Community Atmosphere Model
CAM

Chris Fischer
National Center for Atmospheric Research
Boulder CO

NCAR is sponsored by the National Science Foundation

A) Project Overview

- **CAM is a global atmospheric climate model. It's also a part of the Community Earth System Model (CESM).**
- **CAM is used to simulate the past, current, and future state of the Earth's climate.**
- **CAM is a community model, so there are several people at universities and research centers that use the model.**
- **The core CAM team consists of software engineers and scientists. There job is to make sure new additions to CAM are of the highest quality.**
- **CAM is sponsored by the National Science Foundation, and the U.S. Department of Energy.**
- **The goal of CAM is to produce the best possible climate simulations**

B) Science Lesson

- **CAM models the physics, chemistry, and dynamics of the atmosphere**
- **There are interactions with the ocean, land, sea ice, and land ice.**
- **You can run with either active ocn, lnd, cice, and cism. Or use data models for those components.**

C) Parallel Programming Model

- CAM supports both MPI and OpenMP, so you can run either in a pure MPI mode, or a Hybrid mode.
- CAM is written in FORTRAN90, requires netcdf and MPI libraries
- CAM also requires perl to run the setup scripts, and Subversion to retrieve inputdata.
- CAM runs on unix systems, from a small linux box up to Jaguarpf (Cray XT5)
- The latest version of CAM (5.1) was released to the public on June 2011.
- Future plans for CAM are to add more chemistry, more physics, better dynamics, and higher resolution runs with refined local grids.

D) Computational Methods

- You have the option to use different dynamical cores. fv, eul, sld, and the newer homme.

- Current plans are centered around supporting the homme dycor. We're using this for high resolution runs up to 1/8°.

- In the future we will be adding new dynamical cores that will allows us to have locally refined grids.

E) I/O Patterns and Strategy

- **We are using PIO (Parallel I/O library) that was developed over several years for CCSM (CESM) I/O. PIO is a parallel interface to netcdf and pnetcdf libraries.**

	Current		Future	
	1° CAM	2° WACCM	1/8° CAM	1° WACCM
Output size	3Gb/yr	30Gb/yr	190Gb/yr	120Gb/yr
Initial condition size	45Mb	500Mb	3Gb	2Gb
Restart size	218Mb	1.2Gb	14Gb	5Gb

F) Visualization and Analysis

- Currently use NCL (NCAR command language) scripts to explore the data.
- Work flow is up to the user
- We've started using swift to parallelize our NCL scripts
- Plan to use Parvis (Parallel Analysis Tools and New Visualization Techniques for Ultra -Large Climate Data Sets) in the future

G) Performance

- Currently not using any tools to measure performance, instead rely on timing calls in the code.
- The bottle neck for better performance and scaling is older dynamical cores such as fv.
- The most important features of a performance tool is ease of use, and documentation
- Currently we are adding support for the homme dycore which should give us better scaling on larger processor counts. There are also other dynamical cores that are being developed.

H) Tools

- **Use TotalView, GNU debug, and the good ole print statement for debugging.**
- **No plans to use other tools.**

I) Status and Scalability

- **CAM scales very well. We've seen good scaling of 1/8° homme for up to 130k cores on jaguarpf.**
- **Next year we should be using a new dynamical cores with locally refined meshes that will hopefully scale well to 200k+ cores.**
- **My pains**
  - **Large system downtime**
  - **Analyzing the output**
  - **Large system software updates**
  - **Wait times on queues**
  - **Latency with interactive login nodes**
- **Switching from a fv to homme dynamical core seems to have given us good scalability**

J) Roadmap

- Over the next 2 years, CAM will have more physics, more chemistry, better dynamics, at higher resolutions, with more vertical layers going higher up in the atmosphere.
- This should give us a better understanding to the atmosphere
- What we need to do, is to take the scientists code, and merge it into CAM in such a way as to maintain scalability