Climate Modeling at NCAR: Performance Roadblocks

Tony Craig
National Center for Atmospheric Research (NCAR)

The Conference on High Speed Computing, Salishan
April 22-25, 2002

OUTLINE

- Overview of Climate Modeling at NCAR
  - History
  - Production
  - Performance
- Individual component performance
- Coupling, load balance, coupled performance
- Challenges in the environment
- Summary and Future

CCSM Overview

- CCSM = Community Climate System Model (NCAR)
- Designed to evaluate and understand earth's global climate, both historical and future.
- Multiple executables (5)
  - Atmosphere (CAM), MPI/OpenMP
  - Ocean (POP), MPI
  - Land (CLM2), MPI/OpenMP
  - Sea Ice (CSIM4), MPI
  - Coupler (CPL5), OpenMP

CCSM Platforms

- Currently support
  - IBM Power3, Power4 - NCAR, NERSC, ORNL
  - SGI Origin - NCAR, LANL
  - CPQ - ORNL
- Future?
  - Linux
  - Vector Platform (NEC/Earth Simulator)

History at NCAR

- 1990 - CRAY, CRAY, CRAY
- ~~1996 - NEC~~
- 1997 - SGI O2K

CSM/PCM

- 1999 - IBM SP (power3)
- 2002 - IBM SP (power4)

CCSM

CCSM Performance Summary

- T42 resolution atm and land, 26 vertical levels in atm (128x64x26, 200k cells)
- 1 degree resolution ocean and ice, 40 vertical levels in ocean (320x384x40, 4M cells)
- On 144 processors of IBM power3

  (16 processors/node, 375 Mhz clock, Colony):
  - Model runs about 5.5 simulated years / day
  - Requires about a month to run 150 years
  - Requires O(months) to run a CASE
- Get about 10% of peak FLOPS and getting worse

CCSM Overview (part 2)

- F90
- 500k lines of code
- Community project, dozens of developers
- Code releases
- Multiple levels or parallelism
- Netcdf history files
- Binary restart files
- SCIDAC - DOE
- ESMF - NASA
- Good science top priority

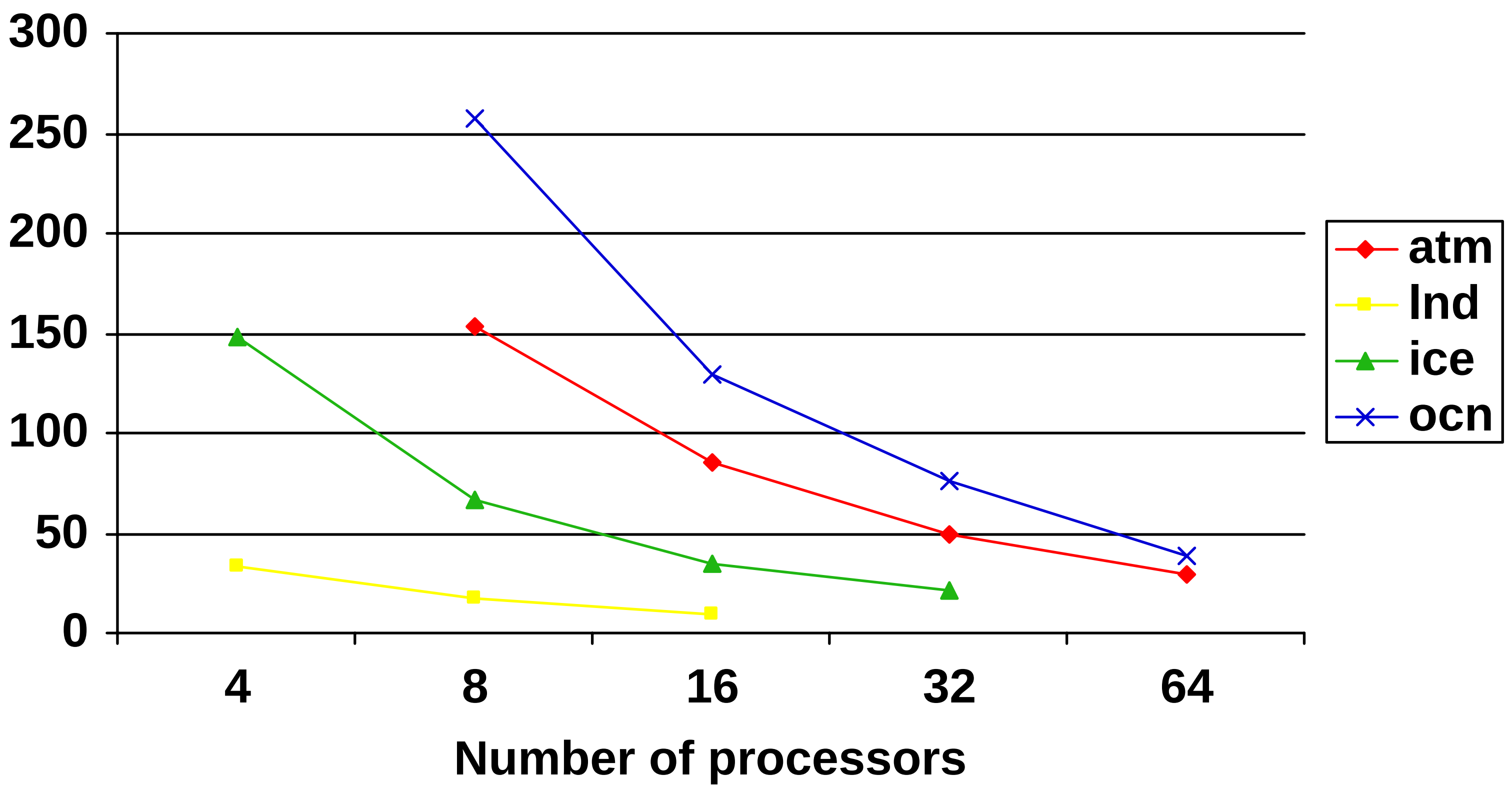
CCSM Production

- Integrate forward in time
- Finite difference and spectral, explicit and implicit methods, vertical physics, global sums
- I/O not a bottleneck (5 Gb / simulated year)
- Restart capability (750 Mb)
- Separate harvesting to mass store
- Transfer subset of files back to NCAR
  - (2 Gb / simulated year)
- Auto resubmit

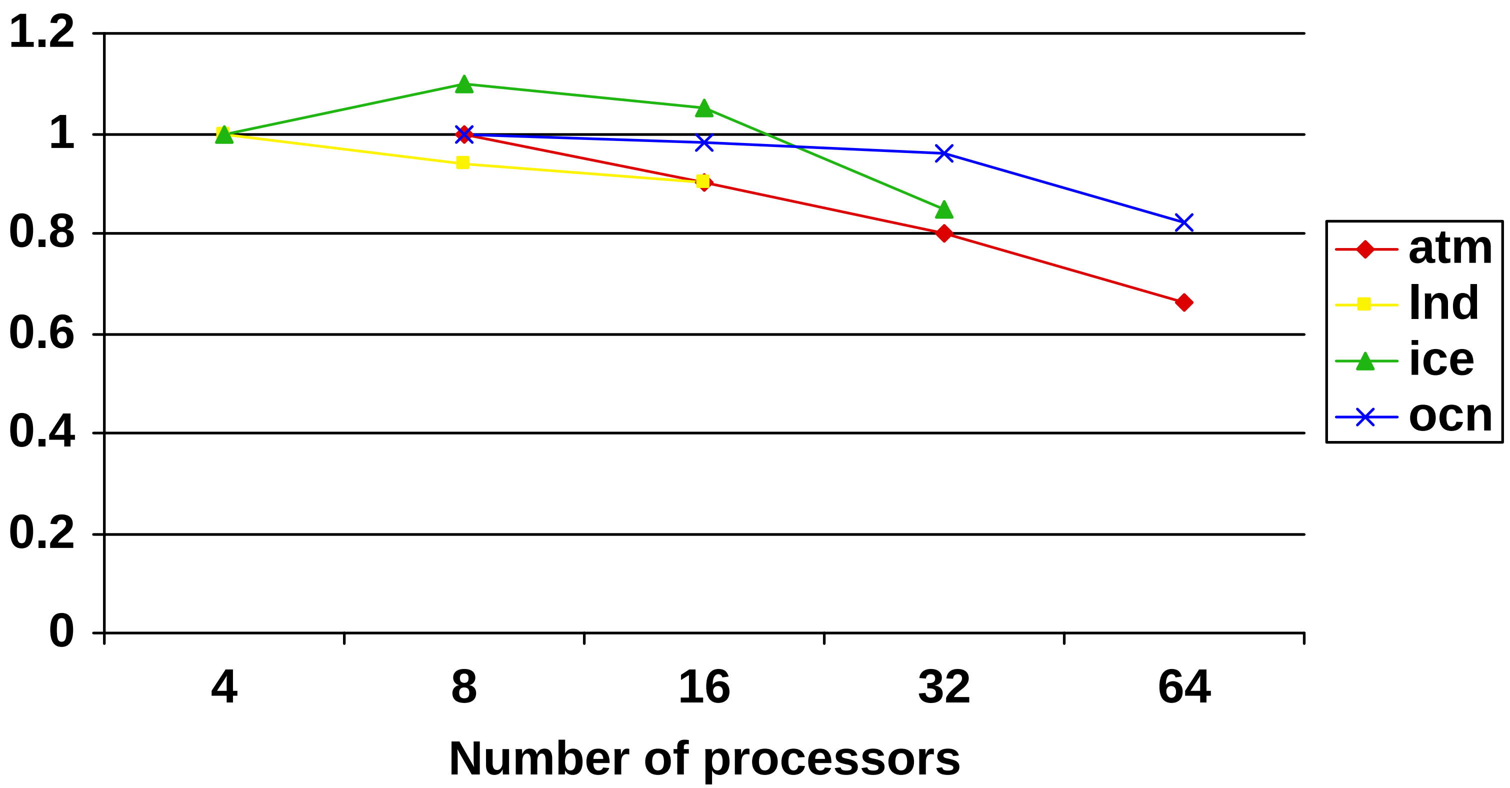
Role of CCSM SE Group

- Model development (science)
- Testing
- Infrastructure (cvs, bug tracking, web)
- Modern SE practices
- Performance
- Portability and hardware issues
- Modern coding practices (F90+)
- RISC vs Vector

Component Timings

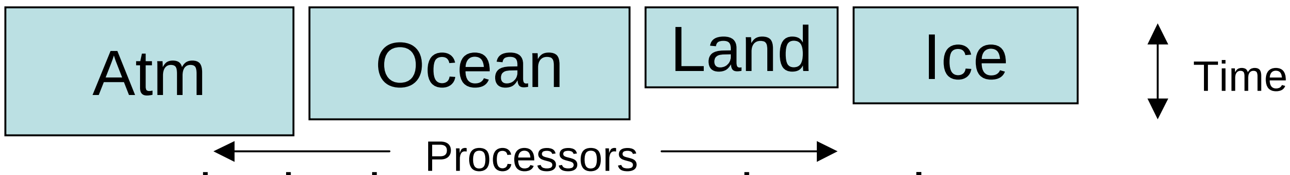


Component Scaling



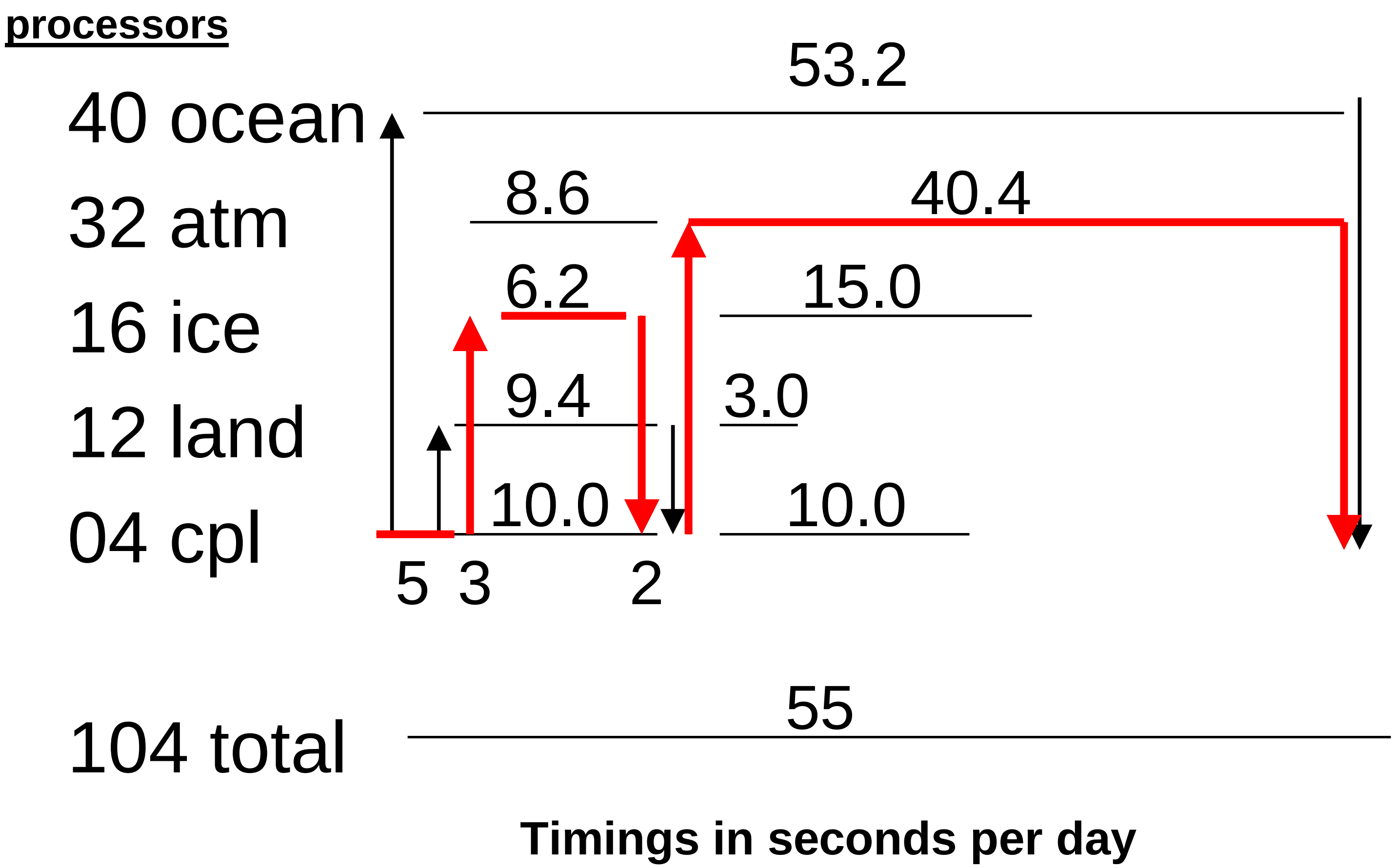
Coupling

- Multiple executables, running concurrently on unique processors sets.



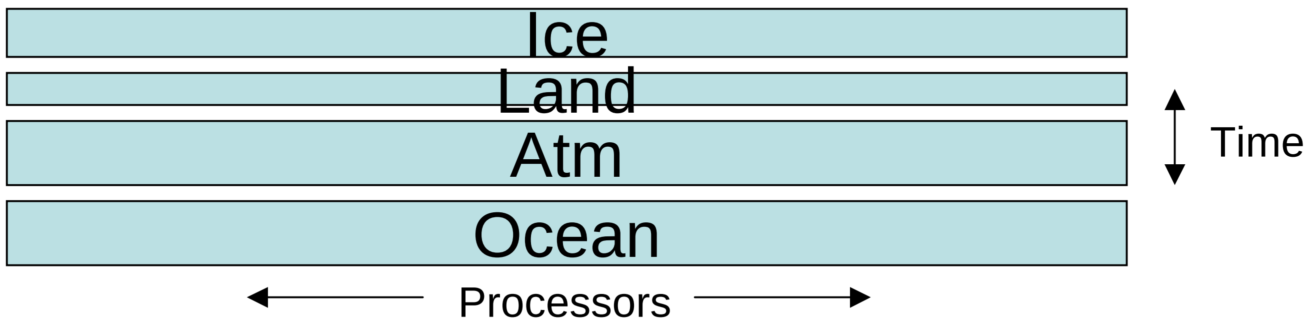
- All communication is root-to-root, via coupler.
- Coupling frequency is 1 hour for atm, land, and ice models, 1 day for ocean.
- 1 send and 1 receive for each component during each coupling interval.
- Scientific requirement that the land and ice models compute fluxes for the atmosphere “first”.

CCSM Load Balancing



Other Coupling Options: single executable, sequential

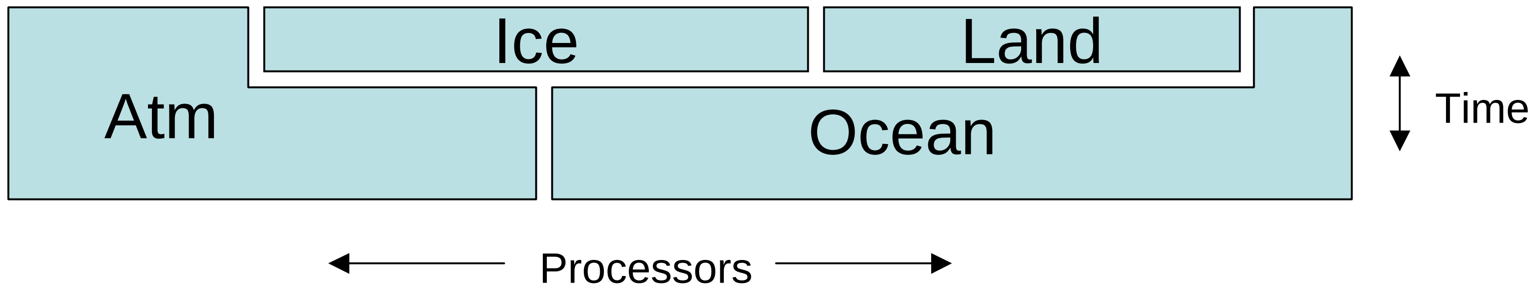
- Single executable, components running sequentially on all processors



  - No idle time
  - Components must scale well to large number of processors

Other Coupling Options: combined sequential/concurrent on overlapping processors

- Single or multiple executable, components running sequentially and/or concurrently on overlapping processor sets.



  - Optimal compromise between scaling and load balance.
  - Requires sophisticated coupler and improvements to hardware and machine software capabilities.
  - CRAY OS/batch did this, now application SW must

Challenges in the Environment

- Batch environment; startup and control of multiple executables
- Tools
  - Debuggers (totalview) inadequate; multiple executables, MPI/OpenMP parallel models, user defined data structures are a problem.
  - Timing and profiling tools inadequate (vampir, jumpshot, etc)
  - Staying away from instrumenting code (risk, robust, #if)
  - Compilers and libraries can be slow, buggy, and constantly changing. Compiler option conflicts.
  - Using print statements and calls to system clock

Challenges in the Environment

- Machines not well balance
  - chip speed
  - interconnect
  - memory access
  - cache
  - I/O
- Each machine is “balanced” differently
- Need to develop “flexible” software strategies

RISC vs Vector

- Data layout; index order
- Data structures; vector of points versus arrays of fields
- Minimize floating operations (RISC, ifs) versus pipelining (Vector, masking)
- Loop ordering and loop structure
- Vectorization impacts parallelization
- Chunking is a compromise
- Multiple functional units on RISC machines
- Memory access, cache

Summary

- Science top priority, community project, model releases
- Machines change rapidly
- Won’t be running on 1000s of pes anytime soon
- Component scaling and CCSM load balance acceptable
- Does hybrid parallelism work?
- Tools unusable

Future

- Increased coupling flexibility
- Continue to work on scalar and parallel performance in all codes
- Look forward instead of just responding to hardware changes
- Take advantage of libraries/collaborations for performance portability - PETSc, ESMF, SCIDAC
- Find some useful tools
- Come up with a strategy for dealing with RISC and Vector architectures

Climate Modeling at NCAR: Performance Roadblocks

Tony Craig

National Center for Atmospheric Research (NCAR)

THE END