

Evaluation Plan for the Cray X1
Oak Ridge National Laboratory
in Collaboration with the SciDAC Scientific Teams and ISICs

“Japanese Computer Is World’s Fastest, as U.S. Falls Back” was the title of the *New York Times* article that heralded the arrival of the “Earth Simulator” to the Top500 list, where it is likely to stay at the Number One position for quite some time. This document outlines the plan for evaluating a computer system at the Oak Ridge National Laboratory (ORNL) that has the potential to match or exceed the performance of the Earth Simulator for many Office of Science research applications - the Cray X1, and, through close collaboration with Cray Inc., increase the realized fraction of peak performance for this and future generations of such computers on scientific and engineering applications. These efforts will eliminate the science gap created by the Earth Simulator and greatly enhance high-end computing in the U.S.

The Earth Simulator achieves its performance advantage not only by a significant financial investment, but also by employing relatively few (5,120) powerful (8 GFLOP/s) vector processors connected to high-bandwidth memory and coupled by a high-performance network. These characteristics have enabled a much greater fraction of peak performance to be obtained on real scientific and engineering applications than has proven feasible on the commodity shared memory processors (SMP) clusters that comprise the mainstream of supercomputing in the U.S.

The X1 is the first of Cray’s new scalable vector systems that offers high-speed custom vector processors, high memory bandwidth, and an exceptionally high-bandwidth, low-latency interconnect linking the nodes. Cray is the only U.S. manufacturer of multi-terascale systems with these characteristics. The processor performance of the Cray X1 is anticipated to be comparable to the NEC SX-6 processor of the Earth Simulator on many Office of Science research applications. A significant feature of this system is that it combines the processor performance of traditional vector systems with the scalability of modern microprocessor-based architectures. The X1 is the first vector supercomputer designed to scale to thousands of processors with a single system image.

As the latest entry in the competitive, high-end computing marketplace, the Cray X1 must be evaluated for both processor and system performance and for production computing readiness. Our evaluation approach focuses on scientific applications and the computational needs of the science and engineering research communities. Among the goals of this project are establishing and maintaining a public record of benchmarks relevant to DOE missions, and establishing an open discussion in the HPC community aimed at improving supercomputer hardware and software for scientific and engineering applications.

Cray’s path into the future scales to nearly 1 PFLOP/s in the latter half of this decade, while significantly improving the price-performance of the technology used. Cray has eagerly solicited our participation in a long-term collaboration focused on making their product even better suited for the Department of Energy (DOE) applications, and to help prioritize their activities and investments so as to meet our performance expectation and delivery schedule. This partnership is an exciting opportunity for science to be once more driving high-performance computing technology, rather than merely consuming commodity products and accepting commodity performance.

The primary tasks of this 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,

Evaluation Plan for Cray X1

- 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 model fully the interactions of computation, communication, I/O and other system services, load balancing, job scheduling, networking, resource allocation, compilers, math libraries, other system software along with application software and Tbytes of data in memory and disk. This is recognized by the “*Applications Performance Matrix*,” which focuses upon the actual performance of full applications rather than a few simple kernels. The “Matrix” presently contains very little climate modeling data and no data on the Cray X1. In order to improve the performance of both applications and the machine, 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 Cray X1 along with detailed interpretation of the raw information, thereby providing a firm basis for future decisions in high performance scientific computing.

A hierarchical approach will be employed to the evaluation, examining low-level functionality of the system and using these results to guide and understand the evaluation using kernels and compact or full application codes. This approach is especially important since the Cray X1 embodies a number of novel architectural features that make it difficult to predict the performance of the most efficient coding styles and programming paradigms. Standard benchmarks will be used when appropriate and to compare with evaluations of other systems, but the major emphasis will be on application-relevant studies, which are described in more detail below.

A distinguishing feature of this effort is that the low-level benchmarks, for example, message passing, and the kernel benchmarks will be chosen so as to model important features of a full application. As will be described, this is crucial in order to predict scalability. The end goal is to predict the expected sustained performance of multi-terascale Cray systems on large-scale simulations, using actual applications codes from climate, fusion, materials science, and chemistry. Specific sub-goals are described in more detail below.

Cray X1 Description

The Cray X1 is an attempt to incorporate the best aspects of previous Cray vector and massively parallel processing (MPP) systems into one design. Like the Cray T-90, the X1 will have high memory bandwidth and high realizable percentage of theoretical peak performance. Like the Cray T3E, the design has a high-bandwidth, low-latency, scalable interconnect, and scalable system software. And, like the Cray SV1, the X1 design leverages commodity CMOS technology and incorporates non-traditional vector concepts, like vector caches and multi-streaming processors.

The Cray X1 multi-streaming processor (MSP) is capable of 12.8 GFLOP/s and has multiple vector pipes designed to increase vector throughput without increasing the inner-loop vector length. Unlike the SX-6, the X1 keeps the relatively short vector length of 64 elements, thus making it easier for algorithms to efficiently use the vector pipes. The primary strategy for fully utilizing the eight vector pipes of a single MSP is parallelism through outer loops and pipelined operations.

Evaluation Plan for Cray X1

Each MSP has 2 MB of cache memory, and the cache has more than enough single-stride bandwidth to saturate the vector units of the MSP. The cache 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-cache systems, like the 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 expected to be just a factor of two slower than stride-one access, not the factor of eight or more seen with typical cache-based systems like the IBM p690, HP Alpha, or Intel IA-64.

It is important to note that the Cray X1's cache-based design deviates from the full-bandwidth design model only slightly. Each X1 processor is designed to have more single-stride bandwidth than an SX-6 processor; it is the yet-higher peak performance that creates the imbalance. A relatively small amount of data reuse, which most modern scientific applications do exhibit, could enable a very high percentage of peak performance to be realized, and worst-case data access could still provide double-digit efficiencies.

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 GB/s between nodes. This is a remarkable number, for it 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. Comparisons with other existing systems are even more unbalanced.

The collected nodes of an X1, eventually containing up to 4096 processors, will have a single system image. A single four-processor X1 node behaves like a traditional SMP, but each processor has the additional capability of directly addressing memory on any other node. 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 OpenMP, outside of a given four-processor node. This remote memory access mechanism is an excellent match for distributed-memory programming models, particularly those using one-sided put/get operations, and it is expected to provide very low latencies and unprecedented inter-processor bandwidths.

In large configurations, the Cray X1 nodes are connected in an enhanced 3D torus, or wraparound mesh. Despite the remarkable bandwidth expected per connection, this topology has relatively low bisection bandwidth compared to crossbar-style interconnects, such as those on the NEC SX-6 and IBM p690. 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, $O(n^{2/3})$, for a 3D torus. Mesh-based systems such as the Intel Paragon, the Cray T3E, and ASCI Red have scaled to thousands of processors.

The primary benefit of full-bisection interconnects is scheduling flexibility. A single job on a mesh-based topology may achieve better performance if scheduled on contiguous nodes, while a job on a full-bisection network can be scheduled on any available nodes. Jobs on mesh-based systems may also be unable to start despite the availability of enough processors because of fragmentation of those processors. This problem was ameliorated on the Cray T3E through automated job migration; jobs were moved to create larger groups of contiguous processors. Additional benefits of implementing such job migration include system-initiated gang scheduling and checkpoint-restart. These capabilities will also be available in the Cray X1. The main reason to prefer contiguous allocation of processors on the X1 is that it enables improved hardware support for remote address translation.

Because of the tightly coupled parallelism of vector processors, high memory bandwidth, high-bandwidth/low-latency interconnect, and scalable systems software, the Cray X1 has the potential to provide more capability for scientific computation in the near term than any other system available or in

Evaluation Plan for Cray X1

late-stage development, including the SX-6. Because of the short vector pipes, multiple vector units, memory caching, and distributed memory of the X1 design, the software optimizations needed for efficiency on the X1 are expected to be similar to the optimizations needed for clusters of traditional SMPs. For example, modern high-frequency commodity processors, like the IBM Power4, have multiple floating-point units with deep pipelines and aggressive memory prefetching. They require moderately large inner loops with no dependencies, just like X1 processors. The expected raw performance and efficiency of the X1 are, however, much higher.

Microbenchmarks

The objective of the microbenchmarks is to characterize the performance of the underlying architectural components of the Cray X1. Both standard benchmarks and customized benchmarks will be used. The standard benchmarks allow component performance to be compared with other computer architectures. The custom benchmarks will permit the unique architectural features of the Cray X1 (distributed vector pipes, and cache and global memory) to be tested with respect to the target applications. The standard benchmarks used for this evaluation will include the LinPack, ATLAS, NAS parallel, Euroben, ParkBench, Streams, HINT, and LMBENCH benchmarks. Since this is the first significant U.S. vector computer in several years, older vector processor benchmarks such as the Livermore Loops will be revisited. In addition, ORNL has custom benchmarks from previous evaluations that measure the performance of message passing, memory hierarchy, OpenMP, and threads.

The architectural-component evaluation will assess the following:

- Scalar and vector arithmetic performance, including varying vector length and stride and identifying what limits peak computational performance;
- Memory hierarchy performance, including cache, local memory, shared memory, and global memory; These tests will utilize both System V shared memory and the SHMEM primitives, as well as UPC and Fortran co-arrays. Of particular interest is the performance of the shared memory and global memory, and how remote accesses interact with local accesses;
- Task and thread performance, including performance of thread creation, locks, semaphores, and barriers; Of particular interest is how explicit thread management compares with the implicit control provided by OpenMP, and how thread scheduling and memory/cache management (affinity) perform;
- Message-passing performance, including intra-node and inter-node MPI performance for one-way (ping-pong) messages, message exchanges, and aggregate operations (broadcast, all-to-all, reductions, barriers); message-passing hotspots and the effect of message passing on the memory subsystem are of particular interest.

System and I/O performance, including a set of tests to measure OS overhead (context switches), virtual memory management, low-level I/O (serial and parallel), and network (TCP/IP) performance; System calls made by an X1 application node are implemented by rescheduling the thread for execution on an O/S node. The performance of this mechanism may have significant implication for the design of I/O or network intensive applications, as suggested already by the design of MPI I/O for the X1, which has its I/O thread locked on an O/S node.

The microbenchmarking effort will also utilize and evaluate the Cray hardware performance monitors, clocks/timers, and profiling tools. Tuning of the computational benchmarks will also provide metrics on various compiler options.

The deliverables for this aspect of the evaluation are the following:

- Results of a standard set of benchmarks and comparisons with other computer architectures;

Evaluation Plan for Cray X1

- Performance metrics of the underlying architectural components, including arithmetic operations (scalar and vector), memory hierarchy (cache, local memory, global memory), message passing, task/thread management, and I/O primitives; and
- Simple analytical models of component performance.

Programming Paradigm Evaluation

As mentioned earlier, the Cray X1 has a number of unique architectural features. Unlike traditional vector machines, the vector pipes are “distributed” within the X1 MSP, and are linked by a cache. As such, the compiler must be able to identify microtasking parallelism (“streaming”) as well as vectorization, and to exploit memory access locality in order to minimize cache misses and maximize bandwidth. Early indications are that coding styles that perform well on the NEC vector systems do not necessarily perform well on the X1. The programming paradigm evaluation will include an evaluation of the Cray X1 compiler, and ORNL will work closely with Cray to identify and resolve problems.

Coding style is anticipated to be as important as the compiler in achieving good single processor performance. Kernels extracted from or representative of the target application codes will be used to examine the performance impact of a variety of coding styles. For example, index and loop orderings and other aspects of the memory access patterns will be carefully examined.

The Cray X1 is a collection of 4-processor, shared memory nodes with a network that supports globally addressable memory. How best to program for the hierarchical parallelism represented by clusters of SMP nodes is still an area of open research in parallel computing. The addition of both streaming and vectorization makes it yet more difficult to determine which of the many shared- and distributed-memory programming paradigms are most appropriate for a given application. Cray is currently planning to support both MPI and OpenMP programming paradigms, as well as the SHMEM one-sided communication library and the Co-Array Fortran and UPC parallel programming languages. System V shared-memory, Multi-Level Parallelism (MLP) and the portable Global Arrays are also possible programming models for the Cray X1.

Thus, the Cray system has multiple options for both inter-node and intra-node communication. Inter-node communication can be through MPI-1 two-sided, point-to-point primitives; MPI-1 collective communication operators; MPI-2 one-sided messaging; or SHMEM one-sided messaging. The same paradigms can be used for communication within a node, or each can be combined with OpenMP, POSIX threads, or System V shared memory in a hierarchical, hybrid approach. Alternatively, the exact mechanism used for inter-processor communication can be hidden behind the parallel syntax of Co-Array Fortran, UPC or Global Arrays.

An initial task in evaluating the system will be assessing the functionality and measuring the costs associated with each programming approach. Each approach will be optimized separately, for example, evaluating the many different communication protocols supported by MPI-1 primitives and comparing point-to-point implementations of the collective communication operations with the vendor-supplied implementations. For each approach the associated peak performance for a set of low-level and kernel codes will be determined.

The goal is not just to choose the best programming paradigm, assuming that one is clearly better than the others, but also to evaluate what can be gained by rewriting codes to exploit a given paradigm. Because of their size, many of the important scientific and engineering application codes will not be rewritten unless significant performance improvement is predicted. The evaluation data will also allow us to interpret the fairness of using a given benchmark code implemented using, for example, pure MPI-1 or OpenMP, when comparing results between platforms.

Evaluation Plan for Cray X1

The deliverables for this aspect of the evaluation are as follows.

- Evaluation of a variety of programming styles appropriate for use in the target application codes, identifying both those well-suited for the Cray X1 and those that should be avoided.
- Performance benchmarks for each paradigm, and conclusions on how to optimize performance. The benchmarks and optimization analysis will be performed for inter-node and intra-node communications separately, as they are likely to differ even within a common paradigm.
- Cross-paradigm comparisons using application-specific kernels, and paradigm recommendations for application codes. Certain of the application codes, for example, the Community Atmospheric Model (CAM) used in climate research, already supports MPI-1/OpenMP hybrid implementations. Some of the other paradigms will require significantly more work to evaluate fairly, and the earlier results will be used to guide this effort, concentrating on the most promising choices.

Initial results of the pure distributed memory and the distributed memory/OpenMP evaluation can be completed quickly and applied to the application benchmarking described in the next section.

Application Evaluation and Benchmarking

The performance and efficiency of applications relevant to the DOE Office of Science in the areas of global climate, fusion, materials, and chemistry will be evaluated. In addition to measures of performance and scalability common to evaluations of microprocessor-based MPP systems, the extent to which Cray compilers and tools can effectively vectorize full application codes will be investigated. The extent to which localized tuning can improve vectorization and efficiency will also be investigated.

ORNL will work closely with the principal investigator(s) leading the Scientific Discovery through Advanced Computing (SciDAC) application teams to identify the leaders and participants in the application evaluation efforts. As described above, initial steps in each domain will be the detailed understanding of selected kernels and critical aspects of the system. At the same time, existing implementations of the full applications (described below) will be ported to the machine. The initial emphasis will be upon correct functioning of the code, and, subsequently, upon sequential and then parallel performance and scalability. Immature compilers and system software will require that this work be performed in close collaboration with Cray. Once a satisfactory port of the existing software has been established, the scientific application evaluation teams will proceed with more detailed examination of the applications, and, as necessary, reformulate and re-implement selected algorithms either for performance or for evaluation of different programming paradigms, as described above. Finally, once the installed computer system has reached the necessary size, and the system and the application software have attained the required levels of robustness and performance, these teams will embark upon an extensive evaluation of the computer as a scientific production resource by performing full-scale simulations.

The following applications will be initial targets for evaluation.

Evaluation Plan for Cray X1

Climate Science: CAM, CLM and POP

The Community Atmospheric Model (CAM) is the atmospheric component of the Community Climate System Model (CCSM), the primary model for global climate simulation in the U.S. and the target of the Climate SciDAC project, “Collaborative Design and Development of the Community Climate System Model for Terascale Computers.” The dynamics algorithm of the CAM is a semi-Lagrangian transport method in combination with a semi-implicit Eulerian spectral method. This code grew out of the spectral models developed at the National Center for Atmospheric Research (NCAR).

The CAM will be used to test the Cray X1 in a variety of ways. The dynamics of atmospheric circulation will test the bandwidth and latency of the X1 interconnect. Simulation of the physical processes in each atmospheric column will test memory bandwidth and vectorization capabilities of the compilers. These studies will provide information on the limitations to performance of non-vector code as well as the style of effective vector coding for the X1. The Community Land Model (CLM) is integrated into CAM, and it will also be used to evaluate memory bandwidth and vectorization capabilities. The transfer of data between the CLM and the atmospheric dynamics will evaluate the X1 interconnect.

After initial performance and scaling measurements of CAM, the effectiveness of targeted tuning, such as modifying physics loops for better vectorization and replacing MPI calls within the dynamics with lower latency SHMEM calls will be investigated. Another investigation will evaluate the feasibility of programming paradigms based on the Multi-level Parallelism (MLP) style of globally addressable blocks. On machines with a single system image, the paradigm has been successful in facilitating scalability by reducing latencies below the message passing interface (MPI) levels. The technique has been successfully applied to the ocean code Parallel Ocean Program (POP) as well as other components of the CCSM. An effort to explore the resolution, scaling and throughput parameter space to understand how best to utilize the X1 for climate science will be undertaken. Target resolutions will range from the current T42 and T85 climate resolutions to high-end weather resolutions.

The POP is the ocean component of CCSM. 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. The POP code is being developed and maintained at Los Alamos National Laboratory (LANL). POP is expected to be amenable to vectorization; the evaluation will test this expectation. The two primary processes in POP will test the Cray X1 in different ways. The “baroclinic” process is three dimensional with limited nearest-neighbor communication and should scale well. The “barotropic” process, however, involves a two-dimensional, implicit solver and will limit scalability. This two-dimensional process will test the latency of the X1 interconnect; early POP results from the NEC SX-6 show that latency is a significant bottleneck. The effectiveness of targeted communication optimizations to minimize latency, such as replacing MPI-1 calls with MPI-2 one-sided calls or SHMEM calls will be investigated.

The target resolutions for ocean calculations will be 1/10 degree and higher horizontal grid spacing. This allows for accurate eddy transport of heat by the ocean currents as well as locating accurately features such as the gulf stream and other important currents. This is the resolution expected on the Japanese Earth Simulator to drive the next series of scientific discoveries. The performance of the key kernels, components, and fully configured coupled models will be documented through reports, conference talks and by providing input to ‘the matrix’ (<http://www.krellinst.org/matrix>). Feedback will be provided to the vendor and input will be sought from experts in optimization at Cray and at other research centers.

Fusion Science: AORSA and Gyro

The All-Orders Spectral Algorithm (AORSA) codes solve for the wave electric field and heating in a stellerator plasma heated by radio-frequency waves. They are important applications in the Fusion

Evaluation Plan for Cray X1

Sciences SciDAC project, “Numerical Calculation of Wave-Plasma Interactions in Multidimensional Systems.” The computation times of AORSA2D and AORSA3D are dominated by the use of ScaLAPACK to solve large dense systems of linear equations. ScaLAPACK shows good efficiency on many computer systems, and the same is expected on the X1. AORSA results from the NEC SX-6 show excellent efficiency using ScaLAPACK, but the results reveal that the matrix generation vectorizes poorly and requires a significant amount of time. The efficiency of the X1 formatrix generation will be evaluated.

Current AORSA development is targeting algorithms that trade smaller linear systems, the solution of which scales as the cube of the problem size for more-expensive matrix generation, that scales as the square of the problem size. The evaluation will help determine the optimal tradeoff for the X1.

The Gyro code solves time-dependent, nonlinear gyrokinetic-Maxwell equations for electrons and ions in a plasma. It is being developed under “The Plasma Microturbulence Project,” a fusion SciDAC project. Gyro uses a five-dimensional grid and propagates the system forward in time using a fourth-order, explicit, Eulerian algorithm. This application has shown good scalability on large microprocessor-based MPPs, and similar scalability is expected on the X1. The extent to which this scalability is enhanced by greater per-processor efficiency will be evaluated.

Materials Science: LSMS

The Locally Self-consistent Multiple-Scattering (LSMS) application uses a real-space, multiple-scattering, Greens function-based method for calculating the electronic structure of materials and for treating the quantum mechanical interactions between large numbers of atoms. LSMS was the first computer model that accurately captured the magnetic interactions responsible for the formation and stabilization of the local magnetic moments observed in neutron scattering data for CuNi and has continued to be extensively used for first-principles materials research. The code scales linearly in the number of atoms, and it is remarkably efficient and scalable on microprocessor-based MPPs. LSMS should approach the theoretical peak performance of the X1. Record setting performance using this method resulted in the award of the 1998 Gordon Bell prize.

The single-processor efficiency of LSMS relies on its use of BLAS3 dense-matrix operations. To scale LSMS to much larger problems, the developers are moving to sparse-matrix formulations, which typically achieve significantly lower efficiency on microprocessor-based systems. The relative advantages of the Cray X1 for these sparse formulations in large number of atom configurations will be evaluated.

Chemistry: NWChem

NWChem is the DOE massively parallel computational chemistry code, and is the first such code designed for efficient parallel execution. It is supported by the Office of Biological and Environmental Research (OBER) as part of the operation of the Environmental Molecular Sciences Laboratory at Pacific Northwest National Laboratory (PNNL). NWChem provides extensive functionality for determining the electronic structure of molecules and solids, including Gaussian-based Hartree-Fock and density functional theory (DFT) with analytic second derivatives, many-body methods such as MP2 (with gradients) and CCSD(T), and plane-wave DFT with a variety of pseudo-potentials and boundary conditions with Car-Parinello dynamics. Also included is an extensive classical molecular dynamics (MD) and free-energy capability that is probably the most efficient such code currently available. The quantum and classical models may be combined to perform mixed quantum mechanics/molecular mechanics (QM/MM) simulations. NWChem has demonstrated efficient scaling to over 1000 processors (MD on a Cray T3E) and the new CCSD(T) capability has been designed for efficient execution up to 10,000 processors.

NWChem is a large code (nearly 1 million lines) with a wide variety of computational kernels that vary in the demands they place upon the underlying hardware. For example, the Gaussian-based DFT and

Evaluation Plan for Cray X1

the MD codes are sensitive to the latency of remote memory access, the CCSD(T) code requires efficient local matrix multiplication, and the MP2 gradients and several other modules require high-performance I/O with both sequential and random access.

In contrast to all of the other applications codes, NWChem uses a distributed, shared-memory programming model supported by the Global Arrays library. Point-to-point message passing is only used in third-party libraries, such as parallel linear algebra or FFT. The NWChem algorithms are typically based upon a non-uniform memory access (NUMA) model. Shared data (in a global array) is assumed to be accessible by any process without the explicit involvement of any other process. This *one-sided access* is critical to realizing the efficiency and ease-of-programming demonstrated in NWChem. Shared memory is assumed to be more expensive to access than local memory (higher latency and lower bandwidth) and the algorithms have been designed to offset this cost by a variety of mechanisms including loop tiling and blocking. The MD uses spatial decomposition, but still employs the Global Array library, rather than message passing, because the one-sided access mechanisms improve the scalability of the code due to increased asynchrony of execution and easier dynamic load balancing.

A plan has been formulated for the PNNL NWChem team (lead by Theresa Windus), the Global Array team (lead by Jarek Nieplocha), and ORNL staff (lead by Robert Harrison) to port and optimize NWChem for use on the X1. Due to the size and complexity of the code, this is a multi-stage project. The initial emphasis will be on porting and optimizing the supporting libraries, optimizing critical kernels, and demonstrating correct sequential and parallel functioning, including passing the full NWChem quality assurance suite. Subsequently, attention will focus upon improving the performance of selected important modules, the overall parallel scalability, and exploring critical aspects of system performance of specific interest to NWChem (e.g., I/O and distributed, shared memory). The MD and two-electron integral modules were initially developed for Cray vector computers, and it is anticipated that they retain many favorable performance characteristics despite subsequent modifications for RISC cache-based machines. However, most of the code has never been tuned for a vector computer, so much work may be necessary to reach a satisfactory base-line performance. In our favor is a wealth of experience of vector computing in computational chemistry and due to this experience it is expected that nearly all of NWChem will eventually attain high performance on the X1.

Deliverables

The deliverables for application evaluation and benchmarking are:

- Performance results for each application, with comparisons to other systems,
- Performance analyses, describing strengths and weaknesses of the X1 relative to other systems, and
- Results of tuning experiments, identifying effective tuning techniques and expectations for performance improvement. The performance results will be documented through reports, conference talks and by providing input to ‘the matrix’ (<http://www.krellinst.org/matrix>). This public forum will allow continuous monitoring of the state of the project and status of the X1.

I/O and System Software Evaluation and Benchmarking

The correctness and usability of the system software are essential components in providing a fully functioning computer system, and a large system suitable for both development and production use makes significant demands on this software. The system software of the Cray X1 provides a variety of unique features, including a single system image, checkpoint and restart, and job migration.

Evaluation Plan for Cray X1

The job-migration feature is a necessity on the X1 because of the requirement that a job must run on a contiguous set of processors to use the X1 interconnect most effectively. Typical job mixes will create distributed gaps, such that enough processors may be available for a given job, but the job cannot start because the processors are not contiguous. With job migration, jobs in process may be moved to different processors to create larger holes. Integration of this feature with the job-management software and its effectiveness at maximizing utilization of the full system will be evaluated.

The single system image will make many aspects of system administration much easier and more efficient than on clusters of SMPs. It has important implications on the scalability and fault tolerance of the system as a whole, however. With large, parallel systems, the ultimate performance of the system can be limited by the reliability and serviceability of the hardware and software. If the system fails frequently or takes a long time to repair, then applications runs are interrupted.

The mean time between failure (MTBF) and mean time to repair (MTTR) for the Cray X1 will be determined for the ORNL environment. The system tools for installing and updating system software, error detection and reporting capabilities of the system, and system monitoring tools will be evaluated. An evaluation of the resilience of the full system to failures of individual components will be performed with the goal of working with Cray to improve the tolerance of the full system to the most common component and system-software failures.

In addition to unique system-software features, the Cray X1 has a unique I/O architecture. All I/O travels over Fiber Channel (FC) links, with "local" disks attached through FC controllers, and external networks attached through external servers that translate from Gigabit Ethernet (GigE) to IP over FC. Configuration variables include the numbers of FC links, FC controllers, RAID arrays, disks per RAID, networking servers, and GigE links. Additional options come from tuning parameters of XFS, the high-performance file system to be delivered with the X1, parameters such as number of stripes, stripe width, and block size. In addition to the I/O load from application codes themselves, checkpoint and restart can impose extreme requirements on file systems. The file system will be configured and tuned in different ways to look for anomalies and performance problems. This work complements the more application-specific evaluations described previously.

Finally, both performance of individual codes and total throughput of the system depend heavily on how well the machine integrates into the total HPC environment. How the Cray system links with external file systems, such as HPSS and NFS, data-analysis servers, and high-bandwidth wide-area networks will be investigated. Again, the Cray X1 has unique configuration variables, including the number of FC links and FC-to-GigE servers.

The deliverables for the I/O and systems software aspect of the evaluation are:

- Functionality and performance of the job-management systems;
- MTBF and MTTR statistics for the X1, as well as a description of all problems and Cray's responses;
- Functionality and performance of the scalability and fault tolerance of the system;
- Data showing the performance of XFS under a variety of configuration and usage patterns, including checkpoint and restart; and
- Recommended tuning parameters and hardware configurations for connecting the Cray X1 to HPSS, NFS, and high-bandwidth wide-area networks.

Evaluation Plan for Cray X1

Scalability Evaluation

Determining the scalability of a system from a given instantiation of an architecture is not a simple exercise. A simpler approach is to attempt to identify system components whose performance will adversely affect scalability.

Extensive studies of the message-passing layers will be conducted, investigating performance behavior under a range of load conditions. Tests will be used that mimic common behavior, such as all-to-all communication patterns, as well as stress tests that create hotspots, with all processors contending for a common resource. Similar studies will be performed with the I/O system. For example, the out-of-core extension of ScaLAPACK to stress test the parallel I/O system will be used. The software is written to use either multiple files local to each processor (when run on a cluster) or access a large shared file on a parallel file system. The software is designed to solve very large systems of linear equations and eigenvalue problems that are several times larger than the total available in-core memory. Large blocks (hundreds of megabytes) of data are transferred in a regular pattern between disk sub-system and memory. Since nearly all available memory is dedicated for computation, it is unlikely that disk caching will be effective.

The emphasis here is different from benchmarking, although the evaluations are similar. The behavior will be measured when scaling problem size, number of processors, or other hardware or software parameters. Then trends are sought that, if extrapolated, would cause performance difficulties for larger systems. If such trends are discovered, the cause will be determined, and whether it is an impediment to scalability will be ascertained. Corrective action for problems identified will be worked with Cray.

Once obvious showstoppers have been eliminated, an attempt will be made to extrapolate performance for well-understood kernels and application codes. This requires codes for which performance models have been constructed. Fortunately, there are a number of codes for which such models exist or for which performance can be bounded from below by characterizations of message-passing behavior or memory pattern accesses. By using a range of problem sizes, these models can be parameterized, validated against existing empirical data, and then be used to estimate performance on a larger system. While the quantitative accuracy of the predictions will be difficult to assess, scaling problems indicated by this analysis should be accurate. The advantage of using models in this study is that the cause of poor scalability will be obvious from the model.

The deliverables for this aspect of the evaluation are hotspot analysis for:

- Inter-node and intra-node communication,
- Shared memory access, and
- Parallel I/O;

as well as

- Trend analysis for both standard and optimized versions of selected communication and I/O patterns;
- Trend analysis for standard and application-specific parallel kernels; and
- Predicting scalability (or lack thereof) of selected kernels and application codes from performance models and lower bounds.

Relation to DOE Science Projects

DOE's science projects -- in accelerator physics, biological science, chemical science, environmental science, fusion science, materials science, nuclear physics, and particle physics -- will be direct

Evaluation Plan for Cray X1

beneficiaries of the knowledge gained from this evaluation project. Initially, selected participants from the major SciDAC scientific application projects will be involved in the evaluation of the Cray X1 either by actively working with the X1 or by acting as a liaison between the evaluation team and the broader DOE science community. In addition, since the evaluation plan addresses questions of performance and programming paradigms using selected benchmarks from each application, scientists will discover firsthand the impact of the CrayX1 on their applications and will share this information with colleagues in other disciplines. The critical performance issues explored as part of this evaluation plan, including the discovery of specialized coding and optimization techniques, will result in a clearly defined path of code modifications and design decisions that must be made in order to adapt the SciDAC scientific applications to the Cray X1 architecture.

Profound implications for the tools and methods projects of the SciDAC Integrated Software Infrastructure Centers will also result from the evaluation process, since vector architectures have been mostly neglected for the past ten years. Without representative hardware, most computer science research and tool development has targeted commodity scalar processors and clusters. Methods research has pushed the scalability issues toward massive, distributed memory configurations. While much of the research will be relevant and applicable to a Cray X1 with thousands of processors, the richness of the Cray software environment and high-bandwidth interconnect will shift the balance. This evaluation will give important information to tool designers and algorithm researchers about the relevance of current designs and implementations.

Finally, as this evaluation includes aspects of the production environment supplied by Cray, it will be of great interest to computational scientists throughout the DOE laboratory complex and to the broader scientific community. Reliability, robustness and ease of use is a relevant question for those intending to use the Cray X1 as an 'extraordinary tool for extraordinary science.' This evaluation will determine how to effectively manage large scientific calculations and to extract information and eventually knowledge, from the large quantities of data. These are the foundations of scientific discovery through advanced computing.