Cray X1 Scientific Library Optimisation

Adrian Tate
HPC Team, Manchester Computing, University of Manchester

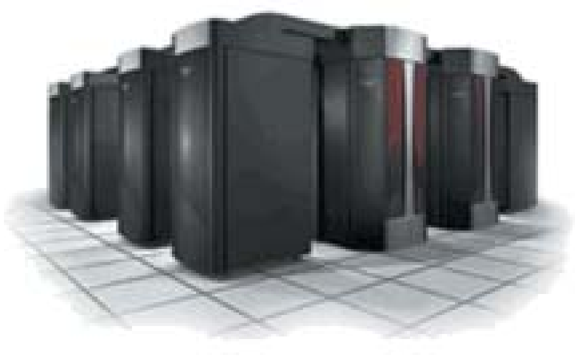

Figure 1: Cray X1

At the Cray User Group Summit in Columbus, Ohio, Cray Inc. announced that the tuning of parallel subroutines within a future release of the X1 scientific library will be carried out by staff of the SVE group at the University of Manchester. This builds on previous work for several CSAR consortia in which the performance of parallel numerical library routines has been successfully increased. By the autumn, ScaLAPACK will be part of the LibSci release, a subset of which will perform optimally after the communication procedures have been overhauled by SVE staff.

The Cray X1 has a number of major architecture differences to its predecessor – the T3E. The distributed-shared memory system is similar to that of a an SGI Origin, but unlike the Origin (or any other system to date) the X1 couples this DSM architecture with 16 vector registers on each processor and with a clock speed of 800 MHz, giving a peak performance of 12.8 Gflops per processor. The system has been designed with MPI codes in mind, but Co-array Fortran, UPC and SHMEM can offer extremely good performance.

Cray are keen to market the machine as highly suited to existing distributed memory applications, and as such are keen to incorporate DM numerical libraries that are efficient and scalable. SVE were selected by Cray because of their experience in creating one-sided communication procedures using SHMEM and Co-array Fortran and because of their knowledge of the complex internal workings of ScaLAPACK and its dependent PBLAS and BLACS libraries. UPC, Co-array and SHMEM operate at very low latency on the Cray X1 and are implemented in such a way that remote data does not enter a local processor's cache but is loaded directly into vector registers. This feature prevents cache invalidation by remote data which has been a problem with implementations in the past, and allows very high speed message passing. In addition, Cray have worked heavily on the Co-array Fortran compiler within Ftn and can enforce pre-fetching with any Co-arrays that appear in an application. Hence, a Co-array version of ScaLAPACK is an attractive prospect and is likely to offer extremely good performance.

An initial pilot project will involve a revision of the communications procedures within the LU factorisation routine pcgetrf/pzgetrf. This will involve complete replacements to four BLACS communication routines and some major alterations to the PBLAS, which govern the communication patterns necessary to achieve parallel execution of Lapack routines. Communications will be a mixture of SHMEM and co-array Fortran, the initial testing of BLACS replacements looks very promising. A longer project will involve a much deeper restructuring of comms patterns within Scalapack, with much work to decrease the programming complexity of specific operations by the capturing of useful functionality into a comprehensive library of efficient programming tools. In addition, Cray are aware of the limitations and restrictions that accompany Scalapack usage, and are interested in the enhancement and development of the user interface to develop a highly tuned, accessible library that may allow less rigorous distribution methods that suit better the attributes of the application.

If you are interested in this work please see the next *CSAR Focus* for an update, or contact Adrian Tate directly:

Contact Details

Telephone: 0161 275 7029

Email: adrian.tate@man.ac.uk