An Evaluation of the Oak Ridge National Laboratory Cray XT3

Sadaf R. Alam, Richard F. Barrett, Mark R. Fahey, Jeffery A. Kuehn, O.E. Bronson Messer, Richard T. Mills, Philip C. Roth, Jeffrey S. Vetter and Patrick H. Worley

International Journal of High Performance Computing Applications 2008 22: 52
DOI: 10.1177/1094342007085019

AN EVALUATION OF THE OAK RIDGE NATIONAL LABORATORY CRAY XT3

Sadaf R. Alam
Richard F. Barrett
Mark R. Fahey
Jeffery A. Kuehn
O. E. Bronson Messer
Richard T. Mills
Philip C. Roth
Jeffrey S. Vetter
Patrick H. Worley

OAK RIDGE NATIONAL LABORATORY
OAK RIDGE, TENNESSEE, 37831 USA
(VETTER@COMPUTER.ORG)

Abstract

In 2005, Oak Ridge National Laboratory (ORNL) received delivery of a 5294 processor Cray XT3. The XT3 is Cray's third-generation massively parallel processing system. The ORNL system uses a single-processor node built around the AMD Opteron and uses a custom chip—called SeaStar—for interprocessor communication. The system uses a lightweight operating system called Catamount on its compute nodes. This paper provides a performance evaluation of the Cray XT3, including measurements for micro-benchmark, kernel, and application benchmarks. In particular, we provide performance results for strategic Department of Energy applications areas including climate, biology, astrophysics, combustion, and fusion. Our results, on up to 4096 processors, demonstrate that the Cray XT3 provides competitive processor performance, high interconnect bandwidth, and high parallel efficiency on a diverse application workload, typical in the DOE Office of Science.

Key words: Cray XT3, Catamount lightweight kernel, performance evaluation, benchmarks, applications

The International Journal of High Performance Computing Applications,
Volume 22, No. 1, Spring 2008, pp. 52–80
DOI: 10.1177/1094342007085019

Figures 1–29 appear in color online: http://hpc.sagepub.com

1 Introduction

Computational requirements for many large-scale simulations and ensemble studies of vital interest to the Department of Energy (DOE) exceed the capabilities of systems currently offered by any U.S. computer vendor. As illustrated in the DOE Scales report (U.S. Department of Energy Office of Science 2003) and the Federal Plan for High-End Computing report (High-End Computing Revitalization Task Force 2004), examples are numerous, ranging from global climate change research to combustion to biology.

Performance of the current class of high performance computer (HPC) architectures is dependent on the performance of all levels of the memory hierarchy, the performance of the system interconnect, and file access performance. The ever-widening gap between processor performance and memory access latency exacerbates this situation. Single processor performance, or the performance of a small system, is relatively simple to determine. However, given reasonable single processor performance, the next important metric is scalability. Here, scalability includes the performance sensitivity to variation in both problem size and the number of processors or other computational resources utilized by a particular application.

ORNL has been evaluating these critical factors on several platforms that include the Cray X1E (Agarwal et al. 2004), the SGI Altix 3700 (Dunigan, Vetter, and Worley 2005), and the Cray XD1 (Fahey et al. 2005). In this paper, we present our evaluation of the Cray XT3 platform with an emphasis on its support for strategic DOE application domains.

2 Cray XT3 System Overview

The XT3 is Cray's third-generation massively parallel processing system. It follows a similar design to the successful Cray T3D and Cray T3E (Scott 1996) systems. The XT3 installed at ORNL uses a single processor node, or processing element (PE). The XT3 connects these processors with a custom interconnect managed by a Cray-designed Application-Specific Integrated Circuit (ASIC) called SeaStar.

2.1 Processing Elements

Each XT3 PE has one Opteron processor with its own dedicated memory and communication resource (see Figure 1). The XT3 has two types of PEs: compute PEs and service PEs. The compute PEs are optimized for application performance and run a lightweight operating system kernel called Catamount. The service PEs run SuSE Linux and are configured for I/O, login, or other system functions.

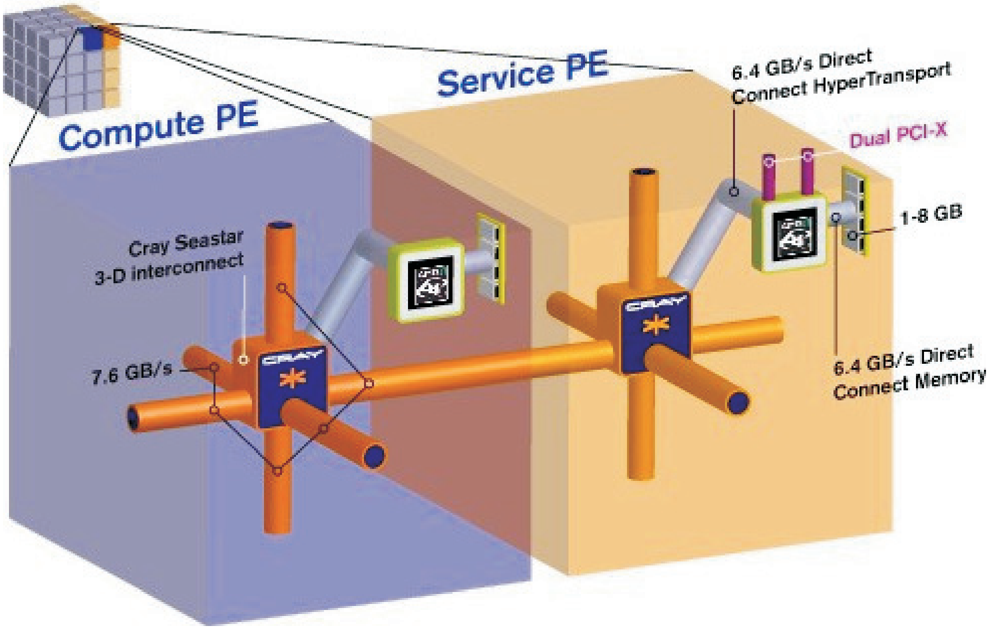

Compute PE
Service PE
6.4 GB/s Direct Connect HyperTransport
Dual PCI-X
1-8 GB
Cray Seastar 3-D interconnect
7.6 GB/s
6.4 GB/s Direct Connect Memory

Fig. 1 **Cray XT3 architecture (*Image courtesy of Cray*).**

The ORNL XT3 uses Opteron model 150 processors. This model includes a single Opteron core, integrated memory controller, three 16 b-wide 800 MHz HyperTransport (HT) links, and L1 and L2 caches. The Opteron core has three integer units and one floating point unit capable of two floating-point operations per cycle (AMD 2004). Because the processor core is clocked at 2.4 GHz, the peak floating point rate of each compute node is 4.8 Gflop/s.

The memory structure of the Opteron consists of a 64 KB 2-way associative L1 data cache, a 64 KB 2-way associative L1 instruction cache, and a 1 MB 16-way associative, unified L2 cache. Each PE has 2 GB of memory but only 1 GB is usable with the kernel used for our evaluation. The memory DIMMs are 1 GB PC3200, Registered ECC, 18 × 512 Mbit parts that support Chipkill. The peak memory bandwidth per processor is 6.4 GB/s. Also, the Opteron 150 has an on-chip memory controller. As a result, memory access latencies with the Opteron 150 are in the 50–60 ns range.

2.2 Interconnect

Each Opteron processor is directly connected to the XT3 interconnect via a Cray SeaStar chip (Figure 1). This SeaStar chip is a routing and communications chip and acts as the gateway to the XT3's high-bandwidth, low-latency interconnect. The PE is connected to the SeaStar chip with a 6.4 GB/s HT link. SeaStar provides six high-speed network links to connect to neighbors in a 3-D mesh topology. Each of the six links has a peak bandwidth of 7.6 GB/s with sustained bandwidth of around 4 GB/s. However, the SeaStar in the ORNL XT3 limits the rate at which the Opteron can inject data onto the XT3 interconnect. In the XT3, the interconnect carries all message passing traffic as well as I/O traffic to the system's Lustre parallel file system.

The ORNL Cray XT3 has 56 cabinets holding 5,212 compute nodes and 82 service nodes. Nodes are connected in a three-dimensional mesh of size 14 × 16 × 24, with torus links in the first and third dimension. The 82 "holes" in the 3-D mesh are due to differences between node packaging for compute nodes and service nodes. Nodes are packaged in modules, with 24 modules per cabinet regardless of node type. Each compute module contains four compute nodes, whereas each service module contains only two service nodes and leaves a two-node "hole" in the 3-D mesh topology.

2.3 Software

The Cray XT3 inherits several aspects of its systems software approach from a sequence of systems developed and deployed at Sandia National Laboratories: ASCI Red

(Mattson et al. 1996), the Cplant (Brightwell et al. 2000; Pedretti et al. 2002), and Red Storm (Brightwell et al. 2005). The XT3 uses a lightweight kernel operating system on its compute PEs, a user-space communications library, and a hierarchical approach for scalable application start-up.

The XT3 uses two different operating systems: Catamount on compute PEs and Linux on service PEs. For scalability and performance predictability, each instance of the Catamount kernel runs only one single-threaded process and does not provide services such as demand-paged virtual memory that could cause unpredictable performance behavior. Unlike the compute PEs, service PEs (i.e. login, I/O, network, and system PEs) run a full SuSE Linux distribution to provide a familiar and powerful environment for application development and for hosting system and performance tools.

The XT3 uses the Portals (Brightwell et al. 2002) data movement layer for flexible, low-overhead inter-node communication. Portals provide connectionless, reliable, in-order delivery of messages between processes. For high performance and to avoid unpredictable changes in the kernel's memory footprint, Portals deliver data from a sending process' user space to the receiving process' user space without kernel buffering. Portals supports both one-sided and two-sided communication models.

Cray provides a Message Passing Interface (MPI; Snir et al. 1998) communication library based on MPICH (Gropp et al. 1996) version 1.2 that uses Portals for data transfer. We used this implementation in all of our parallel benchmark and application experiments on the XT3.

The primary math library on the XT3 is the AMD Core Math Library (ACML). It incorporates BLAS, LAPACK and FFT routines, and is optimized for high performance on AMD-based platforms.

3 Evaluation Overview

Throughout this report, we compare XT-3 performance with that of several other machines, many of which have been rigorously evaluated using important DOE applications as part of the Early Evaluation Project at ORNL. Recent evaluations have included the Cray X1 (Dunigan, Vetter, White et al. 2005), the SGI Altix 3700 (Dunigan, Vetter, and Worley 2005), and the Cray XD1 (Fahey et al. 2005). The primary goals of these evaluations are to 1) determine the most effective approaches for using each system, 2) evaluate benchmark and application performance, both in absolute terms and in comparison with other systems, and 3) predict scalability, both in terms of problem size and in the number of processor. The basic hardware specifications for these and other comparison machines are as follows:

- Cray X1 at ORNL: 512 multi-streaming processors (MSP), each capable of 12.8 Gflop/s. Each MSP is comprised of four single streaming processors (SSPs), operating at 800 MHz for the vector units and 400 MHz for the scalar unit. A node consisting of 4 MSPs shares 16 GB of memory, and 4 node sets are fully connected via a 2-D torus between subsets.
- Cray X1E at ORNL: An upgrade of the above X1, resulting in 1024 multi-streaming processors (MSP), each capable of 18 Gflop/s. Each MSP is comprised of four single streaming processors (SSPs) operating at 1.13 GHz for the vector units and 565 MHz for the scalar unit. A node consisting of 4 MSPs shares 8 GB of memory, and 4 node sets are fully connected via a 2-D torus between subsets.
- Cray XD1 at ORNL: 144 AMD 2.2 GHz Opteron 248 processors, each capable of 4.4 Gflop/s, configured as 72, 2-way SMPs, each with 8 GB of memory, connected by the Cray RapidArray fabric.
- Earth Simulator: 640 8-way vector SMP processor nodes (PNs) connected by a 640 × 640 single-stage crossbar interconnect. Each PN is a system with a shared memory, consisting of 8 vector-type arithmetic processors (APs) and a 16-GB main memory system (MS). The peak performance of each AP is 8 Gflop/s. Each AP consists of a 4-way super-scalar unit (SU) and a vector unit (VU). The AP operates at a clock frequency of 500 MHz with some circuits operating at 1 GHz.
- Opteron cluster at Combustion Research Facility/Sandia (CRF/S): 286 AMD 2.0 GHz Opteron processors, each capable of 4 Gflop/s, with 1 GB of memory per processor, configured as 143, 2-way SMPs with an Infiniband interconnect.
- SGI Altix at ORNL: 256 Itanium2 1.5 GHz processors, capable of 6 Gflop/s, connected by a NUMAlink3 switch, creating a single global shared memory system of 2 TB of shared memory.
- SGI Altix at the National Aeronautic and Space Administration (NASA): Two Altix 3700 BX2 nodes, each consisting of 512 Itanium2 1.6 GHz processors, each capable of 6.4 Gflop/s, connected by a NUMA-link4, creating a single global shared-memory system of 4 TB of memory.
- HP/Linux Itanium2 cluster at the Pacific Northwest National Laboratory (PNNL): 1960 Itanium2 1.5 GHz processors, each capable of 6 Gflop/s. System is configured as 980, 2-way SMP nodes with a Quadrics QsNetII interconnect. 574 compute nodes have 8 GB of memory and 366 compute nodes have 6 GB of memory.
- IBM SP at NERSC: 184 16-way Nighthawk II SMP nodes and an SP Switch2. Each node has two interconnect interfaces, and between 16 and 64 GB of shared memory. The processors are the 375 MHz POWER3-II, capable of 1.5 Gflop/s.

- IBM p690 cluster at ORNL: 27 32-way p690 SMP nodes, each with 32 GB of shared memory, and an HPS interconnect. Each node has two HPS adapters, each with two ports. The processors are 1.3 GHz POWER4, capable of 5.2 Gflop/s.
- IBM p575 cluster at the National Energy Research Supercomputer Center (NERSC): 122 8-way p575 SMP nodes, each with 32 GB of memory and an HPS interconnect with 1 two-link adapter per node. The processors are 1.9 GHz POWER5, capable of 7.6 Gflop/s.
- IBM BlueGene/L at ANL: a 1024-node BlueGene/L system at Argonne National Laboratory. Each BlueGene/L processing node consists of an ASIC with two PowerPC processor cores (2.8 Gflop/s per processor or 5.6 Gflop/s per node), on-chip memory and communication logic. Experiments were run in either "virtual node" (VN) mode, where both processors in the BG/L node were used for computation, or Co-Processor (CP) mode, where one processor was used for computation and one was used for communication. Each node has 512 MB of memory.

4 Micro-Benchmarks

The objective of micro-benchmarking is to characterize the performance of the specific architectural components of the platform. We use both standard benchmarks and customized benchmarks. Using standard benchmarks allows consistent historical comparisons across platforms. The custom benchmarks permit the unique architectural features of the system (e.g. global address space memory) to be tested.

Traditionally, our micro-benchmarking focuses on the arithmetic performance, memory-hierarchy performance, task and thread performance, message-passing performance, system and I/O performance, and parallel I/O. However, because the ORNL XT3 uses single-core processors and a lightweight operating system, we focus only on these areas:

- Memory-hierarchy performance, including all levels of cache.
- Arithmetic performance, including varying instruction mix, identifying what limits computational performance.
- Message-passing performance, including one-way (ping-pong) messages, message exchanges, and collective operations, message-passing hotspots, and the effect of message passing on the memory subsystem.

4.1 Memory Performance

Table 1
STREAM Triad performance.

System	Triad bandwidth (GB/s)
Cray XT3 (ORNL)	4.9
Cray XD1 (ORNL)	4.1
Cray X1E MSP (ORNL)	23.1
IBM p690 (ORNL)	2.1
IBM POWER5 (NERSC)	4.0
SGI Altix (ORNL)	3.7

Table 2
Latency to main memory.

Platform	Latency to main memory (ns)
Cray XT3/Opteron 150/2.4 GHz	51.41
Cray XD1/Opteron 248/2.2 GHz	86.51
IBM p690/POWER4/1.3 GHz	90.57
Intel Xeon/3.0 GHz	140.57
Intel Itanium2 (Altix)/1.5 GHz	140.

The memory performance of current architectures is a primary factor for performance on scientific applications. Table 1 illustrates the differences in measured memory bandwidth for one processor on the STREAM Triad benchmark. The very high bandwidth of the Cray X1 MSP clearly dominates the other processors, but the Cray XT3's Opteron has the highest bandwidth of the other microprocessor-based systems. The XT3 bandwidth we report was measured in April 2006 using the PGI 6.1 compiler. The observed bandwidth is sensitive to compiler, compiler flags, and data placement. A STREAM Triad bandwidth of 5.1 GB/s was measured on the ORNL XT3 using the Pathscale compiler, but that compiler is not currently supported on the ORNL XT3.

As discussed earlier, the choice of the Opteron model 150 was motivated in part to provide low access latency to main memory. Our measurements (Table 2) revealed that the Opteron 150 has lower latency than the Opteron 248 configured as a 2-way SMP in the XD1. Furthermore, it has considerably smaller latency than either the POWER4 or the Intel Xeon and Itanium2, which support multiprocessor configurations (and hence must include logic for maintaining cache coherence that contributes to the main memory access latency). The Itanium2 latency reported in the table reflects the minimum latency for a 64P system.

The memory hierarchy of the XT3 compute node is obvious when probed with the CacheBench tool (Mucci, London, and Thurman 1998). Figure 2 shows that the

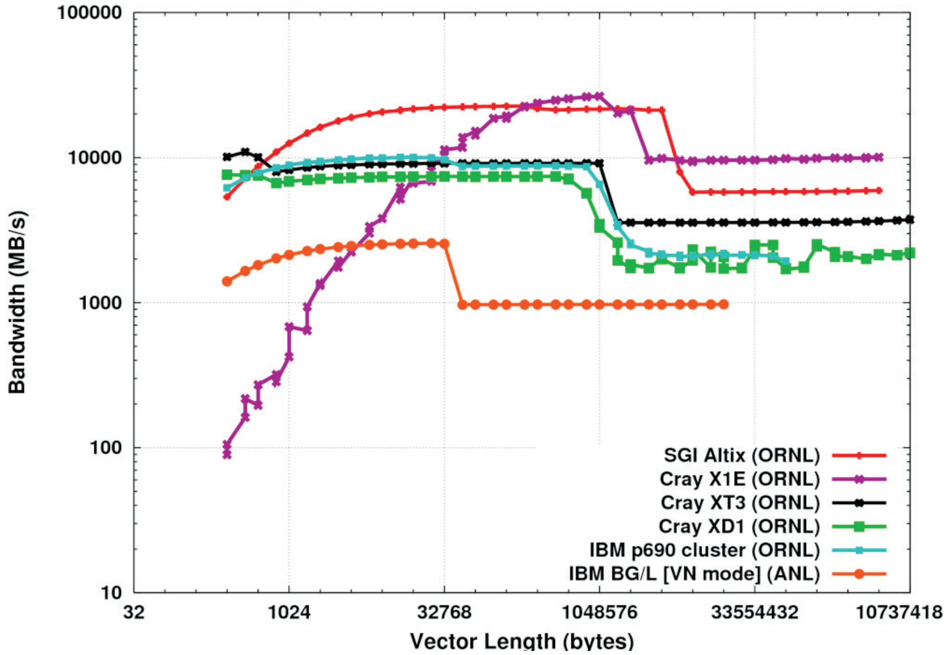


Fig. 2 CacheBench read results for a single XT3 compute node.

system reaches a maximum of approximately 9 GB/s when accessing vectors of data in the L2 cache. When data is accessed from main memory, the bandwidth drops to about 3 GB/s.

4.2 Arithmetic Performance

We use a collection of micro-benchmarks to characterize the performance of the underlying hardware, compilers, and software libraries for common operations in computational science. The micro-benchmarks measure computational performance, memory hierarchy performance, and inter-processor communication. Figure 3 compares the double-precision floating point performance of a matrix multiply (DGEMM) on a single processor using the vendors' scientific libraries. In our tests, the XT3 with the ACML 3.0 library achieved its highest DGEMM performance for matrices of order 1600; the observed performance was 4396 MB/s, approximately 91.6% of the Opteron 150's peak.

Fast Fourier Transforms are another operation important to many scientific and signal processing applications. Figure 4 plots one-dimensional FFT performance using the vendor math library, where initialization time is not included. The XT3's Opteron is outperformed by the SGI Altix's Itanium2 processor for all vector lengths examined, but does better than the POWER4 processor in the p690 and better than the X1E for short vectors.

In general, our micro-benchmark results suggest *performance stability* from the XT3 compute nodes, in that they may not be the best performing for any one of the micro-benchmarks but they perform reasonably well on all of them.

4.3 MPI

Because of the predominance of the message-passing programming model in contemporary scientific applications, examining the performance of message-passing operations is critical to understanding a system's expected performance characteristics when running full applications. Because most applications use the Message Passing Interface (MPI) library (Snir et al. 1998), we evaluated the performance of each vendor's MPI implementation. For our evaluation, we used the Intel MPI Benchmark (IMB) suite, version 2.3 with each system vendor's MPI implementation. In general, the MPI performance of the Cray XT3 was observed to be unexceptional compared to the other systems we tested, and was even observed to be significantly worse for some collective calls with small messages.

Figure 5 and Figure 6 show the latency and bandwidth, respectively, for the IMB PingPong benchmark. Like all

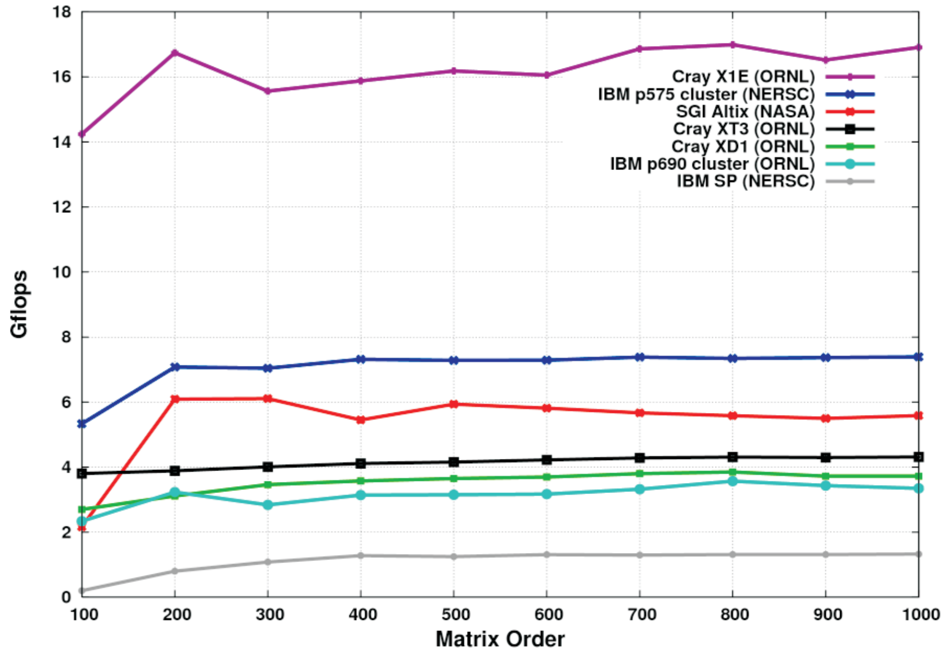


Fig. 3 **Performance of matrix multiply.**

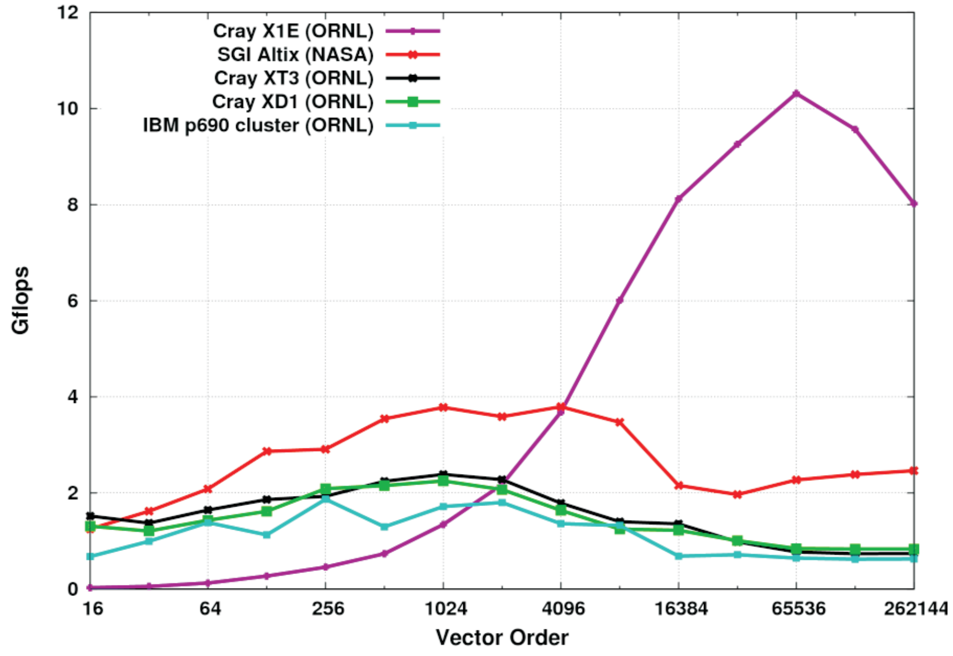


Fig. 4 **Performance of 1-D FFT.**

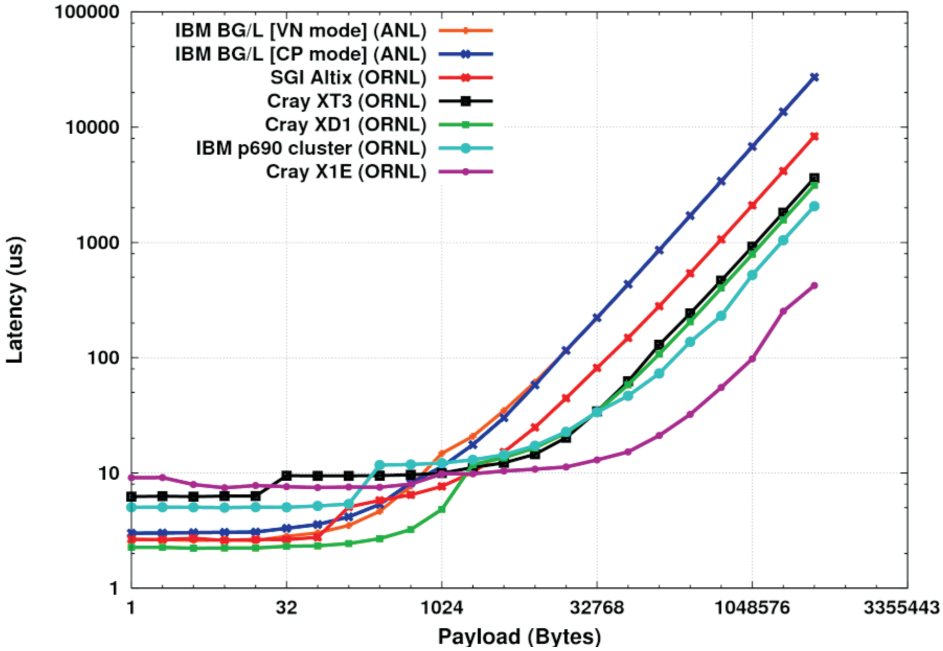


Fig. 5 IMB PingPong benchmark latency.

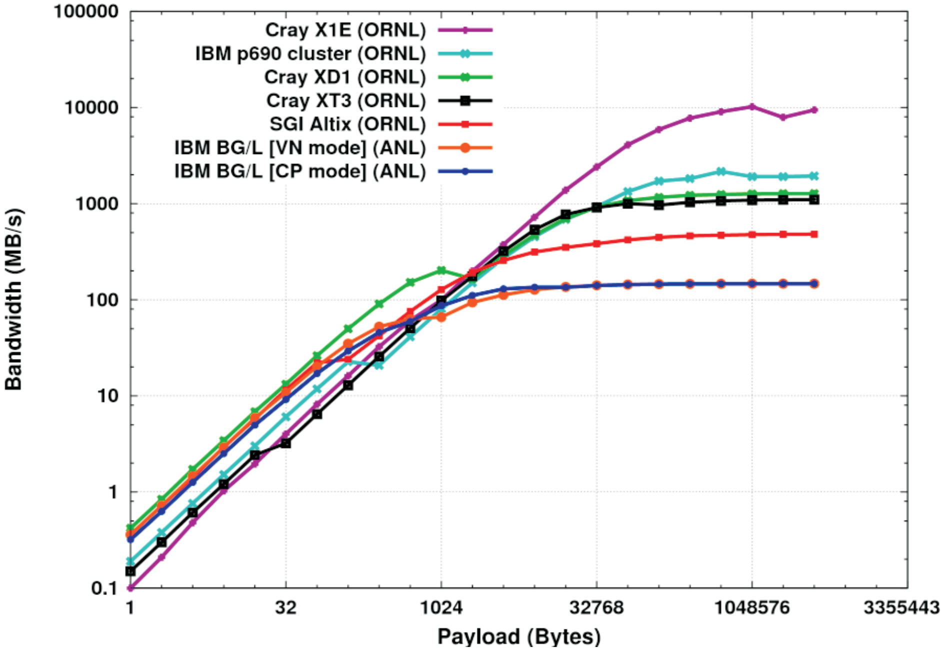


Fig. 6 IMB PingPong bandwidth.

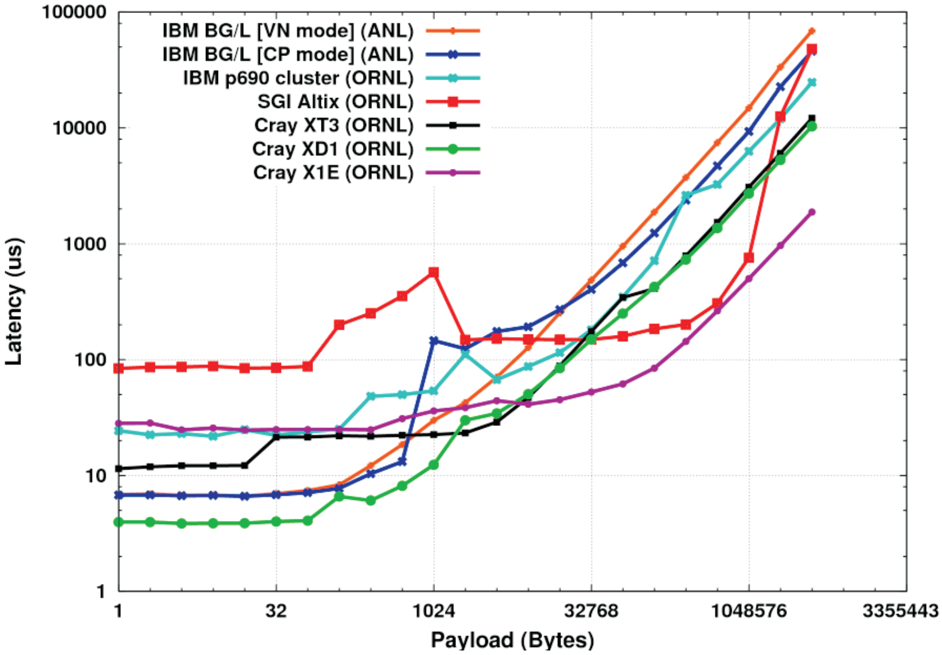


Fig. 7 **IMB Exchange benchmark latency at 128 tasks.**

IMB benchmarks that report both bandwidth and latency, the PingPong bandwidth is calculated from the measured latency so the two figures are different perspectives on the same data. The null message latency on the XT3 was observed to be just over 6 µs, and the maximum bandwidth was 1104 MB/s. Because this observed bandwidth was limited by a network injection problem in the SeaStar chips used in the ORNL XT3, the bandwidth is much less than the reported HyperTransport sustained bandwidth (4 GB/s). The XT3 performance was among the worst of the systems tested for messages smaller than 1 KB, but rises to the middle of the pack for larger messages. These results were collected in April 2006; the short-message latencies are 3% to 5% higher than the latency we measured in November 2005, but the maximum bandwidth is nearly the same. Because the operating system, MPI implementation, and SeaStar firmware have been modified since November 2005, we cannot say with certainty where to attribute the additional overhead. These changes are predominantly visible as changes in latency, from roughly 22 µs in mid-2005 to 6 µs in mid-2006. Peak bandwidth has not changed significantly over the same period.

Figure 7 and Figure 8 show the latency and bandwidth, respectively, for the Intel Exchange benchmark on 128 processors, the largest number of MPI tasks we could obtain across all of our test systems. The Exchange benchmark is intended to represent the behavior of a code performing boundary-exchange communication. In this benchmark, each task performs one-dimensional nearest-neighbor communication using MPI_Isend, MPI_Recv, and MPI_Waitall. The benchmark program measures the time required to send data to its left and right neighbor and to receive data sent by those neighbors. Similar to the IMB PingPong benchmark, bandwidth is computed from the observed latency but considers that each process sends two messages and receives two messages. Because this benchmark measures latency and bandwidth using point-to-point MPI operations when all MPI tasks are communicating, it is a more realistic test of a system's MPI performance than the PingPong benchmark for a large class of scientific applications. For the largest number of MPI tasks we tested on the XT3 (4096), we observed an average latency of 11.99 µs for 4-byte messages and a maximum bandwidth of 1262 MB/s for 512 KB messages. The Cray XD1 showed the best Exchange performance of the systems we tested for messages smaller than 2 KB, whereas we observed the best performance for larger messages with the Cray X1E. The SGI Altix produced interesting behavior with this benchmark, showing performance that was the worst with small message sizes and approached the best with some large

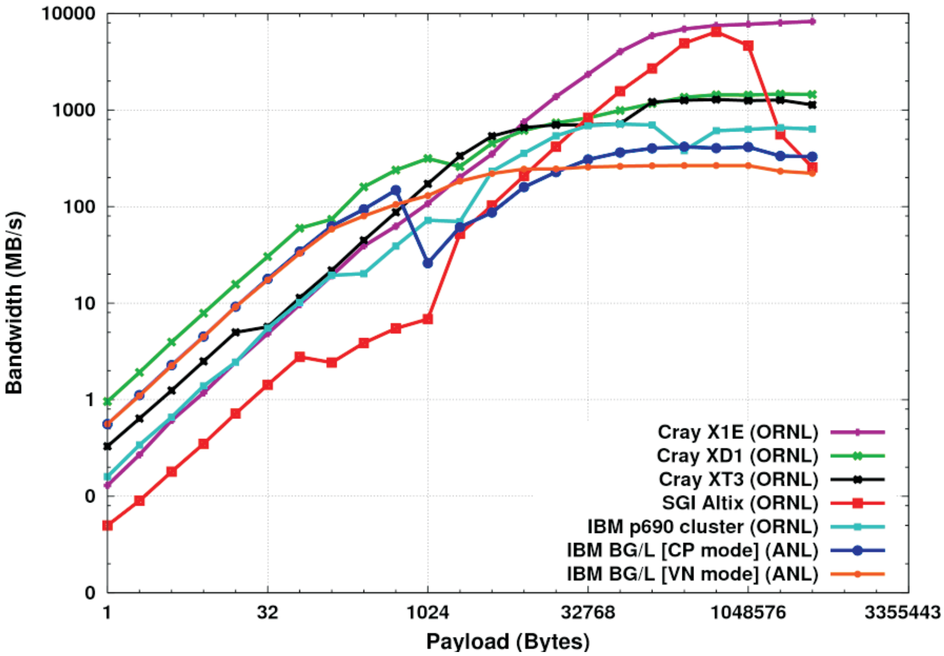


Fig. 8 IMB Exchange benchmark bandwidth at 128 tasks.

message sizes, and showed non-smooth transitions between the two regimes. We suspect this behavior may be due to contention with other jobs running on the Altix when we ran our benchmark.

The MPI_Allreduce operation is particularly important for several DOE simulation applications; for some applications, it is used several times per simulation timestep. Its blocking semantics also require that all tasks wait for its completion before continuing, so the latency of this operation is an important factor with regard to application scalability. The IMB Allreduce benchmark tests the latency of the MPI_Allreduce operation. (The IMB developers do not consider bandwidth to be a well-defined concept for MPI collective operations, so the IMB collective benchmarks including Allreduce do not report a bandwidth measurement.) Our IMB Allreduce latency results are shown in Figure 9. The Cray XT3 Allreduce performance is the worst among the systems tested for small messages, whereas the Cray XD1 and X1E performed very well for small messages and the X1E was superior for messages larger than 2 KB. Note that the excellent performance of Allreduce for small messages on the X1E reflects the implementation of the collective using a transport layer with latency significantly lower than that of MPI.

5 HPC Challenge

HPC Challenge (HPCC; Luszczek et al. 2005) () is a collection of tests designed to benchmark several aspects of a parallel computing system, including its ability to solve a linear system (HPL), to multiply floating point matrices (DGEMM), and to perform updates to random locations in memory (RandomAccess). Several of the tests are run in sequential, embarrassingly parallel, and parallel modes. In this section, we include a sampling of our comprehensive XT3 HPCC results (Kuehn and Wichmann 2006) that complement our single-node and MPI benchmark results.

Figure 10 compares the network latency results for ping-pong, natural-ring, and random-ring operations on the ORNL Cray XT3 across a range of processor counts. Minimum ping-pong latency is (and should be) constant across the range of scales represented in this chart, the other latency curves demonstrate latency increases as the node count is increased. This is expected because for a 3-D torus (or in this case, a 3-D mesh with torus links in the first and third dimension), the number of hops across the working set should increase roughly with the cube root of the number of processors. However, the chart does exhibit an irregularity. Because the natural ring communication pattern on a machine like the XT3 should represent near-

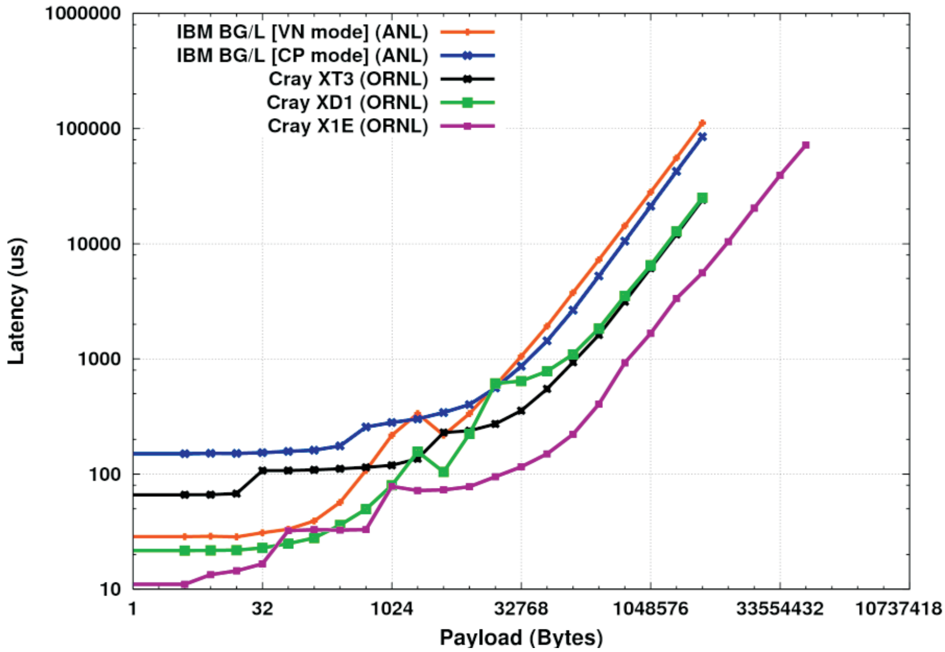


Fig. 9 IMB Allreduce benchmark latency at 128 tasks.

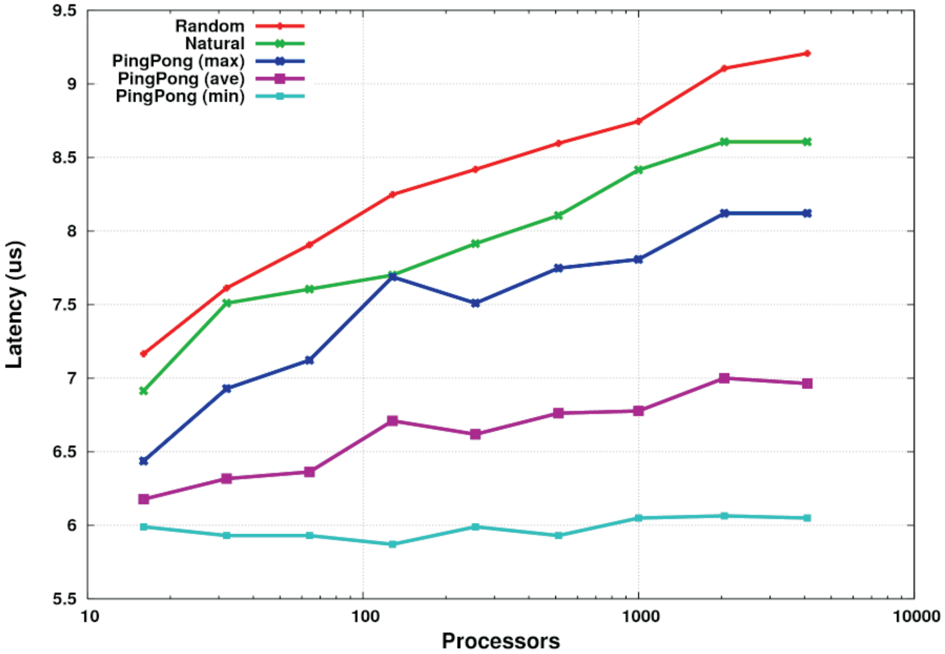


Fig. 10 HPCC network latency results for the ORNL Cray XT3.

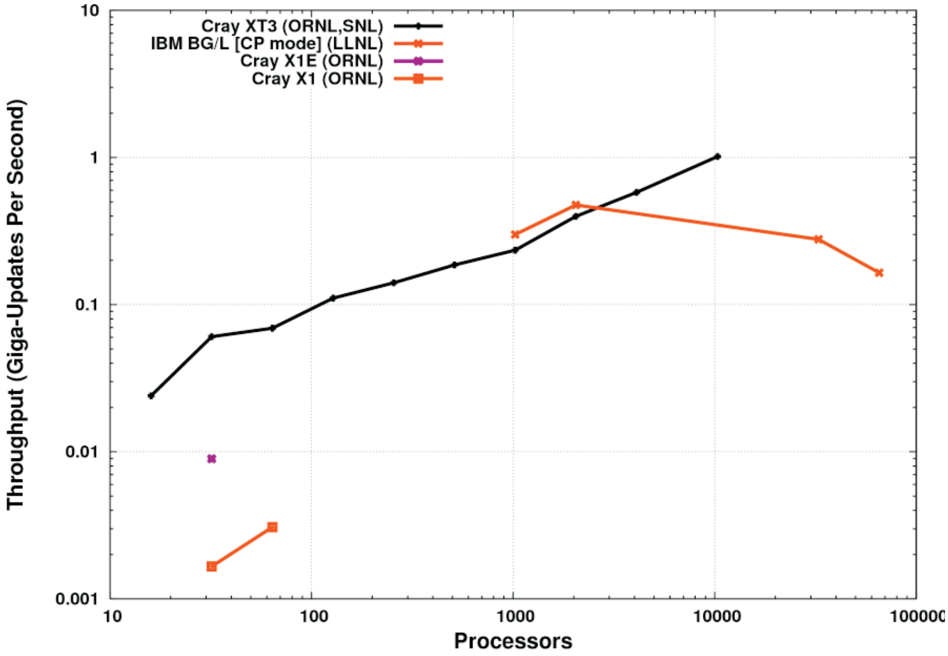


Fig. 11 HPCC MPI RandomAccess throughput.

est node communication, the natural-ring latency should track lower on the chart with the minimum ping-pong latency. However, in this chart the natural ring latency tracks closer to the random-ring latency. The random ring latency represents a communication pattern between nodes which are randomly distributed and thus as the node count increases, the likelihood of randomly chosen nodes being topologically adjacent decreases; it becomes more likely that the random nodes are topologically remote from each other. Because the observed communication latency between nodes which are nominally adjacent in the MPI communicator tracks closely with the observed latency for the random-ring pattern which represents communication latency between nodes which are randomly distributed, this suggests that the job placement is flawed. MPI communicator nodes that are supposed to be topologically adjacent are in fact more randomly spread through the machine. This results from the nodes being numbered according to physical location. However, the wiring topology is serpentine in each dimension to minimize the cable lengths, meaning that nodes that are physically adjacent are topologically remote from each other. Cray has been advised that the batch scheduler's naïve allocation of physically adjacent nodes to a job is, thus, not optimal. Weisser et al. (2006) noted that performance can be improved by making the scheduler aware of the node's physical topology and position within the interconnect. Variations resulting from network topology and node numbering do not seem to affect the observed PingPong bandwidth, which was throttled by the SeaStar network injection bottleneck.

Figure 11 shows unoptimized HPCC MPI RandomAccess results for several systems including the Cray XT3. The BlueGene/L data was obtained from the HPCC web-based results repository and was measured on the systems at Lawrence Livermore National Laboratory using a different algorithm from the standard case that takes advantage of the topology of the interconnect, providing a benefit similar to Strassen's Algorithm over a naïve matrix multiply algorithm. Most of the XT3 results were measured on the ORNL XT3, but the data point for over 10,000 processors was obtained via the HPCC web-based results repository from the Red Storm system at Sandia National Laboratories. The XT3's RandomAccess performance is best understood relative to other systems shown on the chart. For small numbers of PEs, the XT3's lower network latency allows it to outperform the X1 and X1E; in effect, the benchmark ranks the systems by network latency. (These results are obtained using an MPI implementation of the RandomAccess benchmark; using a Unified Parallel C implementation, we observe much better RandomAccess performance than is shown in Figure 11.)

At the upper end of the scale, we observe a crossover between XT3 and BlueGene/L. For smaller processor counts (1024–2048 PEs), BlueGene/L performs slightly better than XT3, with a latency that is significantly less than XT3's. However, because of the restricted memory size of BlueGene/L nodes, the global table size used in the benchmark is significantly smaller. At higher scales (10K–64K PEs) the latencies on the two systems are closer together. Even at larger processor counts, the primary factor dominating BlueGene/L performance is network contention. Even though RandomAccess isn't a "network bandwidth" test, at larger scales the contention induced by the random traffic on BlueGene/L's lower bandwidth network links limits its global performance. In this sense, XT3's higher bandwidth links are less susceptible to the traffic contention that limits BlueGene/L. The lack of intermediate points on the BlueGene/L scaling curves prevents us from drawing strong conclusions about the exact crossover point, but it appears to occur somewhere in the 4K–8K PEs range.

6 Applications

Insight into the performance characteristics of low-level operations is important to understand overall system performance, but because a system's behavior when running full applications is the most significant measure of its performance, we also investigate the performance and efficiency of full applications relevant to the DOE Office of Science in the areas of global climate, fusion, chemistry, and bioinformatics. The evaluation team worked closely with principal investigators leading the Scientific Discovery through Advanced Computing (SciDAC) application teams to identify strategic applications.

6.1 CAM

The Community Atmosphere Model (CAM) is a global atmosphere model developed at the National Science Foundation's National Center for Atmospheric Research (NCAR) with contributions from researchers funded by DOE and by NASA (Collins, Rasch et al. 2006). CAM is used in both weather and climate research. In particular, CAM serves as the atmospheric component of the Community Climate System Model (CCSM; Collins, Bitz et al. 2006).

CAM is a mixed-mode parallel application code, using both MPI (Snir et al. 1998) and OpenMP protocols (Dagum and Menon 1998). CAM is characterized by two computational phases: the dynamics, which advances the evolution equations for the atmospheric flow, and the physics, which approximates subgrid phenomena such as precipitation processes, clouds, long- and short-wave radiation, and turbulent mixing. Control moves between the dynamics and the physics at least once during each model simulation timestep. The number and order of these transitions depend on the numerical algorithm in the dynamics.

CAM includes multiple *dynamical cores (dycores)*, one of which is selected at compile-time. Three dycores are currently supported: the spectral Eulerian solver from CCM (Kiehl et al. 1998), a spectral semi-Lagrangian solver (Williamson and Olson 1994), and a finite volume semi-Lagrangian solver (Lin 2004). The three dycores do not use the same computational grid. An explicit interface exists between the dynamics and the physics, and the physics data structures and parallelization strategies are independent from those in the dynamics. A dynamics–physics coupler moves data between data structures representing the dynamics state and the physics state.

For our evaluation we ported and optimized CAM versions 3.0p1 and 3.1, available for download from http://www.ccsm.ucar.edu/, as described by Worley (2006). We used two different benchmark problems. The first uses the spectral Eulerian dycore with CAM 3.0p1, a 128 × 256 (latitude × longitude) horizontal computational grid covering the sphere, and 26 vertical levels. This problem, which is referred to as T85L26, is a common production resolution used with the CCSM. The second benchmark uses the finite volume (FV) dycore with CAM 3.1, a 361 × 576 horizontal computational grid, and 26 vertical levels. The CCSM community is currently transitioning from the spectral Eulerian dycore to the FV dycore in production runs. This problem resolution, referred to as the "D-grid," is much larger than is envisioned for near-term production climate runs, but represents a resolution of interest for the future.

Figure 12 shows a platform comparison of CAM throughput for the T85L26 benchmark problem. The spectral Eulerian dycore supports only a one-dimensional latitude decomposition of the computational grid, limiting MPI parallelism to 128 processes for this computational grid. OpenMP can be used to exploit additional processors, but the XT3 cannot take advantage of this. By these results, the X1E is 2.5 times faster than the XT3 and the XT3 is 2.1 times faster than the p690 cluster for the same number of processors. Performance on the XT3 and the p575 cluster are similar for small processors counts. OpenMP parallelism gives the p575 cluster an advantage for large processor counts. While performance is reasonable on the XT3 for this benchmark, the limited scalability in the code does not take good advantage of the size and scalability of the ORNL XT3 system.

Figure 13 shows a platform comparison of CAM throughput for the D-grid benchmark problem. The FV dycore supports both a one-dimensional (1-D) latitude decomposition and a two-dimensional (2-D) decomposition of the computational grid. The 2-D decomposition is

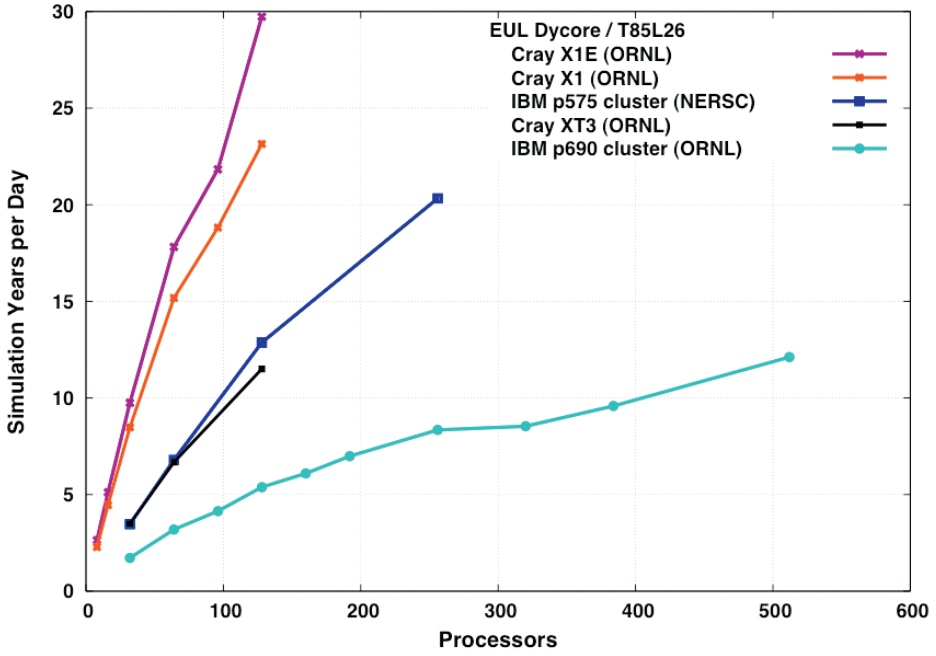


Fig. 12 Platform comparisons using CAM T85L26 benchmark.

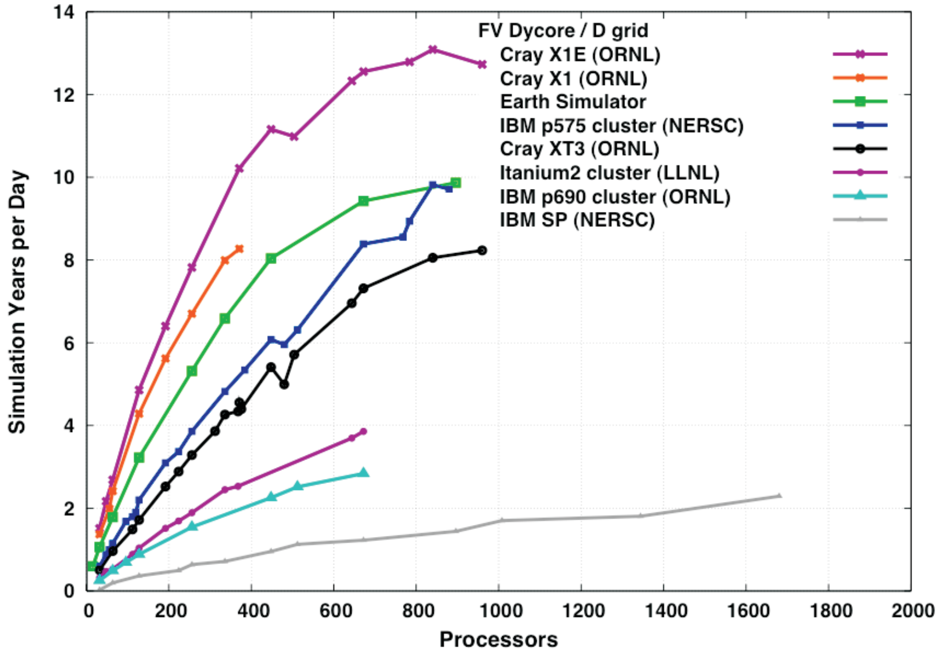


Fig. 13 Platform comparisons using CAM D-grid benchmark.

over latitude and longitude during one phase of the dynamics and over latitude and vertical in another phase, requiring two remaps of the domain decomposition each timestep. For small processor counts the 1-D decomposition is faster than the 2-D decomposition, but the 1-D decomposition must have at least three latitudes per process and is therefore limited to a maximum of 120 MPI processes for the D-grid benchmark. Using a 2-D decomposition requires at least three latitudes and three vertical layers per process, and therefore is limited to 120 × 8, or 960, MPI processes for the D-grid benchmark. OpenMP can again be used to exploit additional processors. OpenMP is used by the Earth Simulator and the IBM systems, but not by the Cray systems. Each data point in Figure 13 represents the performance on the given platform for the given processor count after optimizing over the available virtual processor grids defining the domain decomposition and after optimizing over the number of OpenMP threads per MPI process. For the D-grid benchmark the XT3 performs significantly better than the Itanium2 cluster and the IBM SP and p690 cluster systems. The XT3 performance lags that of the p575 cluster by 10 to 20 percent.

Figure 14 shows plots of the wallclock seconds per simulation day for the dynamics and the physics of the XT3 and the p575 cluster; one plot with linear–log axes (Figure 14a) and one with linear–linear axes (Figure 14b). The IBM system uses OpenMP to decrease the number of MPI processes required, allowing the IBM system to use the 1-D domain decomposition in all experiments. The physics costs are identical up through 200 processors. The performance difference between the p575 cluster and the XT3 for larger processor counts is almost entirely due to the runtime difference in computing a global sum and a write to standard output that occurs each timestep. In contrast, dynamics is always faster on the p575, decreasing from a 40% advantage for small processor counts to a 25% advantage for large processor counts. The performance difference for large processor counts appears to be the result of a higher cost of writes to standard output on the XT3, which increases in relative importance with larger processor counts. For smaller processor counts the reason for the performance difference is not obvious. However, the ratio of peak per processor between the XT3 and p575 is 58%, so some of the performance advantage could result from the processor speed advantage. This is still under investigation.

6.2 Parallel Ocean Program (POP)

The Parallel Ocean Program (POP; Jones et al. 2005) is the ocean component of CCSM (Blackmon et al. 2001). The code is based on a finite-difference formulation of the three-dimensional flow equations on a shifted polar grid. In its high-resolution configuration, 1/10-degree horizontal resolution, the code resolves eddies for effective heat transport and the locations of ocean currents.

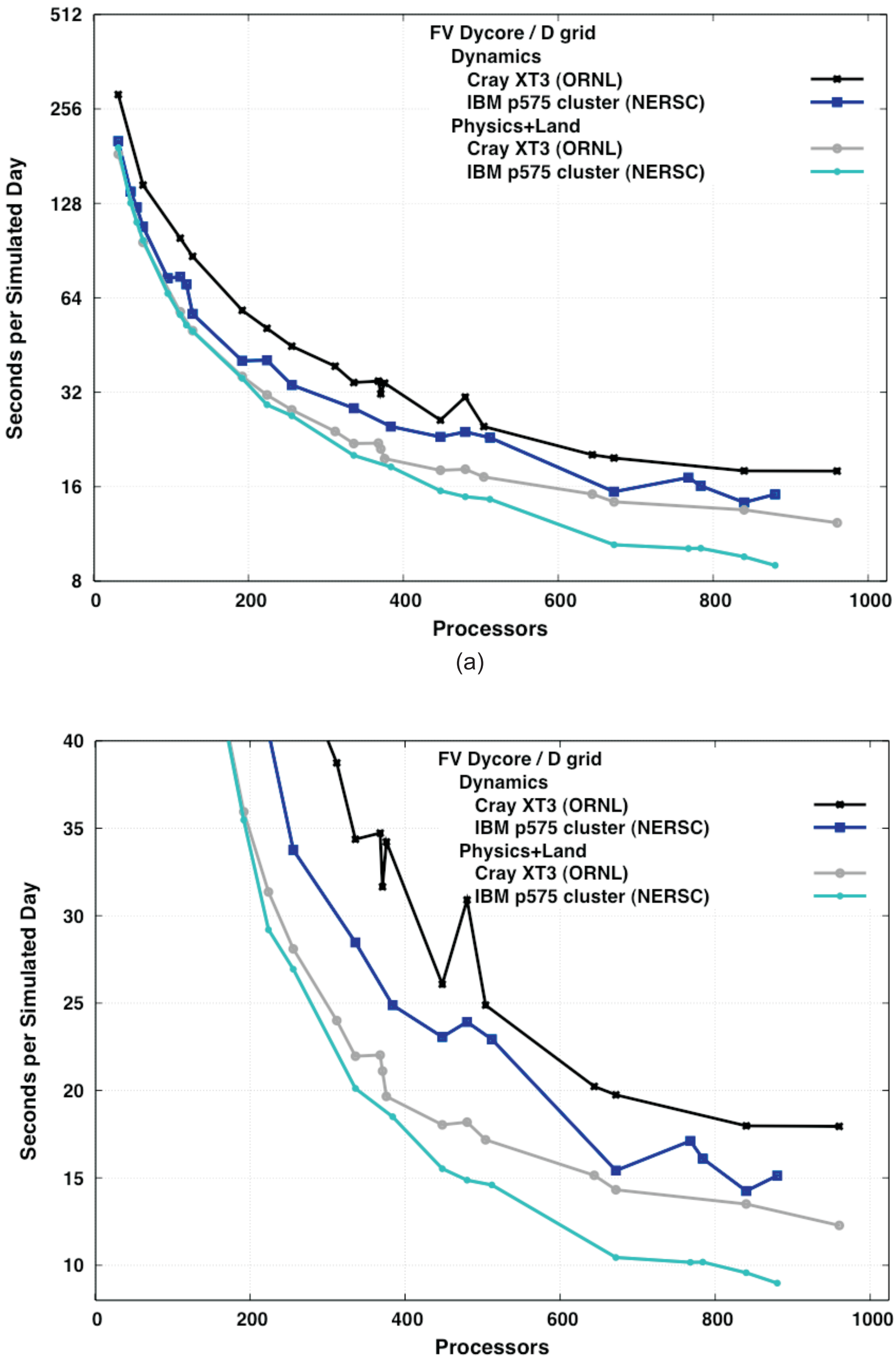
POP performance is characterized by the performance of two phases: baroclinic and barotropic. The baroclinic phase is three dimensional with limited nearest-neighbor communication and typically scales well on all platforms. In contrast, runtime of the barotropic phase is dominated by the iterative solution of a two-dimensional, implicit system using a conjugate gradient method. The performance of the barotropic solver is very sensitive to network latency and typically scales poorly on all platforms.

For our evaluation we used version 1.4.3 of POP and two POP benchmark configurations. The first, referred to as 'x1,' represents a relatively coarse resolution similar to that currently used in coupled climate models. The horizontal resolution is roughly one degree (320 × 384) and uses a displaced-pole grid with the pole of the grid shifted into Greenland and enhanced resolution in the equatorial regions. The vertical coordinate uses 40 vertical levels with smaller grid spacing near the surface to better resolve the surface mixed layer. Because this configuration does not resolve eddies, it requires the use of computationally intensive subgrid parameterizations. This configuration is set up to be identical to the production configuration of the Community Climate System Model with the exception that the coupling to full atmosphere, ice and land models has been replaced by analytic surface forcing.

Figure 15 shows a platform comparison of POP throughput for the x1 benchmark problem. On the Cray X1E, we considered an MPI-only implementation and also an implementation that uses a Co-Array Fortran (CAF) implementation of a performance-sensitive halo update operation. All other results were for MPI-only versions of POP. The BG/L experiments were run in "virtual node" mode. The XT3 performance is similar to that of both the SGI Altix and the IBM p575 cluster up to 256 processors, and continues to scale out to 1024 processors even for this small fixed size problem.

Figure 16 shows the performance of the barotropic portion of POP. While lower latencies on the Cray X1E (when using MPI collectives for global communication and CAF for point-to-point communication) and SGI Altix systems give these systems an advantage over the XT3 for this phase, the XT3 shows good scalability in the sense that the cost does not increase significantly out to 1024 processors. In particular, scaling on the XT3 is superior to that of the p575 cluster and continues to be competitive compared to BG/L. Figure 17 shows the performance of the baroclinic portion of POP. The Cray XT3 performance was very similar to that of both the SGI Altix and the p575 cluster, and shows excellent scalability.

The second benchmark, referred to as "0.1," utilizes a 1/10-degree horizontal resolution (3600 × 2400) and 40



(a)

(b)

Fig. 14 Scaling performance of dynamics and physics for CAM D-grid benchmark, using logarithmic Y-axis (a) and non-logarithmic Y-axis (b).

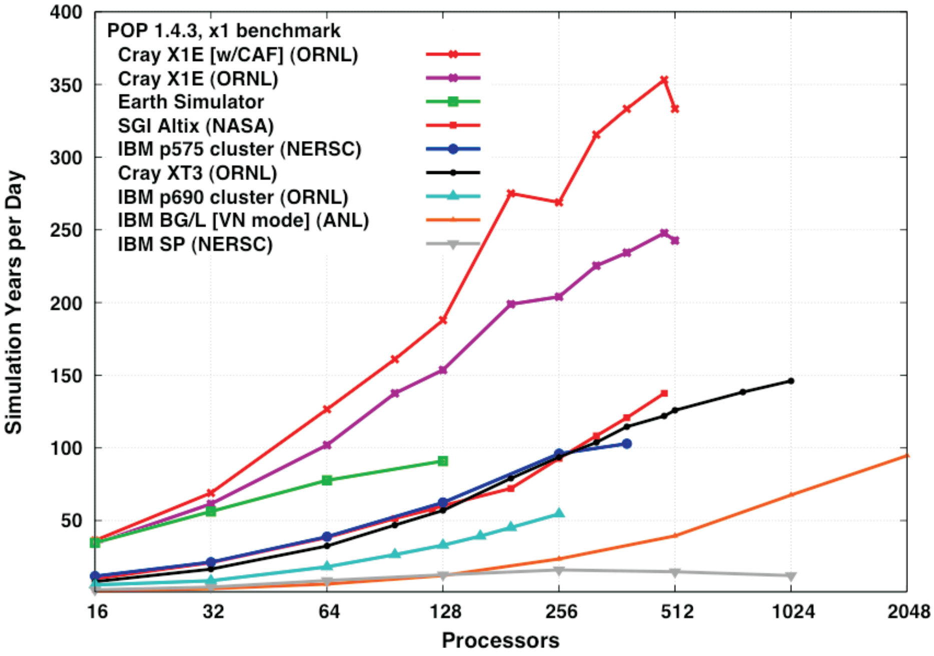


Fig. 15 Performance of POP for x1 benchmark.

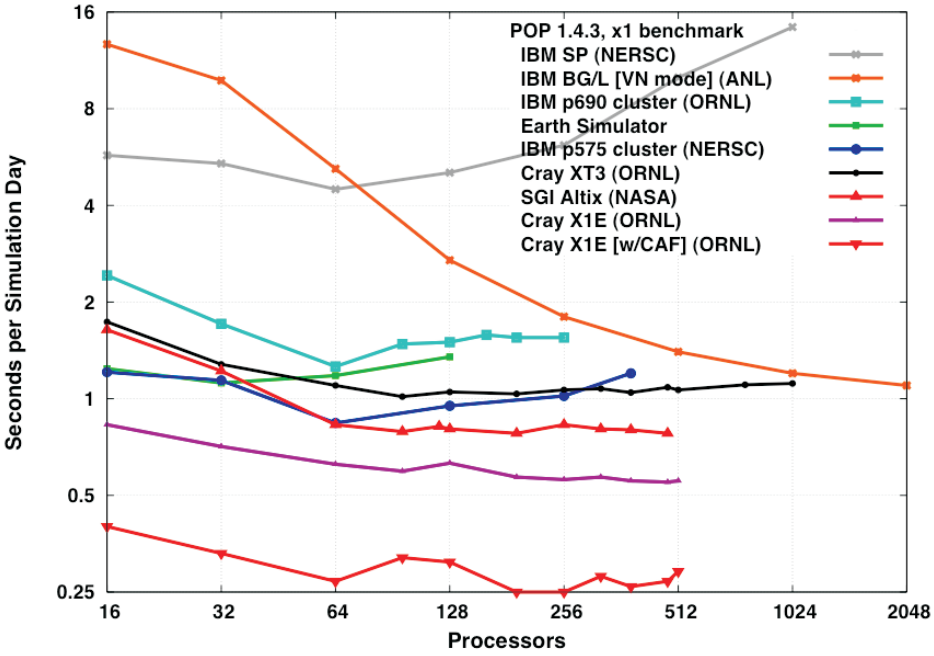


Fig. 16 Performance of POP barotropic phase for x1 benchmark.

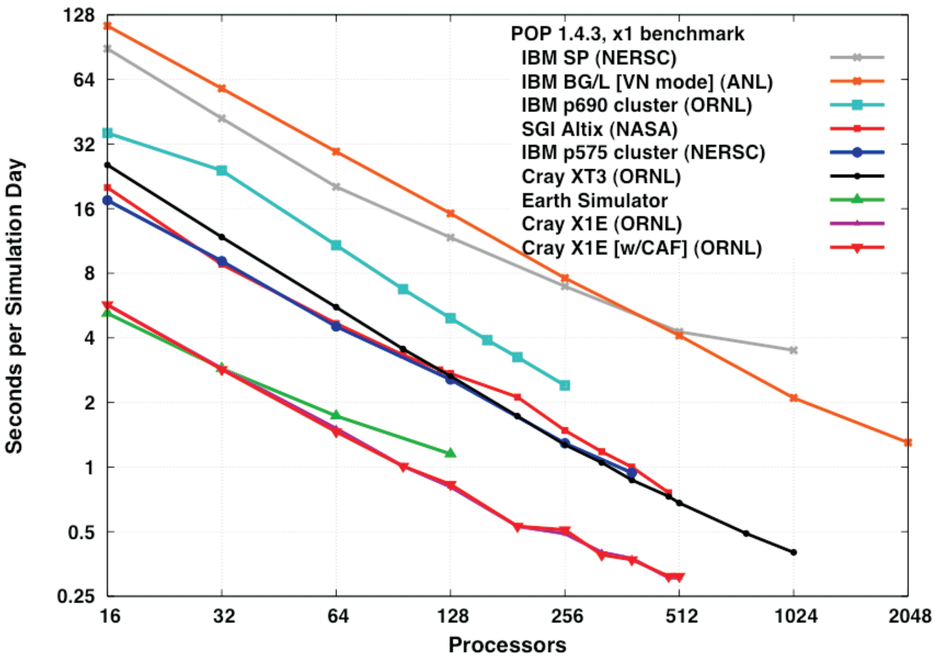


Fig. 17 Performance of POP baroclinic phase for x1 benchmark.

vertical levels. The 0.1 degree grid is also a displaced posed grid with 1/10 degree (10 km) resolution around the equator down to 2.5 km near the poles. The benchmark uses a simple biharmonic horizontal mixing rather than the more expensive subgrid parameterizations used in the x1 benchmark. As mentioned earlier, this resolution resolves eddies for effective heat transport and is used for ocean-only or ocean and sea ice experiments. The cost is prohibitive for use in full coupled climate simulations on contemporary systems.

Figure 18 shows a platform comparison of POP throughput for the 0.1 benchmark. Both performance and scalability on the XT3 are excellent out to almost 5000 processors, achieving 66% efficiency when scaling from 1000 to 5000 processors. Figure 19 shows the performance of both the barotropic and baroclinic phases. From this it is clear that 5000 processors is the practical processor limit on the XT3 as the cost of the barotropic phase dominates that of the baroclinic phase for more than 4000 processors, and is not decreasing. Note that both the X1E and the XT3 demonstrate superlinear speedup in the baroclinic phase, indicating that the problem is still too large to fit into the processor cache even at the maximum processor count. A newer version of POP supports a subblocking technique that should improve cache locality for this benchmark.

6.3 GYRO

GYRO (Candy and Waltz 2003) is a code for the numerical simulation of tokamak microturbulence, solving time-dependent, nonlinear gyrokinetic-Maxwell equations with gyrokinetic ions and electrons capable of treating finite electromagnetic microturbulence. GYRO uses a five-dimensional grid and propagates the system forward in time using a fourth-order, explicit Eulerian algorithm. GYRO has been ported to a variety of modern HPC platforms including a number of commodity clusters. Since code portability and flexibility are considered crucial to this code's development team, only a single source tree is maintained and platform-specific optimizations are restricted to a small number of low-level operations such as FFTs.

For our evaluation, we ran GYRO version 3.0.0 for two benchmark problems, B1-std and B3-gtc. Newer versions of GYRO are now available that achieve better performance on all platforms. However, we have not had the opportunity to benchmark our test systems using the newer versions of the code. Thus the performance data presented here is a consistent measure of platform capabilities, but not a valid evaluation of current GYRO performance.

B1-std is the Waltz standard case benchmark (Waltz, Kerbel, and Milovich 1994). This is a simulation of elec-

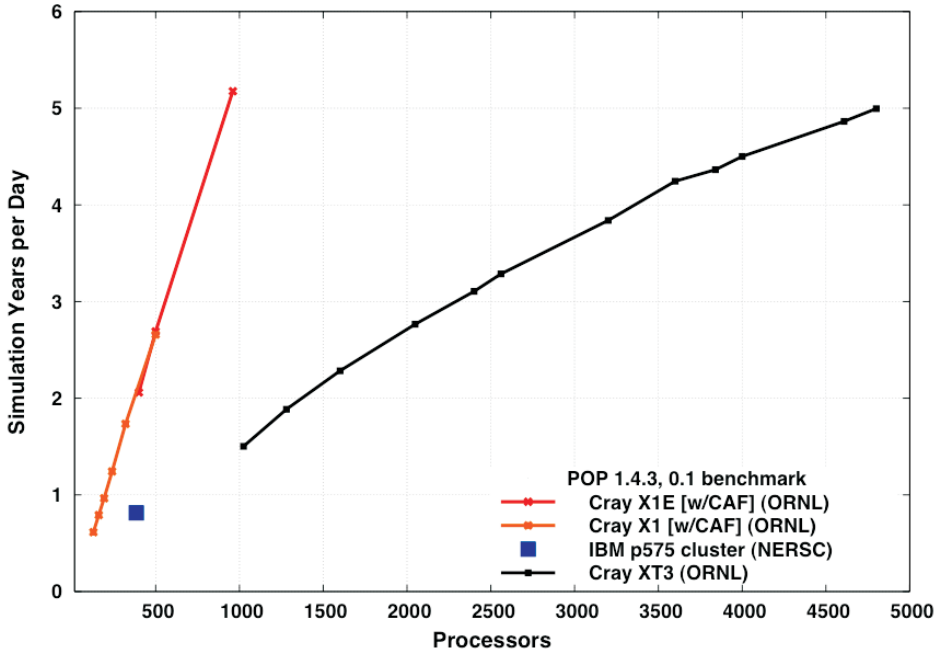


Fig. 18 Performance of POP for 0.1 benchmark.

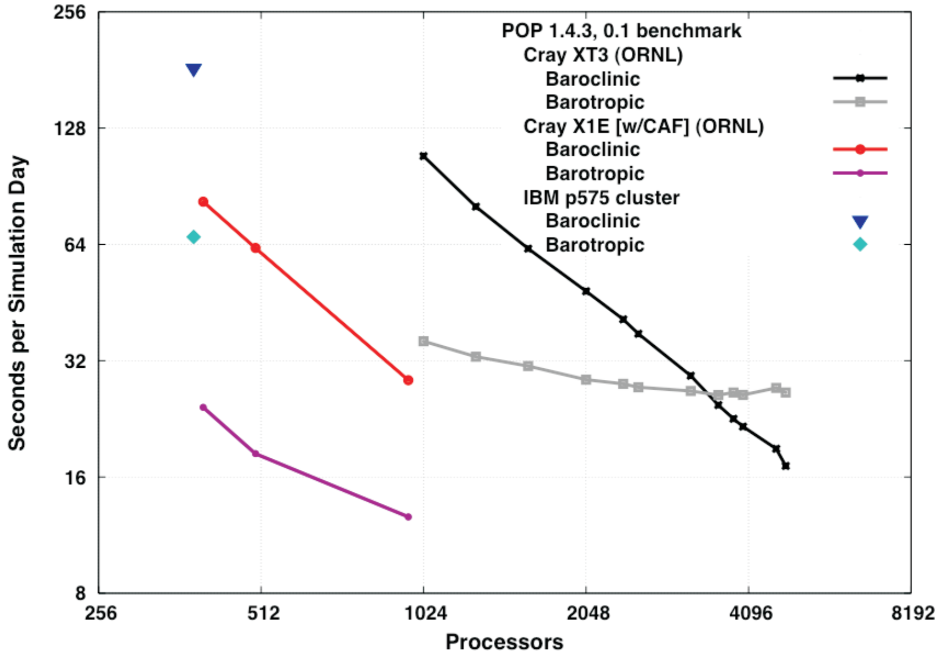


Fig. 19 Performance of POP phases for 0.1 benchmark.

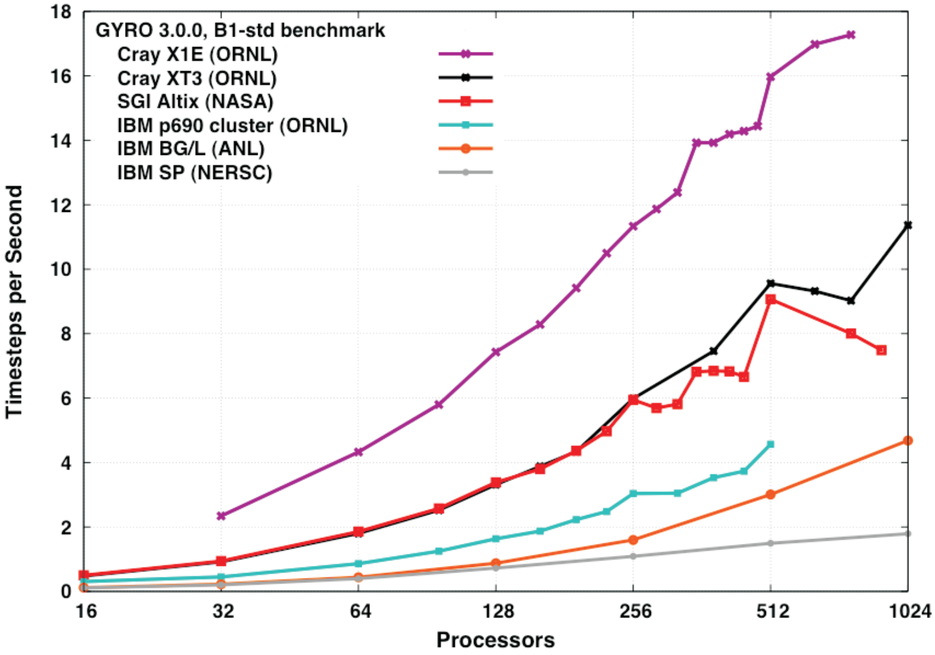


Fig. 20 GYRO performance for B1-std benchmark.

trostatic turbulence using parameters characteristic of the DIII-D tokamak at mid-radius. Both electrons and ions are kinetic, and electron collisions (pitch-angle scattering) are included. The grid is 16 × 140 × 8 × 8 × 20. Since 16 toroidal modes are used, a multiple of 16 processors must be used to run the simulation. Interprocess communication overhead for this problem is dominated by the time spent in “transposes” used to change the domain decomposition within each timestep. The transposes are implemented using simultaneous MPI_Alltoall collective calls over subgroups of processes.

Figure 20 shows platform comparisons of GYRO throughput for the B1-std benchmark problem. Note that there is a strong algorithmic preference for power-of-two numbers of processors for large processor counts, arising from significant redundant work when not using a power-of-two number of processes. This impacts performance differently on the different systems. The XT3 performance is superior to all of the other platforms except the X1E. Scaling on the XT3 is also excellent out to 512 processors.

Figure 21 plots the ratio of the time spent in the communication transposes to full runtime. The transposes for this problem size are sensitive to both latency and bandwidth. By this metric, the communication performance of the XT3 is among the best compared to the other systems up to 512 processors. The somewhat poor latency on the XT3 degrades this performance metric at higher processor counts compared to the X1E and BG/L.

B3-gtc is a high-toroidal-resolution electrostatic simulation with simplified electron dynamics (only ions are kinetic). The grid is 64 × 400 × 8 × 8 × 20. This case uses 64 toroidal modes and must be run on multiples of 64 processors. The 400-point radial domain with 64 toroidal modes gives extremely high spatial resolution, but electron physics is ignored, allowing a simple field solve and large timesteps. As with the B1-std benchmark, interprocess communication overhead for this problem is dominated by the time spent in the transposes.

Figure 22 shows platform comparisons of GYRO throughput for the B3-gtc benchmark problem. As with B1-std, there is an algorithmic preference for power-of-two numbers of processors for large processor counts, The Altix is somewhat superior to the XT3 out to 960 processors, but XT3 scalability is excellent, achieving the best overall performance at 4096 processors. Figure 23 plots the time spent in the communication transposes for this benchmark. Figure 24 plots the ratio of the time spent in the communication transposes to full runtime. The transposes for this problem size are primarily a measure of communication bandwidth. By these metrics, the communication performance of the XT3 is excellent

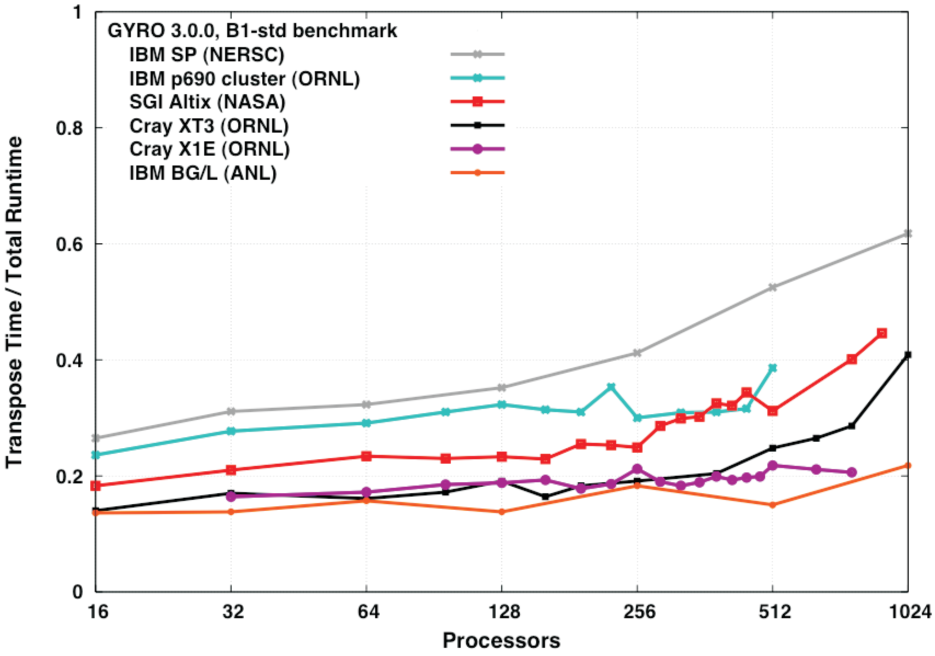


Fig. 21 Ratio of time for GYRO transpose communication to total run time for B1-std benchmark.

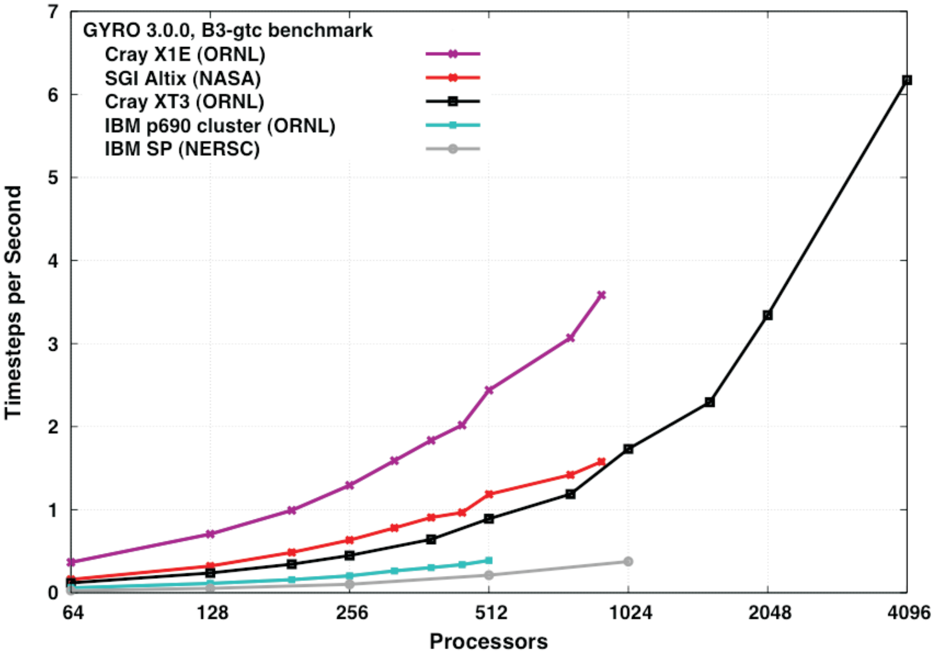


Fig. 22 GYRO performance for B3-gtc benchmark.

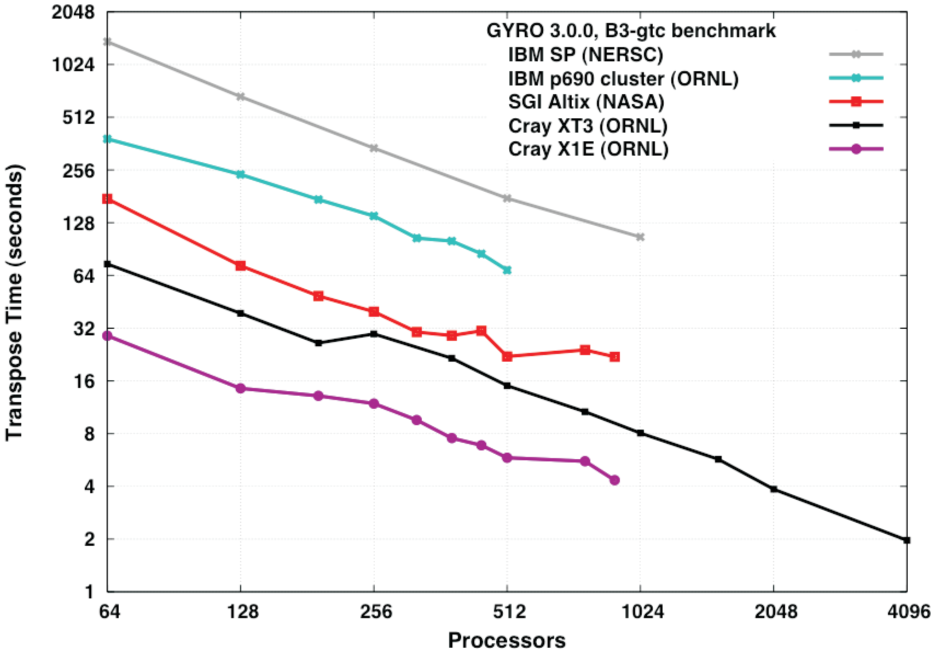


Fig. 23 GYRO transpose communication performance for B3-gtc benchmark.

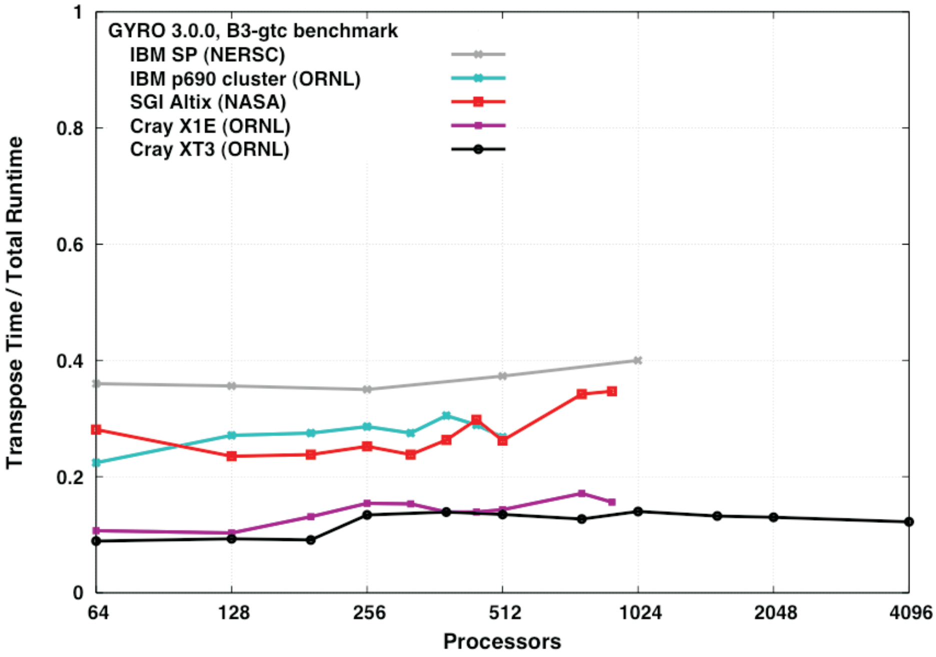


Fig. 24 Ratio of GYRO transpose communication time to total run time for B3-gtc benchmark.

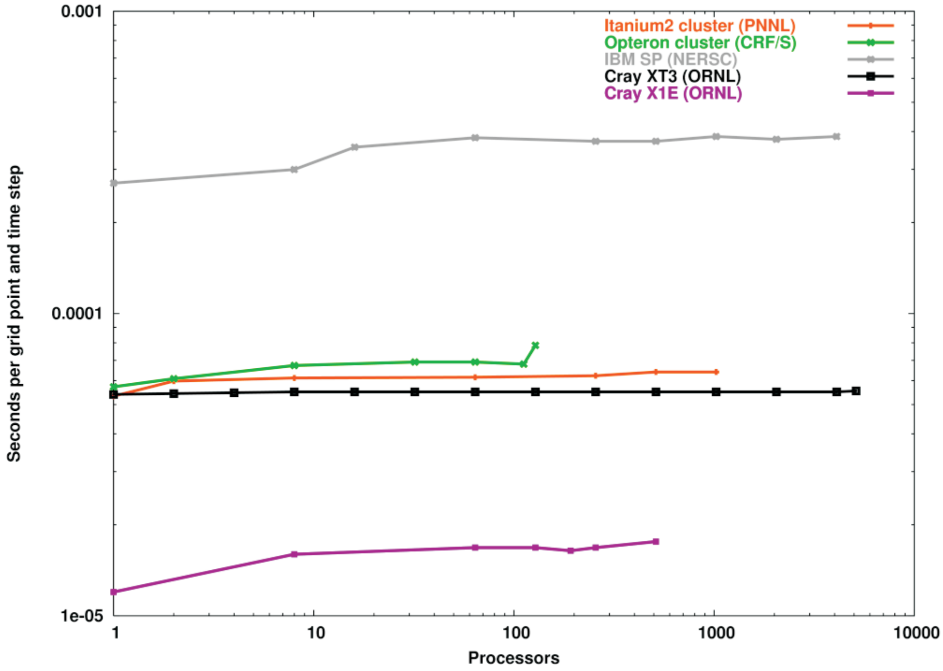


Fig. 25 S3D performance.

compared to the other systems, beating even that of the X1E when the relative speed of the rest of the computation is taken into account.

6.4 S3D

S3D is a code used extensively to investigate first-of-a-kind fundamental turbulence-chemistry interactions in combustion topics ranging from premixed flames (Chen and Im 2000; Hawkes and Chen 2004), to auto-ignition (Echekki and Chen 2003), to non-premixed flames (Mahalingam, Chen, and Vervisch 1995; Hawkes et al. 2005; Sutherland, Smith, and Chen 2005). It is based on a high-order accurate, non-dissipative numerical scheme. Time advancement is achieved through a fourth-order explicit Runge–Kutta method, differencing is achieved through high-order (eighth-order with tenth-order filters) finite differences on a Cartesian, structured grid, and Navier–Stokes Characteristic Boundary Conditions (NSCBC) are used to prescribe the boundary conditions. The equations are solved on a conventional structured mesh.

This computational approach is appropriate for direct numerical simulation of turbulent combustion. The coupling of high-order finite difference methods with explicit Runge–Kutta time integration make effective use of the available resources, obtaining spectral-like spatial resolution without excessive communication overhead and allowing scalable parallelism.

The benchmarked problem is a 3-D direct numerical simulation of a slot-burner Bunsen flame with detailed chemistry. This includes methane-air chemistry with 17 species and 73 elementary reactions. The benchmark is scaled up with increasing processor count, as is the typical use of this code. The simulation used 50^3 grid points per processor, or 640 million grid points in the largest test on the XT3.

Figure 25 shows the simulation throughput in seconds per grid point and time step for this benchmark on various systems. This shows that S3D scales well across several platforms and exhibited a 90% scaling efficiency (almost perfectly flat) on the XT3. This superior scaling can be attributed to the excellent nearest neighbor communication efficiency as documented earlier with the MPI Exchange benchmark in Section 4.3.

6.5 Molecular Dynamics Simulations

Molecular dynamics (MD) simulations enable the study of complex, dynamic processes that occur in biological systems (Karplus and Petsko 1990). MD methods are now routinely used to investigate the structure, dynamics, functions, and thermodynamics of biological mole-

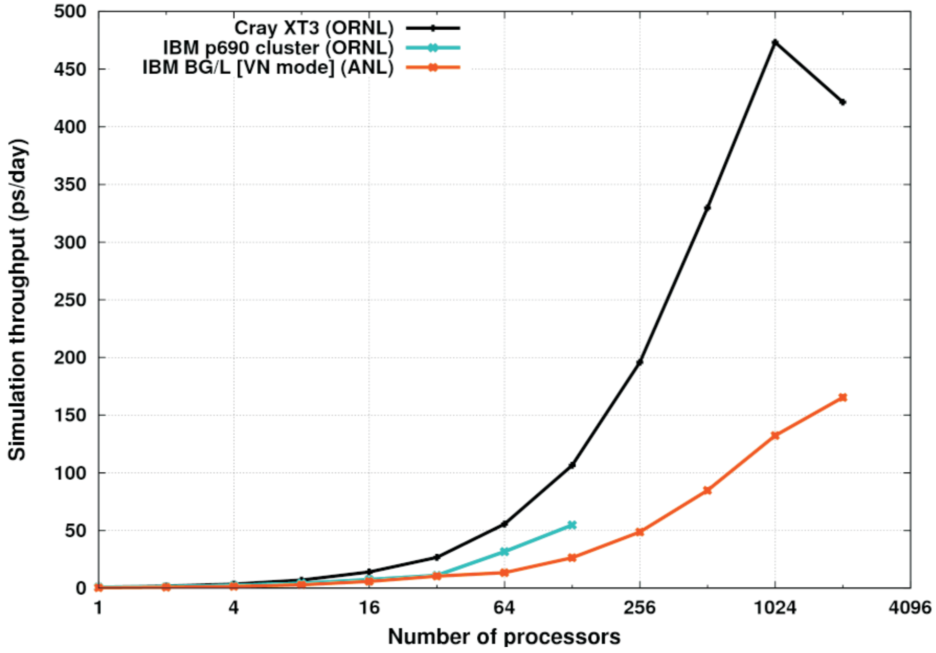


Fig. 26 AMBER simulation throughput.

cules and their complexes. The types of biological activity that have been investigated using MD simulations include protein folding, enzyme catalyzation, conformational changes associated with bimolecular function, and molecular recognition of proteins, DNA, and biological membrane complexes. Biological molecules exhibit a wide range of time and length scales over which specific processes occur, hence the computational complexity of an MD simulation depends greatly on the time and length scales considered. With an explicit solvation model, typical system sizes of interest range from 20,000 atoms to more than 1 million atoms; if the solvation is implicit, sizes range from a few thousand to about 100,000 atoms. The simulation time period can range from picoseconds to a few microseconds or longer on contemporary platforms.

Several commercial and open source MD software frameworks are in use by a large community of biologists, including AMBER (Pearlman et al. 1995) and LAMMPS (Plimpton 1995). These packages differ in the form of their potential function and also in their force-field calculations. Some of them are able to use force fields from other packages as well. AMBER provides a wide range of MD algorithms. The version of LAMMPS used in our evaluation does not use the energy minimization technique.

AMBER consists of about 50 programs that perform a diverse set of calculations for system preparation, energy minimization (EM), molecular dynamics (MD), and analysis of results. AMBER's main module for EM and MD is known as *sander* (for *s*imulated *a*nnealing with *N*MR-*d*erived *e*nergy *r*estraints). We used sander to investigate the performance characteristics of EM and MD techniques using the Particle Mesh Ewald (PME) and Generalized Born (GB) methods. We performed a detailed analysis of PME and GB algorithms on massively parallel systems (including the XT3) in other work (Alam et al. 2006).

The bio-molecular systems used for our experiments were designed to represent the variety of complexes routinely investigated by computational biologists. In particular, we considered the RuBisCO enzyme based on the crystal structure 1RCX, using the Generalized Born method for implicit solvent. The model consists of 73,920 atoms. In Figure 26, we represent the performance of the code in simulation throughput, expressed as simulation picoseconds per real day (ps/day). The performance on the Cray XT3 is very good for large-scale experiments, showing a throughput of over twice the other platforms we investigated (Alam et al. 2006). The bio-molecular MD algorithms implemented in AMBER have high computational intensities, which is reflected in

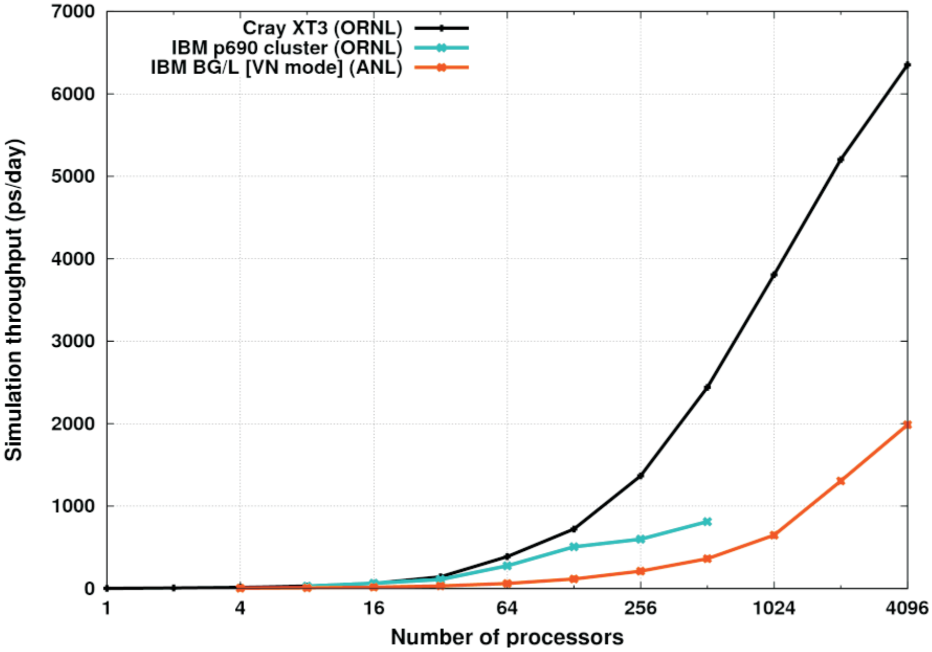


Fig. 27 LAMMPS simulation throughput with approximately 290,000 atoms.

the high throughput results on the XT3 system. At the same time however, the communication volumes in these simulations do not scale with the number of processors. Therefore, the scaling is limited to a few thousand processors.

LAMMPS (Large-scale Atomic/Molecular Massively Parallel Simulator; Plimpton 1995) is a classical MD code. LAMMPS models an ensemble of particles in a liquid, solid or gaseous state and can be used to model atomic, polymeric, biological, metallic or granular systems. The version we used for our experiments is written in C++ and MPI. For our evaluation, we considered the RAQ system that is a model on the enzyme RuBisCO. This model consists of 290,220 atoms with explicit treatment of solvent. We observed very good performance for this problem on the Cray XT3 (see Figure 27), with over 70% parallel efficiency (where parallel efficiency is the speedup divided by the number of processors) on up to 2048 processors and over 40% parallel efficiency on 4096 processor run.

6.6 AORSA

The two- and three-dimensional all-orders spectral algorithms (AORSA; Jaeger et al. 2001) code is a full-wave model for radio frequency heating of plasmas in fusion energy devices such as ITER, the international tokamak project. AORSA solves the more general integral form of the wave equation with no restriction on wavelength relative to orbit size and no limit on the number of cyclotron harmonics. With this approach, the limit on attainable resolution comes not from the model, but only from the size and speed of the available computer.

AORSA operates on a spatial mesh, with the resulting set of linear equations solved for the Fourier coefficients. The problem size is characterized by the total number of Fourier modes retained by the model. The physical process is described using a continuous integral equation involving polynomials. The discrete solution must capture the periodic wave behavior, which is better done using sines and cosines. A Fast Fourier Transform algorithm converts the problem to a frequency space, resulting in a dense, complex-valued linear system. This system is solved using the ScaLAPACK library; in particular routines `pzgetrf` factors the matrix into upper and lower matrices, which `pzgetrs` then uses to compute the solution vector.

Each grid point creates three linear equations, less the point outside of the physical region, so for an M × N grid the linear system is of dimension approximately 0.7*$3*M*N$. For example, the 256 × 256 grid creates a linear system of dimension 124,587 and the 370 × 370

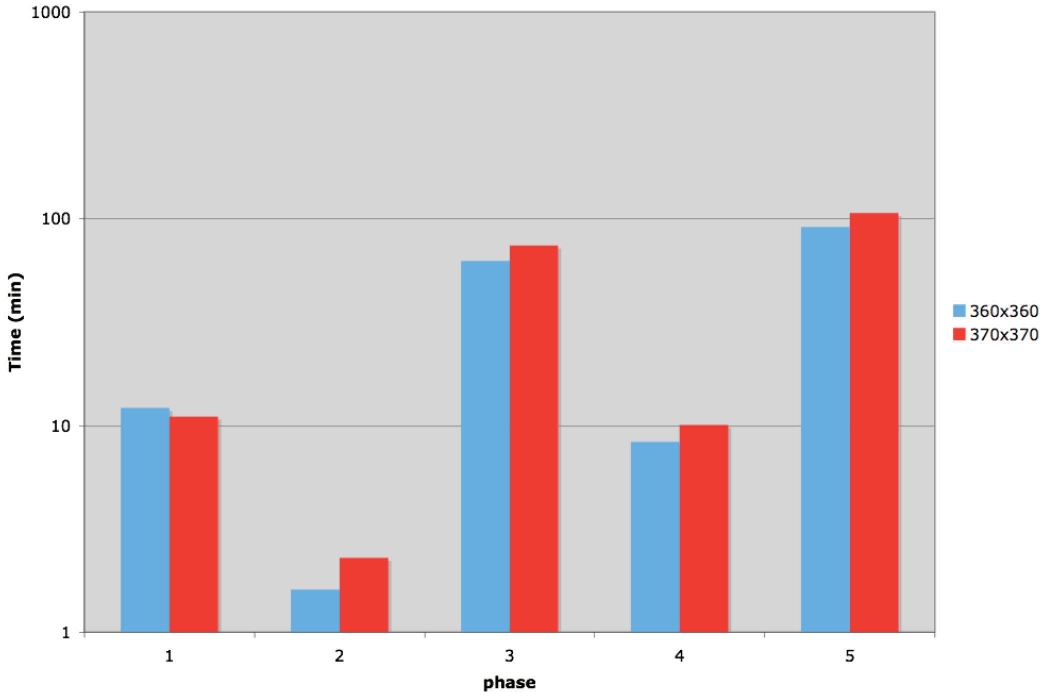


Fig. 28 AORSA elapsed time by phase.

grid creates a linear system of dimension 260,226. Immediate plans call for executing across a 500 × 500 grid, which will result in a dense linear system of dimension approaching 500,000.

In 2005 AORSA was ported to the ORNL XT3, immediately allowing researchers to run experiments at grid resolutions previously unattainable. Up to this point, the finest resolution was 200×200, requiring one hour on 2000 processors on the IBM Power3 Seaborg computer at NERSC. The first large problem run on the ORNL XT3 increased the resolution to 256 × 256, with a runtime of 44.42 minutes on 1024 processors, 27.1 minutes on 2048 processors, and 23.28 on 3072 processors, providing the most detailed simulations ever done of plasma control waves in a tokamak. Since then experiments using even finer resolutions have been run. For example, preliminary calculations using 4096 processors have allowed the first simulations of mode conversion in ITER. Mode conversion from the fast wave to the ion cylcotron wave (ICW) has been identified in ITER using mixtures of deuterium, tritium and helium-3 at 53 MHz. Figure 28 shows the performance of various phases of AORSA execution of a simplified version of this problem executed on 4096 processors. The blue bars are timings for the 360 × 360 grid, the red for the 370 × 370 grid. The phases shown in the figure are 1) calculate the Maxwellian matrix; 2) generate and distribute the dense linear system; 3) solve the linear system; 4) calculate the quasilinear operator; and 5) total time.

The ScaLAPACK solver achieves 10.56 TFLOPS, which is about 53% of peak performance. The difference is attributable to a load imbalance, due in part to the elimination of grid points outside the physical region, as well as the high MPI latencies of the XT3. The former is being addressed by the code development team; the latter is being addressed by Cray.

6.7 PFLOTRAN

PFLOTRAN (Parallel FLOw and TRANsport; Lichtner and Wolfsberg 2004; Hammond, Valocchi, and Lichtner 2005; Lu and Lichtner 2005; Lu et al. 2005; Mills, Lichtner, and Lu 2005) is a state-of-the-art prototype code for modeling multiphase, multicomponent reactive subsurface environmental flows. It is currently being used to understand problems at the Nevada Test Site and the Hanford 300 Area, as well as for geologic CO_2 sequestration studies. The code employs domain decomposition-based parallelism and is built using the PETSc framework (Balay et al. 2004) from Argonne National Laboratory. PFLOTRAN consists of two distinct modules: a flow module (PFLOW) that solves an energy balance

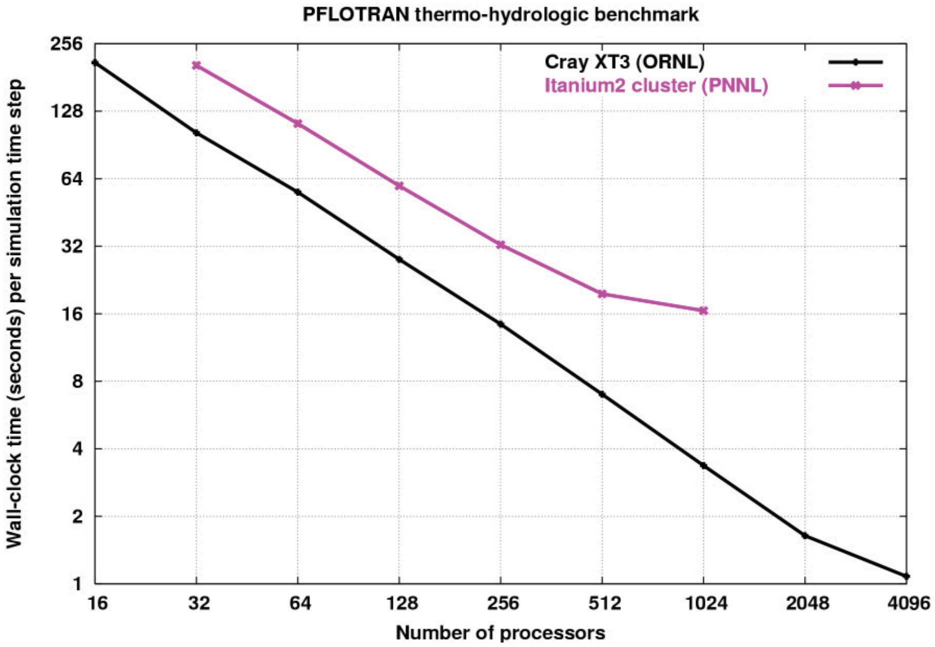


Fig. 29 PFLOTRAN performance.

equation and mass conservation equations for water and other fluids, and a reactive transport module (PTRAN) that solves mass conservation equations for a multicomponent geochemical system. In coupled mode, flow velocities, saturation, pressure and temperature fields computed from PFLOW are fed into PTRAN. For transient problems, sequential coupling of PFLOW and PTRAN enables changes in porosity and permeability due to chemical reactions to alter the flow field.

Governing equations are discretized using an integral finite-volume formulation on an orthogonal structured grid (extension to unstructured grids is planned). Time-stepping is fully implicit (backward Euler). The nonlinear equations arising at each time step are solved using the Newton–Krylov solver framework of PETSc, allowing easy selection of the most appropriate solvers and preconditioners for the problem at hand.

PFLOTRAN has shown excellent parallel scalability. Figure 29 illustrates the performance of the PFLOW module on a modest sized thermo-hydrologic benchmark problem on a 256 × 64 × 256 grid with three degrees of freedom per node (approximately 12.6 million degrees of freedom total). In this case, the linear systems within the Newton method are solved using GMRES(30) with a block-Jacobi pre-conditioner with ILU(0) on each block. The benchmark was run on both the MPP2 Itanium2 cluster (1960 1.5 GHz Itanium2 processors with Quadrics QsNetII interconnect) at PNNL and the Cray XT3 at ORNL. Scaling is exceptionally good on the XT3, with linear speedup on up to 2048 processors, and modest speedup when going to 4096 processors, at which point the modest problem size becomes apparent and the numerous MPI Reductions inside the linear system solver present a scalability barrier. Since reactive flow problems for production runs will often involve 10–20 chemical degrees of freedom per node, we expect to see even better parallel efficiency for problems involving reactive chemistry.

7 Conclusions and Plans

In 2005, Oak Ridge National Laboratory installed a 5294 processor Cray XT3. In this paper we describe our performance evaluation of the system, including microbenchmark, kernel, and application benchmark results. We focused on applications from important Department of Energy applications areas including climate and fusion. In experiments with up to 4096 processors, we observed that the Cray XT3 shows tremendous potential for supporting the Department of Energy application workload. The Cray X1/X1E outperformed the XT3 on many of our test applications, but was limited to 1024 processors (the size of the ORNL X1E system). Also, we expect the X1E to have a higher price-to-performance ratio with its custom vector processors than the commodity Opteron parts. The XT3 interconnect latency is fair for small point-to-point messages, but the interconnect's real strength is its high bandwidth, though that strength is diminished by the current SeaStar network injection problem. The single-core Opteron processors in the ORNL XT3 show good computational capability and exceptional memory access latency. Furthermore, the XT3 showed very good scalability for our test applications.

ORNL plans to upgrade its 25 TF Cray XT3 to a 100 TF system with dual-core processors in 2006. The first stage of this upgrade was completed in August 2006, bringing the system to over 50 TF. We plan to evaluate the upgraded system and compare its performance with that of the existing system as captured in this document. We expect to observe a drop in memory access performance with the dual core processors. Specifically, we expect to observe higher memory access latency due to the addition of cache coherence logic in the memory access path, and lower memory bandwidth per core due to contention between a processor's two cores for the link to memory. How this reduced per-core memory performance will affect our application results is unknown, but our comparison between the evaluation described in this document and our evaluation of the upgraded system will provide valuable insight into the use of multi-core processors in future MPP systems at ORNL.

Acknowledgments

This research was sponsored by the Office of Mathematical, Information, and Computational Sciences, Office of Science, U.S. Department of Energy under Contract No. DE-AC05-00OR22725 with UT-Battelle, LLC. Accordingly, the U.S. Government retains a non-exclusive, royalty-free license to publish or reproduce the published form of this contribution, or allow others to do so, for U.S. Government purposes. Also, we would like to thank Jeff Beckleheimer, John Levesque, Nathan Wichmann, and Jim Schwarzmeier of Cray, and Don Maxwell of ORNL for their assistance in this endeavor. We thank Hongzhang Shan of LBL for collecting GYRO performance data on the IBM SP; James B. White III for collecting POP performance data on the p575 cluster; Yoshikatsu Yoshida for collecting POP performance data on the Earth Simulator; David Parks for collecting CAM performance data on the Earth Simulator; Michael Wehner for collecting CAM performance data on the IBM SP; and Arthur Mirin for collecting CAM performance data on the Itanium2 cluster at LLNL. We thank the National Energy Research Scientific Computing Center for access to the IBM SP, Argonne National Laboratory for access to the IBM BG/L, the NASA Advanced Supercomputing Division for access to their SGI Altix, and the ORNL Center for Computational Sciences (CCS) for access to the Cray X1, Cray X1E, Cray XD1, Cray XT3, IBM p690 cluster, and SGI Altix. The CCS is supported by the Office of Science of the U.S. Department of Energy under Contract No. DE-AC05-00OR22725.

Author Biographies

Sadaf R. Alam is on the research staff in the Future Technologies Group of the Computer Science and Mathematics Division at Oak Ridge National Laboratory. She is also a member of the Scientific Computing Group in the National Center for Computational Sciences. Her research interests are high performance scientific computing and performance evaluation, modeling and projection of parallel systems and emerging processing devices. She earned her Ph.D. from the University of Edinburgh and joined ORNL in 2004.

Richard Barrett is a senior research computational scientist in the Future Technologies Group in the Computer Science and Mathematics Division at Oak Ridge National Laboratory, where he is also a member of the Scientific Computing Group in the National Center for Computational Sciences. His research interests span several areas required for creating effective scientific applications on current and future highest performance computing platforms. Of special interest are the use of programming models and languages; code development tools; performance modeling, analysis, and optimization; computer architectures; inter-process communication mechanisms; the solution of large scale linear systems; and the bridge between research and production computing.

Mark Fahey is a research staff member in the Scientific Computing Group within the National Center for Computational Sciences at Oak Ridge National Laboratory (ORNL). From 2001–2003, he was a research scientist as part of the Joint Institute for Computational Science at ORNL. Mark earned his Ph.D. from the University of Kentucky in 1999. His current interests are in the software engineering areas of portable performance, code design and maintenance of high performance computing codes.

Jeffery Kuehn is a senior high performance computing evaluation researcher in the Future Technologies Group of the Computer Science and Mathematics Division at Oak Ridge National Laboratory, and also a member of the Scientific Computing Group of the National Center for Computational Sciences. His research interests include performance engineering of whole device coupled fusion simulations, detonation theory, computer system evaluation, and micro-benchmarking. He earned his M.S. in mechanical engineering from University of Colorado, Boulder in 1988. Previously employed at National Center for Atmospheric Research and Cray Research, Inc., he joined Oak Ridge National Laboratory in 2005.

Bronson Messer is a computational scientist in the Scientific Computing Group of the National Center for Computational Sciences at Oak Ridge National Laboratory. Trained as an astrophysicist, his primary research interests are in the mechanism and phenomenology of supernovae, both the core-collapse and thermonuclear types. He is also interested in large linear system solution, parallel I/O, and other topics important to large-scale scientific computation. He received his Ph.D. in physics in 2000, and after postdoctoral appointments at ORNL and the University of Chicago, he returned to ORNL in 2005.

Richard Tran Mills is a computational scientist at Oak Ridge National Laboratory (ORNL), where he is a member of the Scientific Computing and Computational Earth Sciences Groups. His research interests include parallel and high performance computing, iterative methods for the solution of linear and nonlinear algebraic equations, and subsurface flow and reactive transport. Richard earned his Ph.D. in computer science in 2004 at the College of William and Mary, where he was a Department of Energy Computational Science Graduate Fellow.

Philip Roth is a computer scientist at Oak Ridge National Laboratory (ORNL), where he is a founding member of the Future Technologies Group. His research interests include performance analysis, prediction, and tools with special emphases on scalability and automation; systems software; virtualization technologies; programming models; and storage for large-scale systems. He earned his Ph.D. in computer science from the University of Wisconsin—Madison in 2005.

Jeffrey Vetter is a computer scientist in the Computer Science and Mathematics Division (CSM) of Oak Ridge National Laboratory (ORNL), where he leads the Future Technologies Group, and a joint Professor in the College of Computing at Georgia Tech. His research interests lie largely in the areas of experimental software systems and architectures for high-end computing. Jeff earned his Ph.D. in computer science from the Georgia Institute of Technology; he joined CSM in 2003.

Patrick Worley is a senior research computer scientist in the Computer Science and Mathematics Division (CSM) of Oak Ridge National Laboratory (ORNL). He earned his Ph.D. in computer science from Stanford University, joining CSM in 1987. His research interests include parallel algorithm design and implementation, especially as applied to geophysical and fusion science simulation models, and performance evaluation and optimization of parallel applications and computer systems. Worley leads the Performance Evaluation and Analysis Consortium End Station at the National Center for Computational Sciences Leadership Computing Facility and is a co-chair of the Software Engineering Working Group for the Community Climate System Model.

References

Agarwal, P. A., et al. (2004). *Cray X1 evaluation status report*, ORNL/TM-2004/13.

Alam, S. R., Agarwal, P., Geist, A., and Vetter, J. S. (2006). Performance characterization of molecular dynamics techniques for biomolecular simulations, In *Proceedings of the ACM SIGPLAN Symposium on Principles and Practice of Parallel Programming (PPOPP)*, New York City: ACM Press, pp. 59–68.

AMD (2004). *Software optimization guide for AMD Athlon™ 64 and AMD Opteron™ processors*, document number 25112.

Balay, S., et al. (2004). *PETSc users manual*, ANL-95/11 – Revision 2.1.5.

Blackmon, M. B., et al. (2001). The community climate system model, *Bulletin of the American Meteorological Society*, **82**(11): 2357–2376.

Brightwell, R., et al. (2000). Massively parallel computing using commodity components, *Parallel Computing*, **26**(2–3): 243–266.

Brightwell, R., Riesen, R., Lawry, B., and Maccabe, A. B. (2002). Portals 3.0: Protocol building blocks for low overhead communication, In *Proceedings of the Workshop on Communication Architecture for Clusters*, Ft. Lauderdale, pp. 164–173.

Brightwell, R., Camp, W., Cole, B., DeBenedictis, E., Leland, R., and Tomkins, J. (2005). Architectural specification for massively parallel computers – an experience and measurement-based approach, *Concurrency and Computation: Practice and Experience*, **17**(10): 1271–1316.

Candy, J. and Waltz, R. (2003). An Eulerian gyrokinetic-Maxwell solver, *Journal of Computational Physics*, **186**(2): 545–581.

Chen, J. H. and Im, H. G. (2000). Stretch effects on the burning velocity of turbulent premixed hydrogen-air flames, In *Proceedings of the Combustion Institute*, **1**: 211–218.

Collins, W. D., et al. (2006). The community climate system model Version 3 (CCSM3), *Journal of Climate*, **19**(11): 2122–2143.

Collins, W. D., et al. (2006). The formulation and atmospheric simulation of the community atmosphere model: CAM3, *Journal of Climate*, **19**(11): 2144–2161.

Dagum, L. and Menon, R. (1998). OpenMP: An industry-standard API for shared-memory programming, *IEEE Computational Science & Engineering*, **5**(1): 46–55.

Dunigan, T. H., Jr., Vetter, J. S., White, J. B., and Worley, P. H. (2005). Performance evaluation of the Cray X1 distributed shared memory architecture, *IEEE Micro*, **25**(1): 30–40.

Dunigan, T. H., Jr., Vetter, J. S., and Worley, P. H. (2005). Performance evaluation of the SGI Altix 3700, In *Proceed-*

ings of the International Conference on Parallel Processing (ICPP), Oslo, Norway, pp. 231–240.

Dunigan, T. H., Jr., Vetter, J. S., and Worley P. H. (2005). Performance evaluation of the Cray X1 distributed shared memory architecture, *IEEE Micro*, **25**(1): 30–40.

Echekki, T. and Chen, J. H. (2003). Direct numerical simulation of autoignition in non-homogeneous hydrogen-air mixtures, *Combustion and Flame*, **134**: 169–191.

Fahey, M. R., Alam, S. R., Dunigan, T. H., Vetter, J. S., and Worley, P. H. (2005). Early evaluation of the Cray XD1, In *47th Cray User Group Conference*. Knoxville, TN.

Gropp, W. D., Lusk, E., Doss, N. E., and Skjellum, A. (1996). A high-performance, portable implementation of the MPI message passing interface standard, *Parallel Computing*, **22**(6): 789–828.

Hammond, G. E., Valocchi, A. J., and Lichtner, P. C. (2005). Application of Jacobian-Free Newton-Krylov with physics-based preconditioning to biogeochemical transport, *Advances in Water Resources*, **28**: 359–376.

Hawkes, E. R. and Chen, J. H. (2004). Direct numerical simulation of hydrogen-enriched lean premixed methane-air flames, *Combustion and Flame*, **138**(3): 242–258.

Hawkes, E. R., Sankaran, R., Sutherland, J. C, and Chen, J. H. (2005). Direct numerical simulation of turbulent combustion: Fundamental insights towards predictive models, *Journal of Physics: Conference Series (SciDAC 2005)*, **16**: 65–79.

High-End Computing Revitalization Task Force (2004). *Federal plan for high-end computing*, Washington, DC.

Jaeger, E. F., Berry, L. A., D'Azevedo, E., Batchelor, D. B., and Carter, M. D. (2001). All-orders spectral calculation of radio frequency heating in two-dimensional toroidal plasmas, *Physics of Plasmas*, **8**(5): 1573–1583.

Jones, P. W., Worley, P. H., Yoshida, Y., White, J. B., and Levesque, J. (2005). Practical performance portability in the parallel ocean program (POP), *Concurrency and Computation: Experience and Practice*, **17**(10): 1317–1327.

Karplus, M. and Petsko, G. A. (1990). Molecular dynamics simulations in biology, *Nature*, **347**: 631–639.

Kiehl, J. T., Hack, J. J., Bonan, G., Boville, B. A., Williamson, D. L., and Rasch, P. J. (1998). The National Center for Atmospheric Research community climate model: CCM3, *Journal of Climate*, **11**: 1131–1149.

Kuehn, J. A. and Wichmann, N. L. (2006). HPCC update and analysis, In *Proceedings of the Cray Users Group 2006 Annual Meeting*, Rhodos, Greece, Cray Users Group, Inc.

Lichtner, P. C. and Wolfsberg, A. (2004). Modeling thermal-hydrological-chemical (THC) coupled processes with applications to underground nuclear tests at the Nevada test site; A grand challenge supercomputing problem, In *Proceedings of the MPU Workshop: Conceptual Model Development for Subsurface Reactive Transport Modeling of Inorganic Contaminants, Radionuclides and Nutrients*.

Lin, S. J. (2004). A vertically Lagrangian finite-volume dynamical core for global models, *Monthly Weather Review*, **132**(10): 2293–2307.

Lu, C. and Lichtner, P. C. (2005). PFLOTRAN: Massively parallel 3-D simulator for CO2 sequestration in geologic media, In *DOE-NETL Fourth Annual Conference on Carbon Capture and Sequestration*.

Lu, C., Lichtner, P. C., Tsimpanogiannis, I. N., and Mills, R. T. (2005). Parametric study of CO2 sequestration in geologic media using the massively parallel computer code PFLOTRAN, In *Proceedings of the AGU Fall Meeting*, San Francisco.

Luszczek, P., et al. (2005). *Introduction to the HPC Challenge benchmark suite*, LBNL-57493.

Mahalingam, S., Chen, J. H., and Vervisch, L. (1995). Finite-rate chemistry and transient effects in direct numerical simulations of turbulent non-premixed flames, *Combustion and Flame*, **102**(3): 285–297.

Mattson, T. G., Scott, D., and Wheat, S. R. (1996). A teraFLOP supercomputer in 1996: The ASCI TFLOP system, In *Proceedings of the 10th International Parallel Processing Symposium (IPPS 96)*, Honolulu, Hawaii, pp. 84–93.

Mills, R. T., Lichtner, P. C., and Lu, C. (2005). *PFLOTRAN: A massively parallel simulator for reactive flows in geologic media*. SC05, (Poster), Seattle, WA.

Mucci, P. J., London, K., and Thurman, J. (1998). *The Cache-Bench report*, CEWES/ERDC MSRC/PET Technical Report 98–25.

Pearlman, D. A., et al. (1995). AMBER, a package of computer programs for applying molecular mechanics, normal mode analysis, molecular dynamics and free energy calculations to simulate the structural and energetic properties of molecules, *Computer Physics Communication*, **91**: 1–41.

Pedretti, K., Brightwell, R., and Williams, J. (2002). Cplant runtime system support for multi-processor and heterogeneous compute notes, In *Proceedings of the IEEE International Conference on Cluster Computing (CLUSTER 2002)*, Chicago, pp. 207–214.

Plimpton, S. J. (1995). Fast parallel algorithms for short-range molecular dynamics, *Journal of Computational Physics*, **117**(1): 1–19.

Scott, S. L. (1996). Synchronization and communication in the T3E multiprocessor, In *Proceedings of the Architectural Support for Programming Languages and Operating Systems (ASPLOS)*, pp. 26–36.

Snir, M., et al., (eds.) (1998). *MPI—The Complete Reference*. 2nd ed. 2 vols, *Scientific and Engineering Computation*. Cambridge, MA: MIT Press.

Sutherland, J. C., Smith, P. J., and Chen, J. H. (2005). Quantification of differential diffusion in nonpremixed systems, *Combustion Theory and Modelling*, **9**(2): 365–383.

U.S. Department of Energy Office of Science. (2003). *A science-based case for large-scale simulation*.

Waltz, R. E., Kerbel, G. R., and Milovich, J. (1994). Toroidal gyro-Landau fluid model turbulence simulations in a nonlinear ballooning mode representation with radial modes, *Physics of Plasmas*, **1** (7): 2229–2244.

Weisser, D., et al. (2006). Performance of applications on the Cray XT3, In *Proceedings of the Cray Users Group 2006 Annual Meeting*, Lugano, Switzerland.

Williamson, D. L. and Olson, J. G. (1994). Climate simulations with a semi-Lagrangian version of the NCAR community climate model, *Monthly Weather Review*, **122**(7): 1594–1610.

Worley, P. H. (2006). CAM performance on the X1E and XT3, In *Proceedings of the 48th Cray User Group Conference*, Eagen, MN, Cray User Group, Inc.