Recent Improvements to the Scalability of the Community Atmosphere Model

Art Mirin

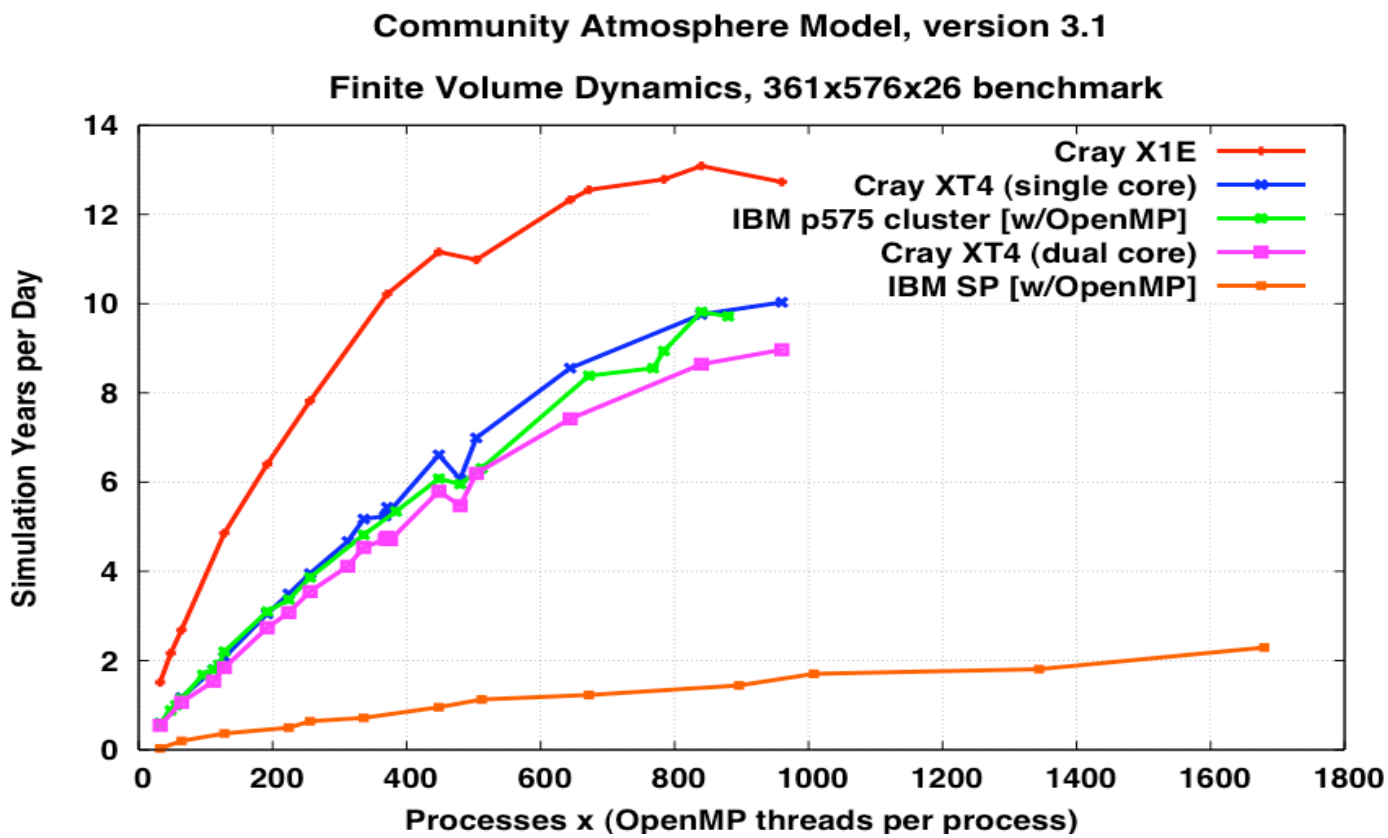
Lawrence Livermore Nat'l. Lab.

Pat Worley

Oak Ridge Nat'l. Lab.

February 12, 2008

CAM scalability is limited



Future computational challenges for CAM

- Support a range of horizontal resolutions efficiently:
  - FV: 2 degree (144x96 grid), 1 degree (288x192 grid), 0.5 degree (576x384 grid), 0.25 degree (1152x768 grid)
- Include atmospheric chemistry, requiring up to a hundred chemical constituents (tracers):
  - increasing both dynamics and physics computation
  - increasing dynamics communication for transposes and tracer advection
- Include cloud resolving physics (increasing physics computation *significantly*)
- Use many more processors efficiently

Today we will discuss…

- Limitations to scalability
- Extending scalability (with a focus on finite-volume dynamical core)
- Future directions

Atmospheric evolution in CAM

- Two primary phases per timestep
  - *dynamics*: advances evolutionary equations for atmospheric flow
  - *physics*: approximates subgrid phenomena, such as precipitation, clouds, radiation, turbulent mixing, …
- Multiple options for dynamics (and more coming):
  - finite-volume (FV) dynamical core
  - spectral Eulerian (EUL) dycore
  - spectral semi-Lagrangian (SLD) dycore

All use tensor product *latitude* x *longitude* x *vertical level* grid over the sphere, but not same grid, same placement of variables on grid, or same domain decomposition in parallel implementation

CAM parallelization strategy

- Domain decomposition, where each subdomain is assigned to a single MPI process; when available, OpenMP is used for additional parallelization
- Dynamics and physics use separate decompositions
  - physics utilizes a fine grain (*chunk*) 2D longitude/latitude decomposition
  - dynamics utilizes multiple decompositions
    - FV: 2D block latitude/vertical and 2D block longitude /latitude, with same number of blocks in each
    - EUL and SLD: 1D latitude in physical space and 1D wave number in spectral space
- Transposes are used to move between decompositions

Original limitations to CAM scalability

- Number of subdomains limited by grid resolution
  - FV dynamics: minimum of 3 grid points in each coordinate direction; with only 26 vertical levels, this severely limited size of the latitude/vertical decomposition
  - EUL and SLD: decompositions one-dimensional, and number of MPI processes constrained by number of latitudes (CAM allows idle processes in the spectral space)
- All domain decompositions required same number of MPI processes
  - number of MPI processes limited by smallest domain decomposition
- Communication cost of transposes, load imbalance, I/O, global diagnostics, … also affect parallel scalability

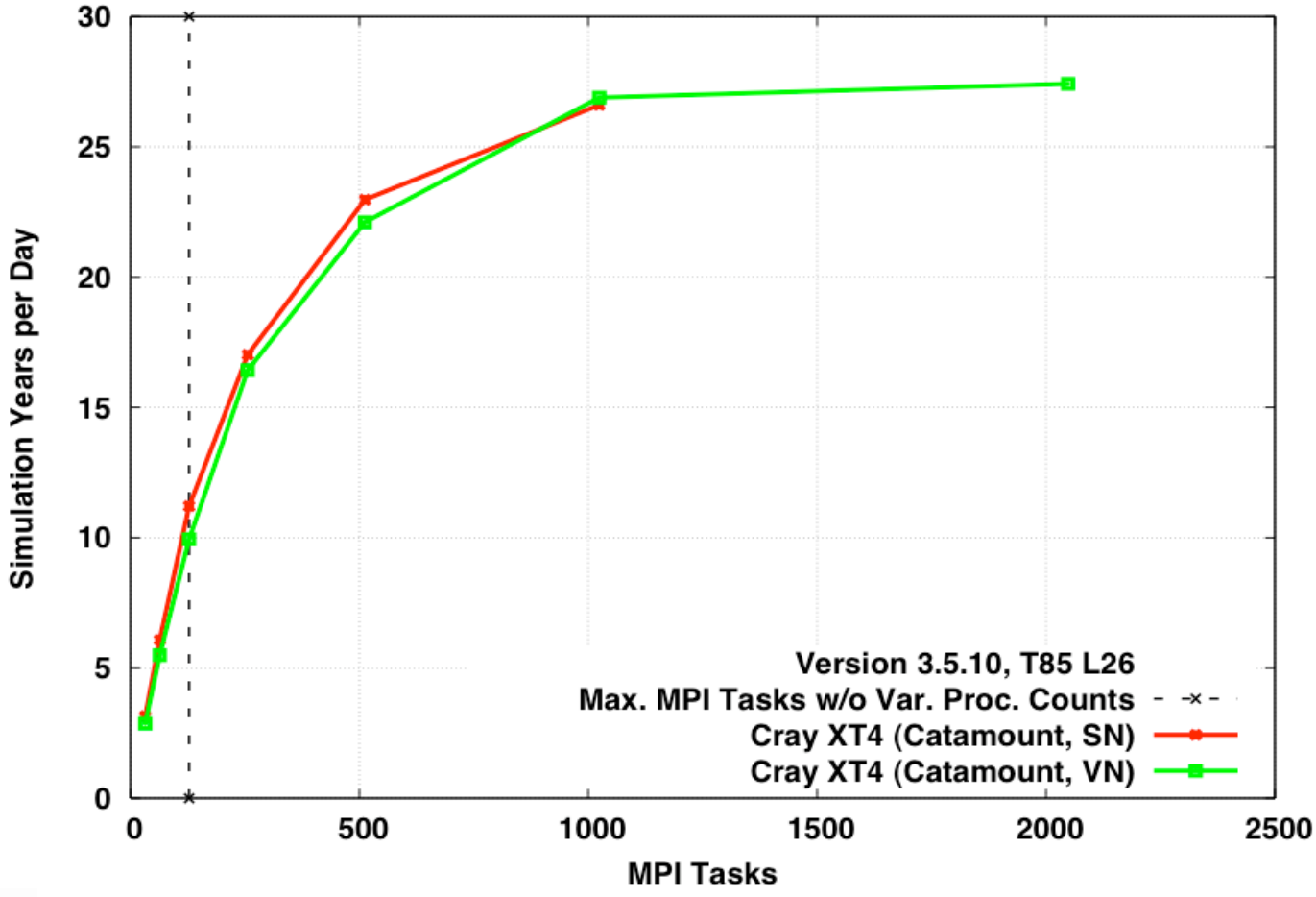
We have extended scalability (and capability)

- Allowing more physics than dynamics processes
- Allowing inactive processes
- Allowing larger vertical decomposition (FV)
- Larger longitude/latitude than latitude/vertical decomposition (FV)
- Overlapping tracer advection with main dynamics (FV)
- Decomposing tracer advection over tracer index (FV)
- Implementing stabilizing terms (Rayleigh friction, additional filtering) to enable tractable operation at 0.25-deg (FV)

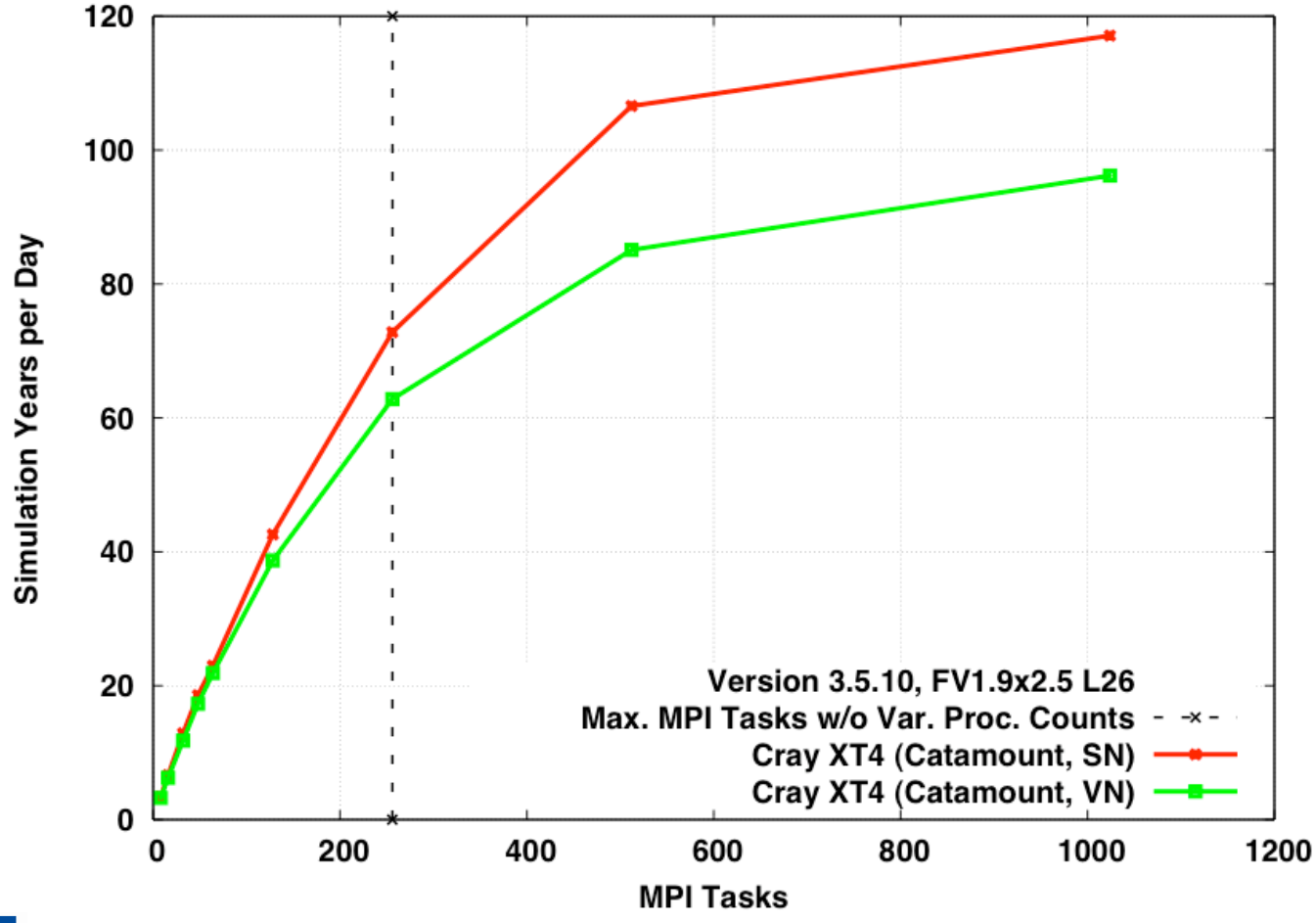
More physics than dynamics processes

- CAM already had capability to distribute physics chunks to processes to achieve better load balance
  - *phys_loadbalance* namelist variable
- The transpose to connect the dynamics and physics (MPI collective call) was generalized to allow the domain and range to have different numbers of processes
  - to use these additional processes, load balancing must be enabled, so *phys_loadbalance* should be nonzero
- The dynamics was generalized to allow processes to not be assigned to subdomains
- Biggest gains are when physics is intense (chemistry, superparameterization)
- FV dynamics benefits more at low resolution

Spectral Eulerian performance



Finite-volume performance



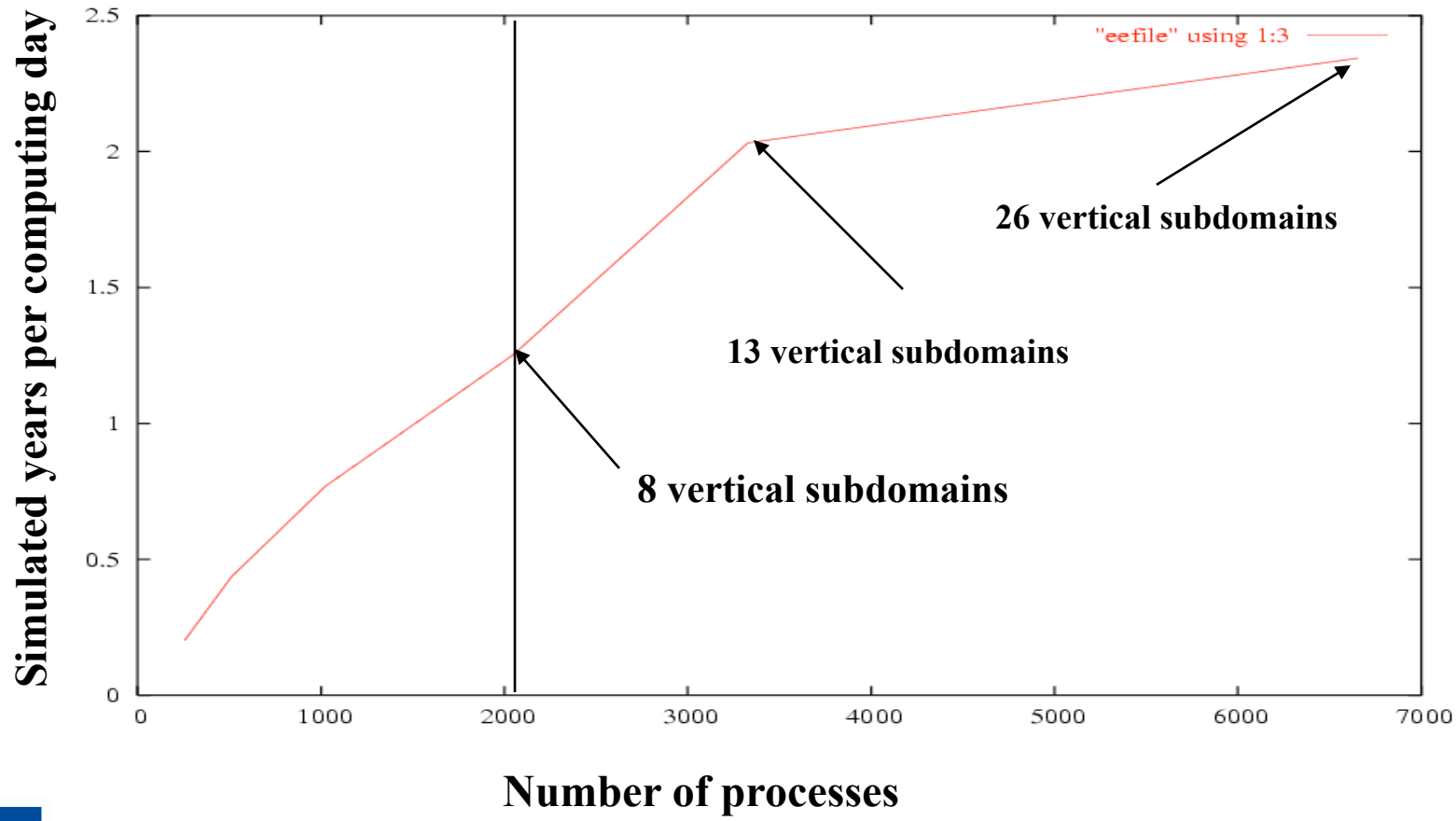
Inactive processes

- CAM now allows processes to be assigned to neither physics nor dynamics
- This can be important when CAM is used within a larger context
  - CCSM sequential execution with low-resolution atmosphere and high-resolution ocean
- Initial approach was to explicitly test for zero chunks and conditionally execute code
- Current approach is to execute without testing
  - compilers generally (but not always) support empty loops, zero-sized arrays, etc.
  - coding is much cleaner

Larger vertical decomposition (FV)

- Original coding required at least 3 levels per vertical subdomain
  - purpose was to avoid potential code breakdowns with small subdomains
  - early scalability analyses indicated no benefit with fewer levels
- One can now have vertical subdomains with a single level
  - essentially no code changes required
  - bit-for-bit agreement for all tests conducted

Throughput at 0.25-deg resolution



Larger lon/lat than lat/vert decomposition (FV)

- Latitude/vertical decomposition used for main dynamics
  - typically restricted by small vertical dimension
- Longitude/latitude decomposition used for flux surface remapping and (sometimes) geopotential
  - columnar operations admit much finer decomposition
- Introduce separate *npes_yz* and *npes_xy* and execute coding conditionally
- Generalize connecting mod_comm transposes to support *npes_xy* > *npes_yz*

Larger lon-lat decomposition, cont.

- Performance tradeoff between increased concurrency and increased communications overhead
  - smaller messages, but more of them
  - typically in latency dominated regime
- Minor performance improvement (typically)
  - results shown later (in this presentation)

CAM evolves with multiple time steps

- Tracer advection (*trac2d*) subcycled with respect to physics
- Main dynamics (*cd_core*) subcycled with respect to tracer advection
- Additional (level-dependent) tracer subcycling (if needed) in *trac2d* (constant winds)

```
do n = 1, nmax
   do k = 1, kmax
      call cd_core
   enddo
   call trac2d
enddo
```

trac2d requires updated winds

outer subcycling (nmax) increases with increasing resolution

Overlap of *trac2d* and *cd_core* subcycles (FV)

- Each *trac2d* subcycle can be overlapped with subsequent (outer) *cd_core* subcycle
- Computational savings greatest at high-resolution and moderate tracer count
- Define auxiliary lat-vert decomposition using second set of *npes_yz* processes
  - use auxiliary decomposition for overlapped *trac2d*
- *Trac2d* requires courant numbers, mass fluxes, pressure thickness, and (at beginning of time step) tracer values
- Use nonblocking communication and overlap to extent possible
  - initial send requests are in last inner subcycle of *cd_core*

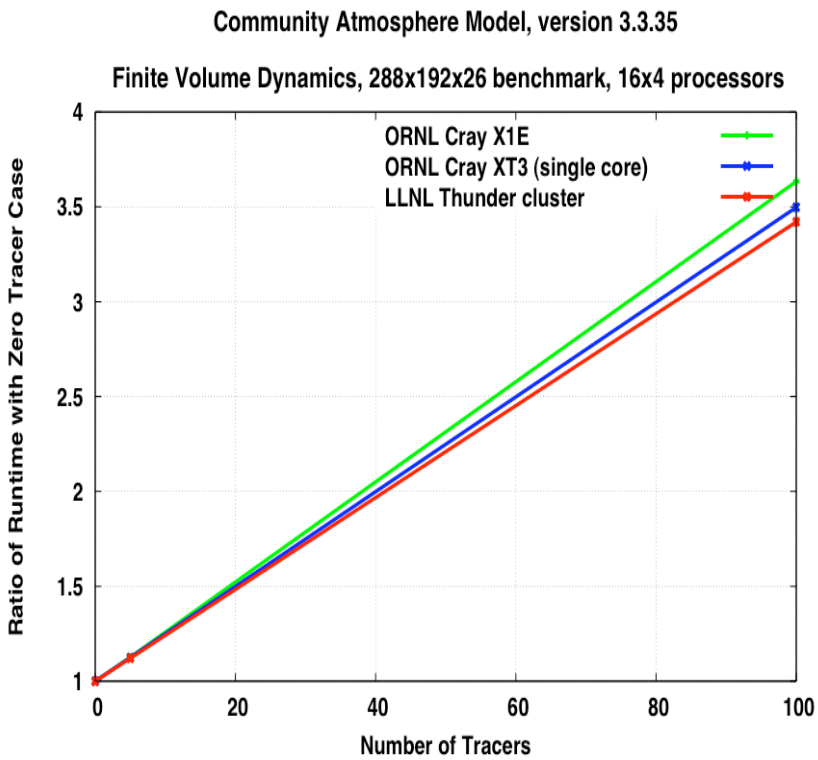
Trac2d/cd_core overlap results

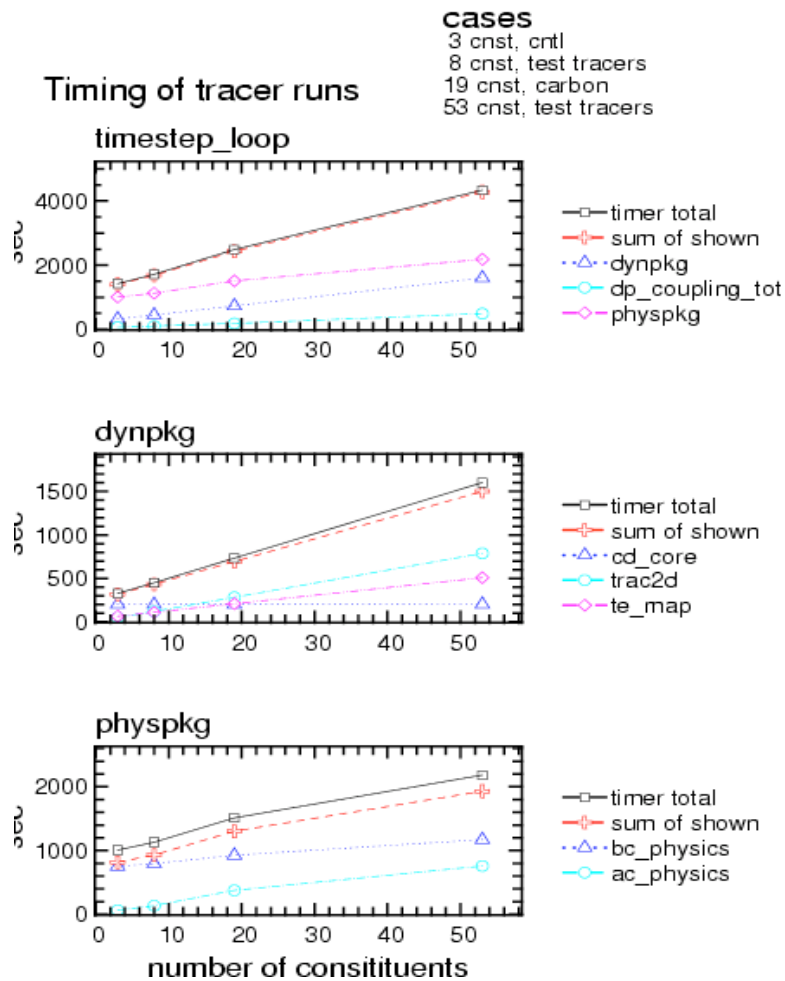
- ***Consider 0.5-deg resolution with 33 tracers (specify 30 extra tracers with tracers_suite)***

feature	Y-lat	Z-vert	*X-lon*	*Y-lat*	npes	overlap	time	improvement
control	128	13	13	128	1664	no	479	---
more phys proc	128	13	13	128	3328	no	435	10%
larger npes_xy	128	13	26	128	3328	no	428	12%
trac2d_ovlap	128	13	26	128	3328	yes	351	36%

Combination of improvements increases throughput by 36%

Each add'l tracer adds ~2.5% to overall run time





Sources of tracer overhead

- For 1-deg resolution, we compare 102-tracer case (using tracers_suite with 99 extra tracers) with control (3 tracers)

code section	3 tracers	102 tracers	add'l cost
ac_phys	1	13	12
bc_phys	24	49	25
p_d_adjust	2	39	37
d_p_coup	3	14	11
xy_to_yz	3	25	22
cd_core	30	32	2
trac2d	3	78	75
te_map	2	17	15
overall	80	322	242

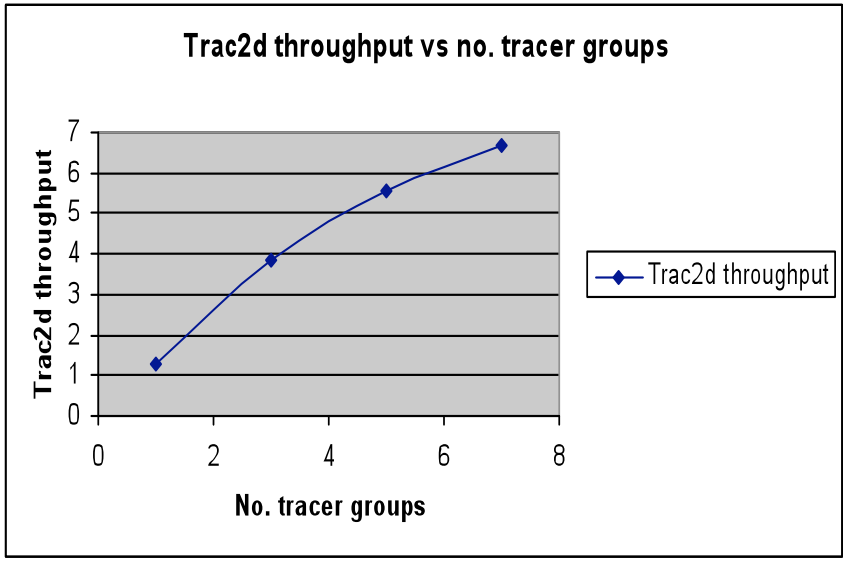
WARNING – timing barriers not used

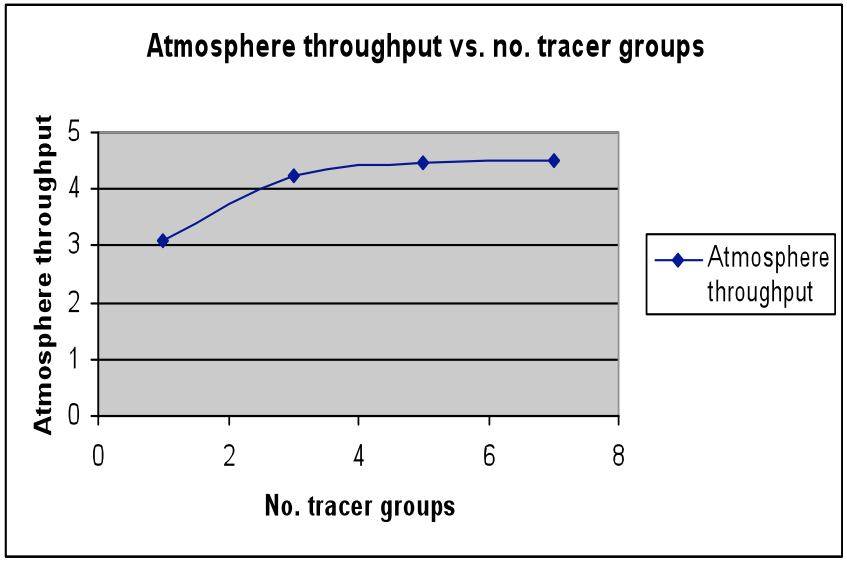
Decomposing *trac2d* over tracer index (FV)

- Applicable to large tracer count at any resolution
- Decompose tracers into *T* groups
- Define *T-1* auxiliary lat-vert decompositions using sets of *npes_yz* processes
- Coding issues similar to *trac2d/cd_core* subcycle overlap
  - most needed quantities must be scattered *T-1* fold
- This option currently can <u>not</u> be used simultaneously with *trac2d/cd_core* subcycle overlap

Tracer decomposition improves throughput

Consider 1-deg resolution, 64x13 (=832 proc) decomposition





The extra processes used for the tracer decomposition affect throughput of other parts of the calculation.

Tracer decomposition improves throughput, cont.

- Consider 1-deg resolution, 64x13 (=832 proc) decomposition

	trac2d	atm run time
control	3	80
99 extra tracers	78	322
5*832 (=4160) proc	78	270
5-way decomp.	18	224

5-way tracer decomposition executes trac2d 4.3 times faster. Overall throughput increases by 44% (this includes benefit of extra processors for physics)

Stabilizing terms at 0.25-deg (FV)

- As resolution increases, CAM develops artificially large high -altitude polar jet in winter
- Top-level January average of zonal wind:

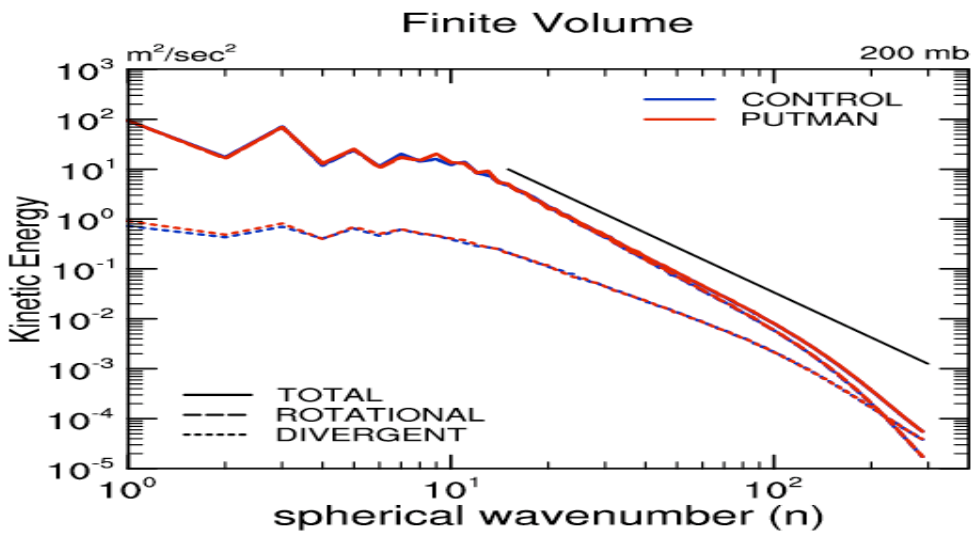
T85	74
1-deg	97
0.5-deg	116
0.25-deg	127

Maximum occurs around 70-deg for all 4 resolutions

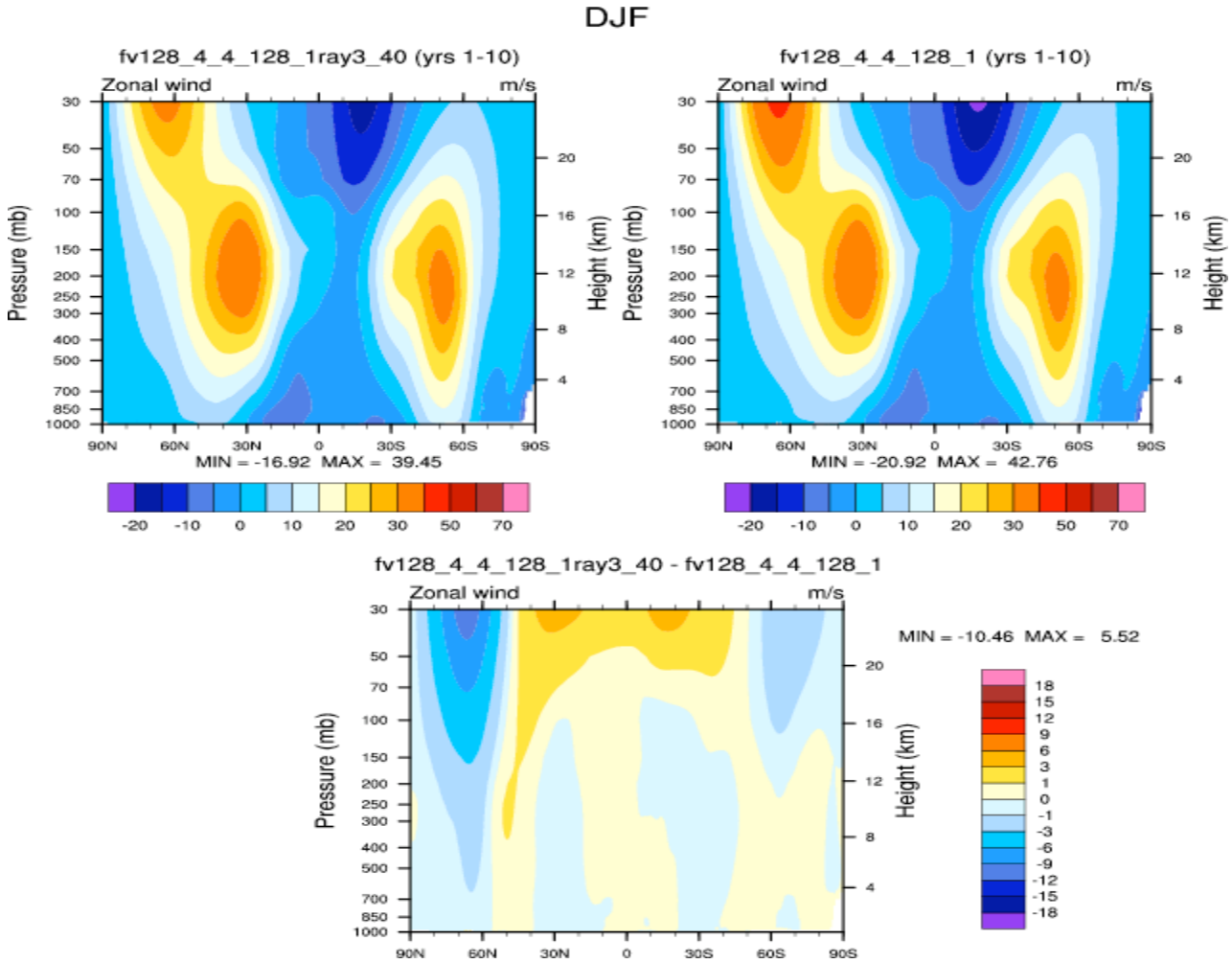
Rayleigh friction and additional filtering

- Rayleigh friction term is energy-conserving and biased at high altitudes
- Additional polar filtering (courtesy Bill Putman) is applied to intermediate winds
- Either approach appears to allow tractable time step while not corrupting solution
  - decadal simulations at 0.5-deg
  - numerical stability tests at 0.25-deg

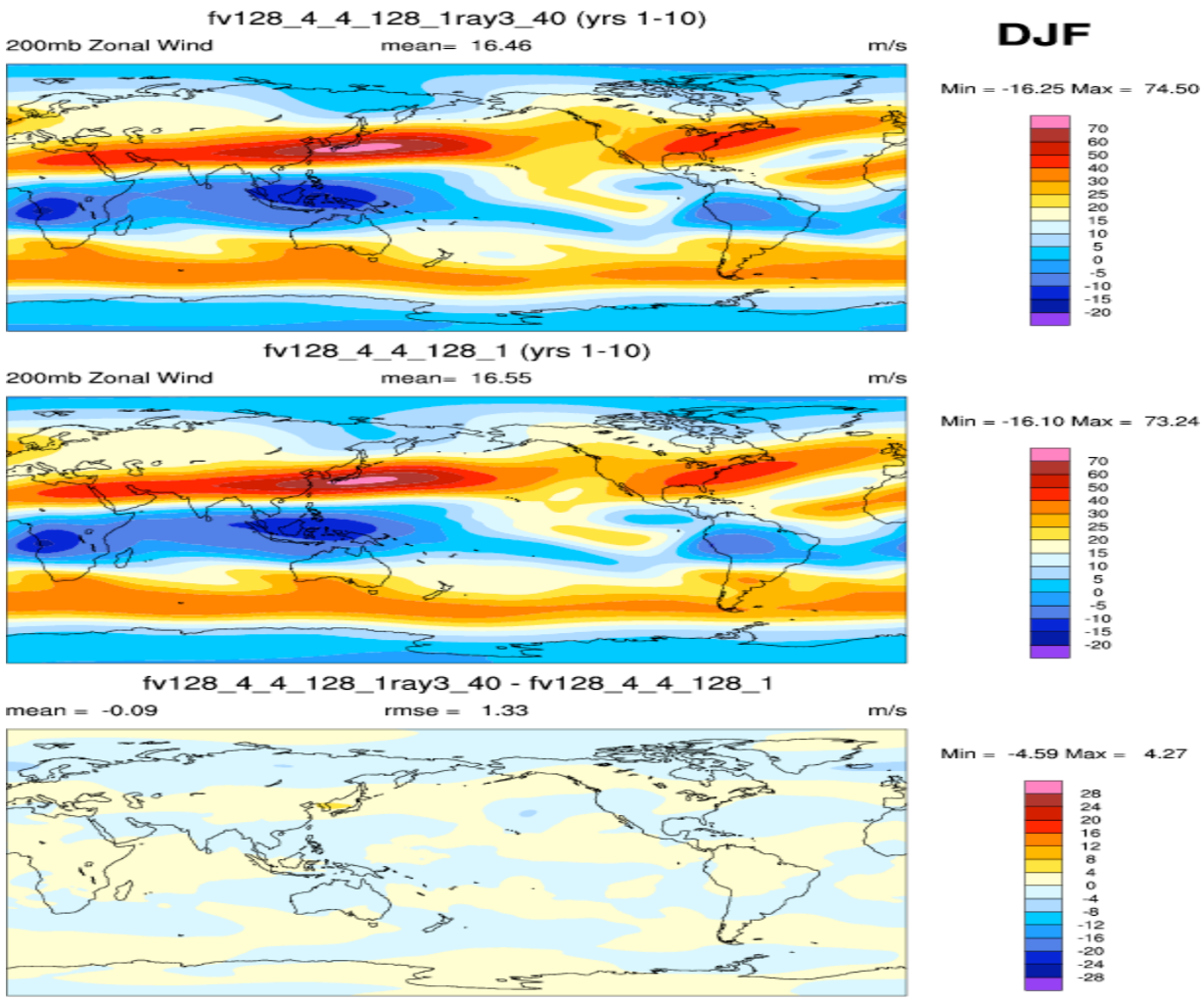
KE spectra: Putman filtering vs control



DJF zonal wind: Rayleigh fric. vs control



200mb zonal wind: Ray. fric. vs control



We have extended scalability (and capability) - I

- **Allowing more physics than dynamics processes**
  - ***intense physics, low resolution FV***
- **Allowing inactive processes**
  - ***code flexibility***
- **Allowing larger vertical decomposition (FV)**
  - ***all resolutions***
- **Larger longitude/latitude than latitude/vertical decomposition (FV)**
  - ***jury still out***

We have extended scalability (and capability) - II

- **Overlapping tracer advection with main dynamics (FV)**
  - ***moderate tracer count, high resolution***
- **Decomposing tracer advection over tracer index (FV)**
  - ***high tracer count, all resolutions***
- **Implementing stabilizing terms (Rayleigh friction, additional filtering) to enable tractable operation at 0.25-deg (FV)**
  - ***not needed at low resolution***

All of these features are on the CAM trunk, except tracer decomposition and larger vertical decomposition

Where do we go from here

- More work needed on large tracer count, larger lon-lat decomposition, chemistry scenarios,…
- Further improvements will be guided by performance characteristics of new benchmarks and new computing platforms
- Scalability improvements to date for standard configurations are useful but relatively modest
  - constrained by polar singularity of standard lon -lat grid
- Breakthrough improvement likely to require alternative grid
  - cubed sphere grid admits to much finer horizontal domain decomposition (e.g., 2-D, space-filling curve)

Acknowledgements

- Research sponsored by the Atmospheric and Climate Research Division and the Office of Mathematical, Information, and Computational Sciences, Office of Science, U.S. Department of Energy under Contract No. DE-AC05-00OR22725 with UT-Battelle, LLC and Contract No. DE-AC52-07NA27344 with Lawrence Livermore National Security, LLC.
- These slides have been authored by contractors of the U.S. Government under contracts No. DE-AC05-00OR22725 and No. DE-AC52-07NA27344, and are released as LLNL Report LLNL-PRES-401144. Accordingly, the U.S. Government retains a nonexclusive, royalty-free license to publish or reproduce the published form of this contribution, or allow others to do so, for U.S. Government purposes.
- This research used resources of the National Center for Computational Sciences at Oak Ridge National Laboratory, which is supported by the Office of Science of the Department of Energy under Contract DE-AC05-00OR22725.