DOE Ultrascale Evaluation Plan of the Cray X1

Mark R. Fahey, Oak Ridge National Laboratory and University of Tennessee
James B. White III, Oak Ridge National Laboratory

ABSTRACT: *In November of 2002, the Center for Computational Sciences at Oak Ridge National Laboratory organized a workshop to develop an evaluation plan for the Cray X1 using applications of relevance to the Department of Energy (DOE) Office of Science. This workshop was followed by application-specific workshops in fusion science, climate modeling, materials science, and biology in early 2003. We describe the findings of these workshops and the resulting plan to evaluate the Cray X1 for ultrascale simulation within the DOE.*

1 Introduction

On August 15, 2002, Dr. Raymond Orbach, Director of the Office of Science of the US Department of Energy (DOE), announced that the DOE would test the effectiveness of the new Cray X1 in solving important scientific problems in climate, fusion, biology, nanoscale materials, and astrophysics [ORNL]. The goal of the evaluation is to predict the capability of ultrascale X1 systems, systems with computation rates of tens of teraflops, on large-scale simulations of critical importance to the mission of the DOE Office of Science.

The Center for Computational Sciences (CCS) at Oak Ridge National Laboratory (ORNL) leads the evaluation project, collaborating closely with other DOE labs and researchers and especially with Cray itself. Since the announcement of the evaluation project, the CCS has hosted a series of DOE-wide workshops, developed an evaluation plan [Bland et al], and fielded a Cray X1 system.

In March of 2003, the CCS accepted delivery of the initial system with 32 MSPs. This system will be upgraded to 256 MSPs by October 2003. For more details, see [Bland].

This report summarizes the evaluation plan, the results of the various application workshops, and early progress implementing the plan. More details on early performance results appear in [Worley] and [White].

2 Evaluation Overview

The primary tasks of the Cray X1 evaluation are the following:

- compare the performance of the X1 with that of other HPC systems,
- determine the most-effective approaches for using the X1,
- evaluate the reliability and performance of the system software and administrative tools,
- predict scalability of the X1, in terms of both problem size and processor count, and
- collaborate with Cray on future system generations using the results of the evaluation.

The plan takes a hierarchical approach to performance evaluation of both software and hardware. For software, it begins with system software, in terms of both performance and stability. This topic is described in further detail in [Bland]. Microbenchmarks describe the raw performance of various subsystems of the X1, while parallel-paradigm evaluations demonstrate the relative strengths and weaknesses of interconnect hardware and communication software. These evaluations complement and help analyze evaluations of full applications. Such analyses will form the basis for the scalability evaluations and predictions that are the final goals of the program.

The initial target of the evaluation work is single-processor performance, investigating vectorization, multistreaming, and memory utilization. These single-processor issues are the emphasis of the current work, using the 32-MSP system now available. As this system grows to 256 MSPs, investigation will proceed with

communication performance, including comparisons of communication libraries and languages, such as MPI, Cray SHMEM, Co-array Fortran, etc.

2.1 Microbenchmarks

The objective of the microbenchmarking effort is to characterize the performance of underlying architectural components of the X1, using both standard and customized benchmarks. These architectural components include the following:

- vector and scalar arithmetic;
- the memory hierarchy, including local and remote shared memory, cache, and registers;
- message-passing performance, including between and within 4-MSP SMP nodes, using MSP- and SSP-based execution;
- process, thread, and stream management, including creation, locks, semaphores, and barriers, with Pthreads, OpenMP, and multistreaming;
- system and I/O primitives, including operating-system overhead, networking, and file I/O operations.

These evaluations are now well underway, and results are available in [Worley].

2.2 Parallel-programming evaluation

The X1 provides various options for implementing inter-process communication, and this evaluation will identify the best techniques for the X1. The options include MPI-1 [MPI1], MPI-2 [MPI2], Cray SHMEM [Barriuso], Global Arrays [Nieplocha], Co-Array Fortran [Co-array], UPC [UPC], OpenMP [OpenMP], and MLP [Taft]. The knowledge of the performance payoffs versus modification effort for each parallel library and language will define optimization strategies for communication-bound applications. Though early results are available [Worley], this phase of the evaluation will be more prominent once larger X1 systems are available.

2.3 Scalability evaluation

The scalability evaluation will attempt to predict scalability from performance models and bounds established through hot-spot and trend analyses. The hot-spot analyses will target potential hot spots within the X1 architecture that may limit scaling. Such hot spots may include communication within or between 4-MSP SMPs, memory contention, and parallel I/O. Trend analyses will target kernel benchmarks and communication and I/O patterns that represent full applications. Scaling studies of these benchmarks and patterns will help validate the predictive models and bounds needed to specify requirements for ultrascale systems. This phase of the evaluation plan clearly requires the largest X1 systems in the current procurement.

2.4 Application Performance

The part of the evaluation plan likely to receive the greatest effort and interest is the application evaluation. This effort targets full scientific applications of interest to the DOE Office of Science that have scientific goals requiring ultrascale computational resources.

The evaluation of the performance, scaling, and efficiency of the chosen applications relies heavily on a complementary evaluation, an evaluation of the ease and effectiveness of targeted tuning for the X1. The remainder of this document describes the application targets for this evaluation.

3 Applications

To identify the appropriate applications to use for the X1 evaluation, the CCS has hosted a series of workshops. The first workshop occurred at ORNL on November 5-6, 2002; it introduced researchers from the DOE Office of Science to the Cray X1 and introduced Cray staff to the application areas of importance to the Office of Science. A series of application-specific workshops followed, covering fusion (February 3-5, 2003), climate (February 6), materials (March 2), and most recently biology (May 9). Future workshops will likely cover chemistry and supernova astrophysics.

The goal of each workshop is to set priorities and plan the work in each application area. The priorities depend on the potential payoff, in terms of performance and scientific results. The work plans attempt to schedule the application pipeline, where this pipeline caries an application through various stages. These stages include porting and development, processor tuning, scalability tuning, and production science runs. The scheduling goal

is to maintain a small number of applications in each stage, thus favoring capability over throughput.

The following sections describe the various application areas, the initial applications selected within each area, and the initial progress on those applications. The current application areas include climate, fusion, materials, and biology. Future application areas are likely to include chemistry and supernova astrophysics.

3.1 Climate

A workshop was held on February 6, 2003, to identify and coordinate efforts in the port of the Community Climate System Model (CCSM) to the Cray X1, including discussions of vectorization and software engineering issues. Within the CCSM, the plan is to evaluate the Community Atmospheric Model (CAM), the Community Land Model (CLM) and the Parallel Ocean Program (POP). The workshop included participants from NCAR, LANL, LBNL, ORNL, NASA-Goddard, CRIEPI, Cray and NEC. The Climate community requires that any optimizations introduced to any of these models are beneficial to all supported systems, including both Cray and NEC.

The current work on CAM involves people from Cray, NEC, NCAR, and ORNL. The radiation and cloud models are the focus of most of the work. NEC expects to have single node optimizations complete by the Fall of 2003, while Cray has recently started and is in the porting and profiling stage. A major issue for CAM is coordination between NEC and Cray, if any, and how they can both arrive at a similar CAM that works well on both architectures.

The land component of the community model (CLM) has been undergoing changes to the data structures to allow for easier extensions later and for maintainability. This is being done with Fortran user-defined types with pointers. Implementing the code as such does not bode well for the vector machines. Similar to CAM, vectorization work on CLM has involvement from Cray, NEC, NCAR, and ORNL, and coordination is a major issue. A promising approach was prototyped at ORNL with the hypothesis that the data structures could be implemented in such a way that they provide similar ease of extensibility and maintenance while being friendly to tuning, optimization, and vectorization. This was tested in the most time consuming routine, Biogeophysics, and significant (serial) improvement was observed. A 20% speedup was witnessed on a Power4 and a 50 times speedup witnessed on the X1 resulting in the X1 being 6 times faster than the Power4 [White].

The Parallel Ocean Program (POP) has been undergoing vectorization and parallel optimizations for months. Again there is multi-institutional involvement with LANL, Cray, NCAR, and CRIEPI and again coordination is an issue. Significant optimizations have been implemented combining both vectorization and Co-Array Fortran on the X1 [Worley].

3.2 Fusion

A Fusion Ultrascale Computing workshop was held February 3-5, 2003. There was multi-institutional participation from General Atomics, PPPL, U. of Iowa, U. of Wisconsin, Cray, and ORNL. The outcome of the workshop with respect to the evaluation plan was that six codes would be ported and analyzed on the X1. These codes are M3D, NIMROD, GYRO, GTC, AORSA, and TORIC and they cover the fusion sub-areas of extended magnetohydrodynamics (MHD), micro turbulence, and radio-frequency plasma interactions. Furthermore, teams were identified for all but one to begin work concurrently. Although six codes may seem too many, this does allow for work to continue even when some ports encounter short- or long-term impediments. For example, the M3D code uses the PETSc library. Until a vector port of PETSc is completed, the M3D work can only be on a functional port.

M3D is a code for simulating the MHD of fusion plasmas in three dimensions. M3D uses finite differences in the radial direction and FFTs in the toroidal and poloidal directions, and PETSc provides the elliptic solver for its quasi-implicit time-integration method. PETSc accounts for 90% of the computation time on microprocessor-based MPPs in M3D, thus a tuned port of PETSc is essential for success of M3D on the X1. Results on the SX-6 have raised issues about the performance of PETSc on vector systems. A vector port of PETSc is estimated to take 6 person-months and will require significant code changes. We plan to soon have an M3D developer on the CCS X1 to get a simple port of PETSc and to obtain some baseline data. ORNL is also trying to help facilitate a more formal PETSc port.

The "Non-Ideal Magnetohydrodynamics with Rotation, and Open Discussion" (NIMROD) code is designed to study three-dimensional, nonlinear,

electromagnetic activity in fusion experiments while allowing for flexibility in the geometry and physic models. NIMROD employs a conjugate-gradient solver with a parallel line Jacobi preconditioner. NIMROD has shown scaling to large numbers of processors on MPP machines despite strong global coupling and complicated data structures. Early profiling of NIMROD on the X1 has revealed a couple weaknesses of the Cray Fortan compiler. Namely, the compiler does not effectively implement the Fortran "reshape" intrinsic. In addition, the compiler is not able to vectorize loops around Fortran "sum"s where the summed objects are pointers. Both problems have been submitted as problem reports to Cray. These two issues alone have a dramatic effect on performance.

NIMROD also extensively uses derived types of pointers rather than allocatable arrays. The compiler in some cases cannot vectorize loops where these data structures are used since it cannot know if pointers point to overlapping data or not, or even if the data are contiguous in memory or not. It is expected the X1 as well as other architectures could benefit by replacing the pointers with allocatable arrays. However, this is not a trivial exercise and would require a significant code rewrite. The success reported above with CLM could influence the decision.

The Gyrokinetic Toroidal Code (GTC) is a three-dimensional particle-in-cell simulation code for microturbulence studies in magnetically confined plasmas. The code solves the nonlinear Gyrokinetic Vlasov-Maxwell system of equations using particle-in-cell methods for the dynamic equations and iterative (including multi-grid) methods for the elliptic field equations, with MPI and OpenMP parallelization. The code has been running on the IBM SP at NERSC using from 64 to more than 2000 processors, with a parallel efficiency of up to 98% and with only 5% of the computing time spent for inter-processor communications. There is strong interest in the evaluation of the performance of the X1 processors for gather and scatter (random access) operations for GTC. GTC has been run on an SX-6 already, but its efficiency was 9% less than that on a Power3. It is currently being ported and optimized to the X1.

The Gyro code solves time-dependent, nonlinear gyrokinetic-Maxwell equations for electrons and ions in a plasma. 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. Gyro has been ported to the X1 and has undergone some vectorization. Since Gyro uses real allocatable arrays as opposed to derived types, the compiler is able to vectorize and multistream loops. To make the vectorization and multistreaming effective though, directives and manual loop interchanges are needed. Code modifications were implemented to vectorize cosine and sine function evaluations that accounted for a non-trivial percentage of compute time. In one instance this meant changing the algorithm. The current optimizations have resulted in a five times speedup over the original port, and performance 35% faster than the Power4. Further improvements are expected.

The All-Orders Spectral Algorithm (AORSA) code solves for the wave electric field and heating in a stellerator plasma heated by radio-frequency waves. 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. However early results show that the Cray ScaLAPACK library requires further tuning.

AORSA performance on the X1 has performed worse than expected, even given that the ScaLAPACK library requires tuning. The matrix scaling was identified as performing extremely poorly, and a fix is being implemented and tested. 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 for matrix generation will also be evaluated.

3.3 Materials

A Materials workshop was held on March 2, 2003 in Austin, TX. The goals of the workshop were to follow up on the ultrascale simulation initiative white papers [Ultrascale], provide a prioritized list of application codes to be ported to the X1, and provide a list of names and projects to be associated with these codes. The selected codes were DCA, FLAPW, LSMS, and Socorro.

The Dynamical Cluster Algorithm (DCA) is implemented with MPI and OpenMP and uses BLAS and PBLAS routines. Significant amounts of time are spent in the DGER and CGEMM calls. On the IBM Power4 for example the BLAS level 2 calls dominate. A two-day port of DCA was performed that included adding directives into a few routines that were identified for optimization. The result was dramatic speedup over the

same run on the Power4. For this test, the time spent on computations became nearly negligible while the time doing I/O became dominant.

The Full Potential Linearized Augmented Plane Wave (FLAPW) method is an all-electron method considered to be the most precise electron structure method in solid state physics. It is used primarily as a validation code and as such is important to a large percentage of the materials community. Cray has begun porting this code.

The Locally Self-consistent Multiple Scattering (LSMS) method is an order-N approach to the calculation of the electronic structure of large systems within the local density approximation. The electronic-structure problem is reduced to that of calculating the single particle Green's function at the central atom of a finite cluster of sites.

This method is highly scalable on an MPP supercomputer since each compute node can be assigned the calculation of the scattering matrix elements, the electron density, and the density of states for the atoms mapped onto it. The code is dominated by matrix multiply calculations. A large amount of time is also spent calculating a partial inverse of a large matrix that is contained within a node (25x25) block. The communication involves exchanges of smaller matrices with neighbors. This code is expected to vectorize well.

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.

Socorro is a highly scalable and extensible density functional code. Socorro is designed from the ground up to work on massively parallel systems such as the ASCI computers, but it works equally well using a single processor. It is written in object-oriented Fortran, with some C included. The libraries used by the code include BLAS, LAPACK, ScaLAPACK, and FFTW. The FFTs are 100x100 and 100^2 of them are done at a time. Roughly ten percent of the code's time is spent in LAPACK calls with most of that spent in eigensolver calls. Porting to the X1 has yet to begin.

3.4 Biology

The Biology Workshop was held on May 9, a few days before CUG. The goals of the workshop, like the previous workshops, were to identify one or two codes of importance to the DOE that we expect to vectorize. However, the biology community is diverse, and in particular, the researchers in Life Sciences/Bioinformatics had different goals. Since they use many codes, identifying one or two codes is not useful. Rather they wanted to specifically talk about what the Cray X1 is good at, and what non-traditional facilities can be accessed so they could exploit these special features. The results of the workshop will be covered more in the presentation.

4 Future Work

It is early in the DOE evaluation of the X1, and so the future is to continue the evaluation which is underway. The workshops held so far have been extremely useful in determining applications to port to the X1, and identifying the teams to do the work. We plan to hold workshops for Chemistry and Astrophysics later this year as well, with similar results.

5 Acknowledgments

The authors thank Pat Worley of ORNL for all his effort on the evaluation plan and the Climate workshop summary. The authors thank Rebecca Fahey of ORNL for her summary of the Materials workshops. The authors also thank the internal CCS reviewers for their comments.

This research was sponsored by the Mathematical, Information, and Computational Sciences Division, Office of Advanced Scientific Computing Research, U.S. Department of Energy, under Contract No. DE-AC05-00OR22725 with UT-Battelle, LLC.

6 About the Authors

Mark R. Fahey, part of the Joint Institute for Computational Science, is in the Scientific-Application-Support Group within the CCS. Mark is the primary CCS liaison for fusion researchers funded by the DOE. He can be reached at faheymr@ornl.gov. James B. White III, a.k.a. Trey White, is in the Scientific-Application-Support Group within the CCS. Trey is the primary CCS liaison for climate researchers funded by the DOE. CUG 2004 will be hosted by the CCS, and Trey is the Local Arrangements Chair. He can be reached at whitejbiii@ornl.gov.

7 References

[ORNL] "Department of Energy's Oak Ridge National Lab to Test New Cray Supercomputer for U.S. Science," http://www.ccs.ornl.gov/PR/craytest.html.

[Barriuso] Barriuso R., and Knies A., *SHMEM User's Gui*de, Cray Resarch, Inc., May 1994.

[Bland et al] Bland A., Dongarra J., Drake J., Dunigan T., Dunning T., Geist A., Gorda B., Gropp W., Harrison R., Kendall R., Keyes D., Nichols J., Oliker L., Simon H., Stevens R., White J., Worley P., and Zacharia T., "Cray X1 Evaluation," Technical Report ORNL/TM-2003/67, March 2003.

[Bland] Bland A., Alexander R., Carter S., and Matney K., "Early Operations Experience with the Cray X1 at the Oak Ridge National Lab for Computational Sciences," proceedings of the 2003 Cray User Group, May, 2003.

[Co-array] *Cray Fortran Co-array Programming Manual*, Cray Private Draft S-3908-42, December 2002.

[MPI1] *MPI 1.1 Standard*, http://www-unix.mcs.anl.gov/mpi/mpich.

[MPI2] *MPI-2: Extensions to the MPI Interface*, http://www-unix.mcs.anl.gov/mpi/mpich.

[Nieplocha] Nieplocha J., Ju J., Krishnan M., Palmer B., and Tipparaju V., "The Global Arrays User's Manual," Pacific Northwest National Laboratory Technical Report Number PNNL-13130, October 2002.

[OpenMP] OpenMP, http://www.openmp.org.

[Taft] Taft J., "Performance of the OVERFLOW-MLP Code on the NASA Ames 512 CPU Origin System," NAS Technical Report NAS-00-005, NASA Ames Research Center, March 2000.

[Worley] Worley P. and Dunigan T., "Early Performance Evaluation of the Cray X1 at Oak Ridge National Laboratory," proceedings of the 2003 Cray User Group, May 2003.

[White] White J., "An Optimization Experiment with the Community Land Model on the Cray X1," proceedings of the 2003 Cray User Group, May 2003.

[Ultrascale] Ultrascale Simulation for Science, http://www.krellinst.org/ultrasim/doe_docs/index.html.

[UPC] Unified Parallel C, http://upc.gwu.edu.