SciDAC
Scientific Discovery through Advanced Computing

Office of Science
U.S. DEPARTMENT OF ENERGY

Performance Engineering for the Next Generation Community Climate System Model

Progress and Plans

Patrick H. Worley
Oak Ridge National Laboratory

June 9, 2008

Office of Advanced Scientific Computing Research
Office of Science, U.S. Department of Energy
Germantown, MD

Argonne NATIONAL LABORATORY

OAK RIDGE National Laboratory

Acknowledgements

- The work described in this presentation was sponsored by the Atmospheric and Climate Research Division and by the Office of Mathematical, Information, and Computational Sciences, both of the Office of Science, U.S. Department of Energy, under Contract No. DE-AC05-00OR22725 with UT-Battelle, LLC, Contract No. DE-AC52-07NA27344 with Lawrence Livermore National Security, LLC, and Contract No. DE-AC02-06CH11357 with UChicago Argonne, LLC.
- These slides have been authored by contractors of the U.S. Government under contracts No. DE-AC05-00OR22725, DE-AC52-07NA27344, and DE-AC02-06CH11357, and are released as LLNL Report LLNL-PRES-404413. Accordingly, the U.S. Government retains a nonexclusive, royalty-free license to publish or reproduce the published form of this contribution, or allows others to do so, for U.S. Government purposes.
- This work used resources of the National Center for Computational Sciences at Oak Ridge National Laboratory, which is supported by the Office of Science of the Department of Energy under Contract DE-AC05-00OR22725.

Background

- Community Climate System Model (CCSM)
  - Fully-coupled, global climate model that provides state-of-the-art computer simulations of the Earth's past, present, and future climate states
  - Comprised of a coupler and four component models: atmosphere, ocean, land, and sea ice
  - Developed at the National Center for Atmospheric Research (NCAR), with contributions from external research groups funded by the National Science Foundation, Department of Energy (DOE) and National Aeronautics and Space Administration (NASA)
- SciDAC-2 science application project *A Scalable and Extensible Earth System Model for Climate Change Science (SEESM)* is working to transform the CCSM into an earth system model that fully simulates the coupling between the physical, chemical, and biogeochemical processes in the climate system.
- SciDAC-2 science application partnership project *Performance Engineering for the Next Generation Community Climate System Model (PENG)* is working with SEESM on the long-term performance engineering of the CCSM.

PENG Organization

Oak Ridge National Laboratory:

- Pat Worley (PI):

Lawrence Livermore National Laboratory:

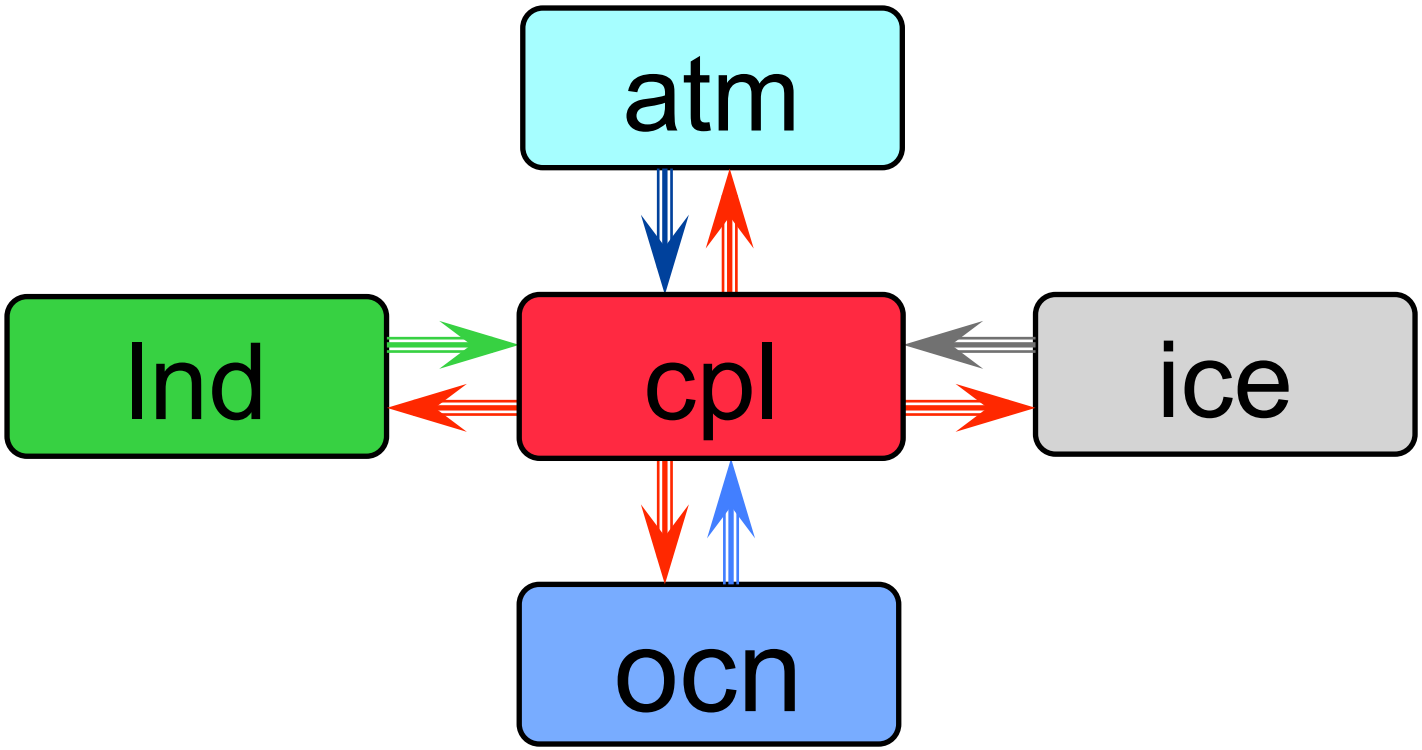
- Art Mirin:

Argonne National Laboratory:

- Jay Larson:
- Ray Loy:

We also collaborate closely with SEESM colleagues, with researchers and software engineers at NCAR, NASA, NOAA, and with others associated With the CCSM developer community.

CCSM Hub and Spoke



- Logical view of coupling architecture
- Cartoon of parallel implementation in concurrent mode

(slide courtesy of CSEG/NCAR)

PENG Activities

1. Porting to new architectures:
   - BG/L and BG/P
   - quad-core Cray XT

   and examining performance at next generation scales (problem size, problem science, processor counts):
   - Current problem size for the atmosphere is a 2 degree horizontal grid (96x144) and 26 vertical levels. Resolutions under consideration include .5, .25, and .125 degree horizontal grids and 30, 48, and 66 vertical levels.
   - New benchmark suite has recently been specified that looks at cost of new physics parameterizations for cloud microphysics, for radiation, for atmospheric chemistry, among others.
   - Running experiments out to 4K processors for current problem sizes, and higher for larger problem sizes.

   to identify performance characteristics and issues and to prioritize project activities.

PENG Activities

2. Identifying and eliminating unscalable algorithms (computation, communication, and/or memory requirements), e.g.
   - Replacing single reader/writer I/O with parallel I/O infrastructure
   - Replacing “master computes” with distributed algorithms that still preserve numerical reproducibility (where required)
   - Replacing undistributed data structures (replicated for “convenience”) with distributed data structures, and modifying associated algorithms accordingly.
3. Identifying and eliminating unnecessary algorithmic restrictions on scalability
   - Limitations in one phase of the code need not limit another phase
   - Identifying other work that can be computed simultaneous to a phase with limited parallelism (and modifying code to support this functional parallelism)

PENG Activities

4. Identifying, analyzing, and optimizing code that performs poorly at scale
   - "standard tricks" applied to situations arising at scale: hiding latency, overlapping communication and computation, performance portable implementations that can take advantage of or address platform -specific characteristics
   - evaluating MPI collectives, point-to-point communication protocols, and alternative messaging layers
6. Identifying and evaluating alternative approaches more suitable at scale, for example
   - new atmosphere dynamics solvers using cubed sphere or icosahedral horizontal computational grid instead of traditional latitude/longitude grid. Active or planned evaluation targets include:
     * HOMME/Spectral Element dynamical core from NCAR and SNL
     * Next generation Finite-Volume dynamical core from NOAA and NASA

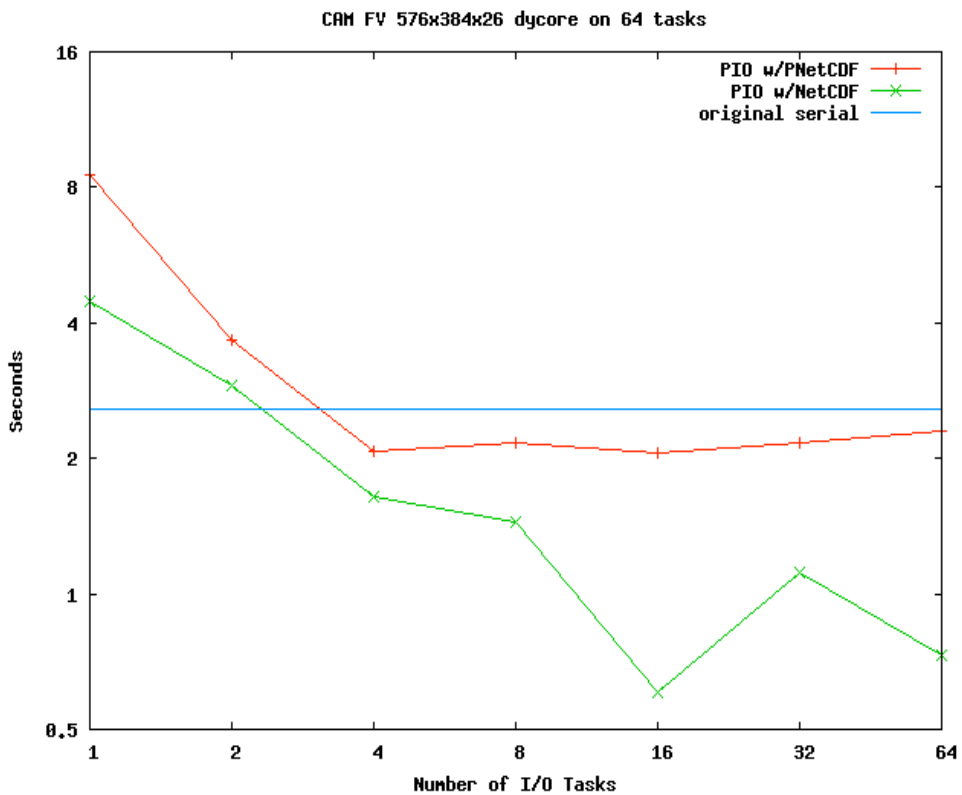
Highlights: CCSM and Blue Gene

- The first port of CCSM3 to Blue Gene/L
  - Starting point: maximum run size of 32 processes on small grids
- Addressed memory scaling problems in coupler
  - Router initialization
    - Overlapping segment storage reduced to 2% original
  - Distributed Sparse Matrix multiply storage
  - Router GSMap storage
- Results
  - Dead models: e.g. T42 gx1v3 32→1024 processors
  - Full models: → T31 gx1v3 1024 processors
    - Executable image size a limiting factor on BG/L
- Coupler fixes will be part of official CCSM4 release
- Initial port to Blue Gene/P

9

Highlights: Parallel I/O in CCSM

PIO improves performance of atmosphere model on 0.5 degree grid, 64 MPI tasks,



PIO in experimental atmospheric model enabling very high resolution runs

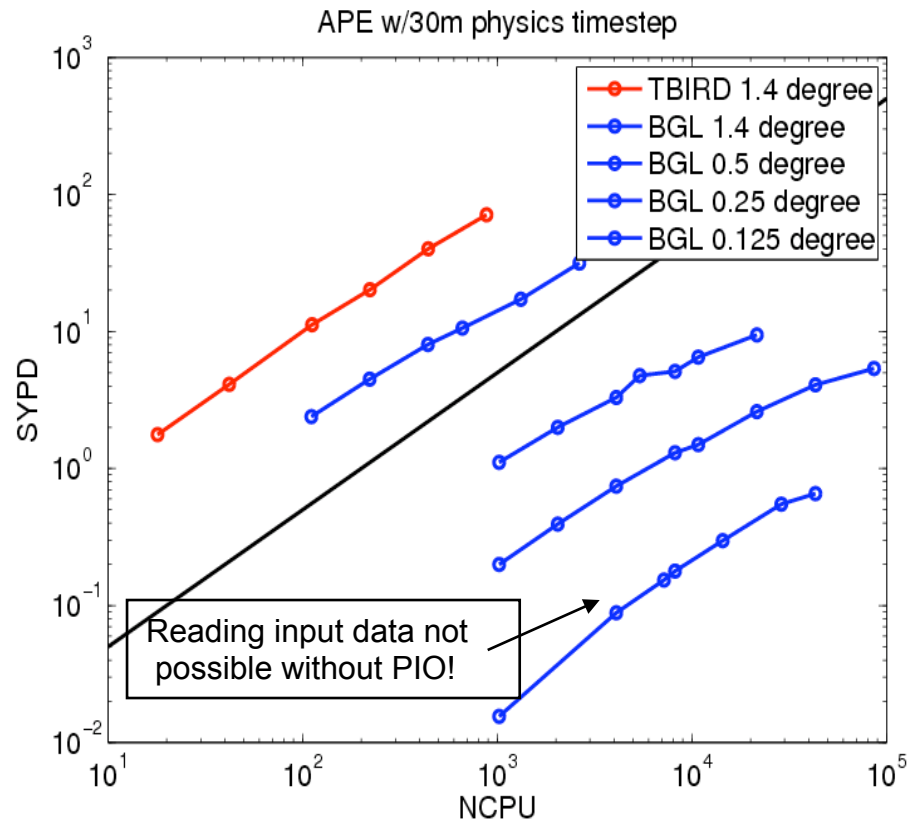


Figure provided by Mark Taylor, Sandia National Lab

CCSM Status:

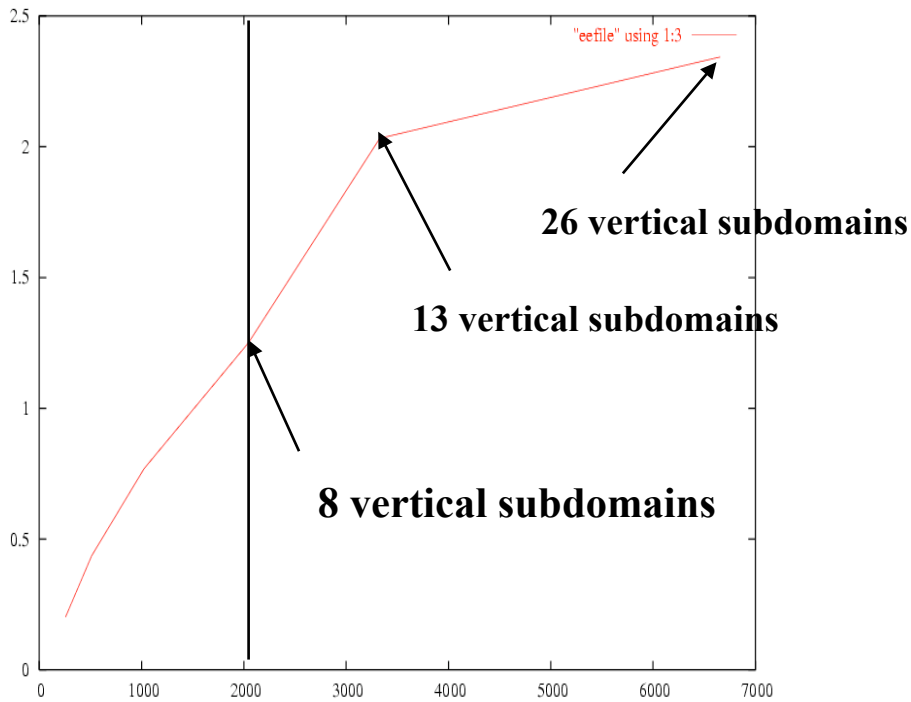
- Atmosphere: read and write history and restart
- Ocean: read and write history and restart
- Land: write history
- Sea Ice and Coupler: in progress

Highlights: More Parallelism in ATM

1. 3X MPI parallelism for entire ATM model (by increased vertical decomposition in dynamics) when using Finite Volume (FV) dynamics
   - *Important now*
2. > 9X MPI parallelism for some phases in ATM model
   a. Decoupling FV dynamics longitude/latitude decomposition from FV dynamics latitude/vertical decomposition
   b. Decoupling physics parallelization from dynamics parallelization for all dynamics options
   - *Important now when physics is more expensive to compute than dynamics and when OpenMP parallelism is limited (or unavailable). For example, we expect the superparameterization technique for resolving clouds physics to be able to use 10,000-way parallelism productively given this capability (and PIO), even for current grid sizes. Cost of physics is increasing relative to dynamics in all future plans, so general applicability will increase in the future.*

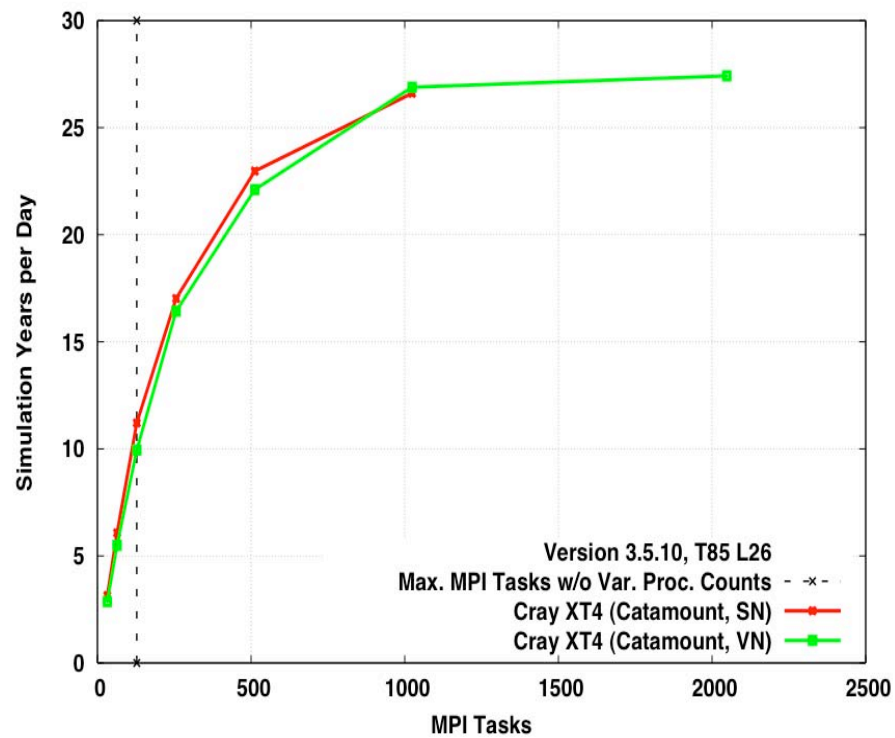
Eliminating Scalability Restrictions

Increased Decomposition in the Vertical



ATM with FV Dynamics: .5 degree (768x1152x26) grid on Atlas system at LLNL

Using More Physics than Dynamics Processes



ATM with Spectral Eulerian Dynamics: T85 (128x256x26) grid on dual-core XT4 at ORNL

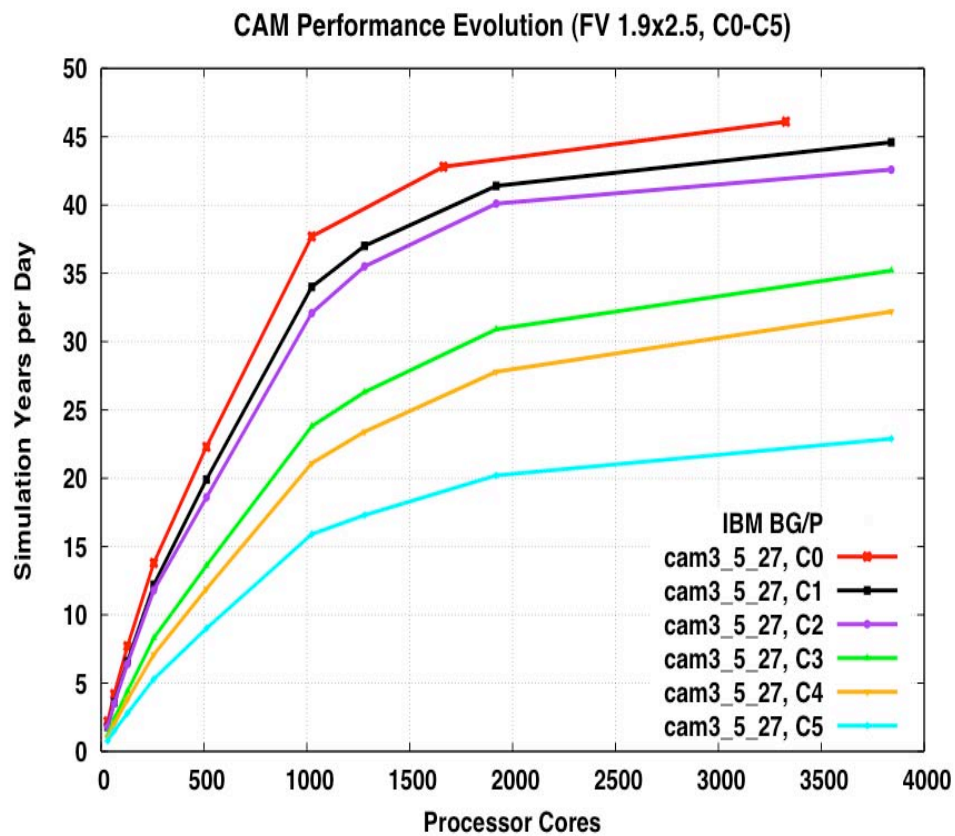
Highlights: More Parallelism in ATM

3. Parallelization over tracer indices in FV tracer advection, with option to overlap with primary dynamics time advance algorithm
   - *In combination with other optimizations, will be very important for full atmospheric chemistry, where just the increase in the number of tracers more than doubles the simulation cost.*

	tracer advection	atm run time
control	3	80
99 extra tracers	78	322
5*832 (=4160) proc	78	270
5-way decomp.	18	224

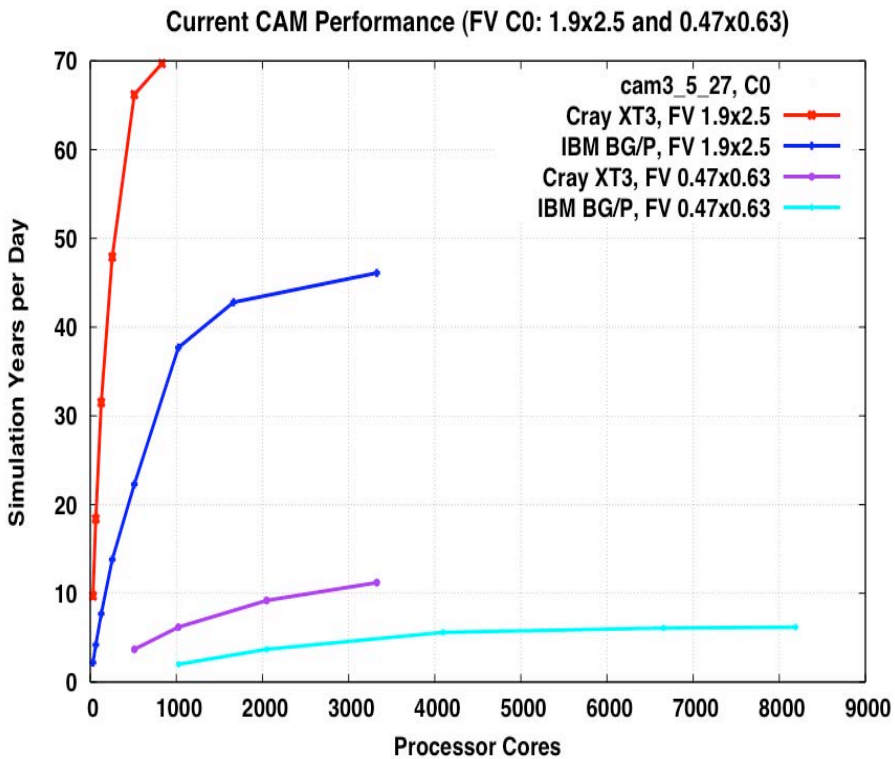
ATM with FV Dynamics: 1 degree (192x288x26) grid on Atlas system at LLNL. Uses 832 processors prior to optimization. Five-way tracer decomposition calculates tracer advection 4.3 times faster. Overall throughput increases by 44%. (This includes benefit of extra processors for physics.)

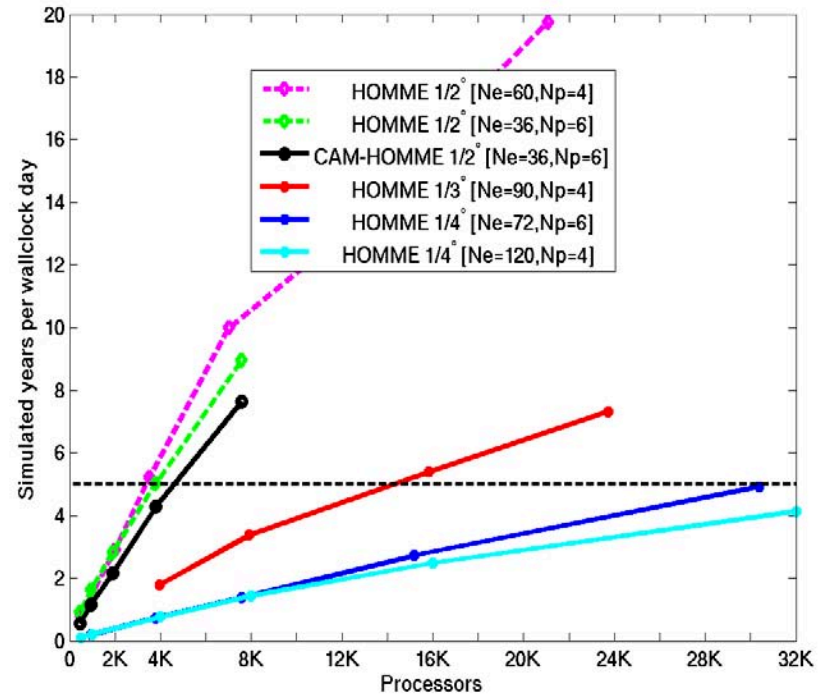
Highlights: Evaluating Performance Impact of New Science in ATM



FV Dynamics: 96x144x26 (C0) and 96x144x30 (C1-C5) on IBM BG/P system for a number of new science options being considered for production runs. Cost of new physics increases cost of model by 2.5X. Cost of other proposed changes, e.g., atmospheric chemistry or 48 vertical levels, will increase cost even more.

Highlights: Evaluating Performance Impact of Higher Resolution on ATM





FV dynamics on dual-core Cray XT3 and quad-core IBM BG/P (left); HOMME on BG/L (right). Performance scalability work is far from complete, especially for higher resolution grid. Achieving further scalability will likely require moving to a more scalable dycore, such as HOMME or cubed sphere FV.

Highlights: Recent Odds and Ends

(Note: N == horizontal grid size, P == number of MPI processes.)

1. Fast reproducible global mean algorithm
   - Replaced master computes algorithm with distributed algorithm, reducing memory and computational complexity from O(N) to O(N/P). Reduced time from 4% of total ATM execution time when using FV dynamics and .5 degree grid on 1664 XT3 processors to ~0.1% (50 times faster).
2. Determination of amount of memory needed to gather distributed data structure into master
   - Replaced O(PN) algorithm by O(N) algorithm. Reduced time from 20% of total ATM execution time when using FV dynamics and .5 degree grid on 1024 BG/P processors to .03% (~1000 times faster).
3. Determination of whether distributed data structure has haloes
   - Replaced O(NlogN) algorithm executed each timestep with a single call during initialization. Reduced total ATM execution time when using FV dynamics for .5 degree grid on 1024 BG/P processors by 3%.

SEESM vs. PENG

Software and Performance Engineering Tasks:

- SEESM
  - Short term
  - Primarily fixing bugs (performance and otherwise) and performance tuning using existing options
  - In support of current (production and development) scenarios
  - In support of current production platforms
  - In support of other SEESM development activities
- PENG
  - Longer term
  - Primarily structural and algorithmic ("big" changes)
  - Targeting (anticipating) issues at larger scales: process count, problem size, and new science
  - Evaluating promise of new architectures
  - Evaluating promise of new algorithms/methods/components

Future Foci

While we have made good progress on a number of fronts, much of the work described here is not yet complete (e.g., PIO), and some is not "complete-able" (e.g., porting to new architectures). In an extension to match the duration of the SEESM, we propose focusing on the following.

1. Performance characterization and evaluation at even greater scale
   a. taking into account new physical processes and other new scientific capabilities that are necessary if the CCSM is to become an Earth System Model, and
   b. targeting the petascale systems that will become available during this timeframe, running with even larger processor counts.
2. Continuing evaluation of alternative computational grids/alternative numerical algorithms within CCSM, quantifying the "promise" of the cubed sphere-based numerical methods within the full atmosphere model with comparable experiments using the current methods.

Future Foci

3. Addressing known opportunities for performance optimization:
   a. Exploiting vectorization (e.g., Opteron SSE, BG/P double Hummer), especially in physics code
   b. New / improved load balancing schemes that address the characteristics of the new physical processes being introduced into the model
   c. Additional minimization of communication overhead and/or evaluating alternative methods for tracer advection
4. Addressing known limiters on scalability
   a. Memory usage in the coupling framework
   b. Memory usage in the atmosphere physics
   c. Memory usage in initialization for all components
   d. (Lack of) support in the coupler for scalability enhancements available in the component models.