SANS

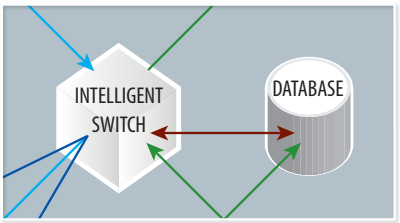
http://icl.cs.utk.edu/sans/

SELF ADAPTIVE NUMERICAL SOFTWARE

Self Adaptive Numerical Software (SANS) is a collaborative effort between different projects that deals with the optimization of software at different levels in relation to the execution environment. The SANS effort intends to identify the commonalities of these different projects and help build a common framework on which these projects can coexist.

ALGORITHMIC DECISIONS

SALSA



The top layer of the SANS system comprises of software that dynamically adapts its workings to the nature of the user data. Such algorithm mediation finds its justification in that many users may lack the expertise to find the best numerical method for their application. With the aid of heuristics, and guided by a database of knowledge gleaned from earlier runs, an automated analysis can pick close to an optimal method for each problem.

The Self-Adapting Large-scale Solver Architecture (SALSA) incorporates heuristics for picking suitable linear solvers, with a range of choices that spans iterative methods, preconditioners, direct solvers, and multigrid methods. Considerations that determine the particular choice include spectral and structural information on the system.

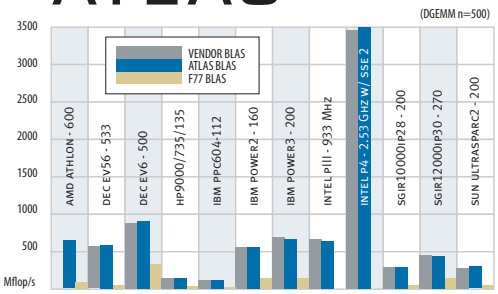
SYSTEM COMPONENTS

On the middle layer of a SANS system one finds software that investigates the availability and state of network resources and lays out a parallel realization of the program accordingly.

LAPACK For Clusters (LFC) aims to bring the performance of ScaLAPACK and the expertise of advanced performance tuning to an average user familiar with the LAPACK interface. The LFC approach hides the complexity of parallel application development, deployment, and use in fashion similar to that of the computational grid. Encapsulation of expert knowledge in high performance parallel, numerical linear algebra enables optimized use of existing hardware resources and software technologies. The modes of computation include spawn-and-wait as well as client-server architectures and are available from scripting environments like Matlab and Python across the various platforms on which these scripting languages are supported.

OPTIMIZED LIBRARIES

ATLAS



The lowest layer of a SANS system provides optimal implementations of computational kernels on a given architecture. In a one-time installation process, a search-direct code generator finds realizations of the desired kernels optimally adapted to the platform.

The Automatically Tuned Linear Algebra Software (ATLAS) package tunes the implementation of BLAS Levels 1, 2, and 3 for each architecture it is run on. It optimizes for cache reuse, register blocking, TLB access, etc. The resulting performance is competitive with, or better than, vendor libraries; on systems where no vendor BLAS library exists it gives the user several factors of improvement over the reference implementation.

SPONSORED BY

THE UNIVERSITY of TENNESSEE
Computer Science Department

ICL UT
INNOVATIVE COMPUTING LABORATORY

CITR
CENTER for INFORMATION TECHNOLOGY RESEARCH

SANS

http://icl.cs.utk.edu/sans/

SELF ADAPTIVE NUMERICAL SOFTWARE

Numerical problem solving is subject to multiple options for an efficient solution process. Thus, there is opportunity for intermediate software layers that assist the user in solving the numerical problem efficiently by adapting in various ways to the computational environment. We call such software layers SANS: Self-Adapting Numerical Software. There are three layers on which SANS can operate.

Firstly, most numerical problem areas know a large repertoire of candidate methods. Sometimes the choice of method can be made if some piece of exact knowledge of the problem data is known, but in many cases the *a priori* determination has to be made on heuristic grounds if the cost of the decision is not to outweigh the actual cost of solving the problem. In this sense, software can adapt to the user data by picking a suitable algorithm or parameterization.

The Self-Adapting Large-scale Solve Architecture (SALSA) is an example of this algorithmic decision layer, applied to linear systems solving. It uses statistical and data mining techniques to tune its heuristics based on a database of performance results.

SALSA

http://icl.cs.utk.edu/salsa/

Secondly, the actual realization of a numerical method on a parallel platform is influenced by dynamic conditions, which are often impossible to access by the user. Thus, an adaptive environment would include a system component that evaluates network conditions and schedules work accordingly.

LAPACK For Clusters (LFC) operates on this level. It aims to provide a user-friendly realization of the ScaLAPACK solver routines, in effect, condensing the expert knowledge needed into a software level. Finally, in a more static sense, computational kernels can be adapted to specific processor architectures, since different implementations can have widely varying performance characteristics. This sort of adaptation is almost completely independent of the user data or dynamic conditions. Thus it can be performed at installation time. Yet another potential for optimization is offered by LFC's newest computational mode, with parallel server and clients running an interactive Matlab or Python shell. The optimization has to be carried at runtime based on user's data size and their type.

LFC

http://icl.cs.utk.edu/lfc/

The third layer optimizes the basic numerical kernels at the installation time. The ATLAS (Automatically Tuned Linear Algebra Software) project has been singularly successful in this respect, delivering automatically tuned BLAS Level 1, 2, and 3 routines that match or exceed the performance of vendor libraries. Its search algorithms generate multiple implementations that take into account caches on multiple levels, Translation Look-aside Buffer, registers, floating-point units, and other hardware characteristics in order to arrive at optimal performance. The recent advances are optimized for various pseudo-parallelism such as Hyperthreading as well as for true multi-CPU designs. These techniques will position the project well in the upcoming multi-core chip era.

ATLAS

http://icl.cs.utk.edu/atlas/

The ACCELS (ACcelerated Compressed-storage Elements for Linear Solvers) project delivers similar improvement on sparse operations but due to inherent structural differences in user's sparse data, a model had to be recently employed that efficiently cuts down the optimization time on variety of modern computing architectures.

THE UNIVERSITY of TENNESSEE
Computer Science Department

ICL UT
INNOVATIVE COMPUTING LABORATORY

CITR
CENTER for INFORMATION TECHNOLOGY RESEARCH