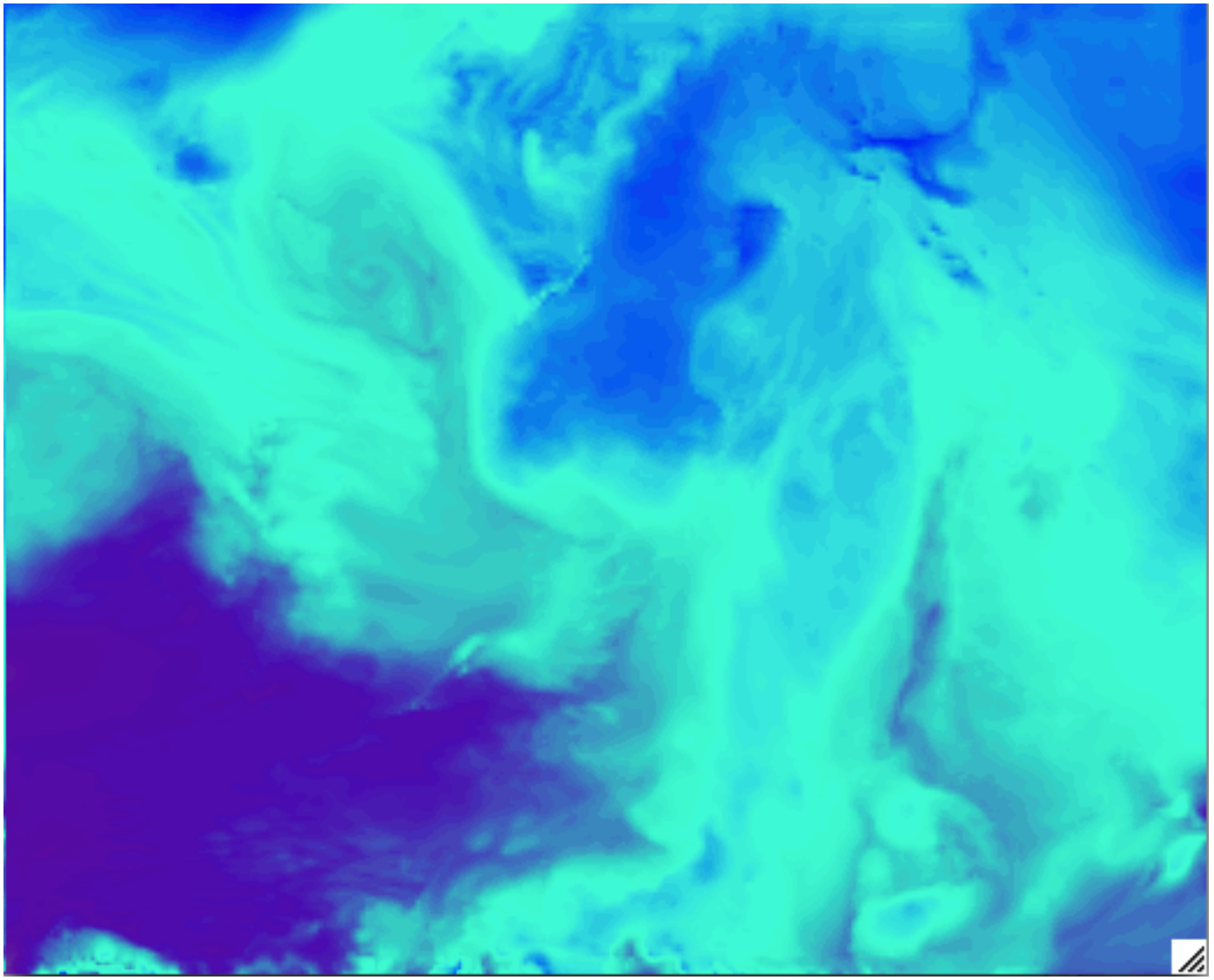
- Metgrid package directly ingests MPAS model output
- No interim re-projection

```
&metgrid
constants_name = 'mpas:120km.grid1.nc'
fg_name = 'mpas.120km'
```

WPS Domain Configuration



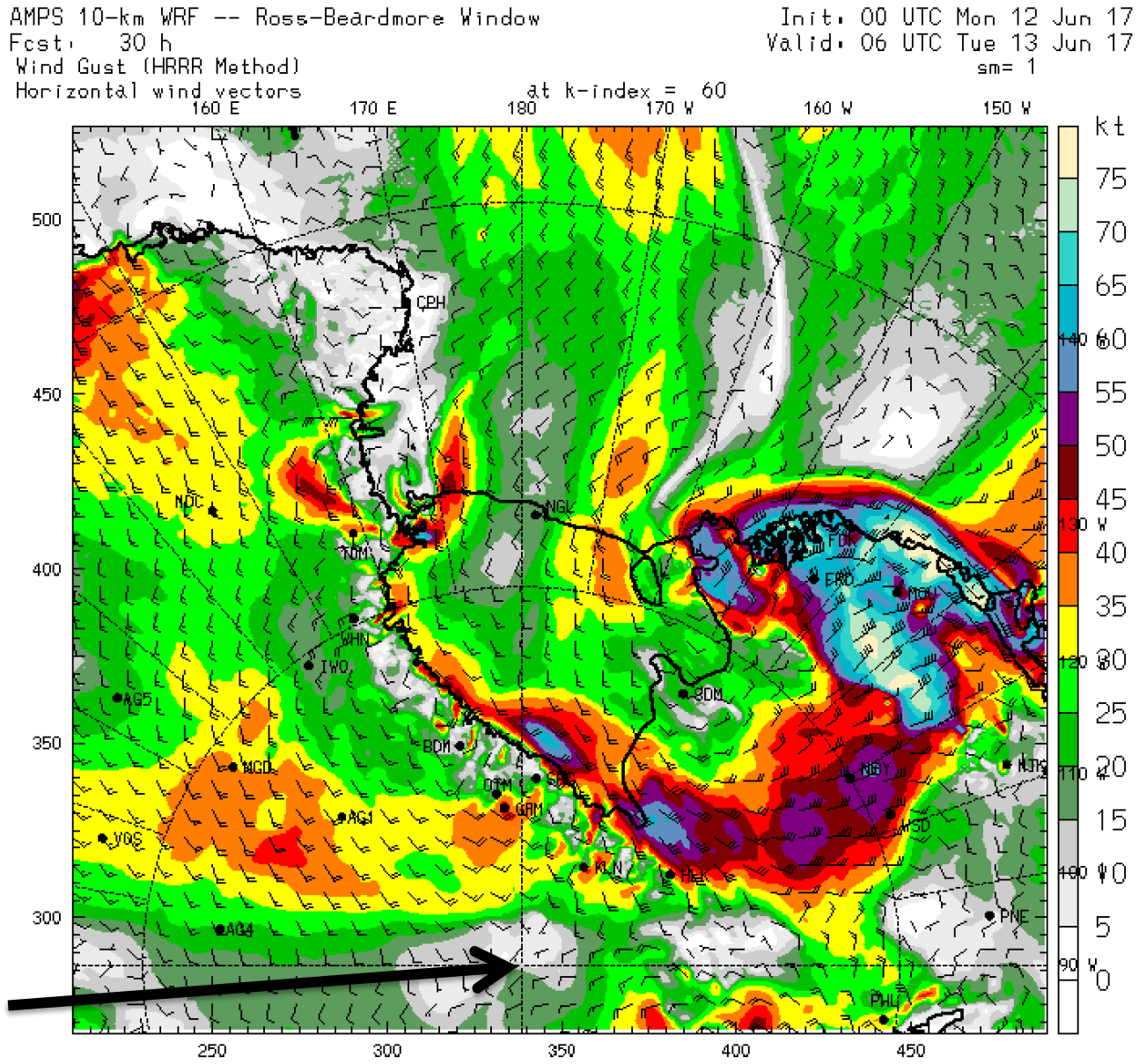




WRF&MPAS Share RIP!

WRF
Wind gust

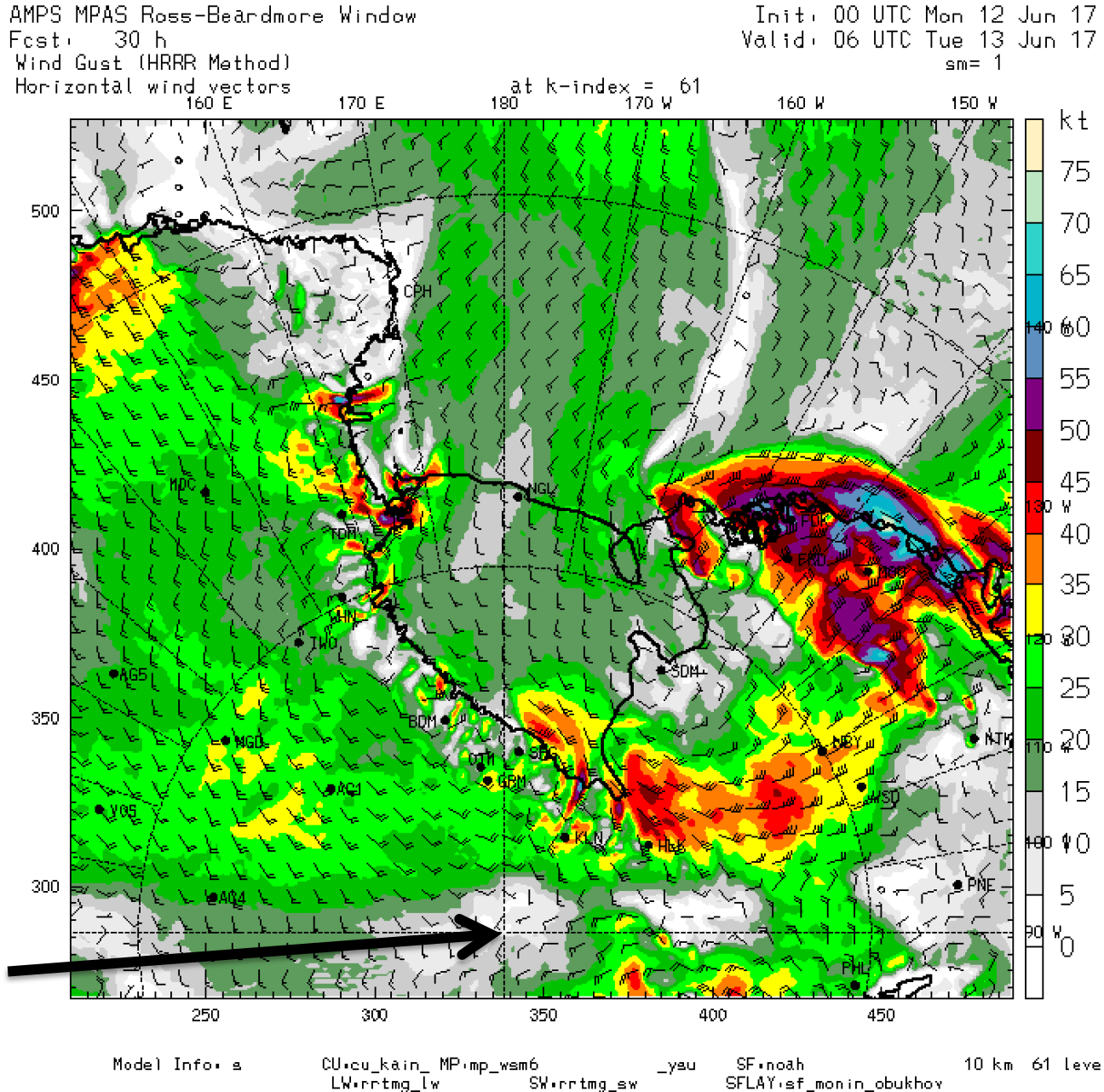
South Pole



WRF&MPAS Share RIP!

MPAS
Wind gust

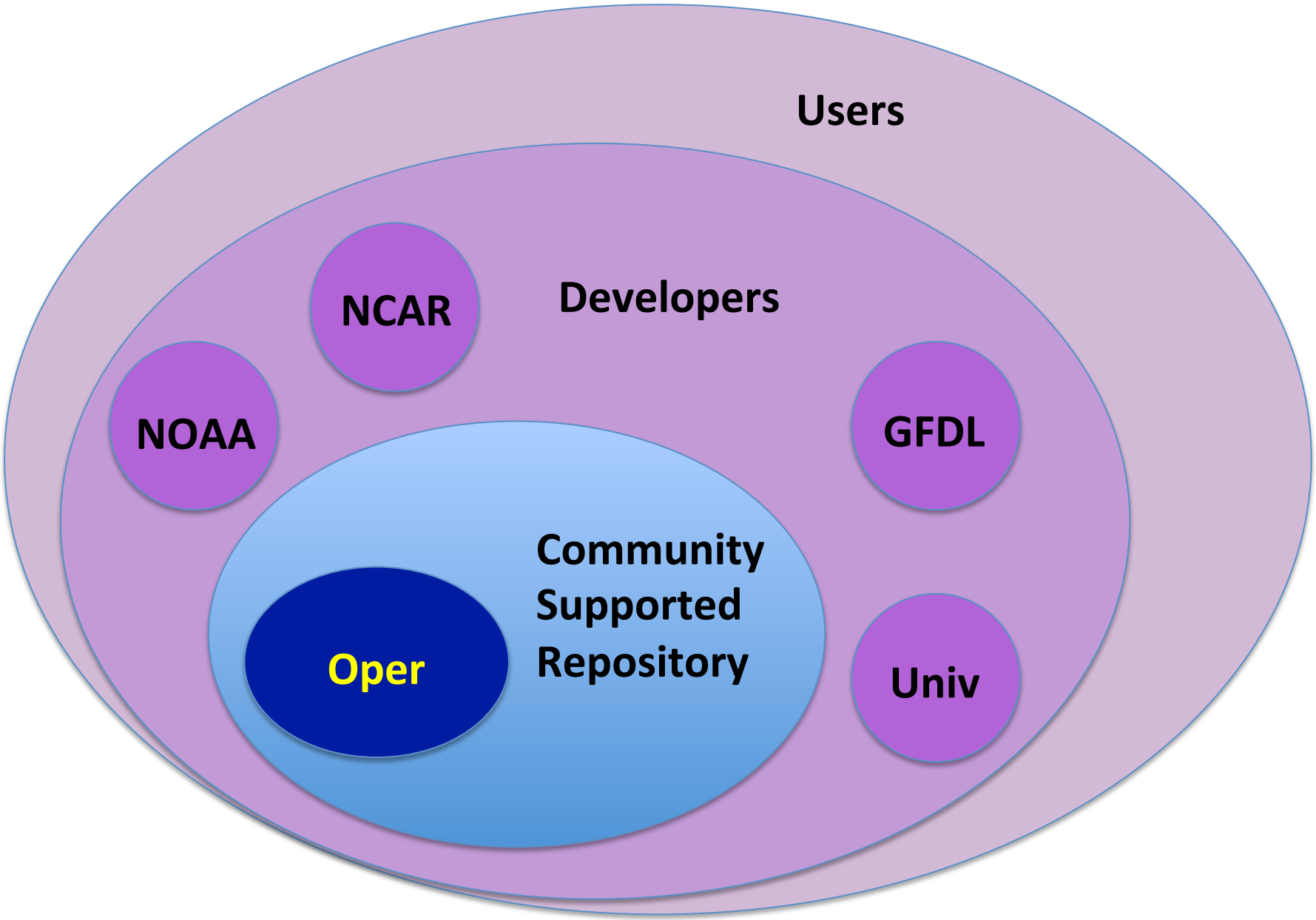
South Pole



WRF&MPAS Share RIP!

- Priscilla Mooney and Kevin Manning
  - Developers and image manufacturers
- RIP
  - Diagnostically rich
- Currently:
  - Intermediate lat-lon domain
  - Direct mapping?? We'll see
- Available next major release

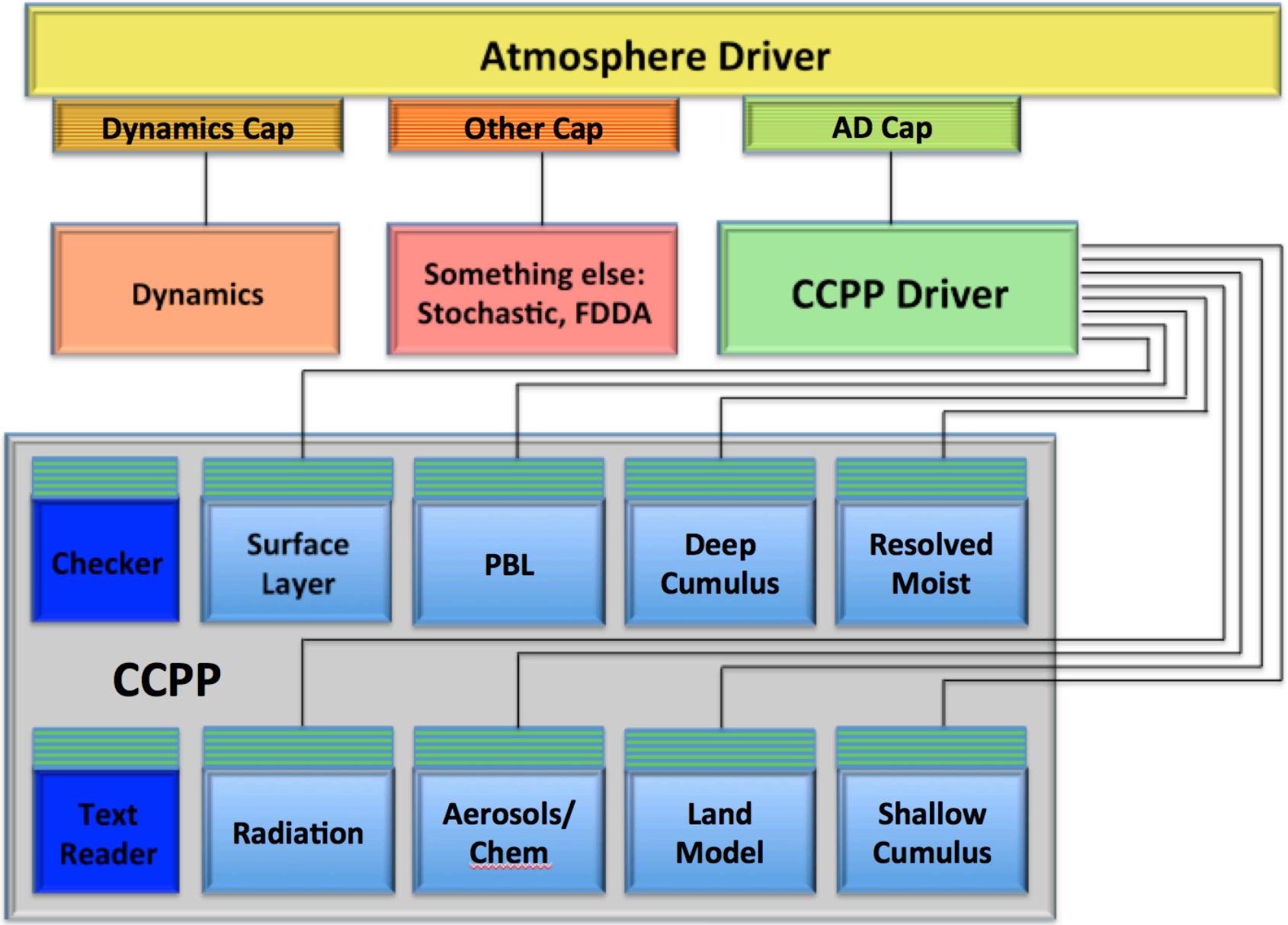
Shared Physics and Driver



Shared Physics and Driver

- Common Community Physics Packages (CCPPs)
- Shared physical parameterization schemes
- Ecosystem assumes git infrastructure
- NCAR: CESM, MPAS, WRF participating
- P1 *The Common Community Physics Package*, Bernardet, et al., DETAILS
- Interoperable Physics Driver: Need a way to call these various physics packages

Shared Physics and Driver

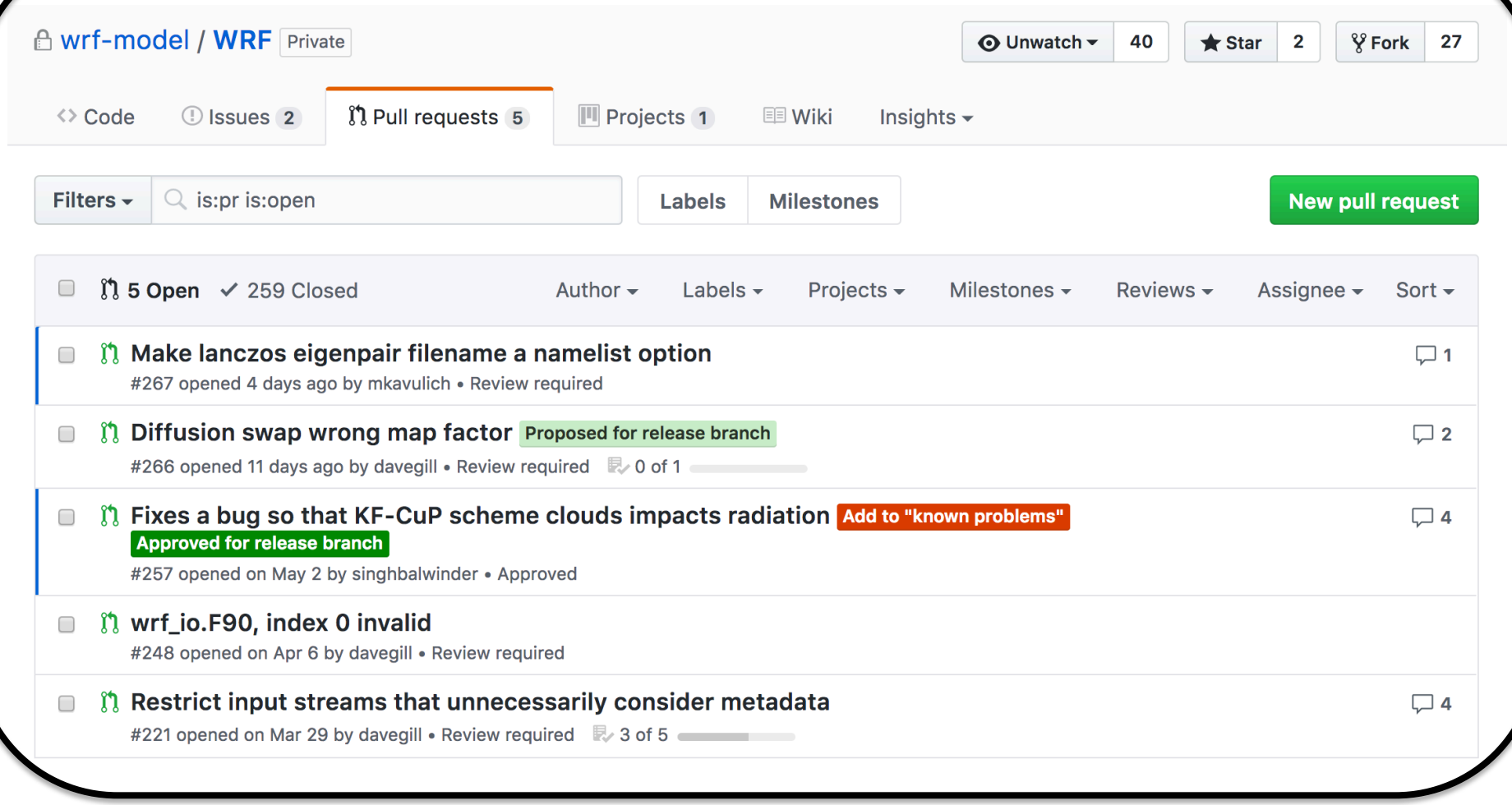


Shared Physics and Driver

```
<?xml version="1.0" encoding="UTF-8"?>
<suite name="RAP">
 <ipd part="1”>     <subcycle loop="1">
    <scheme>RRTMGLW</scheme>
    <scheme>RRTMGSW</scheme>
    <scheme>MYNNSFC</scheme>
    <scheme>RUCLSM</scheme>
    <scheme>MYNNPBL</scheme>
    <scheme>GF</scheme>
  </subcycle>     </ipd>
 <ipd part="2”>     <subcycle loop="1">
    <scheme>THOMPSONAERO</scheme>
  </subcycle>     </ipd>
</suite>
```

- Suite Definition File: SDF
- Text file of individual schemes which compose a “suite”

WRF+WPS on github



- WRFv3.9 transitioned from NCAR based *svn* repository to *git*
- Using github hosting services

WRF+WPS on github

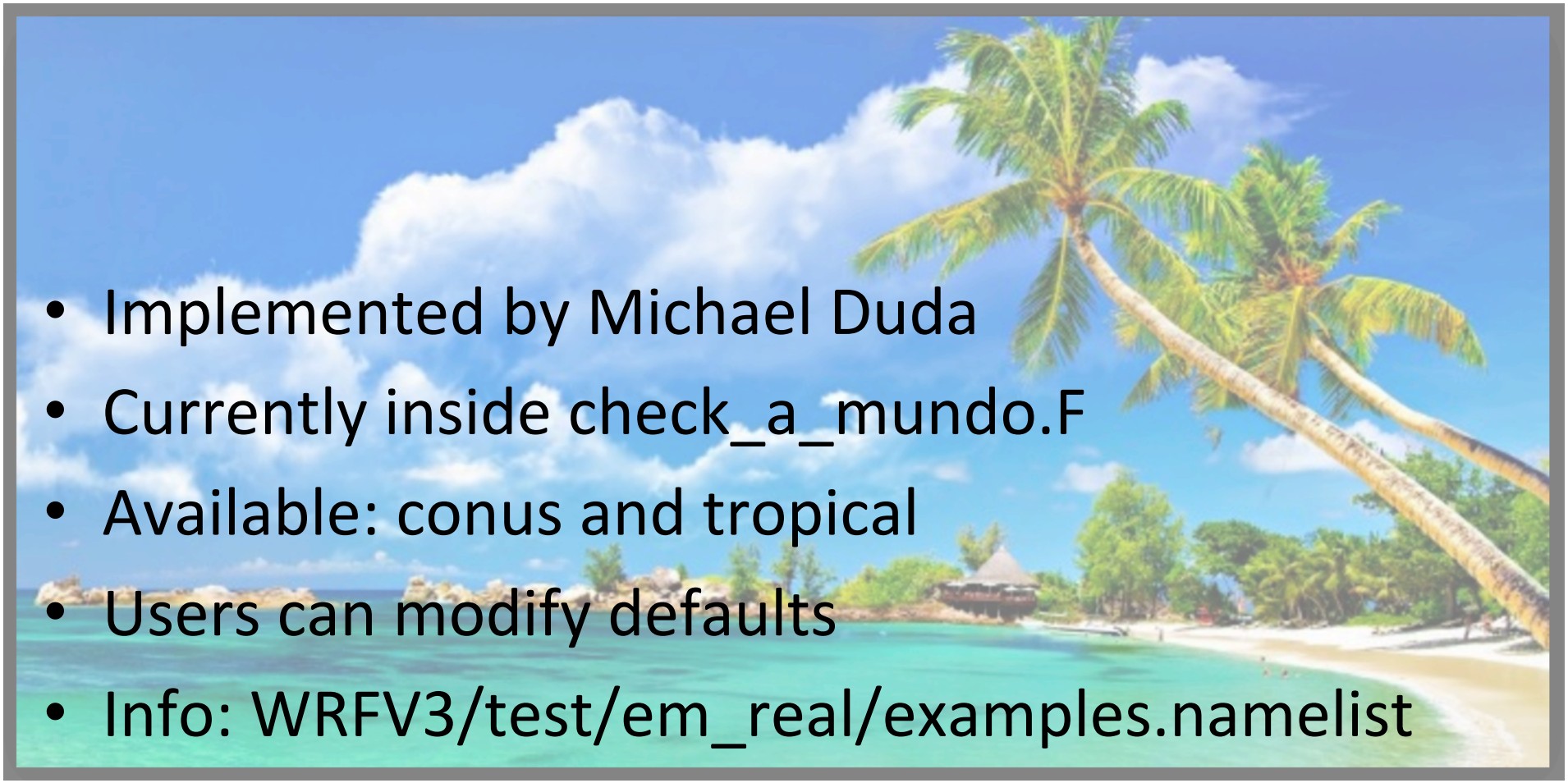
- Access is EASY: send your github User ID to Mike Kavulich **kavulich@ucar.edu**
- ALL DEVELOPERS: contribute code via github pull requests

WRF Suites – Two So Far

```
IF ( model_config_rec % cu_physics(i)         == -1 ) &
   model_config_rec % cu_physics(i)         = TIEDTKESCHEME

IF ( model_config_rec % mp_physics(i)         == -1 ) &
   model_config_rec % mp_physics(i)         = THOMPSON

IF ( model_config_rec % ra_lw_physics(i)      == -1 ) &
   model_config_rec % ra_lw_physics(i)      = RRTMG_LWSCHEME

IF ( model_config_rec % ra_sw_physics(i)      == -1 ) &
   model_config_rec % ra_sw_physics(i)      = RRTMG_SWSCHEME

IF ( model_config_rec % bl_pbl_physics(i)     == -1 ) &
   model_config_rec % bl_pbl_physics(i)     = MYJPBLSCHEME

IF ( model_config_rec % sf_sfclay_physics(i)  == -1 ) &
   model_config_rec % sf_sfclay_physics(i)  = MYJSFCSCHEME

IF ( model_config_rec % sf_surface_physics(i) == -1 ) &
   model_config_rec % sf_surface_physics(i) = LSMSCHEME
```

WRFV3/share/module_check_a_mundo.F

WRF Suites: Tropical

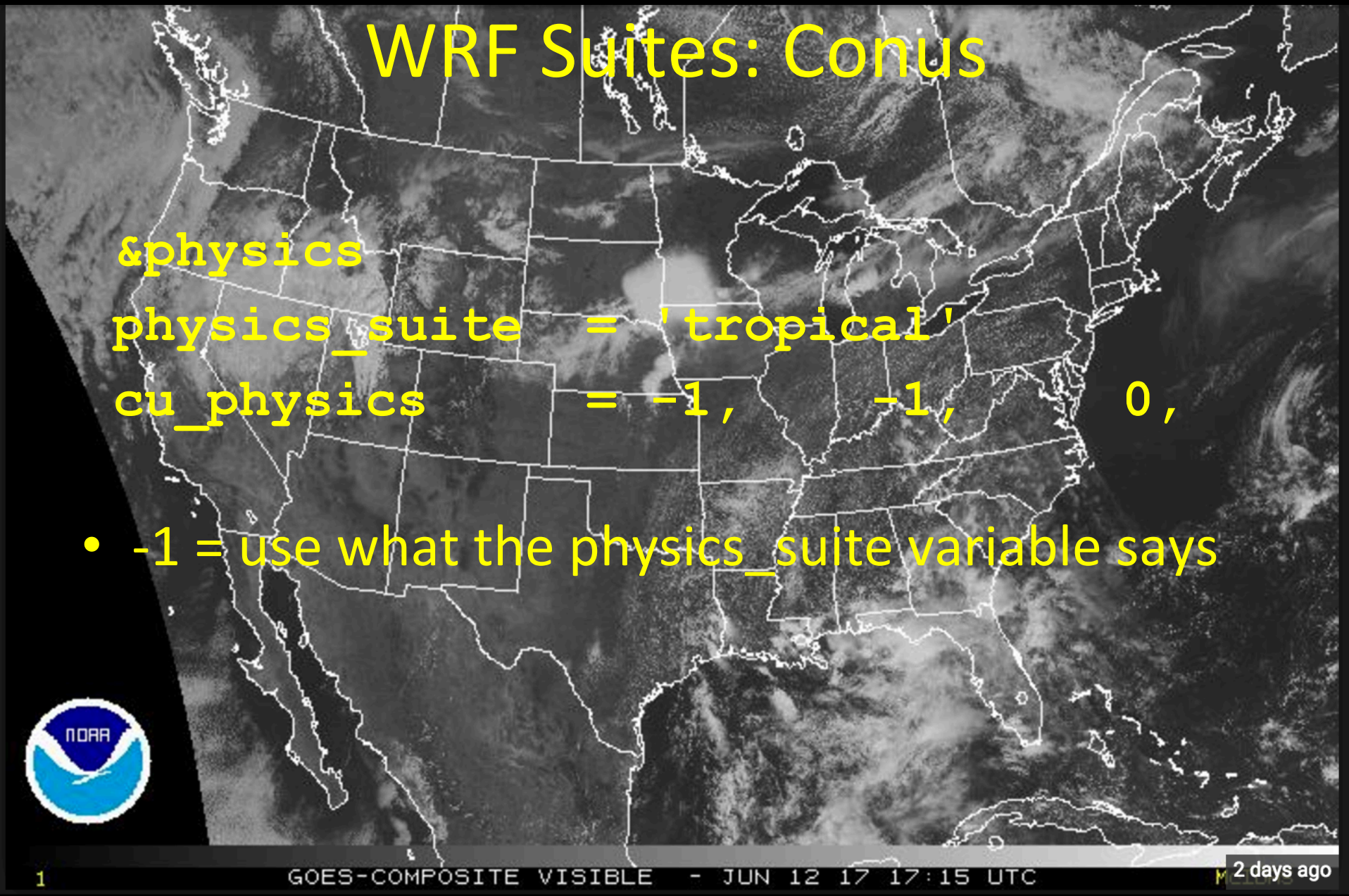


- Implemented by Michael Duda
- Currently inside check_a_mundo.F
- Available: conus and tropical
- Users can modify defaults
- Info: WRFV3/test/em_real/examples.namelist

WRF Suites: Conus

```
&physics
physics_suite  = 'tropical'
cu_physics     = -1,    -1,     0,
```

- -1 = use what the physics_suite variable says



Plans

- WRF v4.0 coming up in the Spring
  - Not tied to backward compatibility
    - LBC – maybe not coupled
    - Moist theta default: changes to all of the ideals + real + DA + post-procs
  - Shared physics
  - Shared infrastructure for physics – maybe
  - HVC:
    - Un-cpp-ize
    - All the ideals

Plans - Physics

- Required of physics developers:
  - Args commented:
    - type, dim, in/out, char name, units
- Physics Requirements (not exhaustive)
  - Args simple Fortran types
  - Routines: run, init, fini??, update??, restart??
  - Nothing is “module USEd”, args or file input
  - 1d assumption
  - Schemes do not call each other (or their subs)