Extending Scalability of the Community Atmosphere Model

Arthur A. Mirin
Lawrence Livermore National Laboratory

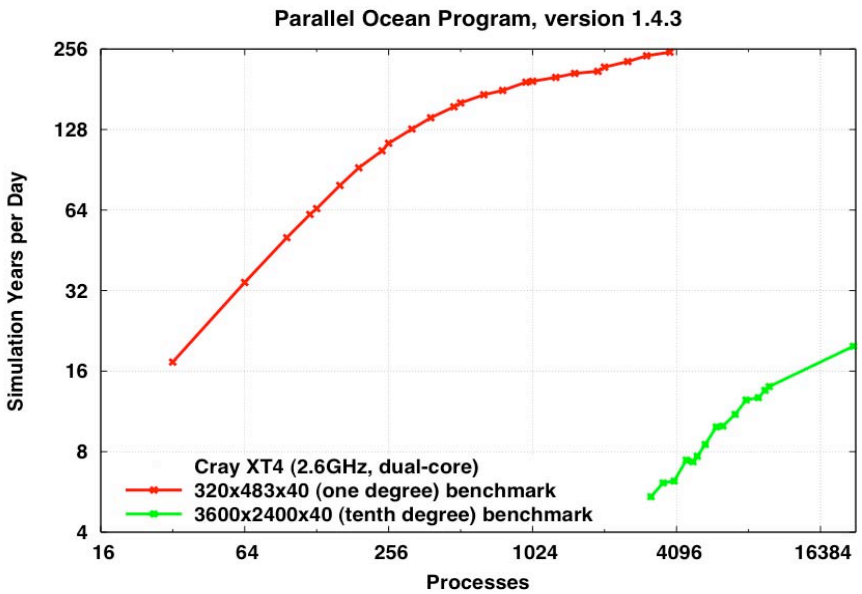
Patrick H. Worley
Oak Ridge National Laboratory

2007 DOE Day of Science
October 29, 2007
Knoxville Convention Center
Knoxville, TN

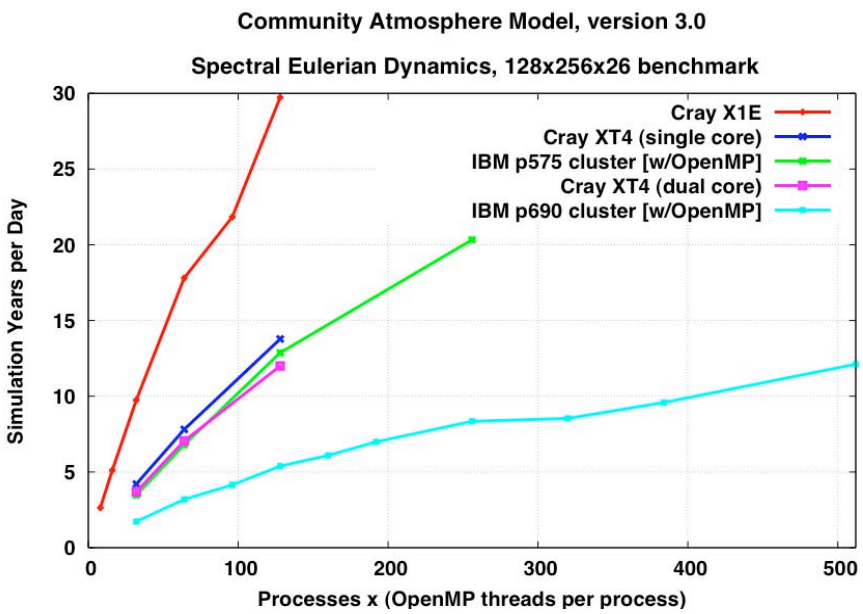
Background

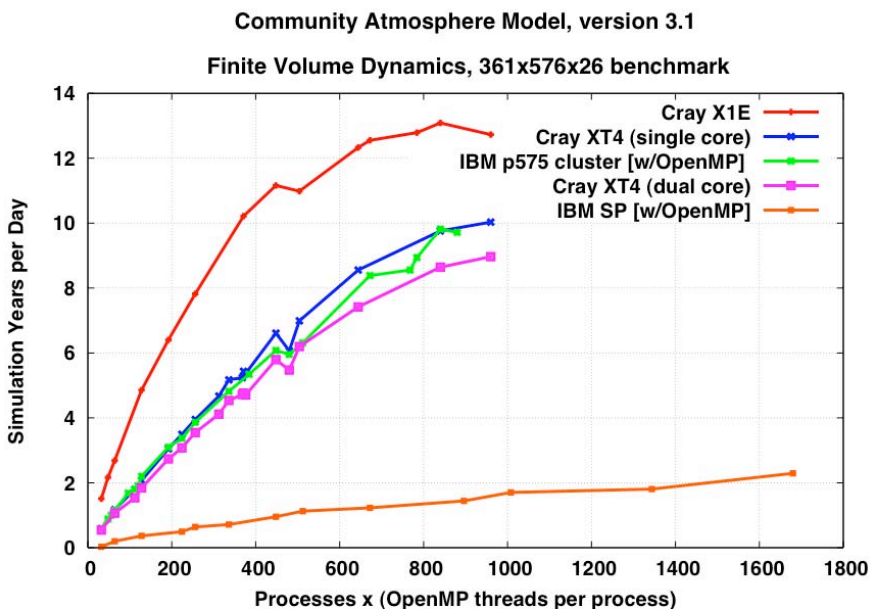
- Community Climate System Model (CCSM)
  - Fully-coupled, global climate model that provides state-of-the-art computer simulations of the Earth's past, present, and future climate states
  - Comprised of a coupler and four component models: atmosphere, ocean, land, and sea ice
  - Developed at the National 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 (PENG)* 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. PENG is a 3 person project: Ray Loy at ANL, Art Mirin at LLNL, and Pat Worley at ORNL.

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

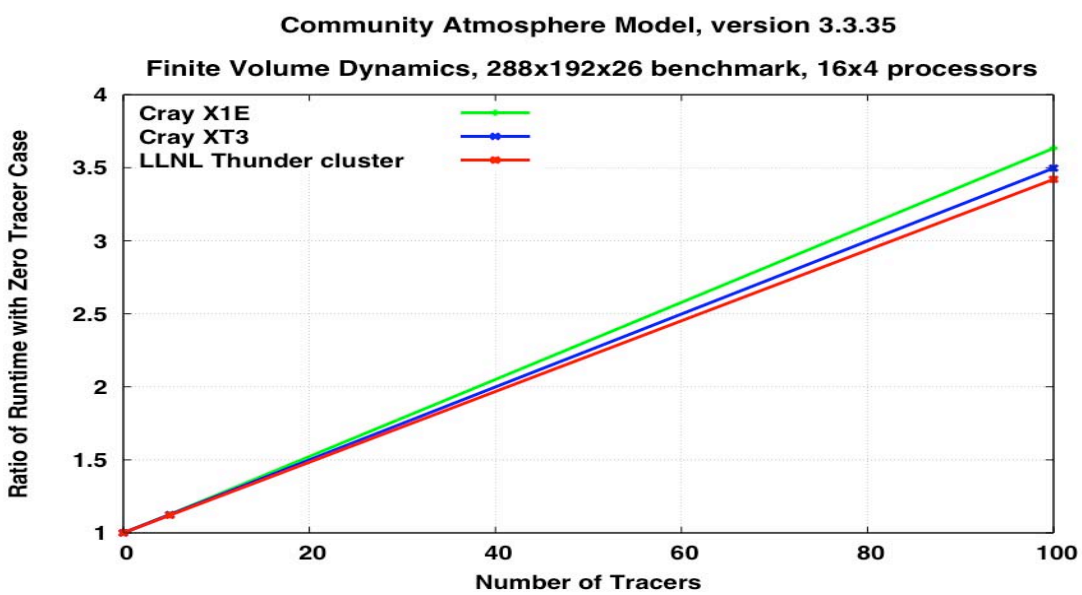
- Domain decomposition, where each subdomain is assigned to a single MPI process. When available, OpenMP is used to parallelize over the set of subdomains assigned to a process and/or over array indices within a subdomain.
- Dynamics and physics use separate decompositions.
  - Physics utilizes a fine grain 2D latitude/longitude decomposition.
  - Dynamics utilizes multiple decompositions.
    - FV: 2D block latitude/vertical and 2D block latitude/longitude
    - 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 number of subdomains employed within the relevant domain decomposition.
- 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

** #1 and #2 checked in on 9/5/2007, and available in cam3_5_10.*

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 the latitude/vertical decomposition to have fewer subdomains than the latitude/longitude decomposition: *complete*
2. Allow the number of active MPI processes to be smaller in the dynamics than in the physics: *complete*
3. Allow auxiliary processes: *initial design and implementation complete*
4. Introduce runtime argument specifying separation between active processes in logical ordering (*stride*), allowing the user to specify which processors are active and which are idle in the dynamics
5. Decompose tracer advection with respect to tracer index (implementing latitude/vertical/tracer decomposition)
6. Consider finer vertical decomposition for tracer advection (vs. main dynamics)
7. Consider overlap of tracer advection (*n* tracer subcycle) with main dynamics (corresponding to (*n+1)* tracer subcycle)

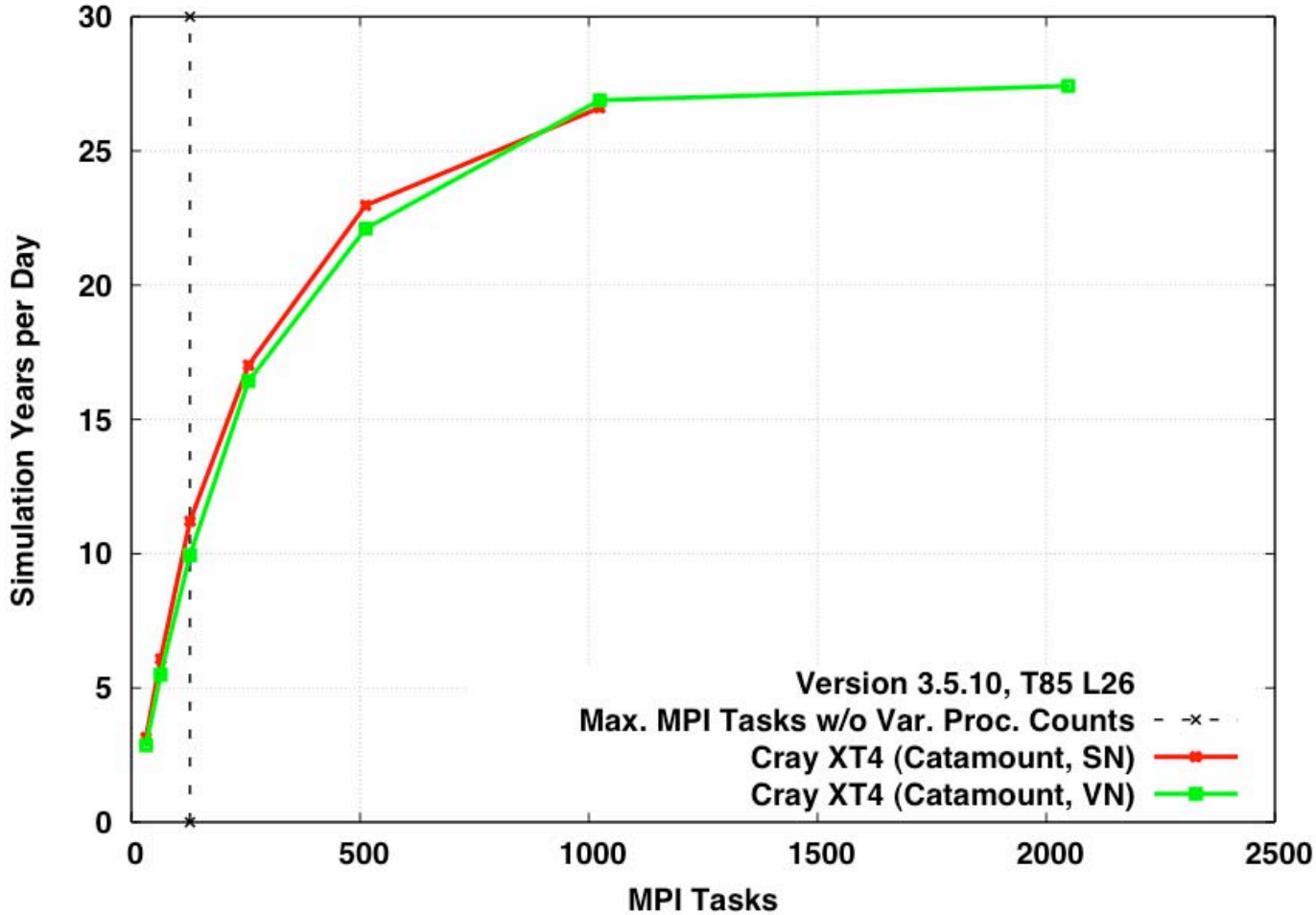
Work Plan: EUL and SLD dynamics

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

Initial Performance Measurements

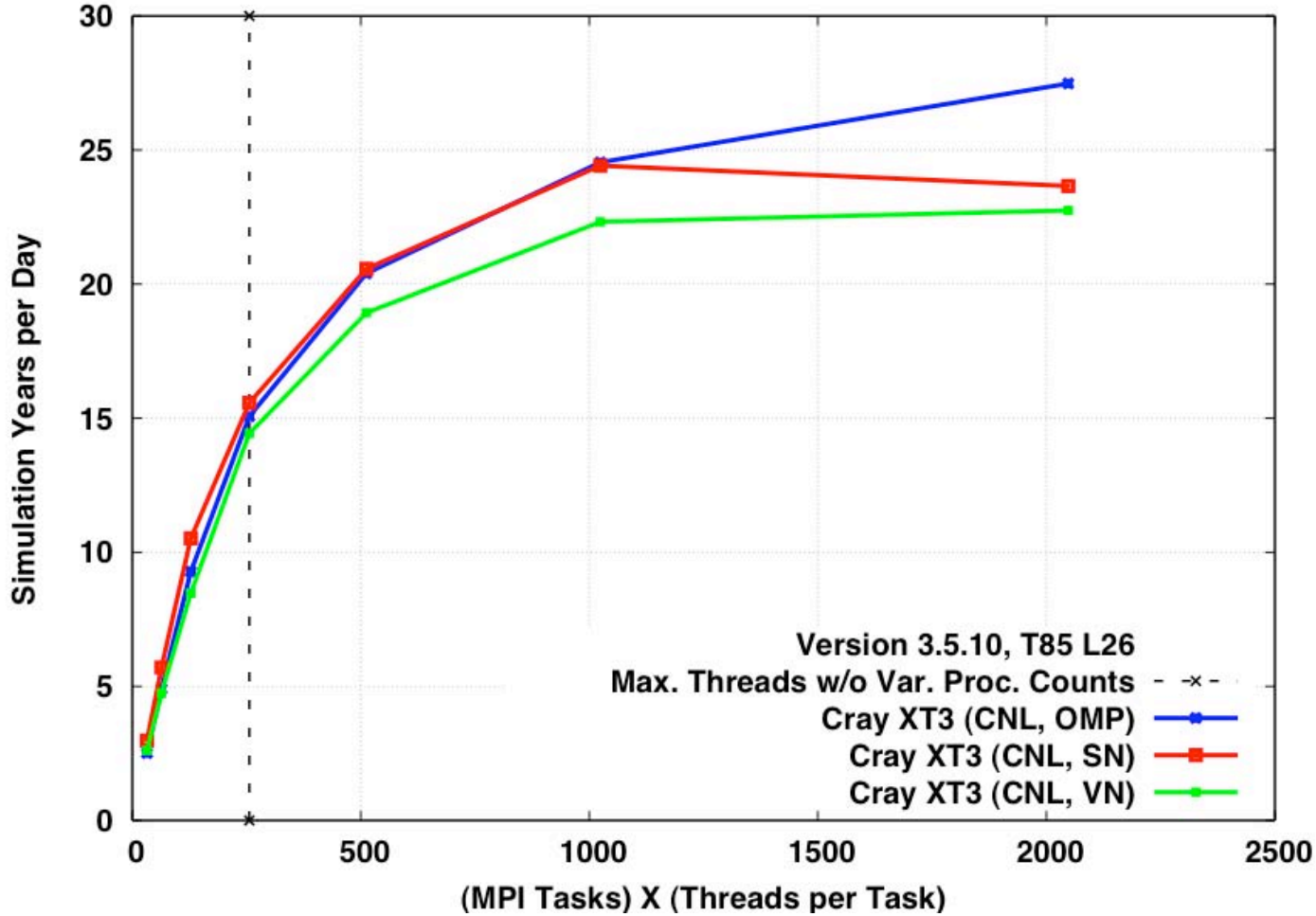
- Results from Cray XT4 running Catamount operating system and from Cray XT3 running Compute Node Linux (CNL) operating system.
  - Both XT4 and XT3 use a 2.6 GHz dual-core AMD Opteron processor. XT4 processor-memory bandwidth is 1.6 times that of XT3. XT4 processor-network bandwidth is 1.5-1.8 times that of the XT3.
  - CNL supports OpenMP; Catamount does not.
  - Experiments run with only one processor core active, leaving the other idle (SN execution mode), using both cores with two MPI tasks (VN mode), and using both cores with one MPI task and two OpenMP threads per task (OMP mode).
- Benchmark data for cam3_5_10 (with an additional optimization to a global mean calculation in the physics) using
  - Finite Volume dycore and 1.9x2.5 degree resolution (96x144 horizontal grid) with 26 vertical levels. Maximum number of MPI tasks without variable process counts: 256.
  - Spectral Eulerian dycore and T85 resolution (128x256 horizontal grid) with 26 vertical levels. Maximum number of MPI tasks without variable process counts: 128.

Spectral Eulerian Performance



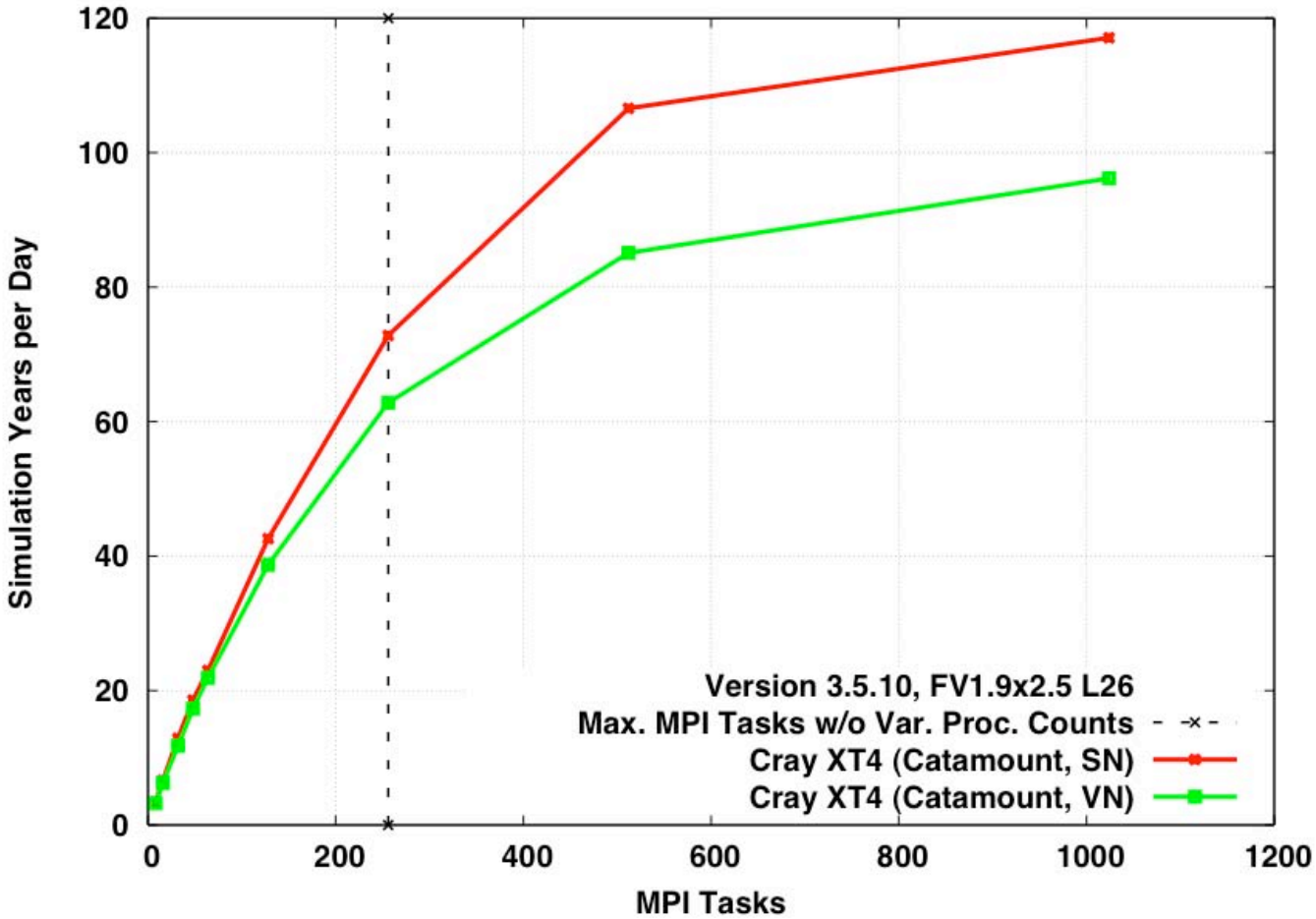
For T85L26, cam3_5_10 can use over 1000 MPI tasks on the XT4, greatly exceeding the prior algorithmic limit of 128 tasks. VN performance is also very close to SN performance for the same task count, indicating that contention between the cores is not an issue.

OpenMP and Spectral Dycore



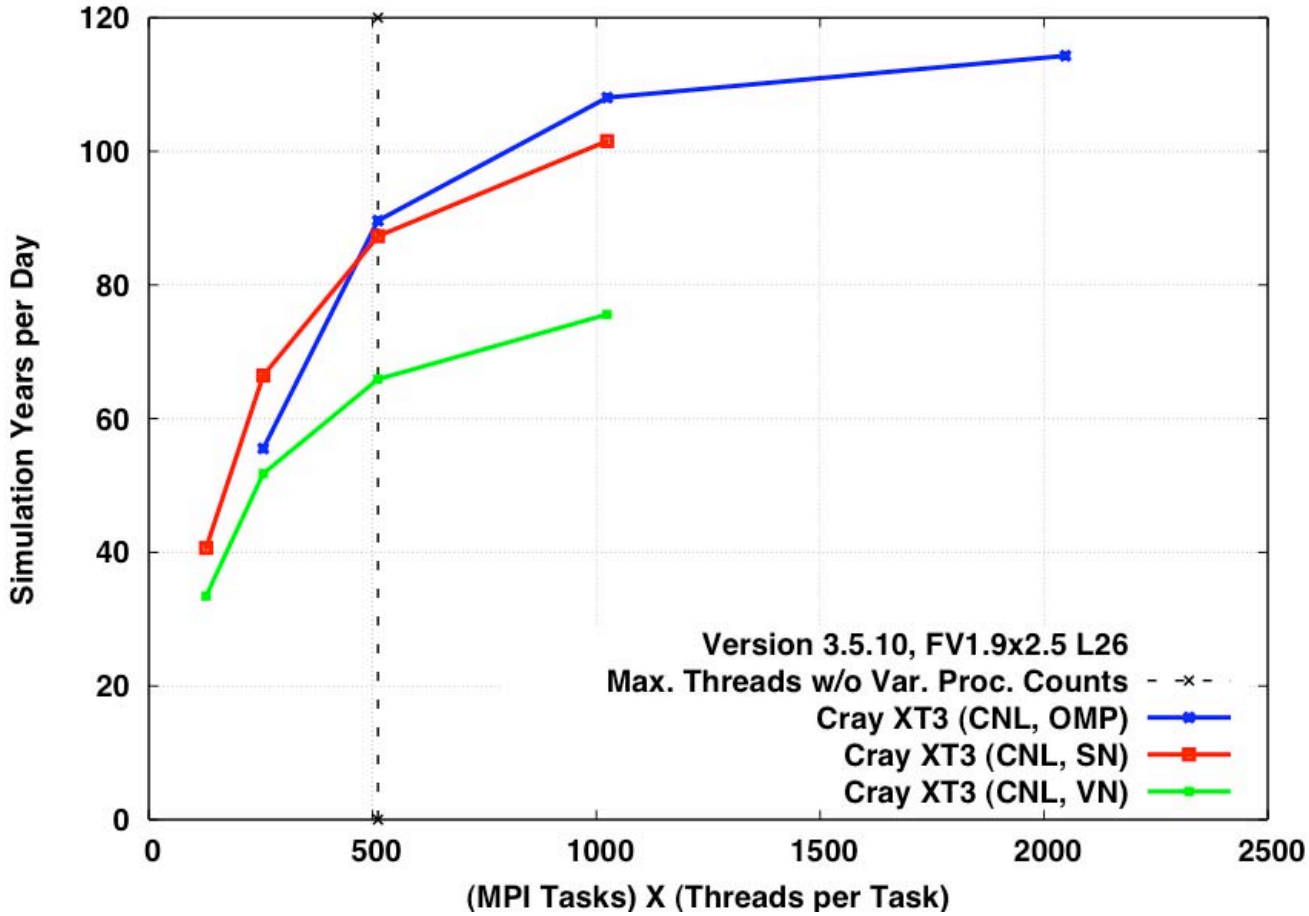
Preliminary data from XT3 system with CNL operating system. For T85L26, the previous limit on total thread count was 128 x 2 = 256. We are now able to use over 2000 threads. Note that with OpenMP we achieve SN-level performance without wasting cores.

Finite Volume Performance: 2 Degree



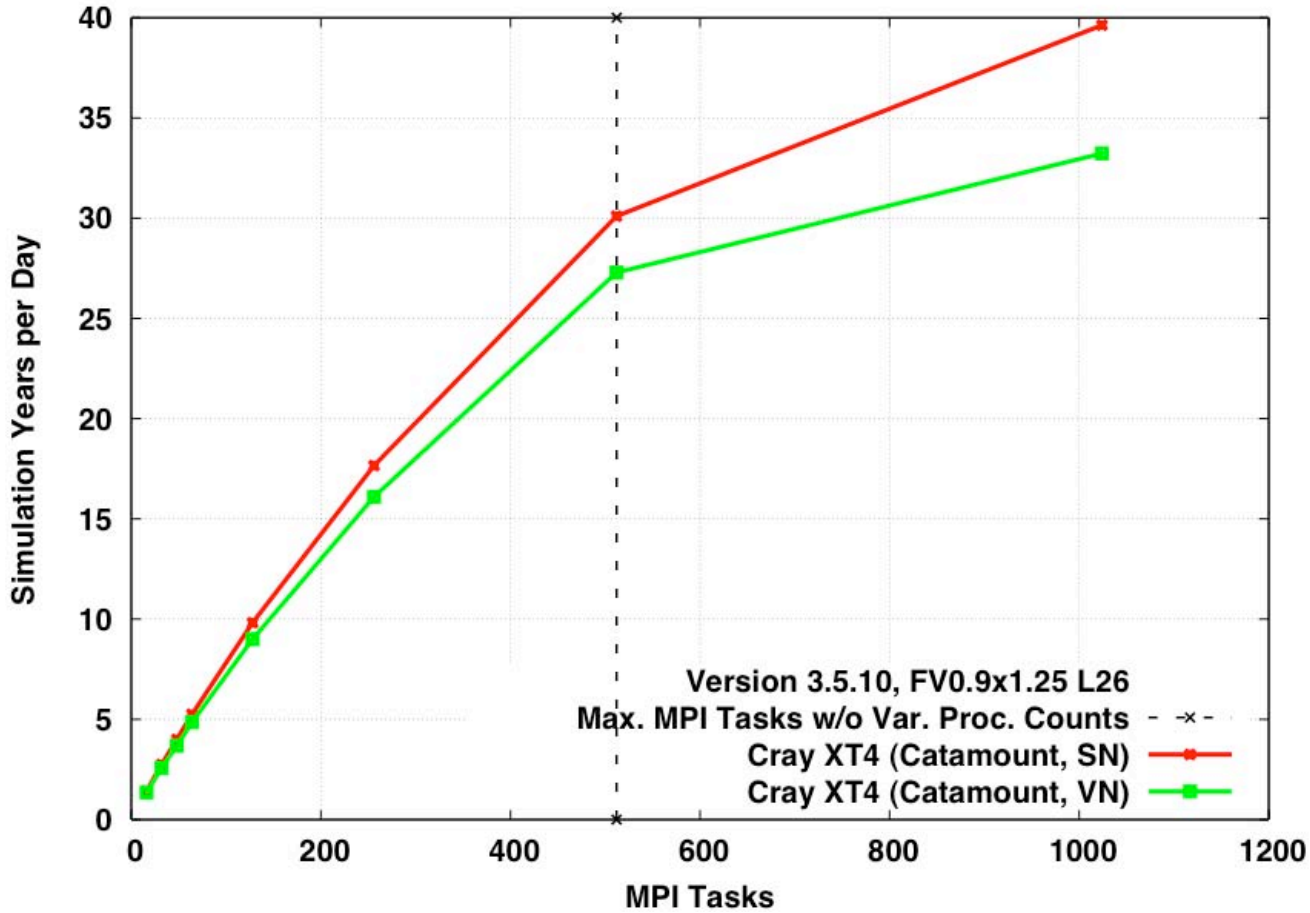
For 1.9x2.5 degree resolution, cam3_5_10 can use 1024 MPI tasks on the XT4, exceeding the original 256 task limit. In this experiment, SN mode was more efficient than VN mode. The *processor stride* runtime option should eliminate most of the performance degradation from using both cores (VN mode), as shown in the spectral dycore benchmarks.

OpenMP and Finite Volume Dycore



Preliminary data from XT3 system with CNL operating system. For 1.9x2.5 degree resolution, the previous limit on total thread count was 236x2 = 512. We are now able to use over 2000 threads. Here OpenMP performs better than using only one of the processor cores (SN mode) for large thread counts, probably reflecting impact of using fewer MPI tasks for the same thread count.

Finite Volume Performance: 1 Degree



For 0.9x1.25 degree resolution, cam3_5_10 can use (at least) 1024 MPI tasks on the XT4, exceeding the original 512 task limit. The performance advantage of doubling the number of tasks (32%) is less here than for the 2 degree benchmark (46%), primarily because physics is a smaller percentage of the runtime when using FV at this resolution and the extra tasks are used most efficiently in the physics.

Summary

- Initial results are promising.
  - Supporting different numbers of active MPI processors in different phases is effective in increasing CAM scalability.
  - OpenMP works well with variable process count, improving scalability even further.
  - This approach to increasing algorithmic scalability will become even more important as new science capabilities are added, for example, when adding atmospheric chemistry or cloud resolving submodels.
- More work needs to be done.
  - Finish FV work plan.
  - Continue performance evaluation, including using 1.0, 0.5, and 0.25 degree resolution grids and other computer systems.
  - Continue evaluation of OpenMP performance.
  - 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 contracts W-7405-Eng-48 and DE-AC52-07NA27344 with 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.