Argonne NATIONAL LABORATORY

... for a brighter future

U.S. Department of Energy

THE UNIVERSITY OF CHICAGO

Office of Science U.S. DEPARTMENT OF ENERGY

A U.S. Department of Energy laboratory managed by The University of Chicago

Parallel-NetCDF and CCSM

Parallel-Netcdf:

Rob Latham, Rob Ross Argonne National Laboratory

Wei-Keng Liao, Alok Choudhary Northwestern University

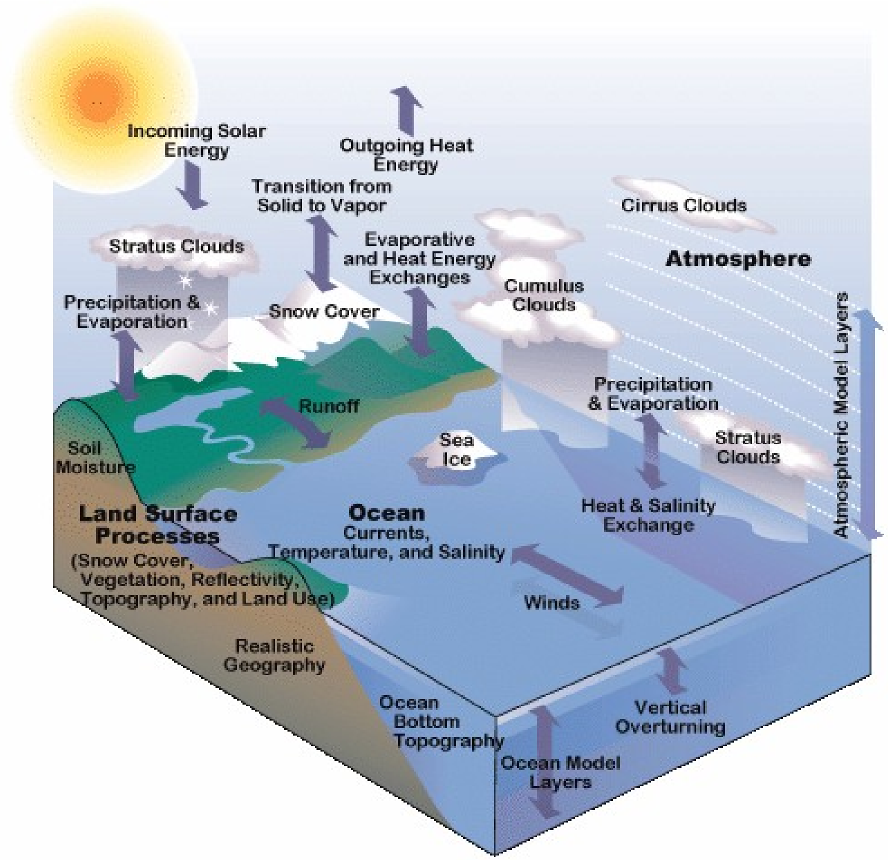
CCSM I/O:

Robert Jacob, Ray Loy Argonne National Laboratory

John Dennis, Mariana Vertensten, Tony Craig National Center for Atmospheric Research

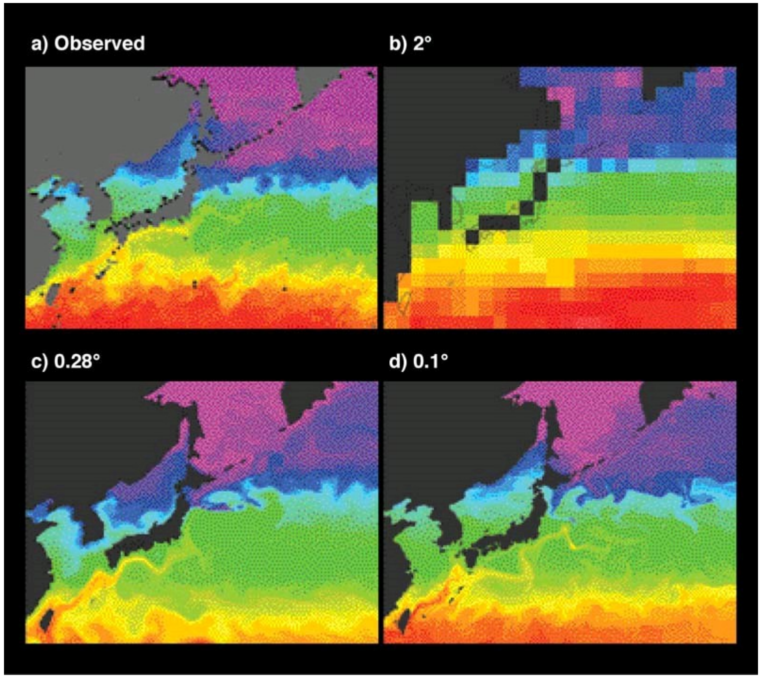
CCSM project

- CCSM: Community Climate System Model
  - model climate
  - simulate historical situations
  - predict future
- Four (mostly) independent components
  - atmosphere (CAM)
  - land (CLM)
  - ocean (POP)
  - sea ice (CICE)
- NCAR-based; many lab, university collaborators
  - 100s of institutions
  - freely available
- One of the IPCC models
  - Nobel-prize winning!



Scalability and leadership class machines

- Many interesting climate structures don't show up in simulation until resolution reaches 1/10th degree
- ...but need at least 5 simulated years/day
- Application groups adapting to new hardware reality
  - Linux clusters:
    - *single core performance levels off*
    - *many cores now*
  - Petascale clusters:
    - *tens of thousands of cores*
    - *each core: small memory, low performance*
  - no more “riding hardware” for performance improvements



BGP Architecture

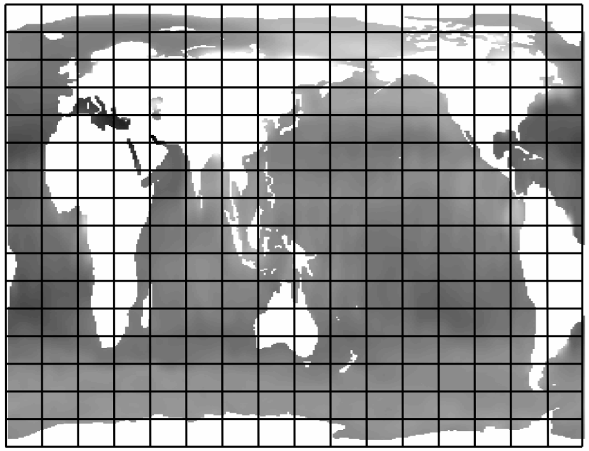
- Similar to BGL design with evolutionary improvements
  - 1024 Compute Nodes (CN) per rack
  - Each node: 4-core PPC 450 @ 850 MHz
  - 2 GB ram per CN
- I/O node forwards system calls
- Argonne “Intrepid” system (a target system for CCSM redesign)
  - 1 I/O node per 64 CN
  - 32k CN (128k cores)
- 512 MB per core:
  - 2x mem/core compared to BGL
  - staging all I/O on rank 0 (still) impossible!

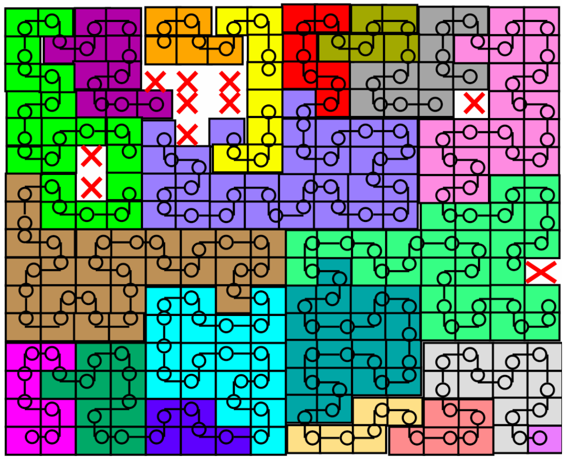
CCSM redesign

- Old CCSM:
  - master coordinator process
  - all I/O from rank 0
    - *this worked great on old Cray systems*
- “Concurrent” vs. “Sequential”
  - concurrent:
    - *each model component runs on a subset of processors*
  - sequential:
    - *components run one at a time*
    - *over all available processors*
- New CCSM
  - less serialization
  - eliminate all global arrays
  - I/O from everyone
    - *Create a new I/O library (PIO) for use by all components*

Data layout

- Load balancing: domain split up into space filling curves
  - "cube the sphere"
  - weight regions by complexity
  - partition equal amounts of "work"
- Example: POP (ocean)
  - can ignore land-only grids
- Decomposition has favorable impact on CPU scalability, less favorable impact on I/O





Parallel-NetCDF

- Joint ANL-NWU project
- Based on original “Network Common Data Format” (netCDF) work from Unidata
  - Derived from their source code
- Same Data Model as serial netCDF:
  - Collection of variables in single file
  - Typed, multidimensional array variables
  - Attributes on file and variables
- High Level Data Mode interface
  - traditional NetCDF access
  - vara (subarray), varm (mapping), vars (strided)
- Collective I/O
- Flexible Mode interface
  - Extends 'vara' 'varm' 'vars' accesses
  - User defines MPI datatype describing layout in memory

CCSM I/O and pnetcdf

- Application access pattern not exact fit for pnetcdf
  - SFC irregular shape doesn't match multidimensional array accesses
  - Can't use pnetcdf flexible mode: SFC regions cannot guarantee monotonically non-decreasing
- MCT (model coupling toolkit)
  - re-arranges data from component layout into 1-D linear arrays
  - takes 1-D arrays and re-arranges into desired component layout
- PIO operations
  - Implements I/O in 1-D arrays
  - (optional) reduce I/O to “worker nodes”
    - *similar to MPI-IO two-phase optimization*
  - Supports I/O through pnetcdf, MPI-IO, and can fall back to serial netcdf

POP-IO

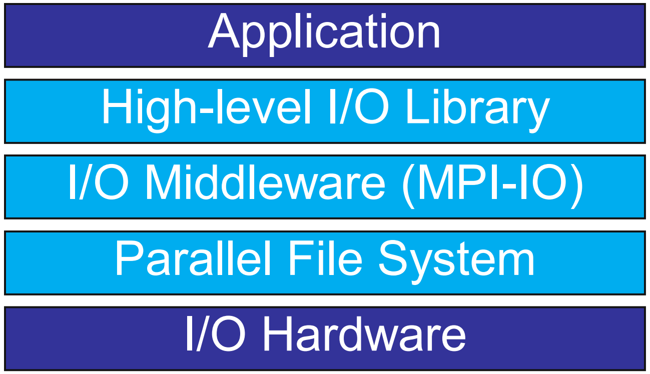
- Application I/O kernel written by John Dennis (dennis@ucar.edu)
- All climate/ocean science removed
- I/O same as full model component
  - e.g. keeps decomposition routines
  - measures I/O performance with and w/o rearrangement step
  - measures I/O performance with MPI-IO directly and with Pnetcdf
- Already been used in I/O study on BGL (Watson) hardware
  - excellent scalability up to 5k processes
  - performance “good enough” at 30k processes, but could be better
  - MPI-IO aggregation hints appear to be either ignored or make no difference in perf

Why pnetcdf?

- climate/weather groups big users of NetCDF data sets
  - pnetcdf fully compatible with serial NetCDF file format
- no parallel I/O in NetCDF-3
  - one file per process
    - *inefficient*
    - *untenable at 128k processes*
  - forward all I/O to rank 0
    - *impossible on BGL: not enough memory*
- pnetcdf provides parallel I/O, common file format

Why not straight to file system?

- I/O software stack
- File system
  - aggregates storage devices
- MPI-IO:
  - collective I/O
  - lower-level but sophisticated optimizations
- High-level library:
  - better interfaces for application developers
    - *already thinking about interfaces closely tailored to application needs*
  - portable, self-describing
  - metadata (attributes)



More information

- CCSM
  - http://www.ccsm.ucar.edu
- Parallel-NetCDF
  - http://www.mcs.anl.gov/parallel-netcdf
- PIO
  - http://swiki.ucar.edu/ccsm/93