Extending Scalability of the Community Atmosphere Model

Art Mirin

Lawrence Livermore National Laboratory

Pat Worley

Oak Ridge National Laboratory

12th Annual CCSM Workshop

June 19-21, 2007

The Village at Breckenridge

Breckenridge, CO

The Community Atmosphere Model (CAM) is a global circulation model originating at NCAR

- The timestep has two primary phases
  - dynamics: evolutionary equations for atmospheric flow
  - physics: columnar processes (typically parameterized) such as precipitation, cloud physics, radiation, turbulent mixing
- There are several choices of dynamical core (dycore); the two most common are
  - finite-volume semi-Lagrangian (FV)
  - spectral Eulerian (EUL)
- The grids used by the FV and EUL dycores are logically rectangular in latitude / longitude / vertical
- Dynamics and physics have separate data structures with explicit data movement between them each timestep

The parallelization strategy uses multiple domain decompositions

- Dynamics and physics use separate decompositions
  - physics utilizes a 2D longitude/latitude decomposition
  - dynamics utilizes multiple decompositions
    - FV dynamics: 2D block latitude/vertical and 2D block longitude/latitude
    - EUL dynamics: 1D latitude in physical space and 1D wavenumber in spectral space
- Decompositions are joined with transposes
- Each subdomain is assigned to at most one MPI task
- Additional parallelism can be exploited through OpenMP

OAK RIDGE National Laboratory

Several factors limit the scalability of CAM

- Number of processes is limited to the number of subdomains
- Number of subdomains is limited by resolution
  - climate simulations have relatively modest resolution
- FV dynamics: small vertical dimension and vertical coupling through geopotential severely limit number of subdomains in latitude/vertical decomposition
- EUL dynamics: decompositions are one-dimensional
- For virtually all decompositions each MPI task must be assigned to at least one subdomain
  - parallelism is constrained by decomposition allowing the smallest number of subdomains

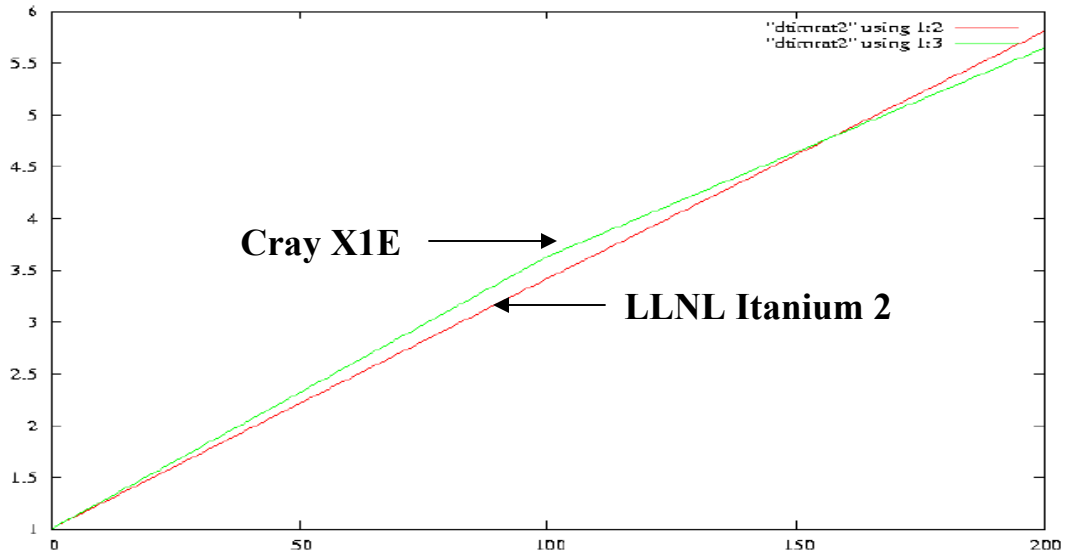
OAK RIDGE National Laboratory

Upcoming climate calculations will be more challenging

- Resolution of 1-deg (288x192 grid), 0.5-deg (576x384 grid), or even 0.25-deg (1152x768 grid) for FV dynamics
- Inclusion of atmospheric chemistry
  - up to a hundred or possibly several hundred chemical constituents (tracers)
  - chemical constituents must be advected
- Inclusion of cloud resolving physics

OAK RIDGE National Laboratory

Advecting tracers can be expensive



Computer time with N tracers normalized by computer time with 1 tracer; each tracer adds just over 2% to the overall run time; calculation takes over 3 times as long with 100 tracers

Variable process count allows different phases of FV computation to achieve their own degree of scalability

- Lagrangian remap (which is columnar) can use much finer decomposition than main FV dynamics
  - limited mainly by cache effects degrading single processor performance for very small subdomains
- Physics (which is columnar) can use much finer decomposition than main dynamics
  - limited mainly by cache effects and to lesser extent by load imbalance
- Tracer advection
  - admits finer vertical decomposition (versus dynamics) since it does not couple vertically
  - can be decomposed over tracer index
  - can be partially overlapped with main dynamics

OAK RIDGE National Laboratory

Variable process count facilitates scalability (cont.)

- Portions of the atmospheric chemistry do not couple vertically and can be decomposed vertically as well as horizontally
- Cloud resolving physics uses much higher resolution in the physics and can therefore utilize many more subdomains

We propose a versatile approach to variable process count for FV dynamics

- Allow the latitude/vertical dynamics decomposition to be smaller than the longitude/latitude dynamics decomposition
- Allow the longitude/latitude dynamics decomposition to have a different number of subdomains than the physics decomposition
- Allow the existence of auxiliary processes that can be employed for alternate decompositions as needed
  - decomposition over tracers during advection
  - finer vertical decomposition for tracer advection
  - overlap of tracer advection and main dynamics (as part of subcycling algorithm)
  - 3-D chemistry decomposition
  - cloud-resolving physics

CAM at 0.5-degree can benefit from variable process count

- Mesh resolution is 576x384x26; assume 100 tracers
- Consider latitude/vertical dynamics decomposition of 96x7
  - —4 latitudes and 4 levels per subdomain
- Consider longitude/latitude dynamics decomposition of 144x96
  - —4 longitudes and 4 latitudes per subdomain
- Consider physics decomposition of 13824 (=144x96)
  - — 16 columns per subdomain
- Decompose tracers into 20 groups for purposes of advection (96x7x20 = 13440 processes)
- Compared to original 96x7 decomposition, we can use additional processes in remap, tracer advection, and physics

We are modifying CAM (FV dycore) for variable process count in several steps

- Allow auxiliary processes (*COMPLETE*)
- Allow longitude/latitude dynamics decomposition to be larger than latitude/vertical decomposition (*COMPLETE*)
- Allow longitude/latitude dynamics and physics decompositions to be of different sizes (*IN PROGRESS*)
- Decompose tracer advection with respect to tracer index (latitude/vertical/tracer decomposition)
- Consider finer vertical decomposition for tracer advection (vs. main dynamics)
- Consider overlap of tracer advection (n^{th} tracer subcycle) with main dynamics (corresponding to $n+1^{st}$ tracer subcycle)

Auxiliary process capability is enabled with minimal code changes

- npes and mpicom continue to refer to the main process set; npes_tot and mpicom_tot refer to the total process set
- npes_xy and mpicom_xy refer to the FV longitude/latitude dynamics decomposition; they currently are equal to npes and mpicom
- Conditional tests are performed to determine if a process is primary or auxiliary
- Code can still be run in familiar mode (no auxiliary processes)
- Code successfully executes with inactive auxiliary processes and gives bit-for-bit agreement with case having no auxiliary processes
  - —see branch *varproc_cam3_3_47*

We encountered several issues in auxiliary process implementation

- We added communicators (when not already present) to *pilgrim* and *mod_comm* subroutine argument lists
  - this is absolutely necessary for handling barrier calls
- We referenced communicators appropriate to the task
  - necessary for transposes and barrier calls
- We had to decide when to conditionally execute certain coding blocks or execute them with trivial values
  - auxiliary processes do not map onto any subdomain
  - execution (versus conditional test) does not necessarily waste resources and can make coding simpler
- We inactivated auxiliary processes for the ocean and ice

CAM (FV) now works with a larger longitude/latitude dynamics decomposition

- npes_yz and mpicom_yz refer to the latitude/vertical dynamics decomposition
  - npes_yz does not exceed npes_xy
  - mpicom_yz is a subcommunicator of mpicom_xy and includes the lowest npes_yz processes
- The biggest challenge was to generalize the dynamics transposes to connect different-sized decompositions
  - modified *pilgrim* (mainly interdecomposition patterns) and *mod_comm* (mainly irregular routines)
  - transpose routines (mp_sendirr, mp_recvirr) called with mpicom_xy
- CAM (unmodified) gives bit-for-bit agreement with runs using a larger longitude/latitude dynamics decomposition

Capability to use different sized lon/lat (FV) dynamics and physics decompositions is in progress

- Dynamics and physics already have distinct variables (npes_xy and npes, resp.) delineating active processes
- Transposes connecting dynamics and physics are already in place (to enable physics load-balancing schemes)
  - need to generalize *pilgrim*-based transposes to different-sized decompositions; non-*pilgrim* transpose options already handle this situation.
- The number of meaningful history variables will not be the same for all processes
  - dynamics variables won't be defined on extra physics processes (or vice-versa)

OAK RIDGE National Laboratory

Tracer advection will be decomposed with respect to tracer index

- For N-fold decomposition over tracers, associate auxiliary process P + K * M with regular latitude/vertical process P, where M is the size of the latitude/vertical dynamics decomposition and K varies from 1 to N-1
  - initialize P + K * M with corresponding subdomain information
  - the idea is to have N copies of the latitude/vertical dynamics decomposition
- Each advection subcycle requires new wind-related information
  - tracer-related information does not need to be fed back until end of tracer subcycling

Dynamics and advection can be partially overlapped

- The FV algorithm employs two levels of subcycling
  - tracer advection is subcycled with respect to physics advance (N-fold)
  - outer dynamics advance (corresponding to tracer advection) is subcycled with respect to tracer advection (M-fold)
- Dynamics advance does not depend on tracers
- Overlap n^{th} tracer subcycle with outer dynamics advance corresponding to $(n+1)^{st}$ tracer subcycle
- This approach is most relevant to high-resolution, where the number of tracer subcycles is the largest
- This provides opportunity to go beyond the limited dynamics scaling

Standard CAM with FV shows modest gains with improvements to date

- **Consider case at 1-deg (288x192) resolution with** phys_loadbalance=2
- **Largest reasonable latitude/vertical dynamics decomposition is 48x7**
  - **companion longitude/latitude decomposition is 7x48**
- **Variable process count allows us to run case with 48x7 latitude/vertical and 14x48 longitude/latitude decompositions (and a 672-way physics decomposition)**

The throughput of the 672-process case (vs. the 336-process case) is 22% greater

OAK RIDGE National Laboratory

Variable process count exposes scaling issues

- Case having 48x7 latitude/vertical and 35x48 longitude/latitude decompositions (1680 total processes) as compared to straight 48x7 case (336 processes) shows 30-fold increase in execution time for CLM restarts (*clm_driver_io*)
  - the gathering procedure scales poorly
- Check_energy diagnostic (gmean) shows small but non-pathological increase
  - in contrast, for 0.25-deg resolution, execution time of *gmean* increases 5-fold in going from 96x7 to 192x7 decomposition. More scalable implementation is being developed.

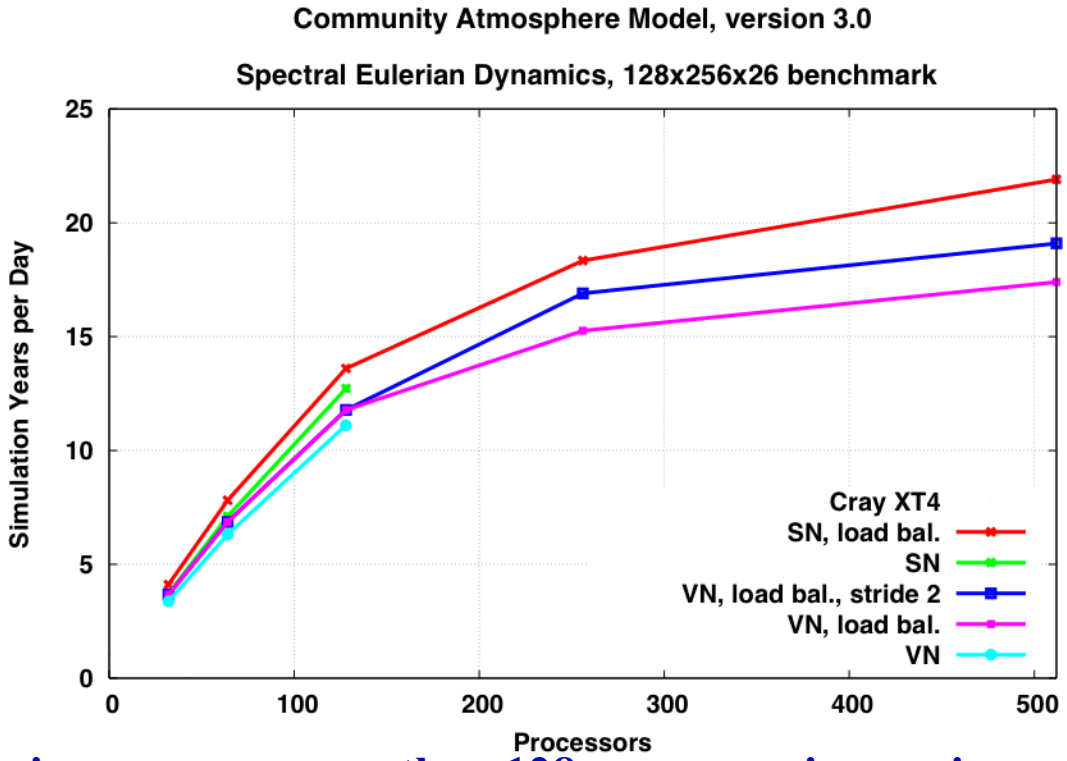
CAM with EUL dycore can now run with more physics than dynamics processes

- The EUL dycore is limited by a 1-D latitude decomposition
  - T85 can be run with at most 128 processes
- CAM has always been able to run with idle processes in spectral space (at most 86 active processes for T85)
- Straight-forward generalization allows assignment of zero work to dynamics processes in physical space computations
- Physics also modified to work when zero work assigned to physics processes
  - necessary when load balancing disabled because physics and dynamics domain decompositions are identical then
- When load balancing enabled, all physics processes are assigned work.

Choice of active EUL dycore processes can be used to optimize performance

- Number of active dynamics processes can be specified by a namelist parameter, but will otherwise be set to the total number of MPI processes or to the number of latitudes, whichever is smaller
- The default is to assign the first Q processes to be active, but a "stride" S can be specified at runtime to instead use processes {0,S,2S,…,P/S} where P is the total number of processes
  - this is useful when running on multicore processors where per core performance is enhanced if not all cores are active
- MPI communicator for active dynamics processes introduced to improve performance of transposes between physical and spectral spaces with EUL dycore

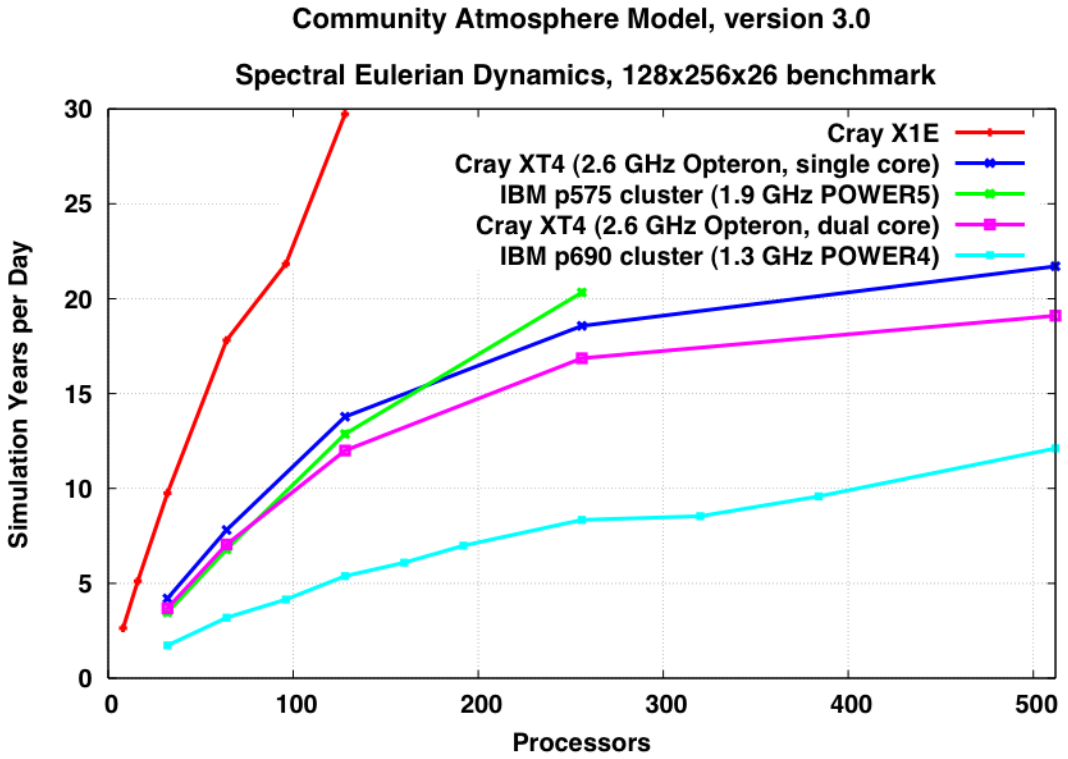
Using additional physics processes with CAM (EUL dycore) pays off



With load balancing, can use more than 128 processes, increasing performance on Cray XT4 by 60%. On XT4, performance is better when using only one core per node (SN) rather than both (VN). Stride 2 optimization (using one core per node in dynamics only) recovers some of the lost performance.

OAK RIDGE National Laboratory

Using additional physics processes with CAM (EUL dycore) still not as effective as OpenMP



IBM POWER5 cluster with OpenMP parallelism and no additional physics processes outperforms Cray XT4 (where OpenMP is not yet an option) with additional physics processes. Note that OpenMP can be used in conjunction with additional physics processes.

Additional physics processes is an unequivocal performance enhancer for the EUL dycore

- **For T85 runs using EUL dycore, physics is twice as expensive as dynamics when using same number of processes in both. When using FV dycore at 0.5-deg, dynamics is (currently) twice as expensive as physics. Thus using more processes in physics than dynamics is more effective for EUL dycore at these resolutions.**
- **For superparameterization version of CAM, cost per column is very high in the physics. Performance results on IBM POWER5 cluster (T42, 1024 processors) show that assigning additional MPI processes to physics is slightly faster than the OpenMP alternative, and will also allow code to run on 1024 IBM BG/L or Cray XT4 processors.***

*** as reported by Marat Khairoutdinov using spectral semi-Lagrangian dycore with same modifications as EUL dycore**

Conclusions

- The scalability of CAM is being extended by way of variable active process count
  - significant performance enhancement with EUL dycore
  - FV dycore code modifications only partially complete
- The procedure will be most effective when including additional physical processes that already scale (e.g., chemistry) or otherwise increase the cost per column (e.g., superparameterization)
- Cases with chemistry will invoke a decomposition with respect to tracer index when performing advection
- Overlapping dynamics (which has been main factor limiting scalability) with tracer advection provides interesting opportunity to extend scalability of FV
  - not effective at coarse resolution

OAK RIDGE National Laboratory

Acknowledgment

- This work was performed under the auspices of the Climate Change Research Division of the Office of Biological and Environmental Research and of the Office of Mathematical, Information, and Computational Sciences, both in the Office of Science, U.S. Department of Energy, by University of California Lawrence Livermore National Laboratory under contract No. W-7405-Eng-48, and by UT-Battelle Oak Ridge National Laboratory under contract No. DE-AC05-00OR22725.
- 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 U.S. Department of Energy under Contract No. DE-AC05-00OR22725.
- This is LLNL Report UCRL-PRES-231564.