SMOKE-RT
for
WRF-Chem
and
Air Quality Forecasting

Carlie J. Coats, Jr.
Baron Advanced Meteorological Systems, LLC
Raleigh, NC

WRF and WRF-Chem:

- Next-generation mesoscale modeling system
  - Targets 1-10 KM resolution grids
  - Modular science processes
  - Integrated atmospheric chemistry option (work in progress... **needs emissions**)
  - HPCC computing framework: OpenMP shared memory + MPI distributed memory hybrid
- Designated successor to MM5, ETA, other models
  - Sponsored by NOAA, EPA, Air Force, Navy, others
- *http://wrf-model.org/index.php*
- *WRF is grid-oriented model; emissions is geopolitical inventory problem:* ***structure mismatch***

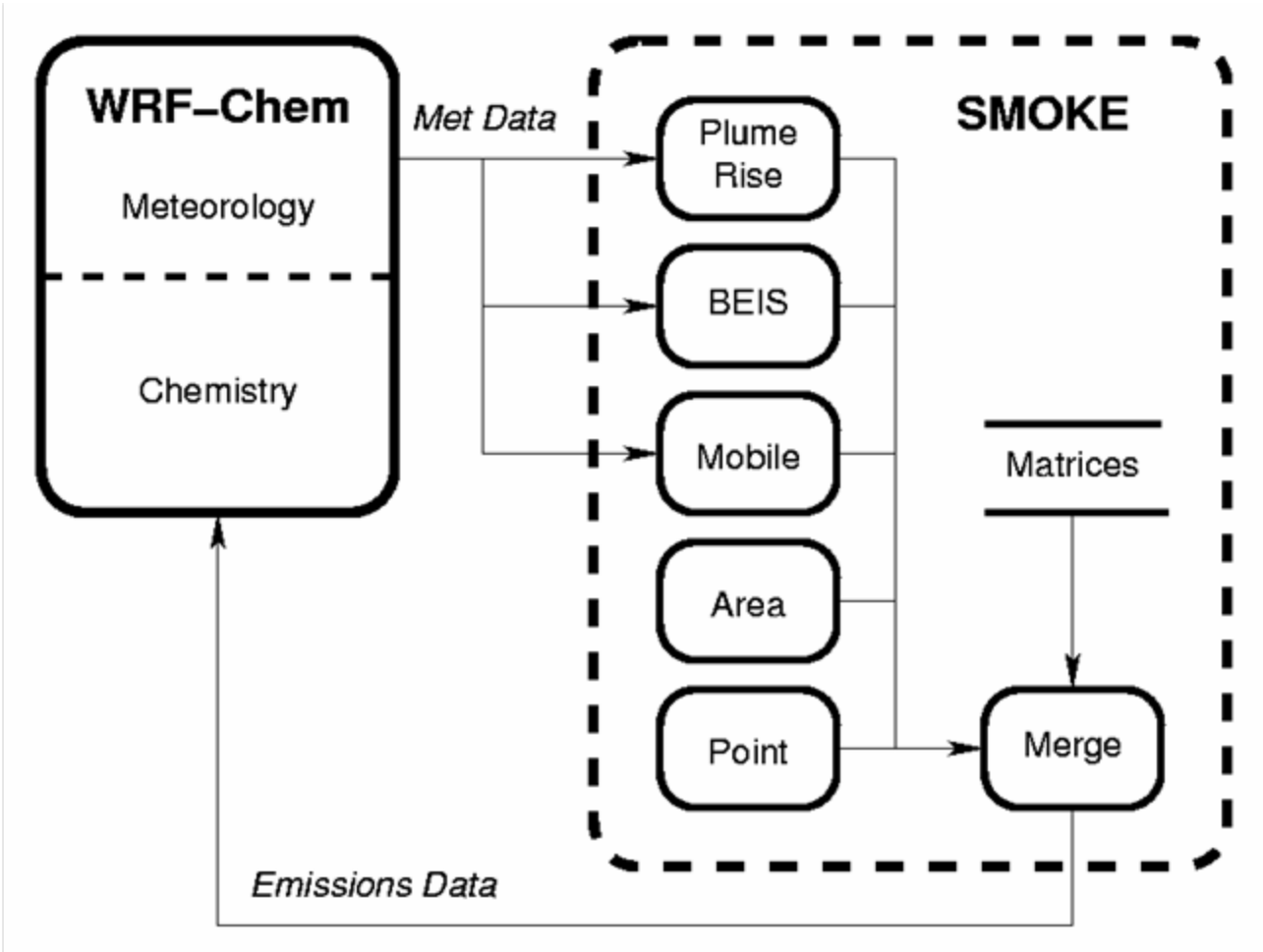
WRF: Layered Model Structure

- *Driver Layer:* manages I/O, communication, nests, state data, decomposition of the grid into distributed-memory parallel “patches” and shared-memory parallel “tiles”
- *Mediation Layer:* interface between Driver and Model layers; contains ***solver***
- *Model Layer:* computational routines for atmospheric processes, called on “tiles” **(scientists work here…)**
- *Registry:* table that describes the desired model to build
  - Variables, science processes, data sets, and how they all interact
  - Build process customizes the source code according to ***`Registry`***: variables, science-modules, data sets

SMOKE and WRF-Chem: *Idea*

- Use WRF "external package" facility to put Models-3 I/O API into WRF
  - Fits in with official "layered-model" architecture
  - Turns out to have "complexities" due to WRF design
- Use I/O API Coupling Mode for "cooperating process" WRF-Chem/SMOKE implementation
  - Met part of WRF-Chem provides met data to SMOKE
  - SMOKE provides met-modulated emissions data to chem part of WRF-Chem
  - "Just works" for both in-line and off-line modeling
- *http://www.baronams.com/projects/wrfchem/*

SMOKE and WRF-Chem -- Concept



Models-3 I/O API *(M3IO)* and WRF(-Chem)

- Fits into WRF architectural design:
  - Driver layer interfaces to I/O through "wrappers"
  - READ, WRITE routines look very much like **`READ3,WRITE3`**
  - ***`Registry`***-driven build system generates I/O calls
- Ostensibly, three easy sub-tasks:
  - Translate date&time/map-projection/grid conventions
  - Build wrappers
  - Additions to ***`Registry, Makefile`***
- Reality is quite different
  - Failure in the underlying WRF systems analysis made this effort much harder than it should have been

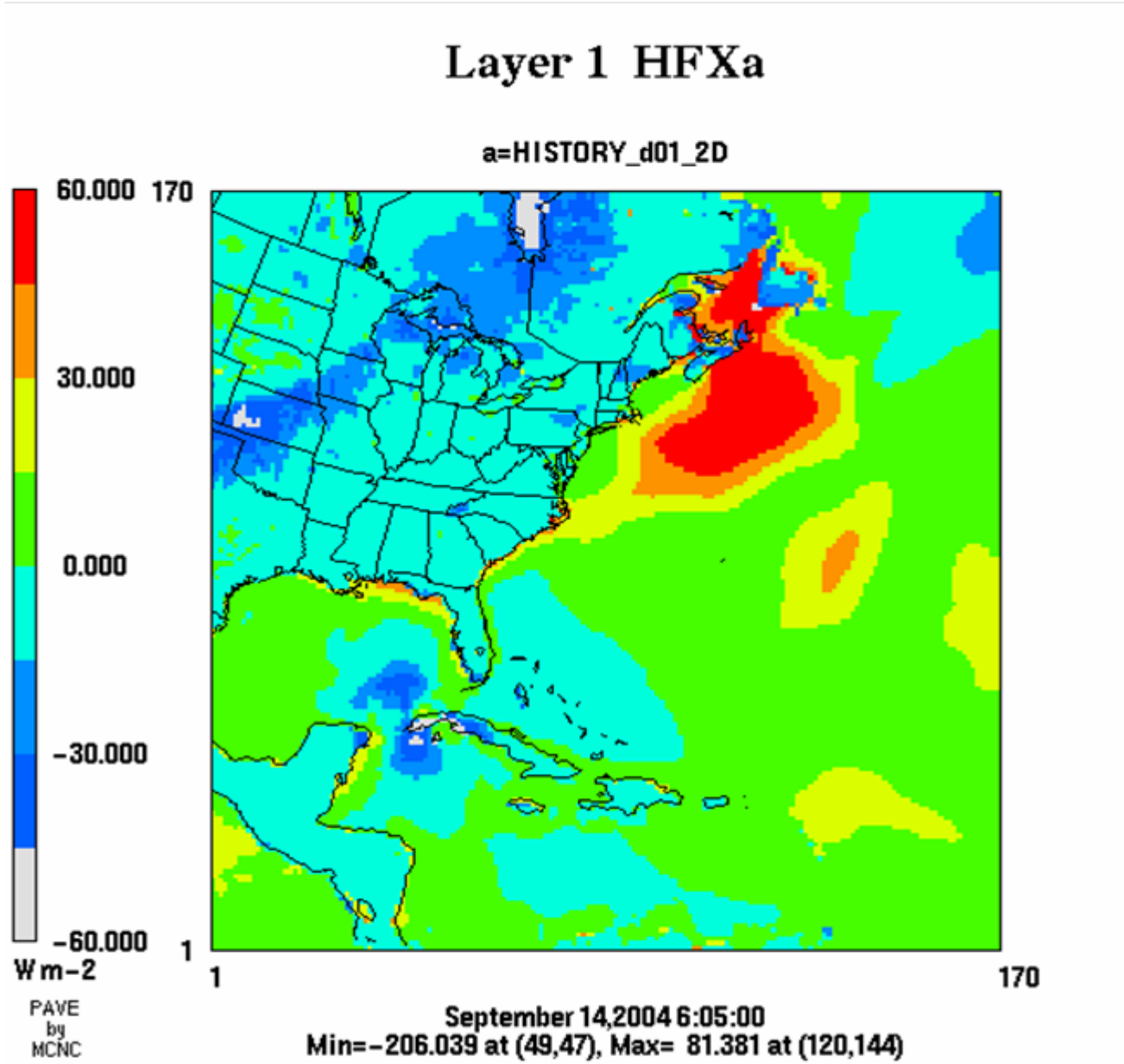
M3IO and WRF (cont'd)

- First pass, learning the architecture: ***about a month***
- READ, WRITE, CLOSE were easy: ***about two hours***
- Date&Time/Map-projection/Grid conventions:
  - ***About three weeks*** (changes from WRFv1 to WRFv2)
  - WRF grid conventions are inadequately precise (and WRF framework makes this difficult to fix)
- File-creation: ***over 8 man-months!***
  - *PROBLEM:* WRF has no real notion of "data set definition"
  - WRF data set creation uses ***eight levels of subroutine calls*** (some indirect—passing subroutines as arguments)
  - File description embedded in ***`Registry`***-generated code (version-specific? ...subject to change? *Yes, to both!*)

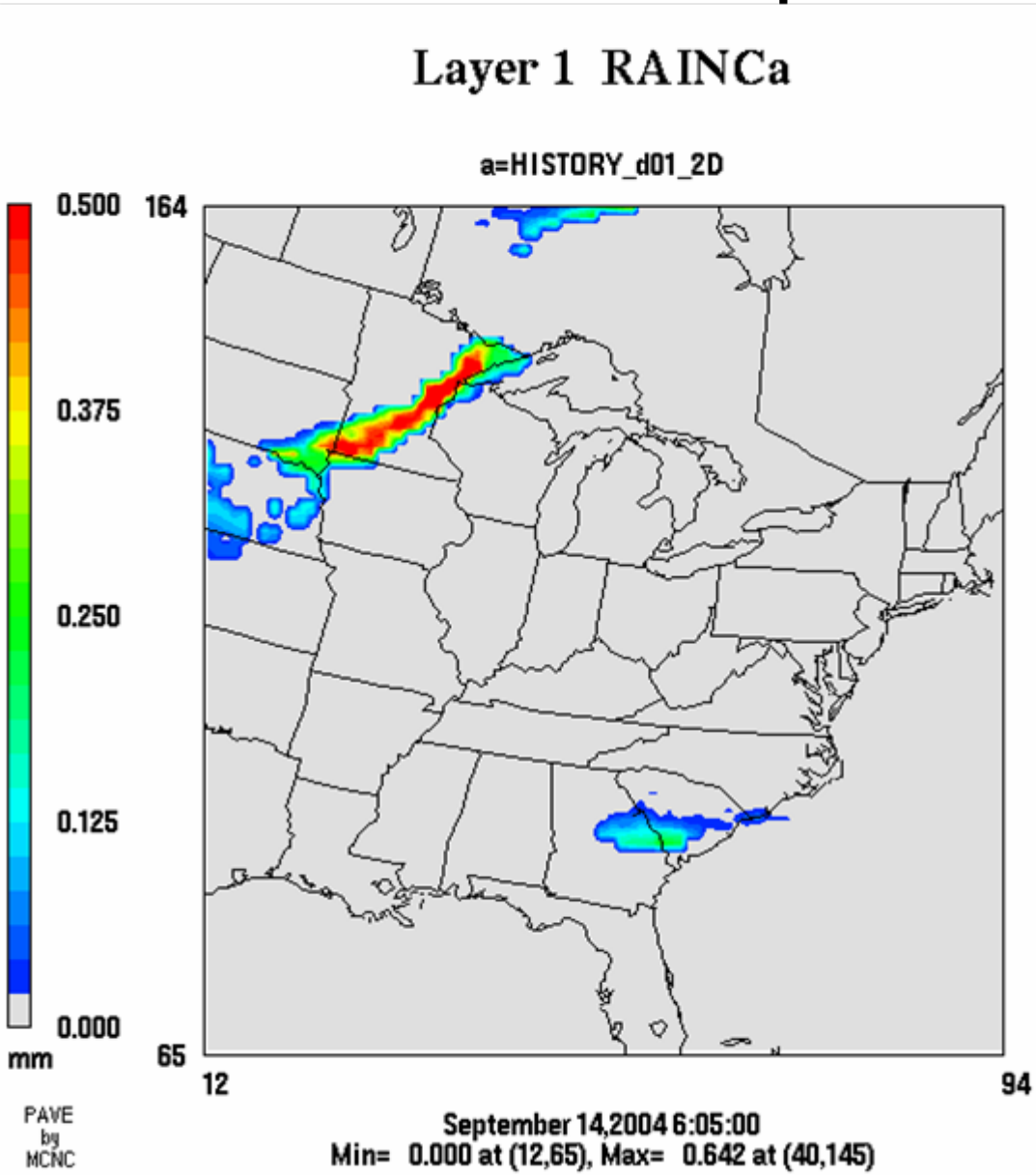
M3IO External package for WRF

- Access to all M3IO modes, tools
  - *NetCDF* file mode, *native-binary* file mode, *PVM*-based coupling mode
  - **`PAVE`**, ***`m3tools`*** package, more...
- Fits (almost) cleanly into WRF build system:
  - **`ext_m3io.F90`**: wrapper code, as specified by WRF
  - **`module_m3io.f90`**: state management for M3IO
  - **`m3io_util.f90`**: time-manipulation, array-transpose, etc.
  - **`m3io_grid.f90`**: interface for M3IO-required metadata not provided by standard WRF framework
  - **`Registry`** mods, other minor mods.

WRF HFX in PAVE



WRF Convective Precip in PAVE



SMOKE and WRF-Chem: Problems

- SMOKE needs to be a ***"time stepped model"*** with the following changes:
  - Do all work for one time step before going on to next time step (get rid of "pollutant groups"!!!)
  - Must work with sub-hourly meteorology input files
  - Must be able to generate sub-hourly emissions output files
  - ***Don't override the user's run specifications!*** This is necessary for coupled-model applications (or for general operational integrity)

SMOKE and WRF-Chem

- ***Parallelization and Optimization***
  - Unrealistic to expect a single-processor SMOKE to cooperate effectively with a WRF model hosted on a system with hundreds of processors
  - Simplified, modular program drivers required in order to do parallelization and optimization

SMOKE and WRF-Chem

- ***Software Engineering:***
  - Build system must be compatible with I/O API
  - Should not make private (and out-of-date) copies of I/O API routines and data structures, nor of external libraries
  - Encapsulation: needs modular/OO (re-)design
  - Referential integrity checks: full primary key should exist within inventory files and should be checked
  - Should check failure status for all library calls
- SMOKE should be ***re-structured*** so that the code is ***comprehensible*** and **maintainable**

SMOKE-RT for WRF and AQ Forecasting

- New versions of meteorology-modulated time stepped emissions programs
  - File-compatible with older versions
  - Parallelization; improved modularity and efficiency
  - **`LAYPOINT, TMPBEIS, TMPMOBIL`**
- New, more powerful **`EMISMERGE`** program
  - Multi-inventory, multi-category merge
  - Single-stage, with improved efficiency
  - Merges for multi-inventory lists of each type

LAYPOINT

- New UI and driver layers
  - Arbitrary user-selected time step (requries compatible met-input time step)
  - Does not override user's run-specifications
  - Uses **`TA,QV,P,Z,U,V`** from met model (compatible with both MM5, WRF(-Chem)) with configurable names
  - OpenMP parallelization, substantial optimization
- Same plume-rise algorithms
- Stack-height adjustment for sub-gridscale terrain
  - Hills and valleys are “part of the stack”

TMPBEIS

- New UI and driver layers
  - Arbitrary user-selected time step
  - Does not override user's run-specifications
  - Uses **`TA2,QV,GSW`** from met model (compatible with both MM5, WRF(-Chem))
  - OpenMP parallelization
  - Optional lapse adjustment to compensate for grid scale met-model terrain height error
- Same BEIS3 algorithms, file compatible with existing **`NORMBEIS3`**

TMPMOBIL

- Replaces mobile-source use of **`TEMPORAL`**
- New UI and driver layers, module structure
  - Arbitrary user-selected time step (up to one hour)
  - Does not override user's run-specifications
  - Uses **`TA,TAMAX,TAMIN`** from met model (compatible with both MM5, WRF(-Chem))
  - OpenMP parallelization, optimization (4.1x faster for East-of-Rockies 45KM test case on SGI Origin)
  - Optional lapse correction for terrain height error
- ***Mobile5b*** for current version (future ***Mobile6-lite?***)

EMISMRG

- Single-stage multi-inventory merge
  - Replaces **`SMKMERGE`** and **`MERGEGRID`** programs
  - Takes (possibly-empty) lists of area-style, mobile-style, BEIS-style, point-style file-sets and date/time specs
  - Resolves hourly, sub-hourly time stepping issues
  - New “standard-week” treatment
  - Optional 3-D sub-grid scale treatment of area, biogenic, mobile emissions using layered terrain-penetration fractions: see poster session
- OpenMP parallel, major efficiency improvements
- Does *not* override user run-specifications

SMOKE Status

- **`LAYPOINT, TMPBEIS, TMPMOBIL, EMISMRG`** complete:
  - Component testing and validation done
  - Integrated system, AQM testing in progress
  - Mobile6 treatment for future?
- Work took longer than anticipated:
  - "Vanilla" **`TEMPORAL`** absurdly complicated—three man-months for what should have been a two-week task
  - M3IO-in-WRF treatment even worse (over eight man-months wasted due to WRF framework mis-design)
- Optional sub-grid scale terrain-error corrections: funded by State of Alabama

Conclusions

- M3IO now works in WRF: some parts easy, data set creation hard.
- M3IO in WRF enables on-line WRF-Chem/SMOKE and off-line WRF/SMOKE/CMAQ.
- WRF-Chem/SMOKE required SMOKE re-engineering; critical parts have been completed.
- Operational use of WRF-Chem/SMOKE requires greatly enhanced SMOKE performance.
- New SMOKE components offer better modularity and comprehensibility, performance, parallelization.