Cray XMT Brings New Energy to High-Performance Computing

The ability to solve our nation's most challenging problems—whether cleaning up the environment, finding alternative forms of energy, or improving public health and safety—requires new scientific discoveries. High-performance experimental and computational technologies from the past decade are helping to accelerate these scientific discoveries, but they introduce challenges of their own.

The vastly increasing volumes and complexities of experimental and computational data pose significant challenges to traditional high-performance computing (HPC) platforms as terabytes to petabytes of data must be processed and analyzed. And the growing complexity of computer models that incorporate dynamic multiscale and multiphysics phenomena place enormous demands on high-performance computer architectures.

Just as these new challenges are arising, the computer architecture world is experiencing a renaissance of innovation. The continuing march of Moore's law has provided the opportunity to put more functionality on a chip, enabling the achievement of performance in new ways. Power limitations, however, will severely limit future growth in clock rates. The challenge will be to obtain greater utilization via some form of on-chip parallelism, but the complexities of emerging applications will require significant innovation in high-performance architectures.

The Cray XMT (figure 1), the successor to the Tera/Cray MTA, provides an alternative platform for addressing computations that stymie current HPC systems, holding the potential to substantially accelerate data analysis and predictive analytics for many complex challenges in energy, national security, and fundamental science that traditional computing cannot do.

The Cray XMT has a unique "massively multithreaded" architecture (sidebar "Cray XMT System Description," p38) and large global memory configured for applications—such as data discovery, bioinformatics, and power grid analysis—that require access to terabytes of data arranged in an unpredictable manner.

The Cray XMT holds the potential to substantially accelerate data analysis and predictive analytics for many complex challenges in energy, national security, and fundamental science that traditional computing cannot do.

Cray, Inc./PNNL

Figure 1. *Deborah Gracio, computational and statistical analytics director; Moe Khaleel, director of computational sciences and mathematics; and senior scientist Andres Marquez, with Pacific Northwest National Laboratory's new Cray XMT supercomputer.*

Computations with more dynamic and adaptive requirements tend to perform poorly on standard architectures.

Cray XMT System Description

The Cray XMT is the commercial name for the new multithreaded machine developed by Cray Inc. under the code name "Eldorado." By leveraging the existing platform of the XT3/4, Cray was able to save non-recurring development and engineering costs by reusing the support IT infrastructure and software, including dual-socket Opteron AMD service nodes, Seastar-2 high-speed interconnects, fast Lustre storage cabinets, and the associated Linux software stacks.

Changes were performed on the compute nodes by replacing the AMD Opterons having a custom-designed multithreaded processor with a third-generation MTA architecture that fits in XT3/4 motherboard processor sockets. Similarly to the previous MTA incarnations, the Threadstorm processor schedules 128 fine-grained hardware streams to avoid pipeline stalls on a cycle-by-cycle basis.

Analogous to the Opteron memory subsystems, each Threadstorm is associated with a memory system that can accommodate up to 16 GB of memory. Each memory module is complemented with a data buffer to reduce access latencies. Memory is hashed and structured to support fine-grain thread synchronization with little overhead. Memory shares a global address space.

The key component of a Seastar-2 network is a full system-on-chip design that integrates six high-speed serial links and a 3D router on each compute node. Sustained random remote accesses peak at around 114 million operations and level off at around 44 million operations with a full complement of 4,000 Threadstorm processors.

Another important element of the Cray XMT is the storage system, which is based on a Lustre version that also has been deployed in the Cray XT4. Lustre has been designed for scalability, supporting I/O services to tens of thousands of concurrent clients. Lustre is a distributed parallel file system and presents a POSIX interface to its clients with parallel access capabilities to the shared objects.

A Natural Platform for Data-Intensive Computing

To understand the potential utility of the Cray XMT, we must first consider the characteristics of applications that perform well on current distributed memory, message-passing machines. Memory locality is a key feature of most successful HPC applications. Simulated physical interactions are low-dimensional and often short-range, leading to computations in which data-access patterns have spatial locality. Spatial locality enables effective use of memory hierarchies and allows for the explicit data partitioning that is necessary for distributed memory computing. Applications without a high degree of spatial locality are quite difficult to execute and scale on most current machines.

Another important feature of current parallel applications is ease of load balancing. Most physical simulations involve meshes or particles with predictable and stable computational requirements. Slowly varying computational requirements can be addressed by infrequent, but expensive, redistributions of data and work amongst processors. Computations with more dynamic and adaptive requirements tend to perform poorly on standard architectures.

Most scientific computations can be performed in a bulk-synchronous manner in which processors alternate between phases of collective communication and computation on local data. This computational structure maps well to distributed memory machines for two reasons. First, messages can be aggregated in the communication phase, so latencies, which are high on current machines, are reduced in importance. Second, the ability to perform computations on only local data allows processors to be maximally efficient. However, this programming style is lacking in flexibility. Data can be transmitted only at pauses between computational steps, and the lack of transmission on demand makes it difficult to exploit fine-grained parallelism in an application.

Thus, current generations of parallel machines are well suited to problems with a high degree of spatial locality, a stable or slowly varying distribution of computational requirements, and a high degree of coarse-grained parallelism that allows for a bulk-synchronous programming model. Fortunately, many important scientific and engineering computations have these characteristics. However, a number of important emerging applications do not.

Problems that deal with massive amounts of high-dimensional information and low locality include social and technological network analysis, such as identification of implicit online communities, viral marketing strategies, quantifying centrality, and influence in interaction networks, and web algorithms; systems biology, such as interactome analysis, epidemiological studies, and disease modeling; and homeland security, such as detecting trends, anomalous patterns from socio-economic interactions, and commu-

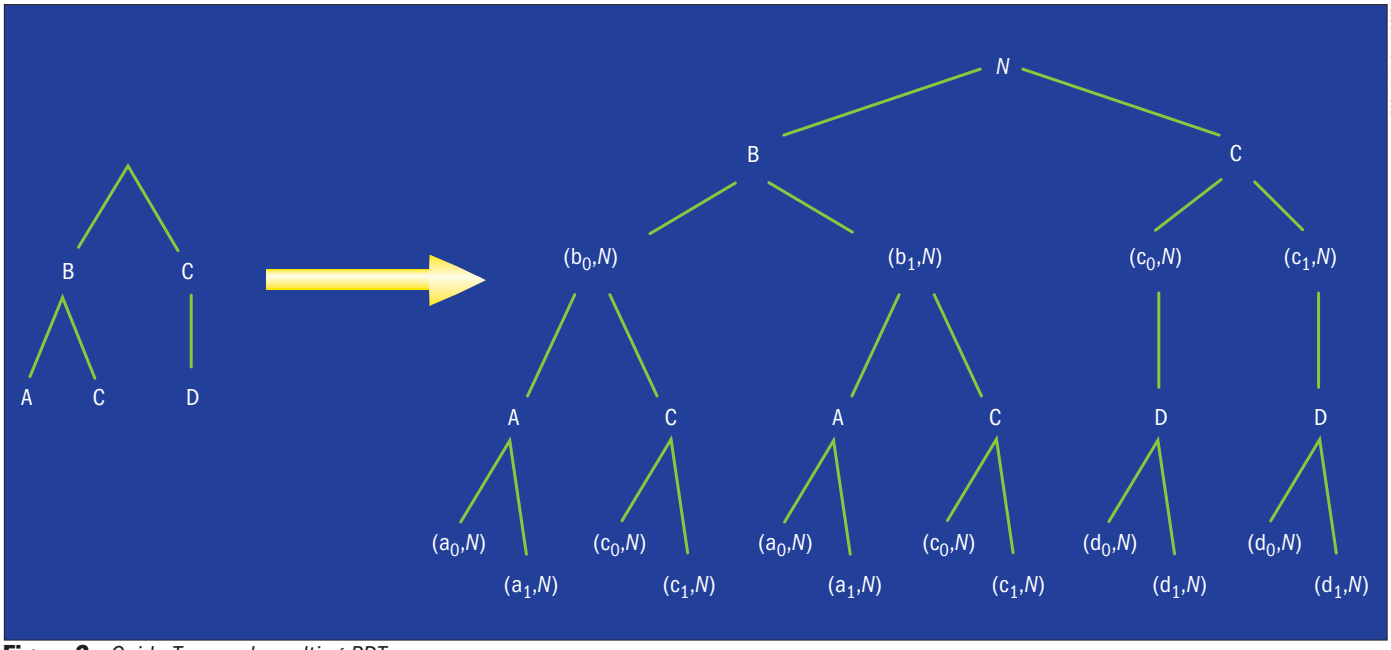

ILLUSTRATION: A. TOVEY

Figure 2. *Guide Tree and resulting PDTree.*

nication data. Further, the data may be dynamically generated and can arise from heterogeneous information sources. Often, there is little computation to do on data items, and the nature of computation can be difficult to characterize, involving a mix of floating-point, integer, and string operations, as well as extensive use of combinatorial algorithms and data structures. For these reasons, these applications are considered data-intensive rather than compute-intensive, and the performance depends more on how well the system manages the application memory access patterns rather than the raw computational capability of the system. Also, parallel approaches to solving these dynamic data problems may not be amenable to a bulk-synchronous formulation, and current message-passing HPC machines lack the ability to exploit fine-grained parallelism in these applications.

In contrast to traditional HPC machines and programming models, the Cray XMT would be a more natural platform for solving data-intensive applications that are characterized by dynamically varying computations and low degrees of spatial locality. The Cray XMT relies on the massive multithreading paradigm of programming to exploit concurrency in applications and tackles the memory-latency issue very differently from traditional HPC architectures. The Cray XMT provides the programmer an illusion of a globally addressable flat memory hierarchy, thus avoiding the need for load-balanced data partitioning and redistribution among processors. Instead, it tolerates latency through hardware support for multiple outstanding memory requests, facilitated by fine-grained threads and fast context switches on each processor. The Cray XMT supports lightweight word-level synchronization primitives for minimizing memory contention among threads. The Cray XMT's massive multithreading approach also supports dynamic load balancing and adaptive parallelism, leading to significant benefits in programmer productivity in the design of algorithms for data-intensive applications. Memory locality and computational intensity of the application have little effect on performance, as long as the programmer identifies and exposes sufficient concurrency at a fine granularity.

Application Case Studies

New applications are being created and existing applications are being mapped to the Cray XMT platform with promising results. Preliminary results indicate that the Cray XMT is able to achieve good performance and scalability on challenging, highly irregular applications.

Application 1: Anomaly Detection for Multivariate Categorical Data (PDTree)

The PDTree application originates in the cyber security domain and involves large sets of network traffic data. Analysis is performed to detect anomalies in network traffic packet headers in order to locate and characterize network attacks, and to help predict and mitigate future attacks. The PDTree application is a special case of a more widely applicable analysis method that uses ideas from conditional probability in conjunction with a novel data structure and algorithm to find relationships and patterns in the data.

Preliminary results indicate that the Cray XMT is able to achieve good performance and scalability on challenging, highly irregular applications.

ILLUSTRATION: A. TOVEY

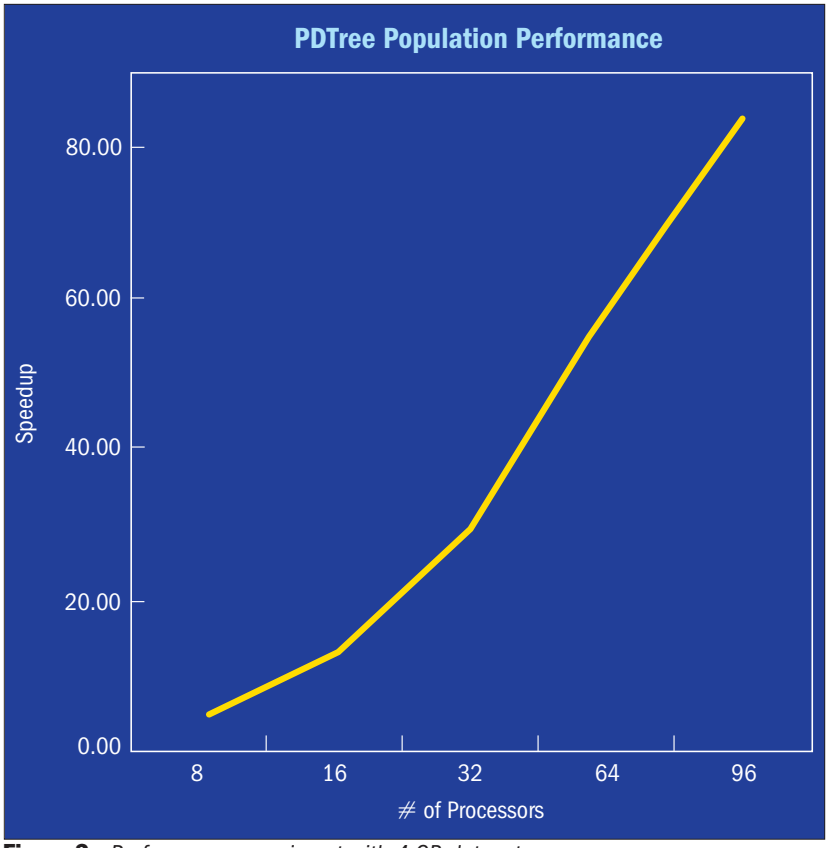

Figure 3. *Performance experiment with 4 GB dataset.*

When dealing with categorical data in multiple variables we can ask, for any combination of variables and instantiation of values for those variables, how many times this pattern has occurred. Because multiple variables are being considered simultaneously, the resulting count table, or contingency table, specifies a joint distribution. Efficient algorithms using contingency tables are critical for implementation of Bayesian networks, log-linear models, Markov random fields, various graph representations, and novel algebraic-based approaches. This is especially important when there are a large number of variables and observations or when some variables take on many distinct values. All of these hold with regard to the massive amounts of data prevalent in cyber security analysis.

The PDTree data structure stores counts for selected combinations of variables in categorical data. The selection of the combinations, specified as a Guide Tree (figure 2, p39), can be derived from a formal statistical model or from requirements regarding memory space. We have designed and implemented a multithreaded, parallel version of the population of a PDTree from a raw, multiple variable, categorical dataset. The implementation utilizes the features found on the Cray XMT to enable the execution of this dynamic, data-dependent application. We used an *n*-ary tree with a specialized root level to implement concurrent, fine-grained tree node updates and insertions and took advantage of the low-cost atomic operations and full/empty bit synchronization provided by the Cray XMT.

For our application, an experiment was designed that streamed a large categorical dataset resident on a parallel Lustre file system—from the Opteron processors on the Cray XMT to the Threadstorm processors—in order to be inserted into the PDTree data structure. The dataset contained 64 million records, and we used a guide tree with nine columns. We streamed the data using 250 MB chunks, which corresponds to approximately four million records. Figure 3 presents the performance results achieved in this experiment. The results show the scalability potential of the Cray XMT on an application with highly irregular, data-dependent memory accesses and high degrees of fine-grained parallelism.

Application 2: Survey Propagation for Complex Systems in Biology

Biology recently has taken a sharp turn into complexity, in ways that are relevant to HPC and statistical physics. Message passing between elements is a common theme in many complex systems, whether they are atoms, economic agents, logical variables, or neurons. Each cell in an organism is an intricate chemical machine that is constructed from a massive number of heterogeneous discrete elements organized into multifaceted, multiscale networks. Large amounts of data about these networks are rapidly accumulating due to high-throughput experimental technologies in the new discipline of systems biology. However, making sense of this deluge of complex data is likely to challenge current paradigms and demand real innovation in advanced algorithms and computing systems.

Recently, tools designed to study spin glasses, as well as the interaction of atomic ensembles, have proven useful in developing more efficient algorithms for solving hard combinatorial problems. This transfer of insight across domains highlights the intriguing similarity between phase transitions in physical systems and sharp thresholds found in computational complexity. Especially interesting is the role of data exchange across probabilistic graphical models, or message passing, as an algorithmic and conceptual framework.

Survey Propagation (SP) is a message-passing algorithm that is able to solve especially large, hard Boolean satisfiability problems, up to 10 million variables close to the threshold of solvability. SP is a generalization of (loopy) Belief Propagation that transmits a probabilistic "sur-

vey" about the variables along the edges of a bipartite factor graph composed of variable and clause nodes. The algorithm exchanges information along the graph until a consensus emerges in a fixed-point solution.

We are investigating SP as an efficient solver for use against computationally hard problems in biology; for example, graph-theoretic problems on transcriptional, metabolic, protein–protein interaction and protein-homology networks. This strategy is motivated by the extraordinary diversity of algorithms needed against large-scale data in modern biology. A powerful central engine can simplify the challenge of applying HPC to a complex array of problems. Hundreds of NP-complete problems from graph theory, network design, set and partition, storage and retrieval, sequencing and scheduling, mathematical programming, game theory, logic, automata and language theory, and optimization have been identified (see Further Reading). The problems in this complexity class can be solved by reduction to another form; for example, from *k*-clique to satisfiability (SAT).

Just as cellular complexity is essentially a mystery under intensive investigation, the complexity of these NP-complete problems is a parallel challenge. We have chosen the graph theoretic problem of finding cliques, or fully connected subgraphs, in protein-homology networks as an initial target because we are easily able to interpret results from a biological perspective using a visual analytics toolkit that we have developed.

Early results have shown some surprises. A simple reduction of a protein-homology network to three-clique unexpectedly provided a bonus by consistently finding much larger cliques. Tuning the properties of these reduction algorithms to a specific purpose will require many computational experiments to define the geometry of their complexity. Thus, an efficient implementation of a general solver such as SP is a workhorse that can produce not only new biological understanding, but theoretical computer science as well.

We have implemented a parallel version of SP on the Cray XMT and have evaluated its performance in the context of randomly generated Boolean satisfiability problems with up to 10 million variables and 50 million clauses in preparation for understanding its performance for problems derived from biological datasets. Figure 4 presents parallel speedup results for SP on a 64-processor Cray XMT for a problem with one million Boolean variables and five million clauses. Scalability and parallel speedup are almost linear for this very challenging combinatorial problem, which is very difficult to execute at scale on traditional HPC systems.

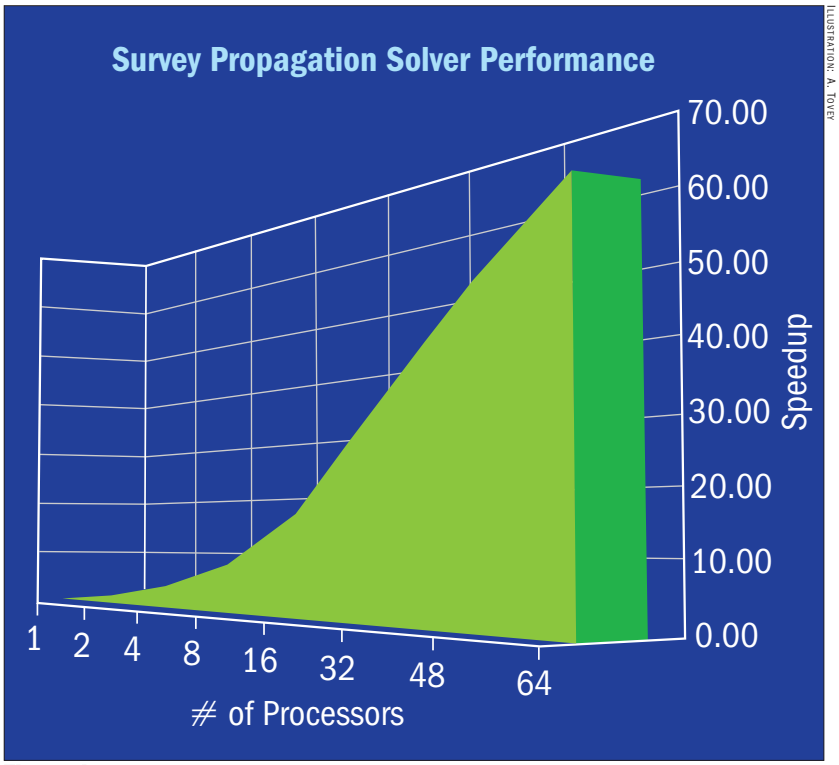

ILLUSTRATION: A. TOVEY

Figure 4. *Performance of Survey Propagation on the XMT for a one million Boolean variable problem.*

Expanding the Use of HPC

The Cray XMT is a disruptive architecture. It supports unprecedented performance for complex, memory-limited applications, and so has the potential to substantially broaden the range of applications benefiting from HPC. A recent workshop sponsored by the DOE Office of Science and the Department of Defense explored possible applications for this machine. In addition to applications in biology and knowledge discovery discussed above, the workshop identified agent-based modeling, cognitive science, combinatorial optimization, and other emerging applications. Scientific communities that depend on these computations have been limited consumers of traditional HPC because their applications do not align well with the capabilities of message-passing clusters. Our hope is that the Cray XMT will better meet their needs and bring new energy and ideas into the HPC world. ●

Our hope is that the Cray XMT will better meet needs and bring new energy and ideas into the HPC world.

Contributors: Daniel Chavarria-Miranda, Deborah Gracio, Andres Marquez, Jarek Nieplocha, Chad Scherrer, and Heidi Sofia, Pacific Northwest National Laboratory; David A. Bader and Kamesh Madduri, Georgia Institute of Technology; Jonathan Berry and Bruce Hendrickson, Sandia National Laboratories; Kristyn Maschhoff, Cray, Inc.

Further Reading

www.nada.kth.se/~viggo/wwwcompendium