Performance and Scalability of the Community Atmosphere Model

Patrick H. Worley
Oak Ridge National Laboratory

The 8th International Workshop on Next Generation Climate Models for Advanced High Performance Computing Facilities

February 23-25, 2006

Hilton Albuquerque

Albuquerque, New Mexico

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.
- These slides have been authored by a contractor of the U.S. Government under contract No. DE-AC05-00OR22725. 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
- Oak Ridge National Laboratory is managed by UT-Battelle, LLC for the United States Department of Energy under Contract No. DE-AC05-00OR22725.

Collaborators

- Arthur A. Mirin (Lawrence Livermore National Laboratory)
- Michael Wehner (Lawrence Berkeley National Laboratory)
- Leonid Oliker (Lawrence Berkeley National Laboratory)
- Dave Parks (NEC)
- Trey White (ORNL)
- Nathan Wichmann (Cray)

Community Atmosphere Model (CAM)

Atmospheric global circulation model

- 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:
  - Spectral Eulerian (EUL) dynamical core (*dycore*)
  - Spectral semi-Lagrangian (SLD) dycore
  - Finite-Volume semi-Lagrangian (FV) dycore

  all using tensor product *longitude* x *latitude* x *vertical level* grid over the sphere, but not same grid, same placement of variables on grid, or same domain decomposition in parallel implementation.
- Separate data structures for dynamics and physics and explicit data movement between them each timestep (in a "coupler")
- Developed at NCAR, with contributions from DOE and NASA
- http://www.ccsm.ucar.edu/models/atm-cam/

Research Goals

- Create baseline performance characterization for the Community Atmosphere Model (CAM) for both current and future problem resolutions.
- Track changes in CAM performance as the model evolves.
- Predict or identify scalability (or other performance) problems early in development process.

Research Tasks

1. Scaling Studies with FV Dycore
   - 1 x 1.25 degree horizontal resolution (C grid)
   - 0.5 x 0.625 degree horizontal resolution (D grid)
   - 0.25 x 0.3125 degree horizontal resolution (E grid)
2. Scaling Studies with EUL Dycore
   - T42L26
   - T85L26
   - T170L26

   On Cray X1E, Cray XT3, Earth Simulator, IBM POWER5 cluster, NEC SX-8, SGI Altix, …(and a number of other Opteron and Itanium2 clusters)
3. Construct a CAM "Performance Model"
4. Repeat / Update 1-3 as add atmospheric chemistry and other new physical processes, and as add additional parallelism into physics.

Status of Research

1. Scaling Studies with FV Dycore
   1. 1 x 1.25 degree horizontal resolution (C grid)
   2. 0.5 x 0.625 degree horizontal resolution (D grid)
   3. 0.25 x 0.3125 degree horizontal resolution (E grid)
2. Scaling Studies with EUL Dycore
   - T42L26
   - T85L26
   - T170L26
3. Construct CAM Performance Model
   - Collecting data on physics, verifying assumptions about effect of diurnal and seasonal cycles.
   - Collecting data on message traffic in FV dycore.

CAM Performance Portability Goals

1) Maximize single processor performance, e.g.
   a) Optimize memory access patterns
   b) Maximize vectorization or other fine-grain parallelism
2) Minimize parallel overhead, e.g.
   a) Minimize communication costs
   b) Minimize load imbalance
   c) Minimize redundant computation

for

- a range of target systems,
- a range of problem specifications (grid size, physical processes, …)
- a range of processor counts

while preserving maintainability and extensibility.

No optimal solution for all desired (platform,problem,processor count) specifications. Approach: compile-time and runtime optimization options.

CAM Performance Optimization Options

- Physics data structures
  - Index range, dimension declaration
- Physics load balance
  - Variety of load balancing options, with different communication overheads
  - SMP-aware load balancing options
- Communication options
  - MPI protocols (two-sided and one-sided)
  - Co-Array Fortran
  - SHMEM protocols

  and pt-2-pt implementations or collective communication operators
- OpenMP parallelism
  - Instead of some MPI parallelism
  - In addition to MPI parallelism
- Aspect ratio of dynamics 2D domain decomposition (FV-only)

Physics Data Structures

- All three dynamics use a tensor product longitude-vertical-latitude (plon x pver x plat) computational grid covering the sphere.
- A *vertical column* is a set of grid points of coordinates (i,*,j). In current physics, computation is independent between vertical columns, and tightly coupled within a vertical column.
- The basic data structure in the physics is the *chunk*, an arbitrary collection of vertical columns. Grid points in a chunk are referenced by (local column index, vertical index).
- Define

  ncols(j): number of columns allocated to chunk j

  nchunks: number of chunks

  begchunk:endchunk: chunk indices assigned to a given process

  pcols: maximum number of columns allocated to any chunk

  pcols and pver specified at compile time.
- Arrays declared as (pcols, pver, begchunk:endchunk).

Physics Computational Structure

- Loops structured as

```
do j=begchunk,endchunk
   do k=1,pver
      do i=1,ncols(j)
         (physical parameterizations)
      enddo
   enddo
enddo
```

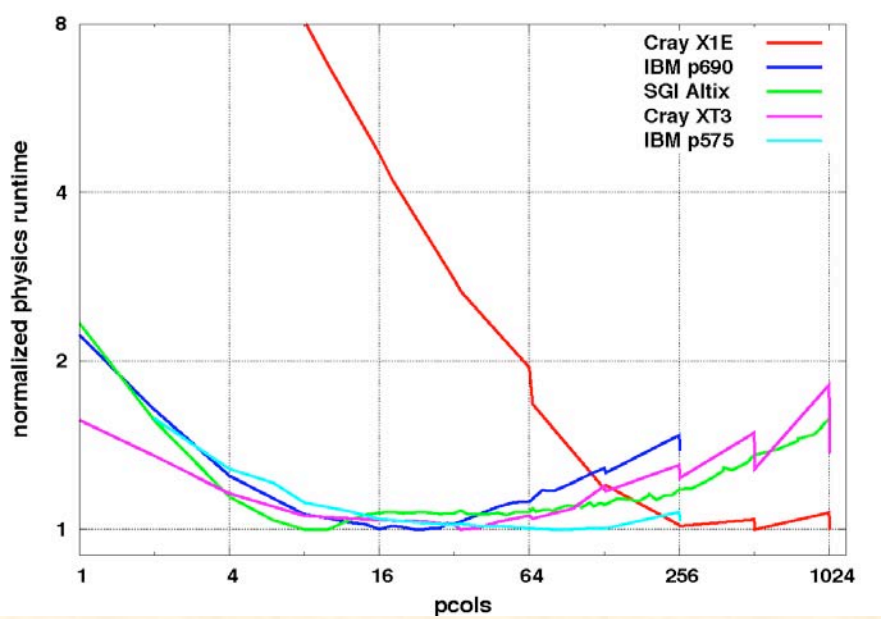
- Inner loop over columns is vectorizable.
- Coarser grain parallelism is exploited over outer loop over chunks, OpenMP can be used to parallize j loop more.
- Length of inner loop can be adjusted for size of cache or for vector length.
- Columns can be assigned to chunk in order to balance load between chunks or to minimize communication cost of coupling between dynamics and physics.

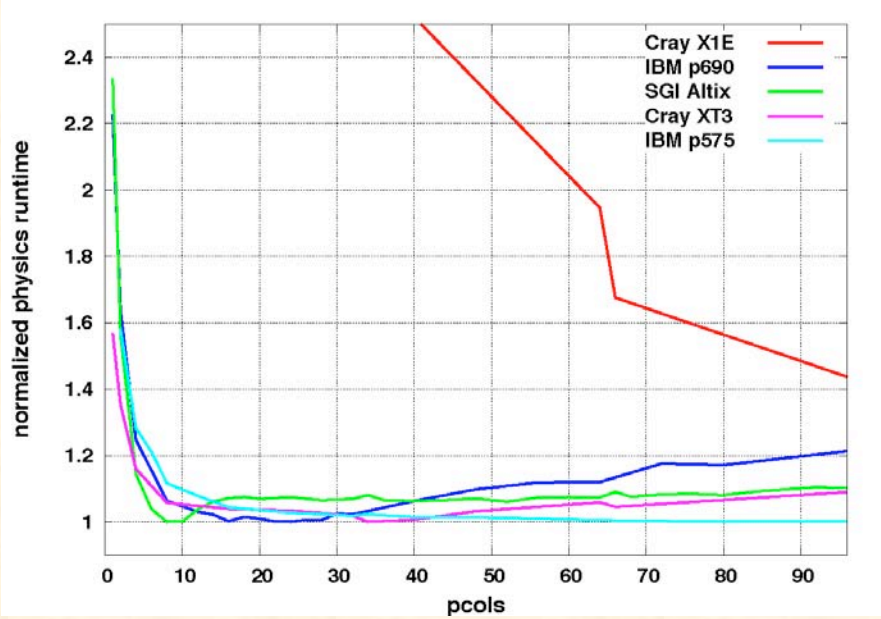
pcols Tuning Experiments

- "Single" processor performance tuning, using MPI only and
  - all processors in SMP node, to include impact of memory contention, or
  - two processors for systems with single processor nodes

  solving problems with
  - 2048 columns (26 vertical levels, 64 x 32 horizontal grid), so 1024 columns per processor in two processor experiments, or
  - 32768 columns (26 vertical levels, 256 x 128 horizontal grid)
- Columns assigned to chunks both "in order" and to balance load between chunks. (Results similar in both cases.)
- CAM executed for one simulation day and for two simulation days. Difference examined to check for atypical start-up costs.
- Varied `pcols` (which necessarily varied `ncols(j)`).
- Physics-only execution times for all processes summed, and results for each `pcols` value normalized with respect to minimum observed time over all experiments for a given platform.

pcols Tuning Experiments





Altix: minimum at pcols = 8

p690: minimum at pcols = 24

XT3: minimum at pcols = 34

p575 : minimum at pcols = 80

X1E : minimum at pcols = 514 or 1026

pcols <= 4 bad for all systems.

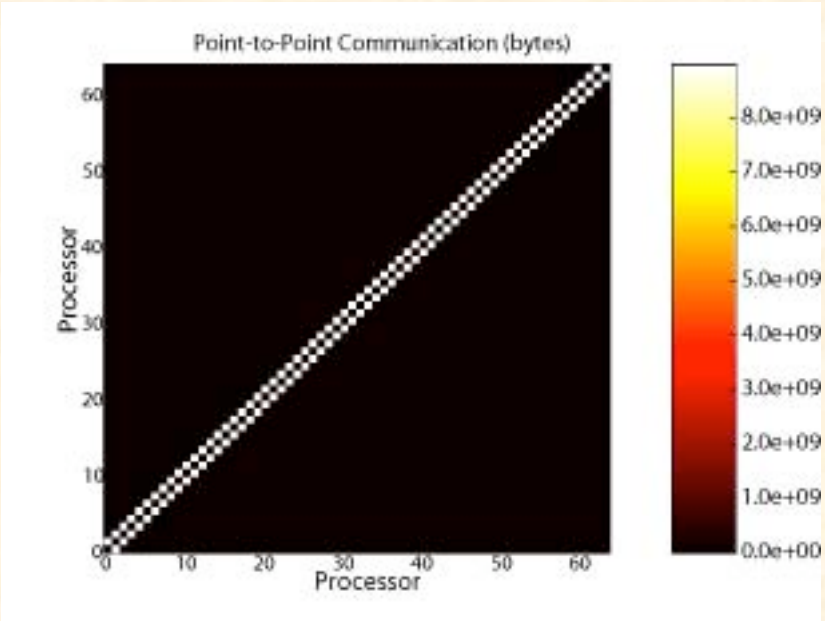
1D vs. 2D Decompositions

- Parallel performance as a function of processor count for different domain decompositions for finite volume dycore:
  - 1D over latitude
  - 2D, defined by a (P/4)x4 virtual processor grid
  - 2D, defined by a (P/7)x7 virtual processor grid

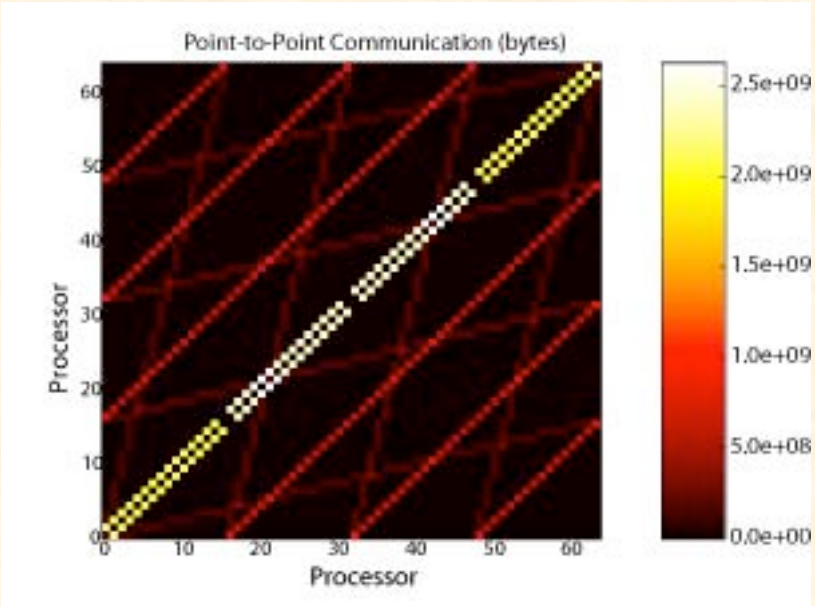
  where first dimension in the processor grid decomposes latitude dimension
- Total runtime for a typical simulation day, measured on
  - Cray X1E
  - Cray XT3
  - IBM p690 cluster

  and used to calculate computation rates.
- Performance results optimized over communication protocol, `pcols`, load balancing, and number of OpenMP threads (separately for each data point).

Communication: Finite Volume Dycore



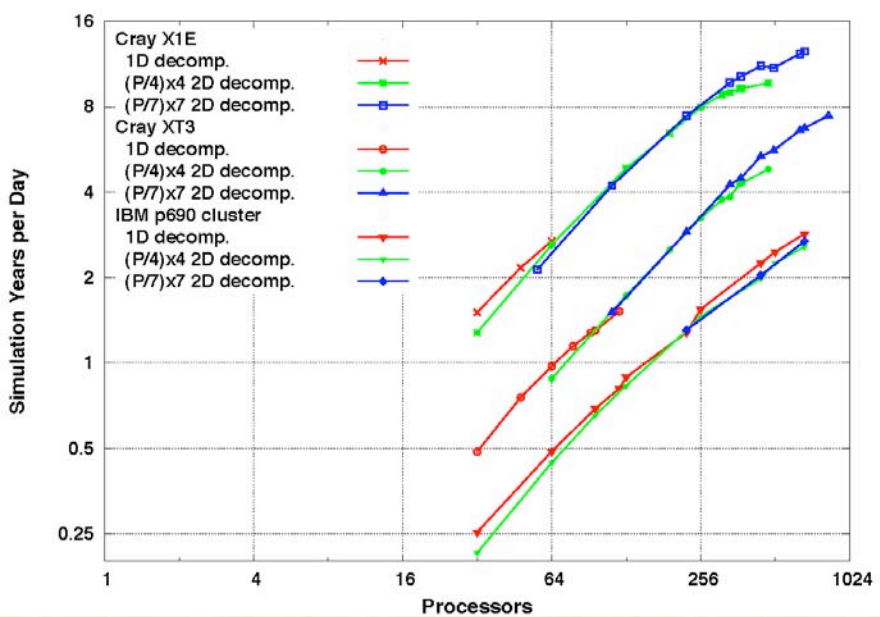
1D Decomposition
64x1

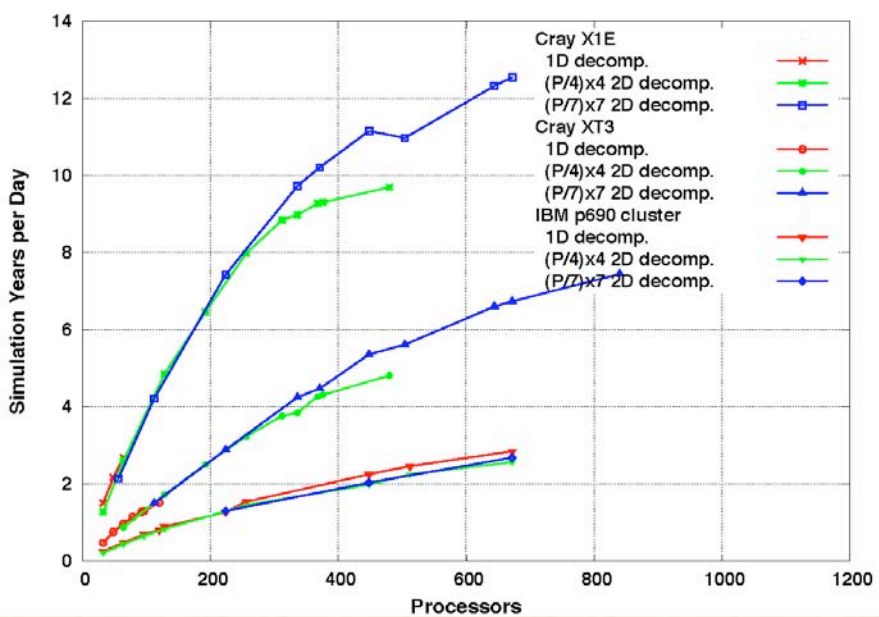


2D Decomposition
16x4

(from L. Oliker, et al, *Leading Computational Methods on Scalar and Vector HEC Platforms,* in SC05 Proceedings.)

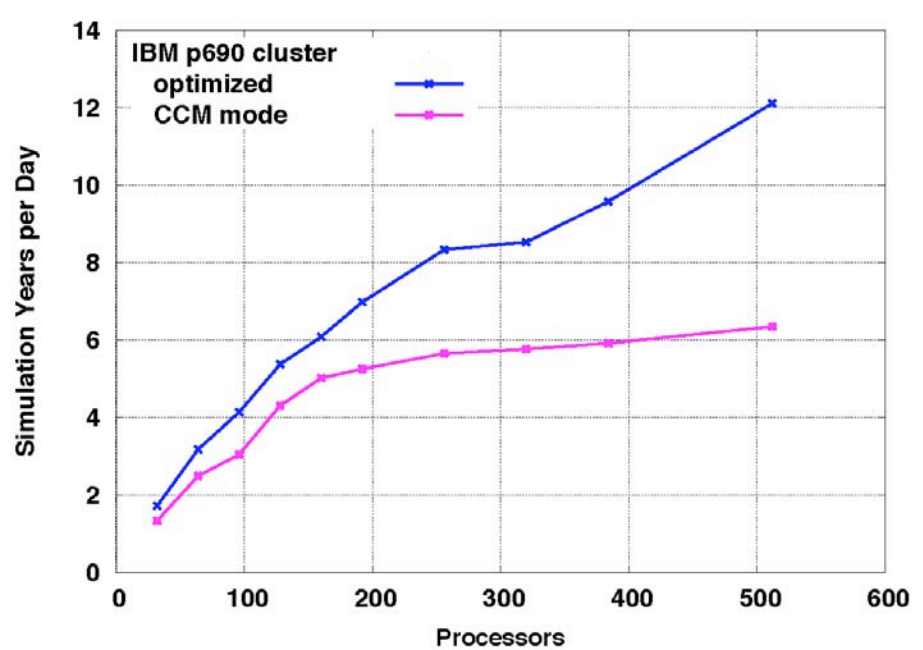
1D vs. 2D: Finite Volume Dycore

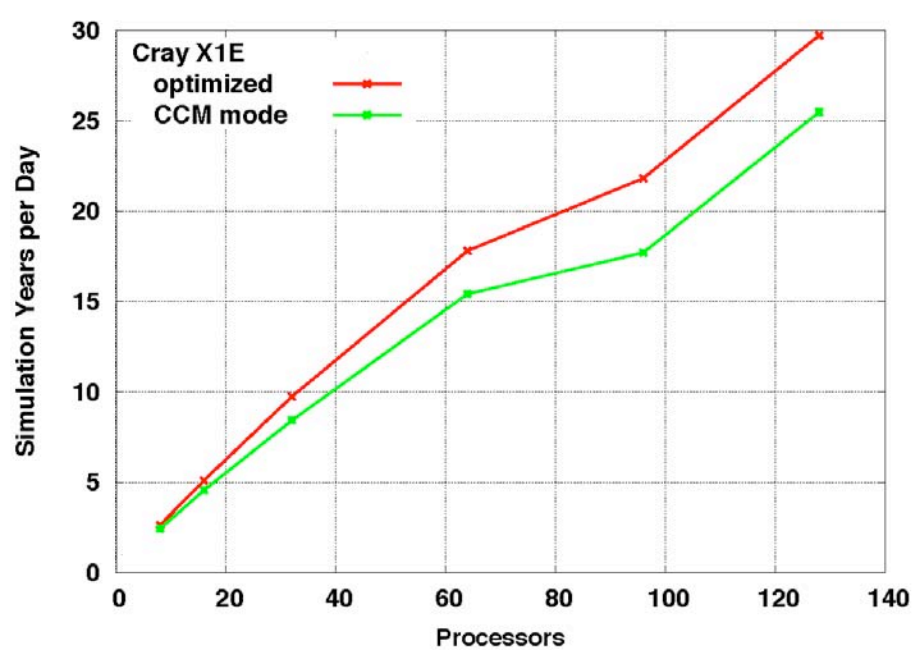




- 2D decompositions become superior to 1D when number of MPI processes used to decompose latitude dimension in 1D exceeds some limit, ~70 on two Cray systems (not using OpenMP). Similarly, (P/7)x7 decomposition begins to outperform (P/4)x4 when the number of MPI processes decomposing latitude in (P/4)x4 exceeds ~70.
- On IBM p690, using OpenMP and the 1D decomposition is superior to using the 2D decomposition up to 672 processors, even when the optimum for 1D uses more threads per process than the optimum for 2D. Note 1D never uses more than 84 MPI processes.

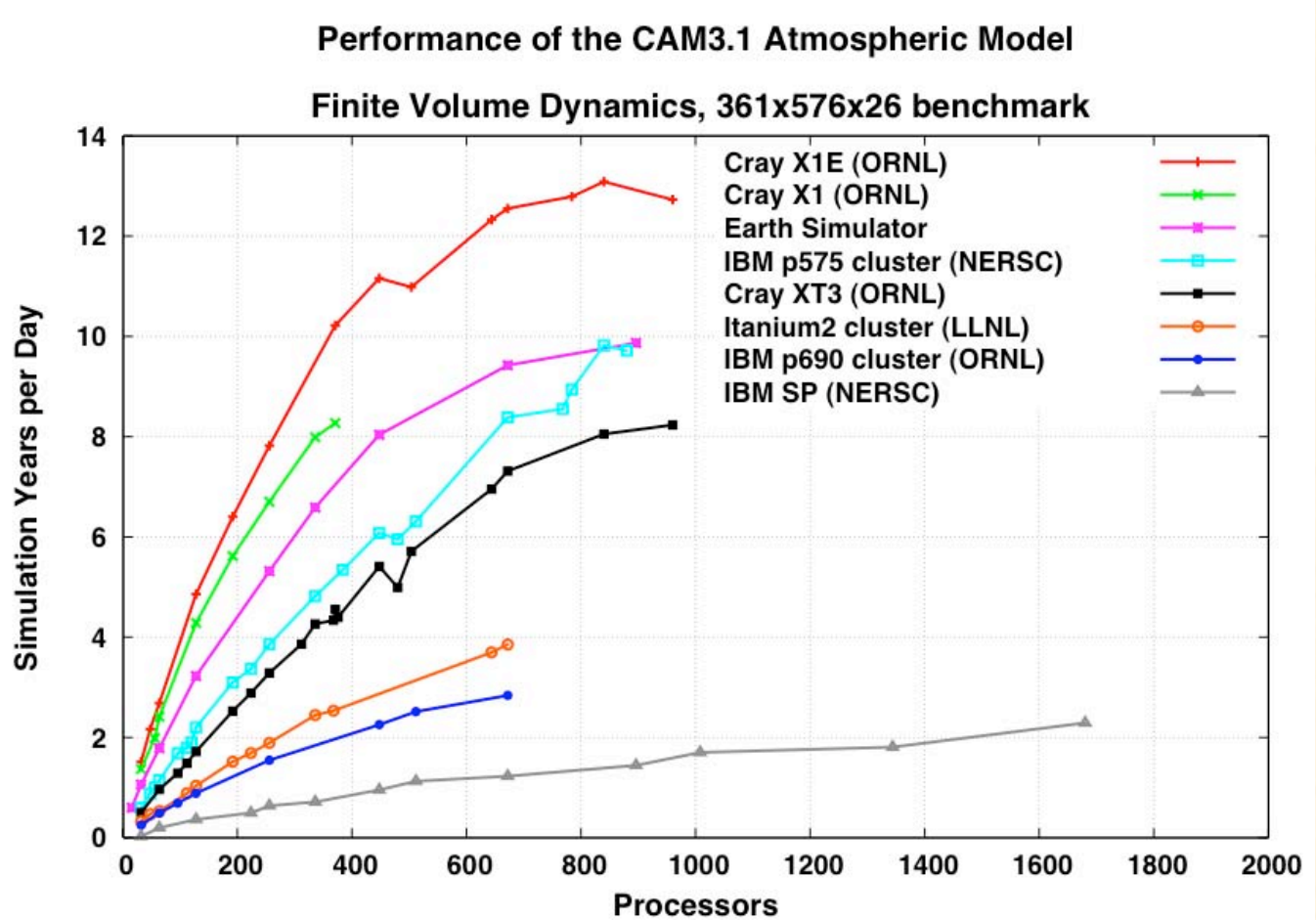
Tuning Impact: EUL Dycore / T85L26





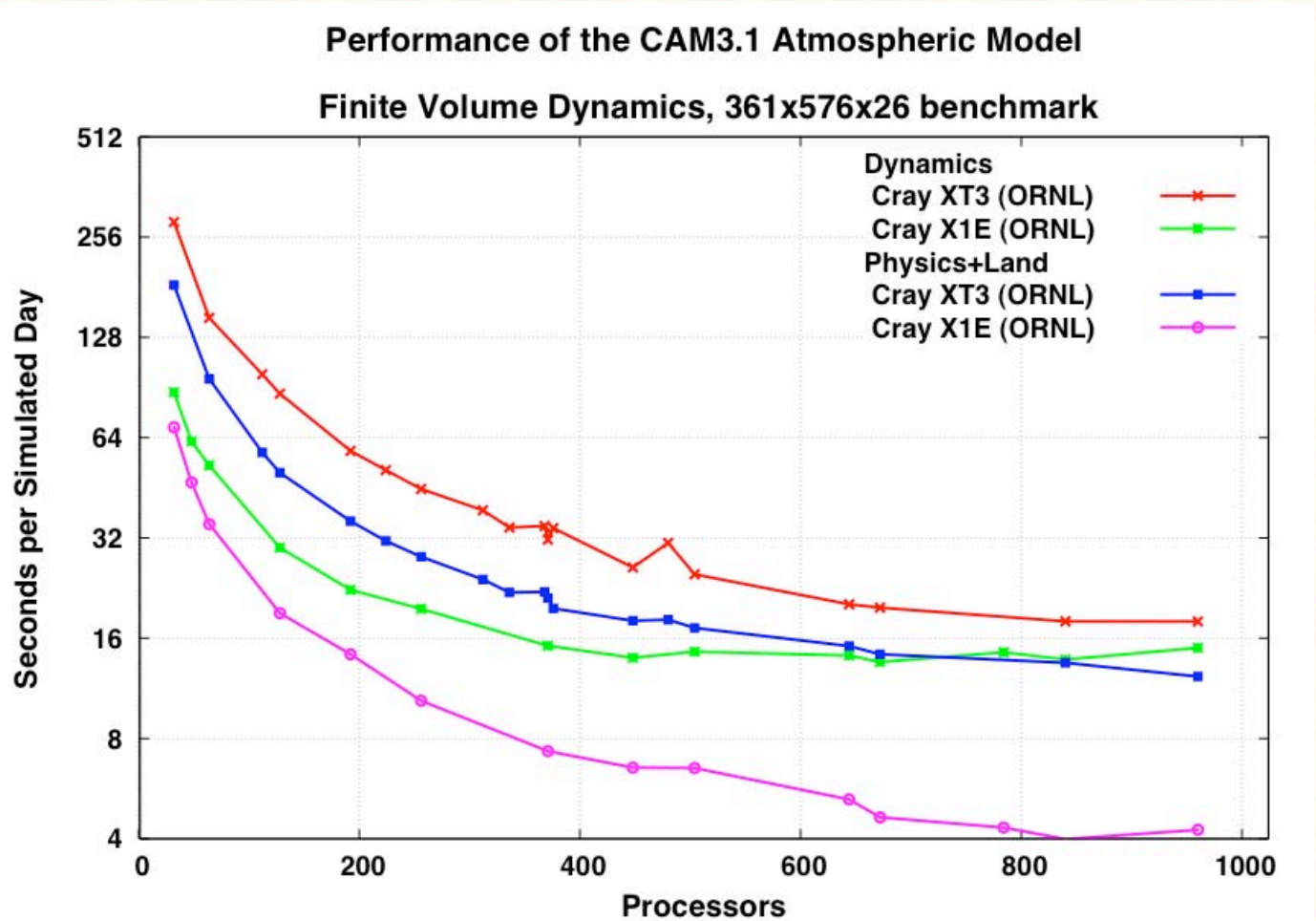
- Big win on IBM, primarily due to ability to use OpenMP parallelism in physics when dynamics parallelism exhausted (at approximately 128 processors).
- As vector length for CCM mode is near optimal on the X1E, performance difference is primarily load balancing.

CAM Platform Comparisons: FV Dycore



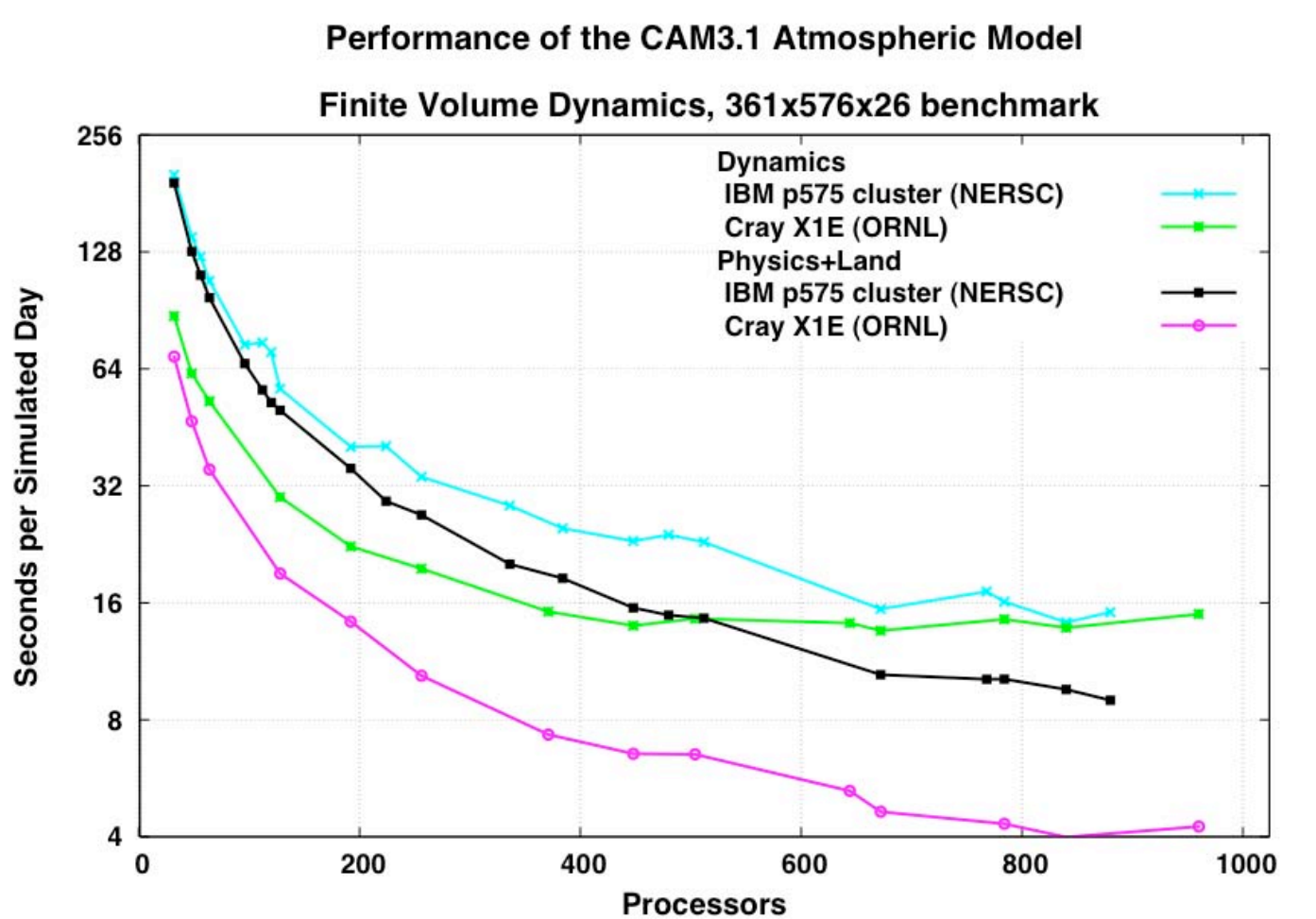
- Earth Simulator results courtesy of D. Parks. LLNL results courtesy of A. Mirin. SP results courtesy of M. Wehner. Maximum number of MPI processes is 960. IBM systems and Earth Simulator use OpenMP to increase scalability.

CAM Performance Diagnosis



- Dynamics and physics scaling very similarly on XT3.
- On X1E, physics scaling much better. (Dynamics best performance at 448 processors.)

CAM Performance Diagnosis



- At large scale, dynamics performance on p575 similar to performance on X1E. However, p575 uses OpenMP and X1E does not in these experiments. P575 using a coarser domain decomposition.

Known Limits to Scalability

1. Requirement that at least 3 latitudes and 3 vertical levels be present in each “block” of domain decomposition within FV dycore. For D grid with 26 vertical levels, limit is 120 x 8 processor grid, or 960 MPI processes.
2. Polar filter introduces load imbalances, especially on vector systems (because the small number of short FFTs do not vectorize well).
3. Physical parameterizations can use many more processors, but is currently limited to the same number of MPI processes as the dynamical core. OpenMP can be used to assign more processors to physics than to dynamics, mitigating this to some degree. There is also some OpenMP parallelism available within the dynamics.
4. On vector systems, additional parallelism in physics is of limited utility, as vector length drop below 220 for more than 960 processors (and drops below 110 in radiation routines).

Proposed Solutions and New Issues

1. Dynamics Scaling?
   - Using altenative (cubed sphere?) computational grid.
2. Physics Scaling?
   - Allow different number of MPI processes in dynamics and in physics (generalizing current OpenMP approach to pure MPI codes).
3. Introduction of atmospheric chemistry will add many new tracers, adding work to both dynamics and to physics.
   - Parallelize advection over species?
   - Parallelize chemistry over species?
   - Use 3D decomposition for chemistry?
4. Increasing model resolution will exacerbate the current I/O bottlenecks and memory impact of the (few) remaining global arrays.
   - Parallel I/O will allow both of these problems to be addressed, hopefully successfully.

Source Material

P. Worley and J. Drake, *Performance Portability in the Physical Parameterizations of the Community Atmospheric Model*, International Journal for High Performance Computer Applications, 19(3), August 2005, pp. 187-201.

A. Mirin and W. Sawyer, *A Scalable Implementation of a Finite-Volume Dynamical Core in the Community Atmosphere Model*, International Journal for High Performance Computer Applications, 19(3), August 2005, pp. 203-212.

W. Putman, S-J. Lin, and B-W. Shen, *Cross-Platform Performance of a Portable Communication Module and the NASA Finite Volume General Circulation Model*, International Journal for High Performance Computer Applications, 19(3), August 2005, pp. 213-223.

L. Oliker, J. Carter, M. Wehner, A. Canning, S. Ethier, A. Mirin, G. Bala, D. Parks, P. Worley, S. Kitawaki, and Y. Tsuda, *Leading Computational Methods on Scalar and Vector HEC Platforms*, in Proceedings of the ACM/IEEE International Conference for High Performance Computing, Networking and Storage (SC05), Seattle, WA, November 12-18, 2005.