



CASL Computational Science Seminar

Title: The LAW Library: Towards Tools For Improved Productivity with HPC Linear Algebra Codes

Speaker: Tim Stitt (ICHEC)

Date: Tue 16th October 2007 at 2:00PM

Location: CASL Seminar Room - Belfield Office Park

Abstract:

As research groups and vendors compete to develop larger terascale and future petascale systems with ever increasing theoretical peak performance, a productivity crisis could be looming, as end-users of these novel architectures are typically presented with non-trivial, unfriendly parallel-programming tools, libraries and development environments, which can require valuable time and prior technical experience, in order to be fully exploited. $i/p\hat{z}$

To remedy this scenario, it is imperative that high-level, user friendly, portable library tools are developed which abstract away the underlying system-dependent complexity and provide the user with simple, yet powerful interfaces, allowing them to concentrate solely on solving their scientific problem without being distracted by lower-level architectural issues. Furthermore, the scientific code developer should not be inconvenienced with maintaining separate codes for each target machine. In the ideal case, a single code repository should be maintained which can be mapped onto

serial, shared-memory or distributed-memory architectures at compile and link time with little user interaction or modification. i/pç

While important HPC libraries such as MPI, LAPACK 3.0, BLAS, ScaLAPACK, PBLAS and PETSc have succeeded somewhat in raising the abstraction bar, they can still arguably be considered as low-level tools, due in part to their mnemonic routine signatures, complicated invocation arguments and reliance on architecture-dependent features, particularly by those users with limited development experience. While historically this development approach may have been unavoidable, modern programming languages possess advanced abstraction features such as modularity and polymorphism whereby current HPC library tool developers should be able to fully abstract away system dependent issues, leaving the developer to concentrate on the scientific domain details, to which they are more comfortable and accustomed. i/pç

In this talk I will introduce a set of high-level wrapper library routines to the high-performance BLAS and PBLAS libraries, which are independent of the target architecture, for a set of common linear algebra operations, by exploiting advanced features of Fortran 95 such as generic functions, derived-types, optional arguments, overloaded operators and source code preprocessing. By providing a single abstract user interface, developers are excused from considering a given architecture, whether it be serial, shared-memory multithreaded or distributed-memory message-passing, when invoking the routines and reasoning about their data storage semantics. It is believed such an environment will result in the rapid prototyping and development of scientific codes with automatic multi-architecture executables being generated during the linking phase with minimal user interaction. i/pç

<mailto:sam.dolan@ucd.ie>