Introducing ParVis

Robert Jacob

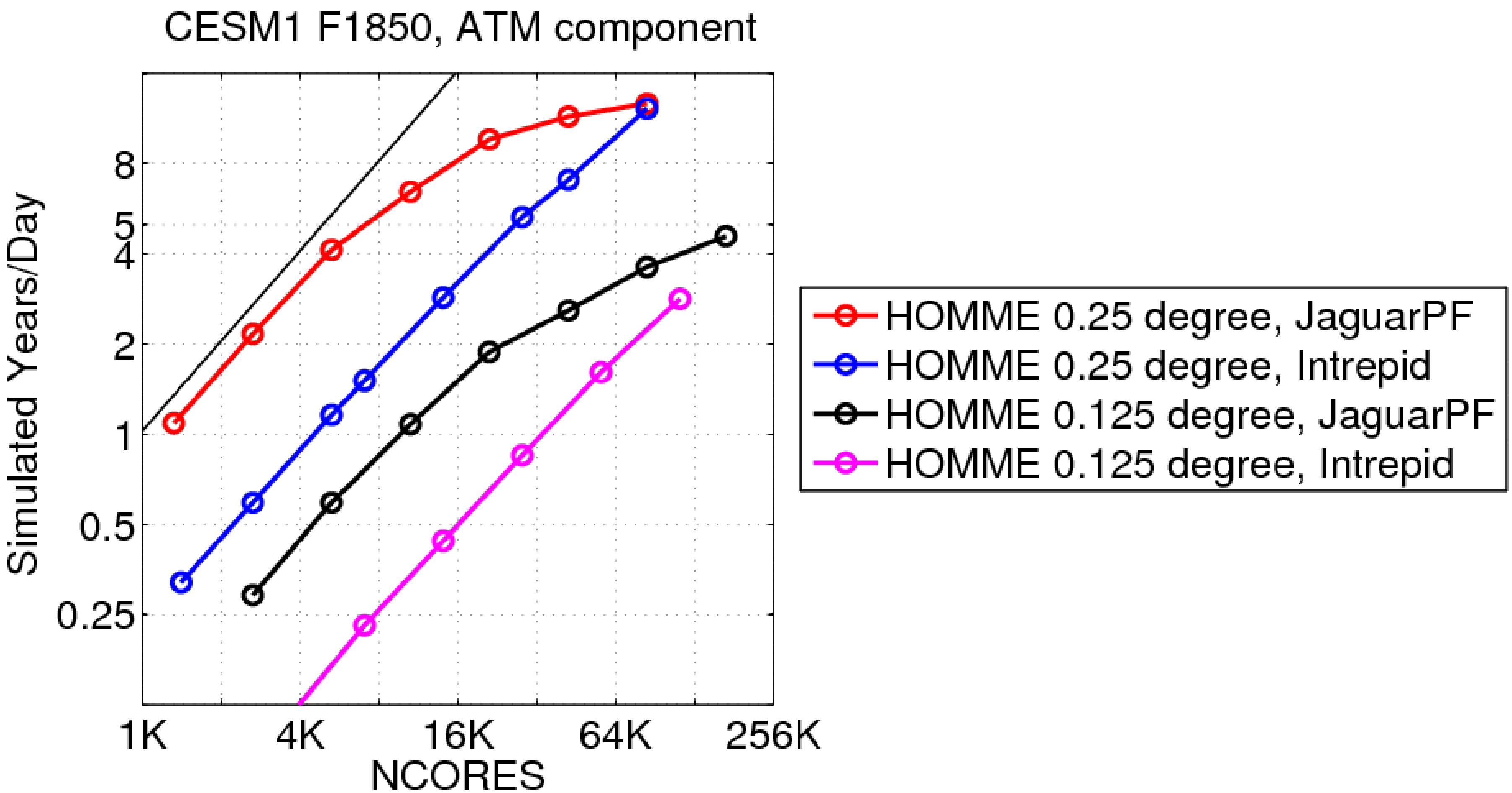
16th Annual CESM Workshop
June 23rd, 2011

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

Motivation:

Ability to gain insight from current and future climate data sets ≠ Capability of current tools

Climate models are now running on 100K cores…



From Mark Taylor, SNL

…and they are outputting lots of data.

- CAM HOMME 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

- CSU GCRM 4km horizontal, 100 levels
  - Single 3D variable (cell center): 16 GB
  - Single 3D variable (cell edge): 50.3 GB
  - Single history file 571 GB
  - 1 year of monthly output: 6 TB
  - 100 years of monthly: .6 PB

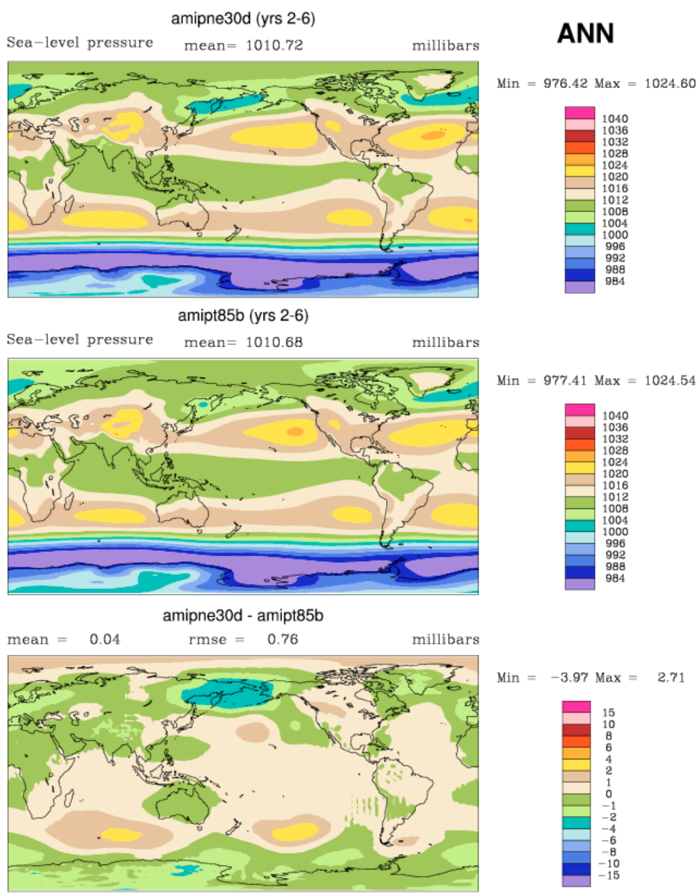
Existing Data Analysis and Visualization (DAV) tools have not kept up.

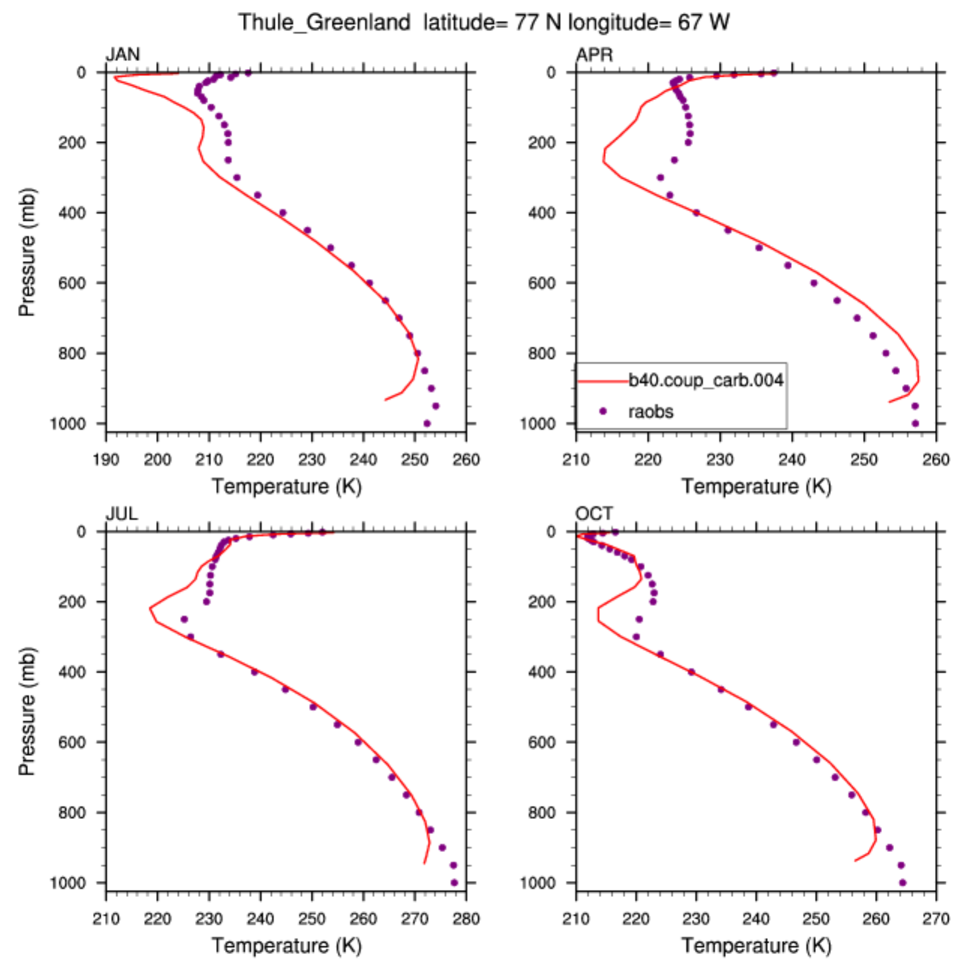
- NCAR Command Languagel (NCL)
- Climate Data Analysis Tools (CDAT)
- Grid Analysis and Display System (GrADS)
- Ferret

} No parallelism

ParVis will speed up data analysis and visualization through data- and task-parallelism.

Parvis philosophy: Insight about climate comes mostly from computationally undemanding (to plot) 2D and 1D figures.



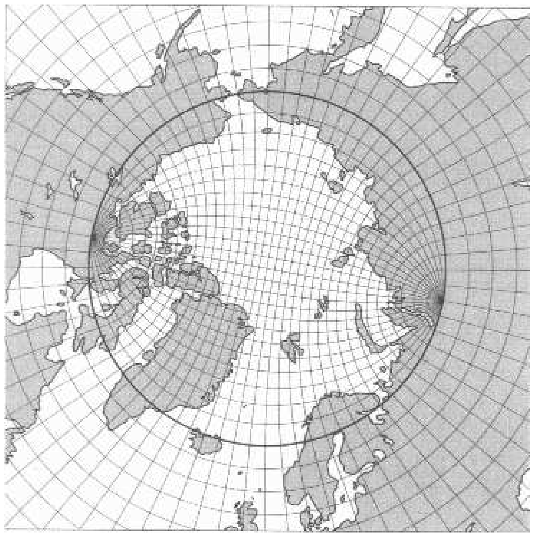


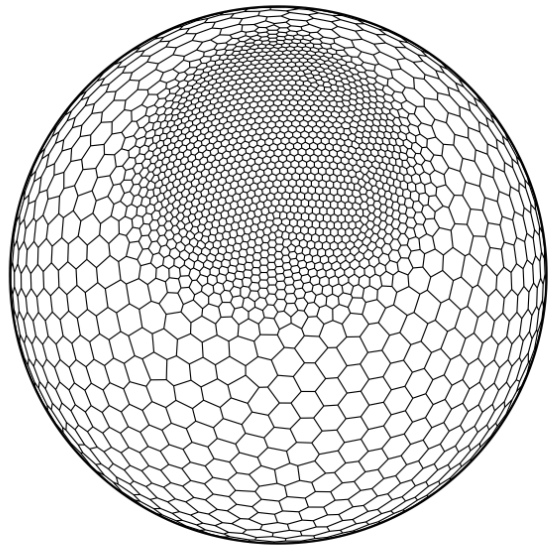
Why? The atmosphere and ocean have a small aspect ratio; 10,000 km vs. 10 km.

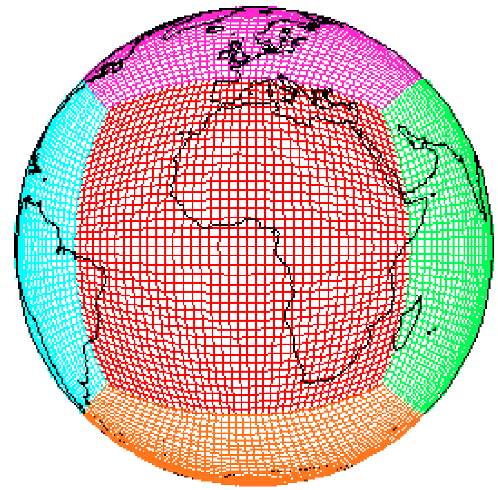
(Philosophy cont'd) The problem is more in making the data for the figures (not directly from the models).

- **Post-processing is an inextricable part of visualization of climate models.**
- **It is the post-processing where the introduction of parallelism could have to largest impact on climate science using current visualization practice**
- **But we should also look to putting powerful graphics processors to work for climate insight.**

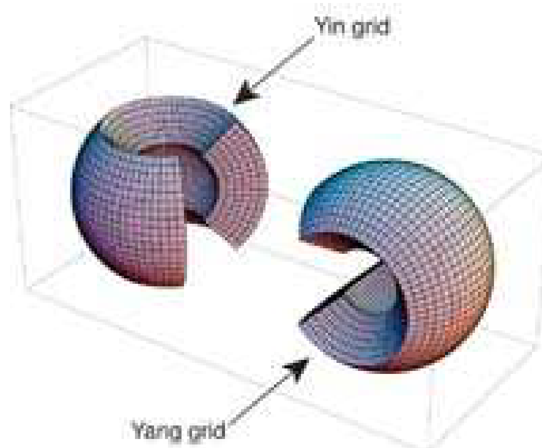
Along with increased data sizes, the data is coming out on new, unstructured or semi-structured grids.







All climate DAV tools require lat-lon grids for their internal adnalysis functions.



Parvis will be more flexible.

Approach

- Use existing tools to speed-up development.
- As much as possible, preserve well-established workflows for analyzing climate data, just speed them up.
- There is a problem *right now* so provide both immediate and long-term help
- Assemble a multi-disciplinary and multi-institutional team to carry out the work.

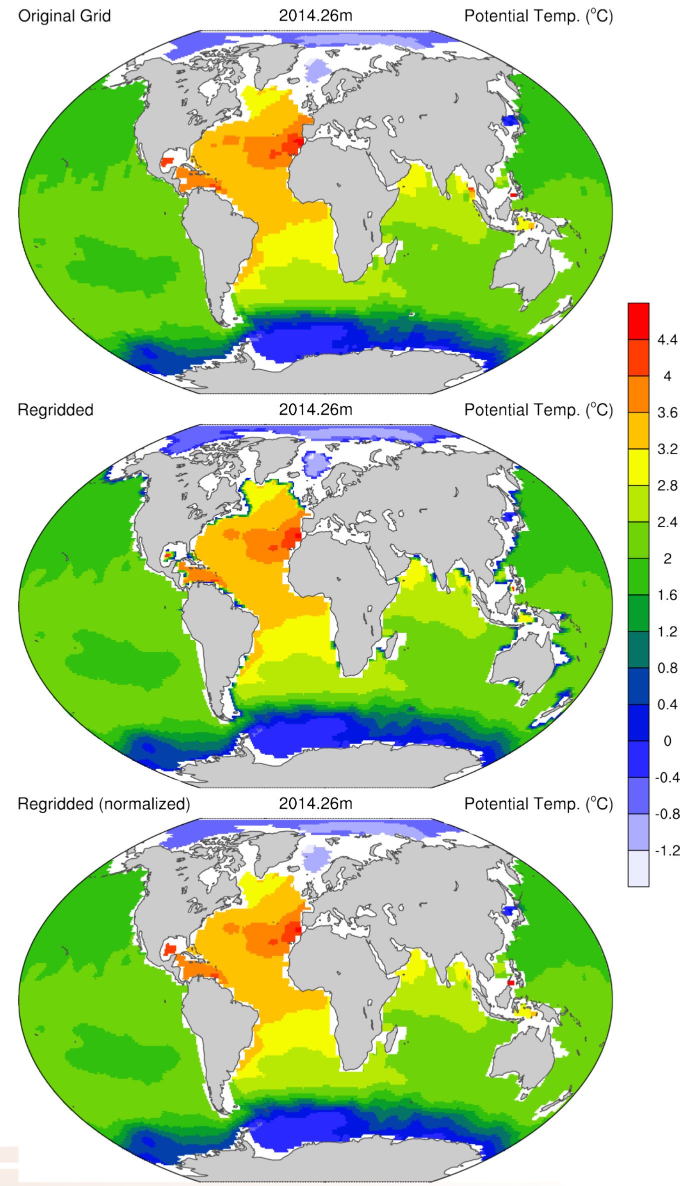
NCAR Command Language (NCL)

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/



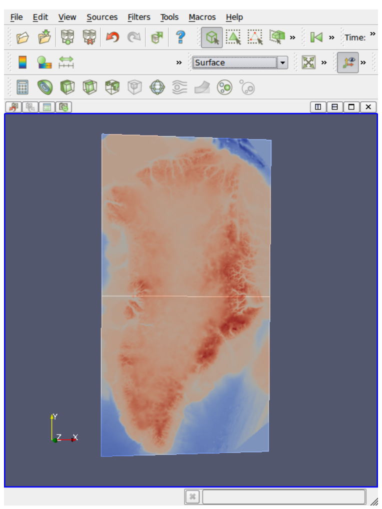
PNetCDF: NetCDF output with MPI-IO

- 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

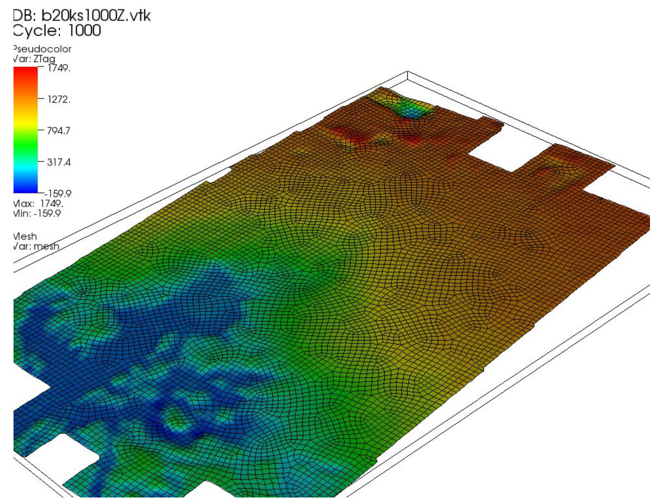
Mesh-Oriented datABase (MOAB)

- MOAB is a library for representing structured, unstructured, and polyhedral meshes, and field data on those meshes
- Uses array-based storage, for memory efficiency
- Supports MPI-based parallel model
  - HDF5-based parallel read/write on (so far) up to 16k processors (IBM BG/P)

- Interfaces with other important services
  - Visualization: ParaView, VisIt
  - Discretization: Intrepid (Trilinos package)
  - Partitioning / load balancing: Zoltan



Greenland ice bed elevation (in Paraview/MOAB)



Jakobshavn ice bed (in VisIt/MOAB)

Intrepid (software, not to be confused with the BlueGene/P at ALCF)

***IN**teroperable **T**ools for **R**apid d**E**velo**P**ment of compat**I**ble **D**iscretizations*

A **Trilinos package** for compatible discretizations:

- An extensible library for computing operators on discretized fields
- Will compute div, grad, curl on structured or unstructured grids maintained by MOAB!

When fully deployed (~2011) will provide

- support for **more discretizations** (FEM, FV and FD)
- optimized **multi-core** kernels
- optimized **assembly** (R. Kirby)

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

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

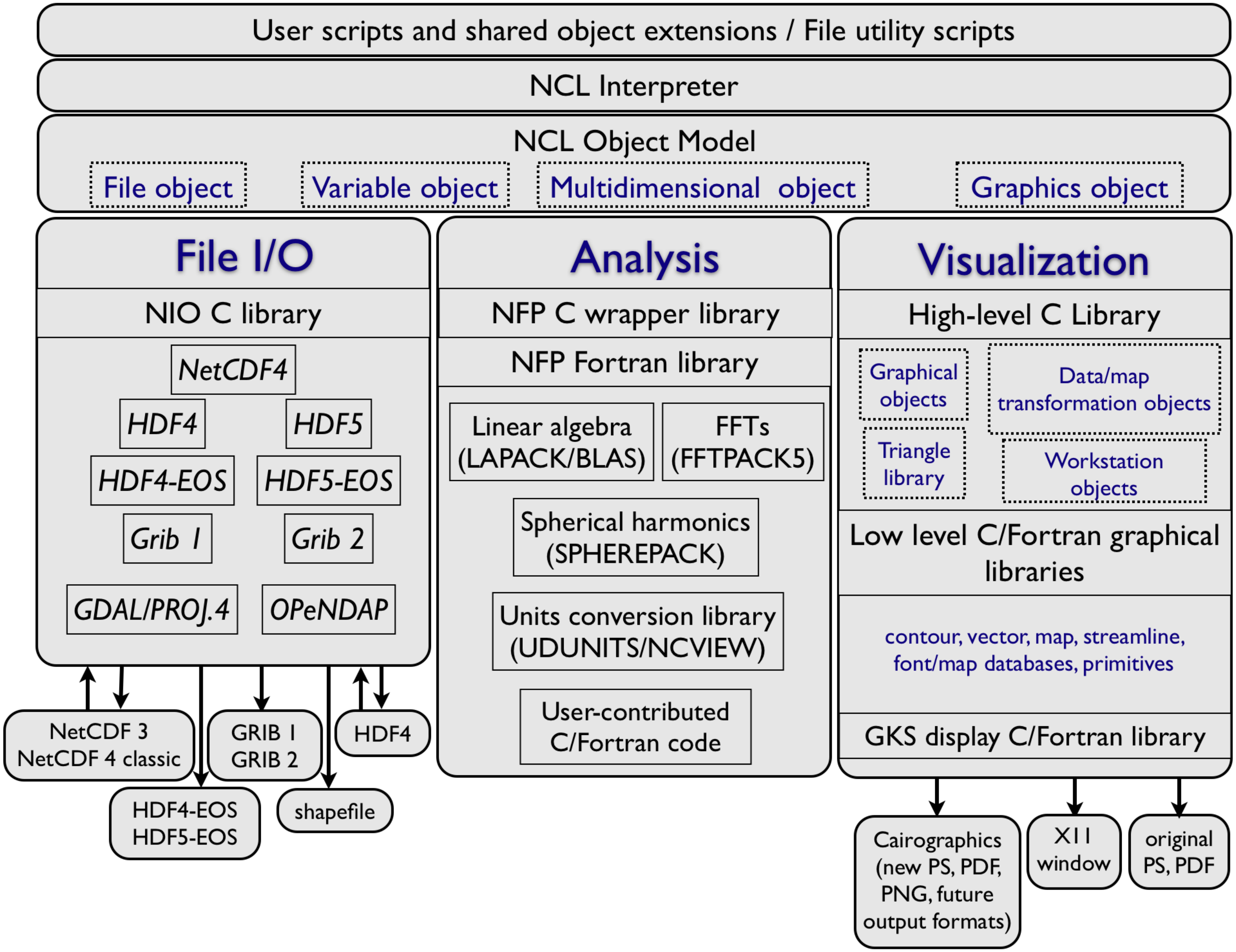
ParCAL: Parallel Climate Analysis Library

- Will be built using PnetCDF, MOAB and Intrepid
  - MOAB and Intrepid have already solved the hard problem of how to represent and operate on structured and unstructured grids distributed over processors.

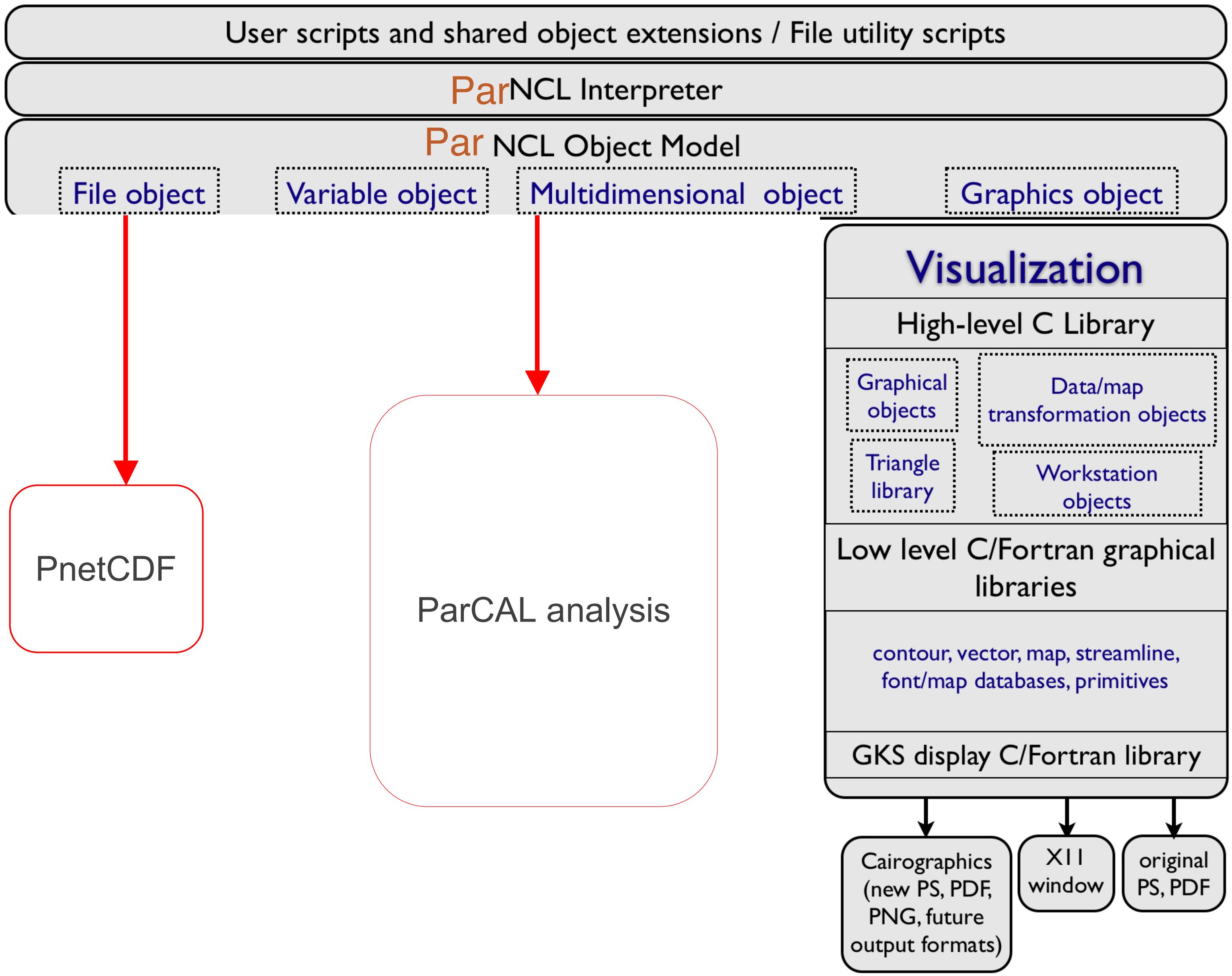
- Will provide data-parallel core to perform typical climate post-processing currently done by either NCL or NCO.

- **Will be able to handle unstructured and semi-structured grids in all operations by building on MOAB and Intrepid. Will support parallel I/O by using PnetCDF.**

NCL architecture



ParNCL architecture



Parvis Hardware Model

- Data Analysis Center's will continue to be a main venue for performing climate-model post-processing.

- Eureka at ALCF
  - 100 compute nodes: each with (2) 2.0 GHz quad-core Xeon servers with 32 GB RAM
  - 200 NVIDIA Quadro FX5600 GPUs in 50 S4s
  - Memory: More than 3.2 terabytes of RAM
  - Peak Performance: More than 111 mostly single precision teraflops of computation use a fraction of electricity compared to alternative architectures.
  - Shares same disk as the big BlueGene/P

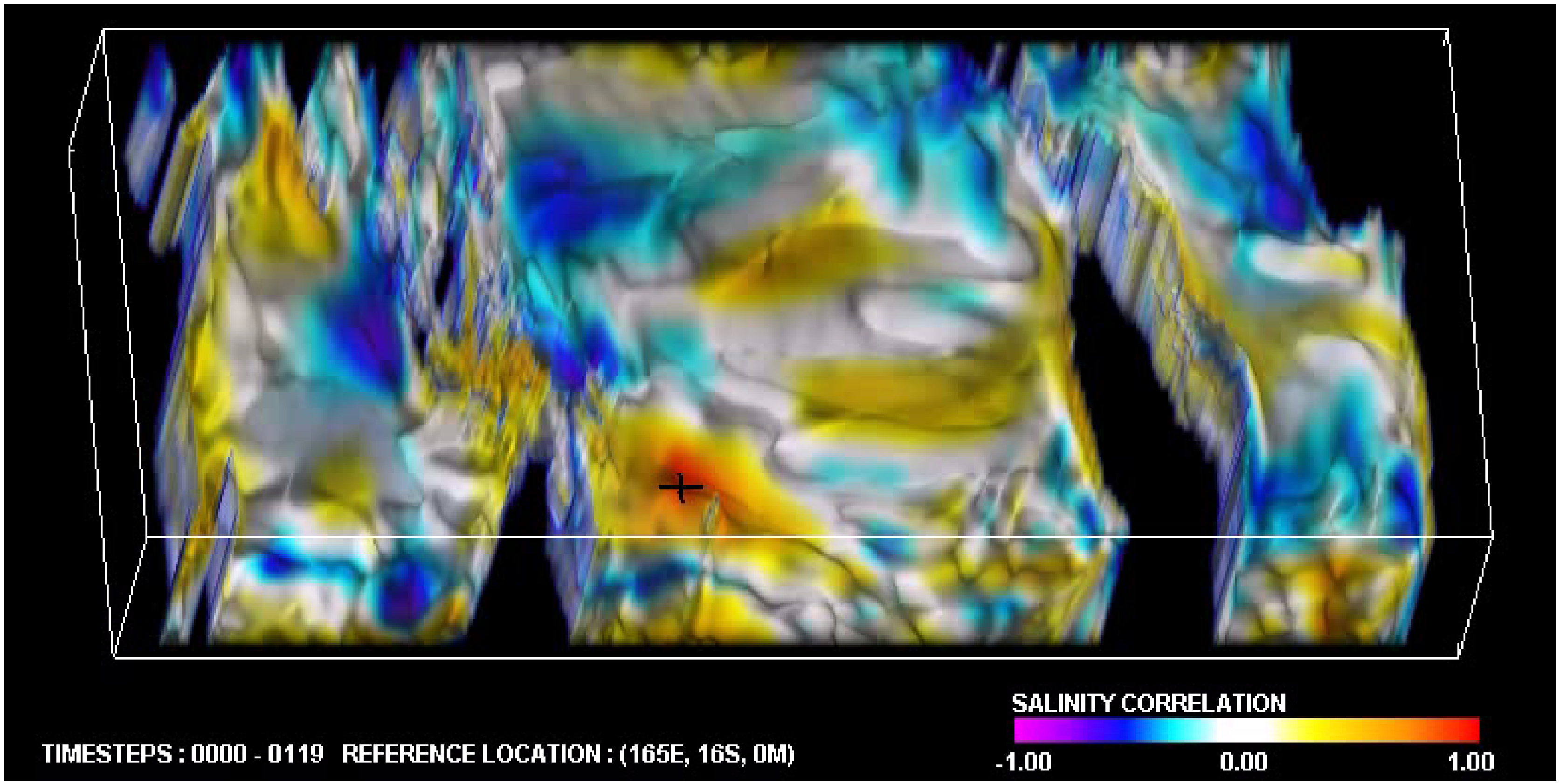
Parvis plan for hardware uncertainty

- Network bandwidth may not grow as fast as data.
- Available disk may not grow as fast as data.
  - Parvis will investigate compression schemes to incorporate in to NetCDF.
- Cloud computing may replace dedicated analysis clusters
  - Parvis will investigate cloud-friendly algorithms for climate analysis based on MadReduce

Parvis will provide immediate help with task-parallel versions of diagnostic scripts using *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.*

Parvis and new visualization: New ways to use 3D images in climate analysis (Interactive Correlation Analysis and Visualization ICAV)



Salinity self-correlation

Parvis Timeline

- Feb 22, 2010: Proposal submitted
- July 7, 2010: Official notice of selection (funds arrived that month.)
- September 29-30, 2010: **Parvis Kickoff meeting at Argonne**
- March 1, 2011: Parvis programmers finally hired!
- April 11-12, 2011: 2nd Parvis team meeting at NCAR.

Personnel

- At Argonne:
  - PI: Robert Jacob
  - PnetCDF developers: Rob Ross, Rob Latham
  - MOAB developer: Tim Tautges
  - Swift developer: Mike Wilde
  - Cloud computing for climate: Jay Larson
  - Visualization and project management: Ian Foster, Mark Hereld
  - Programmers:
    - Jayesh Krishna (**started March 1st**)
    - Xiabing Xu (**started March 7th**)
    - Sheri Mickelson (Argonne/Uchicago Computation Institute)

- At Sandia:
  - Intrepid developers: Pavel Bochen, Dennis Ridzal, Kara Peterson
  - CAM HOMME developer; Parvis testing: Mark Taylor

Personnel

- At PNNL
  - Compression Techniques: Jian Yin
  - GCRM viz developer; Parvis testing: Karen Schuchardt

- At NCAR
  - NCL project lead: Mary Haley
  - NCL head developer: Dave Brown
  - NCL developer: Rick Brownrigg, Wei Huang
  - NCL scientific consultant: Dennis Shea
  - NCAR vis section head, ESG co-PI: Don Middleton
  - CESM lead software engineer: Marian Vertenstein

- At UC-Davis
  - Developer of ICAV tool: Kwan-Lu Ma, Jinrong Xie

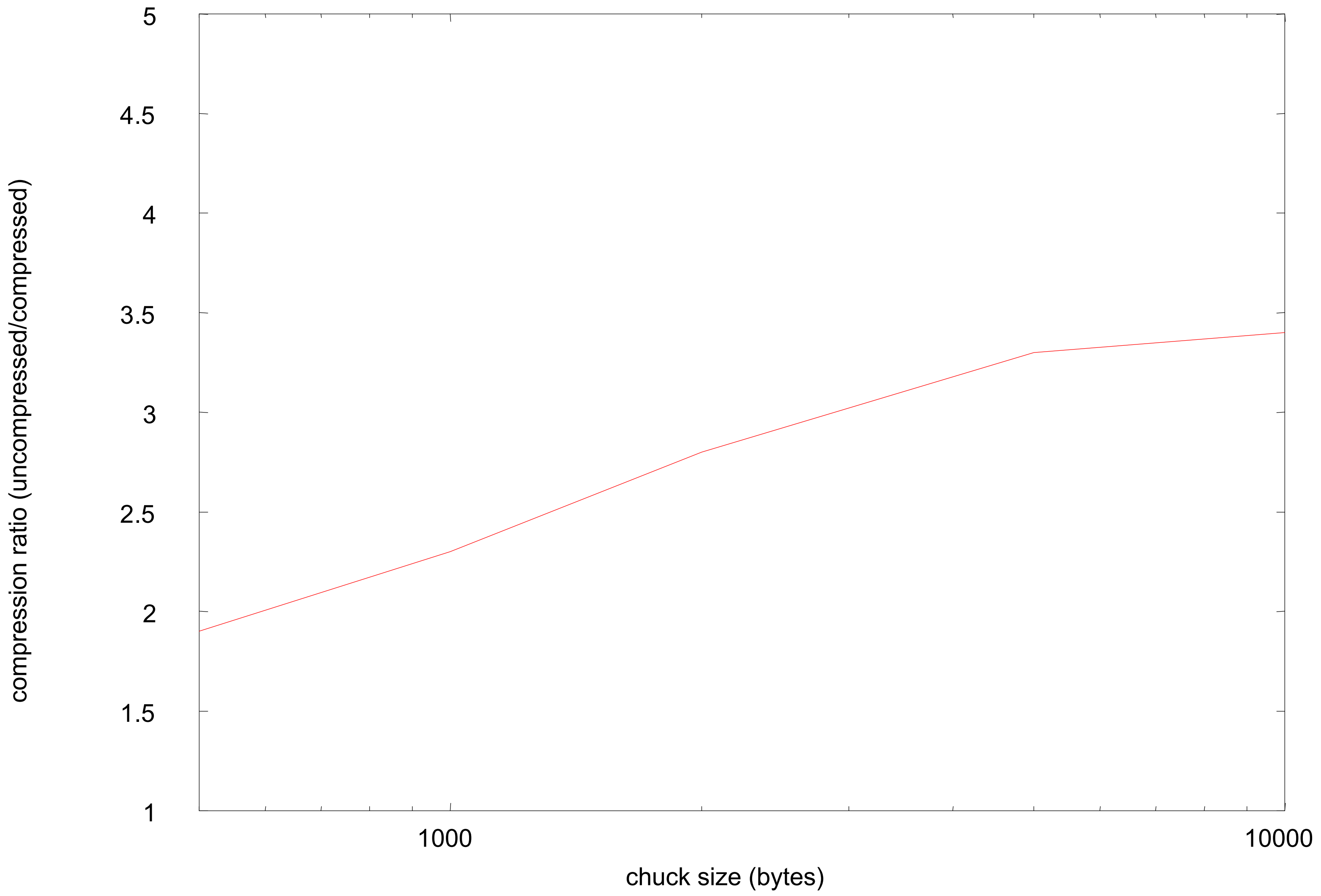
Advisory Panel

- Atmosphere: David Randall (CSU), William Gustafson (PNNL)
- Ocean: Gokhan Danabosaglu (NCAR)
- Sea Ice: Cecelia Bitz (U. Washington)
- Land: David Lawerence (NCAR)

Progress so far….Compression

- Our compression schemes contain two phases: the first phase predicts the next value based on the previous values, the second phase encodes the next value with entropy-based encoding
- We pick up a simple scheme that use neighbor cell value for predication, which yields good compression ratios and compression throughput
- Hardware: Intel Xeon 2.26Ghz, 6Gbytes memory
- Data: GCDM high resolution (about 15km) climate data
- Experiments with various chunk sizes because chunk size can affect compression ratios and degree of parallelism
  – Small chunk size: low compression ratio, high degree of parallelism
  – Big chunk size: high compression ratio, low degree of parallelism

Compression results



Compression will help with data transfer. Also helping - Globus Online

- Globus Online provides a managed file transfer service
  - features include: File Movement, Globus Connect, Performance Optimization, Error Retry, Monitoring, Endpoint Management, and Conditional Transfers and Integrity Checking.
- We used Globus Online to transfer files for testing from NCAR to Argonne
  - 144 528MB files (75GB total) in 10.5 minutes
- Many endpoints supported by default (NERSC, Teragrid, …)
  - Can add your own by downloading app to your laptop or workstation (Mac, Linux, Windows)
- Get free account at www.globusonline.org

Using Swift with diagnostic packages

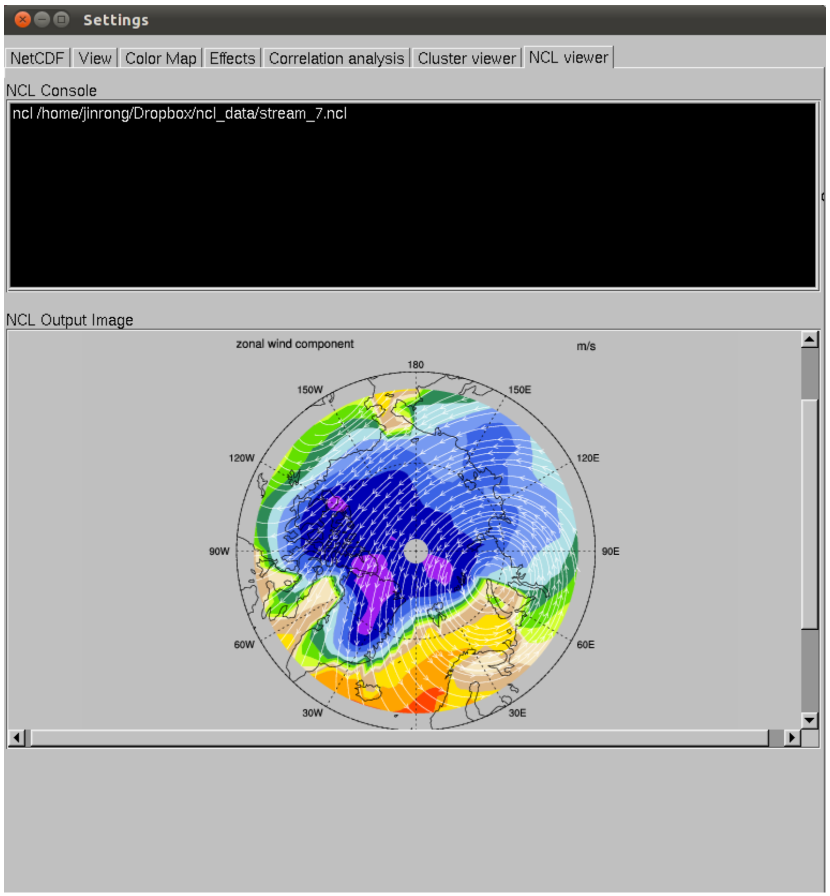
- Updated AMWG diagnostics to latest version (diag110520.csh)
  - "The AMWG diagnostics package produces over 600 postscript plots and tables in a variety of formats from CCSM (CAM) monthly netcdf files."

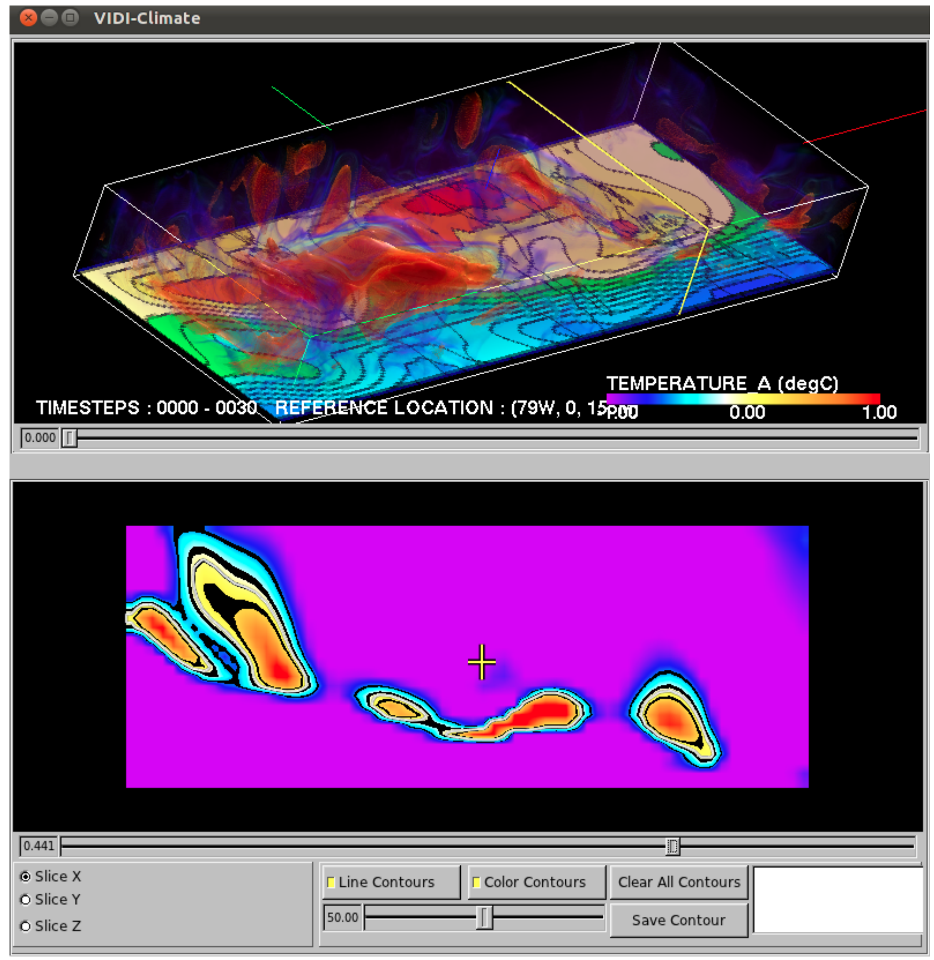
- Now using Swift in both the "data" and "graphics" parts of diagnostics

- Timings with 10 years of 0.5 degree CAM data comparing with observations:
  - Original csh version on one process: 71 minutes
  - Swift and 16 processes: 22 minutes

- Further speedup possible using Pagoda in place of NCO. See talk by Jeff Daily.

Combining ICAV with NCL:
Combining familiar 2D images with 3D views





ParCAL and ParNCL

- First version of ParCAL (with MOAB and PnetCDF) is complete.
  - Implemented in C++
  - Unit testing and in-line documentation with doxygen

- First version of ParNCL working!   Executes an NCL script in parallel with no modifications.
  - Previous work on 64-bit NCL essential
  - NCL headers made C++ compliant
  - Only one NCL function re-written in parallel so far: dim_avg_n

Future work…(lots of coding)

- Continue to develop ParCAL. Integrate Intrepid functionality.
- Implement parallel versions of more NCL internal functions
- Expand test coverage of ParNCL
- Extend Swift approach to other diagnostic packages.
- Bring compression techniques in to PnetCDF.
- Expanding ICAV capabilities

What you can do…

- Tell us where bottlenecks are in your workflow. What NCL commands take to long or need to much memory?

- Provide grids and datasets that don't work well with your current tools.

- Attend our session at 2011 Fall AGU:
  - IN06: Challenges in Analysis and Visualization of Large Earth Science Data Sets.
  - Conveners: Robert Jacob, Dean Williams and Wes Bethel