Comparing Linux Clusters for the Community Climate System Model

Matthew Woitaszek, Michael Oberg, and Henry M. Tufo

Department of Computer Science
University of Colorado, Boulder
{matthew.woitaszek, michael.oberg}@colorado.edu, tufo@cs.colorado.edu

Abstract. In this paper, we examine the performance of two components of the NCAR Community Climate System Model (CCSM) executing on clusters with a variety of microprocessor architectures and interconnects. Specifically, we examine the execution time and scalability of the Community Atmospheric Model (CAM) and the Parallel Ocean Program (POP) on Linux clusters with Intel Xeon and AMD Opteron processors, using Dolphin, Myrinet, and Infiniband interconnects, and compare the performance of the cluster systems to an SGI Altix and an IBM p690 supercomputer. Of the architectures examined, clusters constructed using AMD Opteron processors generally demonstrate the best performance, outperforming Xeon clusters and occasionally an IBM p690 supercomputer in simulated years per day.

1 Introduction

The Community Climate Simulation Model (CCSM) is a coupled climate model used to simulate the Earth’s climate system. Released by the National Center for Atmospheric Research (NCAR), CCSM is the result of collaborative efforts between the National Science Foundation, the US Department of Energy, and the National Aeronautics and Space Administration. CCSM consists of four independently developed component models that simulate the atmosphere (atm), the ocean (ocn), land surfaces (lnd), and sea ice (ice), and a flux coupler (cpl) integrates the four models to simulate the entire Earth climate system [1].

Cluster computer systems have been used for simulations in mesoscale meteorology for the past several years. As early as 1999, the PSU/NCAR Mesoscale Model MM5 was fully functional on Beowulf-style clusters of commodity PCs, and Dorband showed that Beowulf clusters provided the best performance relative to cost [6]. Climate modeling using CCSM, however, has not yet migrated from supercomputer to cluster platforms. The model’s two officially supported target platforms are IBM AIX and SGI IRIX64 [1]. Only the most recent version of CCSM, version 2.0.1, mentions the possibility of executing on Linux systems, and our experience suggests that this support is highly dependent on the compilation environment and specific versions and implementations of MPI.

The pertinent question is: What type of cluster is best suited for running CCSM? The most direct analysis would simply run CCSM on various architectures and report the results, but CCSM's complexity prevents its immediate execution on cluster systems without substantial compilation adjustment. Therefore, this paper presents a performance analysis of two of CCSM's constituent dynamical models, the Community Atmospheric Model (CAM) and the Parallel Ocean Program (POP), on clusters with a variety of processors and interconnects. These two models were selected because they require the greatest amount of computational walltime for a given simulation [3].

The remainder of this paper is organized as follows: Section 2 describes the component models of CCSM, in particular CAM and POP, which we examine in this performance analysis, and Section 3 describes relevant related work. Section 4 introduces our target cluster and supercomputer architectures, and Section 5 presents the performance results. Section 6 continues with a discussion of compiler performance on Intel platforms, and Section 7 examines the performance of the interconnect. The paper closes with future work and conclusions.

2 CCSM Models

The CCSM software package contains three different atmosphere models, two land models, two ocean models, two sea-ice models, and one coupler. In the default CCSM configuration, the following models are utilized: The atmosphere model is the Community Atmosphere Model (CAM), a global atmospheric circulation model developed by NCAR. The ocean model is the Parallel Ocean Program (POP), developed by Los Alamos National Laboratory. The land surface model is the Community Land Model, developed by NCAR. The sea ice component is the Community Sea Ice Model (CSIM4) [1].

On NCAR production runs, CCSM uses the T42L26 (128x64x26) resolution for the atmosphere and land simulations, and 1 degree 40 level (320x384x40) resolution for the ocean and ice simulations [3]. Each component functions only at specific processor counts, and typical configurations are shown in Table 1.

Table 1. Typical parallelization configurations for coupled CCSM models

Grid	atm	lnd	ice	ocn	cpl	Total
T42	32	8	32	48	8	128
T42	64	3	16	40	4	127
T42	32	12	16	48	1	109
T42	32	12	16	40	4	104
T42	32	3	16	40	4	95
T42	16	8	8	40	1	73
T31	48	16	8	8	2	82
T31	32	8	4	4	1	49

3 Related Work

The performance of CCSM, and in particular POP and CAM, has been of interest since the integration of the original models into the coupled climate simulator. In 1999, Drake performed a detailed analysis of the components of the Community Climate Model (CCM) with respect to parallelization performance on the Origin 2000, the Cray T3E, and the IBM Power3 [5]. Noting that "large scientific simulation codes like CCM3 outlive most computing platforms," Drake concluded that the built-in tuning options provide sufficient flexibility to optimize the model and such tuning is necessary to achieve the optimal performance.

Two recent studies examine the portability and performance of POP and CAM. First, Jones examined POP on a variety of architectures, including the Earth Simulator, a Cray X1, an IBM p690 cluster, a SGI Origin3000, and two IBM SPs [8]. Jones noted that POP is dependent on CPU to main memory bandwidth, which our results also demonstrate, and is best suited for vector microprocessors. In a study similar to ours, Worley compared the execution of POP and CAM on several supercomputers, including an IBM p690, a HP AlphaServer SC, a SGI Origin3000, and an IBM SP with WinterHawk II nodes [13]. For CAM, Worley found that the p690 and AlphaServer provided the best performance. For POP, tests run by Jones again indicated that the Earth Simulator and Cray X1 vector supercomputers had the best performance, with the p690 and AlphaServer dramatically lower but above the remaining architectures. While these studies analyze POP and CAM on supercomputer platforms, they did not include commodity clusters in their results.

Several researchers and organizations are working on porting CCSM to function on Linux-based cluster systems. John Taylor's work with CCSM on the Jazz cluster at Argonne National Laboratory was recently introduced into the CCSM 2.0.1 source code as a baseline for providing Linux support, although it retains the disclaimer that tailoring the model to a specific environment may still require a substantial amount of work [1]. While we were unable to apply the port to most of the platforms we tested, and therefore tested only POP and CAM, recent reports indicate that several sites have started to run CCSM on small-scale clusters. For example, the University of Bern Physics Institute reports that CCSM 2.0.1 runs on their 32-processor Linux cluster and is only 1.7 times faster on an IBM SP4 with the same processor counts [12].

4 Platforms

For our tests, we examined several platforms built on the following microprocessor families: Intel Xeon, Intel Itanium, AMD Opteron, and IBM Power4.

4.1 Intel Xeon Clusters

We examined three Xeon clusters. The first cluster consists of 64 dual Xeon 2.4 GHz nodes. Each node has 1 GB of RAM per processor. Each processor contains 512 KB

L2 cache running at core speed with a 400 MHz front side bus. This configuration provides a total memory bandwidth of 3.2 GB/s per processor. Each processor is rated at 65 W of power, with the Northbridge memory controller chip consuming an additional 11 W [7].

The dual 2.4 GHz Xeon cluster uses a Dolphin Interconnect Solutions Wulfkit package to connect the 64 nodes in an 8x8 2D torus with the Scali MPI implementation [4]. The small message latency is about 4 to 5 μs, and the peak bandwidth is approximately 250 MB/s.

The second cluster, which consists of 350 single-processor Xeon 2.4 GHz nodes, is similar to the first system except that half of the processors have 2 GB of memory and half have 1GB. For our benchmarks, we utilized only processors with 1 GB of memory. This system uses Myricom's Myrinet PCIXD solution for the interconnect. The quoted small message latency is about 6.3 μs, and the peak bidirectional bandwidth is approximately 250 MB/s [9].

The third cluster consists of 48 dual-processor Xeon 3.06 GHz nodes with 512KB cache. The nodes were connected with Infiniband, which provides about 250 MB/s bidirectional bandwidth and has a latency of about 4 μs.

4.2 AMD Opteron Clusters

The AMD Opteron clusters consist of 32 dual processor nodes, each containing 2.0 GHz AMD Opteron processors with 4GB of RAM. Each Opteron processor contains 1 MB L2 cache. With the Hyperchannel memory controller integrated on each processor, and the processors connected in a crossbar configuration, each processor has a bidirectional memory bandwidth of 3.2 GB/s [1]. The Opteron processor is rated at 82.1 W. These clusters also employed Myricom's Myrinet PCIXD interconnect.

4.3 Intel Itanium - SGI Altix Supercomputer

The SGI Altix line of supercomputers are powered by Itanium2 processors, with up to 64 in a single-system image. We were given results on two SGI 3700 supercomputers, one with 1.3GHz Itanium2 processors and one with 1.5GHz Itanium2 processors. These Itanium2 processors have a L2 cache of 256KB, and an on-chip L3 cache of 3MB. Each processor has a memory bandwidth of 6.4GB/s, and consumes 130 W. This series uses a proprietary interconnect called NUMAlink3, which is a dual-plane, fat-tree topology that provides 3.2 GB/s bidirectional bandwidth per link. As a non-uniform memory access machine, latency times vary dramatically by distance [11].

4.4 IBM Power4 Supercomputer

The IBM p690 cluster consists of 1.3 GHz Power4 processors, with 8 and 32 processor nodes [9]. Each processor is part of a dual-core chip. Each chip has 3 L2 cache units, connected via a crossbar, for a functional total of 1.41 MB of L2 cache shared by both processors on a chip. The L3 caches are external, running at 1/3 core frequency, and are 32MB in size. Each CPU has a memory bandwidth of 12.8 GB/s, and in the cluster we tested, each processor has 2GB of RAM. The Power4 is rated at 62 W per core, or 124 W per chip. The IBM p690 cluster utilizes an IBM interconnect fabric called Colony, also referred to as the SP Switch2. It has two full-duplex network paths, providing a bidirectional bandwidth of 140 MB/s with a worst-case latency of 15 µs.

5 System Performance Results

Our experimental design highlights the differences in cluster configuration in terms of processor type, the number of processors per node, and the network interconnect. For both POP and CAM, the processor selection is the most important consideration, and the performance of each model in this respect is described in detail below. Most of the clusters we examined use Myrinet D as the MPI network interconnect. One cluster uses a Dolphin 2-D torus interconnect, but its performance is similar to clusters using Myrinet D.

5.1 Community Atmospheric Model

We executed CAM at the T42 resolution used by CCSM on several systems and recorded the execution time reported by the internal timer. Execution time was converted to simulation throughput measured in simulated years per wall clock day (see Fig. 1). The best performing architectures are the SGI Altix 3700 (1500 MHz),

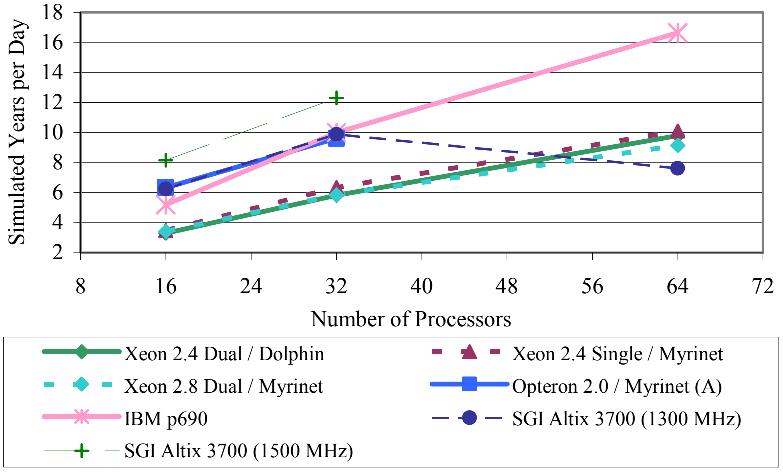

Fig. 1. CAM T42 simulated years per wall clock day by number of processors and platform

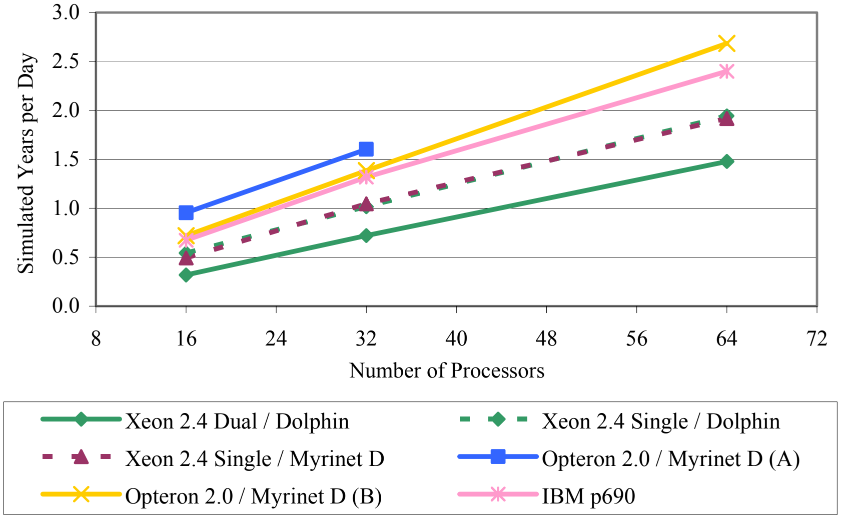

Fig. 2. POP 320x384 simulated years per wall clock day by number of processors and platform

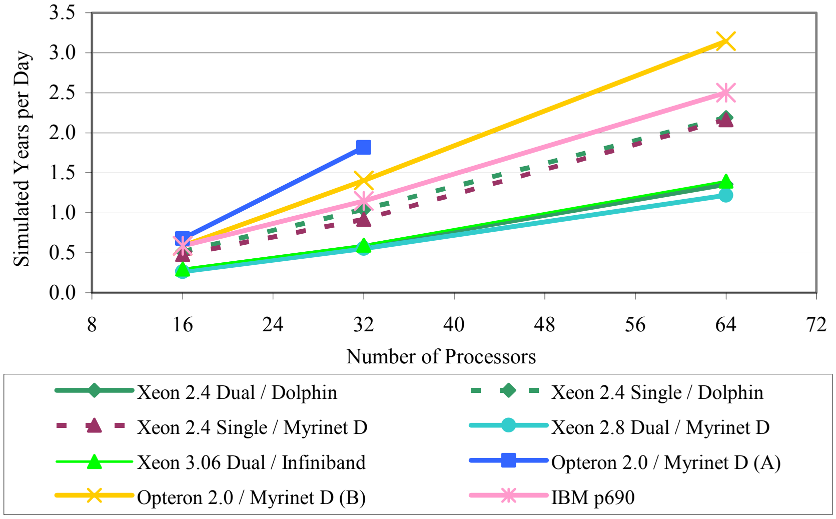

Fig. 3. POP 640x768 simulated years per wall clock day by number of processors and platform

the AMD Opteron cluster, and the IBM p690. The Opteron cluster outperforms or closely matches the performance of the IBM p690 supercomputer, and the Intel Xeon clusters provide the least performance.

5.2 Parallel Ocean Program

We executed POP in two configurations, running at full degree (320x284) and half degree (640x768) resolutions, and recorded the time reported by the internal timer. This omits the startup overhead incurred by the job execution system. The total execution time was converted to job throughput measured in number of simulated years per wall clock day. The POP 320 configuration operates at the resolution used by CCSM in coupled mode, while the POP 640 configuration tests performance in higher resolution situations. For both cases, the AMD Opteron clusters provide the best performance, followed by the IBM p690 supercomputer (see Fig. 2 and Fig. 3). Intel Xeon clusters running one process per node are next, with Intel Xeon clusters running two processes per node providing the worst performance.

These results clearly demonstrate the memory-bound behavior of POP in both resolutions. The processor-memory bus in dual-processor Xeon nodes is particularly ill suited for this code. Running two processes per node on a Xeon cluster provides the lowest performance. Using only one process per node or single processor nodes, however, almost doubles the performance, leading us to conclude that the processor-memory bus is the bottleneck in these systems for this model. We believe that since memory to CPU bandwidth is shared within the dual processor system for the Xeon architecture, using the second processor halves the bandwidth. This does not appear to be the case for the AMD Opteron or p690 architectures. For both POP configurations, the dual processor nodes with only one process per node exhibit slightly better performance than the single processor nodes, as the extra processor is available to handle operating system tasks. Overall, the AMD Opteron clusters provide the best performance.

Of the two Opteron systems examined, one system (labeled A in the figures) outperforms another (labeled B). The vendor believes that the performance difference is caused by the legacy code support requirements of the slower system. That is, the slower system is running in 32-bit kernel mode as required by proprietary kernel modules that have not been ported to a 64-bit version of Linux.

6 Compiler and Optimization Results

While working with computer vendors, processor vendors, and compiler vendors, we noticed a common trend regarding compiler optimization strategy: whichever optimization has been selected is generally considered inferior with respect to those that have yet to be performed. This appears to be a psychological effect. For example, if –O2 optimizations were selected, then aggressive –O3 optimizations are anticipated

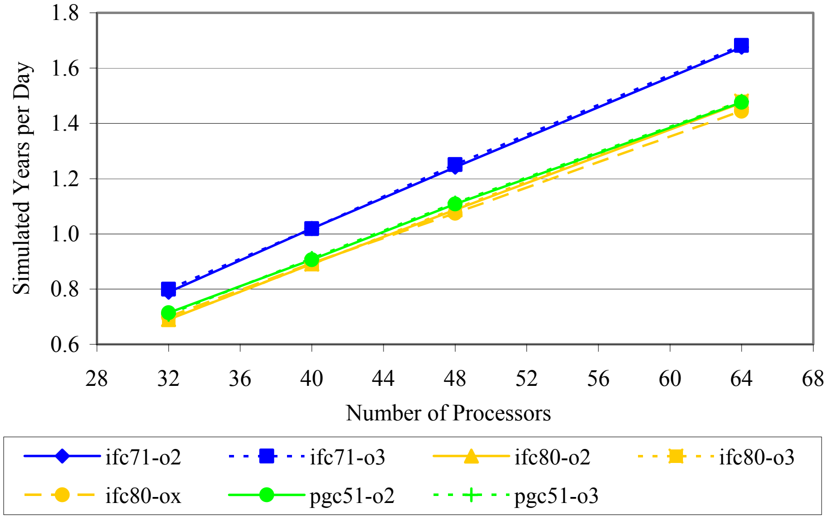

Fig. 4. POP 320x384 simulated years per wall clock day by number of processors and compiler

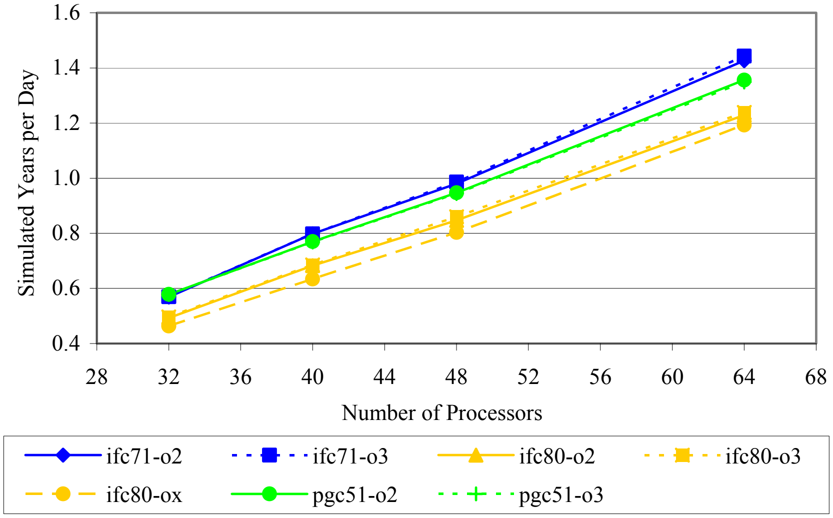

Fig. 5. POP 640x768 simulated years per wall clock day by number of processors and compiler

to provide better performance. If -O3 was selected, then the optimizations are considered too aggressive and -O2 must be examined. We therefore analyzed the impact of compiler selection and optimization options for POP on our Intel 2.4GHz Xeon cluster.

We compiled the POP 320 and POP 640 test suites using the Intel Fortran Compiler (IFC) version 7.1 using -O2 and -O3, IFC version 8.0 using -O2, -O3, and -Nx, and the Portland Group (PGC) Fortran compiler version 5.1 using -fast -O2 and -fast -O3. The tests were run for cases ranging from 8 to 96 processors. POP 320 was run 10 times for each processor count, while POP 640 was run 3 times for each processor count, and the results of each were averaged (see Fig. 4 and Fig. 5).

For both POP 320 and POP 640, IFC 7.1 provides the best performance, followed by PGC 5.1 and then IFC 8.0. For POP 320, the selection of compiler flags -O2 and -O3 for all three compilers is generally unimportant. For various processor counts -O2 and -O3 appear to be arbitrarily slightly faster or slower than each other. Lines on the plots appear to overlap and the difference is not statistically significant. Using IFC 8.0's -nX option, which generates code specifically for the Xeon chip used in the cluster, produces code slower than the -O2 and -O3 options but the difference is extremely small in terms of wall time. We did not perform a fine-grained individual file optimization analysis or more detailed code profiling. In addition, we note that POP appears to be bound by the memory architecture of the underlying platform, and that it may not be the ideal candidate for code optimization analysis. We plan to examine optimization for POP and CAM in more detail in the future.

7 Network Interconnect Comparison

The final element of interest in selecting a cluster for CCSM is the choice of the network interconnect. This is perhaps one of the most important factors in constructing a cluster, as high-performance network systems such as Dolphin and Myrinet are quite expensive when compared to gigabit or fast Ethernet. To examine the impact of the network on the performance of POP and CAM, we ran the software on a Dual Xeon 2.4 GHz cluster using the Dolphin interconnect, a gigabit Ethernet network using a Dell PowerConnect 5224 switch, and a 100Mbps Ethernet network using a HP Procurve 4000M switch.

Our previous results suggested that POP was memory-bound. The results for POP 320 and POP 640 using the Dolphin interconnect and gigabit Ethernet support this, as both networks support executing the program at approximately the same rate of simulated years per day (see Fig. 6 and Fig. 7). That is, neither network appears to be sufficiently slow to impact the execution of the program. In both POP 320 and POP 640, however, execution using 100Mbps Ethernet is clearly slower than when running on the Dolphin and gigabit networks. For POP 640, the execution time for 100Mbps is linear between 8 and 32 processors, presumably as the additional computation outweighs the slower communication network. The less computationally-intensive POP 320 runs show that the simulation rate begins to fall with increasing number of processors.

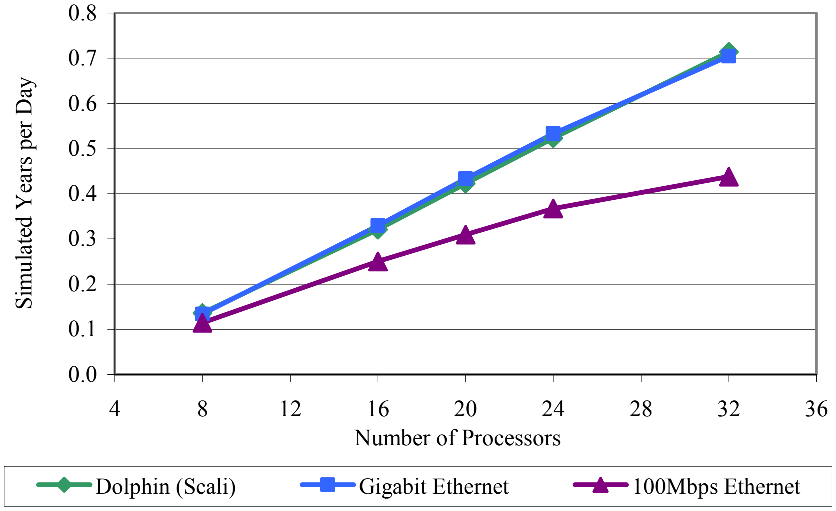

Fig. 6. POP 320x384 simulated years per wall clock day by number of processors and network

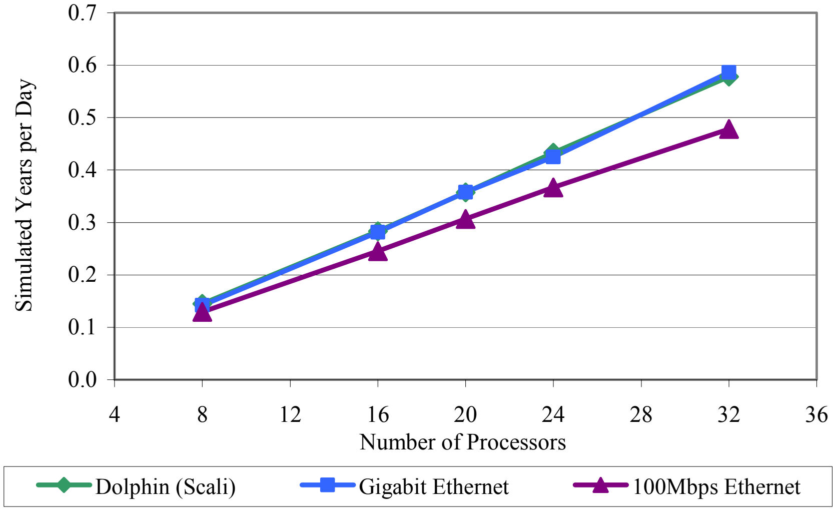

Fig. 7. POP 640x768 simulated years per wall clock day by number of processors and network

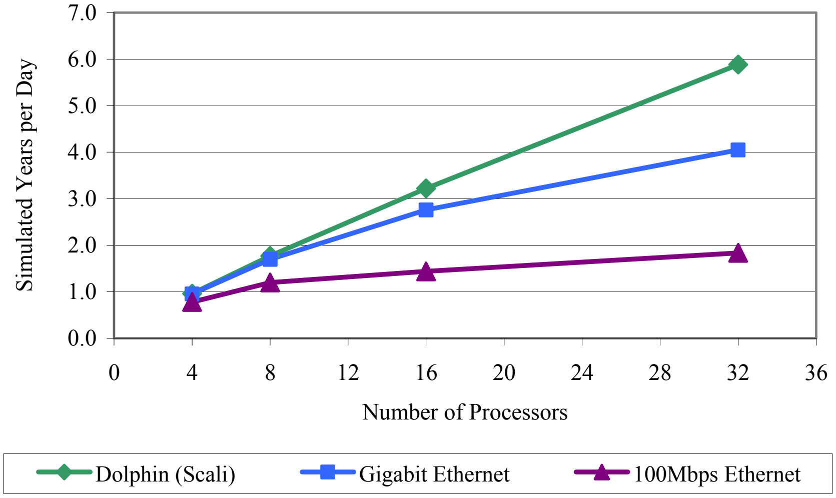

Fig. 8. CAM T42 simulated years per wall clock day by number of processors and network

The CAM T42 results more clearly show the impact of the network interconnect on model simulation rate (see Fig. 8). The 100Mbps Ethernet network demonstrates the worst performance as the integration rate only slightly increases when adding additional processors. The gigabit Ethernet is significantly better than 100Mbps, but the rate begins to flatten with increasing processor counts. The Dolphin interconnect performs best, and our previous experiments show that systems using identical processors with Myrinet exhibit similar performance (see Fig. 1 for CAM, Fig. 2 and Fig. 3 for POP). This leads us to conclude that any high-performance network, such as Myrinet or the Dolphin interconnect, is favorable over commodity gigabit Ethernet solutions.

8 Future Work

This study examined the out-of-box performance of two CCSM component models on commodity cluster systems. While these two models should be instructive of the performance of the entire coupled model, we would have preferred running the entire model on a variety of cluster systems. Our ongoing work includes modifying the build environment of CCSM to function on several other clusters.

We examined CCSM as-is to demonstrate the strengths and weaknesses of current cluster systems running these two models. Once a particular cluster platform has been selected it will then be appropriate to perform parameter studies to determine the optimal tuning for the models' algorithms. As noted by Drake [5], the models contain options useful for tuning performance to a specific processor architecture and interconnect. We intend to explore the effect of these options on a given cluster platform and attempt to leverage the previous work on optimizing CCSM for specific supercomputer architectures.

9 Conclusion

In this paper, we presented simulation throughput measurements for POP and CAM, two dynamical cores constituting part of CCSM, on various cluster and supercomputer systems. The purpose of this analysis is to determine the characteristics of a cluster best suited for further CCSM development, tuning, and performance analysis work. Our results show that clusters based on AMD Opteron processors exhibit performance similar to or better than Xeon clusters and an IBM p690 supercomputer on both POP and CAM.

Acknowledgements

We would like to acknowledge the assistance of the following organizations and individuals: NCAR – John Dennis and Tony Craig. IBM – Jim Tuccillo. Linux Networx – Michael Hall and David Flynn. SGI – Ilene Carpenter. We would also like to thank NCAR, Los Alamos National Laboratory, the Laboratory Computing Resource Center at Argonne National Laboratory, and the University of Texas at Austin for computational time. University of Colorado computer time was provided by equipment purchased under NSF ARI Grant #CDA-9601817 and NSF sponsorship of the National Center for Atmospheric Research.

References

1. Advanced Micro Devices. “AMD Opteron Processor Data Sheet.” http://www.amd.com/us-en/assets/content_type/white_papers_and_tech_docs/23932.pdf, 2003.
2. Buja, L. and Craig, T. *CCSM 2.0.1 User's Guide.* http://www.ccsm. ucar.edu/models/ccsm2.0.1/ccsm/UsersGuide/UsersGuide/UsersGuide.html, 2002.
3. Craig, T. “Climate Modeling at NCAR: Performance Roadblocks.” The Conference on High Speed Computing, Salishan, Oregon. http://www.ccs. lanl.gov/salishan02/craig.pdf, 2002.
4. Dolphin Interconnect Solutions. “Dolphin Interconnect Benchmarks.” http://www.dolphinics. com/products/benchmarks.html, 2004.
5. Drake, J. B., Hammond S., James R., Worley P. H., “Performance Tuning and Evaluation of a Parallel Community Climate Model.” Proceedings of the ACM/IEEE Conference on High Performance Networking and Computing (SC99), Portland, Oregon, 1999.
6. Dorband, J., Kouatchou, J., Michalakes, J. and Ranawake, U. “Implementing MM5 on NASA Goddard Space Flight Center computing systems: a performance study.” IEEE Seventh Symposium on the Frontiers of Massively Parallel Computation, 200-207. http://ieeexplore.ieee.org/iel4/6062/16195/ 00750601.pdf, 1999.

7. Intel. “Xeon Processor with 512KB L2 Cache at 1.80 to 3GHz Datasheet.” http://developer.intel. com/design/xeon/datashts/298642.htm, 2004.
8. Jones, P.W., Worley, P.H., Yoshida, Y., White, J.B. III, Levesque, J. 2003. “Practical Performance Portability in the Parallel Ocean Program (POP), Concurrency Comput. Prac. Exper., in press, 2003.
9. Myricom. “Myrinet Performance Measurements.” http://www.myri.com/myrinet/ performance/index.html, 2003.
10. National Center for Atmospheric Research. “Comparison of CPUs in DSM Systems at NCAR.” http://www.scd.ucar.edu/docs/products/dsm.cpu.specs.html, 2003.
11. SGI. “The SGI Altix 3000 Global Shared-Memory Architecture.” http://www.sgi.com/servers/altix/whitepapers/downloads/altix_shared_memory.pdf, 2004.
12. University of Bern Physics Institute. “KUP Linux Cluster – Performance.” http://www.climate.unibe.ch/cluster/performance.html, 2003.
13. Worley, P. H. “CCSM Component Performance Benchmarking and Status of the Cray X1 at ORNL.” Computing in the Atmospheric Sciences Workshop 2003, Annecy, France. http://www.csm.ornl.gov/~worley/talks/CAS2K3/ CAS2K3.Worley.htm, 2003.