Preparing for Petascale through Expedition Computing

John M. Dennis: dennis@ucar.edu

Mariana Vertenstein: mvertens@ucar.edu

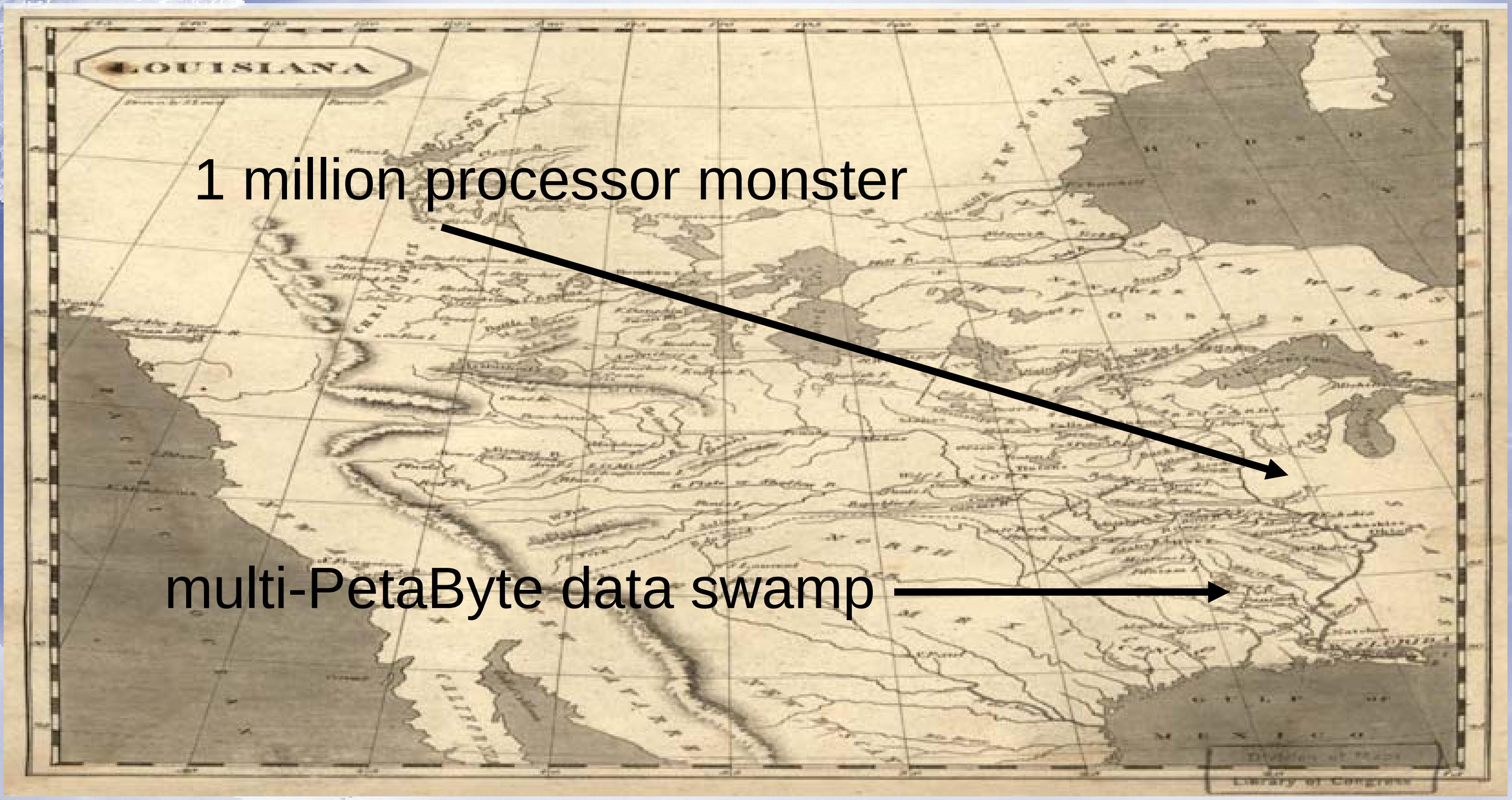
October 16, 2007

U.S. History

- 1803
- Louisiana Purchase

- 1804-1806
- Lewis and Clark Expedition
  - Mapped the west
  - Cataloged 122 species of animals

Petascale Land



Expeditions into the Frontier

- Expeditions are dangerous
  - Eaten by wild animals
  - Deadly diseases
  - Shot full of arrows
  - Arrested
  - Could end in disaster
- Need to explore landscape
  - Map the frontier
  - How do we evolve our code?
  - What science does Petascale enable?

Explore Petascale Land today!

- Increasing common access to large systems
  - LLNL Appro AMD: 9K processors [today]
  - TJ Watson IBM Blue Gene/L: 40K processors [today]
  - ORNL Cray XT3/4 :
    - 22K processors [today]
    - 44K processors [Jan 2008]
  - BNL/SUNY IBM Blue Gene/L: 38K processors [today]
  - NERSC Cray XT4: 19K processors [today]
  - TACC Sun: 55K processors [Jan 2008]
  - ANL IBM Blue Gene/P:
    - 32K processors [Jan 2008]
    - 160K processors [Fall 2008]

Funding Sources

- Department of Energy: CCPP Program Grants
  - DOE-BER
    - DE-FC03-97ER62402
    - DE-PS02-07ER07-06
    - DE-FC02-07ER64340
    - B&R KP1206000
  - DOE-ASCR
    - B&R KJ0101030
- National Science Foundation:
  - Cooperative Grant NSF01

Outline:

- Motivation
- Current Expeditions:
  - POP
  - CICE
  - CLM
  - CPL
  - CCSM
- Requirement: Parallel I/O library (PIO)
- Conclusions

Expedition: POP

POP: Parallel Ocean Program

- Developed at LANL
- Two components:
  - Baroclinic: Finite difference
  - Barotropic: Solve surface pressure (2D) with PCG w/ Diagonal preconditioning
- Preparing POP for Petascale
  - Rework of Conjugate Gradient solver
    - Addition of 1D data-structures [Done] (6 files)
      - Reduces data loaded from memory
      - Reduces data passed between processors
    - Improve pre-conditioner [Underway]
  - Message aggregation for 3D variables [Done] (1 file)
  - Alternative load-balancing algorithm
    - Space-filling curves [Done] (2 files)
  - Parallel I/O [Underway]

POP: Parallel Ocean Program (con't)

- POP @ 0.1°
  - Global eddy-resolving
  - Computational grid: [3600 x 2400 x 40]
  - Land creates problems: Load-imbalance, scalability
  - Evaluate using benchmark:
    - 1 day/ Internal grid / 7 minute timestep

Status of POP

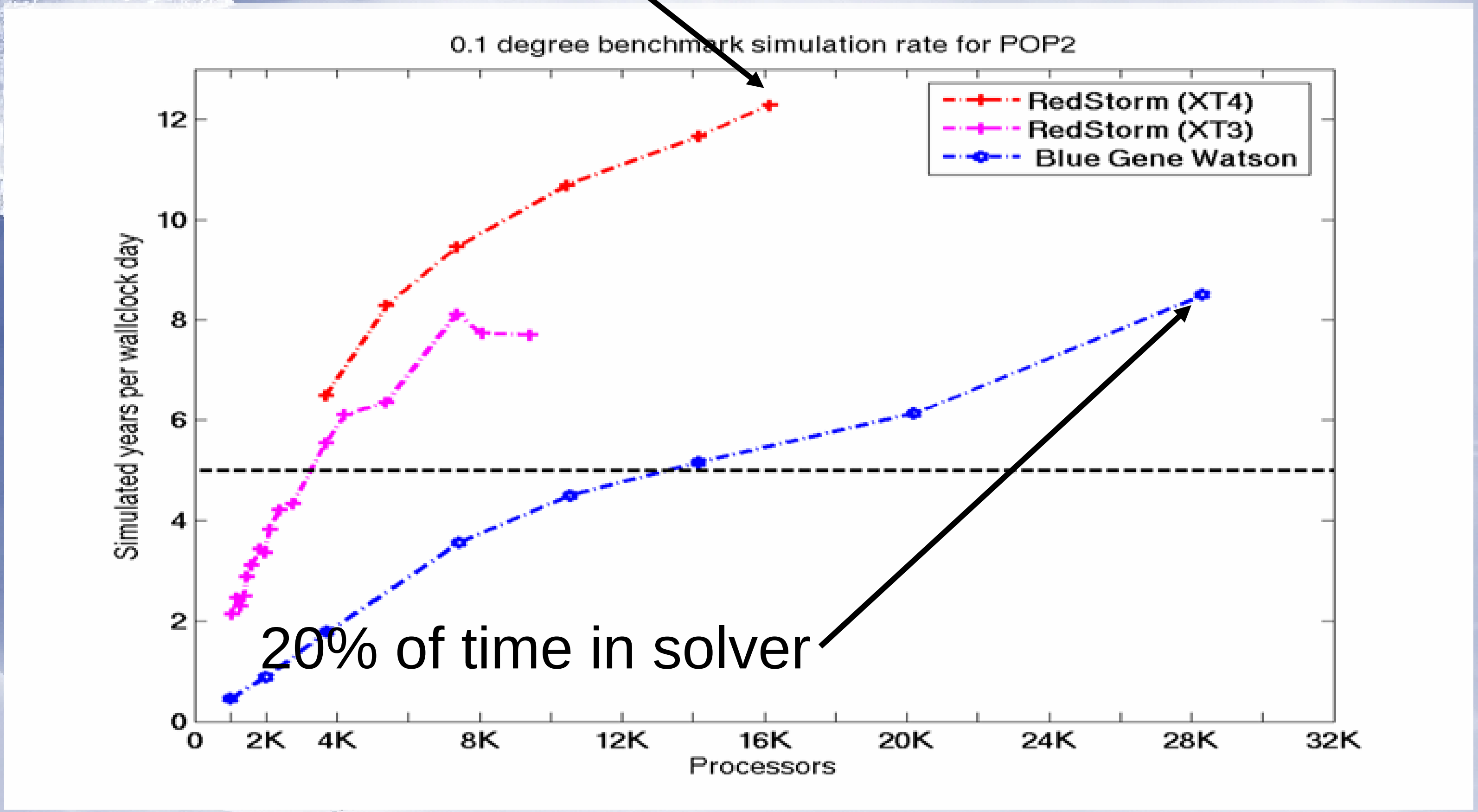
- POP2 benchmark
  - 17K Cray XT4 processors [12.5 years/day]
    - 70% of time in solver
    - Does not include MPI_reduce fixes [P. Worley]
  - 29K IBM Blue Gene/L [8.5 years/day]
- Won BGW cycle allocation

Eddy Stirring: The Missing Ingredient in Nailing Down Ocean Tracer Transport
[J. Dennis, F. Bryan, B. Fox-Kemper, M. Maltrud, J. McClean, S. Peacock]

  - 110 Rack Days/ 5.4M CPU hours
  - 20 year 0.1° POP simulation
    - Includes a suite of dye-like tracers
    - Simulate eddy diffusivity tensor

POP2 0.1° benchmark

71% of time in solver



20% of time in solver

Courtesy of M. Taylor

Lessons from POP

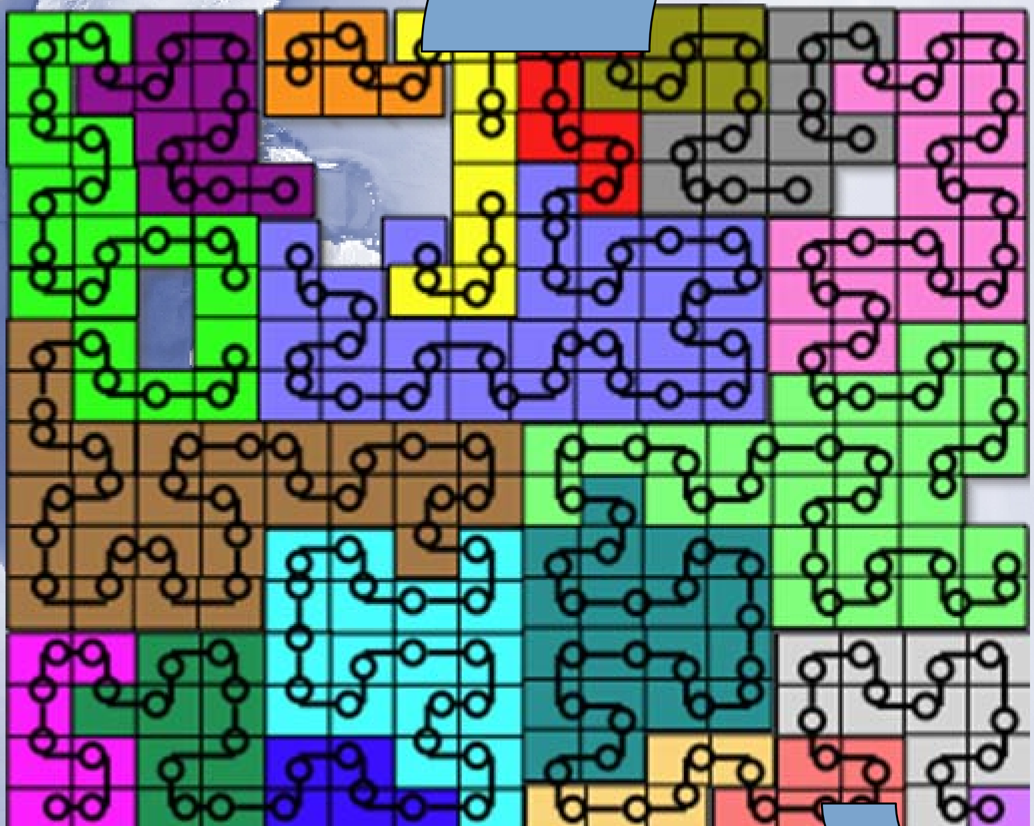
- Utilize very large processor counts
- “Big science” possible in 256 Mbytes/proc
- Status:
  - Completed 9.2 years of spinup [7600 processors]
- Technique reuse
  - Space-filling curves (SFC)
  - Parallel I/O (PIO)
- Arrows:
  - Need Parallel I/O to write history file on BG/L
  - Rediscovering code bugs
  - Huge development/debugging platform

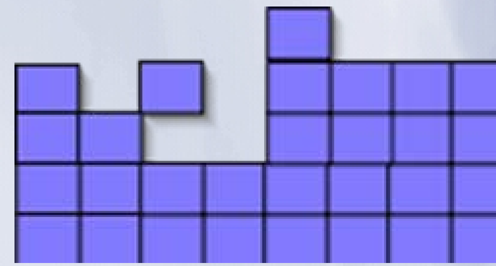
Expedition: CICE

CICE4

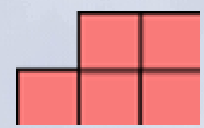
- Developed at LANL (current CCSM3.5 sea-ice model)
- Shares grid and infrastructure with POP
  - Reuse techniques from POP work
- Computational load-imbalance for CICE4 creates challenges:
  - ~15% of grid has sea-ice
  - Use *weighted* Space-filling curves?
- Evaluate CICE4 @ 0.1° (computational grid [3600 x 2400 x 20]) using benchmark:
  - 1 day/ Initial run / 30 minute timestep/ no Forcing
  - 10K Cray XT3 processors
  - 40K Blue Gene/L processors
- Current work
  - Implementation of OpenMP directives
  - Implementation of parallel I/O

CICE4 @ 1° on 20 processors



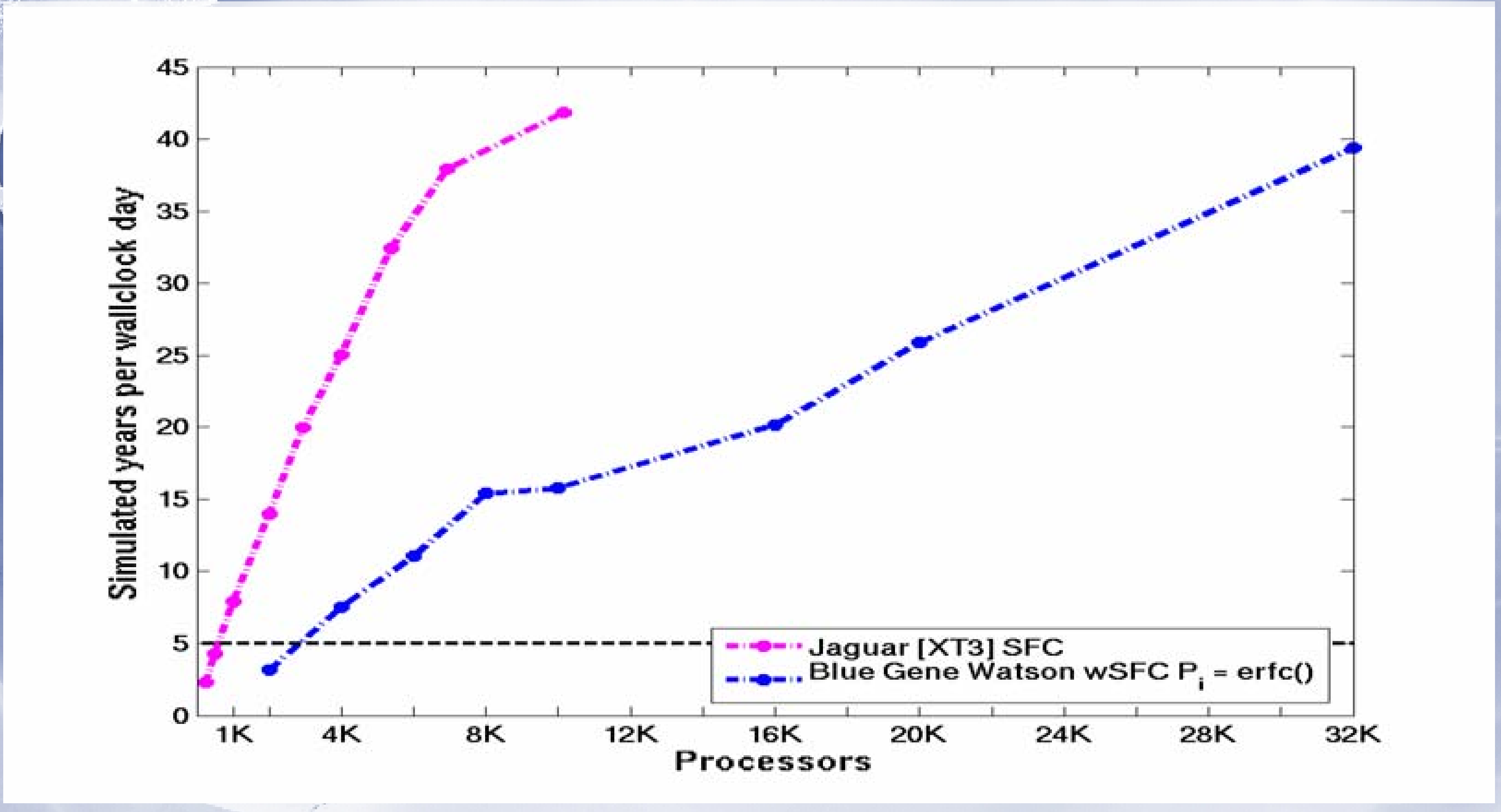


Large domains @ low latitudes



Small domains @ high latitudes

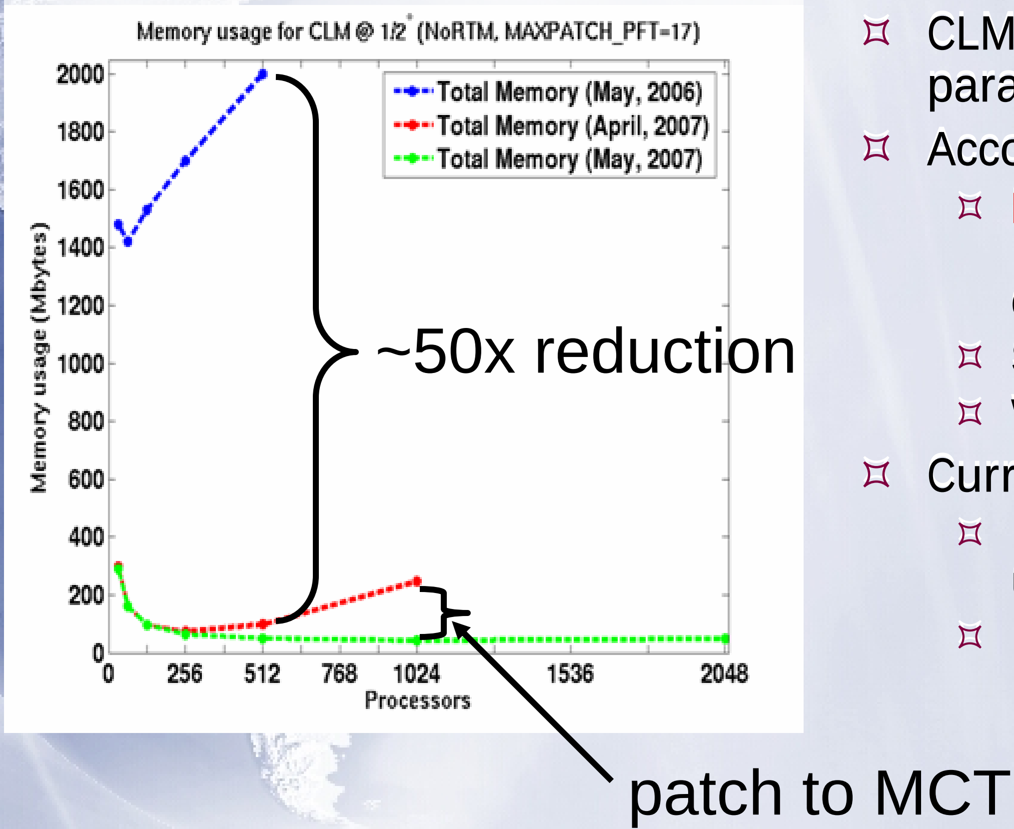
CICE4 @ 0.1°



Simulated years per wallclock day
Processors
0
5
10
15
20
25
30
35
40
45
1K
4K
8K
12K
16K
20K
24K
28K
32K
Jaguar [XT3] SFC
Blue Gene Watson wSFC P_i = erfc()

Expedition: CLM

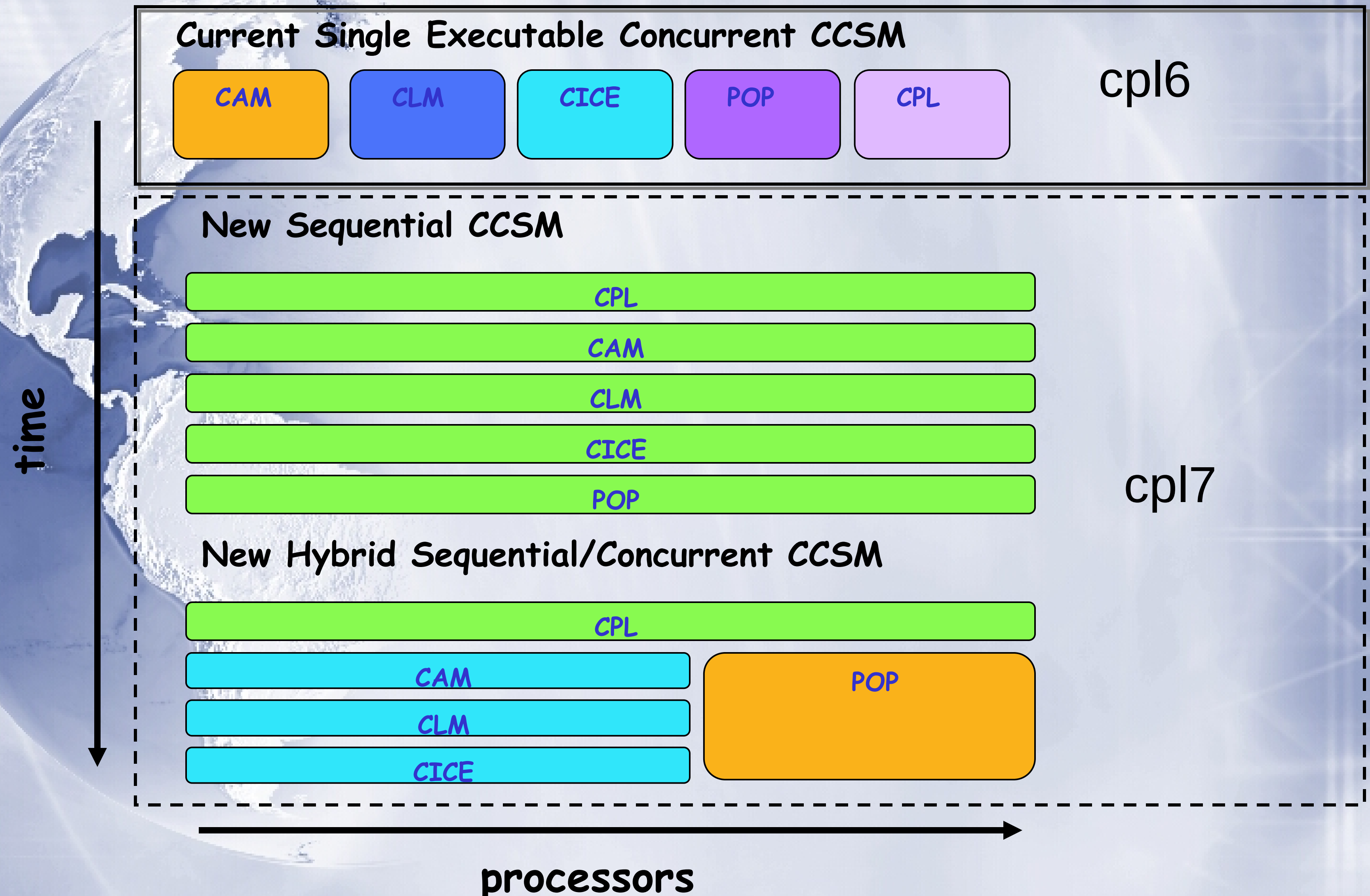
Status of CLM



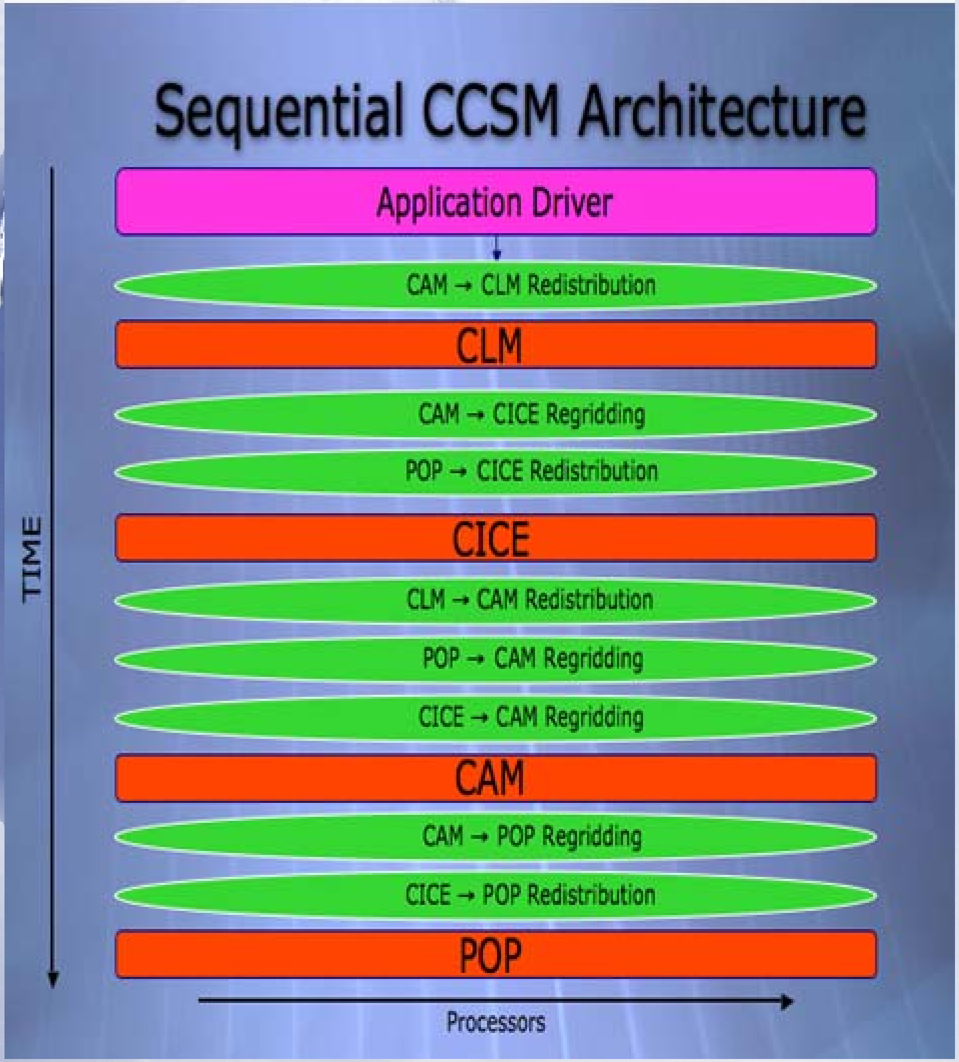
- CLM is inherently embarassingly parallel
- Accomplished (in CCSM3.5)
  - **Elimination of global memory** and reworking of decomposition algorithms
  - Separation of CAM/CLM grids
  - Work of Tony Craig
- Current Work
  - Implementation of parallel I/O using PIO
  - Investigation of scalability at 1/6° & 1/10°

Expedition: CPL7

CPL6 -> CPL7


Current Single Executable Concurrent CCSM
CAM
CLM
CICE
POP
CPL
cpl6
New Sequential CCSM
CPL
CAM
CLM
CICE
POP
cpl7
New Hybrid Sequential/Concurrent CCSM
CPL
CAM
CLM
CICE
POP
time
processors

CPL7: Sequential Coupler



- Simple elegant design
- Eliminates two stage communication
- Possible quasi-local communication
- Places premium on
  - Component scalability
  - Memory usage
- Work of M. Vertenstein, R. Jacob and T. Craig

Advantages of CPL7 System

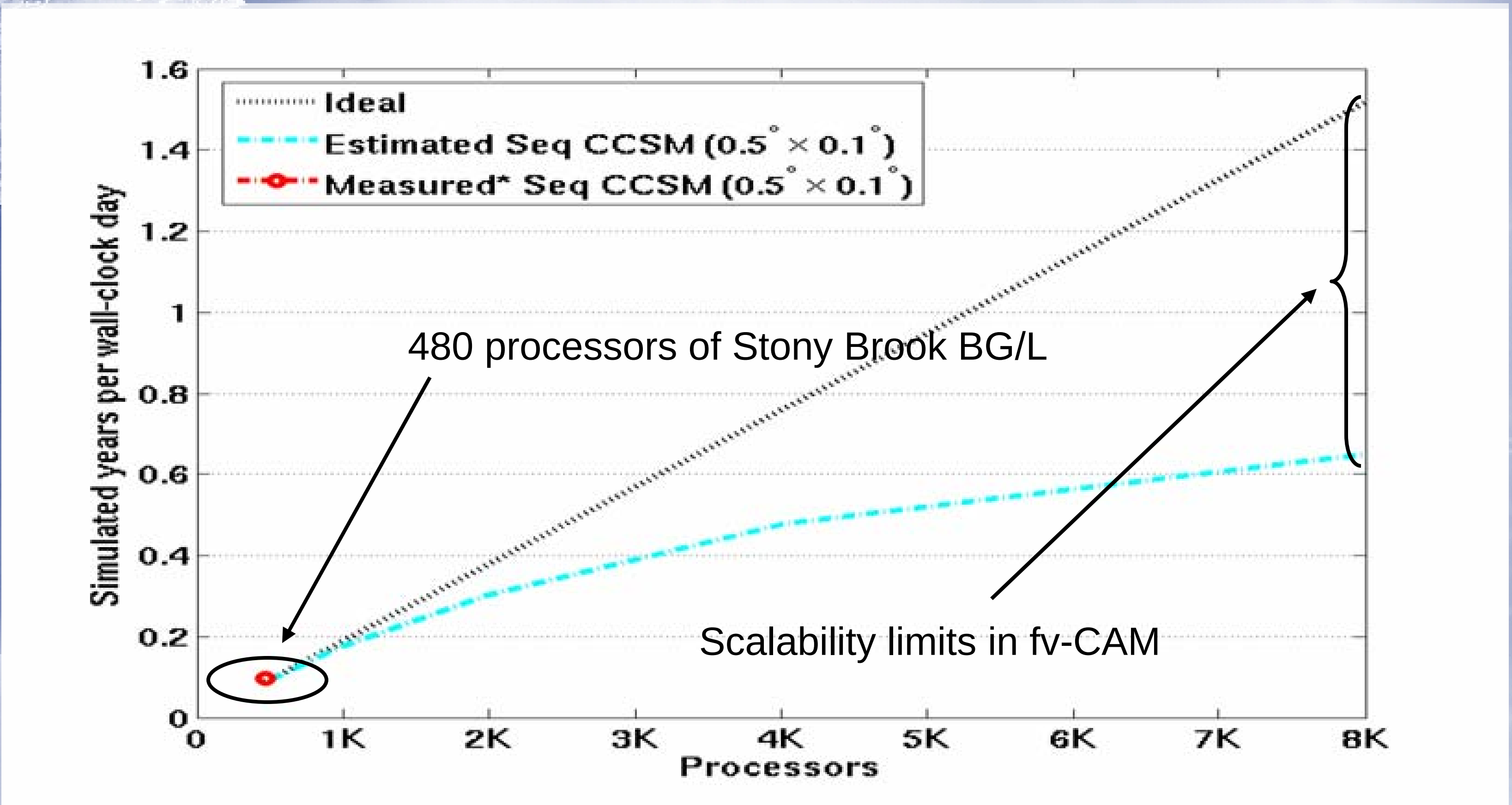
- **Simplicity:**
  - Much more simple to load balance, debug, port and support
- **Efficiency:**
  - More efficient system - eliminate the load imbalance inherent in the concurrent CCSM execution
- **Flexibility:**
  - Permit greater flexibility to construct the system we want to run for a given resolution, architecture and new scientific requirements
- **Code Reuse and Maintainability:**
  - Eliminates the need for separate stand-alone component code base
    - Science done in stand-alone components will no longer be different than science done in full CCSM
- **Standardization:**
  - Standardize coupling interfaces
    - Different coupling frameworks (e.g. ESMF, MCT) can be incorporated and compared without touching core component code

Expedition: CCSM

Target Configuration

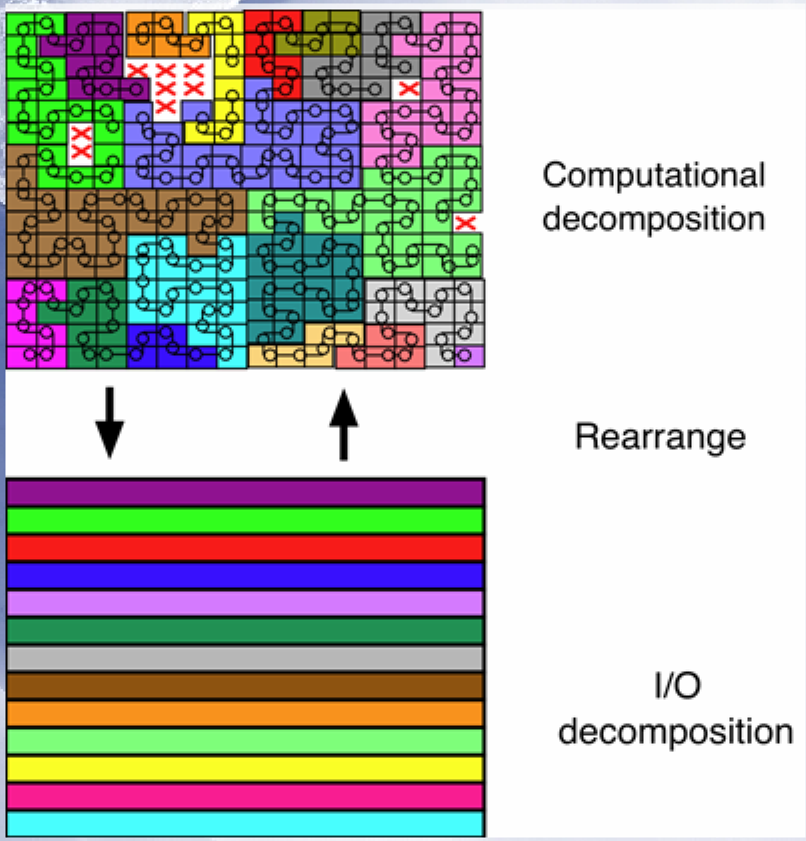
- Multiple 50 year runs [1980-2030]
- Configuration:
  - FV-CAM (0.47° x 0.63°, L26 or L31)
  - CLM (0.47° x 0.63°)
  - POP @ 0.1°
  - CICE4 @ 0.1°
- Scalability of FV-CAM [Mirin & Worley 2007]
  - Enable execution of physics on more processors than dynamics
  - Improvements in performance of MPI reduction
  - Reductions in memory usage now enables execution of FV-CAM @ 0.47° x 0.63° on BGL
- Target: 4K - 32K BG/P processors

Simulation rate (Blue Gene)



Parallel I/O (PIO)

PIO: Parallel I/O library



- Work of J. Dennis, R. Loy, & J. Edwards
- All component models need parallel I/O
  - Serial I/O blows memory
- Supports: binary & netCDF
- Flexibility to adapt to I/O system
- Needed for POP BGW runs
- Rearrangement:
  - MCT
  - ESMF [work in progress]
- Critical for high {resolution,processor counts}

Status of PIO

- Prototype added to POP2
  - Reads restart and forcing files correctly
  - Writes binary restart files correctly
- Prototype implementation in HOMME,CAM [J. Edwards], CLM[Craig]
  - Writes netCDF history files correctly
  - Ongoing work to optimize performance/memory usage
- POPIO benchmark
  - 2D array [3600x2400] (70 Mbyte)
  - Test code for correctness and performance
  - Tested on 30K BGL processors in Oct 06
- Performance
  - POWER5: 2-3x serial I/O approach [GPFS]
  - BGL: 8x serial I/O approach [GPFS, PVFS2]
  - Positive preliminary results on Lustre

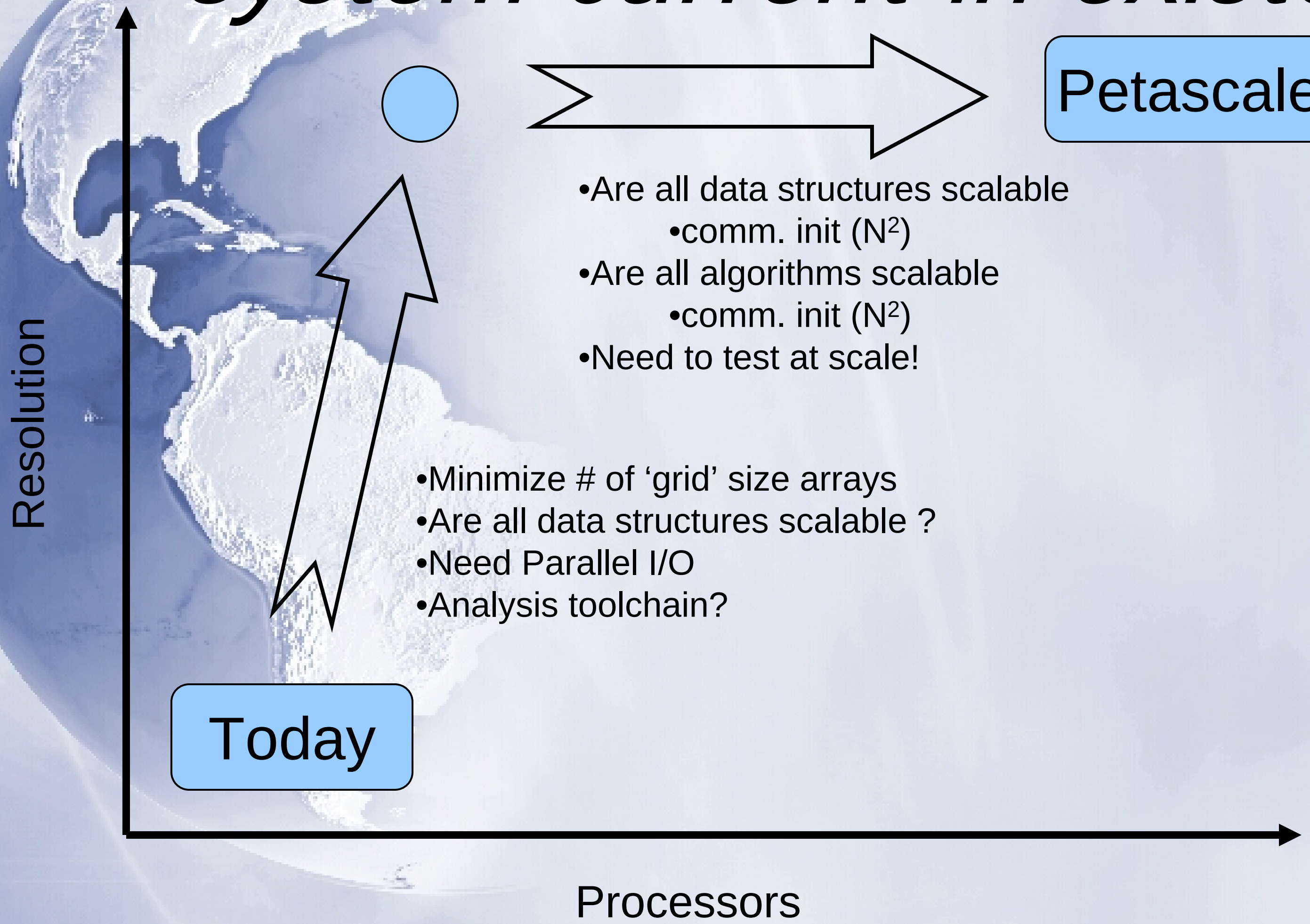
Preparing for Petascale

Preparing for Petascale

- Additional step in software testing
  - Stress testing of components
    - High resolution
    - Large processor counts
- Can you utilize largest machine in existence today?

Why not?

Why can't you utilize largest system current in existence?



- •Are all data structures scalable
  - •comm. init (N^2)
- •Are all algorithms scalable
  - •comm. init (N^2)
- •Need to test at scale!

- •Minimize # of 'grid' size arrays
- •Are all data structures scalable ?
- •Need Parallel I/O
- •Analysis toolchain?

Conclusions

- Petascale CCSM:
  - 2 of 5 components are ready
    - POP: Cray XT4 17K processors [12.5 years/day]
    - CICE: Cray XT3 10K processors [42 years/day]
  - 3 of 5 components are very close:
    - CLM: IBM Blue Gene 2K processors [~70 years/day]
    - CPL7: IBM Blue Gene 7K processors [~40 years/day]
    - CAM: IBM Blue Gene 1K processors [1.4 years/day]
  - Demonstrated on Blue Gene
    - 480 processors of BNL/SUNY machine
  - Common issues for all component models:
    - Parallel I/O
    - Watch memory usage
- Path to Petascale computing:
  1. Test the limits of our codes
  2. Fix resulting problems
  3. Goto 1.

NCAR

Acknowledgements/Questions?

- Thanks to:
  - D. Bailey (NCAR)
  - F. Bryan (NCAR)
  - T. Craig (NCAR)
  - J. Edwards (IBM)
  - B. Fox-Kemper (MIT,CU)
  - E. Hunke (LANL)
  - B. Kadlec (CU)
  - D. Ivanova (LLNL)
  - E. Jedlicka (ANL)
  - E. Jessup (CU)
  - R. Jacob (ANL)
  - P. Jones (LANL)
  - S. Peacock (NCAR)
  - K. Lindsay (NCAR)
  - W. Lipscomb (LANL)
  - R. Loy (ANL)
  - A. Mirin (LLNL)
  - M. Maltrud (LANL)
  - J. McClean (LLNL)
  - T. Qian (NCAR)
  - M. Taylor (SNL)
  - H. Tufo (NCAR)
  - P. Worley (ORNL)
- Funding:
  - DOE-BER CCPP Program Grant
    - DE-FC03-97ER62402
    - DE-PS02-07ER07-06
    - DE-FC02-07ER64340
    - B&R KP1206000
  - DOE-ASCR
    - B&R KJ0101030
  - National Science Foundation Cooperative Grant NSF01
- Computer Time:
  - Blue Gene/L time:
    - NSF MRI Grant
      - NCAR
      - University of Colorado
      - IBM (SUR) program
    - BGW Consortium Days
      - IBM research (Watson)
    - LLNL
  - CRAY XT3/4 time:
    - ORNL
    - Sandia