

# Advances in parallel-split dynamics and physics



Aaron S. Donahue and Peter M. Caldwell

Lawrence Livermore National Laboratory

(ACME Software Modernization CMDV project)

ACME All-Hands Breakout #4

June 6, 2017

# Process Splitting: Introduction

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

- The coarsest granularity is “dynamics” (fluid flow) and “physics” (diabitic processes)
- These two processes must be brought together by a loose coupling mechanism
- The three most common are:
  - Sequential-tendency-splitting (STS),  $se\_ftype=0$
  - Sequential-update-splitting (SUS),  $se\_ftype=1$
  - Parallel-splitting (PS),  $se\_ftype=3$  (proposed)

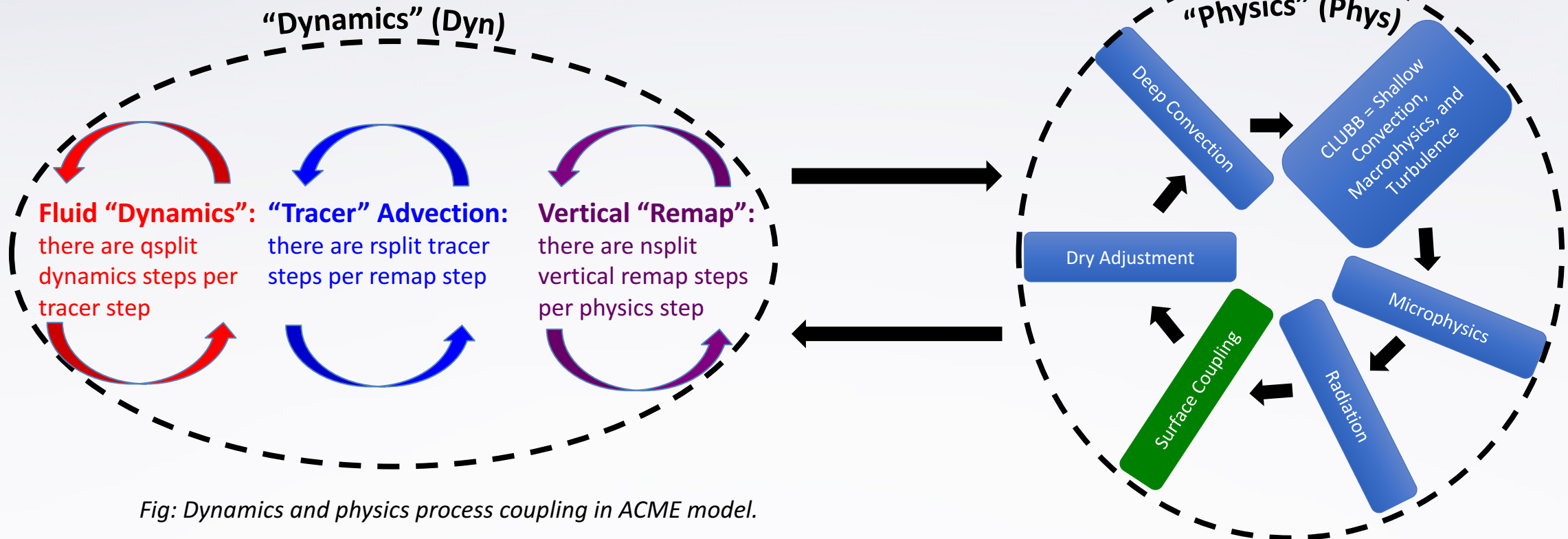
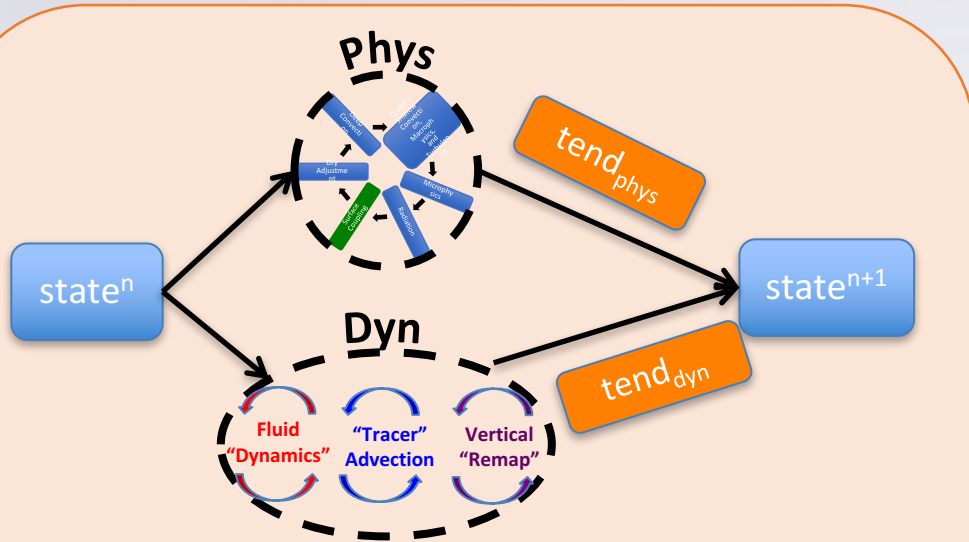
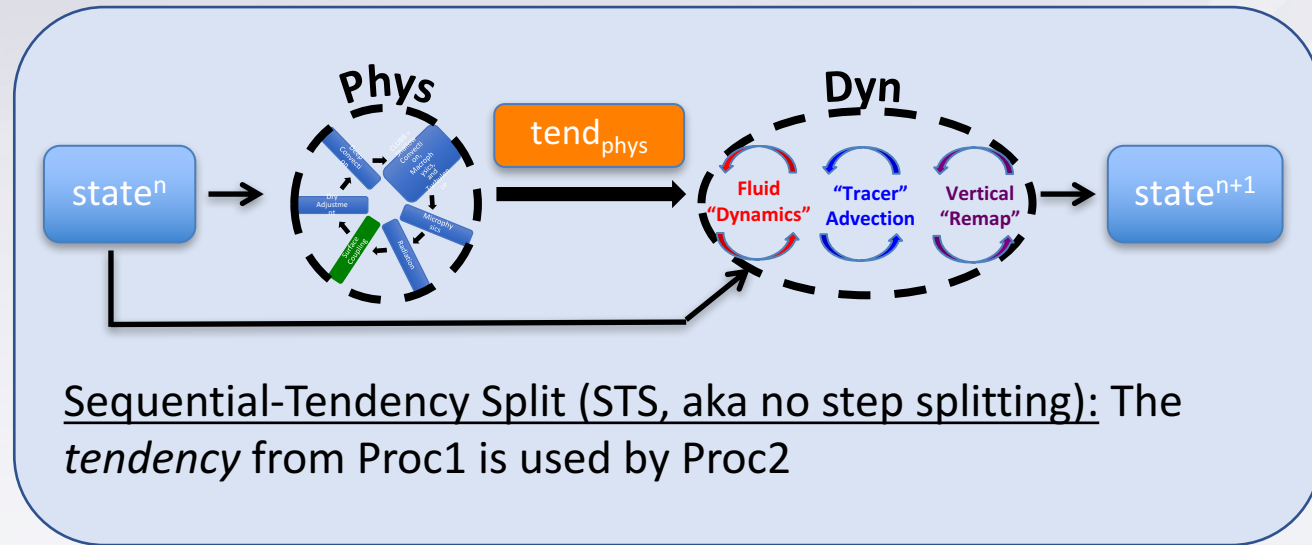


Fig: Dynamics and physics process coupling in ACME model.

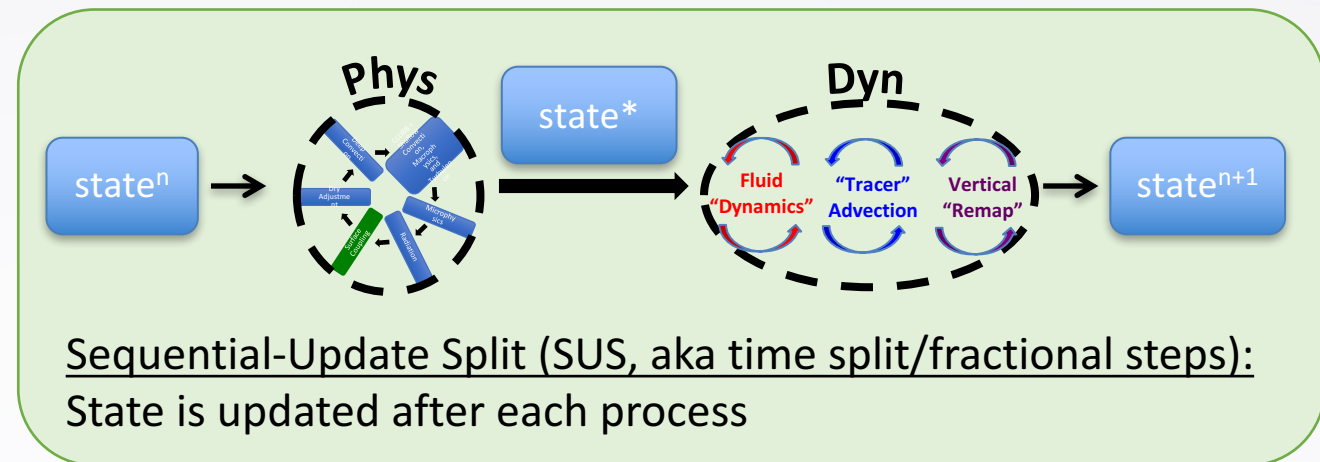
# Process Splitting: Coupling Strategies



Parallel Split (PS, aka process/additive split):  
All processes are computed from the same state



Sequential-Tendency Split (STS, aka no step splitting): The tendency from Proc1 is used by Proc2



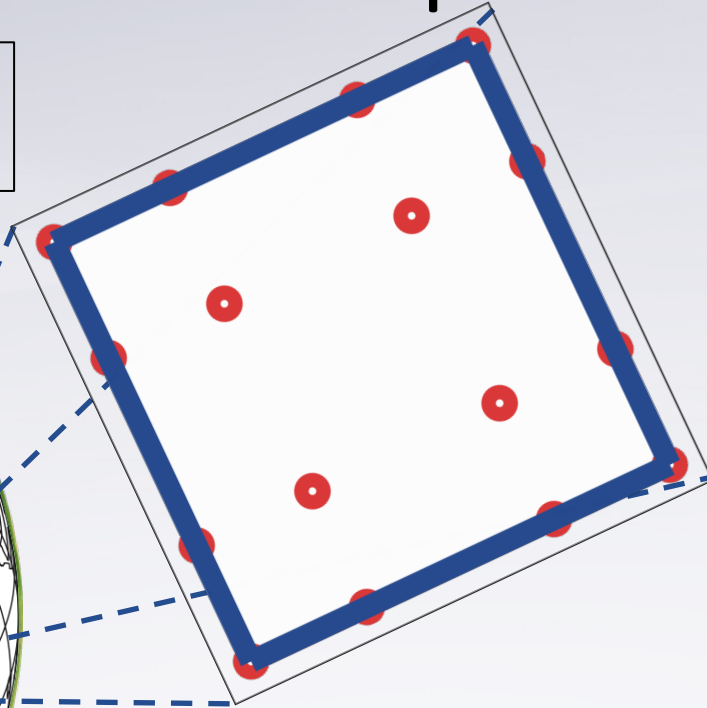
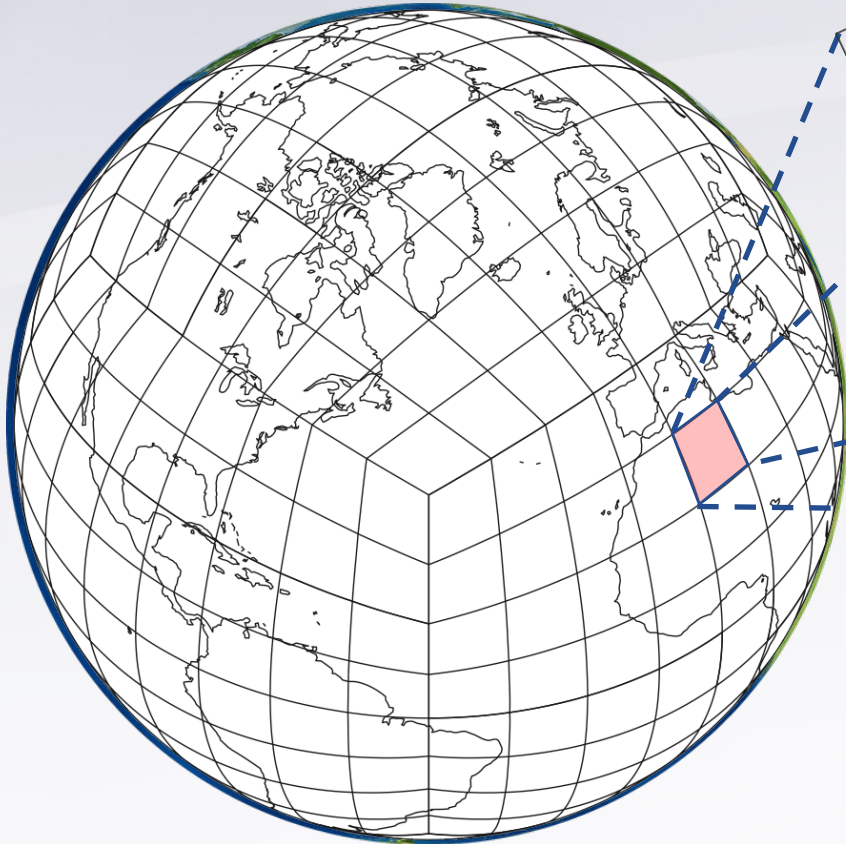
Sequential-Update Split (SUS, aka time split/fractional steps):  
State is updated after each process

# Process Splitting: Domain Decomposition



# Process Splitting: Domain Decomposition

(A) The Earth is divided into a cubed sphere of quadrilateral elements.

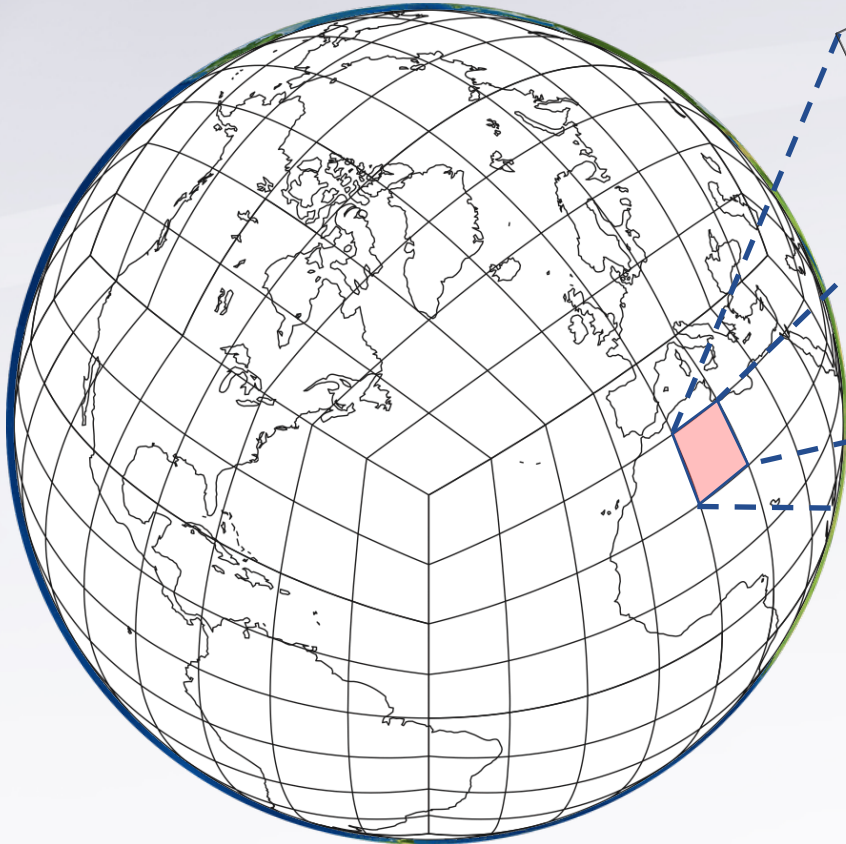


$(1)^0 = 5.4\text{K elements}$   
 $(1/4)^0 = 86.4\text{K elements}$

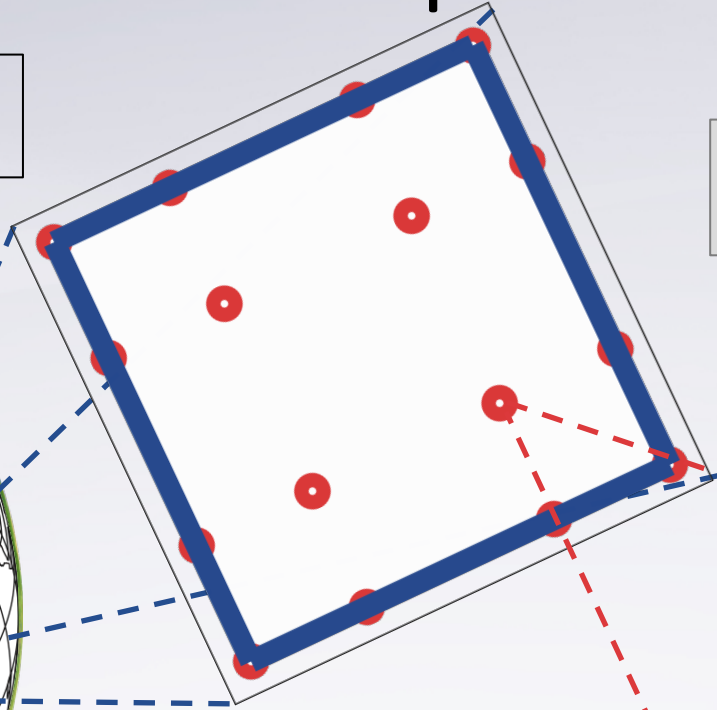
(B) Dynamics is solved on individual spectral elements.

# Process Splitting: Domain Decomposition

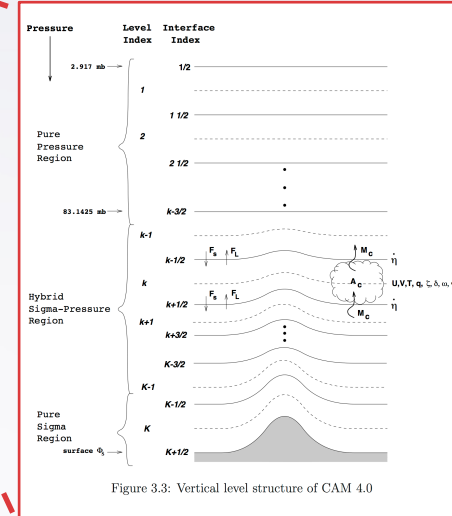
(A) The Earth is divided into a cubed sphere of quadrilateral elements.



(B) Dynamics is solved on individual spectral elements.



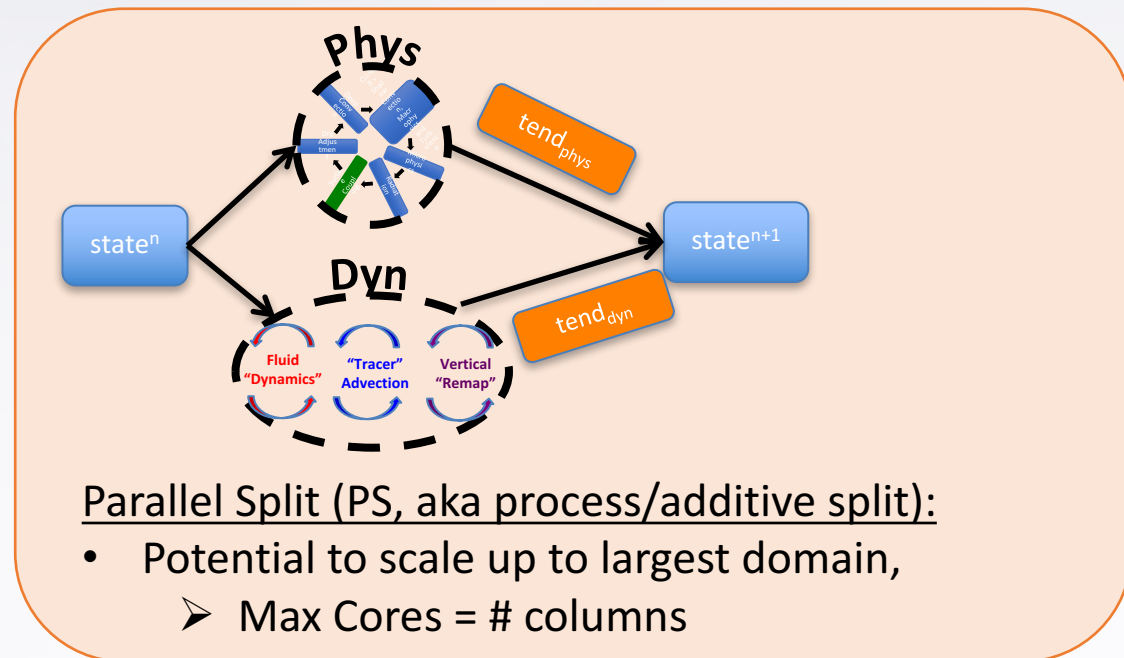
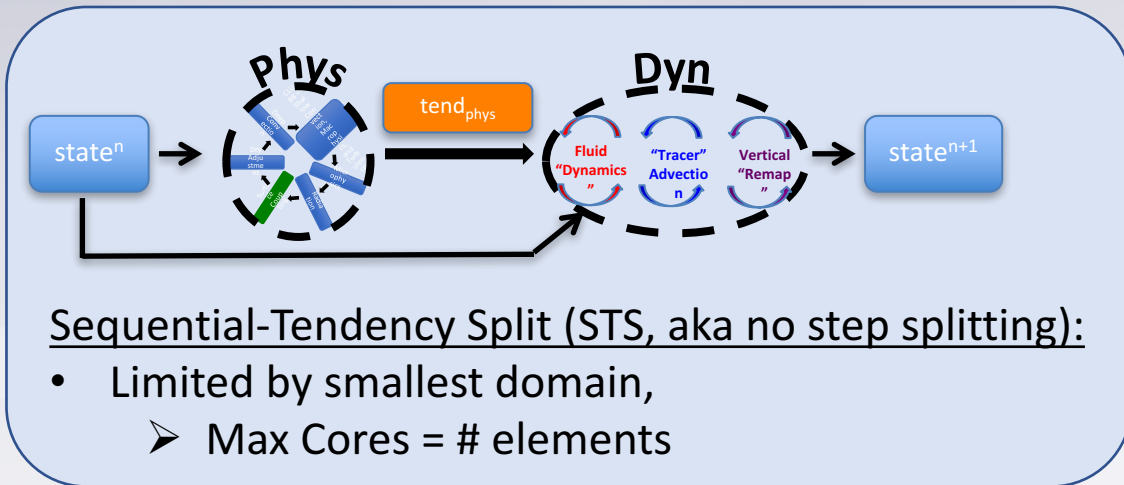
$(1)^0 = 5.4\text{K elements, } 48.6\text{K columns}$   
 $(1/4)^0 = 86.4\text{K elements, } 777.6\text{K columns}$



(C) Physics is solved over a set of columns defined by the Gauss-Lobatto points of a spectral element.

Figure 1: Dynamics and physics domains for the ACME model. (A) cubed sphere, (B) example spectral element, (C) example physics column. Image credit: Dennis *et al.* (2012) Int. J. of High Performance Computing Applications (A and B) and Neale *et al.* (2010) CAM 4.0 (C)

# Process Splitting: Current Scalability



	<i>Dynamics</i>	<i>Physics</i>
$(1)^\circ$	5.4K elements, 48.6K columns	
$(1/4)^\circ$	86.4K elements, 777.6K columns	

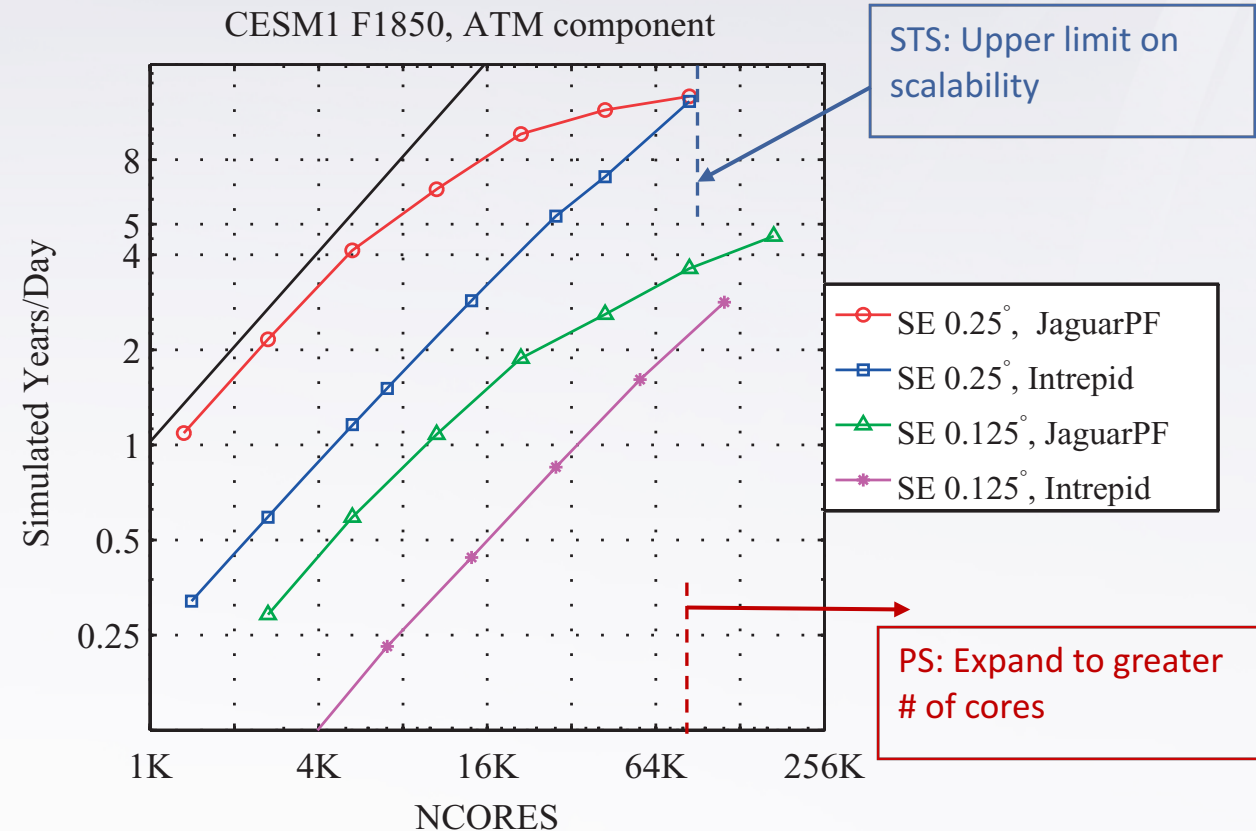
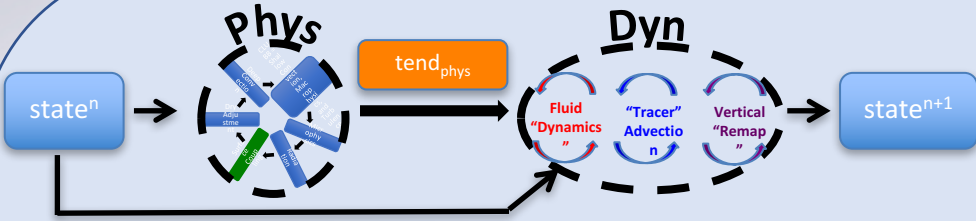
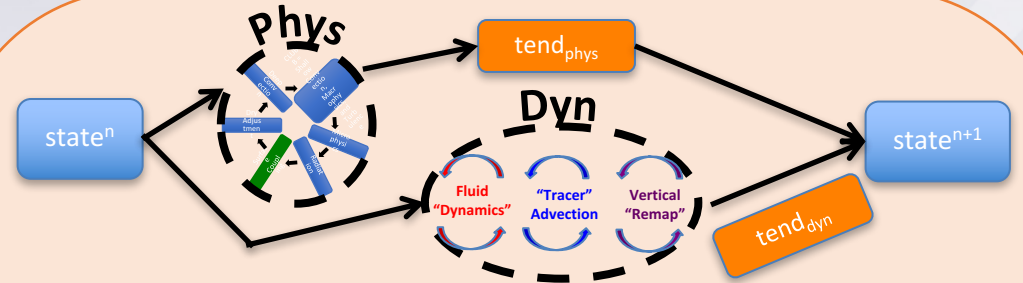
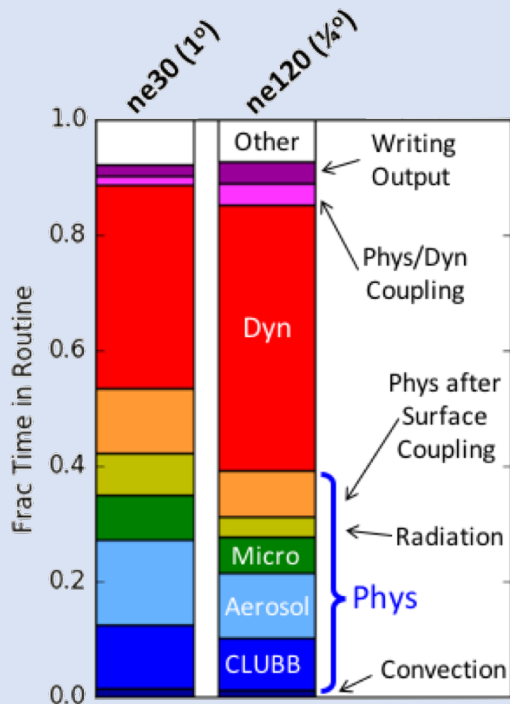


Fig: Scalability of CAM-SE, Dennis et al., "CAM-SE: A scalable spectral element dynamical core for the Community Atmosphere Model" (2012), Int. J. of High Performance Computing Applications.

# Process Splitting: Current Scalability



Sequential-Tendency Split (STS, aka no step splitting):



Parallel Split (PS, aka process/additive split):

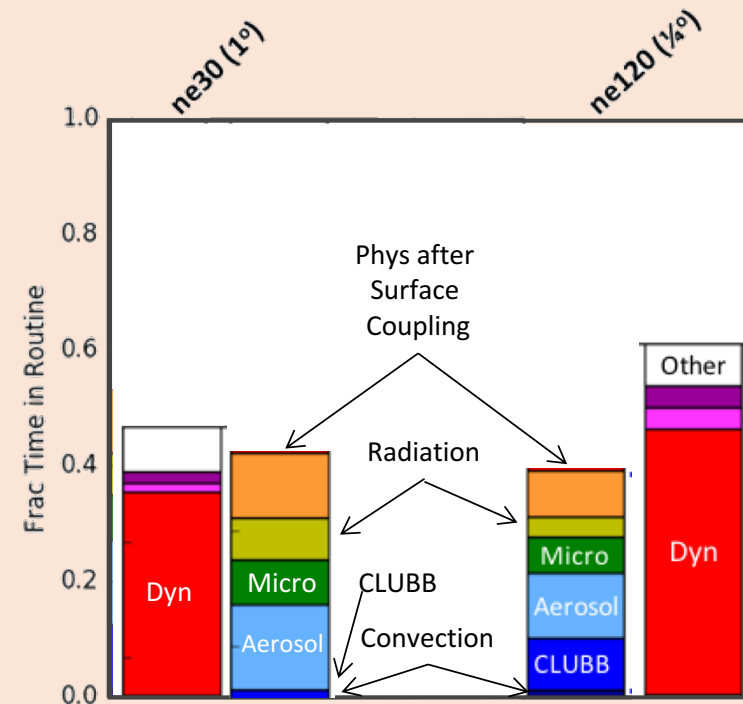


Fig: Fraction of ACME v1alpha7 integration time spent in various processes.



# Parallel-Split: Implementation

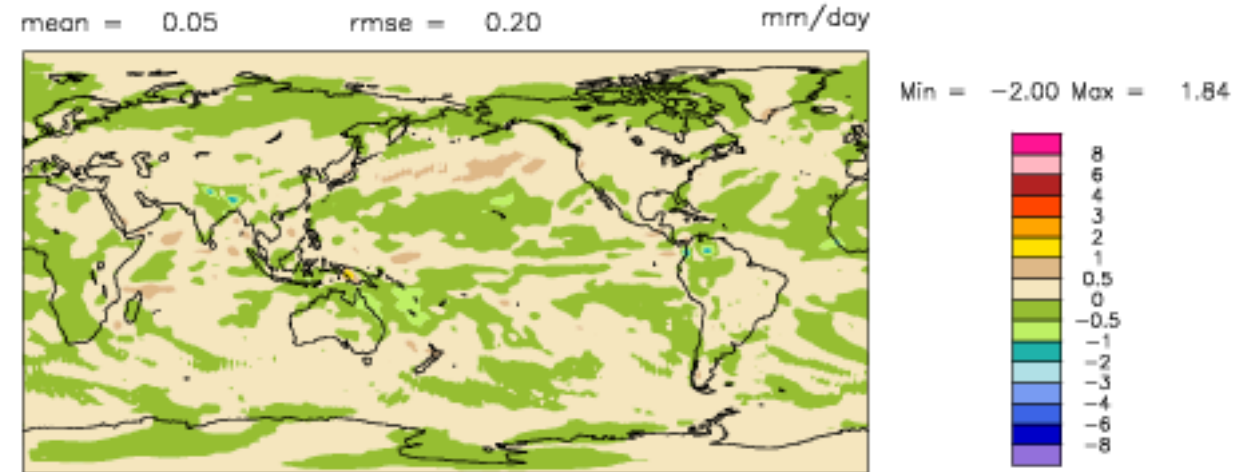
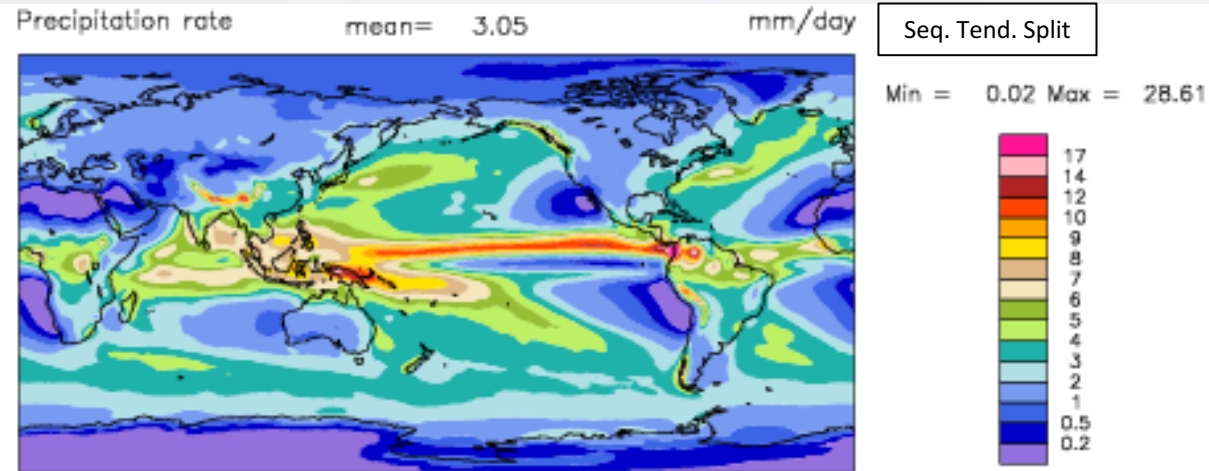
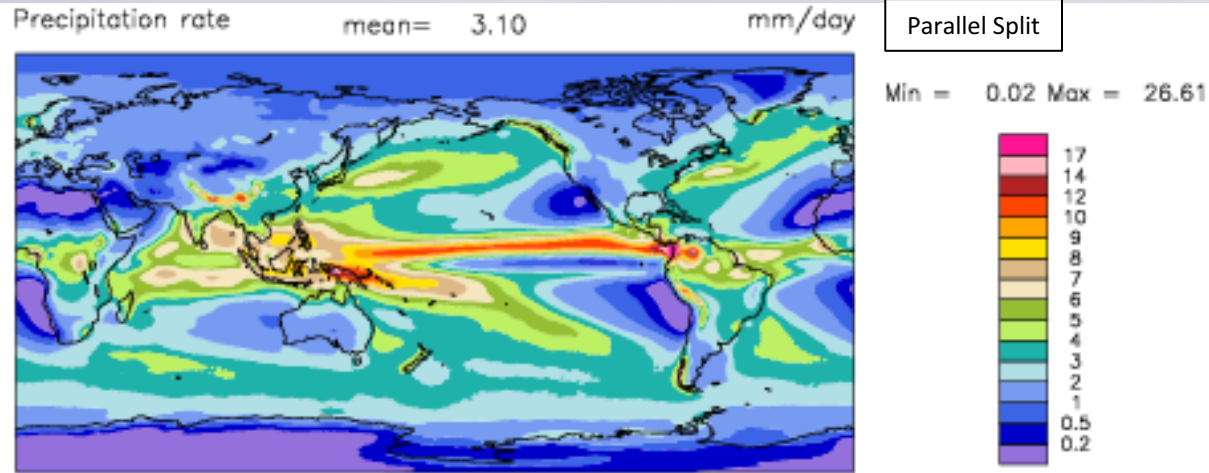
1. Bugfix to allow for dynamics to be solved on a subset of total atmosphere model cores. Namelist variable *dyn\_npes* now works on master. (PR #1393)
2. Adjust *phys\_grid* subroutine to only assign columns “chunks” to physics solving cores.

# Parallel-Split: Potential Issues

1. Bugfix to allow for dynamics to be solved on a subset of total atmosphere model cores. Namelist variable *dyn\_npes* now works on master. (PR #1393)
  2. Adjust *phys\_grid* subroutine to only assign columns “chunks” to physics solving cores.
1. Is there a degraded solution due to change in dynamics/physics coupling mechanism?
  2. How do we handle the mass conservation violations inherent in using a parallel-split approach?
  3. How to implement within current code infrastructure?
  4. Do we actually accomplish improved performance?

# Parallel-Split: degraded solution? **Solution looks good**

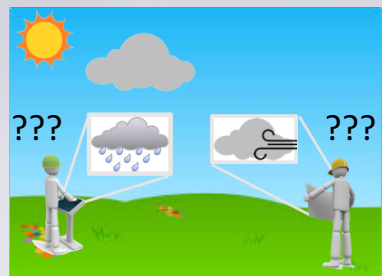
- We have implemented the parallel splitting technique in ACME v0 for dyn and phys, using the same computational cores for both processes, on a  $1^\circ$  domain.
- 10 years of simulation have shown that the method is stable, provided that  $\Delta t = 900s$ .
- Comparison with sequential-tendency splitting (default) shows good results!



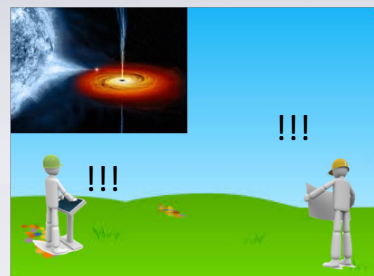
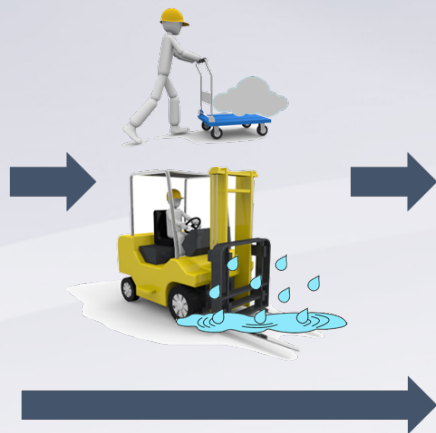
(Parallel Split) - (Seq. Tend. Split)

Fig: precipitation rate from 10 year ACMEv0 runs with; parallel-state splitting (top), sequential-tendency splitting (bottom) and difference (right).

# Parallel-Split: mass conservation? New approaches are promising



Possible to have fluxes that remove more mass than is available.



Leading to negative mass in an element.

- A. Clipping:** Setting all negative masses to zero.
- B. Weighted Horizontal Distribution:** Drawing mass from neighboring nodes horizontally.
- C. Weighted Vertical Distribution:** Drawing mass from neighboring levels vertically.
- D. Full Element Distribution:** Drawing mass from all points within an element.

# Parallel-Split: mass conservation? New approaches are promising



Possible to have fluxes that remove more mass than is available.

Leading to negative mass in an element.

- A. Clipping:** Setting all negative masses to zero.
- B. Weighted Horizontal Distribution:** Drawing mass from neighboring nodes horizontally.
- C. Weighted Vertical Distribution:** Drawing mass from neighboring levels vertically.
- D. Full Element Distribution:** Drawing mass from all points within an element.

Clipping increases the liquid cloud water mass by  $\sim 0.7\%$  per timestep.

Weighted distribution methods dramatically improve the mass conservation properties.

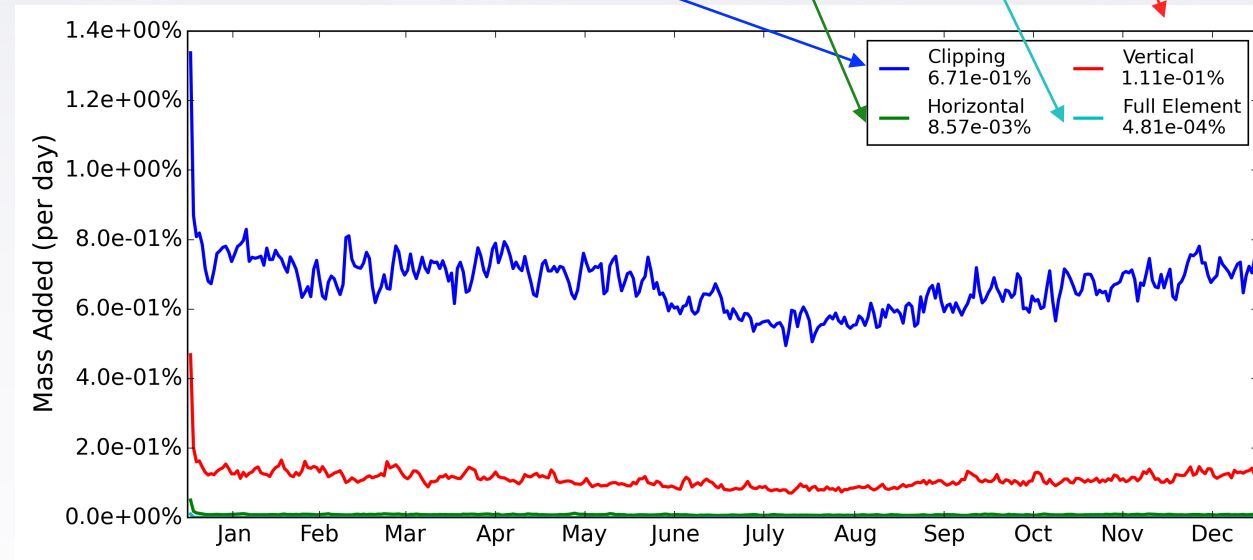


Fig: Daily average mass conservation corrections

# Parallel-Split: implement? **Yes, almost...**

Limit at # of elements, fixed for standard model in PR #1393.

Kinks in performance evidence of poor physics column load balancing.

Standard approach flattens out early on, while parallel-split continues to improve, outperforming the standard approach at maximum core count.

**We are still having coding issues with producing output.**

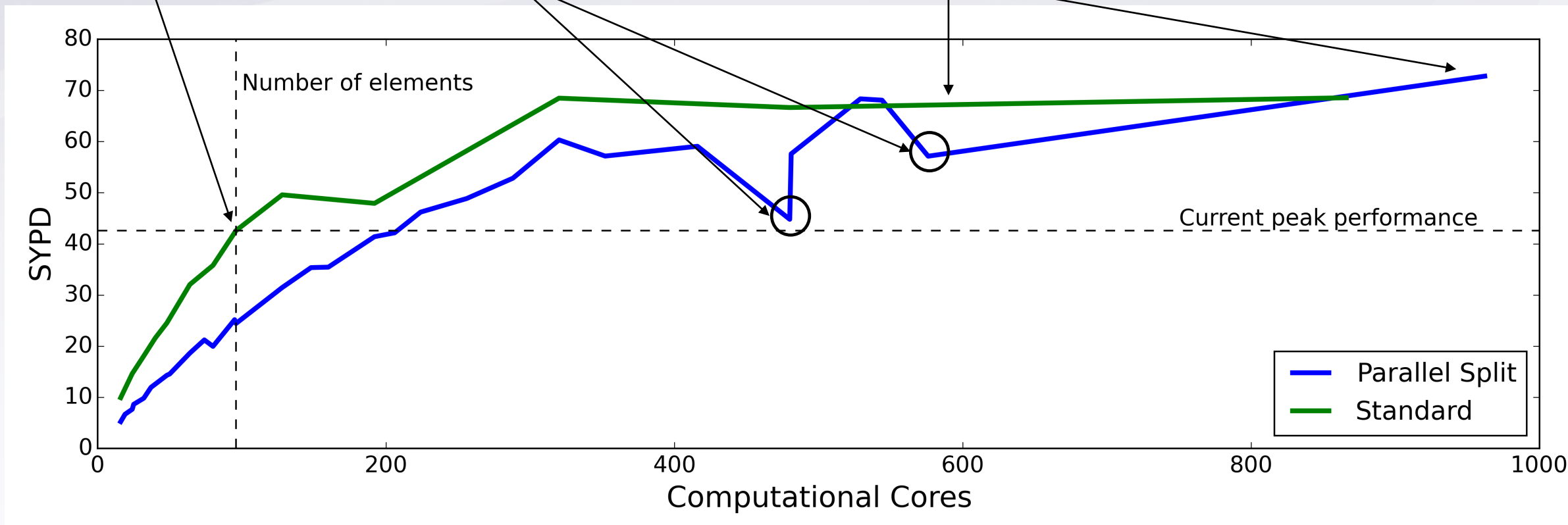


Figure 6: Solution timings for 10-day simulations with no output on the 7.5 degree (ne4) mesh for the standard model and for parallel-split implementation.

# Parallel-Split: improved performance? **Not yet...**

We see a degradation of the model scalability for core counts less than the number of elements for the parallel-split implementation.

We do not see improved performance for core counts greater than the number of elements, STS still outperforms PS except at the max core count option or a few "optimum" cases.

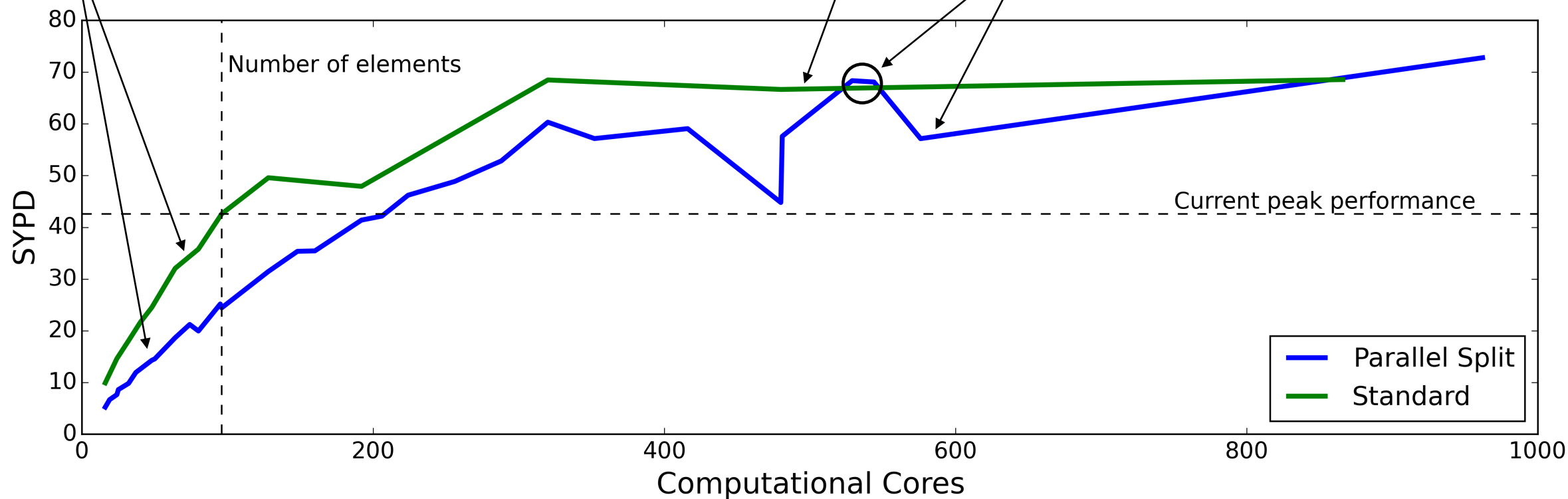


Figure 6: Solution timings for 10-day simulations with no output on the 7.5 degree (ne4) mesh for the standard model and for parallel-split implementation.

# Parallel-Split: improved performance? Solution ideas:

- A 50/50 split of dynamics and physics cores is inefficient and leads to **dynamics cores sitting idle** for long periods.
- Improved performance in terms of solving physics and dynamics separately is traded for **increased communication costs**.

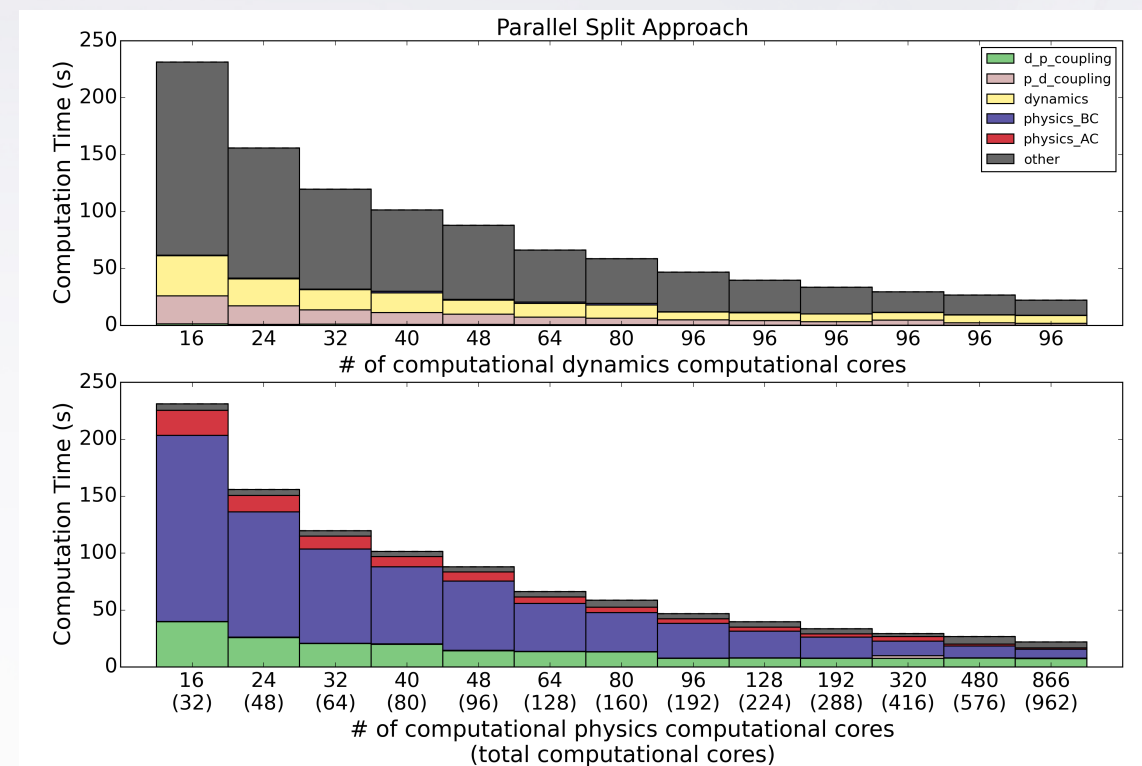
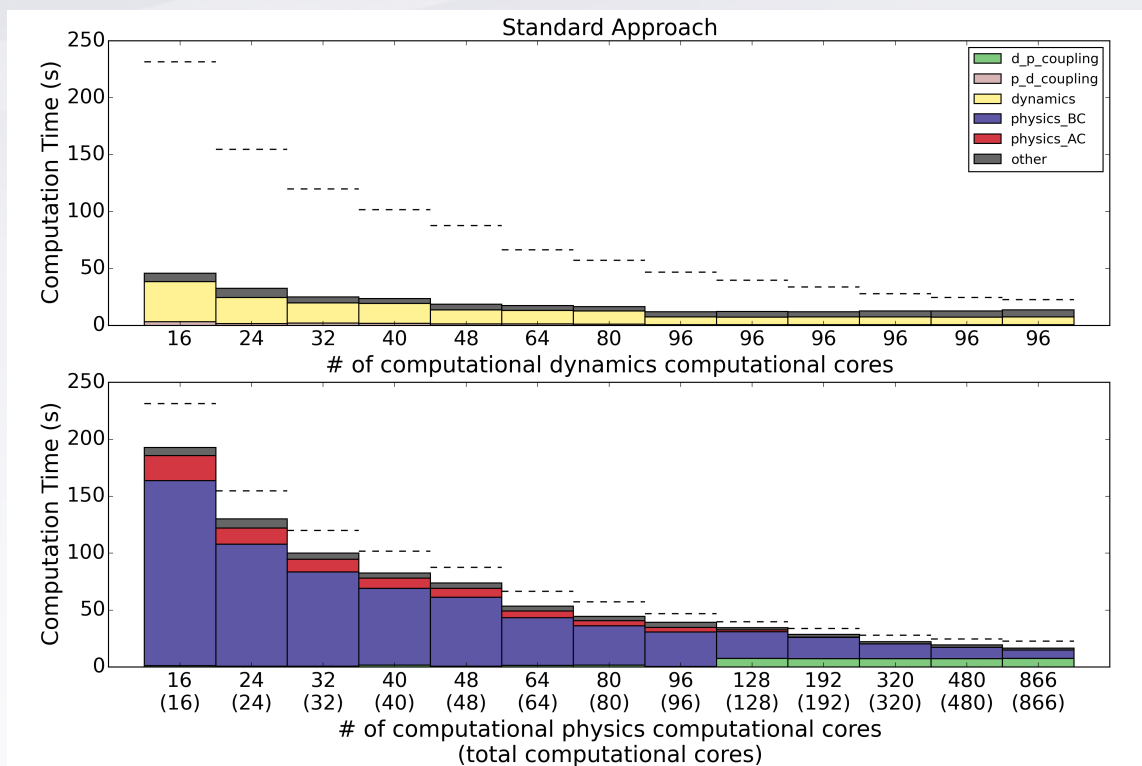


Figure 7: Average computational cost per core for dynamics (dynamics), physics (physics AC and BC) and dynamics-physics communication (d\_p and p\_d coupling) for the standard ACME model (left) and the parallel-split implementation (right). Top panels represent only the cores assigned to dynamics, bottom panels are cores assigned to physics.



# Parallel-Split: improved performance? Solution ideas:

- A 50/50 split of dynamics and physics cores is inefficient and leads to **dynamics cores sitting idle** for long periods.
- Improved performance in terms of solving physics and dynamics separately is traded for **increased communication costs**.

- Determine and implement **optimum** balancing of dynamics and physics computational cores.
- Implement a more sophisticated distribution of cores assigned to dynamics and physics such that most dynamics/physics communication is **inter-compute-node**.

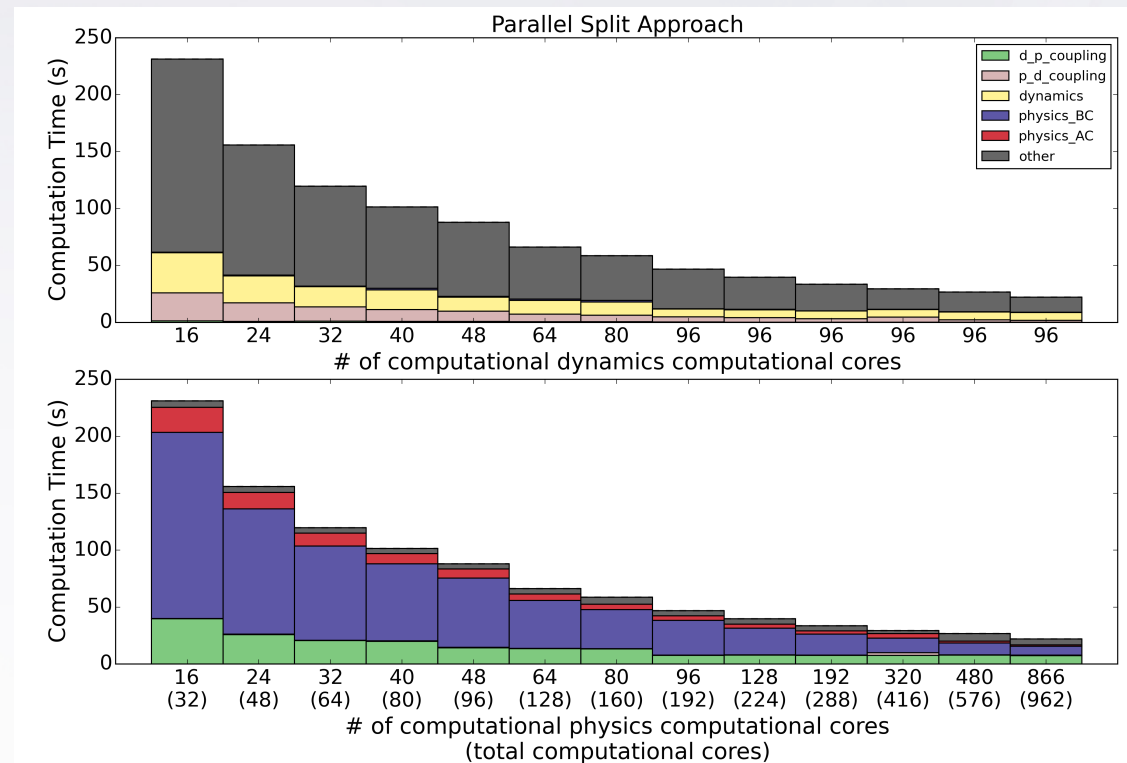
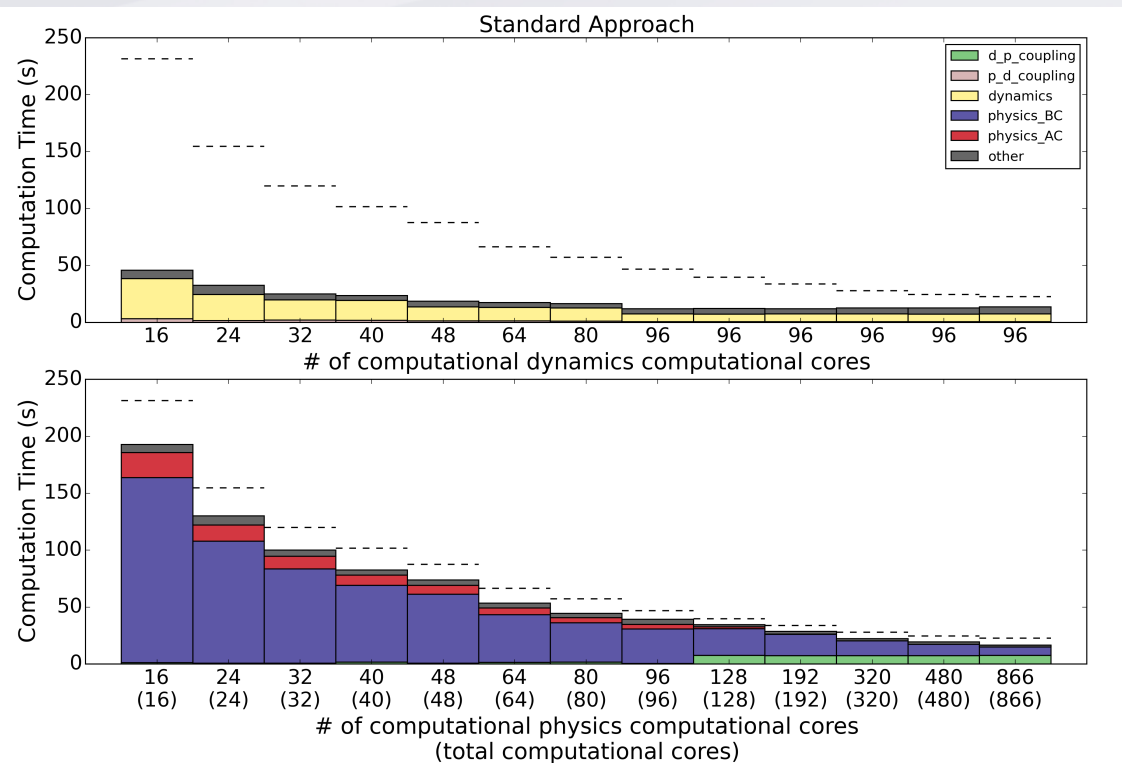
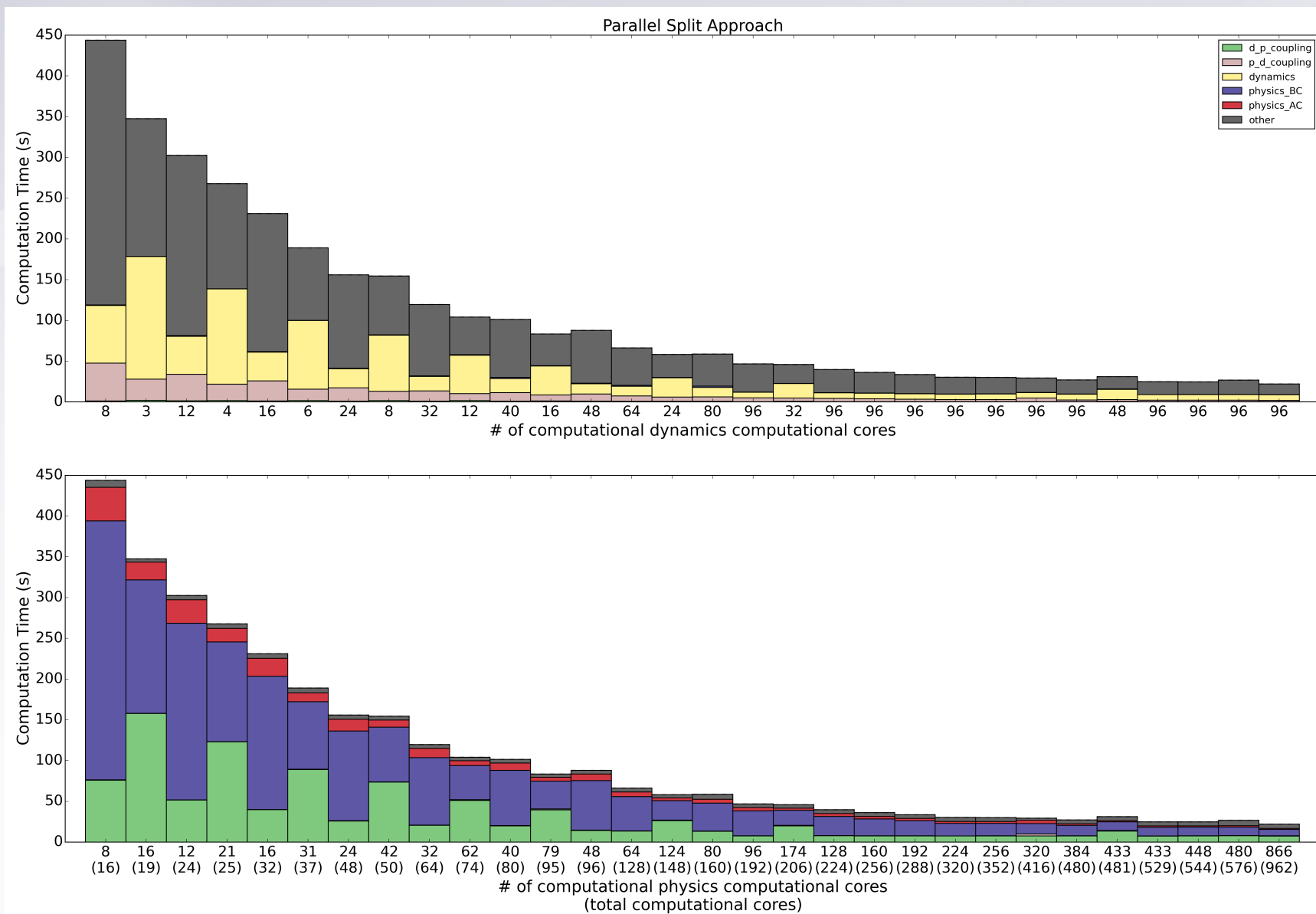


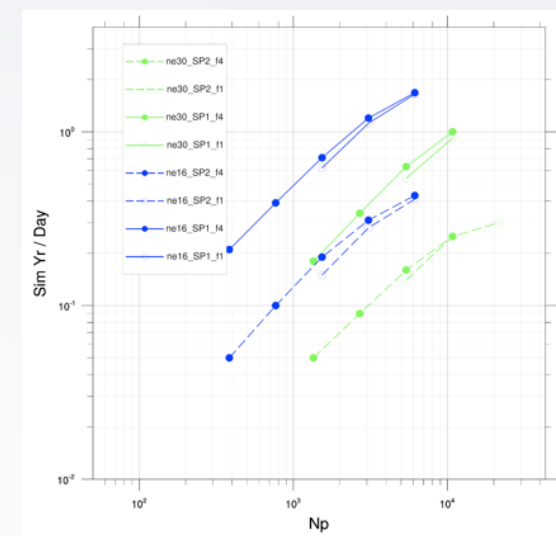
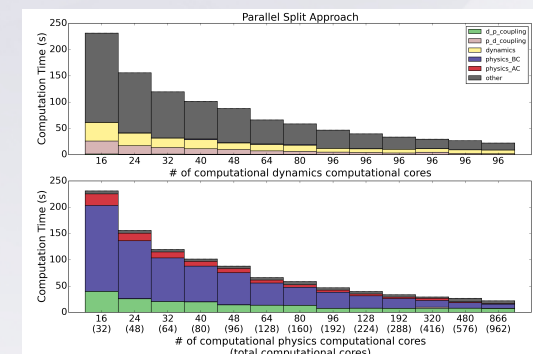
Figure 7: Average computational cost per core for dynamics (dynamics), physics (physics AC and BC) and dynamics-physics communication (d\_p and p\_d coupling) for the standard ACME model (left) and the parallel-split implementation (right). Top panels represent only the cores assigned to dynamics, bottom panels are cores assigned to physics.

# Parallel-Split: improved performance? Optimum balancing:



# Steps forward and other applications:

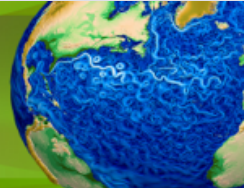
- Fix issues with output in parallel-split implementation.
- Implement a more balanced distribution of dynamics and physics cores over a single computational node.
- Improvement and further testing of mass conservation techniques.
- Possible implementation of parallel-split approach using the product of the next-gen coupler project (also a part of the CMDV project).
- Application of parallel-split in with the current work being conducted on super-parameterization.



# For more info come check out my poster: A13

## P: Advances in the application of parallel split physics and dynamics

Aaron S. Donahue and Peter M. Caldwell  
Lawrence Livermore National Laboratory



### Opportunity

**Improved performance through a change in coupling paradigm:** Dynamics and physics are loosely coupled in the ACME model through sequential-tendency splitting (STS). This has proven to be a limiting factor for scalability of the model, and application of parallel-splitting (PS) provides an opportunity to break past this limit and improve model performance.

### Problem Description

**Dynamics elements vs. physics columns:**

- Dynamics and physics are solved on similar but different domains: a cubed sphere spectral element mesh for dynamics, and a set of individual columns for the physics mesh. See Figure 1.
- The number of physics columns is greater than the number of dynamics elements by a factor of 9
- For standard run setups, dynamics and physics are roughly the same computational cost, see Figure 2.
- New developments such as super-parameterization may make physics more expensive than dynamics in which case parallel-splitting could be critical for improved throughput.

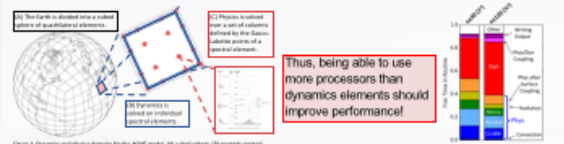


Figure 1: Dynamics and physics domains for the ACME model. (A) Cubed sphere spectral element mesh for dynamics. (B) Individual columns for physics. (C) Dynamics and physics domains for the ACME model. (D) Dynamics and physics domains for the ACME model. (E) Dynamics and physics domains for the ACME model.

Figure 2: Parallel split physics and dynamics integration time spent in various components. The chart shows that physics and dynamics are roughly the same computational cost.

Thus, being able to use more processors than dynamics elements should improve performance!

### Scalability: to the elements...and beyond

**Pushing core counts past element counts:** The ACME model scales up to the number of elements, see Figure 3. For our current suite of meshes this means:

- 5.4K cores for the 1 degree mesh (ne30), and
- 85.4K cores for the 1/4 degree mesh (ne120).

As HPCs get larger so does the limit of available resources:

- ORNL-Titan has 299K cores
- NERSC-Edison has 134K cores
- NERSC-Cori has 658K cores

If we can scale to the number of physics columns we can use an order of magnitude more cores, allowing us to utilize these and future resources more fully.

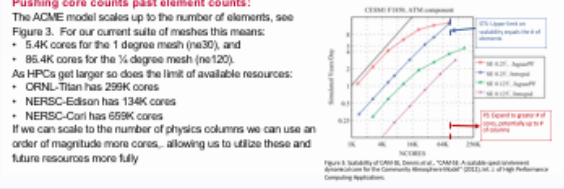


Figure 3: Scalability of the ACME model. The graph shows the number of cores required for different mesh resolutions (ne30, ne60, ne120) and the number of physics columns. The number of physics columns is significantly higher than the number of dynamics elements.

### Preliminary Issues and Solutions

**Solution Fidelity:** Switching to a parallel-split paradigm will change the model solution. Does this result in a degraded solution? **No.** Analysis of 10 year climatologies for STS and PS show very little change in the solution for the full suite of variables produced by the AMV diagnostic toolkit, see example in Figure 4.

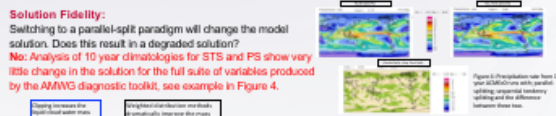


Figure 4: Comparison of 10-year climatologies for STS and PS. The plots show that the solutions are very similar, indicating that the parallel-split paradigm does not result in a degraded solution.

**Mass Conservation:** A drawback to parallel-splitting is the potential for local negative mass with dynamics and physics drawing at the same resource independently, the general approach of clipping these values to zero arbitrarily increases the global mass. See Figure 5. Redistribution of mass over an element either horizontally, vertically, or over the full element provides a tool for addressing mass conservation issues.

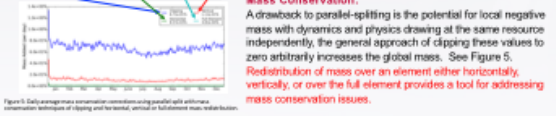


Figure 5: Mass conservation issues with parallel-splitting. The graph shows that local negative mass can occur, which is addressed by clipping these values to zero, leading to an increase in global mass.

### Issues and Proposed Solutions

**Mission accomplished? Not yet:** In our initial simultaneous dynamics and physics implementation, we do not observe improved scalability at high core counts, Figure 6. Why is this?

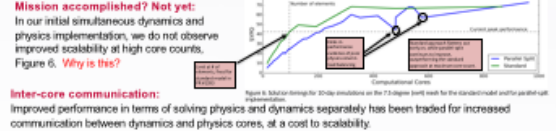


Figure 6: Computational cost versus number of cores. The graph shows that the computational cost increases with the number of cores, but the rate of increase slows down at high core counts, indicating a lack of improved scalability.

**Inter-core communication:** Improved performance in terms of solving physics and dynamics separately has been traded for increased communication between dynamics and physics cores, at a cost to scalability.

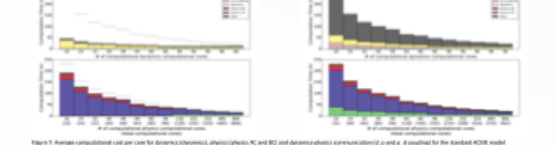


Figure 7: Average communication cost versus number of cores. The graph shows that the communication cost increases with the number of cores, which is a trade-off for improved performance.

**Dynamics/physics balancing per node:** A proposed solution to minimize inter-core communication cost is to balance the assignment of a fraction of cores per node to dynamics and the rest to physics such that most communication between physics and dynamics is done within a compute node. This work is currently underway.

This work funded by the ACME Software Modernization GMDV project

ACME Accelerated Climate Modeling for Energy

Lawrence Livermore National Laboratory  
94550-8002  
aaron@llnl.gov  
climatemodeling.science.energy.gov/acme

U.S. DEPARTMENT OF ENERGY

This work was performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under Contract DE-AC52-07NA27344. M Release Number LLNL-POST-

Thank you!