Performance Evaluation of Ultra-High-Resolution Climate Simulations

John M. Dennis [1] Mariana Vertenstein [2] Anthony P. Craig [3]

Abstract

Currently production climate simulations typically occur at rather coarse resolution, with $\mathcal{O}(100 \text{ km})$ separation between grid points. This choice of resolution is a direct result of limited computing resources. While 100 km resolution is sufficient to resolve global and large continental scale features and trends in the climate system, it is insufficient to resolve regional features. Encouraged by the recent advent of very large-scale compute platforms, we have developed a more scalable and flexible coupling infrastructure that allows for significant increases in resolution and processor counts. Our new coupler infrastructure has enabled the coupling of an atmosphere and land model with either a $\sim$ 25 km (0.25°) or $\sim$ 50 km (0.5°) separation between grid points to an ocean and sea ice model with $\sim$ 10 km (0.1°) separation between grid points. We describe the performance and scalability of ultra-high-resolution climate simulations on as many as 6380 cores on multiple large parallel compute platforms.

1 Introduction

Climate modeling is a Grand Challenge problem [14] requiring enormous computational power. The grand challenge lies in accurately predicting future climate, based on scenarios of anthropogenic emissions and concentrations, across a range of spatial scales from local to global. The current generation of coupled climate system models can simulate most of the continental-scale features of the observed climate. However, these models still cannot simulate climate changes with sufficient regional spatial accuracy. The next generation of models will require much greater spatial resolution, in addition to the inclusion of the full carbon, nitrogen and biogeochemical cycles, the ability to model land-use and ice sheets and more complete representations of sub-grid scale processes to produce regionally improved simulations of climate.

In this paper we will demonstrate the capability of very large scale parallelism to significantly increase the resolution at which it is possible to study climate using current high-performance computing architectures. We limit ourselves to examining the scalability of the upcoming version 4 of the Community Climate System Model (CCSM). CCSM is one of the most extensively used climate models in the world and was an important participant in the Intergovernmental Panel on Climate Change (IPCC) [15]. CCSM is a coupled climate system model that contains component models of the ocean, sea ice, atmosphere and land. The Parallel Ocean Program (POP) [17, 18, 23], developed at Los Alamos National Laboratory is an important multi-agency ocean model used for global ocean modeling. The Community Ice CodE (CICE) [5, 13] also developed at Los Alamos National Laboratory is an extensively used multi-agency code that models sea ice. The Community Atmospheric Model (CAM) [6], is a widely used atmospheric model, whose development is based at the National Center for Atmospheric Research (NCAR), but has a large international community of contributors and collaborators. The Community Land Model (CLM) [12] which was developed at NCAR in collaboration with many national collaborators, models the land surface. Finally the flux coupler (CPL) mediates and controls the exchange of two-dimensional boundary data (flux and state information) between the various component models. The upcoming version 4 of CCSM has a completely new coupling architecture that permits a much greater degree of flexibility in component processor layouts in order to optimize both computation efficiency and model throughput. We note that the CCSM4 system will potentially also include additional component model functionality that is under evaluation, such as atmospheric chemistry, biogeochemistry, and ice-sheet dynamics, which we do not consider here.

The ultra-high-resolution configuration was selected as a a compromise between the needs of climate scientists and what is computationally tractable within the next 5 years. While an cloud-resolving 5 km atmospheric model that may accurately simulate critical convection processes is desirable, it is not yet computational feasible. A 5 km atmospheric model represents a 20-fold increase in resolution and a 160,000-fold increase in the computational cost versus the existing resolution of the atmospheric component used in CCSM. Unfortunately, no computing system capable of sustaining this flop rate will be available in the 5 year time frame. In this study, we determined our target resolutions by holding the integration rate constant and setting resolutions for POP, CICE, CAM and CLM that accurately represent as many physical phenomena as possible.

---

[1] Computer & Information Systems Laboratory , National Center for Atmospheric Research, Boulder, CO 80307-3000 (`dennis@ucar.edu`). The work of this author was supported by the National Science Foundation, and under grants no. #OCI-0749206 and #OCE-0825754. Additional funding is provided through the Department of Energy, CCPP Program Grant #DE-PS02-07ER07-06.

[2] Earth & Sun Systems Laboratory, National Center for Atmospheric Research, Boulder, CO 80307-3000 (`mvertens@ucar.edu`). The work of this author was supported by the National Science Foundation, and under grants no. #OCI-0749206. Additional funding is provided through the Department of Energy, grant no #DE-FC02-97ER62402 and #DE-FC02-07ER64340.

[2] Earth & Sun Systems Laboratory, National Center for Atmospheric Research, Boulder, CO 80307-3000 (`tcraig@ucar.edu`). The work of this author was supported by the U.S. Department of Energy grant no #DE-FC02-97ER62402 and #DE-FC02-07ER64340.

Our resulting target ultra-high spatial resolution has average separation between grid points at the equator of 0.1° for the ocean and sea ice, and either 0.5° or 0.25° for the atmosphere and land models.

In this paper we describe the design and performance of ongoing ultra-high-resolution climate simulations. In Section 2, we describe the background and motivation for this effort. In Section 3 we describe the design of the CCSM4 system architecture. In Section 4 we provide the current performance and scalability of the ultra-high-resolution climate simulations. In particular, we describe the computational platforms used for this scalability study, the method used to load balance the component sets, and the scalability of the resulting application. Finally, in Section 5 we provide conclusions of this study as well as a discussion of future work.

2 Background and Motivation

The CCSM model is comprised of a system of four geophysical components, (atmosphere, land, ocean, and sea ice), which periodically exchange two-dimensional boundary data which each other. In a design decision that dates from the earliest versions of CCSM in the mid 1990s, the exchange of component boundary data occurs only via component communication with a "flux-coupler', thereby resulting in a hub-and-spoke communication pattern. The coupler is also responsible for remapping the boundary-exchange data in space and time. Use of such a coupler to exchange component data simplifies a complex communication pattern that would result from four components simultaneously executing with mutual interaction.

In CCSM3.0, released in 2004, the components were comprised of separate executables running concurrently on disjoint processor sets, including the parallel coupler [7]. This concurrent, multiple executable CCSM architecture was implemented originally to overcome scaling limitations in individual components and to provide a distinct separation between the physical components in order to facilitate independence of model development by separate groups. Subsequent development versions of CCSM now also provide additional support for concurrent component model execution within a single-executable implementation.

The CCSM3 coupling architecture can be contrasted to other widely used coupled climate models. PCM [4] is a single-executable sequential system where each model executes in turn over all processors. FOAM [16] uses a hybrid scheme whereby the atmosphere, land and sea ice are integrated sequentially while the ocean is integrated concurrently. The GFDL FMS [2] system provides support for either concurrent or sequential component execution within a single-executable model. The OASIS coupling system [25] utilizes the framework of the PRISM project [26, 27] and supports concurrent component execution through a flexible runtime environment. Each of these efforts has made trade-offs in their implementation with regard to parallelism, architecture, complexity, flexibility, and performance.

For the recent simulations in support of the Fourth Assessment Report of the Intergovernmental Panel on Climate Change (IPCC) [15], CCSM3, was run at a T85x1° resolution (approximately 1.4° resolution in the atmospheric and land components and 1° resolution in the ocean and ice components), utilizing several hundred processor in total. Although this was one of the higher resolutions models contributing to the report, important features of the climate system are still considered not adequately resolved. Consequently, it is generally accepted that climate model resolution must continue to increase.

Preliminary results for coupled climate simulations performed at under a Lawrence Livermore National Laboratory (LLNL) Institutional Grand Challenge grant [1], and at the National Energy Research Scientific Computing Center (NERSC) suggests that ultra-high-resolution climate resolves several mesoscale processes. We describe two processes that have been observed in these simulations; the winter storm track in the North Atlantic, and the *Cold Wake* behind a hurricane.

Course resolution climate models have difficulty accurately simulating the hydrological cycle. A component of the hydrological cycle, the winter seasonal precipitation rate of the North Atlantic is illustrated in Figure 1. In the left panel in figure 1 an 0.5° atmosphere and land model is coupled to a 1° ocean and sea ice models. In the right panel a 0.5° atmosphere and land model is coupled to a 0.1° ocean and sea ice models. The color in Figure 1 represents precipitation rate (mm/day) during the months of January, February, and March. The contour lines in Figure 1 represent the sea-surface temperatures (SST). Note that the the 0.1° ocean model results in a more accurate simulation of the strong SST gradient associated with the Gulf Stream. A result of this improved representation is that the atmospheric winter storms become stronger and track along the Gulf Stream well offshore versus the lower resolution model, where the maximum precipitation area is confined closer to the coast

We next describe a mesoscale processes not directly related to the hydrological cycle. It was recognized by [3] that a cooling occurs in the ocean just to the right of a hurricane track. This *cold wake*, which has been observed in nature [9], reduces the hurricane intensity. While cold wakes have been generated in high-resolution regional ocean models coupled to weather models [3, 28] they have never been observed in a climate simulation. Previous climate simulations which use high-resolution ocean models performed at the Central Research Institute for Electric Power Industry (CRIEPI) in Japan [8] lacked the resolution in the atmosphere to generate sufficiently strong hurricanes. However within a 0.25° resolution atmosphere coupled to a 0.1° ocean and sea ice performed at LLNL [1] several weak category 4 typhoons were observed. Furthermore,

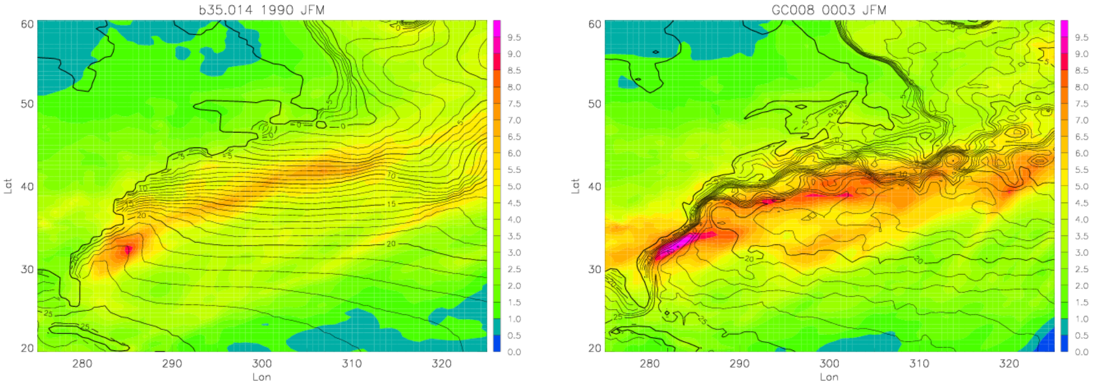

Figure 1: The winter season precipitation rate (mm/day) in the North Atlantic from CCSM using 0.5° atmosphere and land models coupled to a 1° (left panel) and a 0.1° (right panel) ocean and sea ice model. The more realistic representation of the Gulf Stream in the 0.1° Parallel Ocean Program (POP) causes the atmospheric winter storms to become stronger and track along the Gulf Stream well offshore versus the lower resolution model where the maximum precipitation area is confined closer to the coast

one of these typhoons generated a cold wake which is illustrated in Figure 2. Figure 2 is the SST for a region of the Pacific Ocean, south of Japan and east of the Philippines. The location of the typhoon for the past 8 days of the simulation are indicated by an 'x' and are connected by the dashed line. Note the reduction in SST just to the right of the track.

3 CCSM4 system architecture

Increases in the component model resolution and processor counts placed a strain on the concurrent execution model present in CCSM3. In particular, a fully concurrent execution (in time) of each component model is not possible due to restrictions imposed by component data dependencies. Data dependencies between the atmosphere, land and sea ice models partially serializes the system and results in unavoidable idle time. In CCSM3, this idle time is typically 10-20% of the total integration. In addition, the fully concurrent CCSM3 design has repeated proven difficult to debug and maintain.

An alternative coupling architecture has been employed for CCSM4. The new architectures maintains the CCSM3 hub and spoke design but provides new flexibility to allow the the system to be optimally configured in order to account for the data dependencies within the model evolution. The CCSM4 architecture consists of a new driver component, a CPL component along with the CAM, CLM, CICE, and POP component models. The driver component controls the time evolution of the system, the component communications (i.e. the two dimensional boundary exchanges) and the when components write restart files.

As in CCSM3, the CPL component provides the necessary regridding and redistribution of boundary data between model components, as well as being responsible for the calculation of certain fluxes. Regridding of boundary data is necessary when the component models are executing on different computational grids. Redistribution of boundary data is necessary because the component models are potentially executing on different processor sets or with different decompositions.

Unlike CCSM3, the CCSM4 architecture provides the flexibility to support fully concurrent, fully sequential or hybrid execution modes. In the fully sequential configuration, a single set of processors executes all the component models sequentially in time. In the hybrid execution mode, features of both sequential and concurrent execution mode is possible. We illustrate sequential and hybrid layouts in Figure 3. In the sequential layout, illustrated on the left side of Figure 3, all processors execute each component of CCSM in turn. A hybrid sequential/concurrent layout is illustrated on the right side of Figure 3. In the hybrid case, the driver component executes on all processors. The CPL and CAM components execute on a single processor set, CLM and CICE on a subset of the CPL and CAM processor set, while POP executes on set of processors totally disjoint from the CPL, CAM, CLM, or CICE processor set. It is also possible to configure CCSM4 in a fully concurrent layout (as in CCSM3) where disjoint processor sets are allocated to each component.

The flexible CCSM4 coupling architecture has several distinct advantages over the CCSM3 concurrent-only design. First, component processor configurations can be constructed that accurately reflect the data dependencies of the calculation. For example, recent changes to the CAM model prevents the concurrent execution of CAM and CICE as well as CAM and

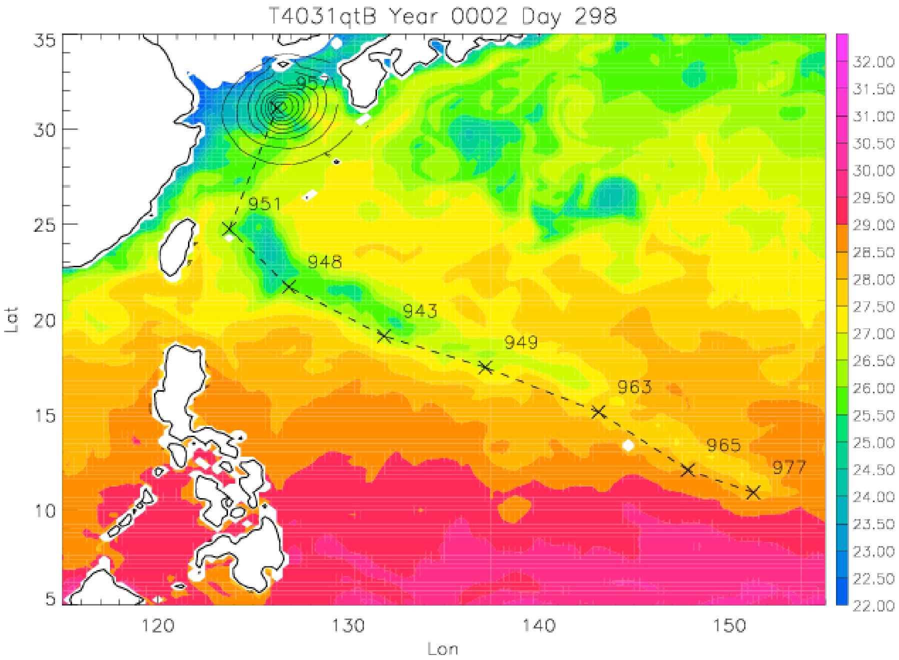

Figure 2: Sea surface temperature in the Pacific Ocean south of Japan and east of Philippines in a ultra-high-resolution coupled simulation that uses a 0.25° atmosphere and land. Note the reduction in sea surface temperature right of a storm track of a category 4 typhoon is consistent with the several degree reduction in temperature observed in nature [9].

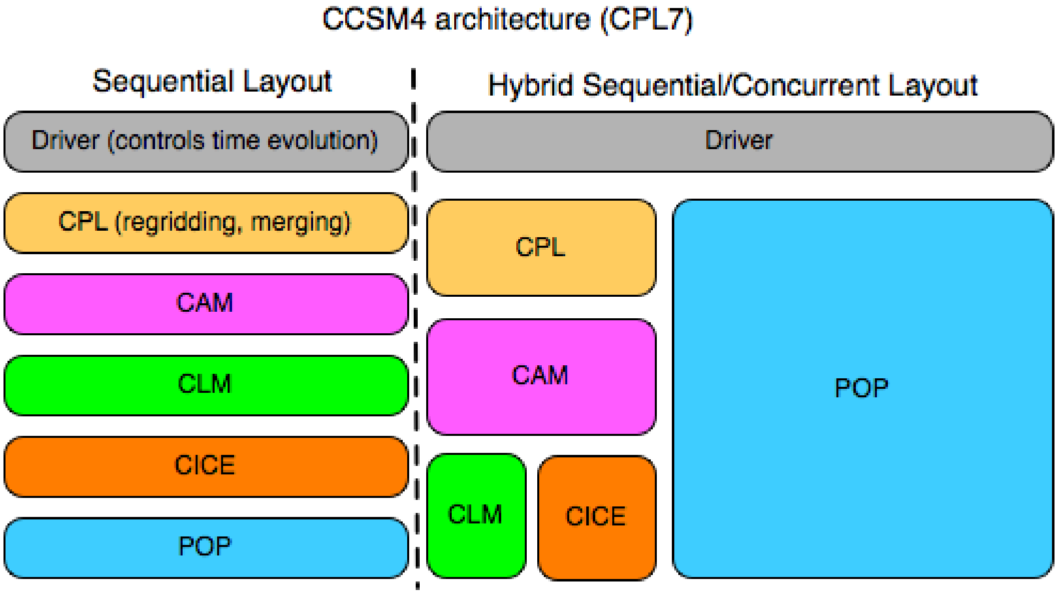

Figure 3: Execution diagram of the CPL7. Flexible design allows for execution in either sequential or hybrid/concurrent execution modes.

CLM. The previous concurrent design would force the idling of processors associated with the CICE or CLM calculations while CAM was executing. Secondly, the ability to have several components execute sequential potentially improves communication efficiency by eliminating unnecessary communication. In particular sequential execution potentially allows communication between models to be achieved through either memory copies or a message passing between a small number of neighboring processors.

The CCSM4 architecture has proven to be significantly easier to port and debug on new platforms. As new architectures become available, a system that can be ported quickly to these platforms will provide new computational and scientific opportunities. Finally, the CCSM4 architecture system also simplifies the process of incorporating new science into the model system since the time sequencing of components is transparent, whereas the time sequencing of component boundary exchange in the concurrent CCSM3 is extremely subtle.

4 Status of the ultra-high-resolution CCSM

We next describe the computational aspects of our ultra-high-resolution CCSM configuration. We note that in previous versions of CCSM parallelism was limited to $\mathcal{O}(100)$ cores with resolutions no greater than 1°. Our ultra-high-resolution configurations represent a enormous leap in available parallelism and model resolution. To limit the scope of this paper, we restrict ourselves to a configuration with a 0.1° ocean and sea-ice coupled to a 0.5° atmosphere and land model. The high-resolution CCSM (CCSMhr05A) benchmark consists of executing this configuration for 5 days without disk I/O. Information about the computational grid is outlined in Table 1. Two distinctive computation grids are used. POP and CICE use a 0.1° tripole grid [21], whereas a 0.5° grid is used by CLM and CAM. Note that the computational grid used by CLM and CAM does not have a fixed separation of 0.5° between grid points but rather an average separation of 0.47° in longitude and 0.63° in latitude.

POP		
Name	grid point separation	# of gridpoints $nx \times ny \times nz$
tx0.1v2	$1/10^\circ$	$3600 \times 2400 \times 42$

CICE		
Name	grid point separation	# of gridpoints $nx \times ny \times icetypes$
tx0.1v2	$1/10^\circ$	$3600 \times 2400 \times 20$

CLM		
Name	grid point separation	# of gridpoints $nx \times ny \times pfts$
0.47x0.63	$1/2^\circ$	$576 \times 384 \times 17$

fv-CAM		
Name	grid point separation	# of gridpoints $nlon \times nlat \times nlev$
0.47x0.63	$1/2^\circ$	$576 \times 384 \times 26$

Table 1: Description of grids used in the ultra-high-resolution CCSM configuration.

Each CCSM component models (CAM, CLM, POP, CICE, and CPL) has its own scalability and performance characteristics. The scalability of POP has been studied extensively and has been successfully scaled to 30,000 cores [10]. Versions of CICE have also been successfully scaled beyond 10,000 cores on a Cray XT-3 [11]. The scalability and performance of CAM has also been studied extensively [20, 24]. All components have to varying degrees benefited from refactoring of algorithms and implementations to reduce memory and load-imbalance. It should be noted that additional refactoring is on-going including the elimination of the serialization of the disk I/O which negatively impacts scalability. To address the lack of parallelism with the disk I/O subsystem, we are developing the Parallel I/O (PIO) library. PIO is a high level I/O library which will provide a standard interface for all models. While PIO has an API for defining user data that closely follows netCDF/pNetCDF, the back-end of PIO can use either MPI-IO or pNetCDF [19] to accomplish parallel disk I/O. Because the disk I/O subsystem is still under development, we concentrate instead on examining the scalability of the purely computational aspects of ultra-high-resolution coupled system by providing performance results without disk I/O cost included. First we describe our computational platforms used for this study.

4.1 Compute Platforms

We present results scalability and performance results for a prototype version of the CCSM4 system (CCSM4_alpha) for three different computational platforms. *Franklin* is a Cray XT-4 system, where each compute node is a single socket 2.3 Ghz AMD Opteron quad-core processor located at the National Energy Research Scientific Computing Center (NERSC). Franklin contains a total of 38,640 cores connected with the Cray Seastar interconnect. *Kraken* is a Cray XT-5 system, where each compute node is a dual socket 2.3 Ghz AMD Opteron quad-core processors located at National Institute for Computational Sciences (NICS). Kraken contains a total of 66,048 cores connected with the Cray SeaStar2+ interconnect. *Atlas* is Appro Linux cluster, where each compute node is a quad socket 2.4 Ghz AMD Opteron dual-core processors located at Lawrence Livermore National Laboratory (LLNL). Atlas uses a switched Infiniband DDR network to connect 9,216 cores.

4.2 Load balancing component sets

As noted in Section 3, CCSM4 has the flexibility to assign cores to components at runtime. For the high-resolution work, we define five common processor configurations: extra small (XS), small (S), medium (M), large (L), and extra large (XL). Table 2 contains the current best processor counts for each of these configurations. Figure 4 illustrates two potential XL configuration that use 4952 and 5844 cores respectively.

We describe the top panel of Figure 4 first that represents the computational cost to simulate a single model day, excluding disk I/O costs on 4952 cores. The x-axis shows the number of cores and the y-axis shows the time each component takes to execute. In this configuration, CAM is placed on the first 1,664 processors and takes a total of 49 seconds to execute a wall-clock day. The CICE model uses the first 1,800 processors and takes 38 seconds to execute a wall-clock day. The CAM and CICE models execute sequentially on the same group of cores. The CLM model executes concurrently with CICE on 16 processors. CPL7 currently takes 10 seconds to complete a model day on 1800 processors. In this configuration a total of 1816 processors are allocated to CAM, CICE, CLM, and CPL7 and require a total of 97 seconds per day. The remaining 3136 processors are allocated to POP, which requires 147 seconds per day. The total time to simulate a model day is therefore 147 seconds. In this configuration 1816 processors are idle for 50 seconds. A more efficient configuration on 5,844 processors is illustrated in the bottom panel of Figure 4. An improved load-balanced CCSM is achieved by increasing the number of processors on which POP executes such that it approximately matches the execution time for the CAM, CICE, CLM and CPL7 components. When POP is executed on 4,028 cores, the time to simulate a model day drops to 107 seconds.

Config size	Type	Processor counts					
		POP	CAM	CICE	CLM	CPL7	TOTAL
XS	Sequential	480	480	480	480	480	480
S	Sequential	1024	1024	1024	1024	480-1024	1024
M	Hybrid	1232-1376	480	480	480	480	1712-1865
L	Hybrid	2464-2634	1024	1024	16	480-1024	3488-3658
XL	Hybrid	3136-4028	1664	1800	16	480-1800	4952-6380

Table 2: CCSM component processor configurations for the ultra-high-resolution baseline configuration.

The design of the CCSM4 coupling architecture allows the necessary flexibility to adjust to changes in component cost and machine performance. For example if the cost of the CAM component of the coupled system significantly increased due an increase in resolution or inclusion of new sub-grid scale processes, the processor count of POP could be adjusted to minimize idle time within the coupled system.

4.3 Scalability of coupled system

We next provide the scalability of the CCSMhr05A benchmark for all five processors configurations on the three compute platforms. Top panel of Figure 5 is the simulated rate for CCSMhr05A. On the x-axis is the number of physical cores, while the y-axis is the number of simulated years per wall-clock day. The value of simulated years per wall-clock day is frequently used by the climate community because it is important measure of the ability to perform research in a timely manner. A typical climate experiment requires tens to hundreds of years of simulation data. A minimum of 5 simulated years per day is often quoted as the minimum useful simulation rate. The simulation rates provided in Figure 5 are for the MPI execution mode. While other versions of CCSM support a hybrid OpenMP/MPI execution mode, the OpenMP/MPI execution mode is not yet supported for high-resolution configuration.

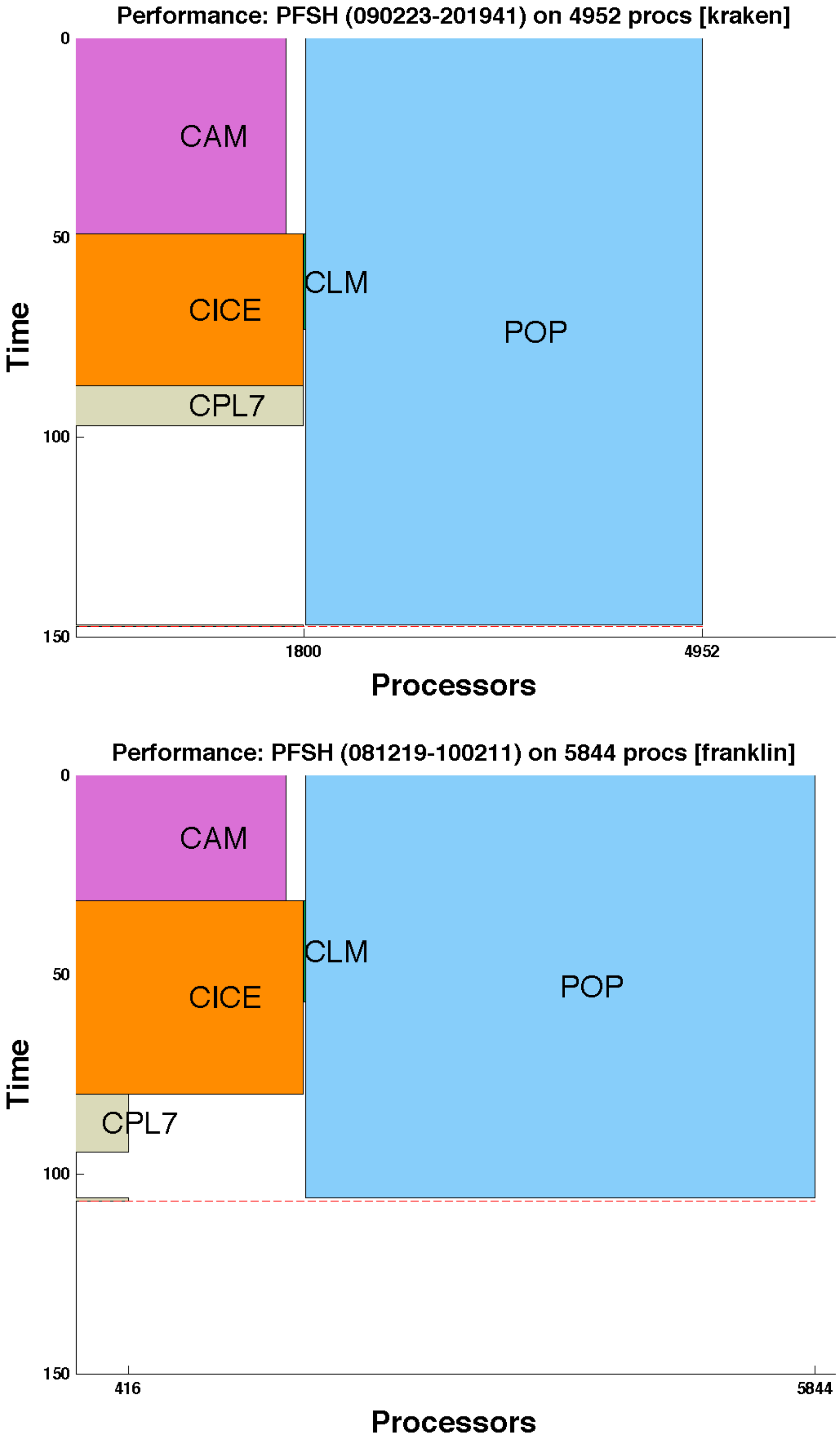

Figure 4: Performance of CCSMhr05A benchmark on a Cray XT-4/5. The top panel, a configuration on 4,952 cores, includes a large amount of idle time due an load imbalanced in the coupled system. The bottom panel, a more efficient configuration on 5,844 cores, has significantly reduced load imbalanced.

It is apparent from the top panel of Figure 5 that the maximum obtainable simulation rate of 2.3 years per wall-clock day is still significantly short of the traditional minimum useful simulation rate for climate. However, it is sufficiently close as to allow for multi-decadale runs. It should be noted that the simulation rate which can be obtained on 480 cores which range from 0.14 to 0.20 years per wall-clock day are entirely inadequate for climate simulations. We include results from small processor counts to analyze the scalability of larger systems.

The results in the top panel of Figure 5 contain several interesting features. First, it should be noted that each platform differs in terms of processor and network configuration. The two systems that are the closest in configuration, Franklin and Kraken have the most similar though not identical scaling characteristics. In particular, the simulation rate for Kraken on 5,844 processor configuration is 13% lower than Franklin. While it is uncertain the reason for the performance lose on Kraken at large processor counts, preliminary investigations suggest that it maybe be due to an increase in OS jitter [22] on Kraken versus Franklin. OS jitter is the prime suspect because a small section of POP, which is very sensitive to message latency and MPI reduction performance, is unexpectedly expensive. It is also instructive to compare the simulation rate of CCSMhr05A benchmark on Atlas versus Kraken. While they both use similar processors, dual-core AMD Opteron on Atlas and quad-core AMD Opteron on Kraken, Atlas uses a commodity Infiniband DDR network versus the proprietary Cray Seastar network. The Infiniband network which has higher latency than Seastar appears to have a large impact on the obtained simulation rate on large core counts. In particular the simulation rate on 3658 Atlas cores is 37% lower than on Franklin.

The bottom panel in Figure 5 is the computational cost to simulate a model year for each configuration. The x-axis is the number of cores while the y-axis is the number CPU hours necessary to simulate a model year. If we only consider those configurations with a simulation rate greater than 1 simulated year per wall-clock day, than the cost ranges from a low of 58,140 CPU hours to a high of 74,280 CPU hours on Kraken. With perfect scaling, the computational cost would be independent of processor count. However load imbalance in the coupled system and scalability issues with the component models increase the cost of the system as processor count is increased. Note that the cost differences between several different XL configurations (4,952, 5,292, 5,844,6380) are due to the minimization of core idle time. The impact Seastar network has on cost to perform the simulation is very apparent in the bottom panel of Figure 5. Using the L configuration, Kraken requires 33% fewer CPU hours per simulated year than Atlas.

5 Conclusion and Future Work

We have described the performance of ultra-high-resolution climate simulations performed on several very large scale parallel compute platforms. The ability to couple ultra-high-resolution climate models together was not even possible in previous versions of CCSM. While the scientific results from these coupled runs are still preliminary, there are indications that it resolves a number of interesting mesoscale phenomenon. In particular the first every hurricane cold wake in a climate model was observed. Additionally, use of an $0.1°$ ocean, which provides a more realistic representation of the Gulf Stream, allows for a more accurate representation of the winter storm track in the North Atlantic.

Access to several large compute platforms has allowed us to examine the scalability of ultra-high-resolution CCSM. We have discovered that is possible to utilize large scale parallelism to nearly achieve climatologically useful simulation rates for a fully coupled system with $0.1°$ ocean and sea ice models to $0.5°$ atmosphere and land models. While our maximum simulation rate of 2.3 simulated years per day is lower than the traditionally minimum of 5 simulated years per day, it is sufficiently close as to make multi-decadale ultra-high-resolution climate simulations possible. It should be emphasized that ultra-high-resolution climate simulations are only possible at the very largest processor counts evaluated. Our scalability evaluation, which includes compute platforms with several different message passing networks, has indicated the importance of a high performance low latency network on the simulation rate of the CCSMhr05A benchmark at large processor counts.

The performance and scalability evaluation presented in this paper is only the start of a more through analysis. In particular other compute platforms should be evaluated including the IBM p575 cluster and IBM Blue Gene/P system. We also intend to port and analyze the performance of the CCSMhr05A benchmark on other very large compute platforms capable of simulating ultra-high-resolution climate when they become available. We also intend to explore the impact that OpenMP/MPI programing paradigm has on scalability. Finally we intend to complete the integration of the PIO library into all component models which will significantly improve the sustained disk I/O bandwidth.

References

[1] Dave Bader, Julie McClean, Art Mirin, Detelina Ivanova, James Boyle, Steve Klein, Bor Chan, Mariana Vertenstein, John Dennis, Frank Bryan, Nancy Norton, Tony Craig, Mathew Maltrud, Elizabeth Hunke, Phil Jones, Pat Worley, and

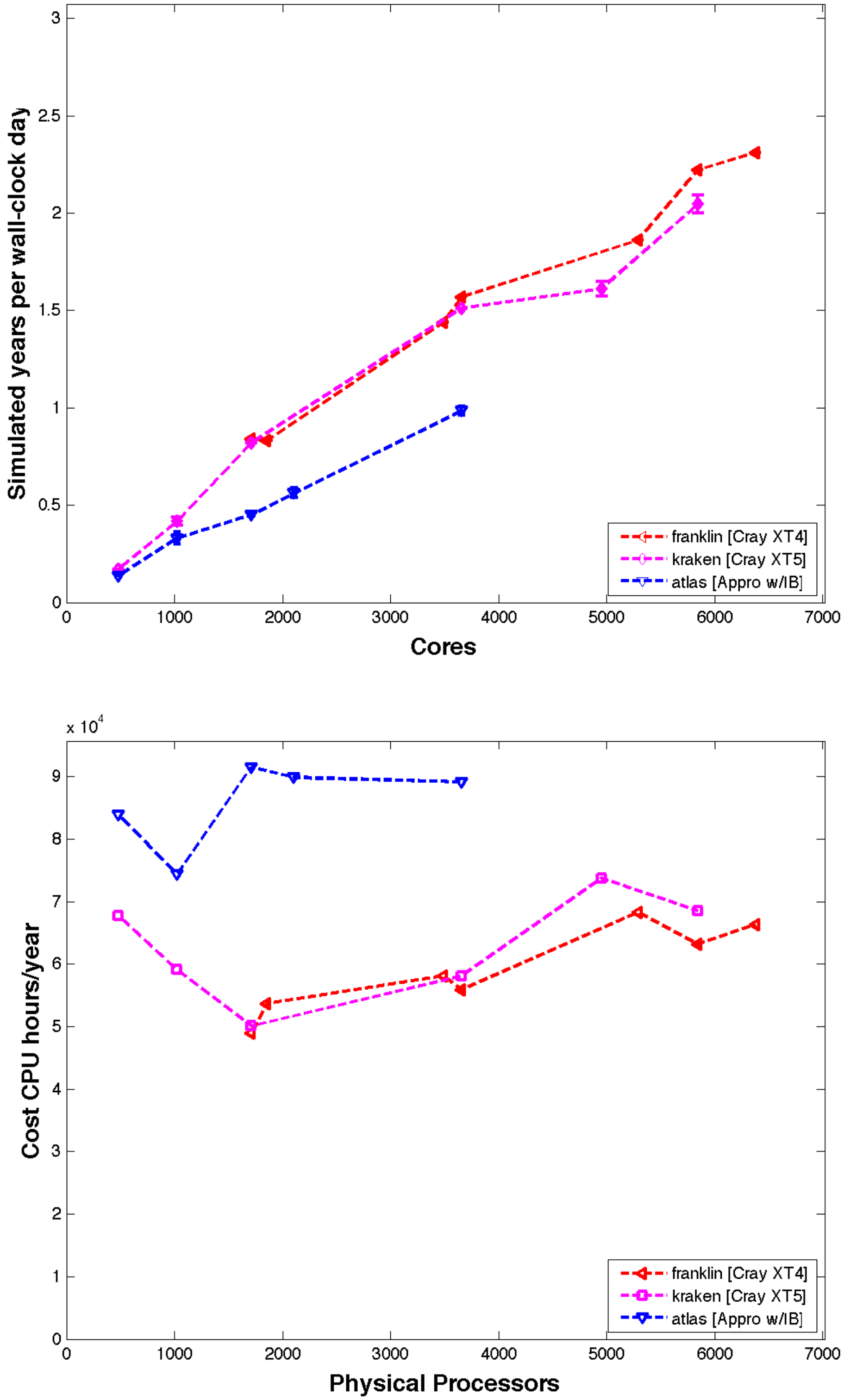

Figure 5: Simulated years per wall-clock day for ultra-high-resolution CCSM (top panel). Simulation cost for ultra-high-resolution CCSM (bottom panel).

Rob Jacob. Ultra-high-resolution coupled climate simulations. Proposal for the LLNL 3rd Institutional Unclassified Computing Grand Challenge, 2008.

[2] V. Balaji. Workshop on FMS at GFDL, September 2004.

[3] Morris A. Bender and Issac Ginis. Real-case simulation sof hurricane-ocean interation using a high-resolution coupled model: Effects on hurricane intensity. *Monthly Weather Review*, 128(4):917–946, 2000.

[4] T. Bettge, T. Craig, R. James, G. Strand, and V. Wayland. Performance of the parallel climate model on the SGI Origin 2000 and the CRAY T3E. In *41st Cray User Group Conference*, Minneapolis, Minnesota, May 1999.

[5] CICE: A Community Ice CodE. http://climate.lanl.gov/Models/CICE, 2008.

[6] W. D. Collins, P. Rasch, B. A. Boville, J.R. Hack, J.R. McCaa, D. L. Williamson, B. P. Briegleb C. M. Bitz, S-J. Lin, and M. Zhang. The formulation and atmospheric simulation of the Community Atmosphere Model version 3 (CAM3). *Journal of Climate*, 19(11):2144–2161, 2006.

[7] T. Craig, R. Jacob, B. Kauffman, T. Bettge, J. Larson, E. Ong, C. Ding, and Y. Hu. CPL6: The new extensible, high performance parallel coupler for the community climate system model. *The International Journal of High Performance Computing Applications*, 19(3):309–327, 2005.

[8] Central Research Institute for Electric Power Industry (CRIEPI). http://criepi.denken.or.jp/en/index.html, 2009.

[9] Eric A. D'Asaro, Thomas B. Sanford, P. Peter Niiler, and Eric J. Terrill. Cold wake of hurricane frances. *Geophysical Research Letters*, 34, 2007.

[10] J. M. Dennis. Inverse space-filling curve partitioning of a global ocean model. In *IEEE International Parallel & Distributed Processing Symposium*, Long Beach, CA, 2007.

[11] J. M. Dennis and H. M. Tufo. Scaling climate simulation applications on IBM Blue Gene. *IBM Journal of Research and Design*, 2007. To appear.

[12] F. M. Hoffman, M. Vertenstein, H. Kitabata, and J. B. White III. Vectorizing the community land model. *International Journal of High Performance Computing Applications*, 19:247–260, 2005.

[13] E. C. Hunke and W. H. Lipscomb. CICE: the Los Alamos sea ice model documenation and software user's manual version 4.0. Technical Report LA-CC-06-012, Los Alamos National Laboratory, T-3 Fluid Dynamics Group, August 2008.

[14] P. G. Huray and et al. A research and development strategy for high performance computing. Executive Office of the President, Office of Science and Technology Policy, 1987.

[15] The Intergovernmental Panel on Climate Change (IPCC), climate change 2007: The physical science basis, summary for policymakers, contribution of working group i to the fourth assessment report of the Intergovernmental Panel on Climate Change. http://www.aaas.org/news/press_room/climate_change/media/4th_spm2feb07.pdf, 2007.

[16] R. Jacob, C. Schafer, I. Foster, M. Tobis, and J. Anderson. Computational design and performance of the Fast Ocean Atmosphere Model, version one. In *Proc. 2001 International Conference on Computatinal Science*, 2001.

[17] P. Jones. Parallel Ocean Program (POP) user guide. Technical Report LACC 99-18, Los Alamos National Laboratory, March 2003.

[18] P. W. Jones, P. H. Worley, Y. Yoshida, J. B. III White, and J. Levesque. Practical performance portability in the Parallel Ocean Program (POP). *Concurrency Comput. Prac. Exper*, 17:1317–1327, 2005.

[19] J. Li, W. Liao, A. Choudhary, R. Ross, R. Thakur, W. Gropp, R. Latham, A. Siegel, B. Gallager, and M. Zingale. Parallel netCDF: A high-performance scientific I/O interface. In *Proceedings of SC2003: High Performance Networking and Computing*, Phoenix AZ, 2003.

[20] A. A. Mirin and P. H. Worley. Extending scalabiity of the community atmosphere model. *J. Phys.: Conf. Ser.*, 78(012082), 2007.

[21] R. J. Murray. Explicit generation of orthogonal grids for ocean models. *J. Comp. Phys.*, 126:251–273, 1996.

[22] F. Petrini, D. J. Kerbyson, and S. Pakin. The case of the missing supercomputer performance: Achieving optimal performance on the 8,192 processors of ASCI Q. In *ACM/IEEE Conference on Supercomputing (SC'03)*, Phoenix, Arizona, Nov. 2003.

[23] The Parallel Ocean Program (POP). http://climate.lanl.gov/Models/POP, 2006.

[24] W. B. Sawyer and A. A. Mirin. A scalable implementation of a finite-volume dynamical core in the community atmosphere model. In *Proceedings of Parallel and Distributed Computing and Systems*, 2004.

[25] S. Valcke. OASIS3 user guide, PRISM support initiative report no 3, 2006.

[26] S. Valcke, D. Declat, R. Redler, H. Ritzdorf, T. Schoenemeyer, and R. Vogelsang. The PRISM coupling and I/O system. In *VECPAR'04, Proceedings of the 6th International Meeting, Vol. 1: High performance computing for computational science*, Universidad Politecnica de Valencia, Valencia, Spain, 2004.

[27] S. Valcke, E. Guilyardi, and C. Larsson. PRISM and ENES: A European approach to Earth system modeling. *Concurrency Computinng: Pract. Exper.*, 17:1–16, 2005.

[28] W. Xhao and S.S. Chen. A coupled atmosphere-wave-ocean framework for high-resolution modeling of tropical cyclones and coast storms. In *Sixth WRF and MM5 User Workshop*, Boulder, Colorado, June 2005.