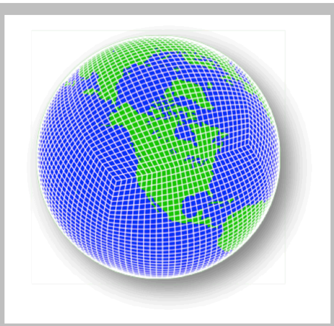
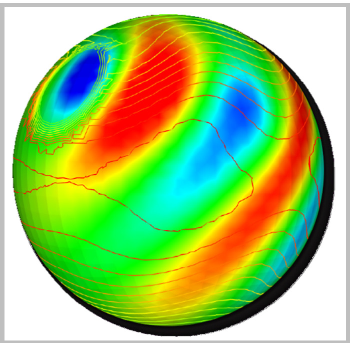
SAND2017-0585C





The Aeras Global Atmosphere Model

Bill Spotz, Sandia National Laboratories

The Aeras Project

- Sandia LDRD project, 2014-16
- "A Next Generation Global Atmosphere Model"
- Use Albany to develop a global atmosphere model suitable for a climate model such as ACME
- All of the standard advantages of using Albany, focusing on:
  - Performance portability
  - Uncertainty quantification
- Suite of models:
  - 2D Shallow water equations
  - X-Z hydrostatic equations
  - 3D hydrostatic equations

The Aeras Project

- Acknowledgements to the team:
  - Pete Bosler
  - Steve Bova
  - Irina Demeshko
  - Jeff Fike
  - Oksana Guba
  - James Overfelt
  - Erika Roesler
  - Andy Salinger
  - Tom Smith
  - Irina Tezaur
  - Jerry Watkins

Aeras Suite of Models

- Shallow Water Equations

$$\frac{\partial \mathbf{u}}{\partial t} = -\omega\hat{\mathbf{k}} \times \mathbf{u} - \nabla\left(\frac{1}{2}\mathbf{u}^2 + gH\right),$$

$$\frac{\partial h}{\partial t} = -\nabla \cdot h\mathbf{u}$$

Shallow water assumptions:

- Atmospheric thickness is independent variable h
- Spatial derivatives are "horizontal" only

- X-Z Hydrostatic

$$\frac{\partial u}{\partial t} + \frac{\partial}{\partial t}\left(\frac{1}{2}u^2 + \phi\right) + \dot{\eta}\frac{\partial u}{\partial \eta} + \frac{RT_v}{p}\frac{\partial p}{\partial x} = 0,$$

$$\dot{\eta}\frac{\partial p}{\partial \eta} = -\frac{\partial p}{\partial t} - \int_{\eta_s}^{\eta}\frac{\partial}{\partial x}\left(\frac{\partial p}{\partial \eta'}\right)\,\mathrm{d}\eta',$$

$$\frac{\partial}{\partial t}\frac{\partial p}{\partial \eta} + \frac{\partial}{\partial x}\left(u\frac{\partial p}{\partial \eta}\right) + \frac{\partial}{\partial \eta}\left(\dot{\eta}\frac{\partial p}{\partial \eta}\right) = 0,$$

$$\frac{\partial T}{\partial t} + u\frac{\partial T}{\partial x} + \dot{\eta}\frac{\partial T}{\partial \eta} - \frac{RT_v}{c_p p}\omega = 0,$$

$$\omega = \frac{\partial p}{\partial t} + u\frac{\partial p}{\partial x}$$

X-Z hydrostatic assumptions:

- Vertical coordinate η is hybrid pressure / terrain following
- Vertical velocity ($\dot{\eta}$) and other quantities are small relative to horizontal velocities

Aeras Suite of Models

- 3D Hydrostatic

$$\frac{\partial \mathbf{u}}{\partial t} + (\zeta + f)\hat{\mathbf{k}} \times \mathbf{u} + \nabla\left(\frac{1}{2}\mathbf{u}^2 + \phi\right) + \dot{\eta}\frac{\partial \mathbf{u}}{\partial \eta} + \frac{RT_v}{p}\nabla p = 0,$$

$$\phi = \phi_s + \int_{\eta_s}^{\eta} \frac{RT}{p} \mathrm{d}\eta',$$

$$\dot{\eta}\frac{\partial p}{\partial \eta} = -\frac{\partial p}{\partial t} - \int_{\eta_s}^{\eta} \nabla \cdot \frac{\partial p}{\partial \eta'} \mathrm{d}\eta',$$

$$\mathrm{RT}_v = (c_p - \mathrm{q}c_v)\mathrm{T},$$

$$\frac{\partial}{\partial t}\frac{\partial p}{\partial \eta} + \nabla \cdot \left(\mathbf{u}\frac{\partial p}{\partial \eta}\right) + \frac{\partial}{\partial \eta}\left(\dot{\eta}\frac{\partial p}{\partial \eta}\right) = 0,$$

$$\frac{\partial T}{\partial t} + \mathbf{u} \cdot \nabla T + \dot{\eta}\frac{\partial T}{\partial \eta} - \frac{RT_v}{c_p p}\omega = 0,$$

$$\omega = \frac{\partial p}{\partial t} + \mathbf{u} \cdot \nabla p$$

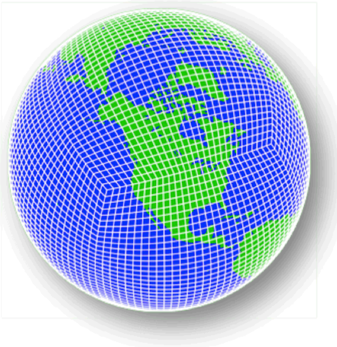
X-Z hydrostatic assumptions:

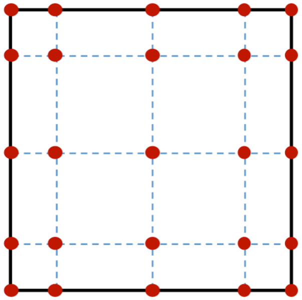
- Vertical coordinate η is hybrid pressure / terrain following
- Vertical velocity ($\dot{\eta}$) and other quantities are small relative to horizontal velocities
- Spatial operator (∇) is "horizontal" only

Proven Numerical Methods

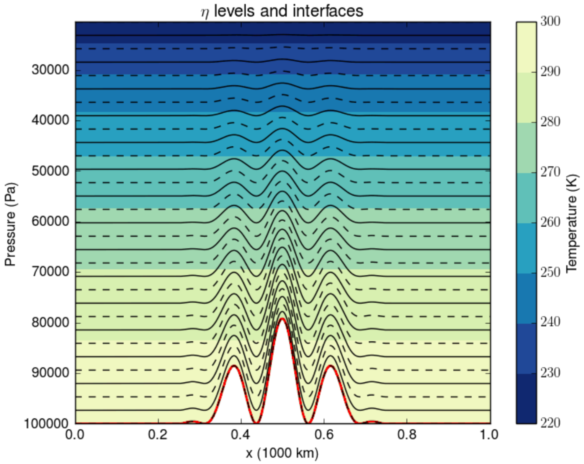
Sandia National Laboratories
Computing Research

- Derivatives in horizontal coordinates:
  - Spectral element method (matrix entries approximated with Gauss-Lobatto quadrature leads to diagonal mass matrix)





- Derivatives in hybrid vertical coordinate:
  - Finite difference method



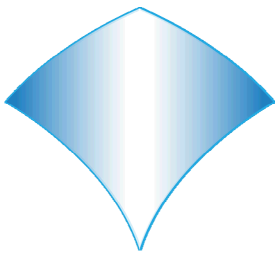
History:
- SEAM
- HOMME
- CAM-SE
  - CESM
  - ACME

- Time stepping
  - Runge-Kutta method
- Stabilization
  - Hyperviscosity $\tau \nabla^4()$

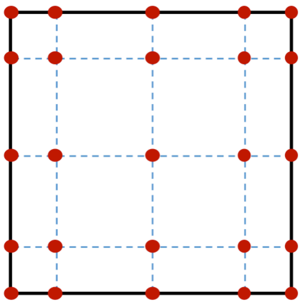
- Focus on next-generation capabilities
- Comparison against existing models

Additions to Albany

- Shell elements (Andy Salinger)
  - Topologically 2D elements on a 3D manifold



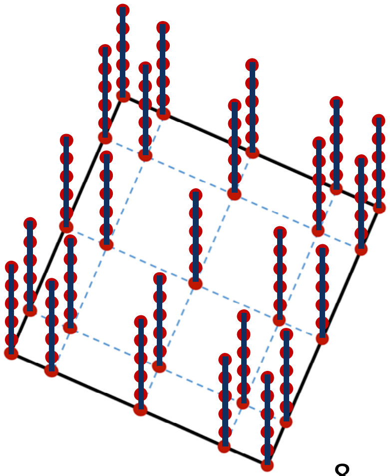
- Spectral elements (Bill Spotz)
  - Input STK mesh of quadrilaterals enriched with Gauss-Lobatto points (parallel algorithm with no communication)
  - Utilize Intrepid support for arbitrary-order elements
  - Each spectral element interpreted as a patch of bilinear elements for output
  - (3D enrichment algorithm designed but not yet implemented)



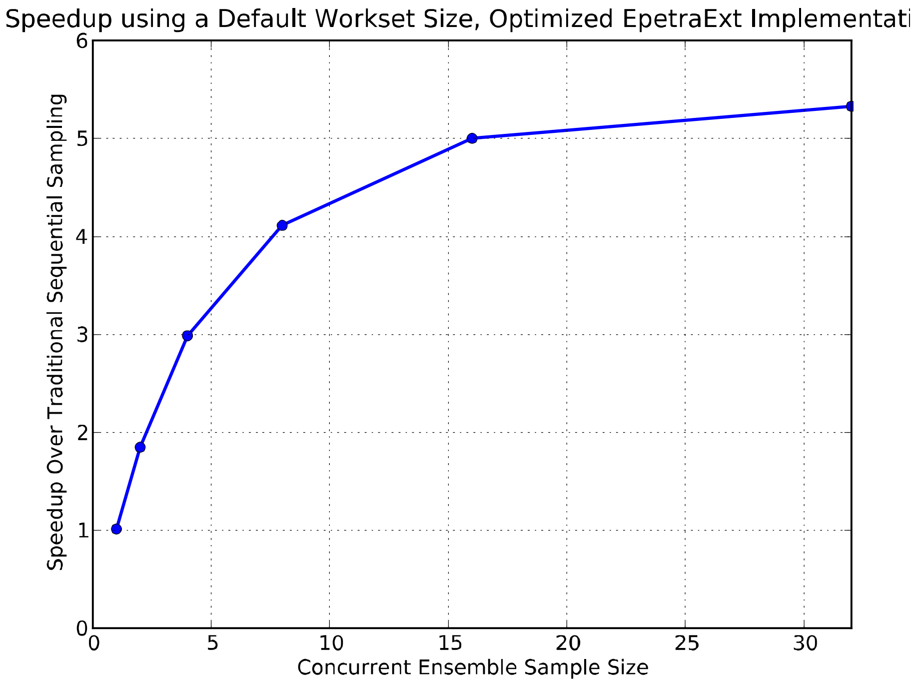
- Explicit time-stepping (Irina Tezaur)
  - Improved efficiency (diagonal mass matrix)
  - Additional Runge-Kutta methods

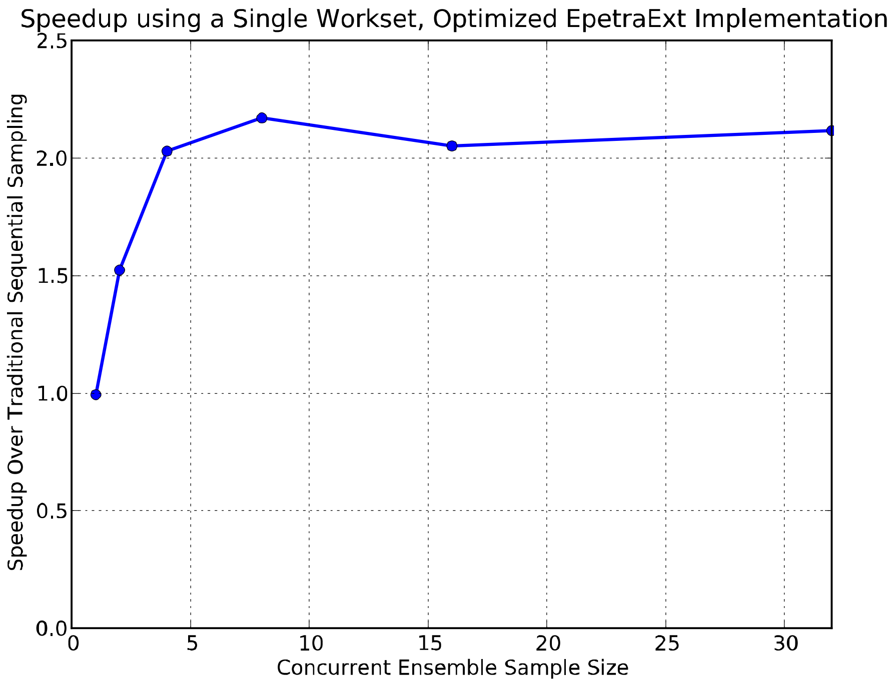
Additions to Albany

- Concurrent samples (Jeff Fike, Andy, Eric Phipps)
  - Run multiple samples of an ensemble for UQ concurrently to improve efficiency
  - Scalar template: double → array of double
  - Operators and model evaluator overloading
- Embedded UQ for transient problems (Andy)
  - Originally, only steady-state problems supported
- Spherical coordinate transformations (Steve Bova, James Overfelt)
  - May seem to be specific to atmosphere, but ice sheet model has expressed interest
- Atmospheric column data structures (Tom Smith, Pete Bosler)
  - Actually is specific to atmosphere…



Concurrent Samples





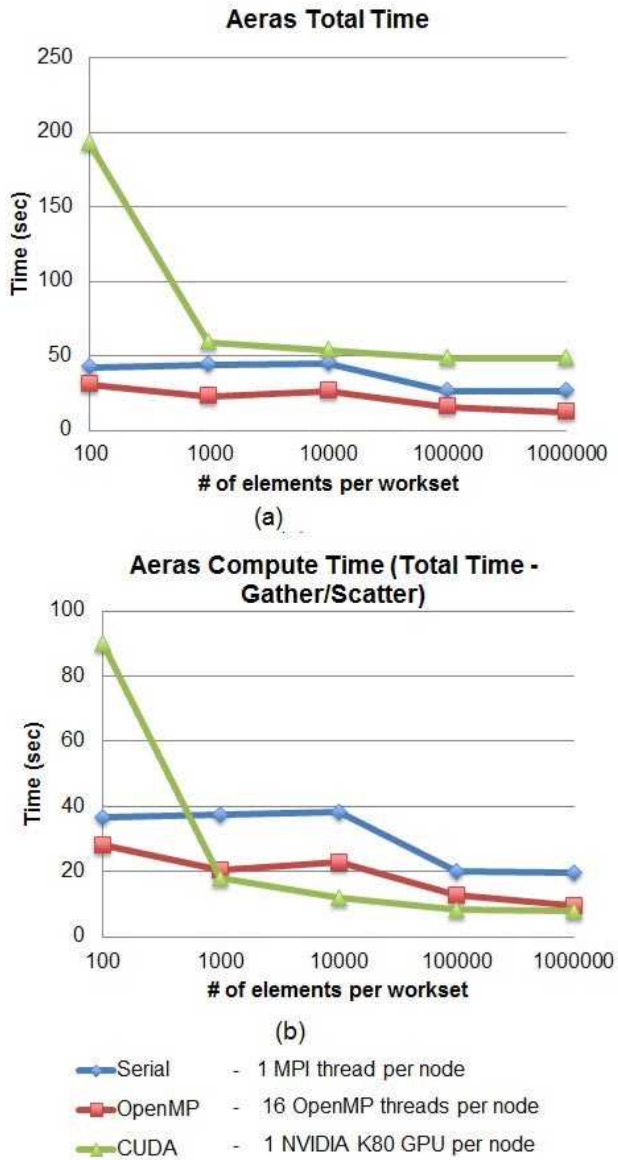
- Observed speedups for original EpetraExt concurrent sample implementation when using a single workset
- Observed speedups for optimized EpetraExt concurrent sample implementation when using a single workset

Performance Portability

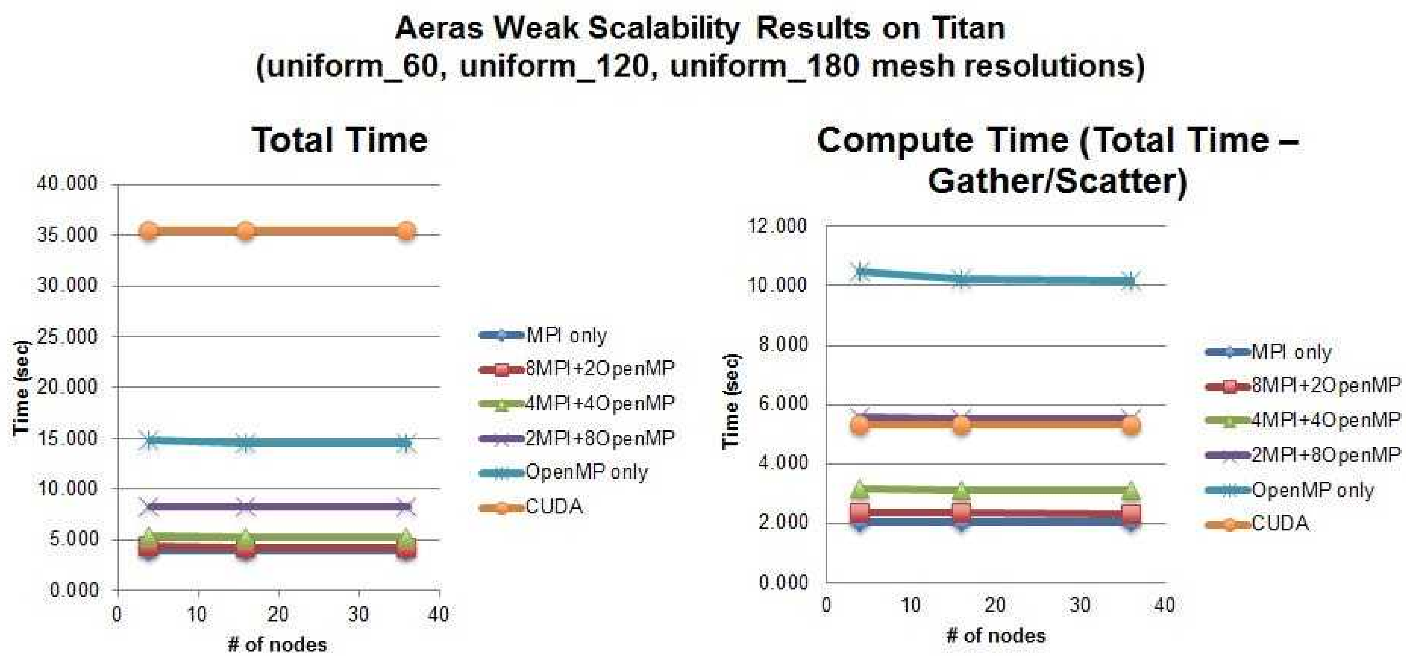
- Strong scalability for Aeras Shallow Water TC5 on Shannon, 0.5° mesh:

  a) Total time as a function of the number of elements per workset

  b) Time without gather/scatter as a function of the number of elements per workset

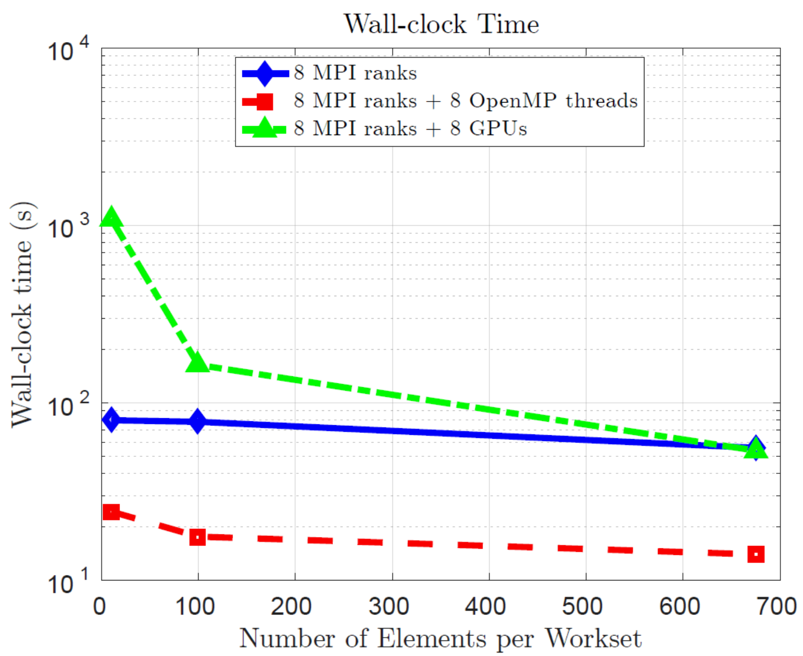


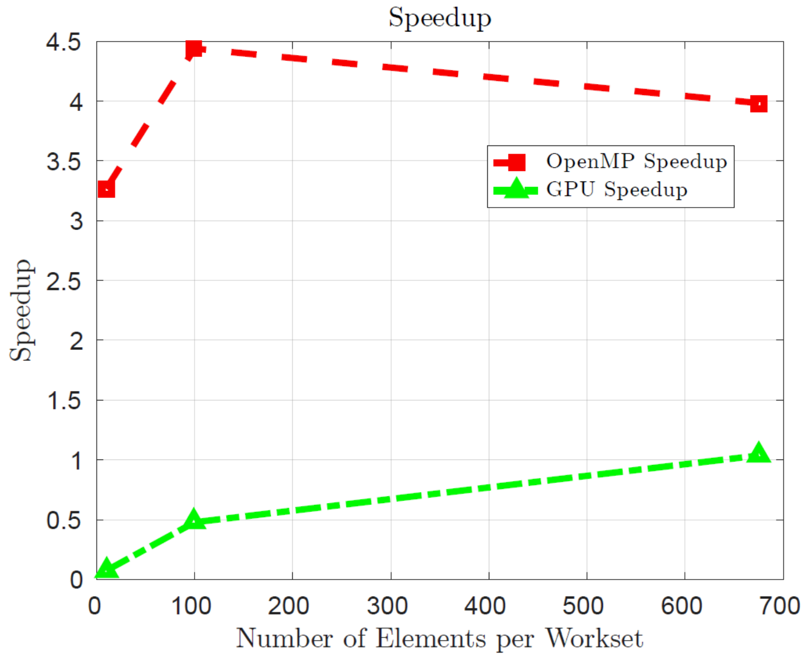
Performance Portability



- Weak scalability for Aeras Shallow Water TC5 on Titan (about 5600 elements per node):
  - Left: total time
  - Right: compute time (right)

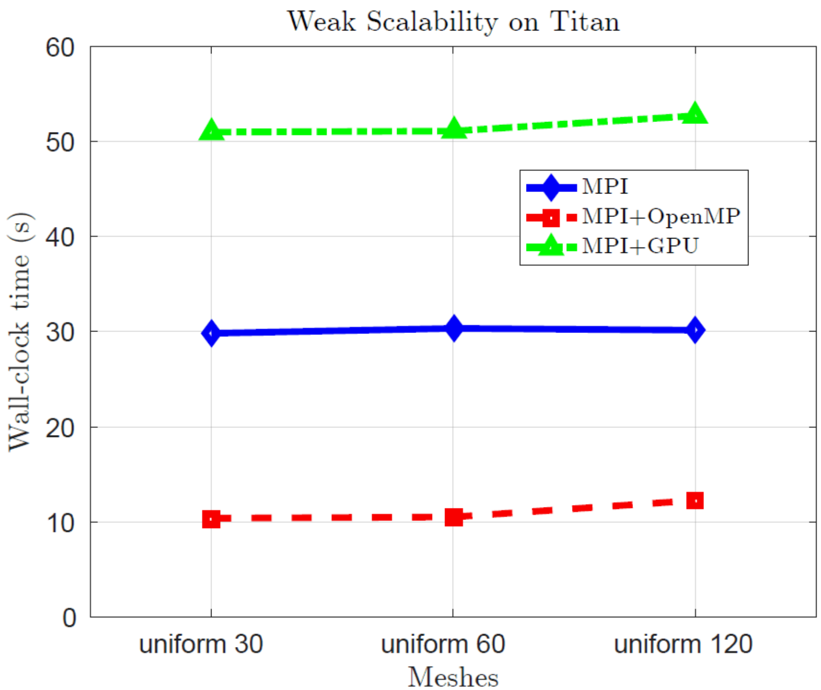
Performance Portability

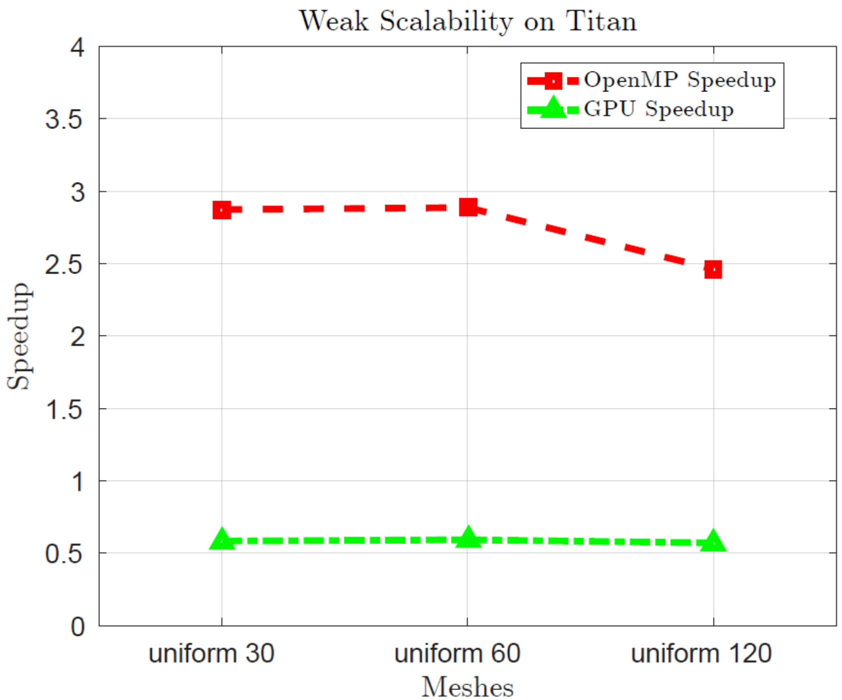




- Wall-clock time as a function of the number of elements per workset for Aeras 3D Hydrostatic baroclinic instability on Shannon for the 1.0° mesh
- OpenMP and Nvidia K80 GPU speedup over MPI as a function of the number of elements per workset for Aeras 3D Hydrostatic baroclinic instability on Shannon for the 0.5° mesh

Performance Portability





- Weak scalability results for the Aeras 3D Hydrostatic baroclinic instability test case on Titan
- OpenMP and Nvidia K20X GPU speedup over MPI for the Aeras 3D Hydrostatic baroclinic instability test case on Titan

The Follow-On to Aeras

- Albany was never a perfect fit for an atmosphere model focused on performance issues
- Communication assumes first- or second-order operators … hyperviscosity violates this
  - We apply Laplacian twice, which is inefficient
- In CAM-SE, hyperviscosity is not applied at every stage of the Runge-Kutta proceedure
  - To Albany, this appears to be solving different governing equations at different time step stages ... would require refactor
- We were never able to get rid of a "fundamental" factor of 2 slowdown
- Not obvious whether follow-on to Aeras should focus on Aeras or on CAM-SE

Aeras or CAM-SE?

Sandia National Laboratories
Computing Research

Aeras

- C++
  - Kokkos already implemented
- Albany inefficiencies
  - Implicit time-stepping
  - Top-level design
  - “Fundamental” 2x slowdown
- Small set of verification and timing tests
  - Need to add new tests
- More complex integration path for ACME

CAM-SE

- Fortran
  - Kernels must be converted to C++ to use Kokkos
- Over a decade of optimizations
- Broad set of verification and timing tests
  - Allows for methodical, step-by-step refactor
- Overall simpler integration path for ACME

Conclusion: CMDV Software proposal would (successfully) propose upgrading **CAM-SE** rather than porting Aeras to ACME

Aeras/Albany Debrief

Advantages to using Albany

- All the built-in capabilities of Albany at our fingertips:
  - Analysis tools, linear algebra, multiphysics, meshing tools, discretizations, derivatives, element fill, postprocessing and other utilities
- Could focus on development of evaluators
- Could leverage work of Albany and Trilinos developers

Disadvantages to using Albany

- Had to wait for Tpetra refactor
- Had to wait for upgrade from STK-Classic to new STK meshes
- Had to wait for Kokkos refactor
- Had to add certain capabilities: shell elements, spectral elements, transient UQ
- Had to work around inefficiencies:
  - Implicit time-stepping
  - Top-level design

My Conclusion: We made *much* more progress on the Aeras project using Albany than if we had started from scratch, and could demonstrate next-gen capabilities that helped secure funding