Improving Model Throughput by Parallel Splitting Atmospheric Physics and Dynamics

Peter Caldwell (LLNL)

Mark Taylor (SNL)

AMWG Workshop (2/9/16)

This work performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under Contract DE-AC52-07NA27344.

Background:

For intellectual and numerical tractability, climate models are broken into components:

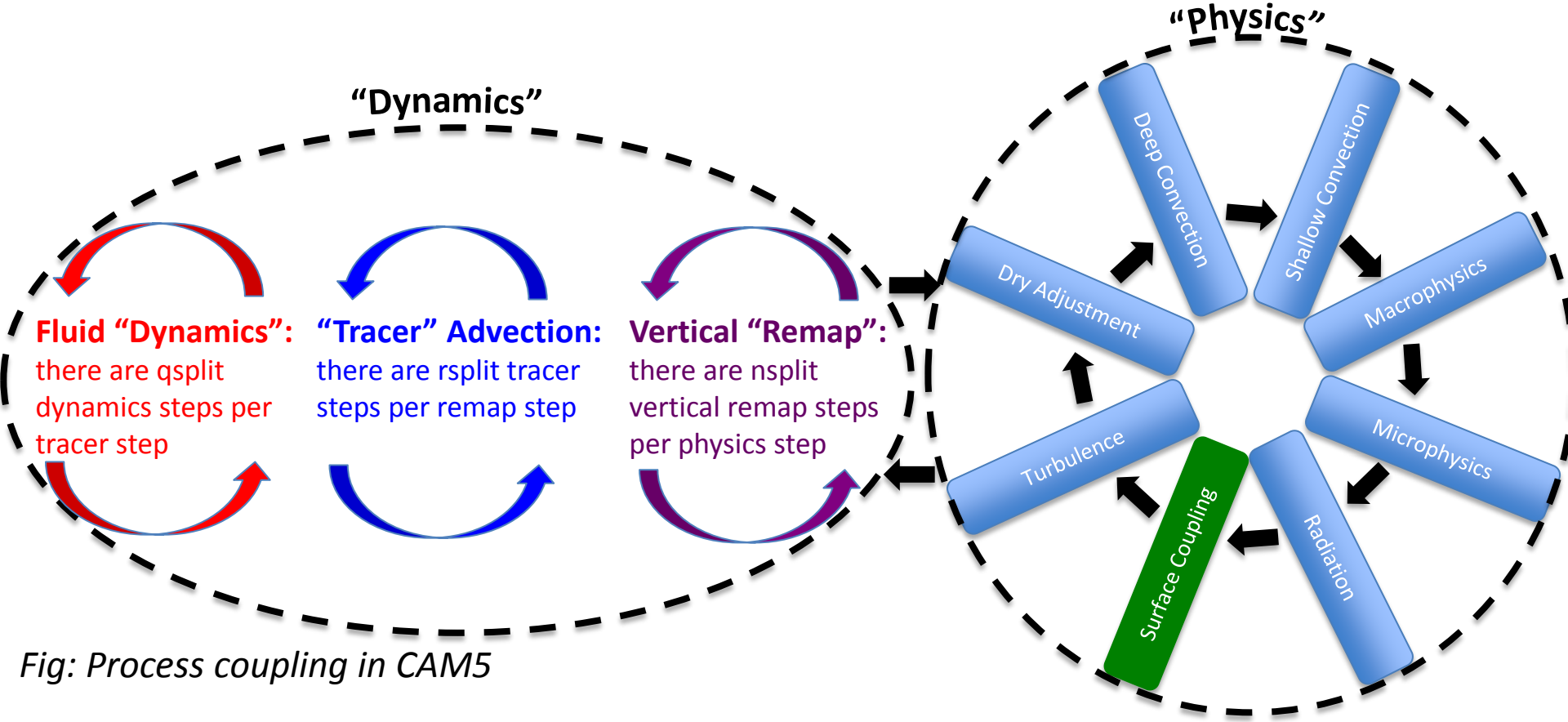


Fig: Process coupling in CAM5

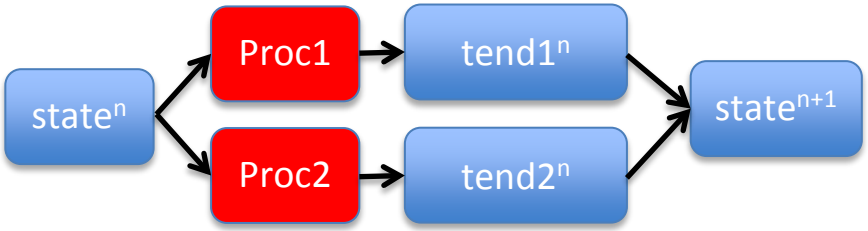
Things to note:

- CAM5 is composed of a variety of processes
- the coarsest granularity is "dynamics" (fluid flow) and "physics" (diabatic processes)
- dynamics uses substepping, so returns a *state* (T=x, q=y...) rather than a *tendency* (dT/dt=x, dq/dt=y...)

Coupling Between Processes

Coupling between processes is fairly crude in GCMs, which is a major source of model error. Typical coupling strategies:

A. Parallel Split (aka process split): All processes are computed from the same state



used in CAM microphysics

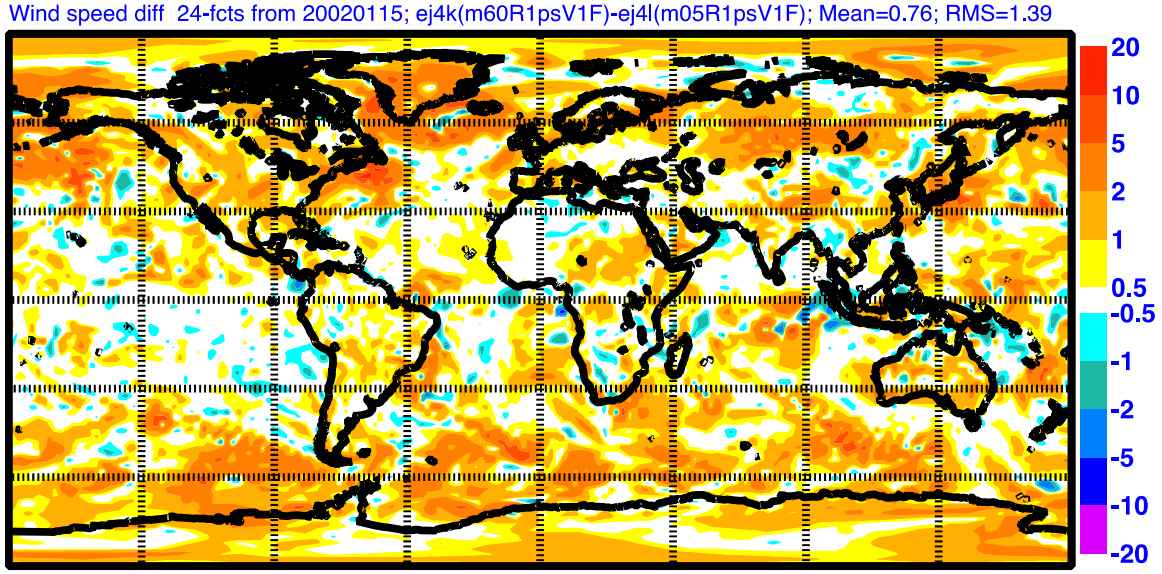
B. Sequential-Update Split (aka time split): The state is updated after each process

used in CAM physics

C. Sequential-Tendency Split: The *tendency* from Proc1 is used by Proc2

used to couple physics and dynamics

Sequential-Tendency Splitting is More Skillful



- Sequential-tendency splitting
  - uses more information
  - couples processes more tightly
  - tends to perform better (see fig)

Fig: 10m windspeed error for (a) parallel-split and (b) sequentially-split versions of the ECMWF model with Δt=60 min (using Δt=5 min as "truth". From Beljaars et al (1991)

But Doing Things in Parallel is Faster!

Dividing work over more cores should increase throughput

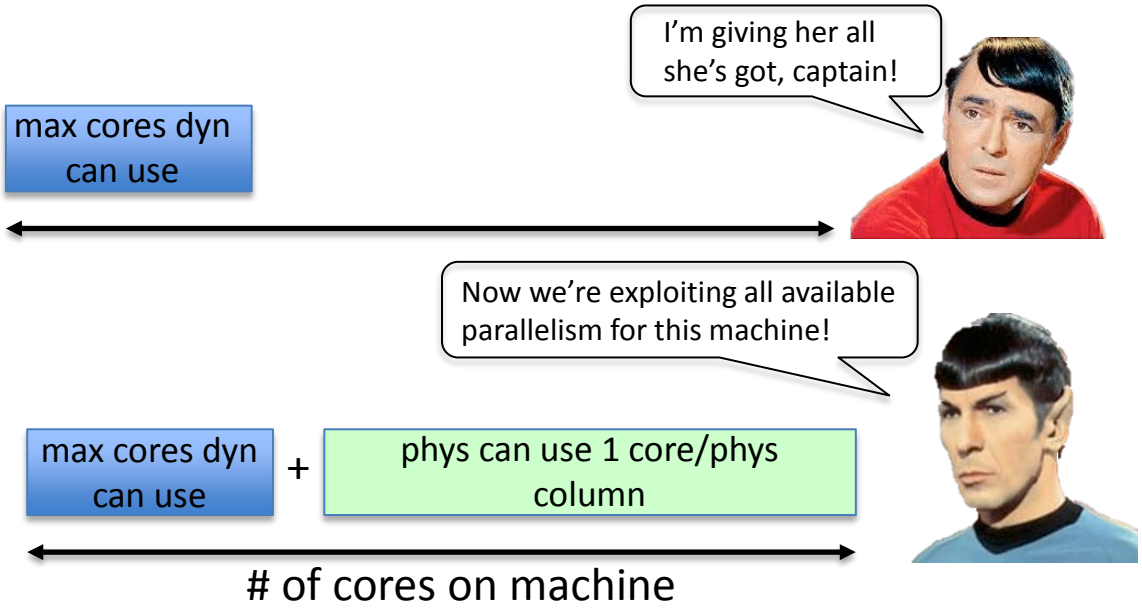


Fig: Max core count in sequential and parallel implementations

Things to note:

- Running phys and dyn in parallel allows us to use more cores
- Since phys requires no columnwise communication, it scales perfectly
  - this allows us to add more sophisticated parameterization with no time penalty

Max cores dyn can use: | # of columns in model | # of cores on machine

ne120 (25 km)	ne120 (25 km)	ne240 (12 km)	Mira	titan	cori p1	cori p2	Aurora/ Summit	Yellow stone	Cheyenne
86K	800K	3M	800K	300K	50K	650K	millions	72,576	145,152

Table: statistics about core usage

Reality Check:

Fig: parallel-splitting enthusiasts

- CAM5-SE scales just fine for all but the biggest core counts
- Running things in parallel makes them *faster*, not *cheaper*

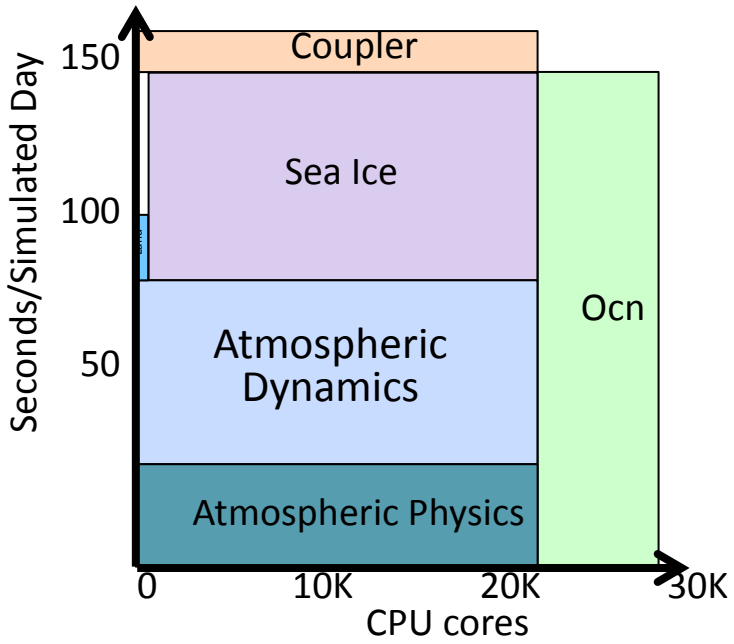
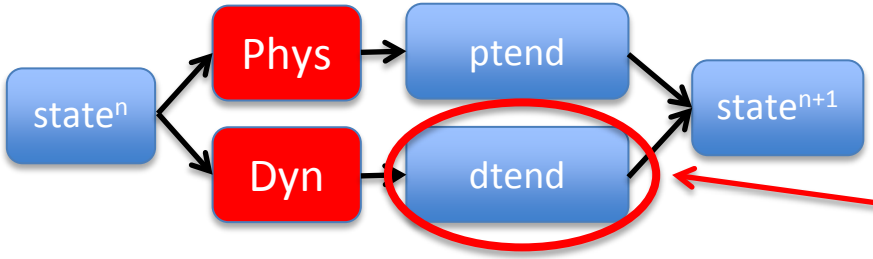


Fig 1: Optimal performance layout for a high-res CESM1 run on Titan

- Completely hiding time spent in phys behind time spent in dyn only reduces atmos run time by ~40% (at ¼° resolution) and total coupled model run time by ~20%

Let's Do This!

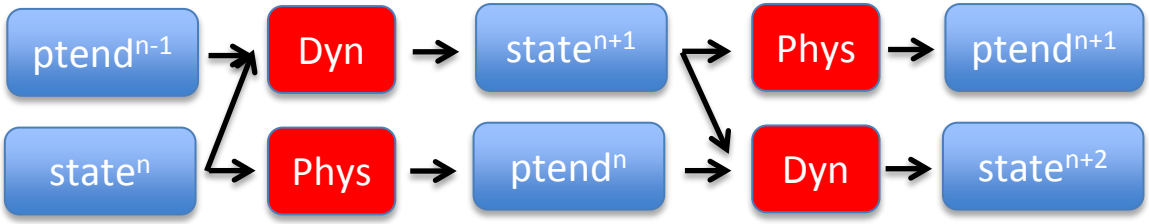
We want:



Oh no! Dyn returns a state, not a tendency

We currently have:

Let's try:



"Lagged-Parallel Splitting": Lag the ptend used by Dyn by 1 step to break dependence between Phys and Dyn.

Lagged-Parallel Splitting

- For initial tests, we lag physics without changing processor layout
- **Results are disastrous** because some physics schemes expect an updated version of the state from last step:
  - macrophysical condensation
  - aerosol activation
  - energy fixer
  - others?

Lesson: be careful of hidden assumptions!

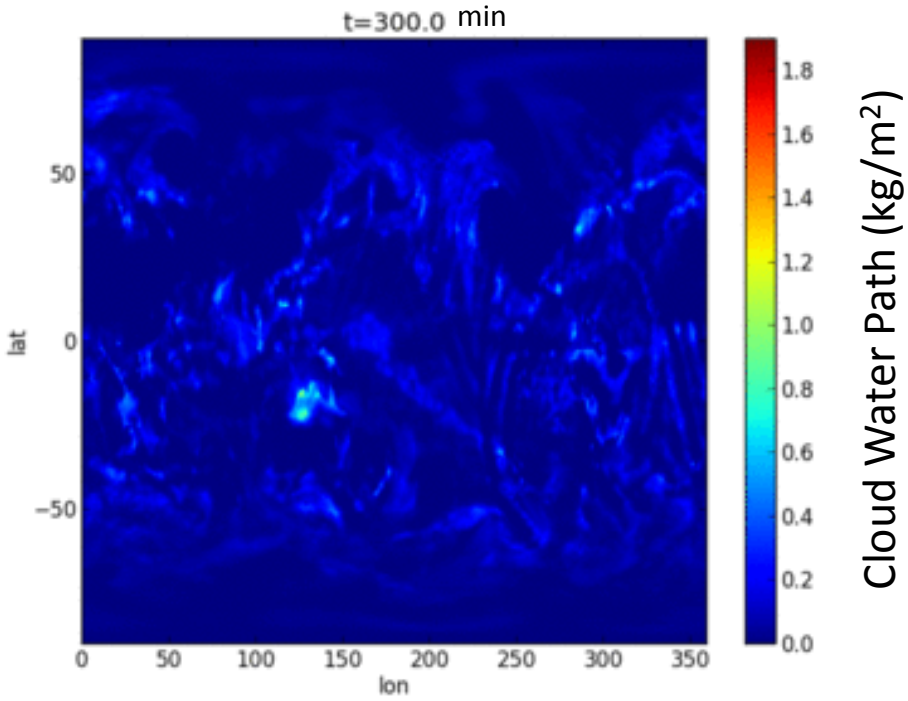
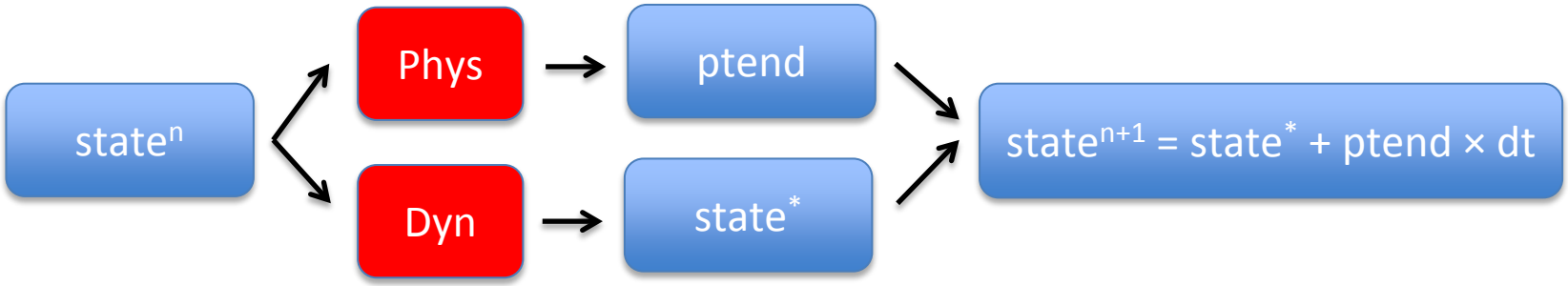


Fig: Cloud water path (TGCLDCWP) every timestep for the 48 timesteps before the model crashes.

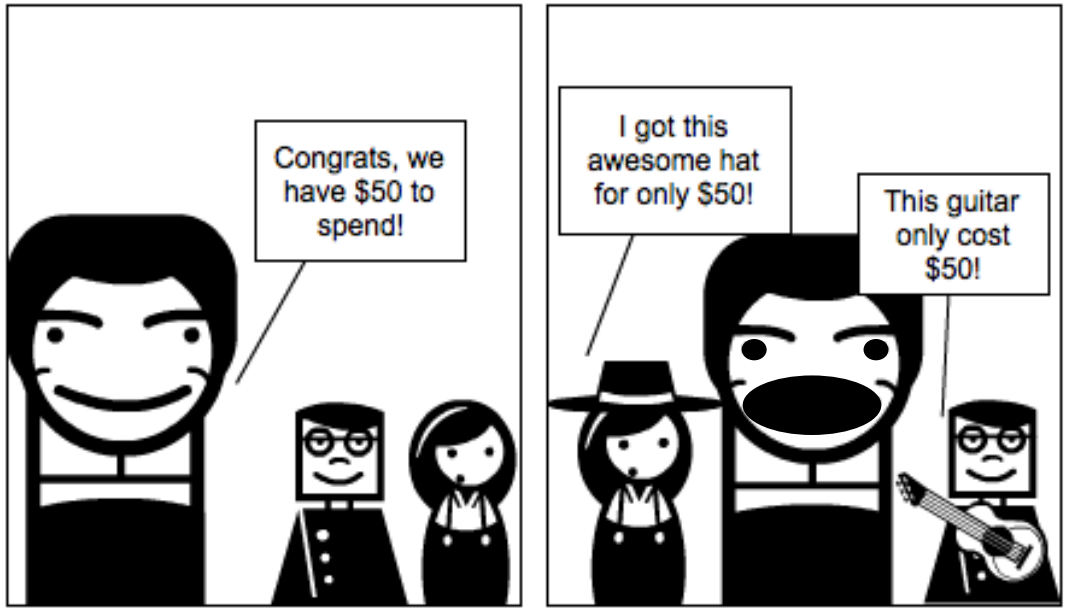
Try 2: Parallel-State Splitting

*Applying ptend **after** dynamics also allows us to parallelize physics and dynamics*



- ptend can no longer be used inside dynamics
- Any timestepping scheme can be used to compute state^{n+1} – what is best?

Parallel-State Splitting - Overconsumption



Parallel yet isolated processes can over-deplete resources when combined.

This problem is encountered any time parallel splitting is used

- if equations are in flux form (moving concentrations from one bin to another), fluxes can be rescaled to prevent negatives without violating conservation (done for microphysics and dynamics)
- at the phys/dyn level, changes are not simple transfers so that won't work. **We currently just set negative values to zero.**
  - This is actually how physical tendencies are already handled in SE dynamics
  - "Clipping" negative tendencies violates conservation, however.
  - Is this acceptable? Is there a better way?

Parallel-State Splitting: Results

- The model seems to run stably and produces something that looks like planet earth.
- Hints of the under-lying grid exist

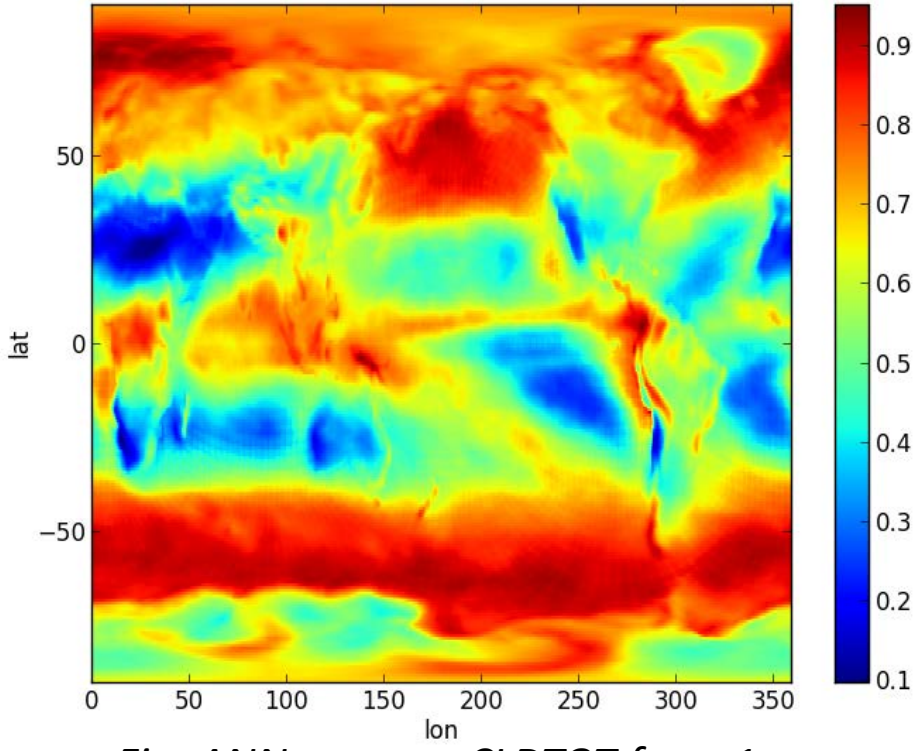


Fig: ANN average CLDTOT from 1 yr simulation with parallel-state splitting.

Conclusions:

- Running physics and dynamics in parallel can speed up high-resolution simulations
  - No time penalty for more sophisticated physics!
- Changing model coupling is hard
  - Constrained by assumptions buried in the code
  - Parallel splitting invites overconsumption/conservation issues