SciDAC
Scientific Discovery through Advanced Computing

Office of Science
U.S. DEPARTMENT OF ENERGY

Issues and Techniques for Performance Optimization at Scale:

Recent Progress with the Community Atmosphere Model

Arthur A. Mirin
Lawrence Livermore
National Laboratory

Patrick H. Worley
Oak Ridge
National Laboratory

March 17, 2009

Climate 2009: The 11th International Specialist Meeting on
Next Generation Models on Climate Change and Sustainability for
Advanced High-Performance Computing

Oak Ridge National Laboratory
Oak Ridge, Tennessee, USA

OAK RIDGE
National Laboratory

Acknowledgements

- Research sponsored by the Atmospheric and Climate Research Division and the Office of Mathematical, Information, and Computational Sciences, Office of Science, U.S. Department of Energy under Contract No. DE-AC05-00OR22725 with UT-Battelle, LLC and Contract No. DE-AC52-07NA27344 with Lawrence Livermore National Security, LLC.
- These slides have been authored by contractors of the U.S. Government under contracts No. DE-AC05-00OR22725 and No. DE-AC52-07NA27344, and are released as LLNL Report LLNL-PRES-410942. Accordingly, the U.S. Government retains a nonexclusive, royalty-free license to publish or reproduce the published form of this contribution, or allow others to do so, for U.S. Government purposes.
- This research 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.

Community Atmosphere Model

- Atmospheric global circulation model with two primary computational phases per timestep
  - *Dynamics*: advances evolution equations for atmospheric flow
  - *Physics*: approximates subgrid phenomena, such as precipitation, clouds, radiation, turbulent mixing, …
- Multiple options for dynamics, including
  - Finite-Volume semi-Lagrangian (FV) dynamical core (*dycore*), using tensor product *longitude* x *latitude* x *vertical level* grid over the sphere
- Separate data structures for dynamics and physics and explicit data movement between them each timestep

Community Atmosphere Model

- Developed at the National Center for Atmospheric Research (NCAR), with contributions from external National Science Foundation (NSF), Department of Energy (DOE) and National Aeronautics and Space Administration (NASA) funded researchers
- Used as the atmosphere component of the Community Climate System Model (CCSM)

Context of this research

- Polar singularity limits ability to decompose domain effectively in longitude on longitude-latitude grid
- Long-term solution is to use a more favorable grid
  - cubed sphere (with, for example, a finite-volume or spectral element dynamical core)
- In near term (including IPCC AR5), we need to maximize performance of the current model formulation (on a longitude-latitude grid).
- Approach: expose additional parallelism and address remaining scaling limiters

CAM Parallelization Strategy

- Domain decomposition, where each subdomain is assigned to a single MPI process. OpenMP can be used to parallelize over the set of subdomains assigned to a process and/or over array indices within a subdomain.
- Dynamics and physics use separate decompositions.
  - Physics utilizes a fine grain 2D latitude/longitude decomposition.
  - FV Dynamics utilizes multiple decompositions: 2D block latitude/vertical and 2D block latitude/longitude.
- Transposes used to move between decompositions. The amount of interprocess communication can vary from none to a complete all-to-all, depending on the runtime choice of domain decomposition and type of load balancing.

(Original) CAM Scalability Limiters

- Number of MPI processes can not be greater than the minimum of the number of subdomains employed within each of the physics and dynamics domain decompositions.
- Number of subdomains is limited by grid resolution.
  - FV: three grid points are required in each coordinate direction in the dynamics decompositions. When coupled with a small number of vertical levels, this severely limits the number of subdomains in the latitude/vertical decomposition.

Other Scalability Limiters

At scale (problem size or processor count):

- **Unscalable computation (master-computes, redundant computation, single process complexity of order P*N, etc.)**
- **Unscalable memory requirements (O(P) size system buffers required for interprocess communications, replicated data structures, …)**
- **Unscalable assumptions (integer overflow, …)**
- **Poor communication performance arising from large number of message requests**
- **Decreased opportunity to hide communication latency**
- **Poor Input/Output (I/O) performance**

Recent Progress in CAM Throughput

We have more than doubled the performance of the Community Atmosphere Model on the Cray XT4/5, and are seeing similar improvements on the IBM BG/P. This has come about through a combination of adding additional parallelism, enabling different sections of CAM to execute at their own process count, implementing improved communication protocols particularly relevant at scale, and removing other scalability limiters.

***Throughput improvement is problem-dependent.**

*** Work carried out over past 2.5 years under SciDAC-2.**

Parallelism Enhancements

- We exposed additional parallelism and enabled different sections of code to execute at their own process count
  - allow one vertical level per subdomain
  - assign more (computational) processes to physics than dynamics
  - advect multiple tracers concurrently
  - larger longitude-latitude than latitude-vertical decomposition
  - overlap of main dynamics and tracer advection subcycles

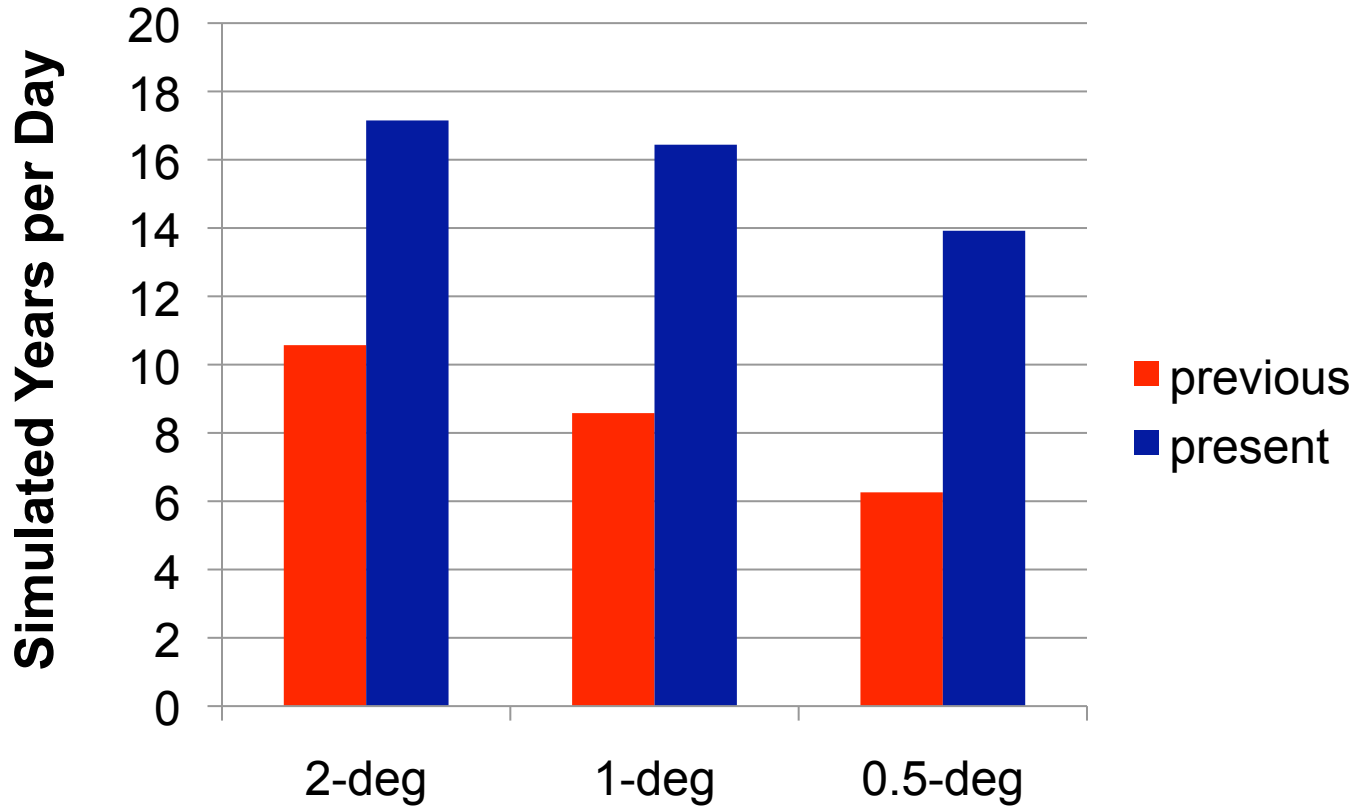
Communication Optimizations

- New performance tuning options:
  - Both point-to-point and MPI collective implementations
  - For point-to-point communication options:
    - hypercube-based ordering of communication requests
    - handshaking (wait to issue send until matching receive is issued)
    - throttling (limit number of outstanding requests)
    - blocking vs. non-blocking send
  - Option to transpose variables one at a time or in blocks (using runtime-specified block sizes)
- Different code sections can use different options

Algorithmic Optimizations

- Fast reproducible distributed sum algorithm that does not require real*16 arithmetic (replaces one-process-computes algorithm)
  - used in numerous locations, in both physics and dynamics
  - used in CCSM coupler
  - under consideration for use in CCSM ocean component
- Reproducible distributed geopotential algorithm (added to existing options based on communication transpose or on real*16 arithmetic)
- Miscellaneous others, for example,
  - replaced O(PN) algorithm by O(N) algorithm

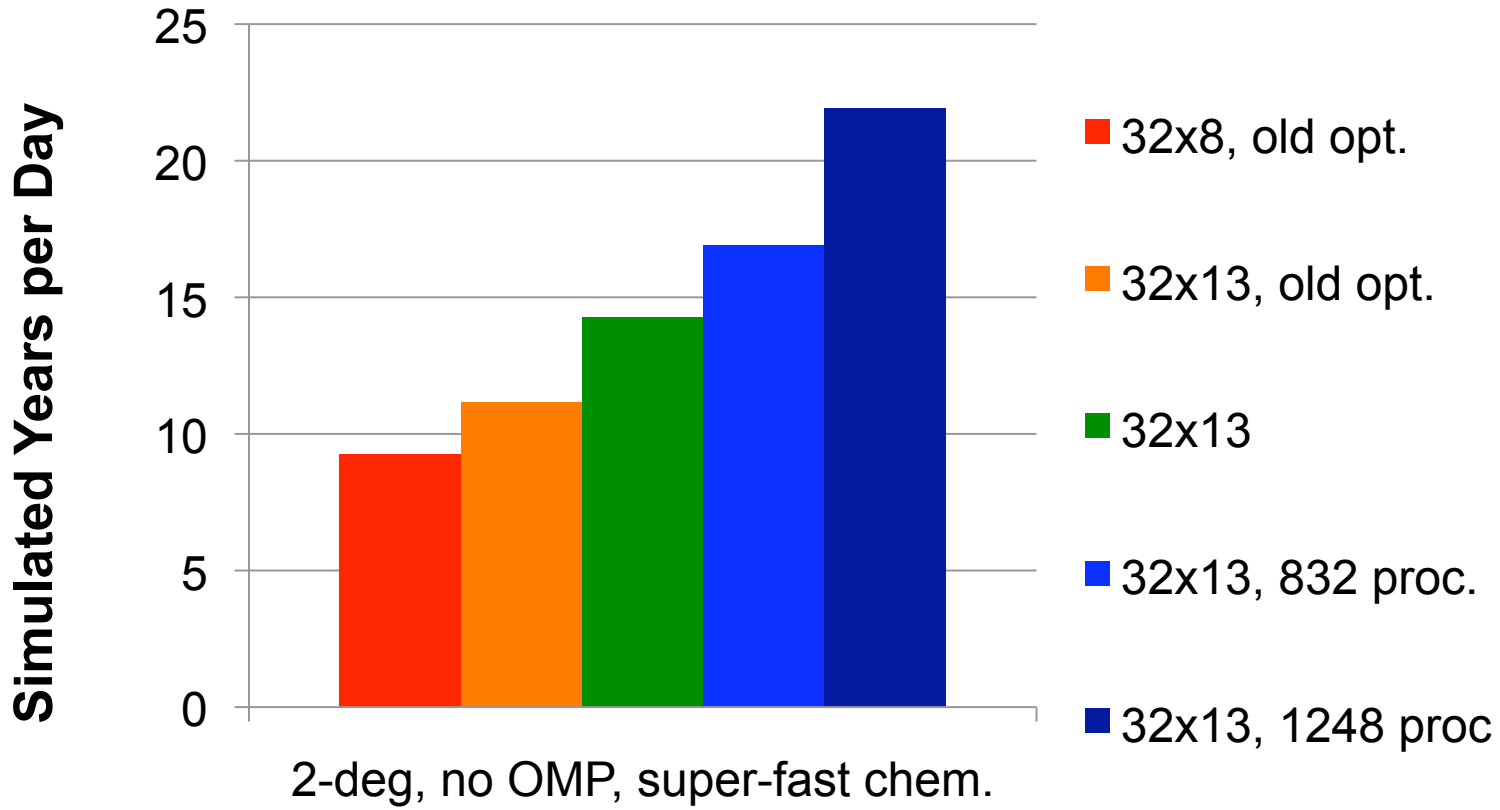
CAM with 4-way OpenMP on Cray XT4



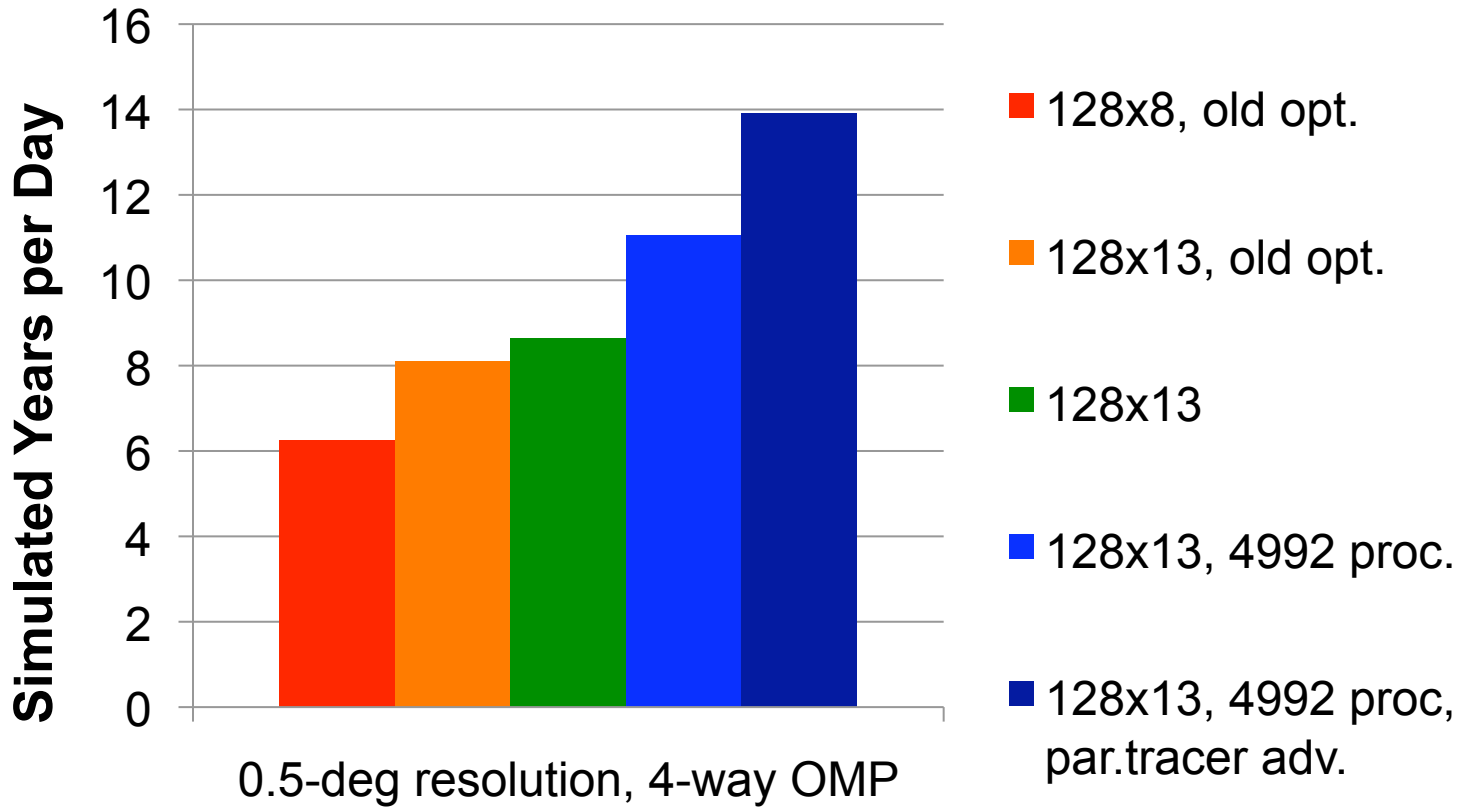
Computational Grids

2-deg: 144x96x26 **1-deg: 288x192x26** **0.5-deg: 576x384x26H**

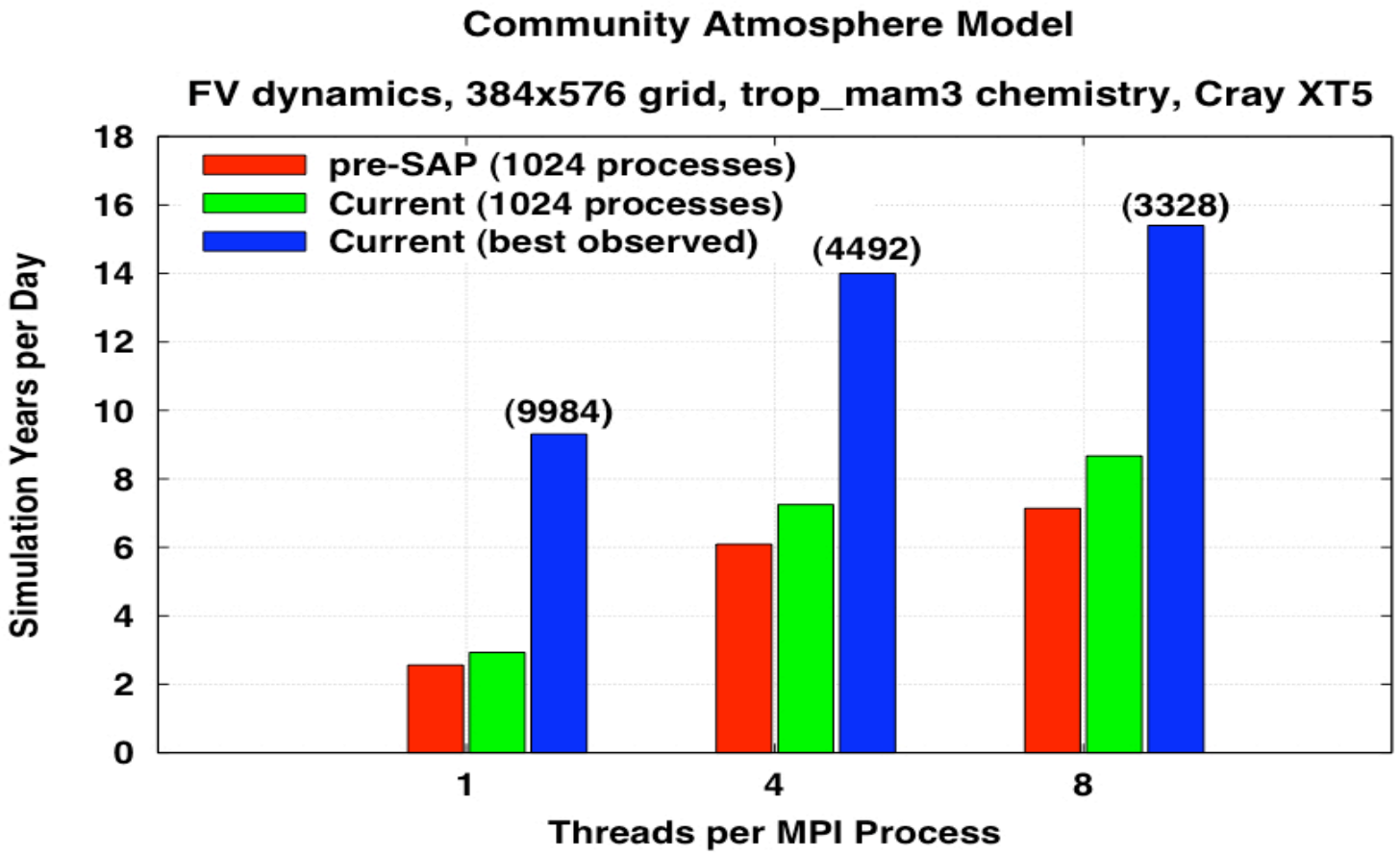
Breakdown of Perf. Improvement



Breakdown of Perf. Improvement

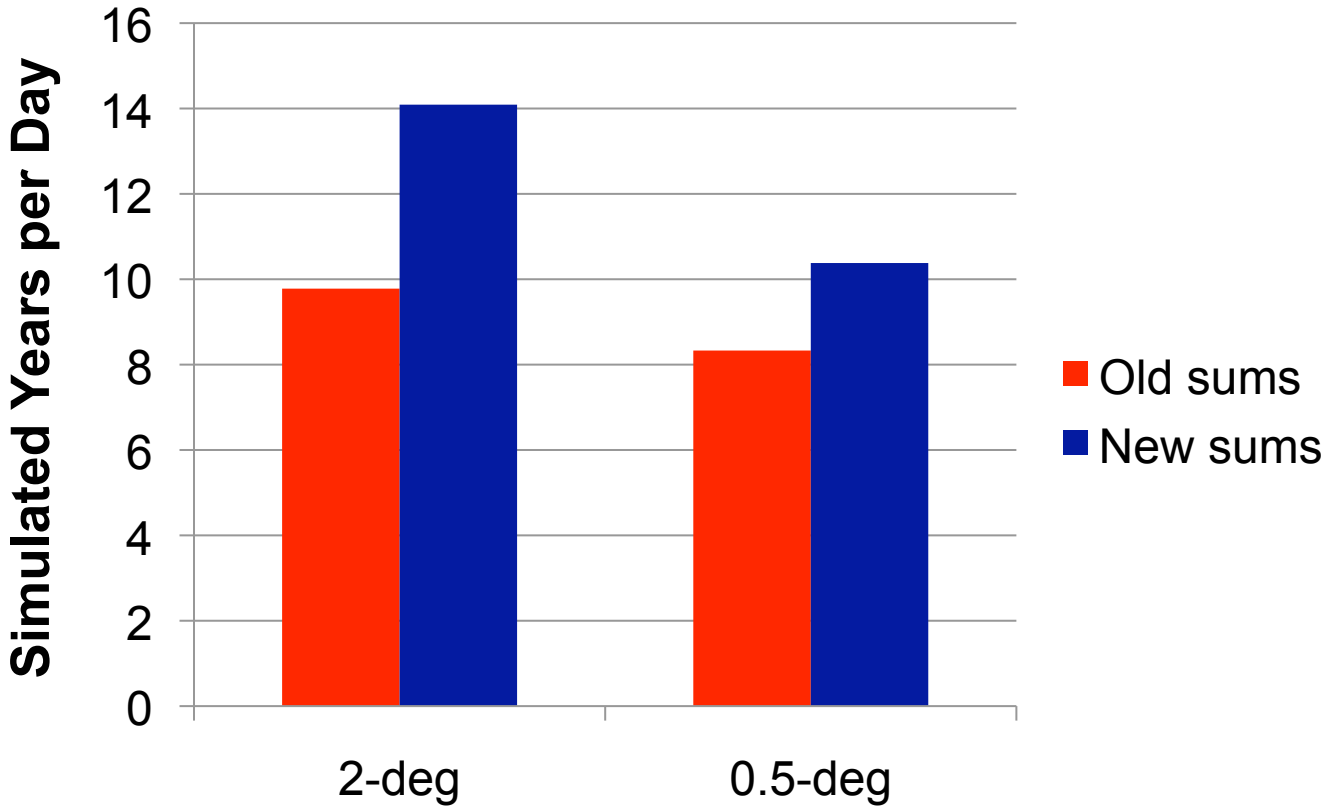


Improved Throughput On Cray XT5



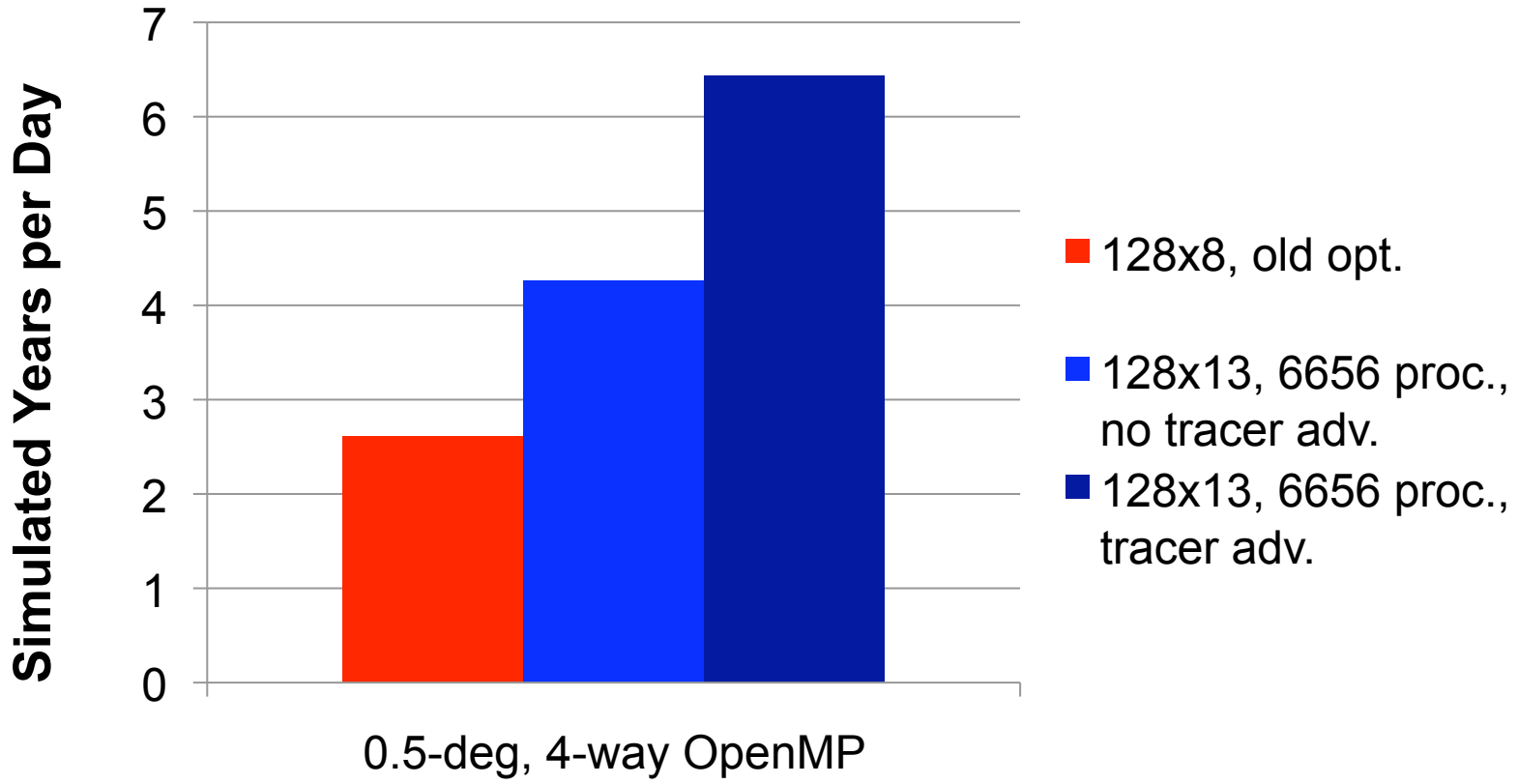
Integer overflow bug that limited thread count to no more than approximately 27,000 threads in this study has now been eliminated.

Fast Reproducible Distributed Sums



Experiments illustrating the performance improvement due to the new reproducible distributed sum algorithm. 2-deg: 2496 MPI tasks; 0.5-deg: 3328 MPI tasks, both with 4 OpenMP threads per task.

Perf. with Full Tropospheric Chemistry



Experiments illustrating performance improvement due to parallel tracer advection option. Over 100 tracers are advected in this configuration.

Ongoing work in support of AR5

- A parallel I/O capability is under active development, with many collaborators within the CCSM community
- Implement and evaluate additional OpenMP parallelism in dynamics (for when MPI parallelism is exhausted)
- Extend atmosphere model scalability improvements to other components, for example,
  - evaluate performance of reproducible distributed sum in the ocean component
  - introduce flow-control and other communication optimizations into parallel I/O layer, to improve both robustness and performance of implementation

Ongoing work in support of AR5

- Improve vectorization (for Opteron SSE and BG/P "double hummer" functional units)
- Improve memory usage throughout the model

Lessons Learned

- Scaling studies expose (new) application limiters, both in performance
  - algorithmic complexity
  - implementation inefficiencies

  and functionality
  - memory requirements
  - violated assumptions (for example, integer overflow)
- Scaling studies expose (new) system characteristics
  - performance variability
  - (physical) topology sensitivities (including I/O)
- Scaling studies are vital for anticipating future problems