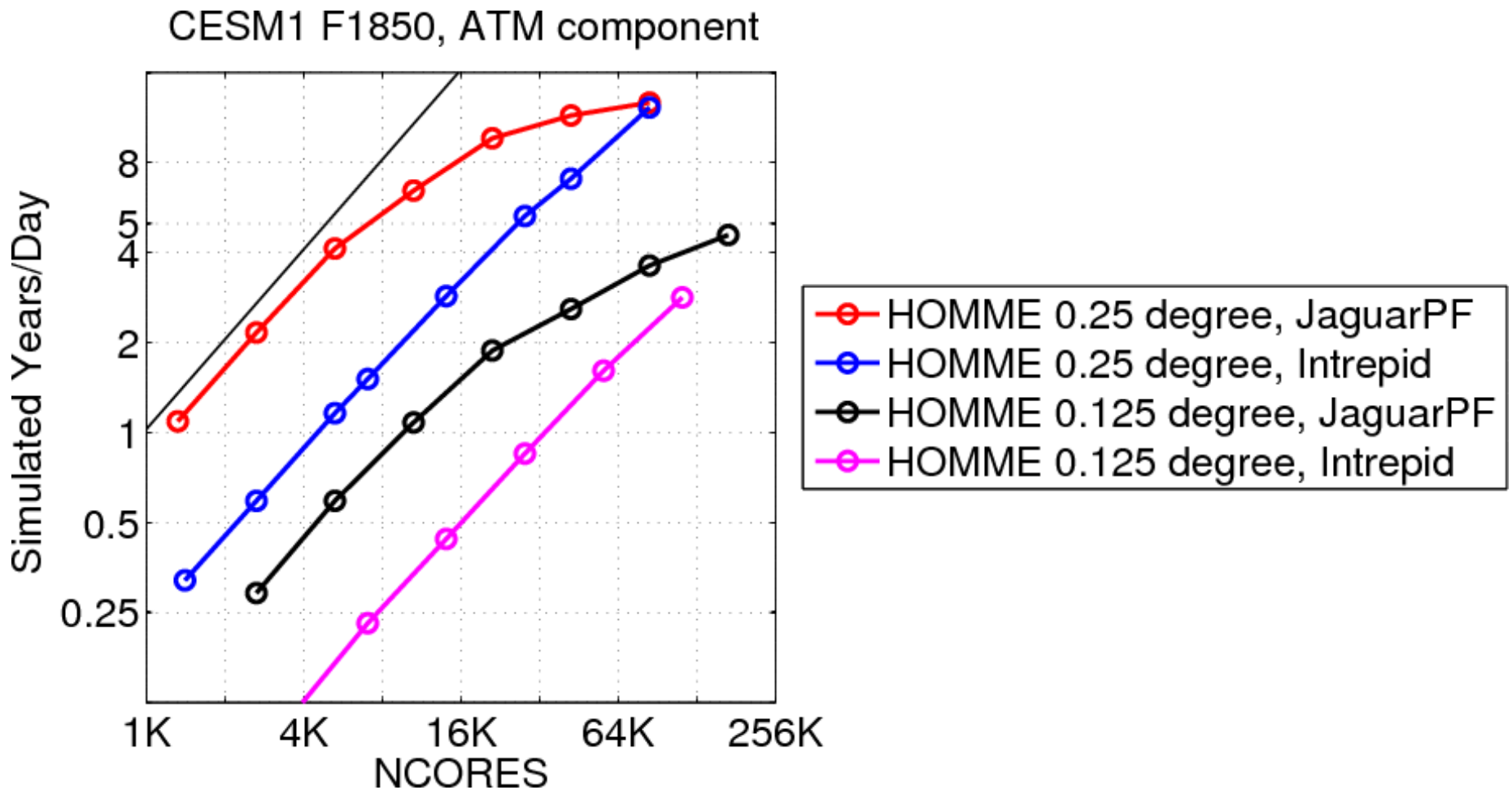
ParVis: Bringing HPC to Climate Model Analysis and Visualization

Robert Jacob, Tim Tautges, Jayesh Krishna, Mike Wilde, Sheri Mickeson, (Argonne) David Brown, Don Middleton, Mary Haley (NCAR) Pavel Bochev, Kara Peterson (Sandia), Kwan-Liu Ma, Jinrong Xie (UC-Davis), Jeff Daly, Jian Yin (PNNL)

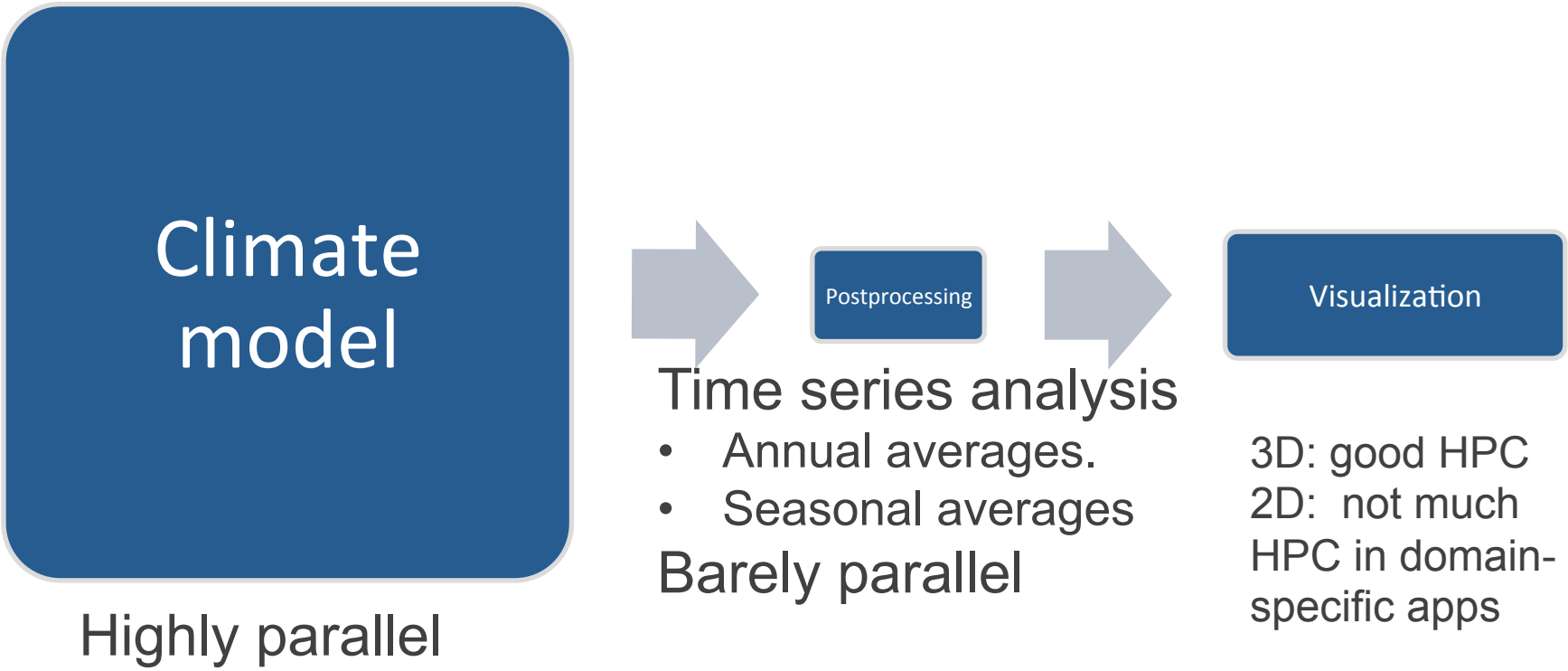
Ultrascale Visualization Workshop

Denver, CO, November 17, 2013

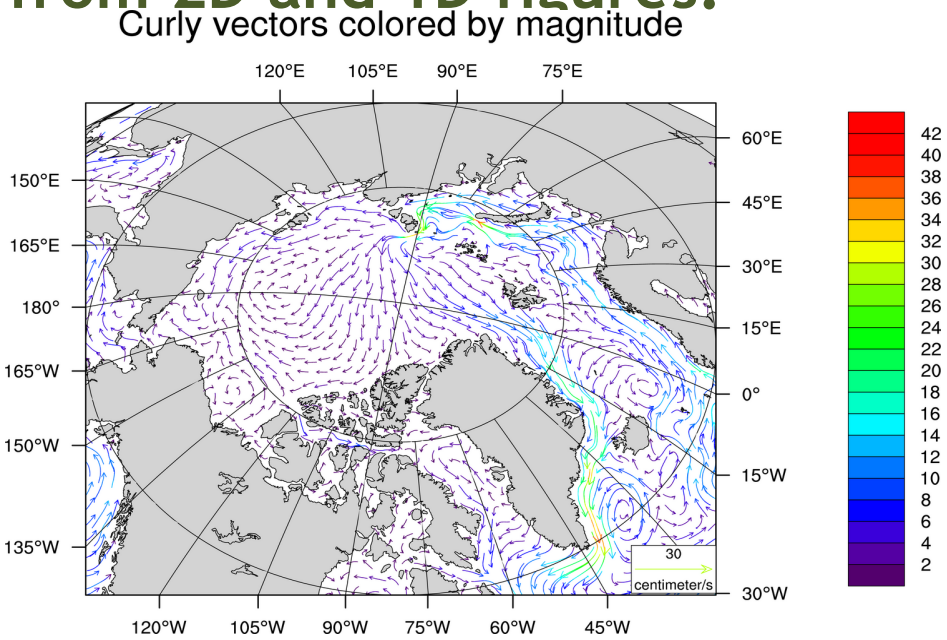
Background: Climate models are long standing HPC applications

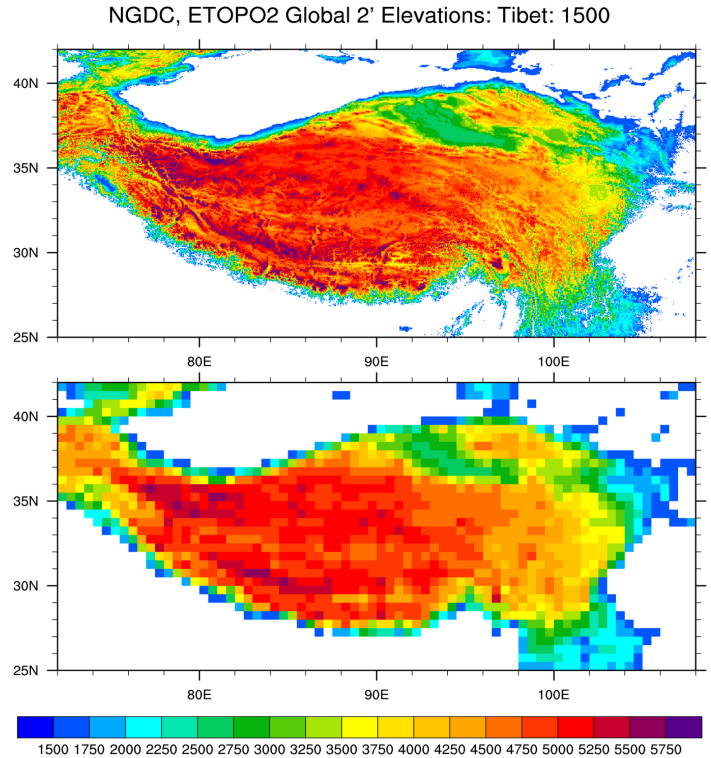


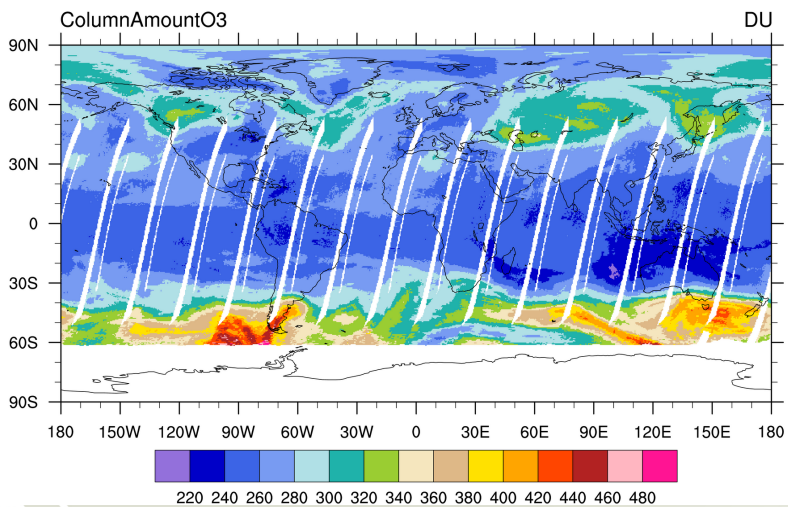
Background: Post processing is integral to visualization and, for most cases, neither are high performing applications. Huge bottleneck!



Background: Much insight about climate still comes mostly from 2D and 1D figures.







Why? The real global atmosphere and ocean have a ***small aspect ratio***; 10,000 km vs. 10 km.

Background: Climate scientist have thousands of lines and person-years of development in domain-specific scripting languages.

- NCAR Command Language (NCL)
- Climate Data Analysis Tools (CDAT)
- Grid Analysis and Display System (GrADS)
- Ferret

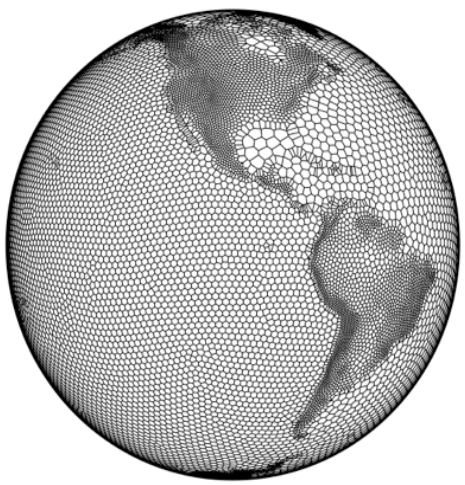
} Existing scripting tools:
*Have **no** parallelism.*
Assume lat-lon grids.

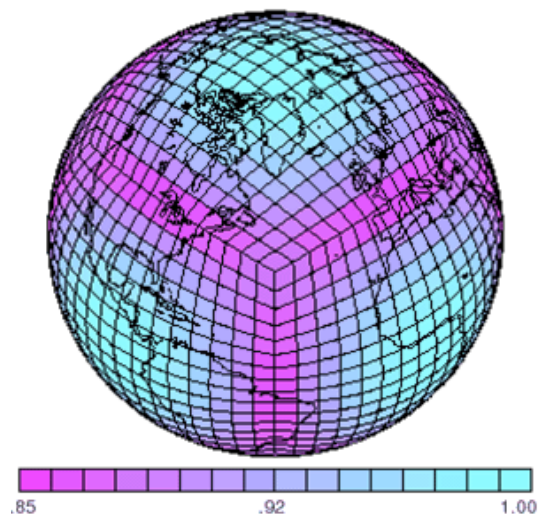
Background: Bottleneck is bad and getting worse.

- CAM-SE at 0.125 degrees
  - Single 3D variable: 616 MB
  - Single 2D variable: 25 MB
  - Single history file: 24 GB
  - 1 year of monthly output: 288 GB
  - 100 years of monthly: 28.8 TB

  (Compare: CMIP3: 35TB CMIP5: 3.3PB, 840 simulations (3 TB average))

Output data getting larger





Grids no longer rectangular. Interpolation necessary before plotting.

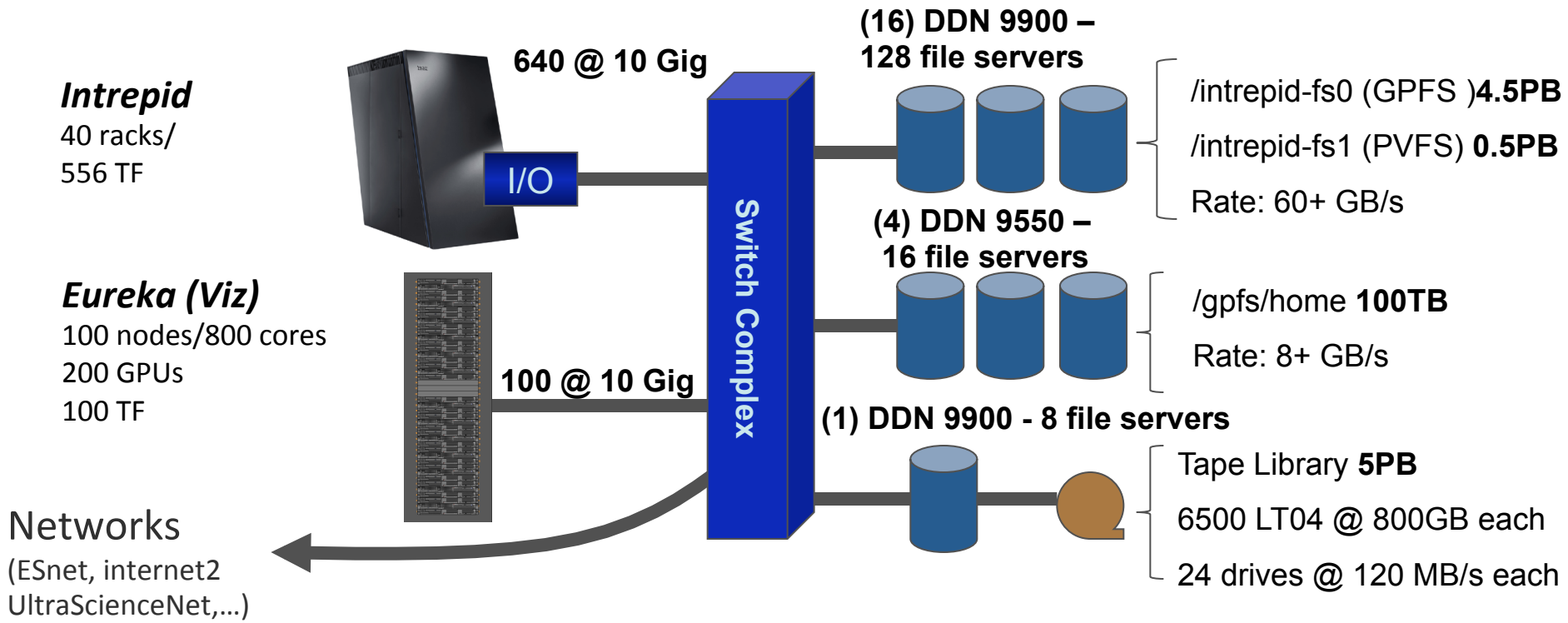
parVis

(Parallel Analysis Tools and New Visualization Techniques for Ultra-Large Climate Data Sets)

Move 2D workflow from single node workstation or laptop
to
Multi-node analysis cluster

AND allow workflow to make accurate calculations on multiple grids including unstructured through advanced libraries.

Example of ParVis' assumed hardware environment: Argonne Leadership Computing Facility Hardware Layout



Building Blocks for ParVis

Task-parallel scripting language *Swift*

- **Swift is a parallel scripting system for Grids and clusters**
  - for loosely-coupled applications - application and utility programs linked by exchanging files
- **Swift is easy to write**: simple high-level C-like functional language
  - *Small Swift scripts can do large-scale work*
- **Swift is easy to run**: a Java application. Just need a Java interpreter installed.

- **Swift is fast**: Karajan provides Swift a powerful, efficient, scalable and flexible execution engine.
  - *Scaling close to 1M tasks – .5M in live science work, and growing*

- Swift **usage** is growing:
  - *applications in neuroscience, proteomics, molecular dynamics, biochemistry, economics, statistics, and more.*

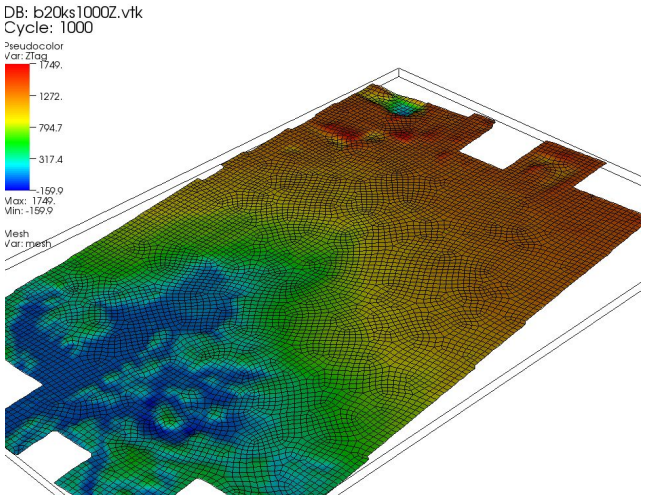
- Development supported mainly by NSF with additional support from NIH, Argonne and UChicago.

PNetCDF: NetCDF output with MPI-IO

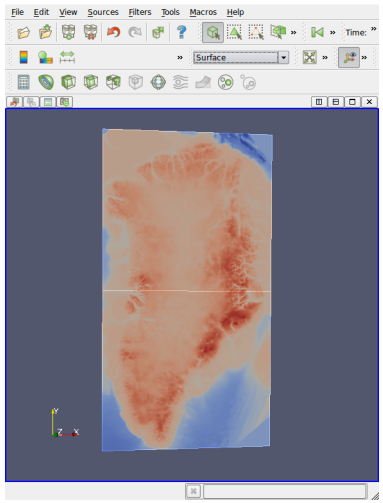
- Based on NetCDF
  - Derived from their source code
  - API slightly modified
  - Final output is indistinguishable from serial NetCDF file
- Additional Features
  - Noncontiguous I/O in memory using MPI datatypes
  - Noncontiguous I/O in file using sub-arrays
  - Collective I/O
- Unrelated to netCDF-4 work
- Support by OASCR

Mesh-Oriented datABase (MOAB)

- MOAB is a library for representing structured, unstructured, and polyhedral meshes, and field data on those meshes
- Supported by OASCR. Part of FASTMATH SciDAC Institute
- Uses array-based storage, for memory efficiency
- Supports MPI-based parallel model
- Interfaces with other important services
  - Visualization: ParaView, VisIt
  - Discretization: Intrepid (Trilinos package)
  - Partitioning / load balancing: Zoltan



Jakobshavn ice bed (in VisIt/MOAB)



Greenland ice bed elevation (in Paraview/MOAB)

http://trilinos.sandia.gov/packages/intrepid/

Developers: P. Bochev, D. Ridzal, K. Peterson, R. Kirby

Intrepid

INteroperable Tools for Rapid dEveloPment of compatIble Discretizations

A Trilinos package for compatible discretizations: a suite of stateless tools for

- Cell topology, geometry and integration
- Discrete spaces, operators and functionals on cell worksets
- Flexible and extensible design: easy to add tools for new cell shapes and basis functions
- Common API for Finite Element, Finite Difference and Finite Volume methods.

Design features

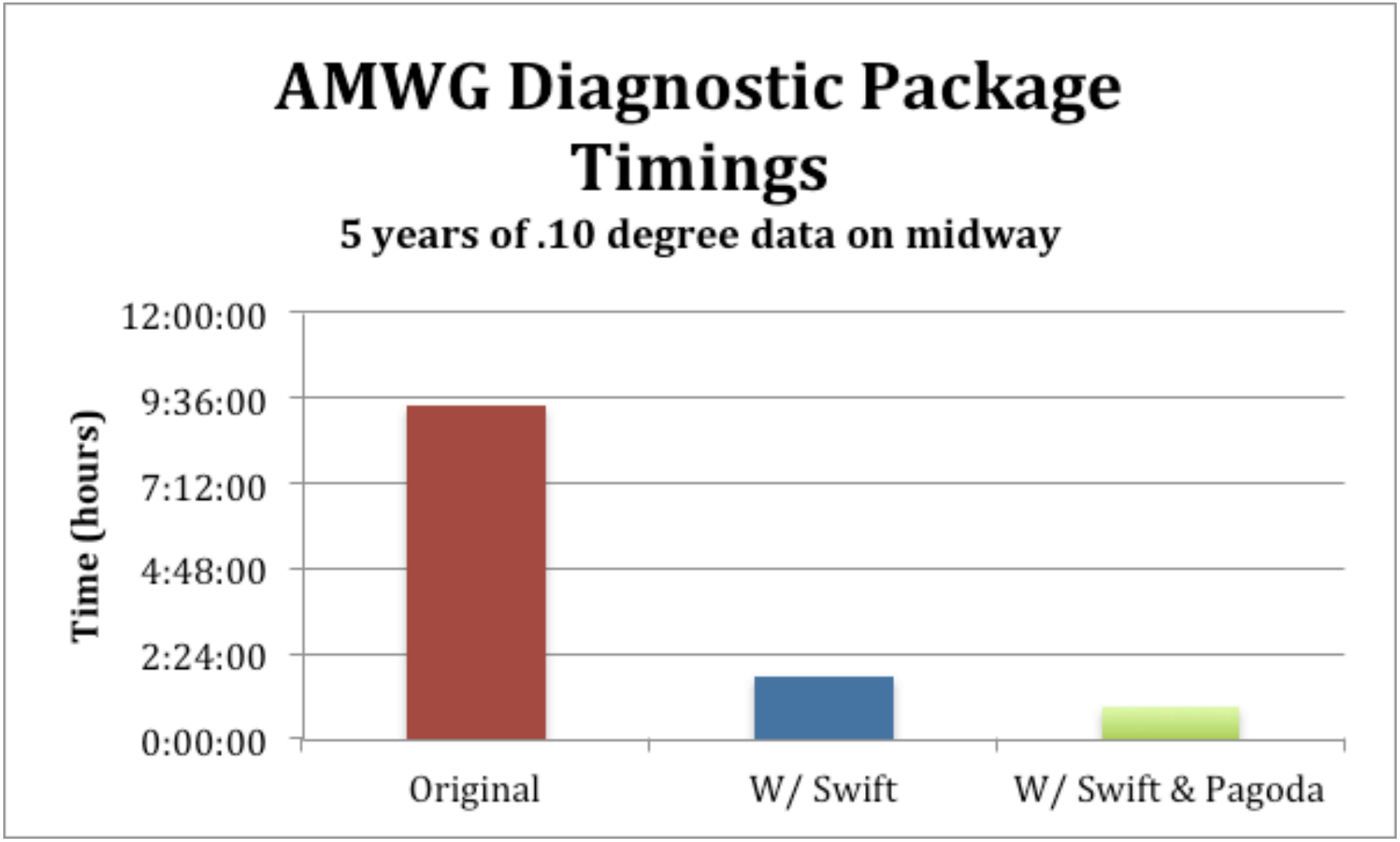
- optimized **multi-core** kernels; optimized **assembly**
- Can compute div, grad, curl on structured or unstructured grids maintained by MOAB.

ParVis results

Task-parallel atmosphere model diagnostics:

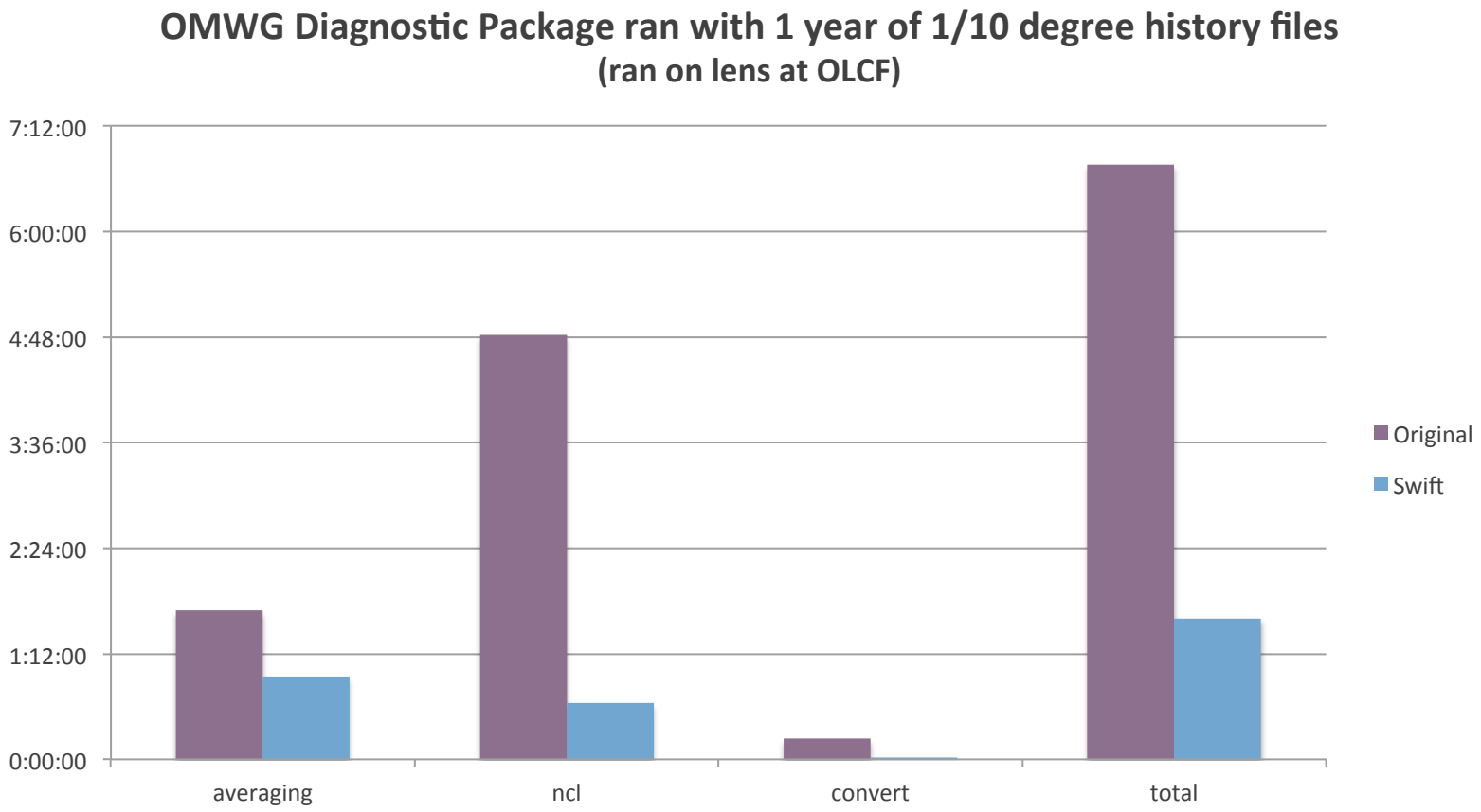
- CESM Atmospheric model working group diagnostics
  - Compares 2 CAM simulations or compares one CAM simulation to observational data
  - Controlled from a top level C-Shell script that calls NCO functions and NCL to create climate average files and over 600 plots that are browsable through a web interface

- ***ParVis' Swift-based AMWG diagnostics***
  - Officially part of version 5.3 of AMWG released in Feb, 2012.
  - Used daily at NCAR
  - Installed on LCF and NCAR machines.
  - Future development will include calling data-parallel functions


AMWG Diagnostic Package Timings
5 years of .10 degree data on midway
Time (hours)
12:00:00
9:36:00
7:12:00
4:48:00
2:24:00
0:00:00
Original
W/ Swift
W/ Swift & Pagoda

Additional task-parallel diagnostics

ParVis has since released task-parallel version of standard diagnostic packages for the ocean, land and sea-ice.



ParGAL - Parallel Gridded Analysis Library

- The main product from ParVis.
  - Data-parallel C++ Library
  - Typical climate analysis functions as found in domain-specific scripting languages.
  - Structured and unstructured numerical grids supported.

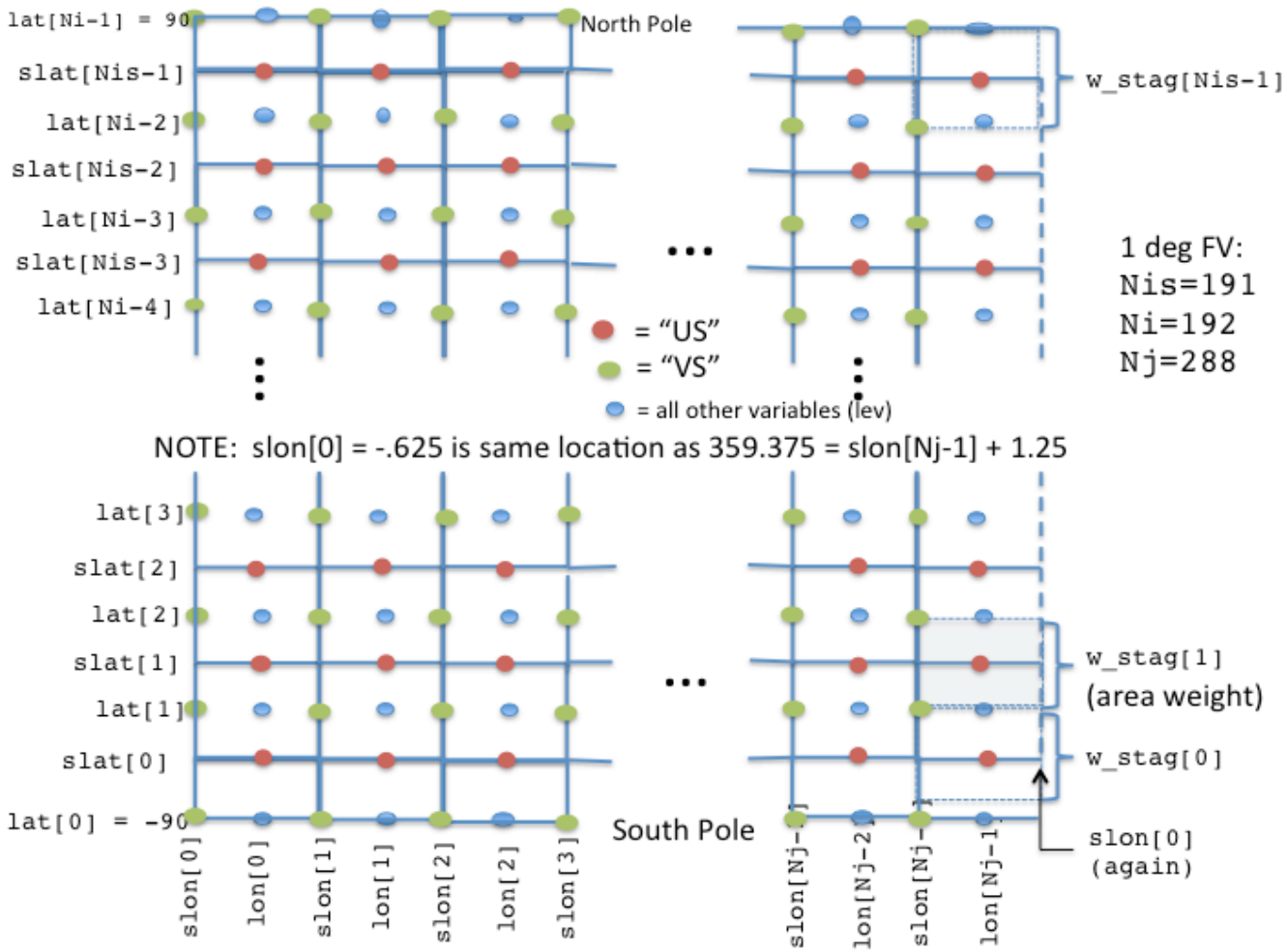
- Built upon…
  - MOAB
  - Intrepid
  - PnetCDF
  - MPI

- Provides a data-parallel core to perform typical climate post-processing operations.

- **Handles unstructured and semi-structured grids in all operations by building on MOAB and Intrepid. Supports parallel I/O by using PnetCDF.**

ParGAL represents discretizations as they are in the model. Algorithms are aware of grid location of data (using MOAB).

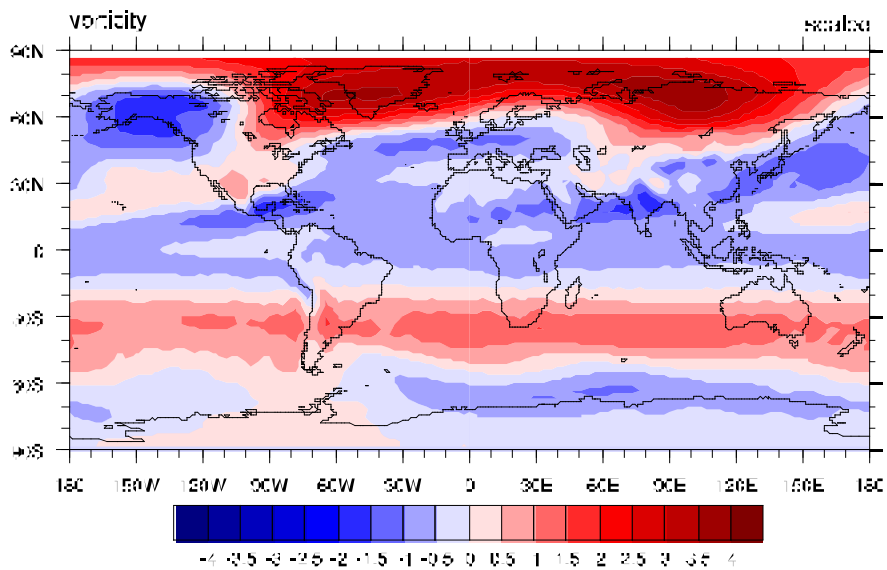
CAM's Finite Volume Grid



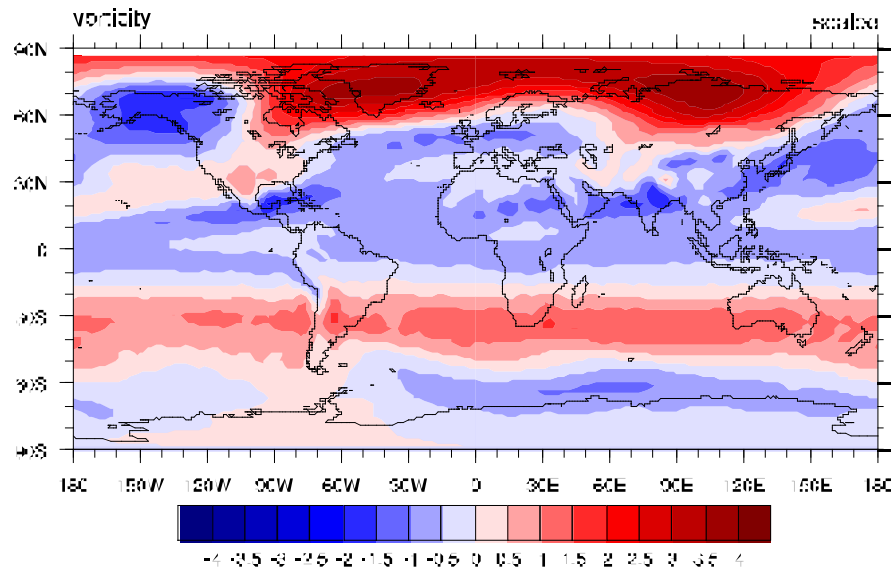
Note: Community should decide on grid metadata standards ASAP

Calculating Vorticity with ParGAL

Intrepid



NCL (uv2vrG_Wrap)

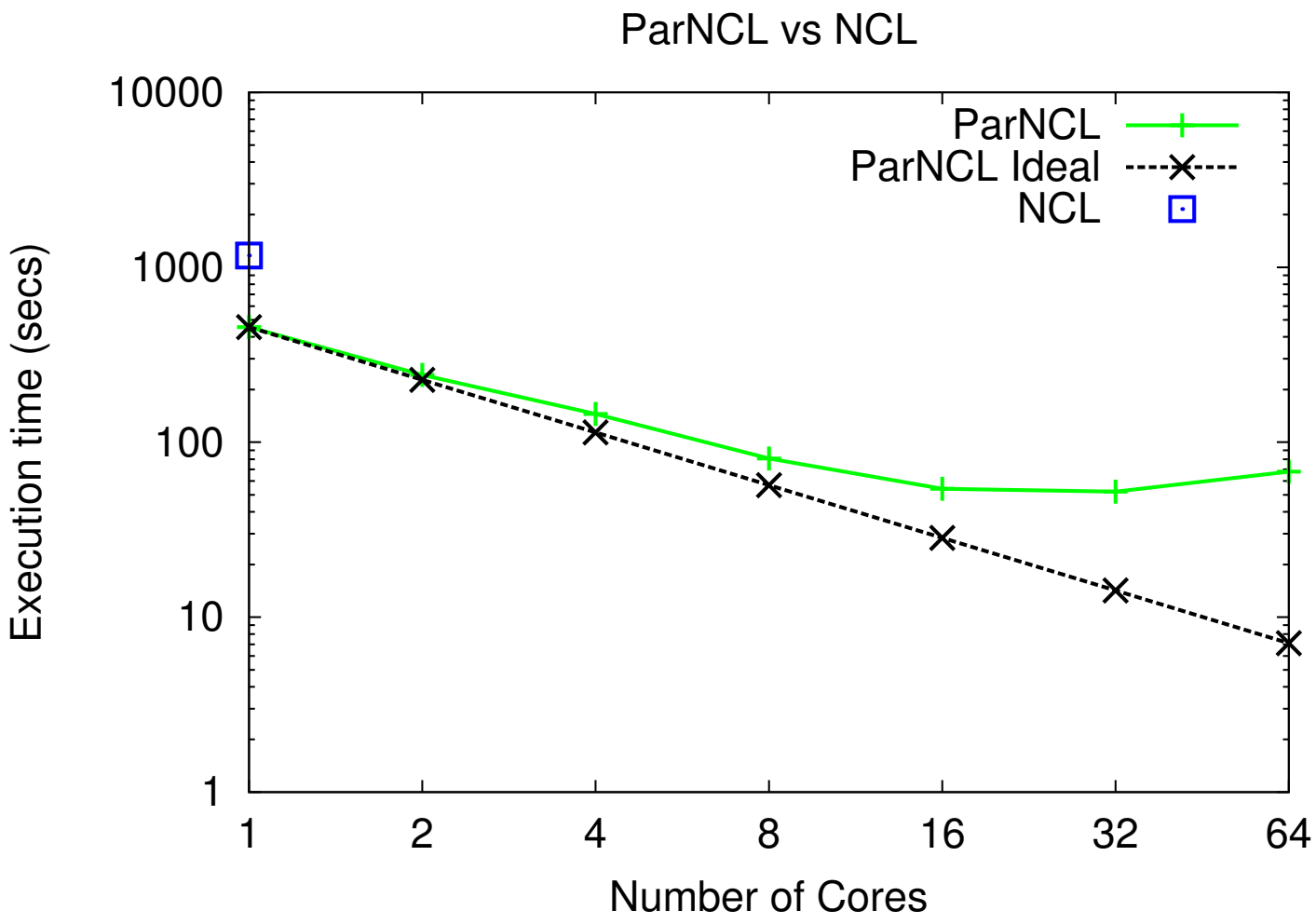


- Calculated locally on each element
- Easily parallelizable
- Global data not required

- Uses spherical harmonics
- Requires global data

$$vorticity = \frac{1}{r\cos\phi}\frac{\partial v}{\partial \lambda} - \frac{1}{r}\frac{\partial u}{\partial \phi} + \frac{u}{r}\tan\phi$$

Calculating Vorticity with ParGAL



Total Execution time for reading 4 timesteps of data and calculating the vorticity field on each level of a 768x1152x26 grid (FV 0.25) vs. number of cores.

ParNCL - parallel version of the NCL langauge.

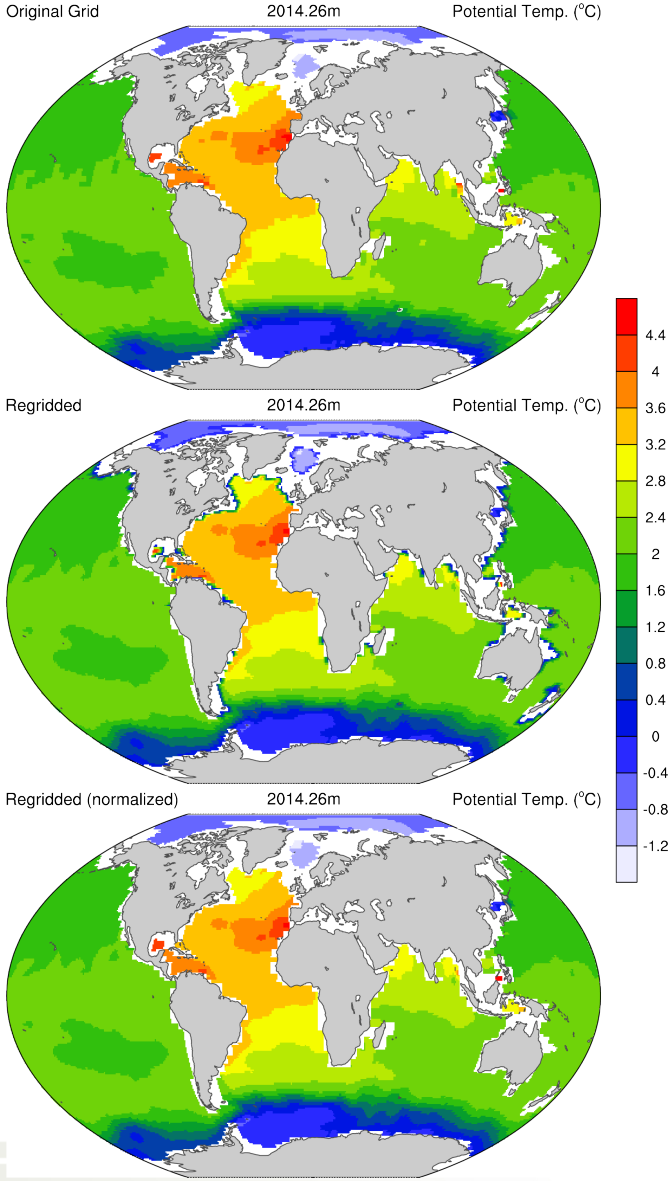
- ParGAL is just a library.
- Need to support calling ParGAL's parallel functions from a scripting interface to support the typical workflow on ultra-large, unstructured data.
- Solution: Convert NCL to parallel.

NCAR Command Language (NCL)

A scripting language tailored for the analysis and visualization of geoscientific data

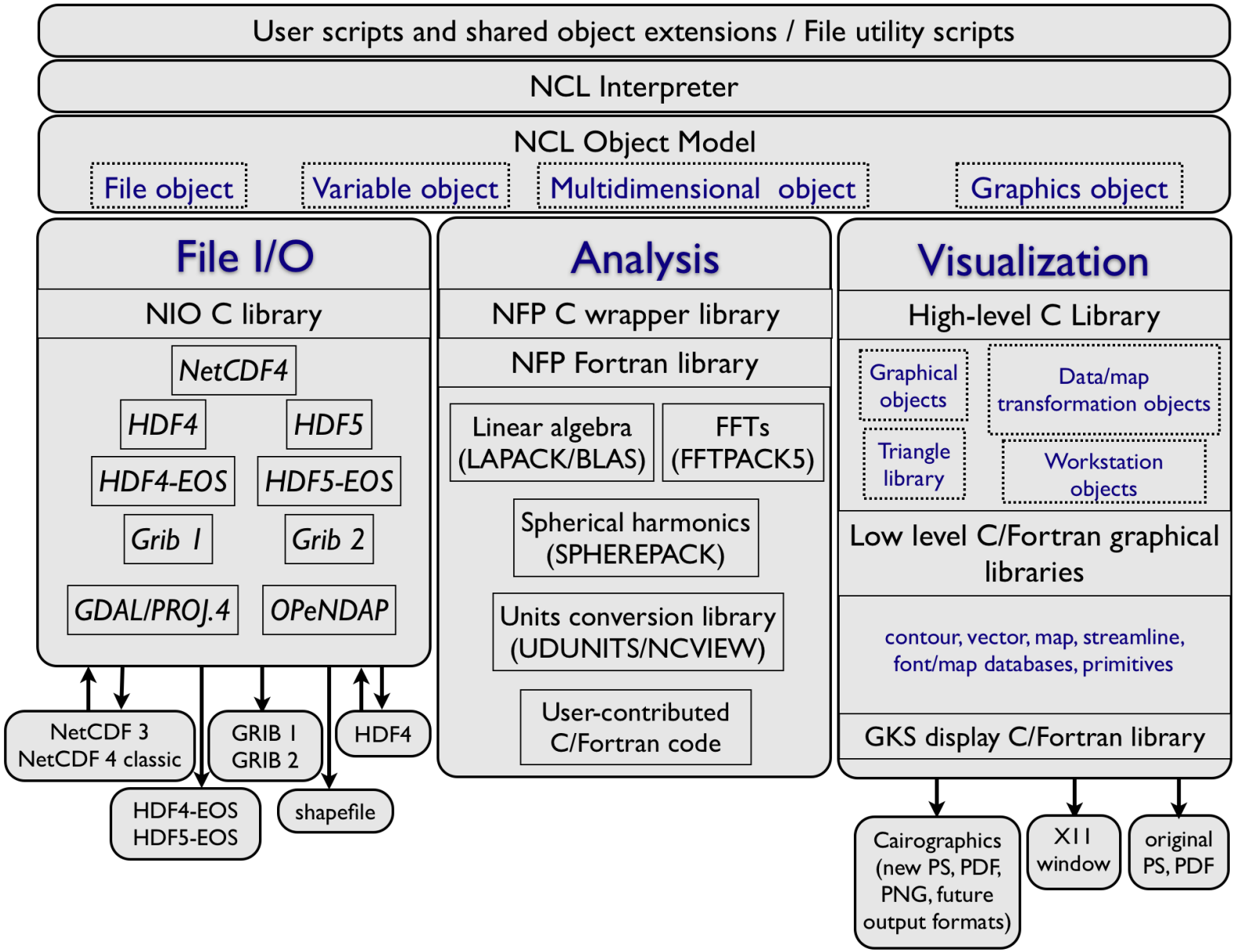
1. Simple, robust file input and output
2. Hundreds of analysis (computational) functions
3. Visualizations (2D) are publication quality and highly customizable

- **Community-based tool**
- Widely used by CESM developers/ users
- UNIX binaries & source available, free
- Extensive website, **regular workshops**

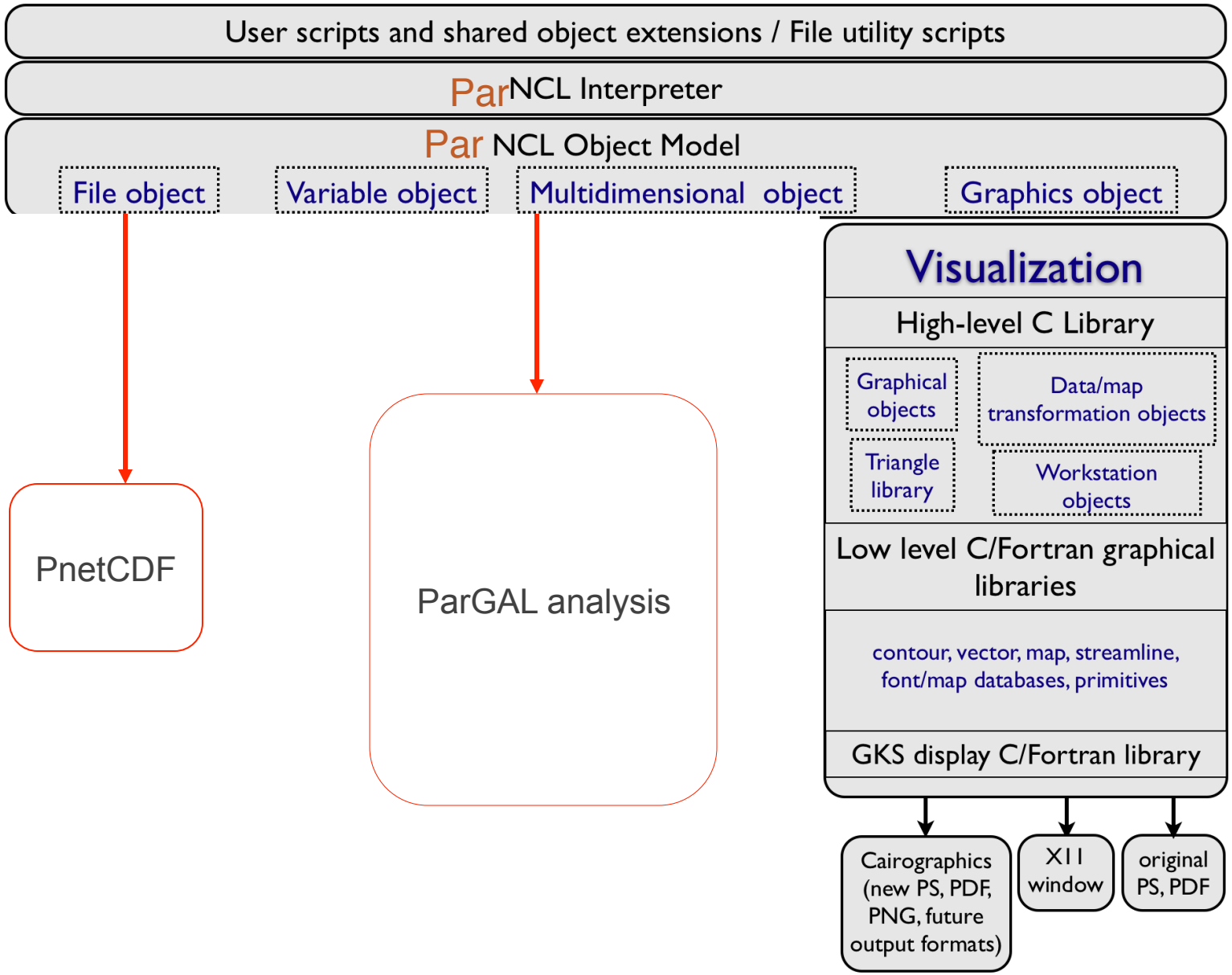


http://www.ncl.ucar.edu/

NCL architecture



ParNCL architecture



An NCL script executed by ParNCL

```
f = addfiles(diri+fili, "r") ;   open file

tt = f[:]->T(0,{500},{-30:30},:) ;  read a section of data.


wks  = gsn_open_wks("ps","parvis_t")  ; open a PS file

plot = gsn_csm_contour_map(wks, tt(:,:),False)
```

Read with parallel I/O

a parallel data object

In the ParNCL interpreter, tt is gathered to one node and passed to normal NCL graphics routines for plotting.

Using ParNCL requires a parallel environment:

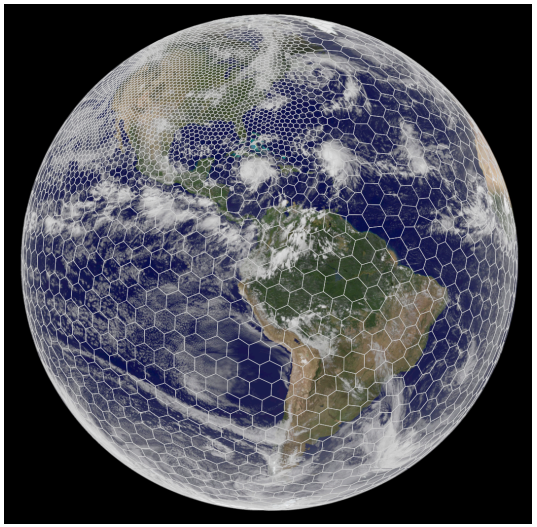
Now: `> ncl myscript.ncl`

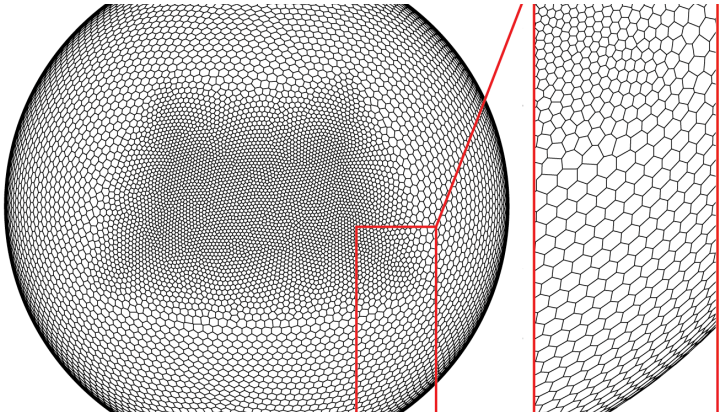
With ParNCL: `> mpirun -np 32 parcnl myscript.ncl`

ParNCL

- ParNCL supports addfiles(), NCL's multi-format file reader
  - Time slices of the variable read from file works
  - Only reads NetCDF for now (using PNetCDF) from CAM.
- Addition, subtraction of distributed multidimensional data.
- Scaling a distributed multidimensional array by a scalar.
- Subsetting a distributed multidimensional array.
- Approximate 40 functions re-implemented with ParGAL
- ParGAL and ParNCL source and ParNCL executable (beta2) all released
  - Installed on LCF analysis clusters

Example of new climate/weather model with unstructured grid





MPAS
Model for Prediction Across Scales

Based on unstructured centroidal Voronoi (hexagonal) meshes using C-grid staggering and selective grid refinement.

Jointly developed, primarily by NCAR/NSF and LANL/DOE, for weather, regional climate, and climate applications

MPAS infrastructure - NCAR, LANL, others.

MPAS - Atmosphere (NCAR)

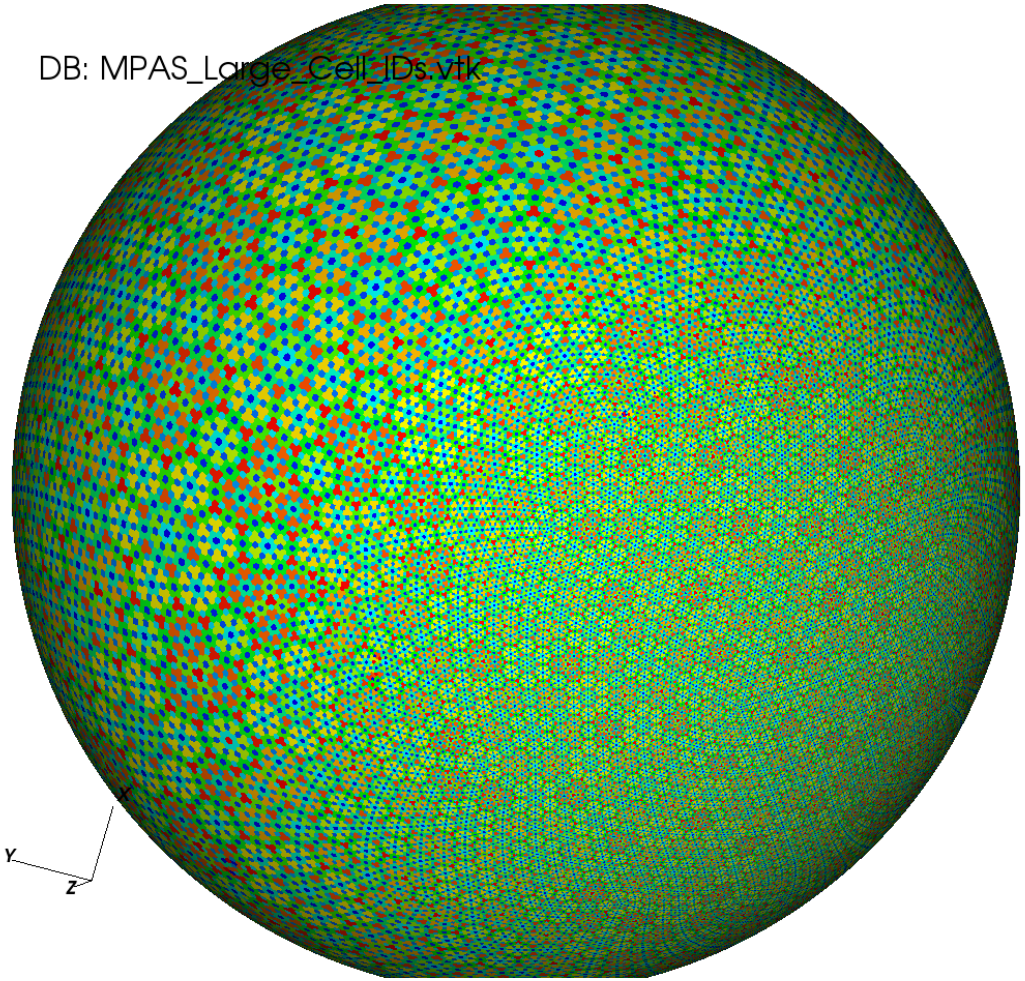
MPAS - Ocean (LANL)

MPAS – Sea Ice, etc

Bill Skamarock, Joe Klemp, Michael Duda,
Sang-Hun Park and Laura Fowler NCAR
Todd Ringler Los Alamos National Lab (LANL)
John Thuburn Exeter University
Max Gunzburger Florida State University
Lili Ju University of South Carolina

Recent ParGAL result: Read 65 million element MPAS grid (3km global horizontal) and 2 variables from a 1.1TB file, construct MOAB mesh representation and write out (51GB):

3 minutes on 400 processors.



Focus on NCL routine *gsn_csm_contour_map* for the 65M cell MPAS output.

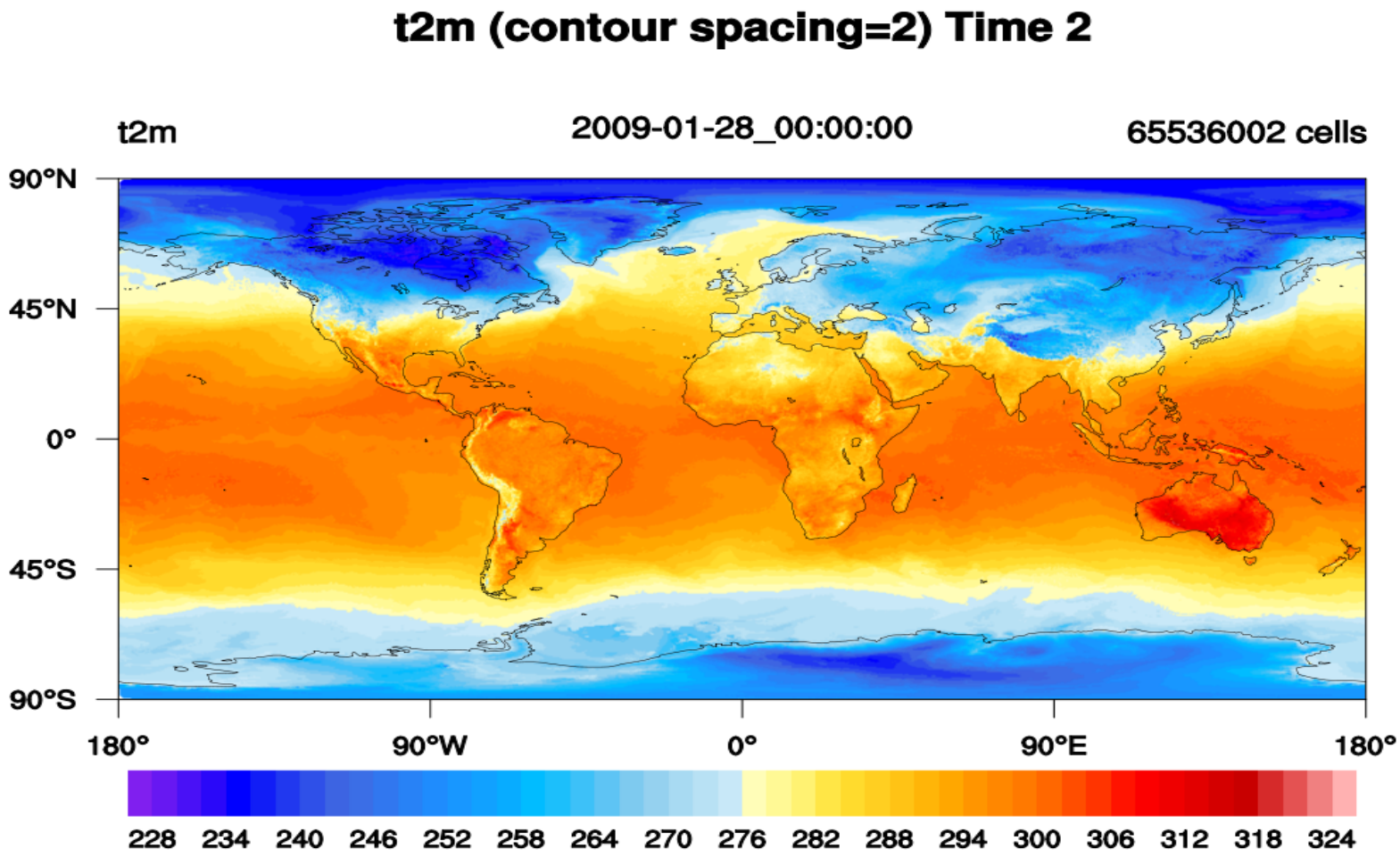
- *gsn_csm_countor_map*: Generic plotting routine: handles any collection of points with associated data (not just MPAS).
  - Creates NCL ContourPlot and MapPlot object
  - Create NCL Mesh data object
  - Call NCL draw routine (delaunay triangulation, connectivity)
  - NCL fill of raster array
  - Pass to GKS (Graphics Kernel System) in NCL
  - Pass to CAIRO library for drawing (PS, GIF, JPEG, PNG)
- Original time to read and draw 1 2d slice: infinite.

NCL raster rendering speedups for 65M cell MPAS grid

- Speedup generation of connectivity data structures
  - Better balancing of binary search trees for points and edges: ~ 46 minutes (1 plot)
  - Eliminate binary search trees altogether by generating data structures directly from output of triangulation routine: ~ 6.5 minutes

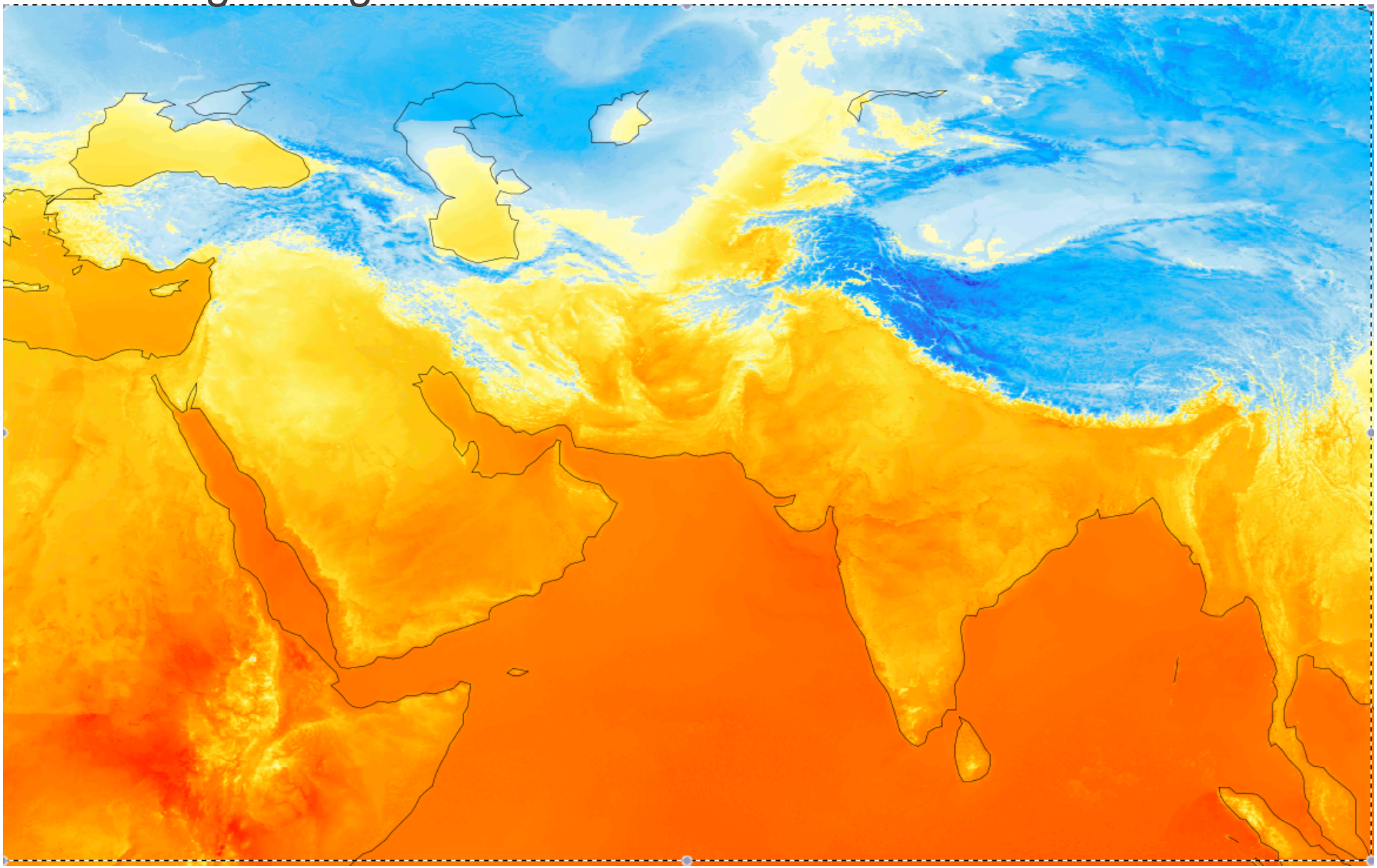
- Sort data into geographical blocks and add some OpenMP threading directives:
  - 2 threads: ~ 4.5 minutes
  - 4 threads: ~ 4 minutes
  - 6 threads: ~ 3.6 minutes
  - 8 threads: ~ 3.5 minutes

- Still do do.
  - Make line and "area fill" contouring work with chunking so that threading can be more widely applied. Currently contour lines don't always behave well at chunk boundaries.
  - Pass MOAB information on connectivity directly to routine.

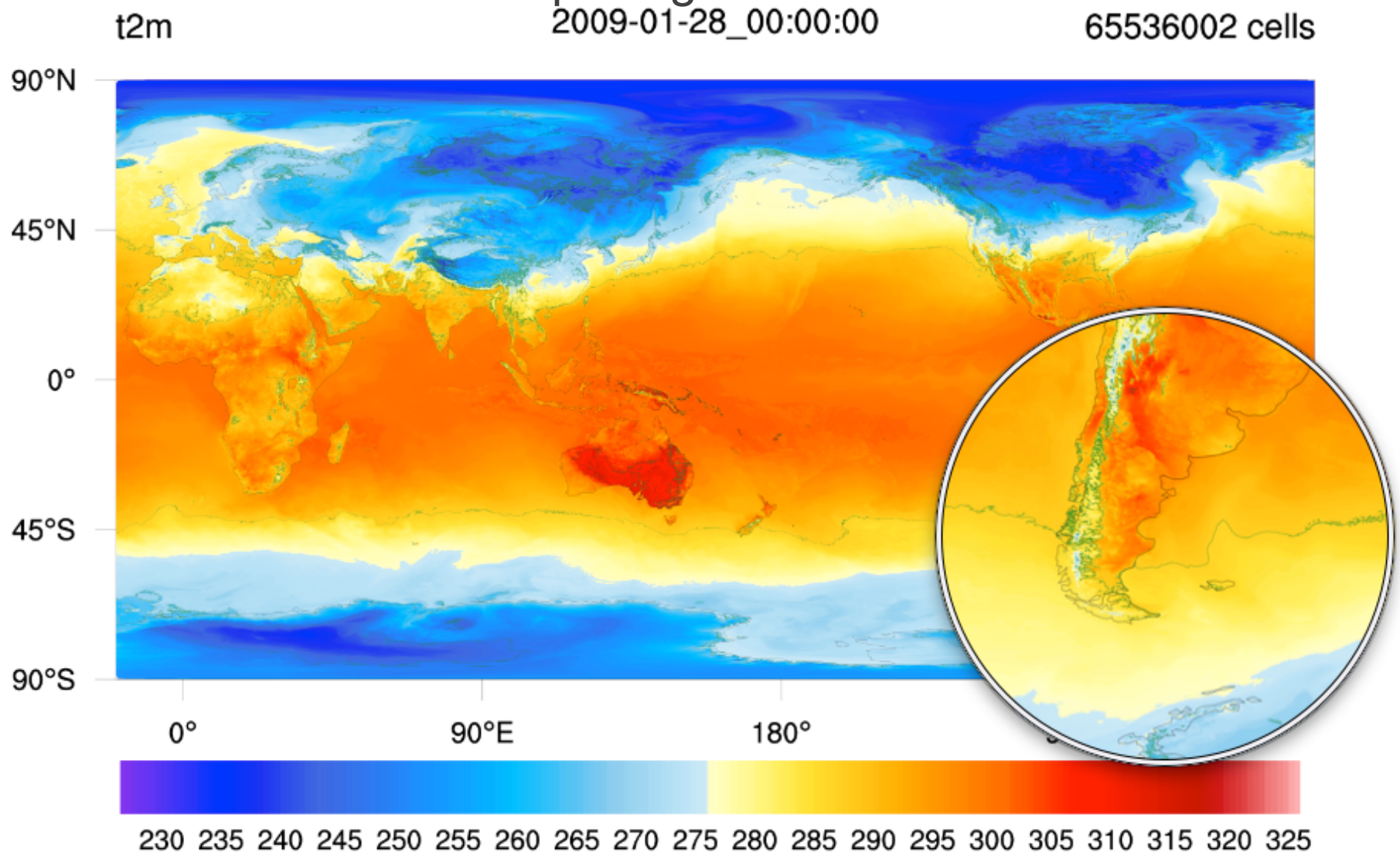

t2m (contour spacing=2) Time 2
t2m
2009-01-28_00:00:00
65536002 cells
90°N
45°N
0°
45°S
90°S
180°
90°W
0°
90°E
180°
228 234 240 246 252 258 264 270 276 282 288 294 300 306 312 318 324

Raster fill

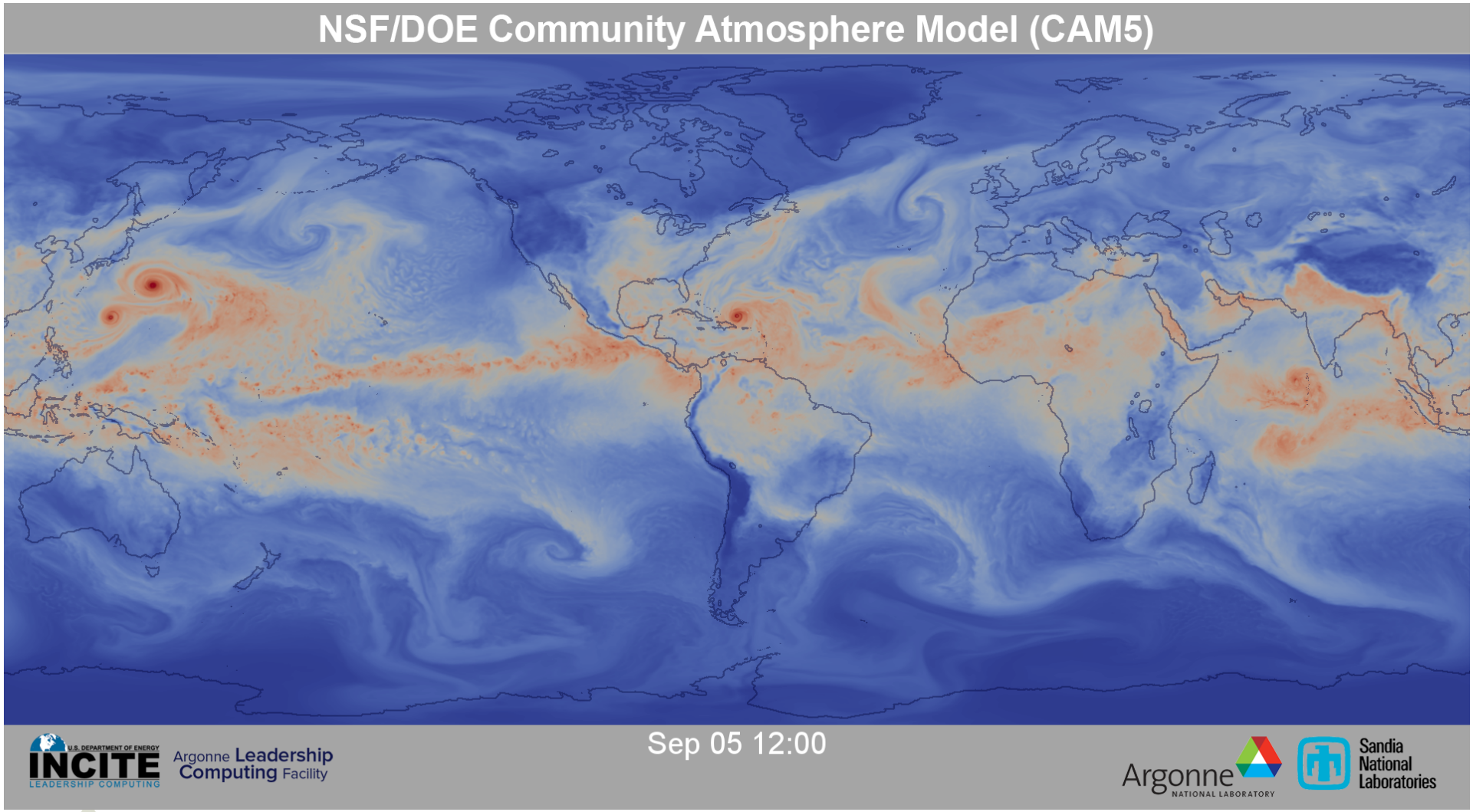
High-res grid subsection of t2m – raster fill



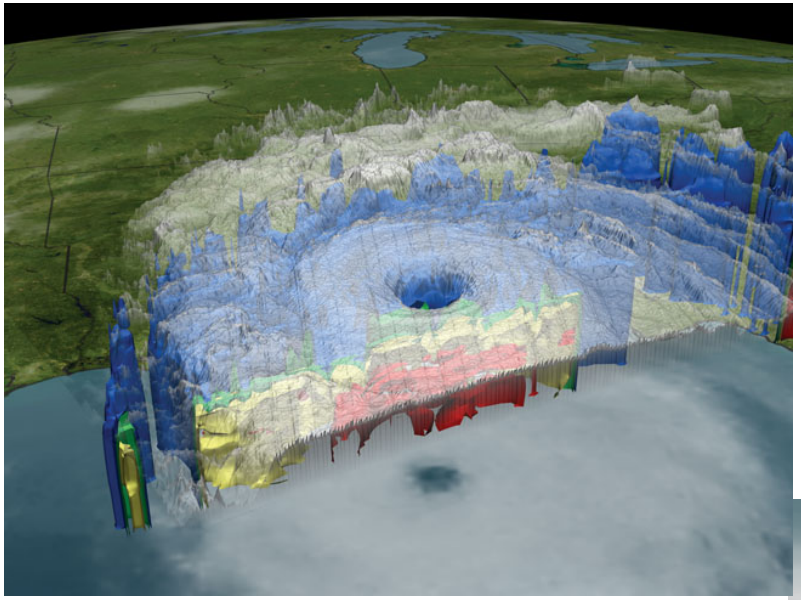
Raster fill spacing: 0.5° K -- line spacing: 20° K



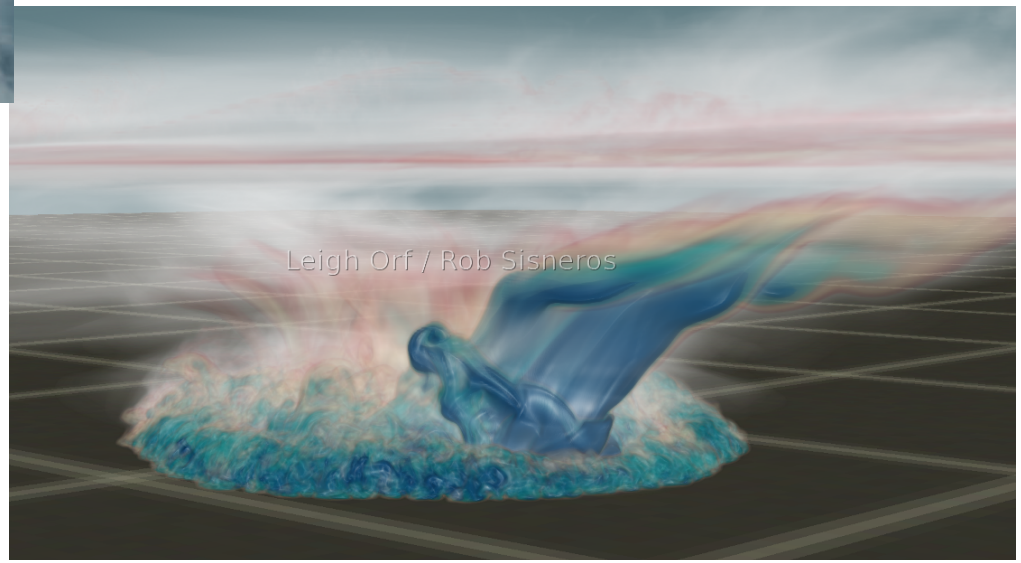
Ultra-high-resolution climate models have many smaller features embedded in them…



These smaller scale features need 3D for understanding (example! Not drawn from previous simulation.)



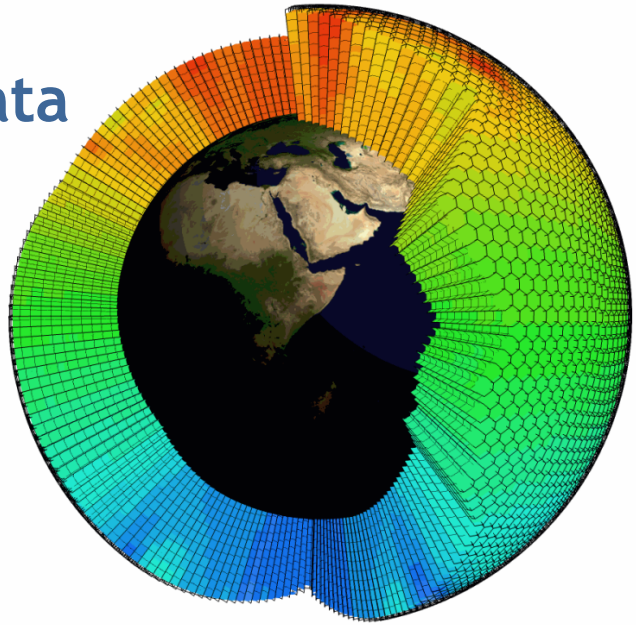
Hurricane Katrina simulation.
USGCRP “Our Changing Planet”, 2007



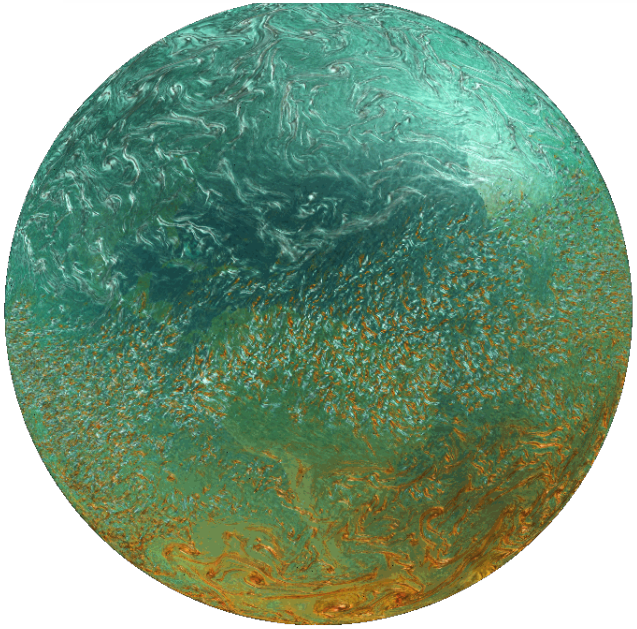
Microburst from thunderstorm.
“Visit” on Bluewaters.
Courtesy Prof. Leigh Orf,
Central Michigan University.
See viz showcase on Wednesday!

The "vis" in ParVis: Interactive Visualization of Large Geodesic Grid Data

Kwan-Liu Ma, Jinrong Xie, UC Davis



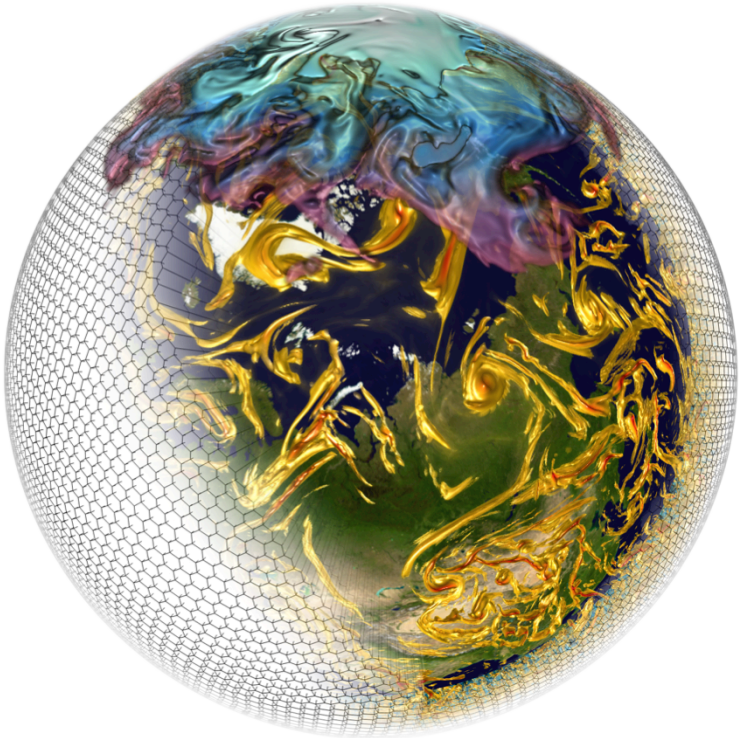
- Geodesic grids gaining popularity in the climate simulation community
- High quality 3D visualization offering new insights into the data
- GPU acceleration and parallelization enabling interactive visualization of large data

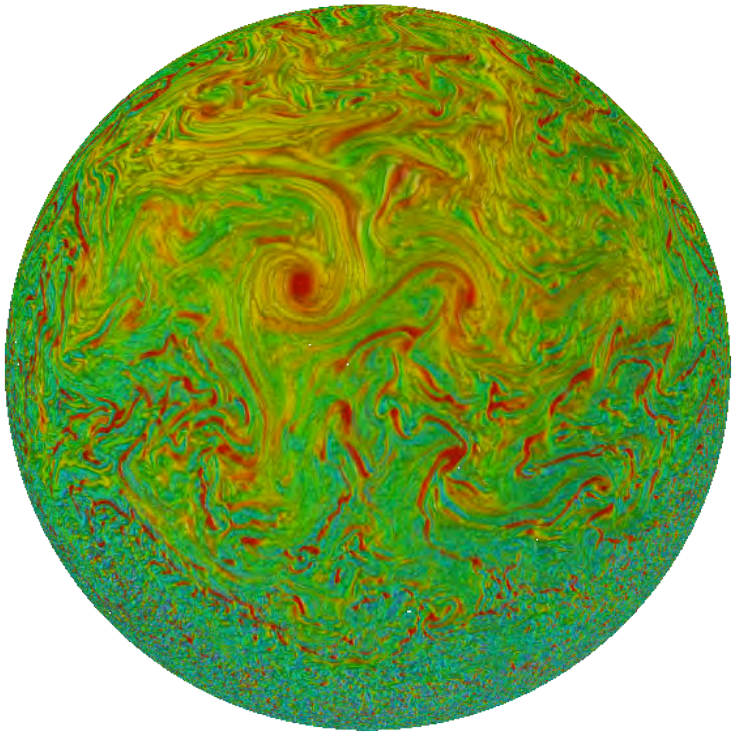


Paper: *Interactive Ray Casting of Geodesic Grids*, Jinrong Xie, Hongfeng Yu, Kwan-Liu Ma, EuroVis 2013.

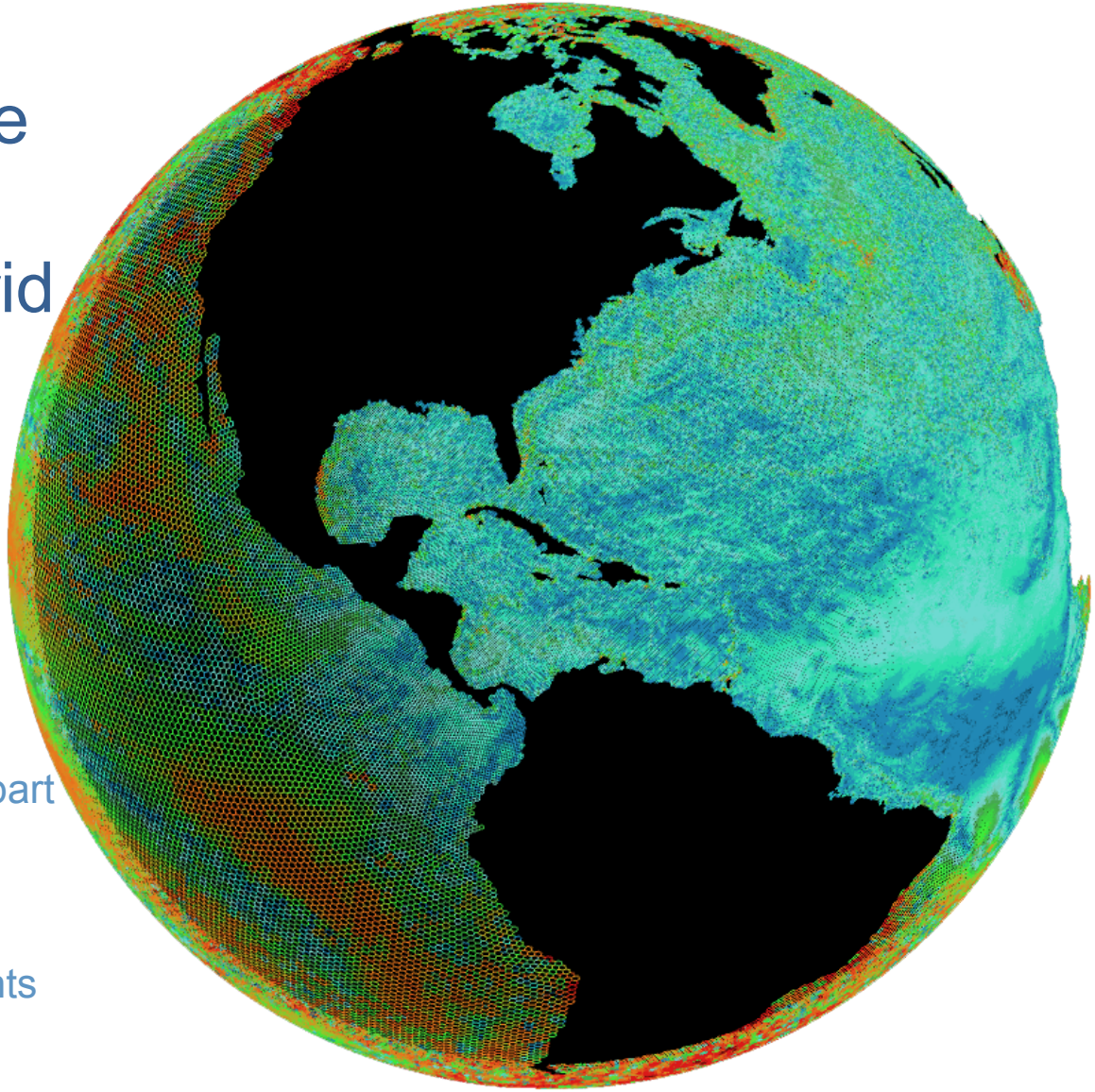
Atmosphere Vorticity of **GCRM** Data

- Rendering avoids reconstructing full 3D connectivity information.
- Avoid pre-partitioning of each hexagonal cell into multiple tetrahedral cells
- Directly uses original simulation data structure making it feasible for in-situ visualization.





Currently we are extending the volume rendering kernel for visualizing MPAS grid data



Ocean Temperature of MPAS Data

(This 3D viz research has been sponsored in part by ParVis and in part by the National Science Foundation through grants OCI-0905008, OCI-0850566 and OCI-0749227. (Department of Energy through grants DE-FC02-06ER25777, DE-FC02-12ER26072, and DE-SC0005334))

parVis

Supported by the Regional and Global Climate Modeling Program of the Office of Biological and Environmental Research of the U.S. Department of Energy's Office of Science

- http://trac.mcs.anl.gov/projects/parvis
- ParVis released software:
  - Swift-enabled task-parallel standard diagnostic scripts
    - For all major components
    - Available on LCFs
    - Fast but still assumes lat-lon
  - ParGAL source code
  - ParNCL source code and binary (beta).
    - Available on LCFs
- Planned output:
  - 1.0 release of ParGAL and ParNCL with support for MPAS grids (by end of this year).
  - Implement some of standard atmosphere diags for native CAM-SE grid using ParNCL