WRF Software: Recent Updates, News, and Recommendations

CF Compliancy

Docker Containers

Expected Releases with "git"

Auxiliary output and Diagnostics

Revisiting WRF Benchmarks

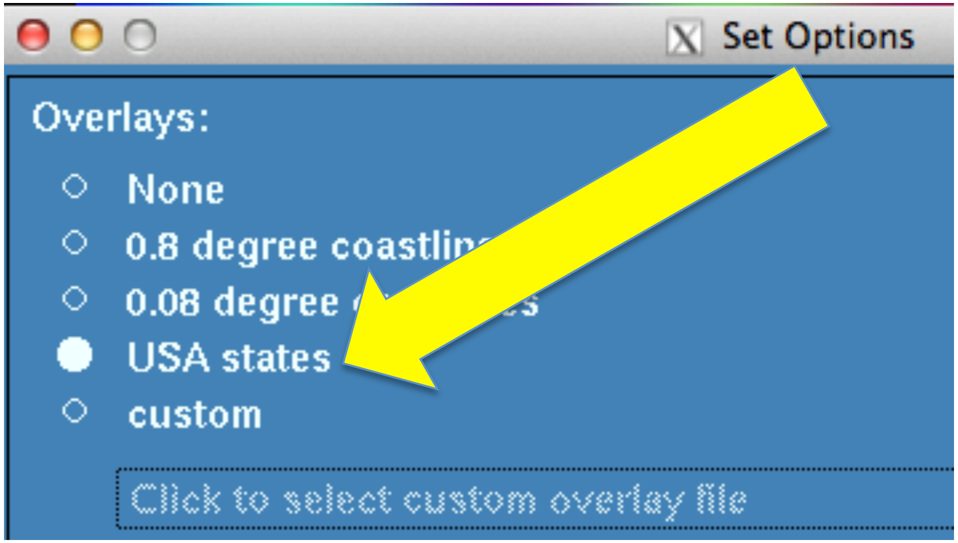
Plans for Future

Dave Gill
John Exby
John Michalakes
Huang Wei
Indraneil Gokhale
Carl Ponder

CF Compliant-ish

If you output the WRF model data with a single time period per file, ncview is able to recognize the WRF projections.

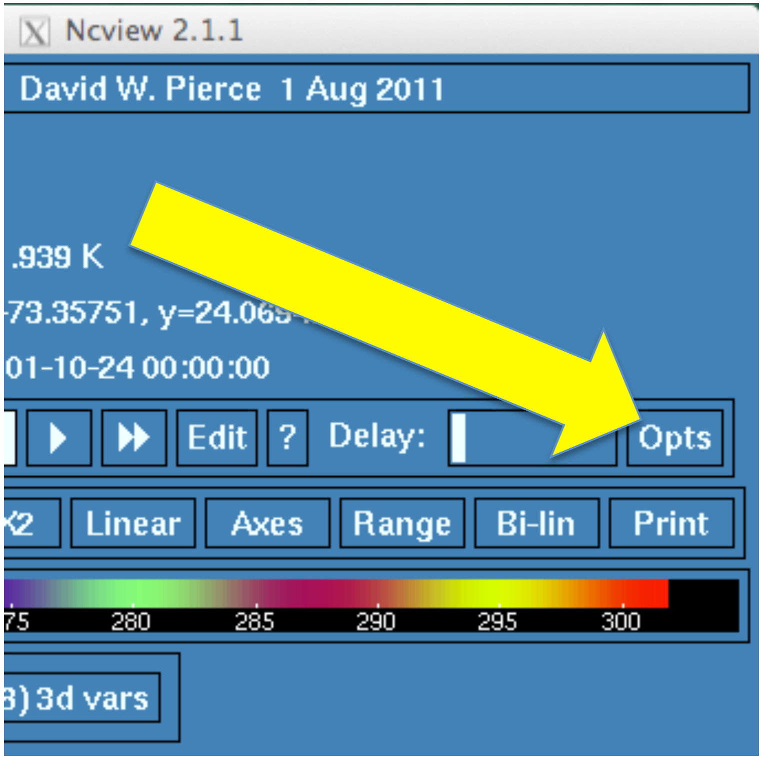
```
&time_control
frames_per_outfile = 1, 1,
/
```



Set Options

Overlays:

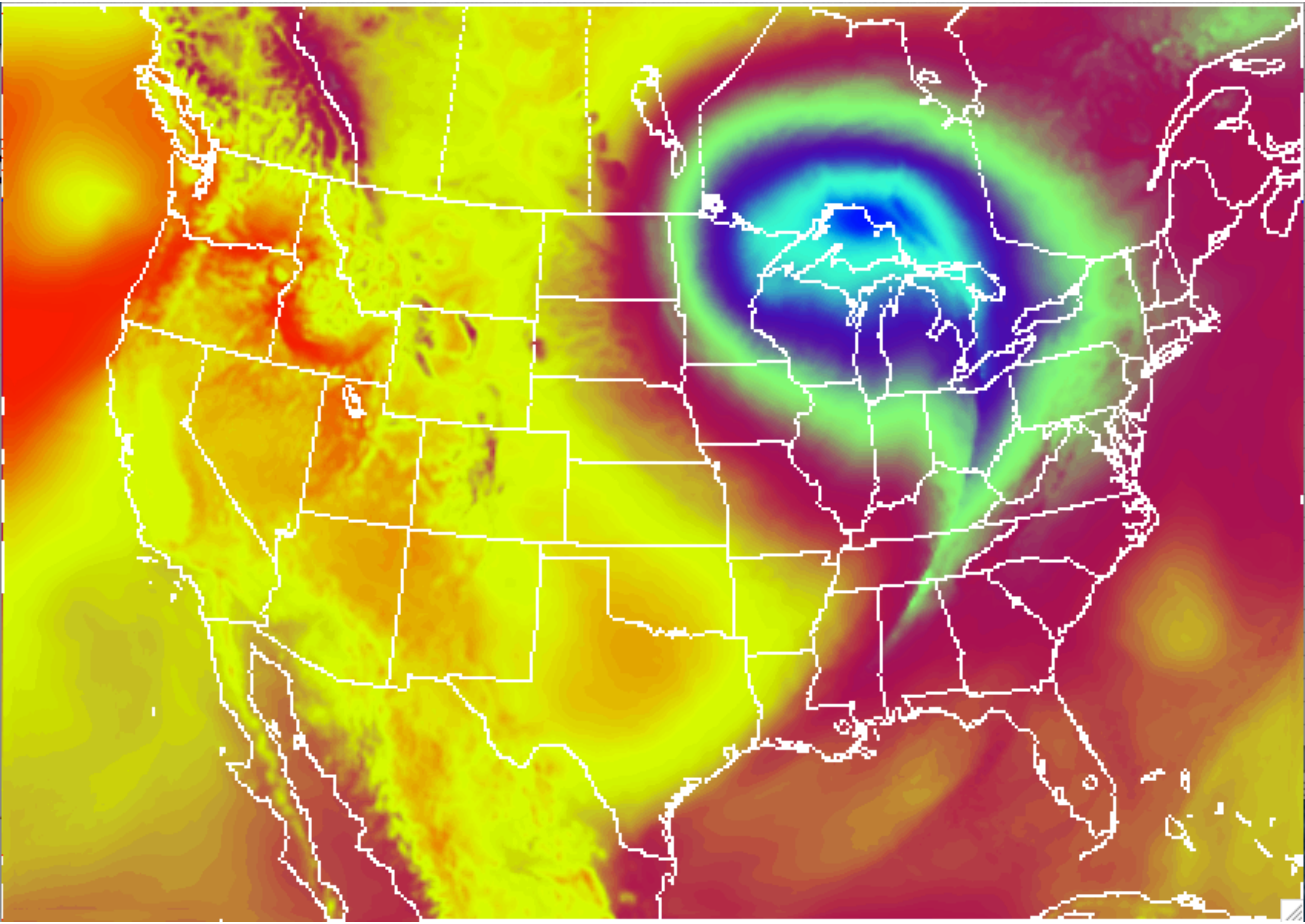
○ None
○ 0.8 degree coastlin
○ 0.08 degree c es
● USA states
○ custom



Ncview 2.1.1

David W. Pierce 1 Aug 2011

.939 K
-73.35751, y=24.069
01-10-24 00:00:00



Diagnostics

```
&time_control
 io_form_history       = 0
 io_form_auxhist23     =  2,
 auxhist23_interval    = 60,   30,    10,
 frames_per_auxhist23  =  1,    1,     1,
 auxhist23_outname     = "PLEVS_d<domain>_<date>”
/

 &diags
 p_lev_diags           = 1
 num_press_levels      = 2
 press_levels          = 50000, 25000
/
```

Diagnostics

```
&time_control
 io_form_history          = 0
 iofields_filename        = "field_list_d01.txt”, …
 io_form_auxhist24        =  2
 auxhist24_interval       = 60,   30,    10,
 frames_per_auxhist24     =  1,    1,     1,
 auxhist24_outname        = "SFC_d<domain>_<date>”
/

&afwa
 afwa_diag_opt            = 1,   1,   1,
/

File:
```

+:h:24:MU,RAINC,RAINNC,U10,V10,T2,Q2,XLAT,XLONG,AFWA_MSLP, REFL_10CM

Docker Containers

One of the biggest hassles new users have with the WRF system is **BUILDING**

→Getting the correct libs set up to work with the compiler

→Finding the right ENV variables and PATH

Wouldn't it be great if we could provide a turn-key WRF system? A complete dev environment?

Docker Containers

Build, Ship and Run

Any App, Anywhere

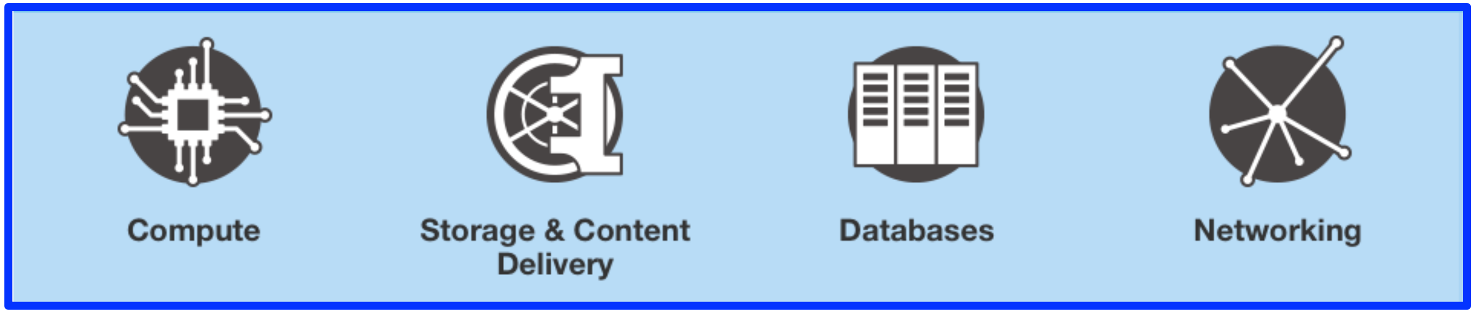
Docker is an open platform for developers and sysadmins of distributed applications.

An NSF project to enable graduate students to use the WRF model on laptops: Mac, Windows, and Linux

… all with the same executable

Done with a VERY small Linux image

Want to Participate?: contact Josh Hacker hacker@ucar.edu



Benchmarking

Revamping the existing benchmark files

Small one is 425x300, 12-km, 3-h: so suitable for most desktop -> department-sized systems

Will provide restart, lateral boundary, and namelist.input for all WRF releases: from **3.0.1.1** upto **3.7**

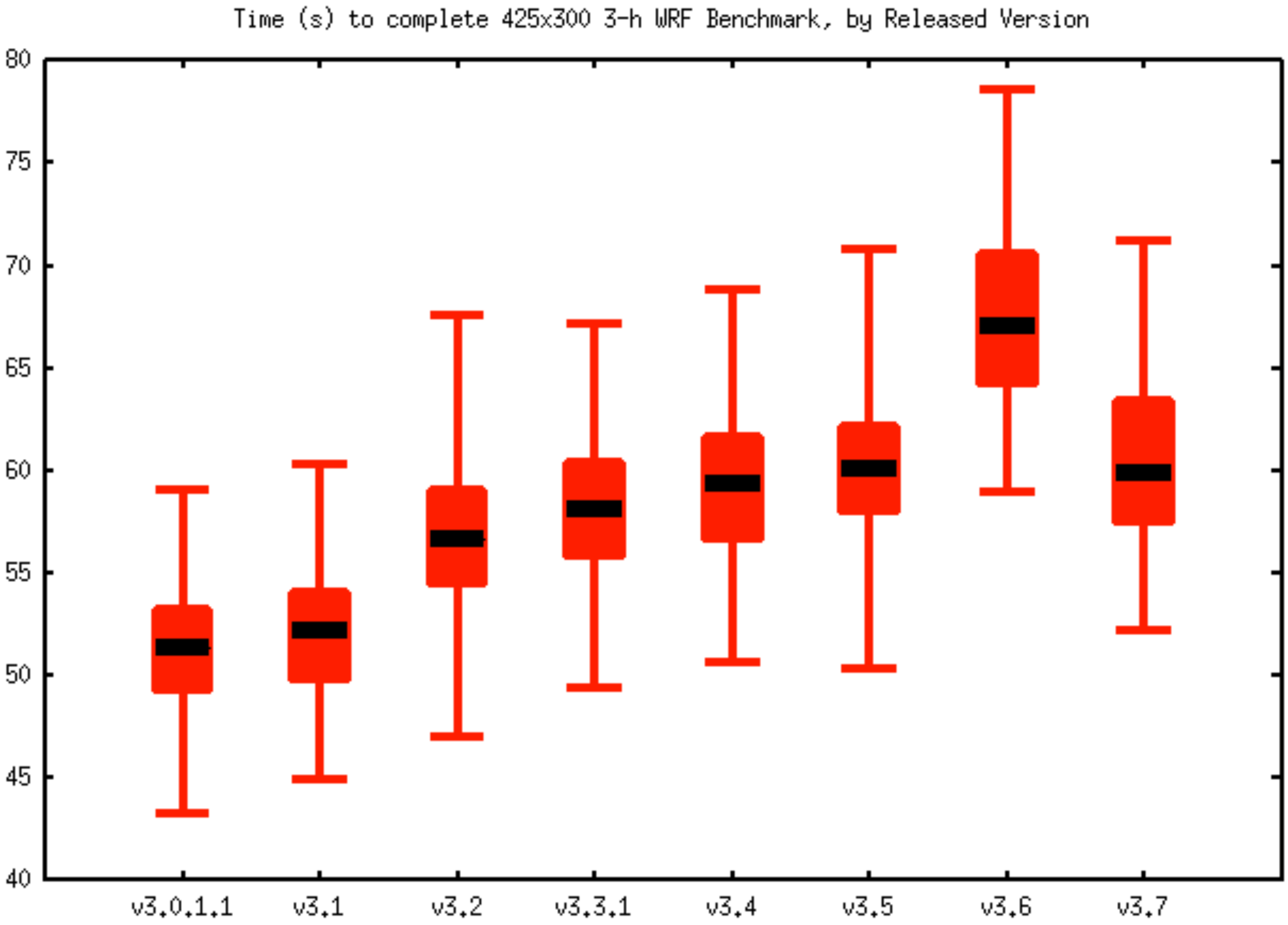
Benchmarking

- 2001 Oct 24 0000 UTC init + 24-h simulation, drop a restart
- 3-h forecast (72 s time step) gives 150 time steps
- No I/O counted in timing
- For 1 through 1024 cores (by powers of 2)

Benchmarking

- Finding **WIDE** variation in timings, but a definite slowing trend
- Ran 200 instantiations of each of 8 WRF model releases

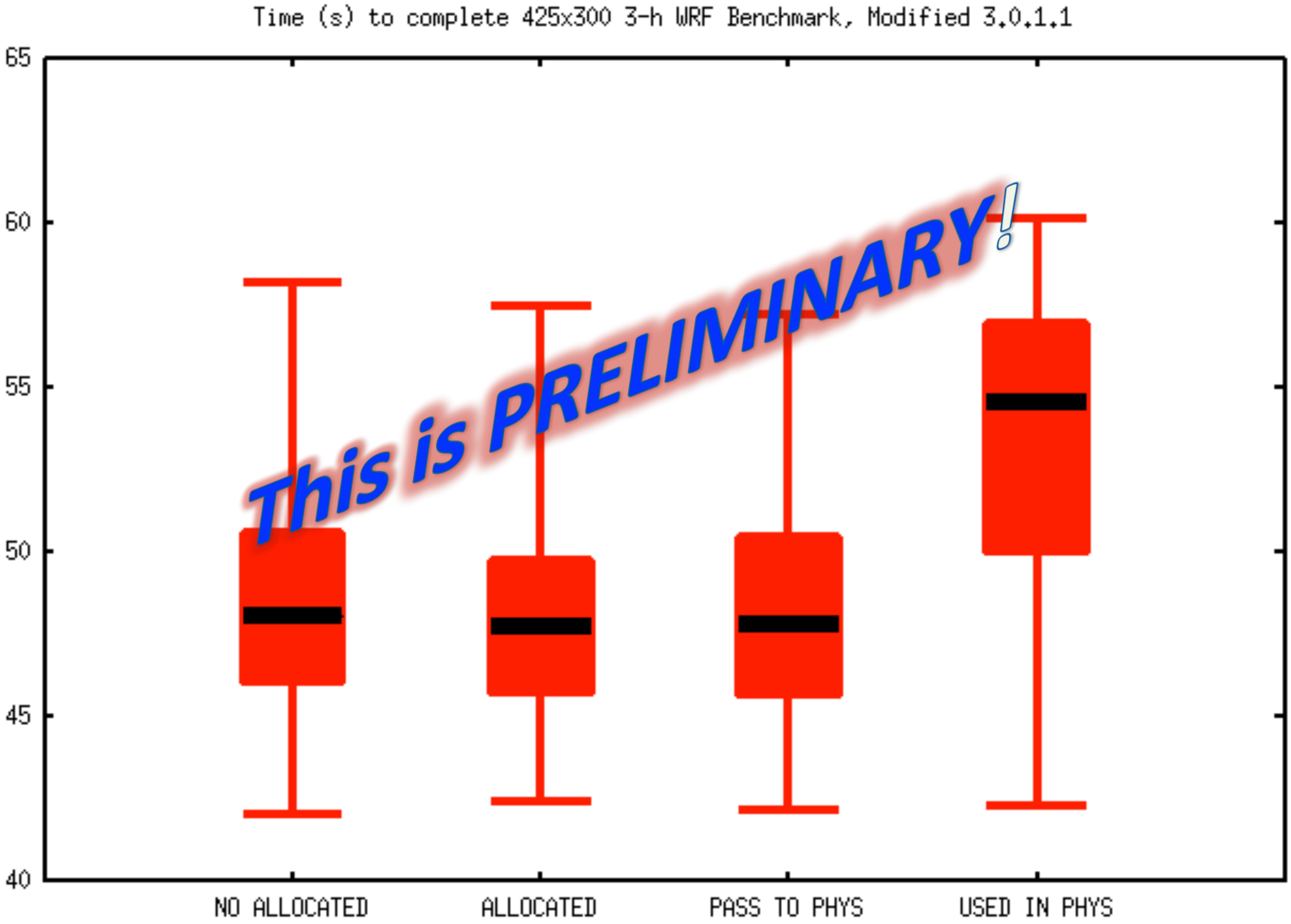
Benchmarking



Benchmarking

- Cause of the slowdown?
- Added new schemes, more arrays to existing schemes, communications
- Start with 3.0.1.1 (200 runs each)
  - Add 3d arrays in Registry, in package
  - Turn package on
  - Pass all arrays from first_rk_step to schemes
  - Assign ONE value per patch in phys routine

Benchmarking

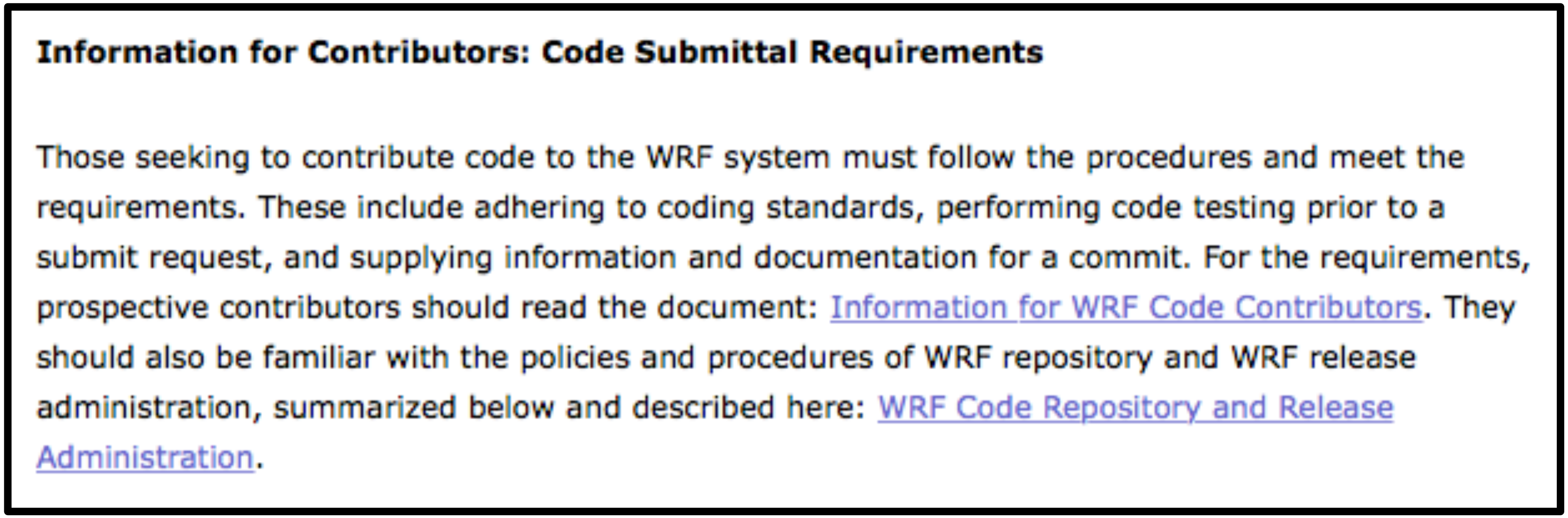


Getting Code into WRF

Reminder: We have existing policies for providing code to the WRF repository

Web search for:

“**`WRF code requirements`**”



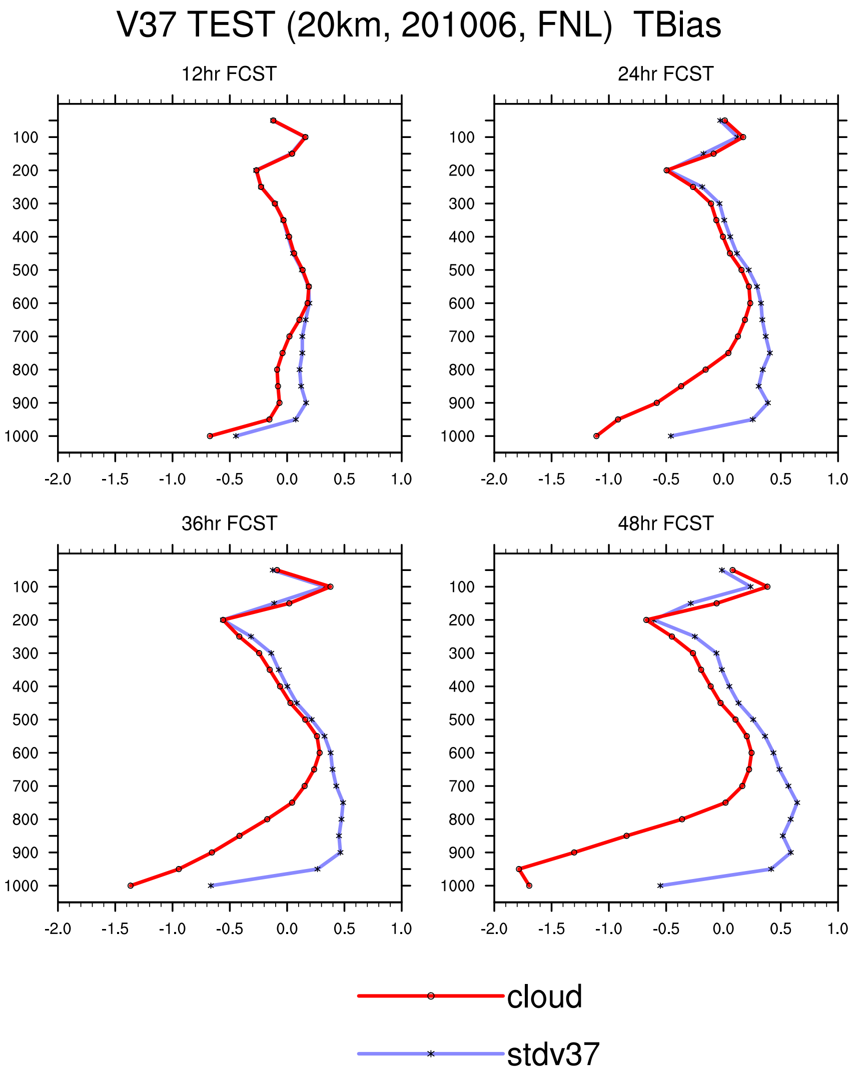
Information for Contributors: Code Submittal Requirements

Those seeking to contribute code to the WRF system must follow the procedures and meet the requirements. These include adhering to coding standards, performing code testing prior to a submit request, and supplying information and documentation for a commit. For the requirements, prospective contributors should read the document: Information for WRF Code Contributors. They should also be familiar with the policies and procedures of WRF repository and WRF release administration, summarized below and described here: WRF Code Repository and Release Administration.

Getting Code into WRF

More testing on developer’s part facilitates **quicker** community access to more thoroughly **vetted** code

Example: vertical profile of model bias in T



Courtesy of Ming Chen, MMM NCAR

Getting Code into WRF

Review panel for code: to be considered

… and to help developers get access to WRF ->

Starting with v3.7.1, WRF will be released in parallel using a public **git** repository

Release branch with restrictions; ***open trunk***

Working through the subversion history and branches, and what gets released

WRF Architecture Working Group

No longer can rely on faster "traditional" chips

Need to aggressively look at accelerators

New Working Group has WRF developers AND hardware/software vendors to steer model in directions likely to be beneficial

WRF Architecture Working Group

Recent years' OpenMP fixes give 30% speed-up in DM *vs* DM+SM (TACC, 400x400 test case) – moving in right direction for larger thread counts

(Davide del Vento, CISL/NCAR, informal presentation)

Might consider atypical (for WRF) source code possibilities for performance sake – **guidance sought from breakout groups**