Cray and AMD Scientific Libraries

Mary Beth Hribar, *Cray Inc.*, **Chip Freitag**, *AMD*,
Adrian Tate, *Cray Inc.*, **and Bracy Elton**, *Cray Inc.*

ABSTRACT: *Cray provides optimized scientific libraries to support the fast numerical computations that Cray's customers require. For the Cray X1 and X1E, LibSci is the library package that has been tuned to make the best use of the multistreamed vector processor based system. For the Cray XT3 and Cray XD1, AMD's Core Math Library (ACML) and Cray XT3/XD1 LibSci together provide the tuned scientific library routines for these Opteron based systems. This paper will summarize the current and planned features and optimizations for these libraries. And, we will present library plans for future Cray systems.*

KEYWORDS: LibSci, ACML, Cray X1, Cray X1E, Cray XD1, Cray XT3, BlackWidow, scientific libraries

1 Introduction

Scientific libraries are a component of Cray's programming environment software. For each Cray system, these libraries provide basic numerical functions that have been highly tuned for that architecture. To access the best performance of Cray systems, use these libraries.

For the Cray XD1™ and Cray XT3™ systems, the AMD Core Math Library (ACML) provides most of the scientific library support. For convenience, ACML is included in Cray's software distribution. On the Cray's vector platforms, the Cray X1/X1E™ series systems and the planned BlackWidow system, LibSci® contains the scientific library routines.

This paper states the features, release schedules and plans for Cray's scientific libraries and ACML.

2 Cray XT3 Libraries

The Cray XT3 is a massively parallel processing (MPP) system designed to provide the scalability, performance and reliability to solve the most demanding high performance computing problems. The Cray XT3 contains processor nodes in a high-bandwidth, low-latency 3-D torus interconnect. Each node includes an AMD Opteron processor, a dedicated memory and a Cray SeaStar™ communication chip. The Cray SeaStar chip contains a HyperTransport™ link to the Opteron processor, a Direct Memory Access (DMA) engine to manage memory accesses, and a router that connects to the system interconnection network. The Cray SeaStar chip offloads the communications processing from the Opteron processor, increasing the efficiency of the computation and communication within an application program.

There are two types of nodes in the system. Service nodes perform the functions needed to support users, administrators, and applications running on compute nodes. These nodes run a full-featured version of SuSE LINUX. Compute nodes currently run a microkernel named *Catamount*, developed by Sandia National Laboratories. Later this year, a lightweight LINUX kernel will also be available for the compute nodes. Programming environment software, including scientific libraries, is provided on both types of nodes.

The Cray XT3 programming environment includes versions of the 64-bit AMD Core Math Library (ACML) to support the GNU and PGI compilers. The Cray XT3 programming environment also includes a scientific libraries package, Cray XT3 LibSci. It contains a much smaller set of library routines than LibSci for the Cray X1/X1E. The Cray XT3 1.4 release of Cray XT3 LibSci will include:

- ScaLAPACK
- BLACS
- SuperLU_DIST

2.1 FFTs for Cray XT3

In the Cray XT3 1.4 release, the only FFTs available to users are those in ACML. Starting with the Cray XT3 1.5 release, there will be more options for FFTs.

2.1.1 FFTW

In the 1.5 release, FFTW will be distributed as part of the Cray XT3 software. Version 3.1.1 of FFTW will be provided first, followed by version 2.1.5 later in 2006.

The first release of FFTW will provide the performance of library as-is. Later, pre-built plans, known as "Wisdom", will be included with the library. The FFTW Wisdom repository provides near optimal plans for the problem sizes and types it spans. (It is near optimal because some items, e.g., data alignment, are not captured via this mechanism.)

Version 2.1.5 will be provided for its distributed memory parallel FFTs. Support for distributed memory parallel FFTs was dropped in the 3.0 release of FFTW and is still not provided in the 3.1.1 release. Users are directed to use the older release to access the parallel FFTs.

2.1.2 Cray FFT interface

Support for the Cray FFT interface will be available in the 1.5 release. The Cray FFT interface will be provided for those routines that map to ACML FFTs. Setting the input parameter ISYS=0 in the Cray FFT interface will result in the corresponding tuned ACML FFT to be used.

Note that not all cases of the single processor Cray FFT interface are supported in ACML. So, only a subset of the total Cray FFTs will be provided. Furthermore, users will need to pay attention to the "table" and "work" array requirements. ACML FFTs use only one auxiliary array while the Cray FFTs typically use two. Consequently, in the Cray FFTs for the Cray XT3 system, the "table" argument will also contain what would otherwise be the separate "work" array. Please consult the man pages for instructions for using the Cray FFTs.

The distributed memory parallel Cray FFTs will be available to users in a later release. These routines call the single processor version of the ACML FFTs.

2.2 ScaLAPACK for Cray XT3

In future releases of XT3 LibSci, ScaLAPACK will be designed to exploit the XT3 communications system by increasing message size and decreasing message count. Since the communications methodology in ScaLAPACK is currently governed by the initial distribution of data amongst processors, this will require a de-coupling of the linear-algebra block size from the distribution block size. When those block sizes are no longer coupled, smaller block sizes can be used for the single processor computations and much larger block sizes can be used in communications. Having the flexibility to control these block sizes separately provides more options for performance tuning.

Cray has been actively supporting the development of the MRRR algorithm, the first $O(n^2)$ eigensolver for the reduced tridiagonal system. The ScaLAPACK version of this algorithm has been developed by Christof Voemel from LBNL. Cray has been supporting this project due to demand from the electronic structures community for better symmetric eigensolvers. XT3 Libsci will include a beta release of the code by the end of 2006.

2.3 Support for Sparse Iterative Solvers

We presently support SuperLU, a direct sparse solver, and are developing library support for iterative solvers. Most sparse systems that use the full scale of the Cray XT3 system will demand application-specific iterative solvers. Recognizing the enormous variety of iterative methods, we will focus on providing optimized building blocks for iterative solvers.

We plan to provide a set of routines that provide the core functionality of the 2002 sparse BLAS standard (sparse matrix-vector and matrix–block product, triangular sparse solution for one or several right-hand sides). We will provide an interface to these routines that is usable by both PETSc and Trilinos, and also from the user interface defined in Algorithm 818 from the ACM TOMS collection. We initially plan to support as sparse matrix representations the common compressed sparse row form, the fixed blocked generalization provided in PETSc and the variable blocked generalization used in Trilinos. As optimizations, we will investigate the preprocessing capabilities from OSKI and its blocked variant of the CSR representation.

2.4 Goto BLAS

Cray is currently in the process of licensing the right to distribute Goto BLAS on the Cray XT3 systems. Kazushige Goto at the Texas Advanced Computing Center developed the Goto BLAS. He tuned them for Translation Lookaside Buffer (TLB) misses, and they have shown better performance than routines tuned for reuse of data in the cache. The Goto BLAS contain single processor and threaded (OpenMP) versions of the BLAS.

Our preliminary tests show that Goto BLAS provide better performance than the ACML BLAS for complex data types. For double precision real data types, the ACML BLAS perform faster. We plan to offer Goto BLAS in addition to ACML so that Cray XT3 users have access to the best performing BLAS routines for all data types.

3 Cray XD1 Libraries

The Cray XD1 is also an MPP system comprised of AMD Opteron processors. The Cray XD1 operating system is based on the SuSE Linux Enterprise Server (SLES) distribution. This operating system supports both 32- and 64-bit applications. The programming environment software includes 32- and 64-bit ACML libraries, with versions to support the GNU and PGI compilers. The OpenMP ACML is available in 32- and 64- bit versions for use with the PGI compilers. ScaLAPACK and BLACS are also provided.

The current software release for the Cray XD1 is 1.4, which will be generally available in the summer of 2006.

4 Cray's Vector Libraries

Scientific libraries for Cray's vector systems are contained in LibSci. In addition, the Libm library contains C and Fortran mathematical intrinsics.

The libraries for Cray's next generation vector system, BlackWidow are based on X1/X1E libraries. BlackWidow will have a faster CPU, and more memory bandwidth than the X1E, though it will not have multistreaming processors (MSP). BlackWidow, like the X1 and X1E, combine vector processors with both shared and distributed memory.

The Cray X1 and X1E systems are constructed of nodes within a node interconnection network, each of which contains four multistreaming processors (MSPs) and globally addressable shared memory. Each MSP contains four single-streaming processors (SSPs).

4.1 LibSci for Cray X1/X1E

LibSci provides Fortran interfaces for all routines. It supports32- and 64-bit default data types. On Cray X1 and X1E systems, LibSci also supports MSP mode and SSP mode. The latest release is LibSci 5.5.

4.1.1 Single processor routines

LibSci contains single processor support for:

- Fast Fourier transform (FFT), convolution, and filtering routines
- Basic Linear Algebra Subprograms (BLAS)
- Linear Algebra Package (LAPACK) routines
- Sparse direct solvers

Libm contains single processor support for:

- scalar mathematical intrinsics, such as EXP, LOG, and SIN
- Vector mathematical intrinsics
- 32-, 64-, and 128-bit real types
- Random number generation
- Other C and Fortran language features

4.1.2 Distributed memory parallel routines

LibSci contains multiprocessor support in a distributed memory environment for:

- FFT routines
- Scalable LAPACK (ScaLAPACK) routines
- Basic Linear Algebra Communication Subprograms (BLACS)

4.1.3 Shared memory parallel routines

LibSci also contains four-way shared memory parallel support across a single node for all Level 3 BLAS routines and for the Level 2 BLAS routines sgemv, dgemv, cgemv, and zgemv. This library is implemented with OpenMP, and including the -lompsci option on the link line accesses it.

4.1.4 Inlining LibSci

There is a small set of LibSci routines that can be inlined with the -O inlinelib option. All Level 1 BLAS routines can be inlined as well as some Level 2 BLAS routines (sgemv, dgemv, cgemv, zgemv, sger, dger, cgerc, cgeru, zgerc, and zgeru).

4.2 LibSci for BlackWidow

LibSci for BlackWidow supports the same functionality as for the Cray X1/X1E systems, with a few additional features. Also, since BlackWidow's processors are not multistreaming processors, LibSci is not needed to support MSP and SSP modes of execution.

4.2.1 Additional shared memory parallel routines

LibSci for BlackWidow will contain additional support for the four-way SMP nodes. In addition to parallel BLAS, there will be parallel FFTs, some parallel LAPACK routines (sgetrf, dgetrf, cgetrf, zgetrf, spotrf, dpotrf, cpotrf, zpotrf, ssytrd, dsytrd, csytrd, zsytrd, ssytrd, dsytrd), and parallel sparse direct solvers.

These parallel routines will be implemented with OpenMP and will be integrated into LibSci. It will no long be necessary to specify the –lompsci option at link time. At execution time, the shared memory parallel routines will be used if the application is launched to run on the SMP node using the –d option.

4.2.2 Support for Sparse Iterative Solvers

Most sparse systems require application-specific iterative solvers. Recognizing the enormous variety of iterative methods, we will focus on providing optimized building blocks for iterative solvers.

As for the XT3, we plan to provide a set of routines that provide the core functionality of the 2002 sparse BLAS standard, with an interface usable by PETSc, Trilinos and Algorithm 818. We initially plan to support as sparse matrix representations the common compressed sparse row form, the fixed blocked generalization provided in PETSc and the variable blocked generalization used in Trilinos. We will also provide two representations optimized for the X1 and X2 vector hardware, with preprocessing capabilities to convert the other representations into either of the vectorizable representations.

4.2.3 Optimizations

LibSci will be tuned for the BlackWidow architecture. This architecture contains different cache and memory sizes from the Cray X1/X1E architecture. Routines that are memory bandwidth bound will need to be evaluated and improved, if necessary.

The BlackWidow architecture contains support for fast one-side communication as the Cray X1/X1E systems do. Communication in ScaLAPACK and the distributed memory parallel FFTs will be tuned to exploit this feature.

5 ACML

The AMD Core Math Library (ACML) is a package of numerical routines tuned specifically for the AMD64 platform processors, including the Opteron. This library provides the basic numerical functions for the Cray XT3 and Cray XD1 systems. ACML contains:

- BLAS
- Sparse Level 1 BLAS
- LAPACK
- ACML FFTs
- Random Number Generators

5.1 Improvements to BLAS and LAPACK

Several improvements were applied to ACML since CUG 2005. Among these is the addition of large array versions especially for the PGI compiler. This fixes an issue when arrays larger than 4 GB are used. Also improvements were applied to dgemm and dtrsmm to improve performance on small (up to 360x360 elements) problems. Also, efficiency improvements were made to larger problems, allowing ACML to match the LINPACK performance of the best available BLAS implementations. Finally, an API for ILAENV was added to the LAPACK to allow programs to dynamically modify various parameters that affect LAPACK performance.

5.2 Random Number Generators

The ACML library now contains a suite of Random Number generators. There are 5 base generators and the option to add a user supplied base generator. The base generators include the NAG Basic, L'Ecuyer's Combined Recursive, Wichman-Hill, Mersenne Twister, and Blum-Blum-Schub. Multiple streams can be provided, as well as generator state preservation and recall. The base generators were tested using the Big Crush, Small Crush and Pseudo Diehard test suites from the TestU01 software library.

There are 26 distribution generators that can produce either continuous or discrete random streams, and have either univariate or multivariate interfaces.

5.3 ACML FFT routines

ACML provides a set of highly tuned FFT routines with an interface to those routines that is unique to ACML. Since there is no established standard for FFTs, this interface is different from the Cray FFT interface. In the Cray XT3 1.5 software release, most ACML FFTs can be accessed through the Cray FFT interface by setting the input parameter `ISYS=0`. For full documentation of the ACML FFT routines, refer to the *AMD Core Math Library User's Guide*. A few highlights are given here.

There are routines to compute one-dimensional, two-dimensional and three-dimensional complex-to-complex FFTs. There is also a routine to compute multiple complex-to-complex one-dimensional FFTs. The standard versions compute in place with unit stride and fixed scale. The expert versions of the routines (routine names are appended with an "X") allow for out-of-place computation with selectable scales. The one-dimensional and two-dimensional expert FFT interfaces also allow a non-unit stride. The three-dimensional expert FFT interface currently only allows a unit stride.

There are also routines to compute one-dimensional real-to-complex and complex-to-real FFTs (single and multiple). These routines compute in place with a unit stride and fixed scale. Also, the complex data is stored in an unusual manner that is documented in the user's guide. Make a note of this format when using complex results from the real-to-complex routines or inputting complex data into the complex-to-real routines.

There are OpenMP versions of the two-dimensional and three-dimensional complex-to-complex FFT routines in ACML, in the `acml-mp` version.

Plan builders were implemented for the ACML FFT routines in the past year. These provide an API that will automatically time how long various combinations of radices take to transform a given problem size. The combination that provided the best performance can then be used for subsequent calls. Although this has an initial up front time cost, FFT performance can be improved up to 15% when large numbers of the same sized FFTs must be computed.

5.4 Fast and vector math functions in ACML_MV

An AMD64 optimized libm is available in glibc in the SuSE SLES9 and SL9.x distributions. Using the fast math routines in ACML_MV can provide further performance improvements. These fast routines may sacrifice accuracy, so the user should determine if the arguments are suitable to use the fast math routines.

The ACML_MV library also contains vector intrinsics of some of the libm routines. These intrinsics can be called in assembly language, or by C compilers that support XMM register m128 data types (such as gcc). The PGI and GCC 4.0 compilers do incorporate these vector intrinsics when producing optimized code. Finally, the ACML_MV library contains array versions of the libm routines. These are callable from C or Fortran, and provide an efficient way to perform the desired transcendental function on an array of *n* input values. The currently supported libm functions in ACML_MV are log, log10, log2, logf, log10f, log2f, exp, expf, sin, cos, sincos, sinf, cosf, sincosf, pow (scalar only), powf and powxf (scalar and vector).

5.5 Upcoming features

A release of ACML is scheduled at the end of June to coincide with ISC 2006. This release will feature optimized level 1 BLAS routines for SSE3 capable Opteron processors, improved performance for 2D complex-complex FFTs, and new expert complex-complex 3D routines that provide for non-unit strides in the 3 dimensions. Other additions will include RNG performance enhancements and new performance measuring examples.

The year-end release of ACML will feature new convolution and correlation routines. In 2007 work will start on Sparse Level 2 and Level 3 BLAS routines. Optimized direct and iterative sparse solvers are also being considered.

6 Summary

Cray offers a set of scientific library routines on each Cray platform. AMD's Core Math Library (ACML) provides highly tuned scientific libraries for Opteron processors, and it is included in the Cray XD1 and Cray XT3 software distributions. Cray and AMD are working together to ensure that ACML contains the performance and features required by Cray's customers.

Cray will add FFTW and Goto BLAS to their software offering on the Cray XT3 systems. These third party software packages provide more performance and portability to users.

Cray continues to improve algorithms and implementations for their scientific libraries. Collaborations with universities and government labs provide assistance with this work, and more joint projects are encouraged.

7 Acknowledgments

The authors would like to acknowledge all of those who contributed to this paper and to thank AMD for their support in providing a presentation at CUG. We acknowledge the following members of the Cray Scientific Libraries Group: Chao Yang who optimizes the BLAS, LAPACK, sparse solvers and the linpack benchmark, Neal Gaarder who tunes libm, and John Lewis who provides technical assistance in all areas of linear algebra and who develops sparse routines.

8 About the Authors

Mary Beth Hribar is the manager of the Cray Scientific Libraries Group. She can be reached at Cray Inc., 411 First Ave S, Suite 600, Seattle WA 98104. Her email address is marybeth@cray.com.

Chip Freitag is the project manager for the math libraries at AMD. He can be reached at AMD, 5204 E. Ben White Blvd, MS 649, Austin TX 78741. His email address is chip.freitag@amd.com.

Adrian Tate is a member of the Cray Scientific Libraries Group. He is the project lead for ScaLAPACK tuning. He can be reached at Cray Inc., 411 First Ave S, Suite 600, Seattle WA 98104. His email address is adrian@cray.com.

Bracy Elton is a member of the Cray Scientific Libraries Group. He is the project lead for FFT development and tuning. He can be reached at Cray Inc., 411 First Ave S, Suite 600, Seattle WA 98104. His email address is elton@cray.com.