Community Earth System Model (CESM)

Sheri Mickelson

Mathematics and Computer Science Division,
Argonne National Laboratory

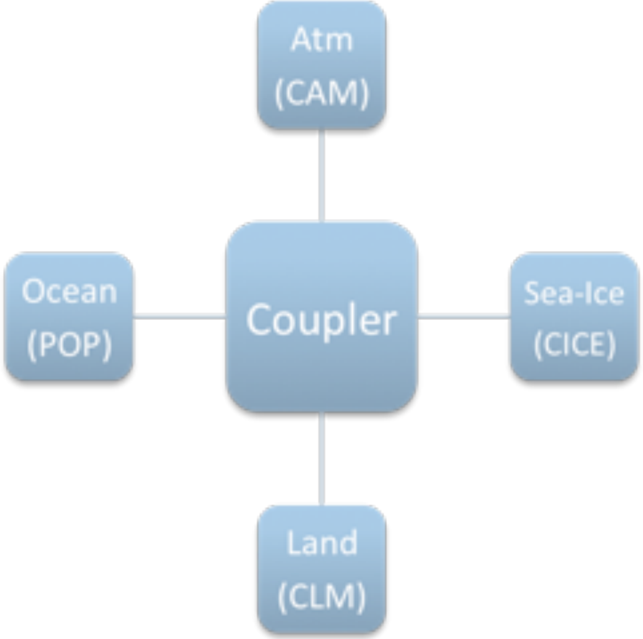
Computation Institute, Argonne/University of Chicago

SciDAC SA: A Scalable and Extensible Earth System Model for Climate Change Science

- PIs: John Drake (ORNL) and Phil Jones (LANL)
- Involves ANL, PNNL, LBNL, LLNL, BNL
- Works closely with the National Center for Atmopsheric Research (NCAR) on climate model development
- Focuses on chemistry, ice sheet modeling, finite element methods, parallelism, performance, and testing

What is CESM?

- The Community Earth System Model (CESM) is a coupled climate model used for simulating the Earth’s climate system.



Programming Model

- About 750,000 lines of Fortran 90
- About 12,000 lines of perl and shell scripts that configure, build, and run the model
- Parallelized with MPI and OpenMP
- Currently supported on
  - Bluefire (NCAR IBM p6)
  - Franklin (NERSC XT4)
  - Hopper2 (NERSC XE6)
  - Intrepid (ANL IBM BG/P)
  - Jaguarpf (ORNL XT5)
  - Edinburgh (NCAR CGD Linux Cluster)

Computational Methods

- Custom PDE Solvers
- Lots of Algebra from the Physics Parameterizations
- Currently lots of work going on developing new grids

Refined Grid

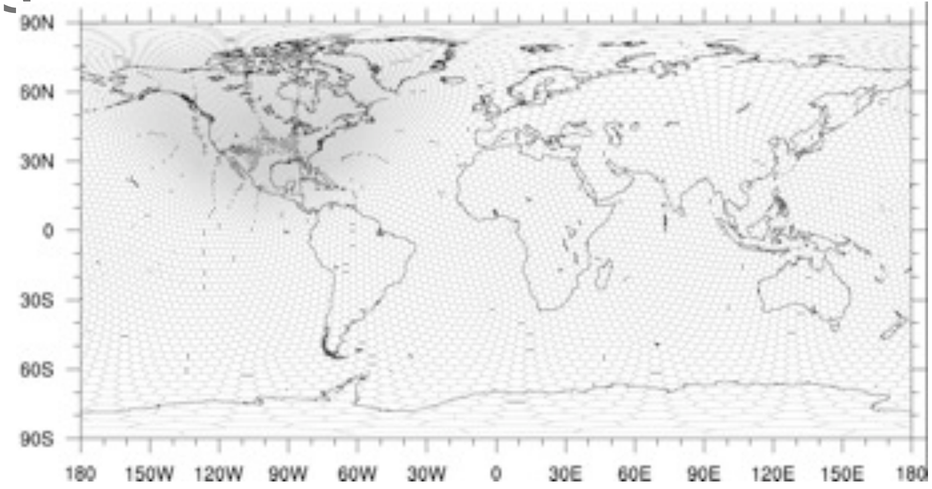


Image by Michael Duda, MMM, NCAR

I/O Patterns and Strategy

- PIO is being used for all components to read/write netcdf data
  - PIO is a wrapper library written by CESM developers
  - Provides one interface for reading/writing using PnetCDF, netCDF, or MPI-IO
  - Communicates between model decompositions and io friendly decompositions
- High-resolution history output size (single monthly average file):

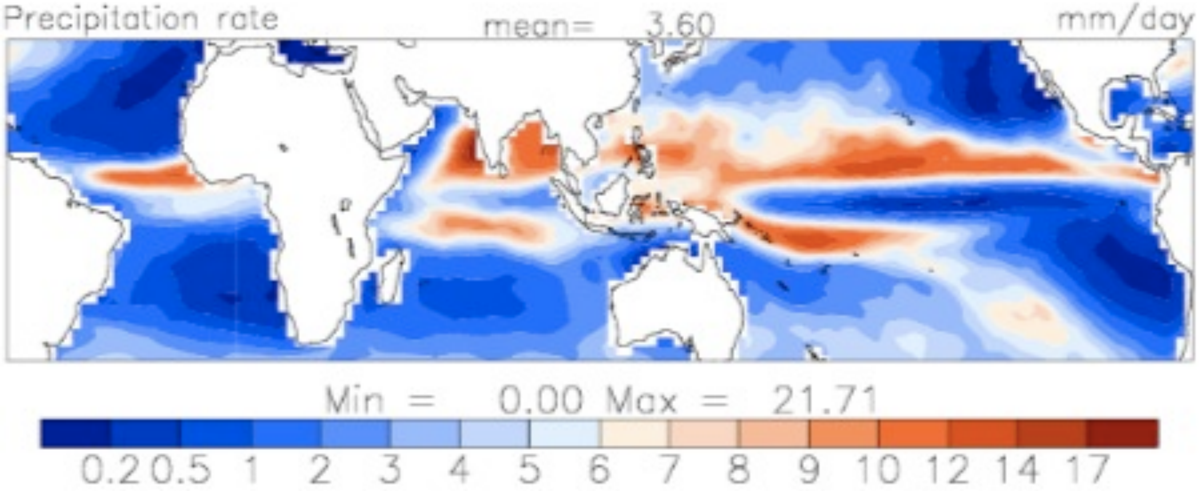
– Atmosphere:	0.8GB
– Ocean:	24GB
– Sea Ice:	4GB
– Land:	0.3GB

- Restart/Checkpoint output:

– Atmosphere:	0.9GB
– Ocean:	29GB
– Sea Ice:	5GB
– Land:	0.2GB

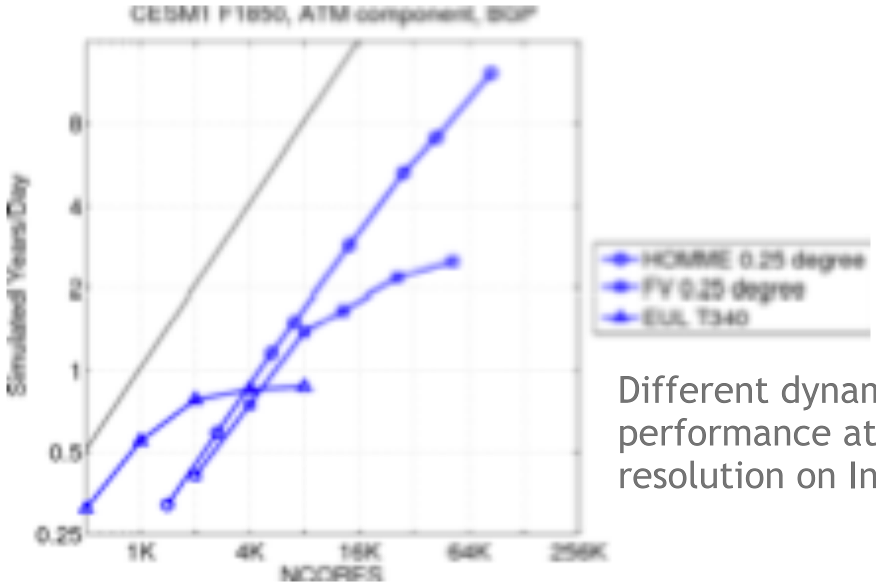
Visualization and Analysis

- All available packages run in serial
- Parallel visualization and analysis tools are being developed under the Parvis project
  - Developing the ParCAL library to analyze data in parallel
  - Using swift to task parallelize workflows



Performance

- Use MPE logging to explore performance
- Current performance bottleneck: Too many variables leading to little cache
- Current scaling bottleneck: Global communication
- Currently working on porting some portions of the code to GPUs and overlaying communication



Different dynamical core performance at equivalent resolution on Intrepid

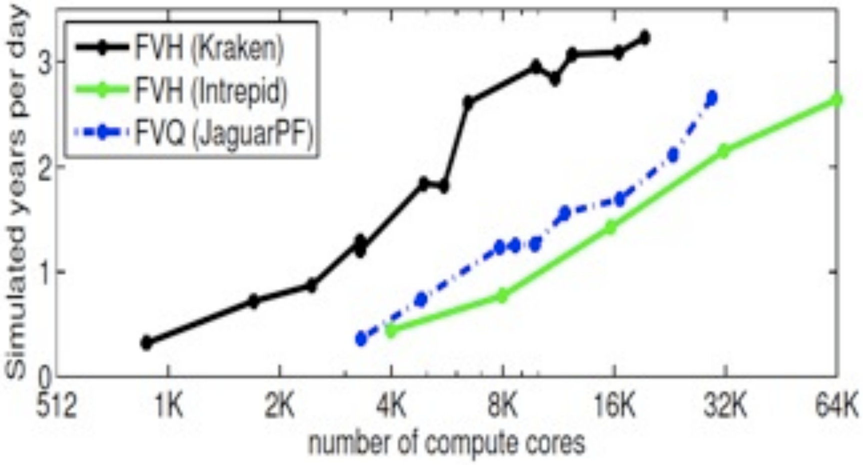
Tools

- Use TotalView and gdb to debug code
- Analysis and visualization tools
  - NCO – several standalone command line programs that take a netcdf input file, operate on it, and outputs the results in a text, binary, or netcdf file
  - NCL – an interpreted language used for data analysis and visualization

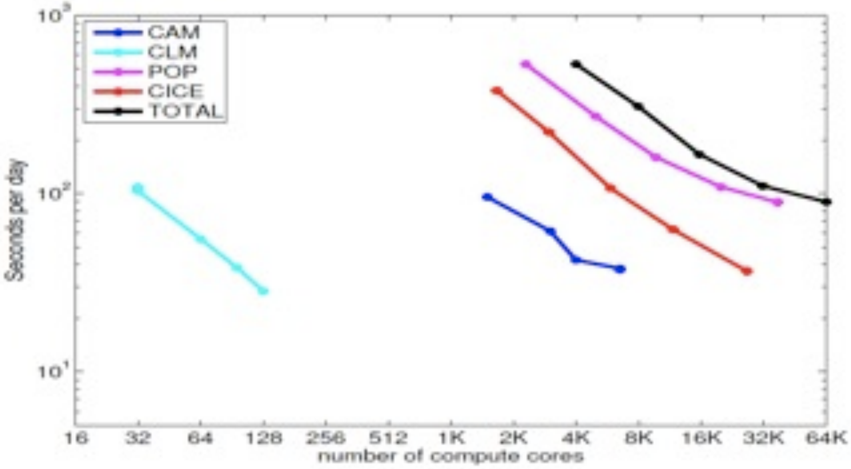
Status and Scalability

- Goal is to be able to simulate 5 years/day at 0.25 degree atmosphere resolution and 0.1 ocean resolution
- We added PIO and replaced algorithms (HOMME for FV) to increase scalability
- Current plans include replacing algorithms in the ocean model to eliminate global communications
- Continuous work on improving component processor configurations
- Pains:
  - Not enough cycles for testing
  - Output data too large for current

Scalability of High-Resolution CESM



Execution time per model component on Intrepid



Roadmap

- Over the next two years we will be heading towards higher resolution and more resolved processes in clouds and carbon cycle
- We hope to be able to make better regional predictions of climate change
- To achieve this we need better scaling algorithms and new algorithms that operate at many spatial scales. We also need algorithms for currently unresolved physical/chemical processes.
- Our plans include a large community effort funded by NSF and DOE