ORNL Cray X1 Evaluation Status Report

P. K. Agarwal[1], R. A. Alexander[1], E. Apra[2], S. Balay[3], A. S. Bland[1], J. Colgan[4], E. F. D'Azevedo[1], J. J. Dongarra[1,5], T. H. Dunigan, Jr.[1], M. R. Fahey[1], R. A. Fahey[1], A. Geist[1], M. Gordon[2], R. J. Harrison[1], D. Kaushik[3], M. Krishnakumar[2], P. Luszczek[5], A. Mezzacappa[1], J. A. Nichols[1], J. Nieplocha[2], L. Oliker[6], T. Packwood[7], M.S. Pindzola[4], T. C. Schulthess[1], J. S. Vetter[1], J. B. White, III[1], T. L. Windus[2], P. H. Worley[1], T. Zacharia[1]

[1]Oak Ridge National Laboratory, [2]Pacific Northwest National Laboratory, [3]Argonne National Laboratory, [4] Auburn University, [5]University of Tennessee, [6]Lawrence Berkeley National Laboratory, [7]Cray Inc.

Abstract

On August 15, 2002 the Department of Energy (DOE) selected the Center for Computational Sciences (CCS) at Oak Ridge National Laboratory♦ (ORNL) to deploy a new scalable vector supercomputer architecture for solving important scientific problems in climate, fusion, biology, nanoscale materials and astrophysics. "This program is one of the first steps in an initiative designed to provide U.S. scientists with the computational power that is essential to 21st century scientific leadership," said Dr. Raymond L. Orbach, director of the department's Office of Science

In FY03, CCS procured a 256-processor Cray X1 to evaluate the processors, memory subsystem, scalability of the architecture, software environment and to predict the expected sustained performance on key DOE applications codes. The results of the micro-benchmarks and kernel benchmarks show the architecture of the Cray X1 to be exceptionally fast for most operations. The best results are shown on large problems, where it is not possible to fit the entire problem into the cache of the processors. These large problems are exactly the types of problems that are important for the DOE and ultra-scale simulation.

Application performance is found to be markedly improved by this architecture:

- Large-scale simulations of high-temperature superconductors run 25 times faster than on an IBM Power4 cluster using the same number of processors.
- Best performance of the parallel ocean program (POP v1.4.3) is 50 percent higher than on Japan's Earth Simulator and 5 times higher than on an IBM Power4 cluster.
- A fusion application, global GYRO transport, was found to be 16 times faster on the X1 than on an IBM Power3. The increased performance allowed simulations to fully resolve questions raised by a prior study.
- The transport kernel in the AGILE-BOLTZTRAN astrophysics code runs 15 times faster than on an IBM Power4 cluster using the same number of processors.
- Molecular dynamics simulations related to the phenomenon of photon echo run 8 times faster than previously achieved.

Even at 256 processors, the Cray X1 system is already outperforming other supercomputers with thousands of processors for a certain class of applications such as climate modeling and some fusion applications.

This evaluation is the outcome of a number of meetings with both high-performance computing (HPC) system vendors and application experts over the past 9 months and has received broad-based support from the scientific community and other agencies.

1 Evaluation Plan for Cray X1

This paper documents progress in the evaluation of the Cray X1 supercomputer installed at the Center for Computational Sciences (CCS) at Oak Ridge National Laboratory (ORNL). The original evaluation plan is described in the Cray X1 Evaluation Plan, ORNL/TM-2003/67 http://www.csm.ornl.gov/evaluation/PHOENIX/PDF/CrayX1-Evaluation-Plan.pdf, dated March 2003. The extended version of this report is available as technical report ORNL/TM-2004/13. Current, detailed configuration information for the systems mentioned in this document is available at http://www.ccs.ornl.gov.

Figure 1: Front view of 256-MSP, 8-cabinet Cray X1 at ORNL's Center for Computational Sciences.

1.1 Motivation

In May of 2002, on the heels of an article in the New York Times proclaiming a crisis in U.S. supercomputing brought on by the Japanese Earth Simulator's performance of 35.8 TFLOP/s on Linpack and over 27 TFLOP/s on a climate model, the Office of Science of the U.S. Department of Energy held a meeting in Washington, D.C., to formulate a response to the threat to U.S. science from the Earth Simulator. In this meeting, Jim Hack of the National Center for Atmospheric Research (NCAR) discussed the abysmal state of U.S. climate simulation capability and challenged the high performance computing (HPC) community to address this problem. The Center for Computational Sciences held meetings with IBM, Cray, HP, and SGI and included computational scientists from climate modeling, materials science, and fusion simulation to understand the capabilities of each manufacturer in building a system to deliver 40 to 100 TFLOP/s of sustained performance on real applications. From this series of meetings, it became clear that most of the U.S. manufacturers were going to continue down a path of delivering large clusters of symmetrical multi-processor systems linked together by either custom or commodity interconnect systems. While these systems have delivered outstanding gains in capacity and price/performance, it was clear to the computational scientists that were part of these discussions that the U.S. was unlikely to recapture the lead in high-end computing measured by sustained performance on a suite of large applications if some changes were not made.

To address this concern, the CCS began discussions with Cray Inc. to evaluate the Cray X1 and its follow-on systems as candidates for what has become known as an Ultra-Scale Simulation computer system. The Cray X1 was intriguing on a number of fronts. The raw performance of the custom vector processor is 12.8 GFLOP/s, more than 50% faster than the SX-6 processor of the Earth Simulator. Nevertheless, fast processors that can only deliver a tiny fraction of their peak speed to real applications have been readily observed. Secondly, the Cray X1 has very high memory bandwidth. Each node board has four processors and more than 200 GB/s of memory bandwidth. This bandwidth dwarfs that of other systems. While not as high (on a bytes per second per FLOP/s basis) as the heyday of vector processors, it is only a factor of two less than the saturation bandwidth needed for the vector processors, an amount that can easily be made up using caching algorithms. Compare this with the IBM Power4 which has only one-tenth of the bandwidth needed to saturate the arithmetic units. A third intriguing aspect of the Cray X1 is the shared memory architecture with very high bandwidth and low latency connections between the nodes. Finally, the vector instruction set tackles one of the bottlenecks in high end computing: the instruction decode and retirement bottleneck of many microprocessors. Taking all of these factors together, it appeared that the Cray X1 offered a truly different choice and one that had a very good chance to deliver high performance for scientific applications.

The CCS submitted a proposal to the Office of Science for the evaluation of the Cray X1. This proposal outlined a series of systems, starting with a 32-processor Cray X1 and eventually culminating in a machine of over 100 TFLOP/s using the follow-on Cray Black Widow architecture. In August 2002, Dr. Raymond Orbach, director of the Office of Science, announced that he was accepting this proposal by funding the initial phases up through a 256-processor system for evaluation. This document provides a status report on progress after nine months of evaluation.

The early results have been very promising. On a fusion application, results are being seen that are 15 times better than on the same number of processors of an IBM Power system, resulting in new scientific capability that is not available on any other computer in the Office of Science facilities. More importantly, on an astrophysics kernel, the results show that the Cray X1 is faster than the IBM Power4 on small problems and this advantage grows as the problem size grows due to a combination of memory bandwidth, interconnect performance, and processor speed. This is exactly the kind of scaling that is required to achieve the goals of an Ultra-Scale Simulation program.

All this performance does come at a cost. Over the last 15 years, most scientific application codes have been either designed or rewritten for massively parallel computers using scalar, cache-based microprocessors. It was thought that the days of vector processors were gone and little consideration was given to vector architectures when redesigning the applications. The result is that most codes require at least some modifications, directives, or restructuring to achieve good performance on the Cray X1. At best, a simple recompilation with the right compiler flags can achieve good results. At worst, major rework of the code is required to allow the vector units to achieve good results. Initial work indicates that most codes require a relatively small amount of effort, usually a few days to two weeks of work by someone with experience on a vector machine. Moreover, many of the optimizations for parallel, cache-based systems end up being important for the scalable vector architecture of the X1. However, this should not be construed to mean that the Cray X1 is the right machine for every application. There are a number of applications that have been tuned to run at a very high percentage of the peak performance on a cluster-based system. For these applications, clusters will always be more cost effective than the X1. However, for the most challenging mission applications, the Cray X1 is proving to be an excellent machine, and one that is certainly worth further study at scale.

The Cray X1 is the evaluation machine; the goal of the project is to deploy an ultra-scale simulation system in 2006. In this timeframe, the follow-on product, code named Black Widow, is the system that Cray would propose. The Black Widow system is being designed to correct some of the deficiencies in the current X1 product and to deliver a seven-fold improvement in price/performance ratio. As a result of the current collaboration between Cray and the CCS, Cray is making important design changes in the Black Widow system. While the nature of these changes are proprietary and cannot be included in an open report, it is important to note that Cray engineers and management are eager to work with the CCS to improve the product and that

Cray is addressing every deficiency that is identified.

2 Evaluation Overview

The ongoing goal of this evaluation is to predict the expected sustained performance of future Cray systems running at tens to hundreds of TFLOP/s on large-scale simulations, using actual applications codes from climate, fusion, materials science, and chemistry. These initial applications were selected for evaluation in discussions with application scientists at workshops held at ORNL in the fall of 2002. Additional workshops with wider application-community participation were held in February and March of 2003 to build consensus on the application and evaluation plans. The organization and progress of the evaluation and specific sub-goals are described in more detail below.

The primary tasks of this continuing evaluation are to:

- evaluate benchmark and application performance and compare with systems from other HPC vendors,
- determine the most effective approaches for using the Cray X1,
- evaluate system and system administration software reliability and performance,
- predict scalability, both in terms of problem size and number of processors, and
- collaborate with Cray to incorporate this information and experience into their next generation designs.

While the performance of individual application kernels may be predicted with detailed performance models and limited benchmarking, the performance of full applications and the suitability of this system as a production scientific computing resource can only be determined through extensive experiments conducted on a resource large enough to run datasets representative of the target applications. It is not possible to fully model the interactions of computation, communication, input/output (I/O) and other system services, load balancing, job scheduling, networking, resource allocation, compilers, math libraries, and other system software, along with application software and terabytes of data in memory and on disk.

To improve the performance of both applications and systems, to predict performance on larger or future systems, and to draw concrete conclusions from system and application level benchmarks, a fundamental understanding of the machine and application kernels is also essential. In essence, this evaluation plan will result in a much more detailed "matrix" for selected applications on the X1 along with detailed interpretation of the raw information, thereby providing a firm basis for future decisions in high-performance scientific computing.

2.1 Methodology

The evaluation has been hierarchical, staged, and open. In the hierarchical approach employed in the investigation of the X1 architecture, the low-level functionality of the system was examined first. Results were then used to guide and understand the evaluation using kernels and full application codes. This approach is important because the X1 embodies a number of novel architectural features that can make it difficult to predict the most efficient coding styles and programming paradigms. Standard benchmarks were used when appropriate, to compare with the performance of other systems, but the emphasis of this evaluation is on application-driven studies.

A hierarchical approach has also been employed in the examination of performance issues. The first step is to establish functional correctness. The second step is to establish performance correctness. Performance expectations are the motivation behind the design and the DOE evaluation of the X1. Verification of these expectations is important to identify and correct hardware and software implementation errors. The third step is to determine "best practice" for the X1, that is, what coding styles and programming paradigms achieve the best performance.

Finally, a hierarchical approach has been applied to the evaluation of system scalability. As larger systems have been installed, variants of earlier experiments have been repeated that examine performance in each new, larger setting. It has been especially important to re-evaluate functional and performance correctness as the system size increases, as these have been problematic in previous large systems from other vendors.

Performance activities have been and will continue to be staged to produce relevant results throughout the duration of the evaluation. For example, subsystem performance is measured as soon as each system arrives, and measured again following a significant upgrade or system expansion. Fortunately, these experiments are conducted relatively quickly. In contrast, the porting, tuning, and performance analysis of complex applications, such as those involving unstructured or dynamic data structures, take much longer to complete, and were started as soon as possible to have an impact on the evaluation. Allocation of system time and manpower to the different aspects of the evaluation has also been staged.

The performance evaluation has been and will remain open. One of the strengths of performance evaluations performed within the Office of Science is the ability to fully disclose the test codes and the results. To ensure the correctness of the approach and interpretation of results, evaluation methodologies and results have been described in public forums and on an evaluation web site for public comment (http://www.csm.ornl.gov/evaluation).

3 Cray X1 Overview

3.1 Cray X1 Hardware and Software

The Cray X1 is an attempt to incorporate the best aspects of previous Cray vector systems and massively-parallel-processing (MPP) systems into one design. Like the Cray T90, the X1 has high memory bandwidth, which is key to realizing a high percentage of theoretical peak performance. Like the Cray T3E, the X1 has a high-bandwidth, low-latency, scalable interconnect, and scalable system software. And, like the Cray SV1, the X1 leverages commodity CMOS technology and incorporates non-traditional vector concepts, like vector caches and multi-

streaming processors.

The X1 is hierarchical in processor, memory, and network design. The basic building block is the multi-streaming processor (MSP), which is capable of 12.8 GF/s for 64-bit operations. Each MSP is comprised of four single-streaming processors (SSPs), each with two 32-stage 64-bit floating-point vector units and one 2-way super-scalar unit. 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 GF/s for 64-bit operations. The four SSPs share a 2 MB "Ecache."

The Ecache has sufficient single-stride bandwidth to saturate the vector units of the MSP. The Ecache is needed because the bandwidth to main memory is not enough to saturate the vector units without data reuse - memory bandwidth is roughly half the saturation bandwidth. This design represents a compromise between non-vector-cache systems, like the NEC SX-6, and cache-dependent systems, like the IBM p690, with memory bandwidths an order of magnitude less than the saturation bandwidth. Because of its short cache lines and extra cache bandwidth, random-stride scatter/gather memory access on the X1 is just a factor of two slower than stride-one access, not the factor of eight or more seen with typical cache-based systems like those based on the IBM Power4, HP Alpha, or Intel Itanium. The X1's cache-based design deviates from the full-bandwidth design model only slightly. Each X1 MSP has the single-stride bandwidth of an SX-6 processor; it is the higher peak performance that creates an imbalance. A relatively small amount of data reuse, which most modern scientific applications do exhibit, can enable a very high percentage of peak performance to be realized, and worst-case data access can still provide double-digit efficiencies.

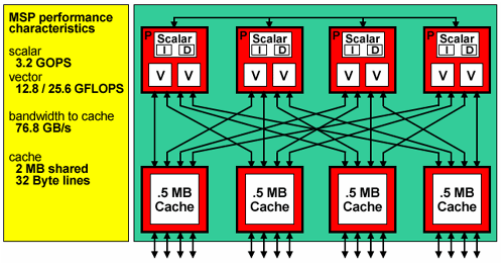

Figure 2: Cray MSP Module.

The X1 compiler's primary strategies for using the eight vector units of a single MSP are multistreaming a (sufficiently long) vectorized loop or multistreaming an unvectorized outer loop. (The compiler is also able to vectorize a "long" outer loop and multistream a shorter inner loop if the dependency analysis allows this.) The effective vector length of the first strategy is 256 elements, the vector length of the NEC SX-6. The second strategy, which attacks parallelism at a different level, allows a shorter vector length of 64 elements for a vectorized loop. Cray also supports the option of treating each SSP as a separate processor.

Four MSPs and a flat, shared memory of 16 GB form a Cray X1 node. The memory banks of a node provide 200 GB/s of bandwidth, enough to saturate the paths to the local MSPs and service requests from remote MSPs. Each bank of shared memory is connected to a number of banks on remote nodes, with an aggregate bandwidth of roughly 50 GByte/sec between nodes. This represents a byte per flop of interconnect bandwidth per computation rate, compared to 0.25 bytes per flop on the Earth Simulator and less than 0.1 bytes per flop expected on an IBM p690 with the maximum number of Federation connections. The collected nodes of an X1 have a single system image.

A single, four-MSP X1 node behaves like a traditional SMP, but each processor has the additional capability of directly addressing memory on any other node (like the T3E). Remote memory accesses go directly over the X1 interconnect to the requesting processor, bypassing the local cache. This mechanism is more scalable than traditional shared memory, but it is not appropriate for shared-memory programming models, like OpenMP1, outside of a given four-MSP node. This remote-memory-access mechanism is a good match for distributed-memory programming models, particularly those using one-sided put/get operations.

Figure 3: Elevated view of 256-MSP, 8-cabinet Cray X1 at ORNL's Center for Computational Sciences.

In large configurations, the X1 nodes are connected in an enhanced 3D torus. This topology has relatively low bisection bandwidth compared to crossbar-style interconnects, such as those on the NEC SX-6 and IBM SP. Whereas bisection bandwidth scales as the number of nodes, O(n), for crossbar-style interconnects, it scales as the 2/3 root of the number of nodes, O(n2/3), for a 3D torus. Despite this theoretical limitation, mesh-based systems, such as the Intel Paragon, the Cray T3E, and ASCI Red, have scaled well to thousands of processors.

3.2 Delivery, Installation, and Operations at CCS

The contract with Cray Inc., for the delivery of the X1 was phased to allow deliveries over a period of nine months. This allowed the CCS to take early delivery of a single cabinet with 32 MSPs as an initial delivery, and follow that with additional MSPs and cabinets as Cray's software capability to support larger configurations grew. The first cabinet was delivered on March 18, 2003, and was installed in the old computer center in building 4500N at ORNL. The system had 32 MSPs, 128 GB of memory, a single I/O cabinet, two Cray Network Subsystems (CNS), one Cray Programming Environment Server (CPES), and 8 TB of disk space. This early configuration allowed initial tests of

the full system, including compilers, networks, file systems, operating system, and applications.

On June 13, 2003, the CCS shut down the computers and moved from the old computer center to the new computer center in the Computational Sciences Building. As part of the move, the X1 was upgraded to a two cabinet configuration with 64 MSPs, 256 GB of memory, and an additional 8 TB of disk space, with two additional CNSs. This configuration allowed tests to determine the impact of additional routers and performance measurements of jobs split over multiple cabinets.

On July 30, cabinets three and four were added to expand the system to a total of 128 MSPs and 512 GB of memory. At this size, many of the flagship applications that run on other CCS resources could now be ported to the X1 for realistic comparison. Evaluators experimented with the order of placement of processes on nodes to see if node order was important for performance.

Table 1: CCS Cray X1 installation progress.

Date	Cabinets	Processors (MSP)	Memory (GB)	I/O (TB)
March 03	1	32	128	8
June 03	2	64	256	16
July 03	4	128	512	16
November 03	8	256	1024	32

The most recent expansion of the system came with the delivery of cabinets five through eight on October 29 and the integration of those cabinets into the system the week of November 15, 2003. The expansion took the system up to 256 MSPs, 1 TB of memory, 32 TB of disk space, eight CNSs, two CPESs, and two I/O cabinets. The full system passed its acceptance test on December 15, 2003.

4 Evaluation Components

4.1 Micro-benchmarks

Both standard and custom benchmarks have been used to characterize the underlying architectural components of the X1. The computational performances of both a single SSP and an MSP have been measured. Using the EuroBen benchmark, hardware performance of add, multiply, divide, and square root as well as software intrinsics (exponentials, trigonometric functions, and logarithms) have been evaluated. An X1 MSP performs about 4 times faster than the Power4 CPU on the IBM p690. Tests demonstrate how performance is affected by vector length, stride, compiler optimizations, and vendor scientific libraries. Figure 4 illustrates the effect of vector length on a 1-D FFT on the X1 and competing architectures. The STREAMS and MAPS benchmarks show the high memory bandwidth the X1 achieves, and Figure 5 shows that the memory performance scales nicely with the number of processors and is competitive with NEC SX-6. Results show that the X1 does not perform well at this time on sparse eigenvalue kernels, but achieves over 90% of peak on dense linear algebra kernels. Tests have been conducted comparing the performance of MPI, SHMEM, coarrays, and UPC. Communication tests include the standard benchmarks (ParkBench and Euroben-dm) to measure latency and bandwidth as a function of message size and distance, as well as custom benchmarks that reflect common communication patterns found in scientific applications (exchange, aggregate reductions, barrier, broadcast). Evaluations of OpenMP performance (up to 16 SSP's) have been conducted on the X1. Using lmbench, the latency in various OS services (file open/close, context switch, virtual memory) has been measured.

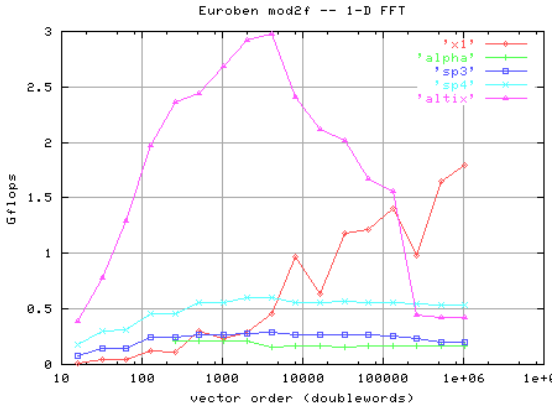

Figure 4: Sustained Flop rate versus 1-D FFT vector length.

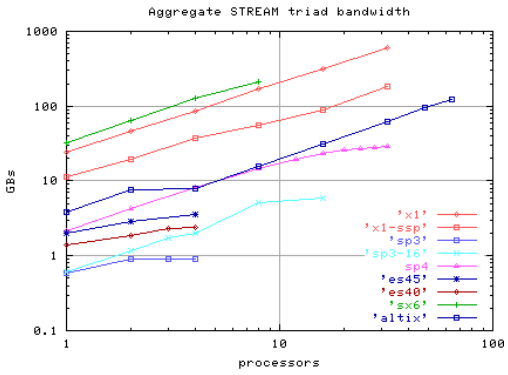

Figure 5: Aggregate stream triad bandwidth.

The micro-benchmarking is nearly complete. Tests will be re-run after any major hardware or software upgrades, and additional low-level I/O benchmarks may need to be developed and analyzed in support of the analysis of the parallel file system.

Detailed performance results are available at http://www.csm.ornl.gov/~dunigan/cray/.

4.2 Kernels and Performance Optimization

The kernel benchmarks bridge the gap between the low-level micro-benchmarks and the resource-intensive application benchmarking. Industry-standard kernels (ParkBench, NAS Parallel Benchmarks, Euroben) have been used as well as kernels that were extracted from the scientific applications. Single processor (MSP) performance and parallel kernels were tested and evaluated with and without the vendor's parallel scientific library. (Some

performance problems have been identified in the Cray parallel library that Cray is correcting.) The performances of these kernels were compared with other architectures and have varied algorithms and programming paradigms (MPI, SHMEM, co-array).

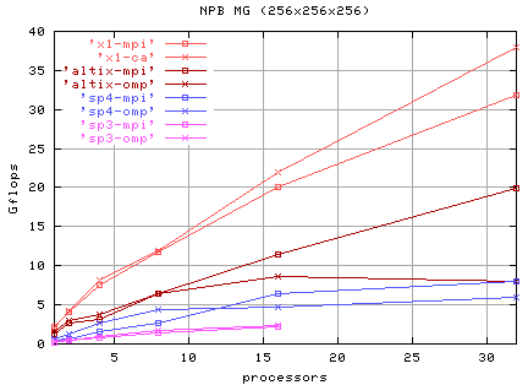

Figure 6: NAS MG scaling.

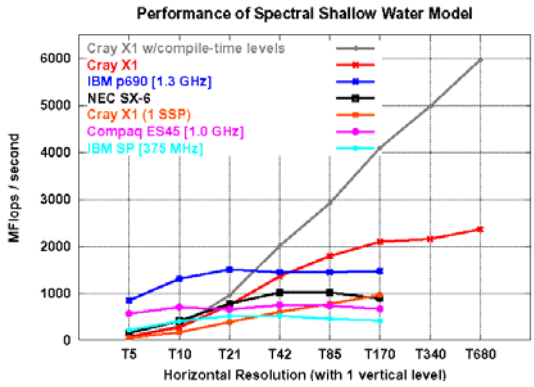

Figure 7: Performance of spectral shallow water model across architectures.

Figure 6 compares the performance of the NAS multi-grid benchmark with various processor counts, architectures, and communication strategies. Extensive comparative analyses were done using a component of the global climate model, the parallel spectral transform shallow-water code (PSTSWM). The PSTSWM kernel supports different problem sizes, algorithms, and programming paradigms. This kernel has been optimized on many parallel computer architectures. On the X1, PSTSWM was used to compare multiple SSP's with MSP's, analyze compiler optimizations (streaming, vectorizing), as well as comparing performance with other supercomputers as shown in Figure 7.

Much of the standard kernel benchmarking is complete, and PSTSWM analysis has provided insight into programming paradigms for the X1. Major hardware and software upgrades may alter the choice of programming paradigm, if so, additional benchmarking will be required. If Cray provides a broader OpenMP implementation, then OpenMP performance will need to be evaluated in relation to MPI, SHMEM, UPC, and co-arrays. Multi-level Parallelism (MLP) or global arrays has not been evaluated yet.

The results of the micro-benchmarks and kernel benchmarks show the architecture of the Cray X1 to be exceptionally fast for most operations. The X1's largest advantages are shown on large problems where it is not possible to fit the entire problem into the cache of the processors. These large problems are exactly the types of problems that are important for the DOE and ultra-scale simulation.

4.3 HPC Challenge Benchmarks

The HPC Challenge benchmark consists at this time of 5 benchmarks: HPL, STREAM, RandomAccess, PTRANS, and Latency/Bandwidth. HPL is the Linpack TPP benchmark. The test stresses the floating point performance of a system. STREAM is a benchmark that measures sustainable memory bandwidth (in GB/s); RandomAccess measures the rate of random updates of memory. PTRANS measures the rate of transfer for larges arrays of data from multiprocessor's memory. Latency/Bandwidth measures (as the name suggests) latency and bandwidth of communication patterns of increasing complexity between as many nodes as is time-wise feasible. Table 2 illustrates the impressive performance of the Cray X1 on a variety of these measurements. See http://icl.cs.utk.edu/hpcc/ for complete details.

Table 2: HPC Challenge benchmark results (condensed version).

Computer	Run	Processors	HPL (system performance)	PTRANS (system performance)	*STREAM Triad (per CPU)	Random Access MPI (per CPU)	Random Ring Latency (per CPU)	Random Ring Bandwidth (per CPU)
System	Type	#	TFlop/s	GB/s	GB/s	Gup/s	usec	GB/s
Cray X1	base	64	0.522	3.229	14.99	0.00521	20.34	0.9407
HP AlphaServer SC45	base	128	0.19	1.507	0.803	0.00133	37.31	0.0278
HP Integrity zx6000	base	128	0.331	4.607	1.956	0.00145	32.44	0.0435
IBM p690	base	64	0.181	0.477	1.148	0.00057	101.96	0.0149
IBM p690	base	256	0.654	0.833	1.19	0.00031	373.99	0.0046
SGI Altix	base	32	0.131	0.7	0.896	0.00152	7.71	0.0258
SGI Altix	base	128	0.52	0.727	0.746	0.00089	24.34	0.0193

4.4 Climate Science

The Community Climate System Model (CCSM) is the primary model for global climate simulation in the U.S. The CCSM represents a massive, distributed-software development project supported jointly by the DOE Office of Science, NSF, and NASA, and it is the target of the climate SciDAC project, "Collaborative Design and Development of the Community Climate System Model for Terascale Computers." Over the past year, CCSM development has been particularly aggressive in preparation for major new science runs for the Intergovernmental Panel on Climate Change (IPCC). In addition to significant new science in areas such as atmospheric chemistry and dynamic land vegetation, the CCSM is undergoing re-parameterization and validation for higher resolutions, all of which significantly increases the computational expense of each run.

During this time of already increased development activity, vectorization has risen to a high priority not just because of the availability of the X1 in the CCS, but thanks to a large allocation of time on the Earth Simulator for the National Center for Atmospheric Research (NCAR) to perform IPCC runs. Adding or returning vectorization to the CCSM components has been a cooperative effort among scientists and application specialists at NCAR, ORNL, LANL, ANL, NEC, Cray, and CRIEPI in Japan. Production IPCC runs on the Earth Simulator began in January 2004, and validation of IPCC components have completed on the X1.

The CCSM is built on four component models: the Community Atmosphere Model (CAM), the Community Land Model (CLM), the Parallel Ocean Program (POP), and the Los Alamos Sea Ice Model (CICE). Derivatives of these are coupled with a fifth component (CPL). Each component has seen significant activity as part of the X1 evaluation.

Early application analysis showed that CLM would be particularly challenging for vectorization because of recent changes to its data structures. After prototyping of vectorizable data structures by the CCS as part of the X1 evaluation, the CLM developers at NCAR and ORNL performed a major rewrite of the code. The new code is now undergoing validation and performance analysis. Early performance results show that the vectorized code is a dramatic improvement even for non-vector systems; on the IBM p690, it is almost twice as fast as the original code. And the vectorized version of the main computational routine is still five times faster on the X1 than on the IBM p690.

CAM, while it has data structures that are more amenable to vectorization, has algorithms that have proven challenging for vectorization. NEC, NCAR, and CRIEPI have made significant modifications to the CAM to support vectorization, while NCAR developers have made additional modifications to support the IPCC science. Work continues to port the many CAM changes to the X1 and incorporate additional optimizations by Cray and ORNL.

Conversely, POP was an early success on the X1. POP consists of two computational phases; one is computation bound and the other is latency bound. The computation-bound phase vectorizes well, and the latency-bound phase takes good advantage of the low-latency interconnect. Scaling has been limited by the Cray MPI implementation, but this implementation has been improving steadily. In the mean time, localized replacements of MPI with Co-Array Fortran have provided leadership performance and scalability on the X1. At 128 processors (MSPs), the X1 is seven times faster than the IBM p690 and 1.5 times faster than the Earth Simulator for a relatively small problem. With additional vector tuning and for larger problems, this advantage is expected to increase.

A vectorized version of the full CCSM is to undergo validation testing on the Earth Simulator, and work has begun to port this full version to the X1 and perform additional tuning. In the meantime, the early success with POP has allowed the CCS to support early production science runs of coupled POP and CICE simulations by LANL scientists. The X1 is currently performing new science runs at a horizontal resolution of 0.4 degrees, and work is under way to perform a "heroic" run at 0.1 degrees.

4.5 Fusion Science

4.5.1 GYRO

Advances in understanding tokamak plasma behavior are continuously being realized, but uncertainties remain in predicting confinement properties and performance of larger reactor-scale devices. The coupled gyrokinetic-Maxwell (GKM) equations provide a foundation for the first-principles calculation of anomalous tokamak transport. GYRO 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.

The current performance is best described by comparing timings across architectures as Table 3 shows. Early performance tests for the benchmark problem DIII-D simulation which simulates the experimental levels of turbulent radial transport showed GYRO performance on 32 MSPs to be closely comparable to 256 Power3 processors.

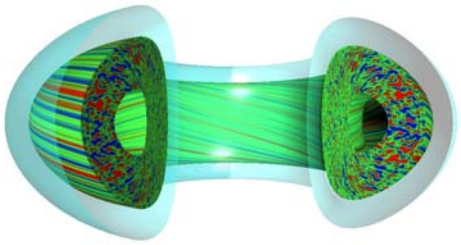

Figure 8: Turbulent potential fluctuations in shaped toroidal geometry.

The developers have since run GYRO in production mode on the X1 and observed the performance difference to be larger. For example, on transport barrier studies, 1024-processor production cases on Power3 give a relative timing of 16 seconds between output stages, whereas 32 MSPs on

the X1 for the same size case take 25 seconds per output stage. Practically, this means that more than 512 Power3 processors are required to match the performance available from 32 X1 MSPs.

Furthermore, hardware counter data has shown that GYRO has achieved 125.831 Gflops/sec with 32 MSPs. This is 3.92 Gflops/sec per processor or 30.7% of peak. Probably the most significant reason for this relatively high percent of peak realized is because an average vector length of 62 is attained.

In summary it has been shown that GYRO benefits greatly from running on the X1: GYRO has achieved approximately 35% of peak with large runs. This benefit was the product of a nontrivial, but not unreasonable, optimization effort. The result was achieved by modifying 14 out of 140 routines where most of the modifications were the insertion of directives. Consequently, the new version of GYRO is in fact slightly faster on MPPs.

Table 3: GYRO scaling.

System	Processors	Seconds per Timestep
IBM SP (Nighthawk II)	256	2.75
IBM p690	32	13.5
Cray X1	16 MSPs	4.8
Cray X1	32 MSPs	2.74
Cray X1	48 MSPs	2.05

Also, the GYRO developers find that the X1 is more flexible in terms of grid selection. Because the Power3 system offered groups of 16 processors per frame, applications were forced to use multiples of 16 processors for runs. This in turn limits toroidal modes to multiples of 16 (a parallelization constraint demands that the number of processors be a multiple of the multiple of modes). With the X1, any number of toroidal modes can be used.

4.5.2 NIMROD

NIMROD stands for Non-Ideal Magnetohydrodynamics with Rotation, an Open Discussion Project. It is designed for flexibility in geometry and physics models and for efficient computation from PCs to MPPs. NIMROD uses semi-implicit and implicit methods in combination with high-order spatial representation.

NIMROD uses distributed SuperLU. At this time, even with SuperLU not optimized for the X1, early results show a 2 MSP run approximately 5 times faster than 2 Power4 CPUs. The future requires doing more comparisons between the X1 and other architectures. If the X1 demonstrates capability beyond that of other architectures at larger processor counts, then it would be interesting to see how it does on very large problem sizes not done before in an effort to accomplish new science.

4.5.3 AORSA3D

A mature version of AORSA3D was used for acceptance testing on the X1. With the new ScaLAPACK solvers, 8 GFlops/s/proc with 240 MSPs was observed. The pre- and post-processing parts of the code need hand-tuning as they are becoming a significant portion of the runtime.

Work on porting a newer version of AORSA2D, targeting an ITER-scale run, followed by a port of AORSA3D is underway. With the improved system performance, the expectation is to accomplish new science.

4.6 Materials Science

4.6.1 DCA-QMC

An early success on the Cray X1 was the model of "cuprate" high-temperature superconductors developed by Mark Jarrell and others at the University of Cincinnati and Thomas Maier and Thomas Schulthess at ORNL. The model, which uses Jarrell's Dynamic Cluster Approximation (DCA), performs quantum-Monte-Carlo (QMC) computations dominated by double-precision rank-one matrix updates, "DGER," and complex single-precision matrix multiplication, "CGEMM." The X1 has several advantages over competing, cache-dependent systems in running the DCA-QMC model:

DGER performs few computations per memory operation and provides little opportunity for cache blocking, so the large memory bandwidth of the X1 greatly accelerates this operation. CGEMM can be blocked for cache effectively and runs well on cache-dependent systems. Each QMC process has a fixed startup cost. Using fewer, more-powerful processors reduces time being spent on redundant work.

Early performance benchmarks showed that 8 X1 MSPs were more than 3 times faster than 32 p690 processors, and 32 MSPs were more than 12 times faster. Production runs began in early Fall of 2003.

The productions runs on the X1 typically use two-dimensional cluster sizes of 32 and 64. Before the availability of the X1, cluster sizes of only 4 or 8 were possible. With these smaller cluster sizes, the model was not producing the correct two-dimensional physics. The larger cluster sizes enabled by the X1 have restored the correct two-dimensional physics.

Current runs often use 50-90% of the 256 MSPs on the X1. Expansion of the X1 will enable the modeling of multiple 2D clusters arranged in 3D structures that may be the first to computationally reproduce high-temperature superconductivity. Future ultrascale systems may then enable solution of the inverse problem, designing new materials that provide desired characteristics.

4.6.2 LSMS

LSMS was ported to the Cray as an SSP application. The design of LSMS maps each processor to an atom, so by using SSPs, more atoms can be modeled in a run. For production-size runs, LSMS scales nearly linearly. For example, an 880 SSP run is only 3% slower than a 32-SSP run where the amount of work per SSP is same for both.

These production-like runs attain approximately 53% of peak with only some compiler optimizations. No hand-tuning has been implemented. Future work will involve putting a newer version of LSMS on the X1. Also, since BLAS subroutines only account for 65% of the profile, there are LSMS routines that can benefit from hand-tuning.

4.7 Chemistry

4.7.1 NWCHEM and ARMCI

The NWCHEM chemistry package is built on Global Arrays (GA), which, in turn, is built on ARMCI. Good performance of NWCHEM depends on the development of a high performance implementation of ARMCI, a portable remote memory copy library http://www.emsl.pnl.gov/docs/parsoft/armci on the Cray X1. The port uses global shared memory of the X1 directly and delivers performance competitive to other communication interfaces available on the X1. As Figure 9 shows, the current implementation of ARMCI_Get consistently performs at higher bandwidth than MPI send/receive.

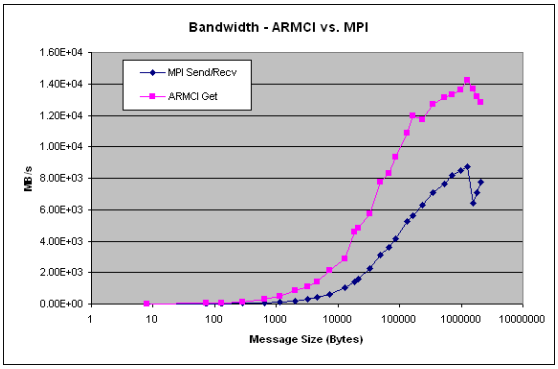

Figure 9: Bandwidth comparison between ARMCI_Get and MPI_Send/Recv.

4.7.2 GAMESS

General Atomic & Molecular Electronic Structure System (GAMESS) is a computational chemistry code used for simulating chemical reactions and calculating molecular properties. The X1 did not initially perform well on this application because GAMESS is almost entirely scalar code (deeply-nested loops full of if-tests being one of the prime culprits). Extensive work went into improving performance of this code for the X1, resulting in speedups of 3x to 8x, depending on the job. Direct-SCF RHF calculations were most affected by these changes, and further optimization work is ongoing. The first round of optimizations, submitted in mid-2003, have been accepted by Iowa State University, and are currently available with the official distribution of GAMESS. Further optimizations are pending testing and acceptance.

Early analysis showed that MSP mode for GAMESS on the X1 was approximately 3x slower than SSP mode. This is because, as for vectorization, the complicated loop nesting within GAMESS prevents the compiler from streaming all but the innermost loops (in most cases). In addition, GAMESS has a built-in dynamic load-balancing scheme which makes it highly parallel-efficient. Now that extensive rewrites have been done to certain key routines, it may be possible to implement an MSP version of GAMESS.

One of the greatest advantages to running GAMESS on the X1 is the large amount of memory available. With 16GB of memory available per X1 node, GAMESS jobs requiring enormous memory can be run on the X1 when memory limitations prevent them from being run elsewhere.

4.8 Astrophysics

4.8.1 AGILE-BOLTZTRAN

One of the primary codes used by the Terascale Supernova Initiative (TSI) has been ported to the Cray X1 at the ORNL CCS. AGILE-BOLTZTRAN is a neutrino radiation hydrodynamics code, specifically designed to model core-collapse supernovae in spherical symmetry. The code makes use of a one-dimensional, adaptive-mesh, implicit hydrodynamics module tightly coupled to a Boltztran solver for the neutrino radiation field based on the method of discrete ordinates extended to handle both general and special relativistic effects in hydrodynamic flows and the fermionic nature of neutrinos. Each computational cycle generally consists of a hydrodynamics solve operator (direct solve of a relatively small linear system) split from a transport step (the construction and solution of a large, unsymmetrical, block tridiagonal system) and a recalculation of several microphysical quantities needed by the transport solve that are functions of various hydrodynamic variables (floating point intensive filling of several large, multidimensional data structures).

Current performance evaluation of AGILE-BOLTZTRAN will concentrate on measuring the performance differences between the Cray X1 and the IBM p690 for the transport step. The construction and solution of the large, sparse linear system at the heart of the transport solution is a computational kernel that will carry over to future simulations in two and three spatial dimensions essentially unchanged. Conversely, the hydrodynamics algorithm will be wholly different in multiple spatial dimensions, and memory and CPU limitations will necessitate a replacement of current inline interaction physics calculations with a set of pre-computed multidimensional tables. The current linear system solver is a specially tailored ADI-like iterative solver due to D'Azevedo et al. The differences in the structure of the linear system resulting from a move to multiple spatial dimensions will be such that this solver, or a closely related derivative, will continue to be the primary solution implementation.

The important metric describing the size of the transport calculation is the product of the number of angular bins used to resolve the neutrino direction cosines and the number of energy groups used to resolve the spectra of those neutrinos. Typical production runs of the serial version of AGILE-BOLTZTRAN have used phase space resolutions of 6 angular quadrature bins and 12 energy groups. These calculations have been performed on spatial grids of approximately 100 radial zones. These choices produce a transport problem of overall size roughly 14500x14500, with dense diagonal block sizes of 146x146.

Single processor performance for this typical problem size was measured on both the X1 and the p690 at the ORNL CCS. The X1 exhibited a speedup of over 5x compared to the p690, running at 1.84 GLOPs vs. 366 MFLOPs. In upcoming ray-by-ray radiation hydrodynamics

simulations, these differences should produce wallclock speedups between the two machines of essentially 5 times.

A much higher resolution problem size was also tested: 16 energy groups and 16 angular bins, distributed over 100 processors (with each MSP of the X1 as a processor) of each machine. The current use of large, temporary data structures in the interaction physics modules produces executables on each processor of just less than 2GB, requiring the use of at least 100 processors. The X1 produces a speedup for this problem of more than 15x over the p690 times, and wallclock times suggestive that production runs at this resolution are viable. These kinds of phase space resolutions are more typical of those that will be used in two and three-dimensional simulations. In general, the X1 significantly outperforms the p690 on an important computational kernel for TSI, especially for large problem sizes.

4.9 Biology: AMBER and NAMD

AMBER (Assisted Model Building with Energy Refinement) is a widely used software package for atomistic modeling of biological systems using molecular dynamics (MD) simulations, developed at University of California, San Francisco. Parallelization of MD codes is of wide interest to the biological community because with the current computational resources, MD modeling falls short of simulating biologically relevant time scale by several orders of magnitude. The ratio of desired and simulated time scale is somewhere between 100,000 – 1,000,000. In the past, developers of AMBER were successful in achieving good speed on the Cray T3E with up to 256 processors. See http://www.psc.edu/science/kollman98.html. This enabled 1 microsecond MD simulation of a small protein in water (12,000 atoms) over a period of about 3 months. Today's biological systems of interest consist of millions of atoms, which will require substantially more computing power of hundreds to thousands of processors for extended periods of time.

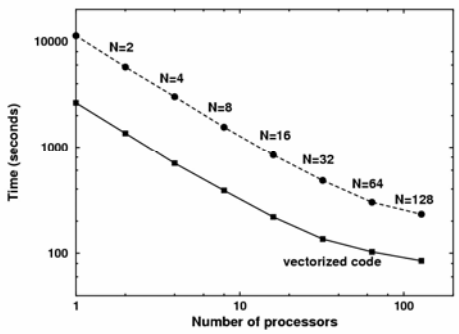

Figure 10: AMBER Time-to-solution on Cray X1.

Pratul K. Agarwal of the Computer Science and Mathematics Division (CSMD) at ORNL, in collaboration with Jim Maltby of Cray Inc., is currently working on optimizing AMBER on X1 for parallel execution on several hundreds of processors by evaluating each loop for vectorization in the commonly used Particle Mesh Ewald (PME) subroutines. Initial benchmarking results on 128 processors of the X1 show good speed up, with vectorization reducing the run time significantly (see Figure 10). Figure 11 shows that speed up continues to improve for higher numbers of processors as the number of atoms in the simulation grows. The most time consuming part of MD runs is the calculation of the non-bonded interactions, therefore, other non-bonded interactions subroutines (Generalized Born) are also being optimized for vector processors of the X1.

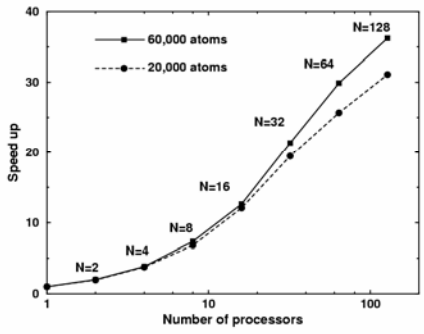

Figure 11: AMBER speedup on Cray X1.

NAMD is another popular parallel MD program developed by the Theoretical and Computational Biophysics Group at University of Illinois at Urbana-Champaign. Work is also underway to port NAMD to X1 and optimize it for longer simulation periods.

4.10 Atomic and Molecular Physics

The oldest and still one of the largest divisions of the American Physical Society is the Division of Atomic, Molecular, and Optical (AMO) Physics. In the Division, basic research studies are carried out in the areas of quantum information, atomic and molecular structure and collisional dynamics, clusters, atomic and molecular processes in external fields, and matter waves and condensation. Due to the pervasive nature of atoms and molecules, applications of these basic research studies are legion: from nanoscale fabrication to observations of the large scale structure of the universe.

Theoretical physicists at ORNL, LANL, and several U.S. and UK research universities have formed a consortium to study the many-body quantum dynamics of atoms and molecules utilizing the most advanced computing platforms. Advances in understanding are applied to many general science areas, including controlled fusion, atmospheric chemistry, cold atom condensate dynamics, laser interactions with matter, and observational astrophysics. The consortium is supported by grants from the U.S. Department of Energy, the UK Engineering and Physical Sciences Research Council, the U.S. National Science Foundation, and the U.S. National Aeronautics and Space Administration. AMO research computer codes are run on IBM-SP massively parallel machines at the DOE/NERSC, ORNL/CCS, the Daresbury/CSE, and the NSF/SDSC, as well as other types of computing machines at LANL.

Beginning in the fall of 2003, several AMO research computer codes have been tested in full production mode on

the Cray X1 at the ORNL/CCS. As an example, the TDCC3d code is run on the IBM-SP or the Cray X1 with no coding changes. The TDCC3d code partitions a three-dimensional radial lattice over the processors using standard Fortran with MPI. With a partitioning over 64 processors, the TDCC3d code runs 15 times faster on the Cray X1 than the IBM-SP at the DOE/NERSC. Only when the total memory needed to run a large coupled channel case exceeds that currently available on the Cray X1 is the code forced to run on the much slower, but larger, NERSC machine.

At present Dr. James Colgan of LANL has been running the TDCC3d code on the Cray X1 to calculate all of the dynamical processes found in the photoionization of the Li atom. His calculation of the triple photoionization of Li represents the first non-perturbative quantum solution of four-body Coulomb breakup. The total cross section results are in reasonable agreement with pioneering experimental measurements made at the LBNL Advanced Light Source.

Dr. Mitch Pindzola of Auburn University has been running the TDCC3d code on the Cray X1 to calculate all of the dynamical processes found in the electron ionization of the He atom. His calculation of the electron double ionization of He again solves four-body Coulomb breakup. Whereas radiative selection rules limit the photoionization of Li calculation to one overall symmetry, many total partial wave symmetries are needed in the electron ionization of He calculation, making the latter a truly daunting computational challenge. So far complete results have been obtained for L = 0 to L = 3 only.

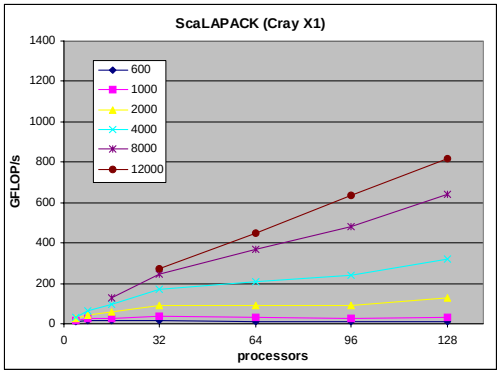

Figure 12: ScaLAPACK performance for matrix multiplication on the Cray X1.

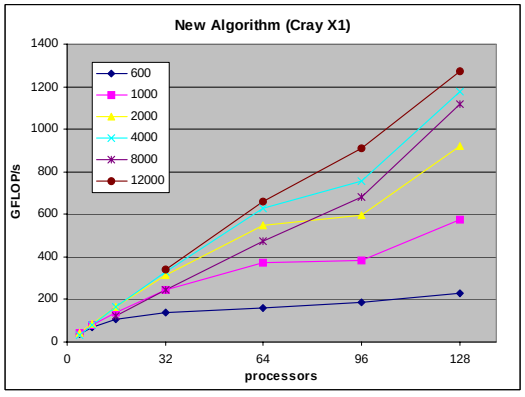

Figure 13: New algorithm for matrix multiplication on Cray X1.

4.11 Dense Matrix Operations on Cray X1

Krishnakumar and Nieplocha have developed a new algorithm [KrNi2004] on the Cray X1 that exploits Cray optimized DGEMM, performance characteristics of global shared memory on that machine, and ARMCI interface for communication. Figure 12 and Figure 13 show performance comparisons of the new algorithm to the Cray-provided PDGEMM out of PBLAS/Scalapack library by matrix size.

4.12 Sparse Matrix Operations on Cray X1

The Cray X1 was evaluated in the context of sparse linear algebraic computation. The sparse matrix operations are key parts of many important scientific applications (in particular implicit unstructured mesh codes solving nonlinear partial differential equations (PDEs)). It is well known in the literature that many of the popular storage formats for sparse matrices (e.g., compressed row storage) perform poorly on vector processors. In the past, some sparse matrix formats (like jagged diagonal, segmented scan, etc.) were proposed that fair relatively better but not close to the performance of dense matrix operations (which vectorize well and have high data reuse).

For the evaluation purpose, a simple (but very important) kernel of sparse linear algebra was employed: matrix vector multiplication. The sparse matrix is stored in diagonal format. After reasonable level of optimization, obtained about 14% of the theoretical peak was obtained on one multi-streaming processor (MSP) of Cray X1. To understand this performance, a performance analysis based on the memory hierarchy of this MSP was carried out. It turns out that the sparse matrix vector multiplication is memory bandwidth limited (even though the available memory bandwidth is relatively high as compared to most commodity processors) and the best that can be hoped for is about 20-30% of the theoretical peak (by doing extensive custom data structure redesign). It should be noted that the implicit algorithms that employ sparse matrix linear algebra to solve PDEs are optimal in terms of computational complexity.

In addition to the sparse matrix vector multiplication, there are many other important computational phases in implicit PDE codes (like finite element assembly, mesh manipulation, preconditioner generation, etc.) that will require significant effort to adapt to Cray X1. Cray does not currently provide any support for sparse linear algebra libraries, however initial ports of these libraries will be available from Cray in early 2004 and Cray is very interested in working with any groups to port and optimize sparse solver libraries.

In summary, simple mathematical models taking into account the latency and bandwidth of the system provide an upper bound of approximately 30% of peak for most sparse matrix computations. Even that, however, would require heroic programming effort since the required data structures and algorithms are not those used in conventional off-the-shelf systems (COTS) such as the Pentium, IBM Power series etc. Roughly, for certain sparse matrix calculations (like matrix vector product) as an upper bound, one could achieve about twice the ratio of achieved to peak

performance on the X1 compared to COTS systems, with a complete rewrite of the software.

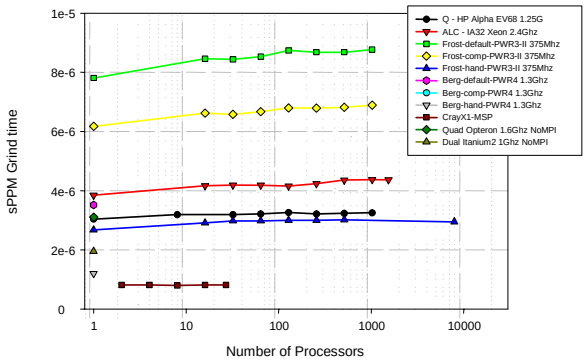

Figure 14: sPPM performance results on a scaled problem where the vertical axis represents the time to solution for one timestep of sPPM.

4.13 sPPM

sPPM solves a 3-D gas dynamics problem on a uniform Cartesian mesh, using a simplified version of the Piecewise Parabolic Method. The algorithm makes use of a split scheme of X, Y, and Z Lagrangian and remap steps, which are computed as three separate sweeps through the mesh per timestep. Message passing provides updates to ghost cells from neighboring domains three times per timestep.

After spending approximately two days porting sPPM to the X1 during a workshop at ORNL, the best performance was obtained across all the systems tested as Figure 14 illustrates.

4.14 Cosmology

From the application area of cosmology, the Microwave Anisotropy Dataset Computational Analysis Package (MADCAP) was examined. MADCAP implements the current optimal general algorithm for extracting the most useful cosmological information from total-power observations of the Cosmic Microwave Background (CMB). The CMB is a snapshot of the Universe when it first became electrically neutral some 400,000 years after the Big Bang. The tiny anisotropies in the temperature and polarization of the CMB radiation are sensitive probes of early Universe cosmology and measuring their detailed statistical properties has been a high priority in the field for over 30 years. MADCAP was designed to calculate the maximum likelihood two-point angular correlation function (or power spectrum) of the CMB given a noisy, pixelized sky map and its associated pixel-pixel noise correlation matrix. http://crd.lbl.gov/~oliker/Vecpar04/vecpar04_draft.pdf provides more details.

4.14.1 X1 Porting

Porting MADCAP to vector architectures is straightforward, since the package utilizes ScaLAPACK to perform dense linear algebra calculations. However, it was necessary to rewrite the dSdC routine that computes the pixel-pixel signal correlation matrices. The basic structure of dSdC loops over all pixels and calculates the value of Legendre polynomials up to some preset degree for the angular separation between these pixels. On superscalar architectures, this constituted a largely insignificant amount of work, but since the computation of the polynomials is recursive, prohibiting vectorization, the cost was appreciable on both the X1 and the SX-6. The dSdC routine was therefore rewritten so that at each iteration of the Legendre polynomial recursion, a large batch of angular separations, was computed in an inner loop. Compiler directives were required to ensure vectorization for both the vector machines.

Table 4: Per-processor performance of MADCAP on target machines

	Power3		Power4		SX-6		X1	
P	GFlops/s	% Peak	GFlops/s	% Peak	GFlops/s	% Peak	GFlops/s	% Peak
Small synthetic case: N_p = 8192, N_b = *16*, without gang-parallelism								
1	0.844	56.3	1.98	38.1	4.55	56.9	5.36	41.9
4	0.656	43.7	1.23	23.7	2.66	33.2	4.26	33.3
9	0.701	46.7			—	—	3.22	25.2
16	0.520	34.7	0.731	14.1	—	—	2.48	19.4
25	0.552	36.8			—	—	1.84	14.4
Large real case: N_p = 14966, N_b = 16, with gang-parallelism								
4	0.831	55.4	1.90	36.5	4.24	53.0	5.66	44.2
8	0.688	45.9	1.62	31.2	3.24	40.5		
16	0.615	41.0	1.26	24.2	—	—		
32	0.582	38.8	0.804	15.5	—	—		
64	0.495	33.0	0.524	10.1	—	—		

4.14.2 Performance Results

Table 4 first shows the performance of MADCAP for a small synthetic dataset with N_p = 8192 and N_b = 16 on each of the four architectures under investigation, without the use of gang-parallelism. Note that for maximum efficiency, the processor sizes are set to squares of integers. As expected, the single processor runs achieve a significant fraction of the peak performance since the analysis is dominated by BLAS3 operations (at least 60%). However, as more processors with a fixed data size are used, the density of the data on each processor decreases, the communication overhead increases, and performance drops significantly. Observe that the X1 attains the highest raw performance, achieving factors of 6.5x, 3.5x, and 1.6x speedup compared with the Power3, Power4, and SX-6 respectively on 4 processors. With increasing processor counts, the fraction of time spent in BLAS3 computations decreases, causing performance degradation. The average vector length was about 62 per SSP (vector length 64) for all processor counts. It is therefore surprising that MADCAP did not attain a higher percentage of peak on the X1.

Table 4 also shows the performance of the gang-parallel implementation of MADCAP for a larger real dataset from the Millimeter-wave Anisotropy Experiment Imaging Array (MAXIMA), with Np = 14996 and Nb = 16. MAXIMA is a balloon-borne experiment with an array of 16 bolometric photometers optimized to map the CMB anisotropy over hundreds of square degrees. The gang parallel technique is designed to preserve the density of data during matrix-matrix multiplication and produce close to linear speedup. In this regard, the results on the Power3 and Power4 are somewhat disappointing. This is because the other steps in MADCAP do not scale that well, and the efficiency of each gang suffers due to increased memory contention.

4.15 I/O and System Software Evaluation

4.15.1 Stability

The X1 evaluation has led to a variety of system stability and scalability issues and improvements. The following is a list of some significant system challenges and achievement over the evaluation period thus far.

The upgrade to more than one cabinet suffered from a number of bent pins on the connectors for the interconnect cables. The bent pins were difficult to detect and caused unusual failures. This experience led to a change in the Cray installation procedure that uses boroscopes to inspect connectors. This has eliminated the problem.

As applications scaled beyond 64 MSPs, high variability and latency in barriers and other similar collective-communication operations were noticed. The problem was solved by synchronizing clock interrupt handling across the system, which keeps synchronizing processes from waiting in user mode for other processes to perform system tasks. Now, the less onerous problem of jobs issuing interrupts and slowing down other jobs remains. Also, Cray engineers are investigating localizing interrupts to avoid this issue.

When the 256-MSP system was reconfigured from four fully-populated cabinets to eight half-populated cabinets, the topology of the X1 interconnect changed. This change led to another increase in the variability and latency of barrier operations. The solution this time was in the barrier software itself. Cray hired a communication expert who is now optimizing the Cray MPI for greater scalability. Early results include a barrier routine that is slightly faster for small process counts but much more scalable. The new barrier time increases by less than 50% going from 32 to 880 processes (using SSPs), while the previous barrier increased by a factor of eight.

One of the high-priority applications, the Community Climate System Model (CCSM), performs simulations of the global climate by coupling components that separately model the atmosphere, oceans, land, and sea ice. Each of these components, along with the coupler itself, uses a separate executable. Cray did not originally support multiple executables in a single parallel job, but the need for this capability caused them to accelerate development of it. CCSM now has this capability, and its input has led Cray to consider the additional capability of allowing each executable to specify a different number of threads per process. This additional capability is unlikely to be available on X1, but Cray is considering modifying future architectures to support it. The critical capability is for multiple executables; differing thread counts is a tuning option. It is expected that other coupled simulations, such as those proposed for fusion, biology, and astrophysics, may also use multiple executables in a single job.

Compilation time is an ongoing issue with the X1, but some progress has been seen and more is expected. Cray now provides a cross compiler for use on Sun systems and prototyped using the Cray Programming-Environment Server (CPES) for both CPES services and direct login and cross compilation. The tests were successful, and now production is run this way. Compilation directly on the CPES can be up to twice as fast as from the X1. Further significant improvements are expected as the CPES moves from Sun to Intel architectures and as the CPES gets direct access to X1 files through the ADIC StorNext file system. The CPES currently accesses files through NFS from the X1.

Continuing hardware issues on the X1 include multi-bit memory errors and processor parity errors, many of which result in full-system panics. Cray engineers have significantly reduced the frequency of these failures, but they have not been eliminated. The CCS continues to work with Cray on strategies to reduce the effects of these failures. The CCS and Cray have worked together to identify specific applications that demonstrate the problem and help identify failing parts.

4.15.2 Collaborative Activities with Cray

This section describes four collaborations that CCS has undertaken with Cray to contribute to the success of the X1.

ADIC SAN planning

The Cray system currently supports only XFS local file systems. In order to scale, it is expect that a SAN file system with Fibre Channel (FC) attachment of the disks is needed.

Cray has developed a relationship with ADIC to support their StorNext file system on the Cray X1.

CCS has been heavily involved with Cray's plan to roll out this support. It loaned Cray a Brocade Silkworm 3900 switch to test for compatibility with the CCS's environment. The two have jointly developed plans to start extensive testing of StorNext functionality and performance in the 1st Quarter of 2004.

The plan is to test both Cray-branded and non-Cray-branded FC disk arrays with clients on the X1, the Cray Programming Environment Server (CPES), a Solaris based system, SGI's Altix system and IBM SP systems at the CCS. There will be good characterizations of the performance that the StorNext product can produce in both a Cray environment as well as for the whole HPC complex when it is complete.

Cray Network Server (CNS) optimization

In order to maximize performance by minimizing the interrupt load on the X1, all I/O devices that attach directly to the X1 are buffered using 64K byte blocks. Network attachment is achieved by interposing the Cray Network Server (CNS), a Dell 2650 Linux server, whose primary purpose is to take network traffic that arrives in either 1500 byte or 9000 byte buffers and reblock it into 64K byte blocks to be sent over an IP over FC connection to the X1. A second purpose is to notice TCP traffic and run a so-called "TCP_assist" function to speed up TCP.

In 2003, CCS has developed an open source replacement for the binary-only version of TCP_assist supplied by Cray (which gets it from a sub-contractor in binary-only form). A copy of that replacement code was delivered to Cray under a GNU Public License (GPL) for their use. It is the CCS's understanding that Cray plans to use its code for future releases of the CNS.

In 2004, the CCS is planning to install the Net100 modifications to the TCP stack into the production CNS to test its applicability to traffic in and out of the X1. This work has already been accomplished in test mode on a second CNS. The Net100 modifications attempt to tune a TCP/IP connection to maximize bandwidth over wide area network connections. Testing at ORNL shows that these modifications are also useful in increasing the bandwidth for all connections.

Since IP over FC is the only way to connect peripherals to the X1, a range of tests are planned that will involve putting services into the CNS so that a much better performance of these services can potentially be achieved. Two examples come to mind. First is the deployment of an HPSS mover on a Linux platform that talks to the Cray with IP over FC. If this test proves fruitful, users will see dramatically increased network performance to HPSS. A second experiment is to put a Spinnaker Network's NFS server into a CNS thereby connecting it to the X1 with IP over FC. The Spinnaker technology is being deployed as the replacement for DFS home directories in the HPC complex.

4.15.3 Access control

The HPC complex currently keeps all user account information in a database that is not local to any of the supercomputers. The current technology employed is a combination of the Distributed Computing Environment's (DCE) registry and the Lightweight Directory Application Protocol (LDAP). Unicos/MP does not support either of these two authorization and authentication services. The CCS has been working with Cray to define the requirement and develop an implementation plan. Support for LDAP on the X1 is expected in the future.

4.15.4 Developing the next OS for Black Widow

Cray and the CCS have been developing plans for CCS to test the Black Widow OS in the 2004 time frame. A small part of the CCS's existing X1 is planned for these tests. The CCS will be the first site to deploy the new operating system outside of Cray's software development center. The advantages of this early testing to both Cray and the CCS are enormous.

4.16 Scalability Evaluation

The initial X1 installation at ORNL had 32 MSP processors. Over the course of the summer, this was increased to 64, then 128, and finally to 256 processors. The OS was also upgraded a number of times, with the last major upgrade at the end of September having a significant impact on the performance of large processor count runs.

Many of the micro-benchmark, kernel, and application codes described above have been used to evaluate the performance scalability as the system has grown and the application and system software evolved. Communication micro-benchmarks (e.g., MPI point-to-point, all-reduce, and all-to-all) were used to determine the network subsystem scalability. Two application codes in particular were used for scalability analyses. POP was used because of its sensitivity to small message performance. Lack of performance scalability of POP in August led directly to the September OS update. POP performance still does not scale well beyond 128 processors for the small benchmark problem. While performance will eventually stall for any fixed size problem, the current scalability issue has been identified as an OS performance problem and solutions are currently being investigated within Cray.

The GYRO fusion code was also used because of its sensitivity to large message performance. Two fixed size problems were supplied by the developer, small and very large. Unlike POP, where system interrupts and other performance vagaries are limiting performance on large processor counts, GYRO scalability is good for the small problem, excellent for the large, and smooth for both. For the GYRO results, the vector modifications were only incorporated into the released version of the model and the benchmark problems specified in late October. However, the high bandwidth requirements are being easily handled by the X1 network, and performance appears to be primarily a function of changing vector length and the slower decrease in communication cost as compared to computation cost as more processors are used.

Tracking POP scalability will continue as the implementation and system software change. The performance of GYRO will also be analyzed. Additional

application codes will be added as they become available, and they will be added to the scalability test suite.

5 Outreach Activities

This evaluation plan was the outcome of a number of meetings with both HPC system vendors and application experts over the past 9 months. Highlights from several of these meetings are given below.

5.1.1 Climate Modeling System Workshop

The Climate Modeling System Workshop was held at IBM Research in Yorktown Heights, June 10-12, 2002, and included representatives from IBM, ORNL, and NCAR. The topics of discussion were the computational requirements of the next generation of climate model, along with what hardware and system configurations and performance would be sufficient to meet those needs. Over 25 researchers attended this meeting.

5.1.2 ORNL Cray X1 Tutorial

The ORNL Cray X1 Tutorial was held on November 7, 2002 at ORNL (http://www.ccs.ornl.gov/CrayX1/Tutorial/index.html). ORNL staff described the acquisition and evaluation plans, and Cray staff made presentations on the X1 hardware, tools, performance, and optimization strategies. Approximately 101 researchers attended this tutorial. On November 7, 2002, ORNL hosted a tutorial for an initial introduction to Cray X1 for scientists across the DOE. Attendees included representatives from Ames Lab, ANL, BNL, Colorado State, LANL, LBNL, NCI, Office of Science, Old Dominion, ORNL, PNNL, PPPL, SLAC, SNL, SUNY, UCSD, UIUC, University of Maryland, and UT.

5.1.3 Cray-Fusion UltraScale Computing Workshop

The Cray-Fusion UltraScale Computing Workshop was held at the Garden Plaza Hotel in Oak Ridge, Tennessee, February 3-5, 2003 (http://www.csm.ornl.gov/meetings/fusion2.html). Topics of discussion were the characterization of models/codes in terms of physics objectives and models; characterization of performance extensions needed for making significant advances; identification of code/code kernel/algorithm for analysis on a range of platforms; and development plans for carrying out this task. CCS and Cray staff worked with application specialists to identify applications for the initial Cray evaluation. Attendees included application specialists from CCS and Cray, scientists from ORNL and PPPL, teleconference with scientists from General Atomics and University of Wisconsin. The group identified several applications for initial Cray evaluation: Aorsa, GTC, Gyro, M3D, Nimrod, and Toric.

5.1.4 Doe SciDAC Workshop: Porting CCSM to the Cray X1

The DOE SciDAC Workshop: Porting CCSM to the Cray X1, was held at the National Center for Atmospheric Research in Boulder, Colorado, February 6, 2003 (http://www.csm.ornl.gov/meetings/climate2.html). The goal of the workshop was to identify and coordinate efforts in the porting of the Community Coupled System Model (CCSM) to the Cray X1, including discussions of vectorization and software engineering issues. Attendees included scientists from LANL, NCAR, and ORNL and application specialists from CCS, Cray, and NEC.

5.1.5 Computational Materials Science: Early Evaluation of the Cray X1

The Computational Materials Science: Early Evaluation of the Cray X1 was held at the Hyatt Regency in Austin, Texas, March 2, 2003 (http://www.csm.ornl.gov/meetings/materials2.html). The goals of the workshop were to follow up on ultra-simulation initiative white papers; provide a prioritized list of application codes that will be ported to the Cray X1; and provide a list of names and research projects affiliated with these codes. Materials researchers met with CCS and Cray staff to identify applications for the initial Cray evaluation. CCS staff presented a Cray X1 introduction and early DCA-QMC results. Attendees included application specialists from CCS and Cray and scientists from Ames Lab, BNL, ETH Zurich, Florida State, LBNL, LLNL, Mississippi State, NCSU, ORNL, SNL, University of Cincinnati, University of Florida, University of Georgia, University of Minnesota, University of Washington, and William and Mary. Applications identified for initial Cray evaluation included DCA-QMC, FLAPW, LSMS, Paratec, PWSCF, and Socorro.

5.1.6 Prototype Application Workshops

During the Prototype Application Workshop, (ORNL, November 5-6, 2002) Cray staff presented an introductory tutorial and then participated in application sessions with ORNL scientists to discuss priority applications: fusion, astrophysics, materials, chemistry, biology, and climate. Scientists from PNNL, the National Cancer Institute (NCI), and the University of Tennessee (UT) also participated.

5.1.7 Biology Workshop

A computational biology workshop was held at ORNL on May 9, 2003. CCS staff presented a Cray X1 tutorial; CCS and Cray staff worked with biology researchers to identify specific application areas for the Cray evaluation. These areas included molecular dynamics, genome comparison, and biophysics. Two applications were identified for possible Cray evaluation: Amber and Prospect.

5.1.8 Cray Hands-On Workshop I

A Cray hands-on workshop was held at Oak Ridge on June 3-5, 2003. Users were given hands-on access to the CCS Cray X1 with support from CCS and Cray specialists. Several optimization tutorials were presented. Leonid Oliker and Jonathan Carter, LBNL, ported TLBE and Madcap and began work on UPC tests. Andrew Canning, LBNL, started the Paratec port and identified issues with ScaLAPACK and FFTs. Jarek Nieplocha, PNNL, made significant progress on optimization techniques for ARMCI. Satish Balay and Dinesh Kaushik, ANL, identified extensive changes to

PETSc data structures needed for vectorization.

5.1.9 Cray Hands-On Workshop II

A second Cray hands-on workshop was held at Oak Ridge on July 23-25, 2003. Users were given hands-on access to the CCS Cray X1 with support from CCS and Cray specialists. Several optimization tutorials were presented. Tony Craig, Julie Schramm, and Vince Wayland, NCAR, began porting CCSM components to the X1. Stephane Ethier, PPPL, continued work on GTC vectorization. Hongzhang Shan, LBNL, ported and ran memory-performance benchmarks. Jeff Vetter, then of LLNL, ported several applications including sPPM and SMG2000.

5.1.10 CCSM Vectorization Workshop (organized by NCAR)

On October 15, 2003 at NCAR, NCAR, and ORNL scientists met with application experts from CCS, NEC, and CRIEPI to plan for porting, tuning, and validation of all CCSM components for vector architectures, including Cray X1 and Earth Simulator.

6 Additional Publications

- A.S. Bland, R. Alexander, S.M. Carter, and K.D. Matney, "Early Operations Experience with the Cray X1 at the Oak Ridge National Laboratory Center for Computational Sciences," Proc. 45th Cray User Group Conference, 2003.
- A.S. Bland, J.J. Dongarra, J.B. Drake, T.H. Dunigan, Jr., T.H. Dunning, Jr., A. Geist, B. Gorda, W.D. Gropp, R.J. Harrison, R. Kendall, D. Keyes, J.A. Nichols, L. Oliker, H. Simon, R. Stevens, J.B. White, III, P.H. Worley, and T. Zacharia, "Cray X1 Evaluation," Oak Ridge National Laboratory, Oak Ridge, TN, Report ORNL-TM-2003/67, 2003.
- T.H. Dunigan, Jr., M.R. Fahey, J.B. White, III, and P.H. Worley, "Early Evaluation of the Cray X1," Proc. ACM/IEEE Conference on High Performance Networking and Computing, SC03, 2003.
- M.R. Fahey and J.B. White, III, "DOE Ultrascale Evaluation Plan of the Cray X1," Proc. 45th Cray User Group Conference, 2003.
- J.B. White, III, "An Optimization Experiment with the Community Land Model on the Cray X1," Proc. 45th Cray User Group Conference, 2003.
- P.H. Worley and T.H. Dunigan, Jr., "Early Evaluation of the Cray X1 at Oak Ridge National Laboratory," Proc. 45th Cray User Group Conference, 2003.
- M. Krishnakumar and J Nieplocha, "A parallel matrix multiplication algorithm suitable for scalable shared and distributed memory systems," Proc IPDPS'04.
- L. Oliker, R. Biswas, J. Borrill, A. Canning, J. Carter, M. J. Djomehri, H. Shan, D. Skinner, "A Performance Evaluation of the Cray X1 for Scientific Applications," Proc. VECPAR 04: 6th Int'l Meeting High Performance Computing for Computational Science (submitted).

7 References

1. OpenMP Architecture Review Board, "OpenMP: A Proposed Standard API for Shared Memory Programming," October 1997. Available from http://www.openmp.org/openmp/mp-documents/paper/paper.ps.
2. Z. Lin et at., *Phys Rev. Lett.* **88**, 195004, 2002.