

Extending Scalability of the Community Atmosphere Model

Arthur A. Mirin

Lawrence Livermore National Laboratory

Patrick H. Worley

Oak Ridge National Laboratory

SciDAC 2007

June 24-28, 2007

Westin Copley Place

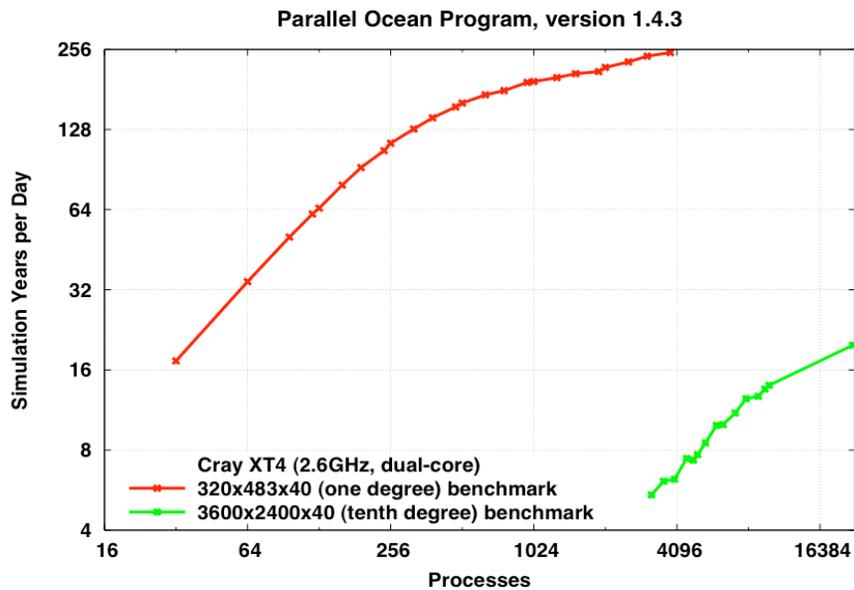
Boston, MA



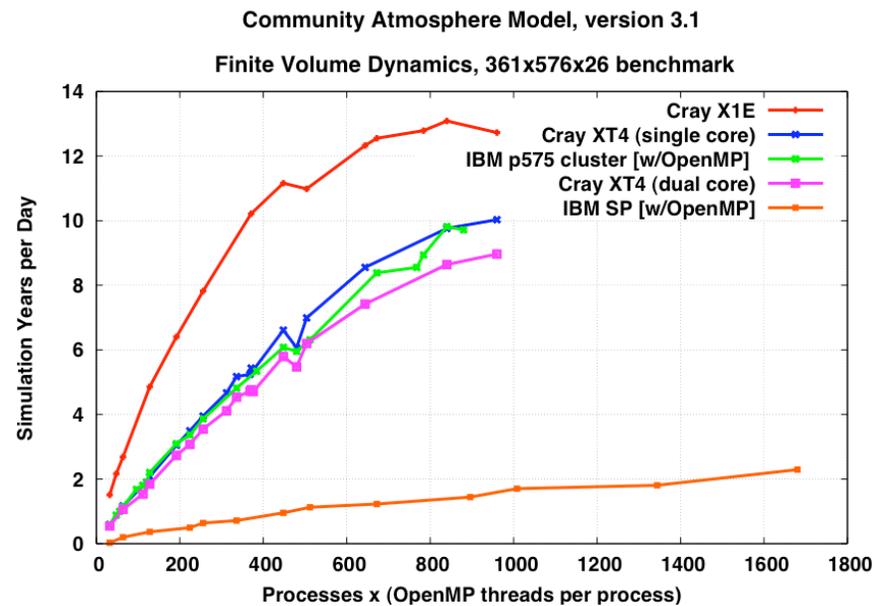
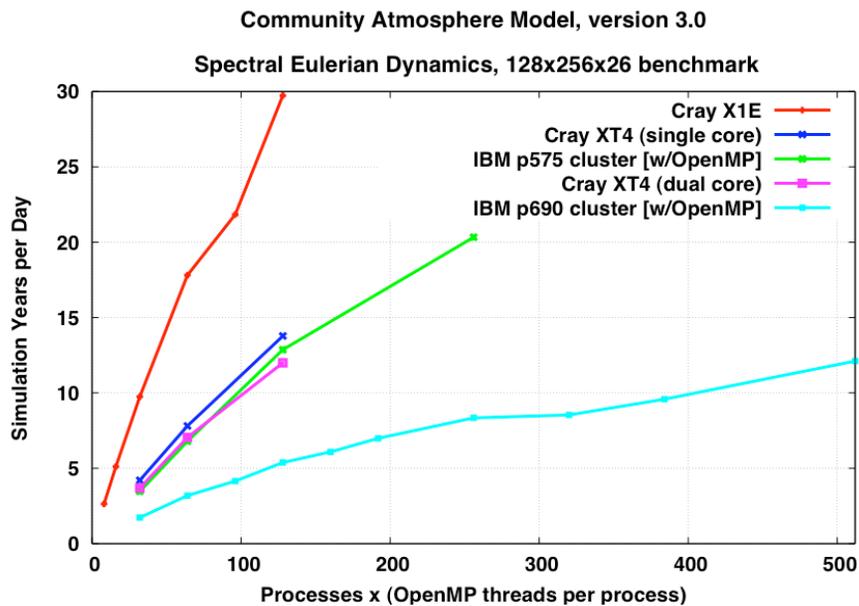
Background

- 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 Model* is working with SEESM on the long-term performance engineering of the CCSM, with an emphasis on improving problem size and processor count scalability and on planning for new science capabilities.

Scalability of CCSM components



- For both small (current production size for climate simulations) and large benchmark problems, POP ocean code can use thousands of MPI processes effectively on the Cray XT4
- For CAM atmosphere model, MPI scalability is severely limited (to 128 and 960 processes, respectively, for representative small and large benchmark problems). Improved MPI scalability would improve performance even when OpenMP parallelism is available.



Community Atmosphere Model (CAM)

- Atmospheric global circulation model
- Primary consumer of computer resources in typical CCSM simulations
- Timestepping code with two primary 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 (and more coming):
 - Finite-Volume semi-Lagrangian (FV) dynamical core (*dycore*)
 - Spectral Eulerian (EUL) dycore
 - Spectral semi-Lagrangian (SLD) dycore

all using tensor product *latitude x longitude x vertical level* grid over the sphere, but not same grid, same placement of variables on grid, or same domain decomposition in parallel implementation
- Same grid but separate data structures for dynamics and physics, and explicit data movement between them each timestep

CAM Parallelization Strategy

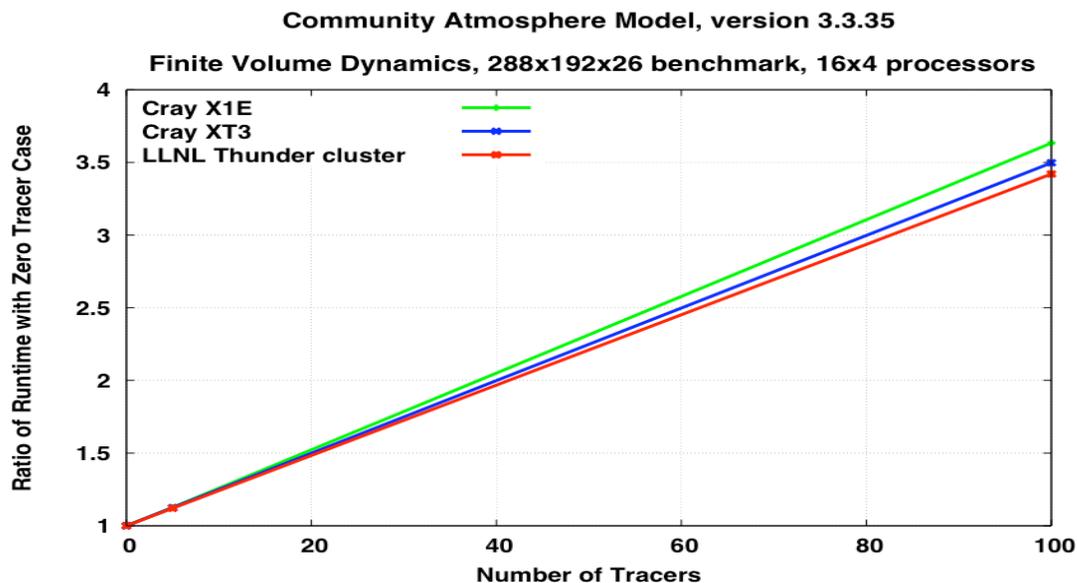
- Domain decomposition, where each subdomain is assigned to a single MPI process. When available, OpenMP is used to parallelize over the set of subdomains assigned to a process and/or over array indices within a subdomain.
- Dynamics and physics use separate decompositions.
 - Physics utilizes a fine grain 2D latitude/longitude decomposition.
 - Dynamics utilizes multiple decompositions.
 - FV: 2D block latitude/vertical and 2D block latitude/longitude, with same number of blocks in each
 - EUL and SLD: 1D latitude in physical space and 1D wavenumber in spectral space
- Transposes are used to move between decompositions.

CAM Parallel 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 (and climate simulations employ relatively modest resolutions).
 - 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.
 - EUL and SLD: decompositions are one dimensional, and the number of MPI processes can not be greater than the number of latitudes. (CAM allows processes to be idle in the spectral space decomposition.)
- Communication cost of transposes, load imbalance, I/O, global diagnostics, ... also affect parallel scalability.

Future Computational Challenges for CAM

- Requirement to support a range of horizontal resolutions efficiently:
 - FV: 2 degree (96x144 grid), 1 degree (192x288 grid), 0.5 degree (384x576 grid), 0.25 degree (768x1152 grid)
- Inclusion of cloud resolving physics (increasing physics computation *significantly*)
- Inclusion of atmospheric chemistry, requiring up to a hundred chemical constituents (tracers):
 - increasing both dynamics and physics computation
 - increasing dynamics communication for tracer advection



Each additional tracer adds approx. 2% to the overall runtime. Runtime is 3.5 times as long when using 100 tracers.

Opportunities to Improve Scalability

1. Lagrangian remap in FV dynamics is columnar (coupling in the vertical only) and can use a much finer decomposition than the main FV dynamics.
 - Scaling is limited by cache effects degrading performance for very small subdomains.
2. Physics is columnar and can use much finer decompositions than the main dynamics (FV, EUL, or SLD).
 - Scaling is limited by cache effects and, to a lesser extent, by load imbalance.
3. Tracer advection
 - Admits finer vertical decomposition (compared to dynamics) since it does not couple vertically,
 - Can be decomposed over tracer index, and
 - Can be partially overlapped with main dynamics.
4. Portions of the atmospheric chemistry do not couple vertically and can be decomposed vertically as well as horizontally.
5. Cloud resolving physics uses much higher resolution and can therefore utilize many more subdomains.

Initial Approach: Variable Process Count

1. Allow the FV latitude/vertical decomposition to have a different number of subdomains than the FV latitude/longitude decomposition, thus allowing a different number of active MPI processes to be used in the respective phases.
2. Allow the number of active MPI processes to be different in the dynamics (FV, EUL, SLD) and in the physics.
3. Allow the existence of auxiliary processes that can be employed for alternate decompositions as needed, such as
 - Decomposition over tracers during advection,
 - Finer vertical decomposition for tracer advection,
 - Overlap of tracer advection and main dynamics,
 - 3D decomposition in physics for chemistry, and
 - Additional subdomains in cloud-resolving physics

Potential Increase in Scalability: An Example

- FV dynamics, 0.5 degree grid (384x576x26), 100 tracers
 - Consider a latitude/vertical dynamics decomposition based on a 96x7 virtual processor grid (approx. 4 latitudes and 4 levels per subdomain)
 - Consider a latitude/longitude dynamics decomposition based on a 96x144 virtual processor grid (4 longitudes and 4 latitudes per subdomain)
 - For the physics, consider a decomposition into 13824 subdomains (=96x144) (16 vertical columns per subdomain)
 - Decompose tracers into 20 groups of 5 for purposes of advection (96x7x20 = 13440 processes)

Compared to the original 96x7 decomposition, we can use approximately 20 times as many processes for the Lagrangian remap, tracer advection, and physics. The performance advantage from this approach depends on the relative amount of runtime spent in the code where only 96x7 processes are active.

Work Plan: FV dynamics

1. Allow auxiliary processes: *complete*
2. Allow the latitude/vertical decomposition to have fewer subdomains than the latitude/longitude decomposition: *complete*
3. Allow the number of active MPI processes to be smaller in the dynamics than in the physics: *in progress*
4. Decompose tracer advection with respect to tracer index (implementing latitude/vertical/tracer decomposition)
5. Consider finer vertical decomposition for tracer advection (vs. main dynamics)
6. Consider overlap of tracer advection (n tracer subcycle) with main dynamics (corresponding to $(n+1)$ tracer subcycle)

Initial FV Performance Impacts

Results from Cray XT4 (2.6 GHz dual-core AMD Opteron processor):

1. 1.9x2.5 degree resolution (96x144x26 grid)

- 256 processes for latitude/vertical decomposition (maximum)
- 256, 512, and 1024 processes for other decompositions

| | secs/day | | | phys/dyn & phys/land comm. | |
|------|----------|----------|---------|----------------------------|------|
| | total | dynamics | physics | | |
| 256 | 3.63 | 1.37 | 1.61 | 0.18 | 0.08 |
| 512 | 2.94 | 1.61 | 0.83 | 0.31 | 0.04 |
| 1024 | 3.05 | 1.88 | 0.44 | 0.36 | 0.08 |

2. 0.9x1.25 degree resolution (192x288x26 grid)

- 512 processes for latitude/vertical decomposition (maximum)
- 512, 1024, and 2048 processes for other decompositions

| | secs/day | | | phys/dyn & phys/land comm. | |
|------|----------|----------|---------|----------------------------|------|
| | total | dynamics | physics | | |
| 512 | 8.80 | 4.38 | 3.00 | 0.35 | 0.54 |
| 1024 | 6.95 | 4.54 | 1.60 | 0.28 | 0.10 |
| 2048 | 7.45 | 5.34 | 0.80 | 0.57 | 0.06 |

Initial FV Performance Impacts

Results from Cray XT4 (2.6 GHz dual-core AMD Opteron processor):

3. 0.5x0.625 degree resolution (361x576x26 grid)
 - 960 processes for latitude/vertical decomposition (maximum)
 - 960 and 1920 processes for other decompositions

| | | secs/day | | phys/dyn & phys/land comm. | |
|------|-------|----------|---------|----------------------------|------|
| | total | dynamics | physics | | |
| 960 | 24.9 | 16.5 | 6.00 | 0.84 | 0.29 |
| 1920 | 21.0 | 15.9 | 3.10 | 0.70 | 0.15 |

-
- Doubling number of processes improves throughput by 1.23, 1.27, and 1.19 times, respectively.
 - Most of the improvement is due to improved performance in physics, for which doubling processes approximately halves runtime.
 - Using additional processes for dynamics latitude/longitude decomposition can cause dynamics runtime to increase. Need to try to further optimize communication within the dynamics.
 - For these benchmarks, using additional processes only in physics may be preferred. Adding atmospheric chemistry (and additional tracers) will change performance characteristics.

Work Plan: EUL and SLD dynamics

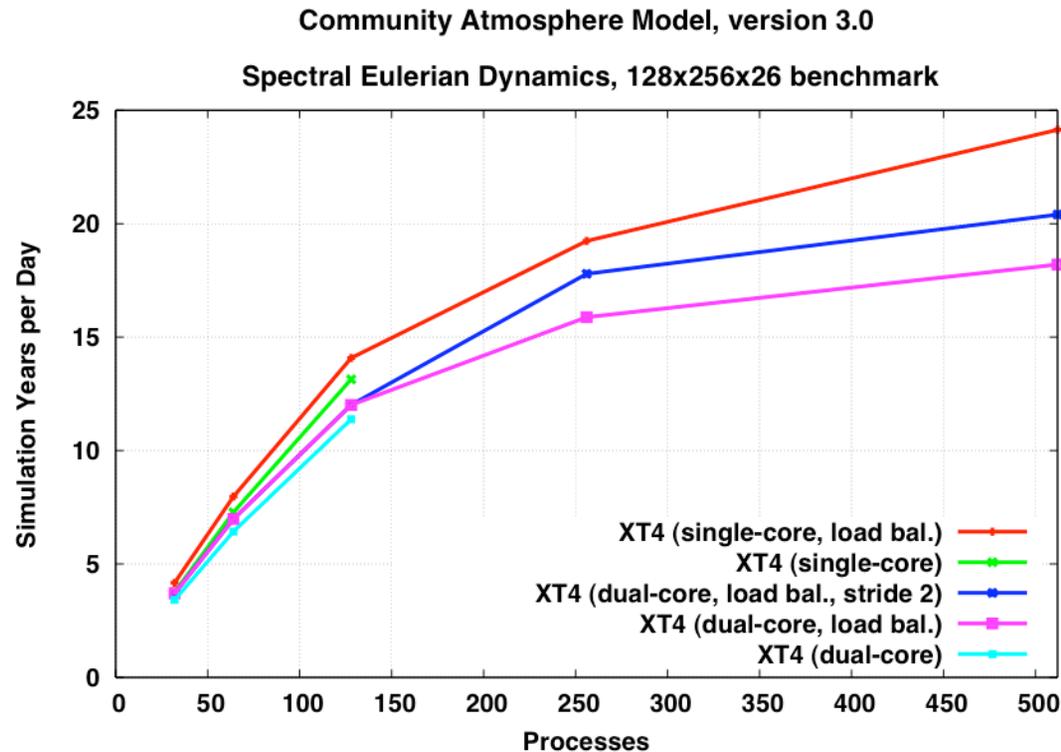
1. Allow the number of active MPI processes to be smaller in the dynamics than in the physics: **complete**
2. Introduce runtime arguments specifying number of active dynamics processes and stride (separation between active processes in logical ordering), allowing the user to specify which processors are active and which are idle in the dynamics: **complete**

EUL Performance Impacts

Results from Cray XT4 (2.6 GHz dual-core AMD Opteron processor):

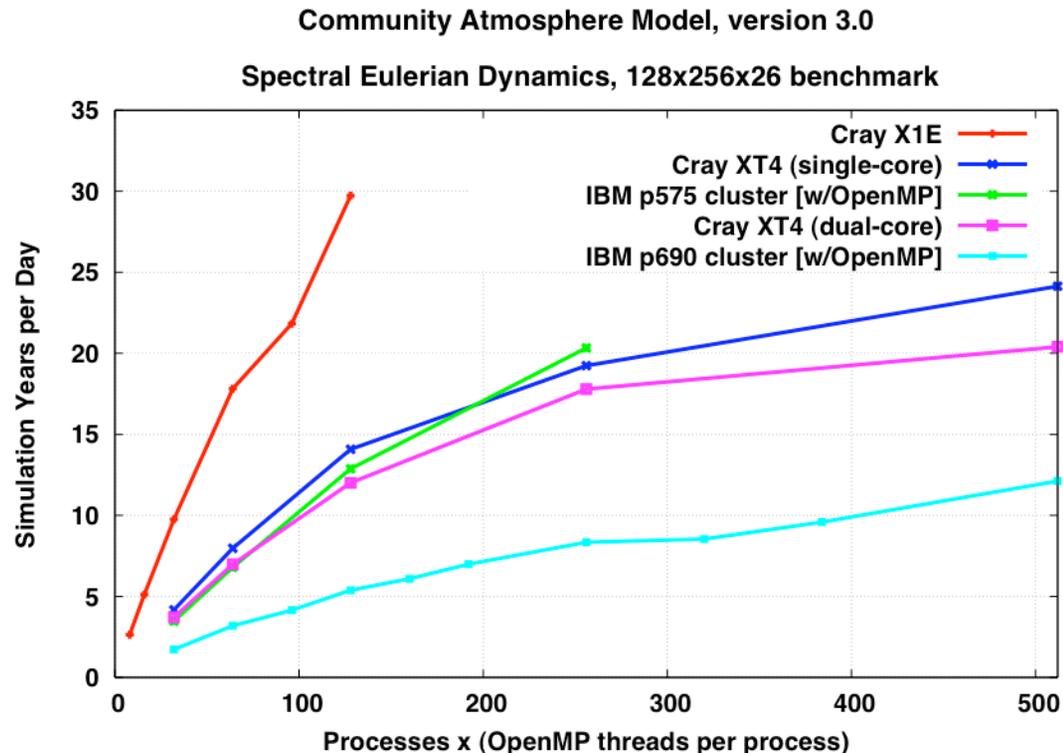
- T85 resolution (128x256x26 grid)
 - 32, 64, and 128 processes for dynamics latitude decomposition (128 is maximum)
 - 32, 64, 128, 256, 512 processes for physics decomposition
 - Physics load balancing is the mechanism for distributing work to more processes in the physics than in the dynamics. Without load balancing, the same decomposition (and the same number of processes) is used in the physics and the dynamics.
 - Both computation and communication performance are improved (for a fixed number of processes) when using only one processor core per node on the XT4. By specifying “stride 2” when using more than 128 processes, at most one core is used per node in the dynamics. Otherwise, active dynamics processes are allocated to both cores in the active nodes.
 - Also looking at performance when using only one core per node for both physics and dynamics (‘single-core’)

EUL Performance Impacts



- Using additional processes improves throughput by 1.61 times (single-core) and 1.62 times (dual-core).
- Using 1024 processes (not shown here) does not improve performance further - the cost of communication between physics and dynamics and between physics and surface models (land, ice, ocean) is increasing, offsetting further decreases in physics runtime.

Variable Process Count vs. OpenMP



- Fixing MPI process count at 64 and using 2 and 4 OpenMP threads per process on the p575 cluster is more efficient than using additional MPI processes on the XT4, probably because OpenMP can also be used within the dynamics and is not subject to increasing MPI communication costs.
- Experimental cloud resolving version of CAM (super-parameterization approach with SLD dynamics on 64x128x26 grid) using 512 processors on IBM p575 cluster is slightly faster using variable process count than using OpenMP.

Summary

- Initial results are promising.
 - On systems without OpenMP, we can now use additional MPI processes to increase throughput. We expect that the combination of additional MPI processes and OpenMP to be useful on systems with OpenMP.
 - This approach to increasing algorithmic scalability will become even more important as new science capabilities are added, for example, when adding atmospheric chemistry or in cloud resolving models.
- More work needs to be done.
 - Finish FV work plan
 - Add ability to specify which processors are active and which are idle in FV dynamics
 - Evaluate performance on systems with OpenMP
 - Address new scalability limiters that are becoming obvious as the process count increases (land/atmosphere coupling, communication between latitude/longitude and latitude/vertical decompositions, I/O, ...)

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. W-7405-Eng-48 with the University of California Lawrence Livermore National Laboratory.
- These slides have been authored by contractors of the U.S. Government under contracts No. DE-AC05-00OR22725 and No. W-7405-Eng-48, and are released as LLNL Report UCRL-PRES-231564. 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, and of the National Energy Research Scientific Computing Center, which is supported by the Office of Science of the U.S. Department of Energy under Contract No. DE-AC03-76SF00098.