Early Evaluation of the Cray XT3

Jeffrey S. Vetter Sadaf R. Alam Thomas H. Dunigan, Jr.
Mark R. Fahey Philip C. Roth Patrick H. Worley

Oak Ridge National Laboratory
Oak Ridge, TN, USA 37831
{vetter,alamsr,duniganth,faheymr,rothpc,worleyph}@ornl.gov

Abstract

Oak Ridge National Laboratory recently received delivery of a 5,294 processor Cray XT3. The XT3 is Cray's third-generation massively parallel processing system. The system builds on a single processor node—built around the AMD Opteron—and uses a custom chip—called SeaStar—to provide interprocessor communication. In addition, the system uses a lightweight operating system on the compute nodes. This paper describes our initial experiences with the system, including micro-benchmark, kernel, and application benchmark results. In particular, we provide performance results for strategic Department of Energy applications areas including climate and fusion. We demonstrate experiments on the installed system, scaling applications up to 4,096 processors.

1 Introduction

Computational requirements for many large-scale simulations and ensemble studies of vital interest to the Department of Energy (DOE) exceed what is currently offered by any U.S. computer vendor. As illustrated in the DOE Scales report [32] and the High End Computing Revitalization Task Force report [18], examples are numerous, ranging from global climate change research to combustion to biology.

Performance of the current class of HPC architectures is dependent on the performance of the memory hierarchy, ranging from the processor-to-cache latency and bandwidth to the latency and bandwidth of the interconnect between nodes in a cluster, to the latency and bandwidth in accesses to the file system. With increasing chip clock rates and number of functional units per processor and the lack of corresponding improvements in memory access latencies, this dependency will only increase. Single processor performance, or the performance of a small system, is relatively simple to determine. However, given reasonable sequential performance, the metric of interest in evaluating the ability of a system to achieve multi-Teraop performance 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 X1 [1], the SGI Altix 3700 [13], and the Cray XD1 [15]. This report describes initial evaluation results collected on an early version of the Cray XT3 sited at ORNL. Recent results are also publicly available from the ORNL evaluation web site [25]. We have been working closely with Cray, Sandia National Laboratory, and Pittsburgh Supercomputing Center to install and evaluate our XT3.

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 [29] systems. As in these previous systems, the XT3 builds upon a single processor node, or processing element (PE). However, unlike the T3D and T3E, the XT3 uses a commodity microprocessor—the AMD Opteron—at its core. The XT3 connects these processors with a customized interconnect managed by a Cray-designed Application-Specific Integrated Circuit (ASIC) called SeaStar.

2.1 Processing Elements

As Figure 1 shows, each PE has one Opteron processor with its own dedicated memory and communication resource. 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. In contrast, the service PEs run SuSE Linux and are configured for I/O, login, network, or system functions.

The XT3 uses a blade approach for achieving high processor density per system cabinet. On the XT3, a compute blade holds four compute PEs (or nodes), and eight blades are contained in a chassis. Each XT3 cabinet holds three chassis, for a total of 96 compute processors per cabinet. In contrast, service blades host two service PEs and provide PCI-X slots..

The ORNL XT3 uses Opteron model 150 processors. As Figure 2 shows, this model includes an Opteron core, integrated memory controller, three 16b-wide 800 MHz HyperTransport (HT) links, and L1 and L2 caches. The

1-4244-0054-6/06/$20.00

Opteron core has three integer units and one floating point unit capable of two floating-point operations per cycle [3]. Because the processor core is clocked at 2.4 GHz, the peak floating point rate of each compute node is 4.8 GFlops.

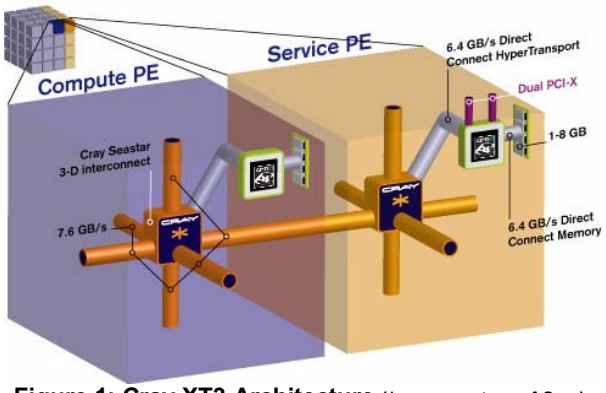

Figure 1: Cray XT3 Architecture (*Image courtesy of Cray*).

The memory structure of the Opteron consists of a 64KB 2-way associative L1 data cache, a 64KB 2-way associative L1 instruction cache, and a 1MB 16-way associative, unified L2 cache. The Opteron has sixteen 64b integer registers, eight 64b floating point registers, sixteen 128b SSE/SSE2 registers, and uses 48b virtual addresses and 40b physical addresses. The memory controller data width is 128b. 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 x 512 mbit parts that support Chipkill. The peak memory bandwidth per processor is 6.4 GB/s.

In general, processors that support SMP configurations have larger memory access latencies than those that do not support SMP configurations, due to the additional circuitry for coordinating memory accesses and for managing memory coherence across SMP processors. Furthermore, on-chip memory controllers enable smaller access latencies than off-chip memory controllers (called "Northbridge" chips in Intel chipsets). The Opteron 150 used in our XT3 does not support SMP configurations because none of its HyperTransport links support the coherent HyperTransport protocol. 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. These observations are quantified in Section 4.1.

The Opteron's processor core has a floating-point execution unit (FPU) that handles all register operations for x87 instructions, 3DNow! operations, all MMX operations, and all SSE and SSE2 operations. This FPU contains a scheduler, a register file, a stack renaming unit, a register renaming unit, and three parallel execution units. One execution unit is known as the adder pipe (FADD); it contains a MMX ALU/shifter and floating-point adder. The next execution unit is known as the multiplier (FMUL); it provides the floating-point multiply/divide/square root operations and also an MMX ALU. The final unit supplies floating-point load/store (FSTORE) operations.

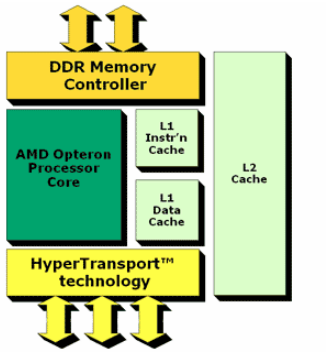

Figure 2: AMD Opteron Design (*Image courtesy of AMD*).

2.2 Interconnect

Each Opteron processor is directly connected to the XT3 interconnect via a Cray SeaStar chip (see Figure 1). This SeaStar chip is a routing and communications chip and it 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 path. SeaStar provides six high-speed network links to connect to neighbors in a 3D torus/mesh topology. Each of the six links has a peak bandwidth of 7.6 GB/s with sustained bandwidth of around 4 GB/s. In the XT3, the interconnect carries all message passing traffic as well as I/O traffic to the system's Lustre parallel file system.

Each SeaStar ASIC contains:

- **HyperTransport link** [4]—this enables the SeaStar chip to communicate with the Opteron processor over parallel links without bus arbitration overheads.
- **PowerPC 440 processor**—this communications and management processor cooperates with the Opteron to synchronize and to schedule communication tasks.
- **Direct Memory Access (DMA) engine**—the DMA engine and the PowerPC processor work together to off-load message preparation and demultiplexing tasks from the Opteron processor.
- **router**—the router provides six network links to the six neighboring processors in a 3D torus topology.
- **service port**—this port bridges between the separate management network and the Cray SeaStar local bus. The service port allows the management system to access all registers and memory in the system and facilitates booting, maintenance and system monitoring. Furthermore, this interface can be used to reconfigure the router in the event of failures.

The ORNL Cray XT3 has 56 cabinets holding 5,212 compute processors and 82 service processors. Its nodes are connected in a three-dimensional mesh of size 14 x 16 x 24, with torus links in the first and third dimension.

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 [23], the Cplant [7, 27], and Red Storm [6]. 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. Catamount is the latest in a sequence of lightweight kernel operating systems developed at Sandia and the University of New Mexico, including SUNMOS [21], Puma [33], and Cougar. (Cougar is the product name for the port of Puma to the Intel ASCI Red system.) For scalability and performance predictability, each instance of the Catamount kernel runs only one single-threaded process and does not provide services like 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 [8] 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. Portals supports multiple higher-level communication protocols, including protocols for MPI message passing between application processes and for transferring data to and from I/O service PEs.

The Cray XT3 programming environment includes compilers, communication libraries, and correctness and performance tools [11]. The Portland Group's C, C++, and Fortran compilers are available. Cray-provided compiler wrappers ease the development of parallel applications for the XT3 by automatically including compiler and linker switches needed to use the XT3's communication libraries. The primary XT3 communication libraries provide the standard MPI-2 message passing interface and Cray's SHMEM interface. Low level communication can be performed using the Portals API. The Etnus TotalView debugger is available for the XT3, and Cray provides the Apprentice^2 tool for performance analysis.

The primary math library is the AMD Core Math Library (ACML). It incorporates BLAS, LAPACK and FFT routines, and is optimized for high performance on AMD platforms. This library is available both as a 32-bit library for compatibility with legacy x86 applications, and as a 64-bit library that is designed to fully exploit the large memory space and improved performance offered by the new AMD64 architecture.

3 Evaluation Overview

As a function of the Early Evaluation project at ORNL, numerous systems have been rigorously evaluated using important DOE applications. Recent evaluations have included the Cray X1 [12], the SGI Altix 3700 [13], and the Cray XD1 [15].

The primary goals of these evaluations are to 1) determine the most effective approaches for using the 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 number of processors. We employ a hierarchical, staged, and open approach to the evaluation, examining low-level functionality of the system first, and then using these results to guide and understand the evaluation using kernels, compact applications, and full application codes. The distinction here is that the low-level benchmarks, for example, message passing, and the kernel benchmarks are chosen to model important features of a full application. This approach is also important because several of the platforms contain novel architectural features that make it difficult to predict the most efficient coding styles and programming paradigms. Performance activities are staged to produce relevant results throughout the duration of the system installation. For example, subsystem performance will need to be measured as soon as a system arrives, and measured again following a significant upgrade or system expansion.

For comparison, performance data is also presented for the following systems:

- Cray X1 at ORNL: 512 Multistreaming processors (MSP), each capable of 12.8 GFlops/sec for 64-bit operations. Each MSP is comprised of four single streaming processors (SSPs). The SSP uses two clock frequencies, 800 MHz for the vector units and 400 MHz for the scalar unit. Each SSP is capable of 3.2 GFlops/sec for 64-bit operations.
- Cray X1E at ORNL: 1024 Multistreaming processors (MSP), each capable of 18 GFlops/sec for 64-bit operations. Each MSP is comprised of four single streaming processors (SSPs). The SSP uses two clock frequencies, 1130 MHz for the vector units and 565 MHz for the scalar unit. Each SSP is capable of 4.5 GFlops/sec for 64-bit operations. This system is an upgrade of the original Cray X1 at ORNL.
- Cray XD1 at ORNL: 144 AMD 2.2GHz Opteron 248 processors, configured as 72, 2-way SMPs with 4GB of memory per processor. The processors are

interconnected by Cray's proprietary RapidArray interconnect fabric.

- Earth Simulator: 640 8-way vector SMP nodes and a 640x640 single-stage crossbar interconnect. Each processor has 8 64-bit floating point vector units running at 500 MHz.
- SGI Altix at ORNL: 256 Itaninium2 processors and a NUMAlink switch. The processors are 1.5 GHz Itanium2. The machine has an aggregate of 2 TB of shared memory.
- SGI Altix at NASA: Twenty Altix 3700 Bx2 nodes, where each node contains 512 Itanium2 processors running at 1.6 GHz with SGI's NUMAflex interconnect. We used two such nodes, connected by a NUMAlink4 switch.
- IBM p690 cluster at ORNL: 27 32-way p690 SMP nodes and an HPS interconnect. Each node has two HPS adapters, each with two ports. The processors are the 1.3 GHz POWER4.
- IBM SP at the National Energy Research Supercomputer Center (NERSC): 184 Nighthawk(NH) II 16-way SMP nodes and an SP Switch2. Each node has two interconnect interfaces. The processors are the 375MHz POWER3-II.
- IBM Blue Gene/L at ANL: a 1024-node Blue Gene/L system at Argonne National Laboratory. Each Blue Gene/L processing node consists of an ASIC with two PowerPC processor cores, on-chip memory and communication logic. The total processing power per node is 2.8 GFlops per processor or 5.6 GFlops per processing node.
- IBM POWER5 at ORNL: An IBM 9124-720 system with two dual-core 1.65 GHz POWER5 processors and 16 GB of memory, running Linux.

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. The standard benchmarks allow 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 with respect to the target applications.

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 XT3 has a single processor node and it uses a lightweight operating system, we focus only on these areas:

- Arithmetic performance, including varying instruction mix, identifying what limits computational performance.
- Memory-hierarchy performance, including levels of cache and shared memory.
- Message-passing performance, including intra-node, inter-node, and inter-OS image MPI performance for one-way (ping-pong) messages, message exchanges, and collective operations (broadcast, all-to-all, reductions, barriers); message-passing hotspots and the effect of message passing on the memory subsystem are studied.

Table 1: STREAM Triad Performance.

Processor	Triad Bandwidth (GB/s)
Cray XT3	5.1
Cray XD1	4.1
Cray X1 MSP	23.8
IBM p690	2.1
IBM POWER5	4.0
SGI Altix	3.8

Current, detailed micro-benchmark data for all existing evaluations is available at our Early Evaluation website [25].

4.1 Memory Performance

The memory performance of current architectures is a primary factor for performance on scientific applications. Table 1 illustrates the differences in measured memory bandwidth on the triad STREAM 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.

Table 2: Latency to Main Memory.

Platform	Measured 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

As discussed earlier, the choice of the Opteron model 150 was motivated in part to provide low access latency to main memory. As Table 2 shows, our measurements 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, which both support multiprocessor configurations.

The memory hierarchy of the XT3 compute node is obvious when measured with the CacheBench tool [24]. Figure 3 shows that the 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 Scientific Operations

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 4 compares the double-precision floating point performance of a matrix multiply (DGEMM) on a single processor using the vendors' scientific libraries. The XT3 Opteron achieves 4 GFlops using the ACML version 2.6 library, about 83% of peak.

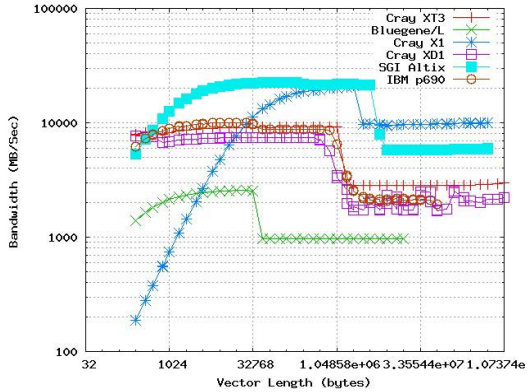

Figure 3: CacheBench read results for a single XT3 compute node.

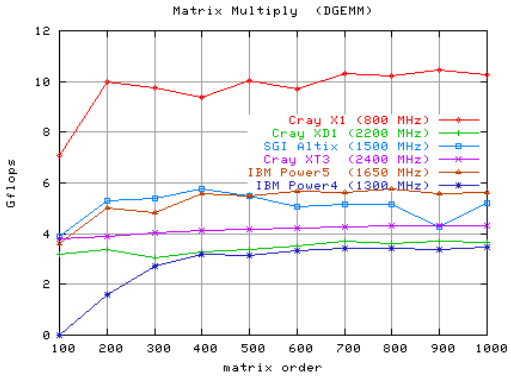

Figure 4: Performance of Matrix Multiply.

Fast Fourier Transforms are another operation important to many scientific and signal processing applications. Figure 5 plots 1-D FFT performance using the vendor library (-lacml, -lscs, -lsci or -lessl), 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 X1 for short vectors.

In general, our micro-benchmark results show the promise of the Cray XT3 compute nodes for scientific computing. Although the Cray X1's high memory bandwidth provided a clear benefit over the other systems we considered, and the SGI Altix and IBM Power5 systems gave better performance for several micro-benchmarks, the XT3 showed solid performance, and in some cases, it performed better than these other systems for short vector lengths.

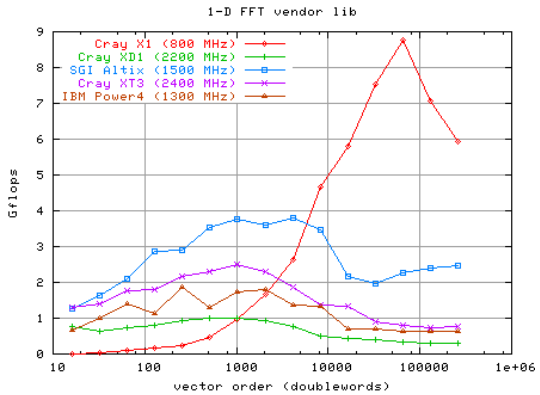

Figure 5: Performance of 1-D FFT using vendor libraries.

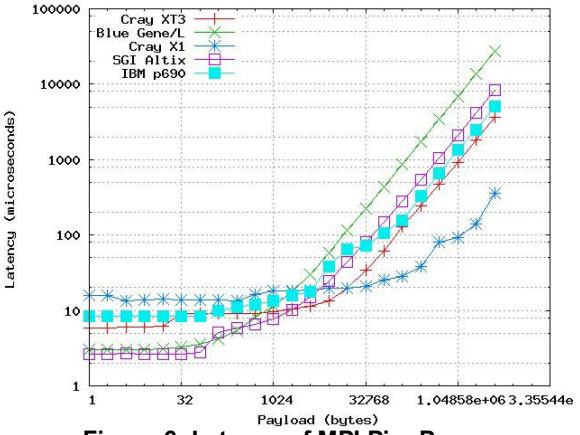

Figure 6: Latency of MPI PingPong.

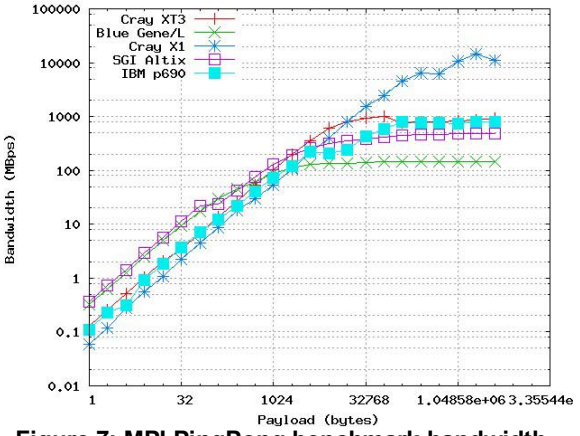

Figure 7: MPI PingPong benchmark bandwidth.

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 [30], we evaluated the

latency and bandwidth of each vendor's MPI implementation. For our evaluation, we used the Pallas MPI Benchmark suite, version 2.2 (now offered by Intel as the Intel MPI Benchmarks).

Figure 6 and Figure 7 shows the latency and bandwidth, respectively, for the Pallas MPI PingPong benchmark. We observe a latency of about 8 microseconds for a 4 byte message, and a bandwidth of about 1.0 GB/s for messages around 64KB.

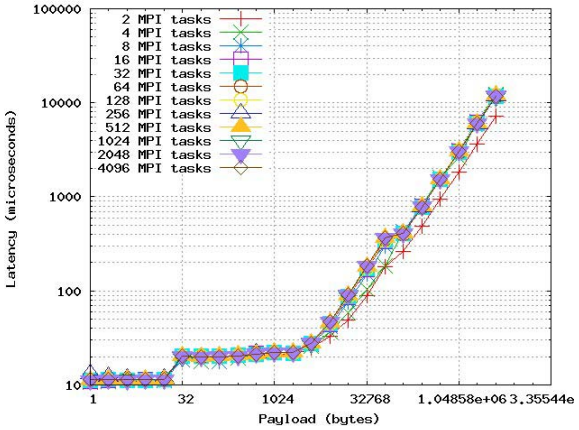

Figure 8: MPI Exchange benchmark latency.

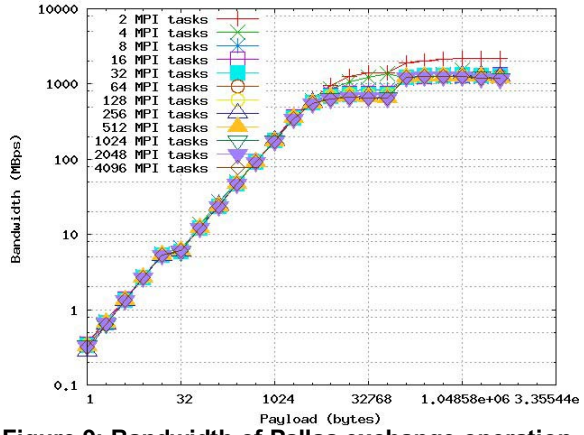

Figure 9: Bandwidth of Pallas exchange operation.

Figure 8 and Figure 9 show the latency and bandwidth (respectively) for the Pallas exchange benchmark at 4,096 processors. This test separates all tasks into two groups, and then uses the MPI_SendRecv operation to transfer data between pairs of tasks, where the endpoints are in separate groups. As opposed to the PingPong operation, which transfers messages between only two tasks, the exchange benchmark has all pairs transferring messages at the same time. The average latency of these transfers is higher, on the order of 20 microseconds for a 4 byte message. The bandwidth is also less than that for the PingPong test, but it reaches an average of nearly 700 MB/s for an individual transfer, in the context of 2,048 simultaneous transfers.

The latency for an MPI_Allreduce operation across various payload sizes and processor counts is shown in Figure 10. The MPI_Allreduce operation is particularly important to several DOE simulation applications because it may be used multiple times within each simulation timestep. Further, its blocking semantics also requires that all tasks wait for its completion before continuing, so latency for this operation is very important for application scalability.

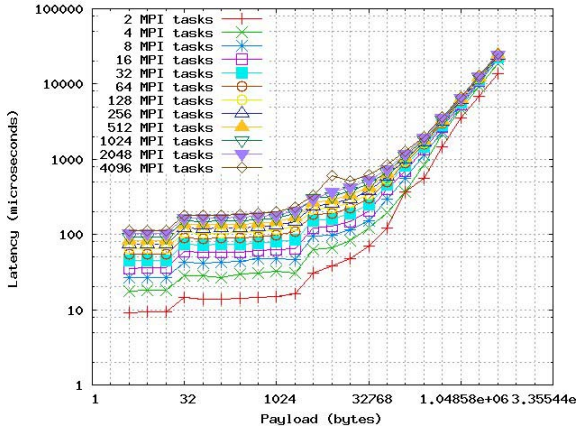

Figure 10: Latency for MPI_Allreduce of across 4,096 processors.

5 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 important applications.

5.1 Parallel Ocean Program (POP)

The Parallel Ocean Program (POP) [19] is the ocean component of CCSM [5] and is developed and maintained at Los Alamos National Laboratory (LANL). 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.

For our evaluation, we used a POP benchmark configuration called x1 that represents a relatively coarse resolution similar to that currently used in coupled climate models. The horizontal resolution is roughly one degree (320x384) 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 a 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.

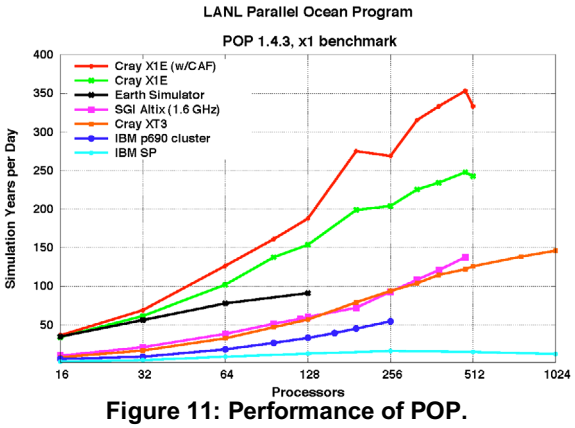

Figure 11: Performance of POP.

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, performance of the barotropic phase is dominated by the performance of a two-dimensional, implicit solver whose performance is very sensitive to network latency and typically scales poorly on all platforms.

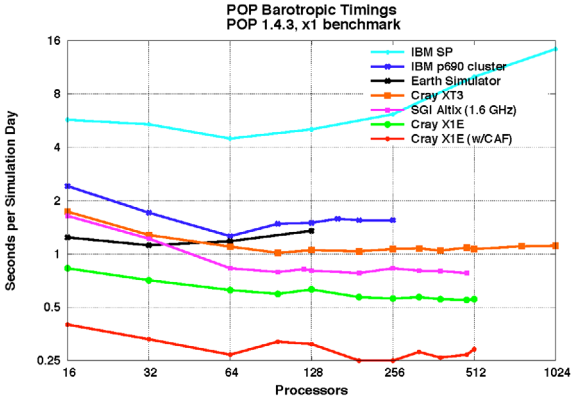

Figure 12: Performance of POP barotropic phase.

Figure 11 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 XT3 performance is similar to that of the SGI Altix up to 256 processors, and continues to scale out to 1024 processors even for this small fixed size problem.

Figure 12 shows the performance of the barotropic portion of POP. While lower latencies on the Cray X1E and SGI Altix systems give them an advantage over the XT3 for this phase, the XT3 showed good scalability in the sense that the cost does not increase significantly out to 1024 processors. Figure 13 shows the performance of the baroclinic portion of POP. The Cray XT3 performance was very similar to that of the SGI Altix, and showed excellent scalability.

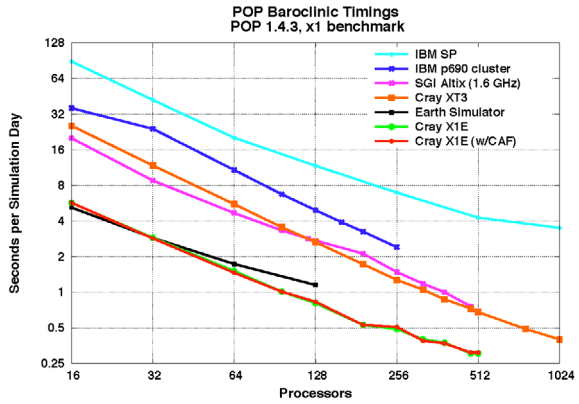

Figure 13: Performance of POP baroclinic phase.

5.2 GYRO

GYRO [9] 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. Ports to new architectures often involve nothing more than the creation of a new makefile.

For our evaluation, we ran GYRO for the B3-GTC benchmark problem. Interprocess communication for this problems is dominated by "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 14 shows platform comparisons of GYRO throughput for the B3-GTC 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 Altix is somewhat superior to the XT3 out to 960 processors, but XT3 scalability is excellent, achieving the best overall performance at 4,096 processors.

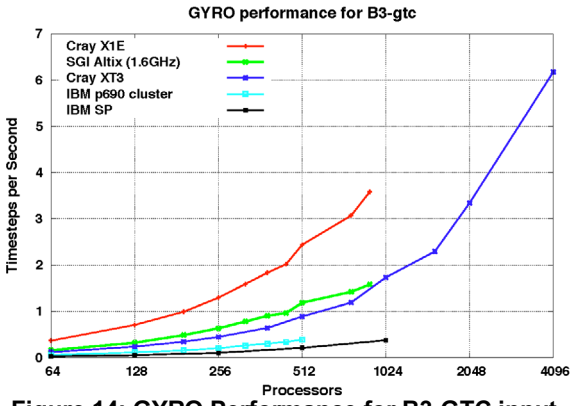

Figure 14: GYRO Performance for B3-GTC input.

5.3 S3D

S3D is a code used extensively to investigate first-of-a-kind fundamental turbulence-chemistry interactions in combustion topics ranging from premixed flames [10, 16], auto-ignition [14], to nonpremixed flames [17, 22, 31]. It is baesd 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 very appropriate for direct numerical simulation of turbulent combustion. The coupling of high-order finite difference methods with explicit Runge-Kutta time integration make very effective use of the available resources, obtaining spectral-like spatial resolution without excessive communication overhead and allowing scalable parallelism.

For our evaluation, the problem configuration is a 3D direct numerical simulation of a slot-burner bunsen flame with detailed chemistry. This includes methane-air chemistry with 17 species and 73 elementary reactions. This simulation used 80 million grid points. The simulation is part of a parametric study performed on different Office of Science computing platforms: the IBM SP at NERSC, the HP Itanium2 cluster at PNNL, and the ORNL Cray X1E and XT3. Figure 15 shows that S3D scales well across the various platforms and exhibited a 90% scaling efficiency on the Cray XT3.

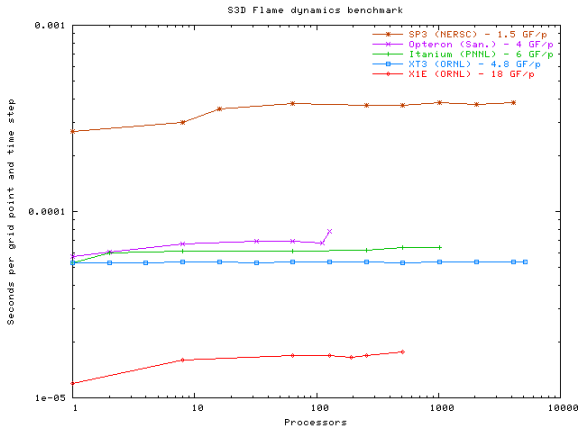

Figure 15: S3D performance.

5.4 Molecular Dynamics Simulations

Molecular dynamics (MD) simulations enable the study of complex, dynamic processes that occur in biological systems [20]. The MD related methods are now routinely used to investigate the structure, dynamics, functions, and thermodynamics of biological molecules and their complexes. The types of biological activity that has been investigated using MD simulations include protein folding, enzyme catalysation, conformational changes associated with bimolecular function, and molecular recognition of proteins, DNA, 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 a 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 atoms to about 100,000.. The time period of simulation can range from pico-seconds to the a few micro-seconds or longer.

Several commercial and open source software frameworks for MD calculations are in use by a large community of biologists, including AMBER [26] and LAMMPS [28]. These packages use slightly different forms of potential function and also their own 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, which is commonly used in biological simulations.

AMBER. 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 parllel systems (including the XT3) in other work [2].

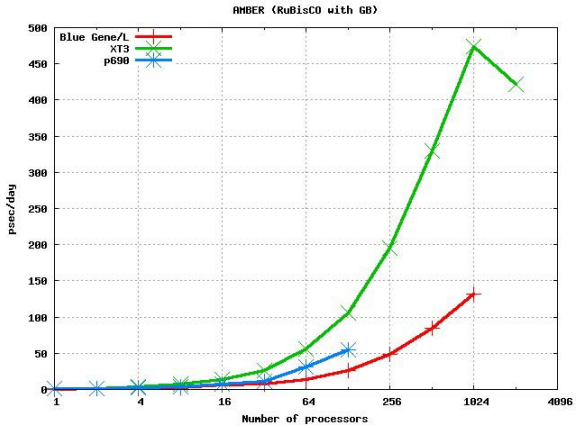

Figure 16: AMBER Simulation Throughput

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 menthod for implicit solvent. The model consists of 73,920 atoms. In Figure 16, we represent the performance of the code in simulation throughput, expressed as simulation pico-seconds per real day (psec/day).The performance on the Cray XT3 is very good for large scale experiments, showing a throughput of over twice the other architectures we investigated.

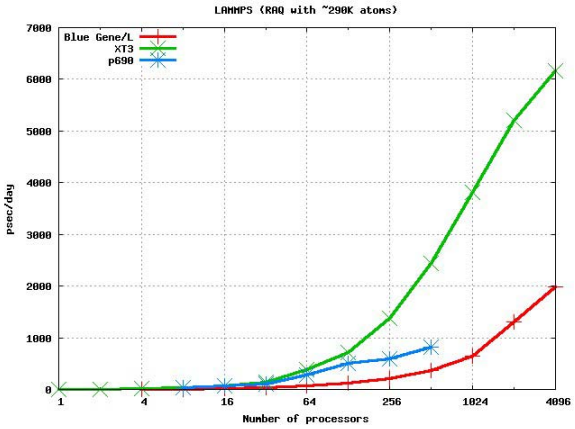

Figure 17: LAMMPS parallel efficiency with approximately 290K atoms

LAMMPS. LAMMPS (Large-scale Atomic/Molecular Massively Parallel Simulator) [28] 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 which 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 17), with over 60% efficiency on up to 1024 processors and over 40% efficiency on 4096 processor run.

6 Conclusions and Plans

Oak Ridge National Laboratory has received and installed a 5,294 processor Cray XT3. In this paper we describe our performance evaluation of the system as it was being deployed, including micro-benchmark, 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, with good scalar processor performance and high interconnect bandwidth when compared to other microprocessor-based systems.

Acknowledgements

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 all their assistance in this endeavor.

We gratefully acknowledge Jeff Kuehn of ORNL for collection of performance data on the BG/L system and Hongzhang Shan of LBL for collection of performance data on the IBM SP. 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.

References

[1] P.A. Agarwal, R.A. Alexander *et al.*, "Cray X1 Evaluation Status Report," ORNL, Oak Ridge, TN, Technical Report ORNL/TM-2004/13, 2004, http://www.csm.ornl.gov/evaluation/PHOENIX/PDF/CRAYEvaluationTM2004-15.pdf.

[2] S.R. Alam, P. Agarwal *et al.*, "Performance characterization of bio-molecular simulations using molecular dynamics," Proc. ACM SIGPLAN Symposium on Principles and Practice of Parallel Programming (PPOPP), 2006.

[3] AMD, "Software Optimization Guide for AMD Athlon™ 64 and AMD Opteron™ Processors," Technical Manual 25112, 2004.

[4] D. Anderson, J. Trodden, and MindShare Inc., *HyperTransport system architecture*. Reading, MA: Addison-Wesley, 2003.

[5] M.B. Blackmon, B. Boville *et al.*, "The Community Climate System Model," *BAMS*, 82(11):2357-76, 2001.

[6] R. Brightwell, W. Camp *et al.*, "Architectural Specification for Massively Parallel Computers-An Experience and Measurement-Based Approach," *Concurrency and Computation: Practice and Experience*, 17(10):1271-316, 2005.

[7] R. Brightwell, L.A. Fisk *et al.*, "Massively Parallel Computing Using Commodity Components," *Parallel Computing*, 26(2-3):243-66, 2000.

[8] R. Brightwell, R. Riesen *et al.*, "Portals 3.0: Protocol Building Blocks for Low Overhead Communication," Proc. Workshop on Communication Architecture for Clusters (in conjunction with International Parallel & Distributed Processing Symposium), 2002, pp. 164-73.

[9] J. Candy and R. Waltz, "An Eulerian gyrokinetic-Maxwell solver," *J. Comput. Phys.*, 186(545), 2003.

[10] J.H. Chen and H.G. Im, "Stretch effects on the Burning Velocity of turbulent premixed ydrogen-Air Flames," Proc. Comb. Inst, 2000, pp. 211-8.

[11] Cray Incorporated, "Cray XT3 Programming Environment User's Guide," Reference Manual S-2396-10, 2005.

[12] T.H. Dunigan, Jr., J.S. Vetter *et al.*, "Performance Evaluation of the Cray X1 Distributed Shared Memory Architecture," *IEEE Micro*, 25(1):30-40, 2005.

[13] T.H. Dunigan, Jr., J.S. Vetter, and P.H. Worley, "Performance Evaluation of the SGI Altix 3700," Proc. International Conf. Parallel Processing (ICPP), 2005.

[14] T. Echekki and J.H. Chen, "Direct numerical simulation of autoignition in non-homogeneous hydrogen-air mixtures," *Combust. Flame*, 134:169-91, 2003.

[15] M.R. Fahey, S.R. Alam *et al.*, "Early Evaluation of the Cray XD1," Proc. Cray User Group Meeting, 2005, pp. 12.

[16] E.R. Hawkes and J.H. Chen, "Direct numerical simulation of hydrogen-enriched lean premixed methane-air flames," *Combust. Flame*, 138(3):242-58, 2004.

[17] E.R. Hawkes, R. Sankaran *et al.*, "Direct numerical simulation of turbulent combustion: fundamental insights towards predictive models," Proc. SciDAC PI Meeting, 2005.

[18] High-End Computing Revitalization Task Force (HECRTF), "Federal Plan for High-End Computing," Executive Office of the President, Office of Science and Technology Policy, Washington, DC 2004.

[19] P.W. Jones, P.H. Worley *et al.*, "Practical performance portability in the Parallel Ocean Program (POP)," *Concurrency and Computation: Experience and Practice*(in press), 2004.

[20] M. Karplus and G.A. Petsko, "Molecular dynamics simulations in biology," *Nature*, 347, 1990.

[21] A.B. Maccabe, K.S. McCurley *et al.*, "SUNMOS for the Intel Paragon: A Brief User's Guide," Proc. Intel Supercomputer Users' Group, 1994, pp. 245-51.

[22] S. Mahalingam, J.H. Chen, and L. Vervisch, "Finite-rate chemistry and transient effects in direct numerical simulations of turbulent non-premixed flames," *Combust. Flame*, 102(3):285-97, 1995.

[23] T.G. Mattson, D. Scott, and S.R. Wheat, "A TeraFLOP Supercomputer in 1996: The ASCI TFLOP System," Proc. 10th International Parallel Processing Symposium (IPPS 96), 1996, pp. 84-93.

[24] P.J. Mucci, K. London, and J. Thurman, "The CacheBench Report," University of Tennessee, Knoxville, TN 1998.

[25] Oak Ridge National Laboratory, *Early Evaluation Website*, http://www.csm.ornl.gov/evaluation, 2005.

[26] D.A. Pearlman, D.A. Case *et al.*, "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, 1995.

[27] K. Pedretti, R. Brightwell, and J. Williams, "Cplant Runtime System Support for Multi-Processor and Heterogeneous Compute Notes," Proc. IEEE International Conference on Cluster Computing (CLUSTER 2002), 2002, pp. 207-14.

[28] S.J. Plimpton, "Fast Parallel Algorithms for Short-Range Molecular Dynamics," in *Journal of Computational Physics*, vol. 117, 1995

[29] S.L. Scott, "Synchronization and Communication in the T3E Multiprocessor," Proc. Architectural Support for Programming Languages and Operating Systems (ASPLOS), 1996, pp. 26-36.

[30] M. Snir, S. Otto *et al.*, Eds., *MPI--the complete reference*, 2nd ed. Cambridge, MA: MIT Press, 1998.

[31] J.C. Sutherland, P.J. Smith, and J.H. Chen, "Quantification of Differential Diffusion in Nonpremixed Systems," *Combust. Theory and Modelling (to appear)*, 2005.

[32] US Department of Energy Office of Science, "A Science-Based Case for Large-Scale Simulation," US Department of Energy Office of Science 2003, http://www.pnl.gov/scales.

[33] S.R. Wheat, A.B. Maccabe *et al.*, "PUMA: An Operating System for Massively Parallel Systems," *Journal of Scientific Programming (special issue on operating system support for massively parallel systems)*, 3(4):275-88, 1994.