Experience with the Full CCSM *

J. B. Drake †
Oak Ridge National Laboratory

P. H. Worley ‡
Oak Ridge National Laboratory

I. Carpenter §
Cray Inc.

M. Cordery ¶
Cray Inc.

Abstract

We present our experiences porting and initial performance of the Community Climate System Model (CCSM3) on the Cray X1. This is the primary model for global climate research in the US and is supported on a variety of computer systems. It will also be used for assessing impacts of climate change for the Intergovernmental Panel on Climate Change (IPCC) Fourth Assessment. The CCSM is the result of a community modeling effort sponsored by the National Science Foundation, the Department of Energy, and the National Aeronautics and Space Administration. The model development is coordinated by the National Center for Atmospheric Research, and the code has demonstrated performance portability across vector and cache based parallel architectures due to software engineering implementing a tunable internal data structure. The application is composed of five executables run in MPMD mode: a sea ice model (CSIM), a land model (CLM), an ocean model (POP), an atmospheric model (CAM) and a flux coupler (CPL6). Each component model communicates with the coupler using MPI and exploits SPMD distributed memory parallelism with MPI.

1 Introduction

CCSM, the Community Climate System Model [1, 6], is a coupled model for simulating the earth's climate systems. It is unique in that it was developed at the National Center for Atmospheric Research (NCAR) with significant contributions from the US Department of Energy, National Aeronautic and Space Administration (NASA) and the university community to provide the research community state of the art simulation capabilities in a freely available code. CCSM is composed of four separate model components that simultaneously simulate the atmosphere, ocean, land surface and sea-ice, and a central coupler component. Each component model integrates the evolution equations of general circulations as well as the physical processes of solar energy absorption and radiation, cloud processes and biological/ecological interaction. The component models progress in the simulation taking independent time steps linked together by regridded state and flux data passed through the coupler. In CCSM3, the dynamical atmospheric model is the Community Atmospheric Model (CAM) [4], a global atmospheric

*This research was 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-Batelle, LLC. 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.

†Computer Science and Mathematics Division, Oak Ridge National Laboratory, P.O. Box 2008, Bldg. 5600, Oak Ridge, TN 37831-6016 (drakejb@ornl.gov)

‡Computer Science and Mathematics Division, Oak Ridge National Laboratory, P.O. Box 2008, Bldg. 5600, Oak Ridge, TN 37831-6016 (worleyph@ornl.gov)

§Cray Inc., 1340 Mendota Heights Rd., Mendota Heights, MN 55120 (ilene@cray.com)

¶Cray Inc., 411 First Avenue South, Seattle, WA 98104 (mcordery@cray.com)

general circulation model developed from the NCAR CCM3 [15]. The ocean model is the Parallel Ocean Program (POP 1.4.3) [10, 14, 21] from Los Alamos National Lab. The sea ice model is the Community Sea-Ice Model (CSIM4) [3, 13, 20]. The Community Land Model (CLM3) [2, 8, 19] is the same land model used with the un-coupled version of CAM. The coupler is CPL6 [5]. The four component models each exchange data only with the coupler, and each is a separate executable.

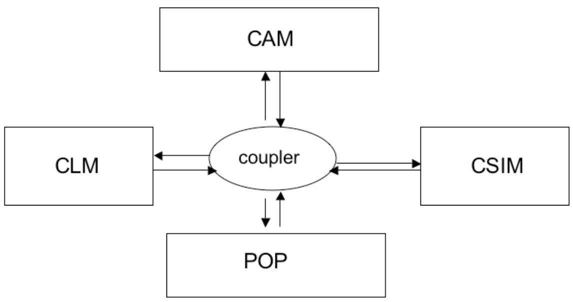

FIGURE 1.1: CCSM components and interactions

The atmospheric resolution corresponds to a 1.4 degree (140km) horizontal mesh and models 26 vertical layers from the surface into the stratosphere with a variable, sigma-pressure hybrid coordinate. The atmospheric solution is based on an Eulerian, semi-implicit, spectral formulation of the primitive flow equations. The ocean model is based on a free-surface, baroclinic formulation with implicit barotropic solve. Both the ocean and sea-ice models use a 1 degree displaced polar grid with a finite difference discretization. The dynamic ice moves according to a visco-plastic rheology. The land model incorporates calculations of soil moisture and vegetation properties that simulate processes on diurnal to seasonal timescales.

The coupled model is typically run in a dedicated mode for century long simulations. Typical throughput for these simulations using 192 processors on an IBM p690 cluster [17] is five simulated years per day, so that a single simulation requires twenty days to complete. Since ensembles are often desired to bound the chaotic response of the climate and generate accurate climate forecast statistics, optimizing performance on the Cray X1 [7] impacts the scientific productivity of climate researchers.

In addition to the dynamical models, data cycling versions for each component are available, which facilitate porting and testing. These components simply read existing data sets and pass data to the coupler. This paper describes our experience porting a prerelease verion of CCSM3.0 to the Cray X1. CCSM3.0 will be released in June 2004.

2 Porting Strategy

The port to the Cray X1 resolved the issues of verification, validation and optimization. These were addressed for each component separately and then for the coupled system as a whole. A number of institutions were involved in the vectorization of individual components, including NCAR, Oak Ridge National Laboratory (ORNL), Los Alamos National Laboratory (LANL), the Arctic Region Supercomputing Center (ARSC), Cray, NEC, and the Japanese Central Research Institute of the Electric Power Industry (CRIEPI).

In addition to the component models, we ported the coupled system framework, which includes the coupler and the utilities it uses:

- MCT - Model Coupling Toolkit [16] from Argonne National Laboratory (ANL),
- MPEU - Message Passing Environment Utilities from NASA,
- MPH - Multi Program Handshaking utility [9] from Lawrence Berkeley Laboratory (LBL) .

In a new computational environment, the CCSM build and test procedures must also be ported. This is not reported on here.

2.1 Porting Issues

Most of the issues we encountered were related to word length. On the Cray X1, default word lengths for both integers and reals are 32 bits. CAM needs to be compiled with `-s real64` to auto-promote reals to 64 bits if not explicitly declared. On the X1, all object files must be built with the same auto-promotion flags so all libraries and component models need to be compiled with `-s real64`. As a result, minor changes were needed in utilities and POP. Other porting issues included checking old Cray-specific code (included when `ifdef CRAY` is true) to determine whether they should be retained. Those related to word length were not (as the old Cray vector systems used 64 bit reals and integers) but many of those related to operating system calls, such as replacing `getenv()` with `pxfgetenv()`, are needed on the X1. We include these with a new `UNICOSMP` cpp token.

Finally, CCSM required a new multiple binary capability to be added to the Cray MPI environment.

2.2 Strategy

The coupler was ported and validated using data models to exercise the coupling framework. After verifying that the coupler worked correctly, we added the non-vectorized component models one at a time to debug the coupled system. The component models were vectorized and optimized in stand-alone mode. Finally, vectorized versions of the dynamic models were included in the most recent version of the CCSM.

Validation of results from the component models on the Cray X1 involved standard error growth tests comparing the vectorized and non-vectorized versions and comparison of results with the vectorized version run on other hardware platforms. Finally, climate statistics were gathered from long simulations and compared with the statistics generated on other platforms using previous versions of the model.

3 Component Optimization and Strategy

An optimization was not accepted if it impacted the simulation adversely. The rules for optimization changes to the models were:

- no significant impact on performance on other target systems,
- solution must be independent of number of processors used,
- no change of solution on other platforms greater than that resulting from a change in the ordering of arithmetic, and then only after showing a clear performance benefit,
- only limited amount of architecture-dependent code (no large scale `#ifdef` sections for different systems).

These rules are followed to ensure maintainable code that runs well on multiple platforms.

Because the software we were porting was still under development, there were frequent updates to the models (both science and optimization changes). The land and ice models were optimized and, where necessary, rewritten[12]. The stand-alone ocean model has been separately optimized for the X1 [22], but not all those optimizations have been included in the version of POP used in the CCSM. We expect the performance of CCSM to be determined primarily by the performance of the CAM atmospheric model, so we focused most of our attention there.

3.1 CAM Optimization

Several groups worked on vectorization of CAM for the Cray and NEC systems, with loose coordination through the CVS repository at NCAR. In addition, there were frequent updates to the model as new modules were added to support the IPCC simulations.

Performance is determined by that of two distinctly different submodels, the dynamical core ("dynamics") and the physical parameterizations ("physics"). As with many atmospheric models, performance profiles were relatively flat across the subroutines, and there is not a single kernel to optimize. However, the loop structure and general layout of the data are regular from routine to routine within the dynamics and physics, respectively, and some general optimization strategies were identified.

Physics optimizations The physics consists of short-wave and long-wave radiation calculations along with the moist processes associated with clouds and convective adjustment. The simplest optimization involved function calls within loops, which inhibit vectorization and streaming. In particular, the function `estblf` is often called within loops. Optimization involved compiling with `-Omodinline` for certain modules to allow more extensive inlining.

Another pattern needing attention on the X1 is error checks that involve I/O, such as the following loop:

```
do i=1,N
  err=f(i)-g(i)
  if(err > tol) then
 write(6,fmt) msg,i,err
   call endrun()
  end if
end do
```

The presence of write statements forces the loop to be scalar, and the call to `endrun()` inhibits streaming. This loop can be changed as follows to allow both streaming and vectorization:

```
j=0;
jerr=0.0
do i=1,N
  err = f(i)-g(i)
  if(err > tol) then
         j=i
    jerr=err
end if
end do
if(j > 0) then
write(6,*)msg,err,j
   call endrun()
end if
```

This type of modification was made in a number of routines, including `qneg3` and `aerosols`. Note that this does not degrade performance on nonvector systems except when the error test is satisfied, in which case the (minor) performance degradation is unimportant.

More substantial optimization was needed in the short and long wavelength radiation routines. In the original code, these were not streamed or vectorized due to a complex cloud algorithm, and there were few opportunities for long vectors.

Optimization of the long-wave radiation routine `radclwmx` included inserting `!DIR$ CONCURRENT` directives for loops with indirect addressing (`i=indx(j)`) and forcing streaming over the number of columns.

Optimization of the short-wave radiation routine `radcswmx` was done several times for both NEC and Cray systems. Our first optimization strategy on the X1 was to vectorize across spectral bands and stream across number of columns. This was very simple to implement and gave a good performance improvement. However, short vector lengths (19) provide relatively inefficient performance compared to what one might get by vectorizing over columns (vector lengths up to 256). Unfortunately, it is not trivial to vectorize over columns because only columns representing locations that are currently receiving solar radiation ("daylight columns") are involved in these computations. A second pass at vectorization was done by NEC in which new data structures and routines were introduced to assist in vectorizing over the number of daylight columns. This version introduced additional complexity and overhead to compress and expand data and provided no significant performance boost on the X1 over the previous version.

Dynamics Optimizations While the physics calculations scale well to high processor counts, scaling was initially poor in the spectral dynamical core. Both communication methods and load imbalance were addressed.

Careful analysis showed that significant load imbalanced was caused by streaming loops with length less than four. This was fixed by moving streaming to loops with more work, such as loops over the number of latitude bands. In addition, using load balancing methods previously implemented in the code reduced load imbalance effects in the physics calculations substantially.

CAM does a number of transposes and other all-to-all communication patterns. Two approaches were taken for optimization of the communication-intensive routines. The first involved replacing point-to-point communication patterns with MPI collective routines, which substantially improved performance. We also investigated replacing MPI with Co-Array Fortran within the MPI wrapper routines. This approach increased the amount of synchronization needed, but this was offset by faster point-to-point communications that was both streamed and vectorized. It has not yet been determined whether the benefits of this approach outweigh the goal of minimizing platform-specfic code.

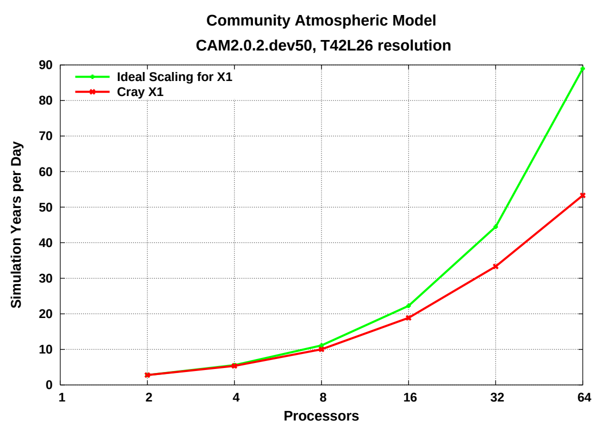

FIGURE 3.1: T42 Performance of CAM on the X1

The performance achieved on the T42 and T85 resolution problems is shown in Figs 3.1 and 3.2. (T42 corresponds to a 2.8 degree horizontal resolution, while T85 corresponds to the 1.4 degree horizontal resolution. Both problems use 26 vertical layers.) Both tests used the "dev50" version of CAM version 2.0.2. The code scales to 64 MSPs for T42 and 128 processors for T85, each of which represents the case of having just one latitude line per processor. In addition, we reached our target of over 20 simulated years per wallclock day on the T85 problem. For reference, Fig. 3.2 includes data from the

IBM p690 cluster at ORNL. On the IBM, OpenMP parallelism is used in the physics, enabling the use of more than 128 processors. Most of the above mentioned CAM optimizations are in the latest CAM and CCSM releases. In particular, the performance of CAM version 3.0 is similar to the dev50 performance described here.

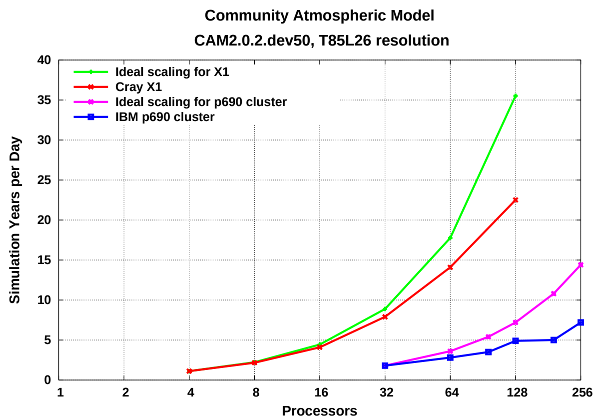

FIGURE 3.2: T85 Performance of CAM on the X1

3.2 CLM optimization

The original CLM2.2 model in CCSM contained data structures that were not well suited to vector processors. Data structures were based on a hierarchy of pointers to derived data types containing scalar quantities scattered throughout memory. The lowest level loops over "plant functional types" had loop lengths of 1 to 20, and the snow/soil loops had negligible work.

This model was substantially re-written to make it more vectorizable. The goal was to develop a single code that runs well on both scalar and vector architectures while maintaining the hierarchical nature of the data structures. Loops over columns were moved into science subroutines, and vectorization was done over these outer loops rather than over the short inner loops over plant functional types and soil/snow levels. Additional optimization included unrolling short loops, interchanging loops, fusing loops and inlining subroutines. Cray streaming directives (CSDs) were added as an optional replacement for OpenMP directives for high level loops.

The new code is substantially more efficient on both vector and scalar systems, running 25.8 times faster on the Cray X1 than the original code and 1.8 times faster on IBM systems. In addition, it uses less memory and the new vector-friendly data structures simplify history updates and reduce the complexity and number of gather/scatter operations. These modifications are included in the latest CLM and CCSM releases. For more information on the CLM modifications and performance, see [11] and [12].

3.3 Coupler

No X1 specific optimization has been done for the coupler. A sparse matrix-vector multiplication routine is structured to permit vectorization. A few porting modifications were needed for word length issues.

4 Coupled model configuration and optimization plan

Optimal performance of CCSM requires determining the right allocation of processors for each component to load balance the five executables. Because the performance of the coupled model is primarily determined by the performance of the atmospheric model, we expect to use 128 processors for the atmospheric component at the T85 resolution and achieve a simulation rate of 20+ simulated years per day. We will need smaller numbers of processors for each of the other components, for an expected total of about 180 processors. The goal is to use enough processors on each of the other models that none of them fall behind, but not so many that those processors sit idle while waiting for data from the coupler. Testing of the performance achieved as a function of the number of processors used in each component has just begun on the X1, but the conjectured balance is

- CAM 128 MSPs,
- POP 24 MSPs,
- CLM 12 MSPs,
- CSIM4 8 MSPs,
- CPL6 8 MSPs.

Concurrent to this work, scientific validation of the coupled system is being done by scientists from NCAR and ORNL.

While the coupled model performance should be close to stand-alone CAM performance, the initial performance is approximately six times slower than expected. Some compiler optimizations have not yet been enabled because of issues building the coupled system and some machine-specific optimizations (primarily in POP) have not yet been included in the CCSM release because of their impact on code readability and performance on other platforms. We are in the process of analyzing the performance data to determine other sources of performance problems, and hope to achieve the expected performance by the end of the summer, 2004.

5 Conclusions

The full CCSM has been ported to the X1 with validation and optimization ongoing. The CCSM build system will include Makefiles, scripts and many source code modifications necessary for running on the Cray X1. Significant optimization of each component has been done by groups at Cray, ORNL, NCAR, NEC and CRIEPI. Performance of individual components is excellent, showing good utilization of the vector architecture and high bandwidth distributed memory subsystem. Initial performance of the coupled model is currently poor and is the subject of continuing work since the coupled model with vectorized components was only available a few days before the Cray User Group meeting.

6 Acknowledgements

This research used resources of the Center for Computational Sciences at Oak Ridge National Laboratory, which is supported by the Office of Science of the U.S. Department of Energy (DOE) under Contract No. DE-AC05-00OR22725. This work sponsored by the DOE/ Office of Science under the SciDAC [18] program.

7 About the Authors

Ilene Carpenter is the group leader for environmental applications at Cray. Her work focuses on optimization and performance analysis of weather and climate models on new systems. She can be reached at ilene@cray.com .

Matthew Cordery is a member of the environmental applications group at Cray. His work focuses on porting, optimization and benchmarking of climate and weather applications. He can be reached at mcordery@cray.com .

John Drake is the group leader of the Computational Climate Dynamics Group in the Computer Science and Mathematics Division at Oak Ridge National Laboratory. His research is in numerical methods and parallel algorithms for atmospheric models. He can be reached at drakejb@ornl.gov .

Patrick Worley is a Senior Research Scientist in the Computer Science and Mathematics Division at Oak Ridge National Laboratory. He has been conducting early evaluations of advanced computer architectures since the early 1990s. He also does research in performance evaluation tools and methodologies, and designs and implements parallel algorithms in climate and weather models. He can be reached at worleyph@ornl.gov .

References

[1] M. B. Blackmon, B. Boville, F. Bryan, R. Dickinson, P. Gent, J. Kiehl, R. Moritz, D. Randall, J. Shukla, S. Solomon, G. Bonan, S. Doney, I. Fung, J. Hack, E. Hunke, and J. Hurrel, *The Community Climate System Model*, BAMS, 82 (2001), pp. 2357–2376.

[2] G. B. Bonan, K. W. Oleson, M. Vertenstein, S. Levis, X. Zeng, Y. Dai, R. E. Dickinson, and Z.-L. Yang, *The land surface climatology of the Community Land Model coupled to the NCAR Community Climate Model*, J. Climate, 15 (1998), pp. 3123–3149.

[3] B. P. Briegleb, C. M. Bitz, E. C. Hunke, W. H. Lipscomb, and J. L. Schramm, *Description of the Community Climate System Model version 2 Sea Ice Model*, tech. rep., National Center for Atmospheric Research, Boulder, CO 80307, May 2002. http://www.ccsm.ucar.edu/models/ice-csim4.

[4] W. D. Collins, P. J. Rasch, *et. al.*, *Description of the NCAR Community Atmosphere Model (CAM 3.0)*, NCAR Tech Note NCAR/TN-464+STR, National Center for Atmospheric Research, Boulder, CO 80307, 2004.

[5] CCSM Coupler. http://www.ccsm.ucar.edu/models/cpl6/.

[6] Community Climate System Model. http://www.ccsm.ucar.edu/.

[7] Cray Inc., *Cray X1*. http://www.cray.com/products/systems/x1/.

[8] Y. Dai, X. Zeng, R. E. Dickinson, I. Baker, G. B. Bonan, M. G. Bosilovich, A. S. Denning, P. A. Dirmeyer, P. R. Houser, G. Niu, K. W. Oleson, C. A. Schlosser, and Z.-L. Yang, *The Common*

Land Model, Bulletin of the American Meteorological Society, 84 (2003), pp. 1013–1023.

[9] C. H. Q. Ding and Y. He, *MPH: a library for coupling climate component models on distributed memory architectures*, in Proceedings of the IEEE/ACM SC2003 Conference, Nov. 15-21, 2003, IEEE Computer Society Press, Los Alamitos, CA, 2003.

[10] D. K. Dukowicz and R. D. Smith, *Implicit free-surface method for the Bryan-Cox-Semtner ocean model*, J. Geophys. Res., 99 (1994), pp. 7991–8014.

[11] F. Hoffman, M. Vertenstein, P. Thornton, K. Oleson, and S. Levis, *Community Land Model version 3.0 (CLM3.0) developer's guide*, Tech. Rep. ORNL/TM-2004/119, Oak Ridge National Laboratory, Oak Ridge, TN, June 2004.

[12] F. M. Hoffman, M. Vertenstein, H. Kitabata, J. B. W. III, P. H. Worley, and J. B. Drake, *Adventures in vectorizing the Community Land Model*, in Proceedings of the 46th Cray User Group Conference, May 17-21, 2004, R. Winget and K. Winget, ed., Eagen, MN, 2004, Cray User Group, Inc.

[13] E. C. Hunke and J. K. Dukowicz, *An elastic-viscous-plastic model for sea ice dynamics*, J. Phys. Oceanogr., 27 (1997), pp. 1849–1867.

[14] P. W. Jones, *The Los Alamos Parallel Ocean Program (POP) and coupled model on MPP and clustered SMP computers*, in Making its Mark – The Use of Parallel Processors in Meteorology: Proceedings of the Seventh ECMWF Workshop on Use of Parallel Processors in Meteorology, G.-R. Hoffman and N. Kreitz, eds., World Scientific Publishing Co. Pte. Ltd., Singapore, 1999.

[15] J. T. Kiehl, J. J. Hack, G. Bonan, B. A. Boville, D. L. Williamson, and P. J. Rasch, *The National Center for Atmospheric Research Community Climate Model: CCM3*, J. Climate, 11 (1998), pp. 1131–1149.

[16] J. W. Larson, R. L. Jacob, I. T. Foster, and J. Guo, *The Model Coupling Toolkit*, in International Conference on Computational Science, V. N. Alexandrov, J. Dongarra, B. A. Juliano, R. S. Renner, and C. J. K. Tan, eds., vol. 2073 of Lecture Notes in Computer Science, New York, 2001, Springer, pp. 185–194.

[17] K. Matsubara, A. H. J. Tenh, and B. Blanchard, *IBM e-server pSeries 670 and pSeries 690 System Handbook*, IBM Redbook, International Business Machines, Inc., May 2003. Form #SG24-7040-02.

[18] Office of Science, U.S. Department of Energy, *Scientfic Discovery Through Advanced Computing*. (available from http://www.sc.doe.gov/ascr/mics/scidac/SciDAC_strategy.pdf), March 2000.

[19] K. Oleson, Y. Dai, G. Bonan, M. Bosilovich, R. Dickinson, P. Dirmeyer, F. Hoffman, P. Houser, S. Levis, G.-Y. Niu, P. Thornton, M. Vertenstein, Z.-L. Yang, and X. Zeng, *Technical description of the Community Land Model (CLM)*, NCAR Tech Note NCAR/TN-461+STR, National Center for Atmospheric Research, Boulder, CO 80307, 2004.

[20] J. L. Schramm, *Community Sea Ice Model (CSIM) user's guide version 4.0*, tech. rep., National Center for Atmospheric Research, Boulder, CO 80307, August 2002. http://www.ccsm.ucar.edu/models/ice-csim4.

[21] R. D. Smith and P. Gent, Ed., *Reference manual for the Parallel Ocean Program (POP), ocean component of the Community Climate System Model (CCSM2.0)*, Tech. Rep. LAUR-02-2484, Los Alamos National Laboratory, Los Alamos, NM, 2002.

[22] P. H. Worley and J. Levesque, *The performance evolution of the Parallel Ocean Program on the Cray X1*, in Proceedings of the 46th Cray User Group Conference, May 17-21, 2004, R. Winget and K. Winget, ed., Eagen, MN, 2004, Cray User Group, Inc.