Sisal Language Project

Paring Parallel Production Prices

Objective

The objective of the Sisal Language Project is to simplify the process of writing scientific programs on parallel supercomputers by developing high-performance functional compilers and runtime systems and mixed-language programming environments.

Impact

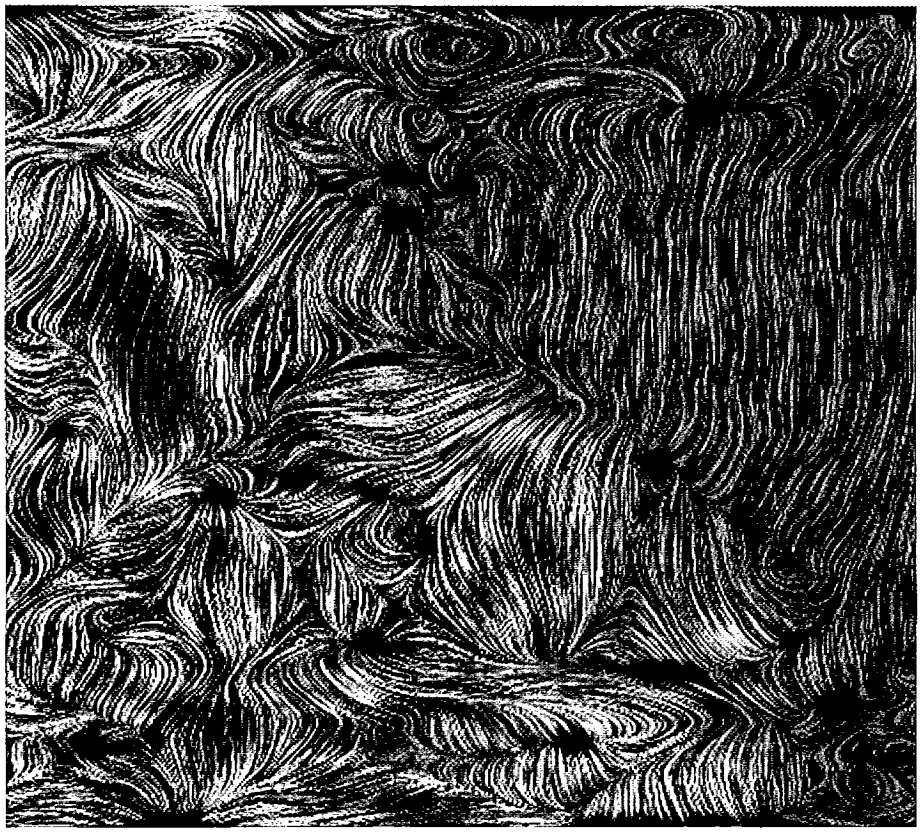
When appropriately coupled with traditional languages, functional languages such as Sisal provide a low-cost approach to developing parallel computing applications that still provide high performance and portability. This is of major significance in an area where costs are projected to top $1 trillion.

Despite the commercial availability of multiprocessor computer systems, the number of parallel scientific and commercial applications in production use today remains small. Based on the cost of writing sequential imperative software, it is estimated that the worldwide cost of developing software will reach $450 billion in 1995. Even if only a quarter of the software is parallelized, the cost will soar to $1 trillion. Functional programming can reduce this cost and still provide high performance and portability.

Parallel programming in imperative languages, such as Fortran and C, has proven difficult. These languages were developed for sequential computer systems and do not naturally support parallelism. In addition to expressing the algorithm, the programmer must encode the program's synchronization and communication operations, ensure data integrity, and safeguard against race conditions. The extra programming complexity increases costs, and the time-dependent errors created by the use of these languages can frustrate even the most experienced programmers.

Technology

Functional languages, such as Sisal, promote the construction of correct parallel programs by isolating the programmer from the complexities of parallel processing. Based on the principles of mathematics, Sisal exposes implicit parallelism through data indepen-



Computational fluid dynamics field imaged by a line integral convolution algorithm written in Sisal.

dence and guarantees determinate results. A functional program comprises a set of mathematical expressions that map inputs to outputs. Sisal programs are referentially transparent and free of side effects, deadlock, and nondeterminacy. The determination of data dependencies, scheduling of operations, the communication of data values, and the synchronization of concurrent operations are realized automatically by the compiler and runtime system. The programmer does not and cannot manage these operations. Sisal programs that run correctly on a single processor are guaranteed to run correctly on any multiprocessor, regardless of architecture. Relieved of the most onerous chores of parallel programming, the software developer is free to concentrate on algorithm design and application development.

Today, most Sisal programs outperform equivalent Fortran programs compiled using automatic vectorizing and parallelizing software, and run comparably to hand-written parallel Fortran codes. Due to the simplicity of the functional programming model, the development cost of the Sisal programs is five to seven times less than that of the Fortran programs. Most importantly, the Sisal Foreign Language Interface allows Sisal programs to call C and Fortran and allows C and Fortran programs to call Sisal. Thus, a user needs to rewrite only the computational kernel of the application in Sisal to realize most of the benefits provided by massively parallel computer systems. Since the computational kernel is usually a small fraction of the code, programming costs are reduced further.

Mature Sisal systems exist for any shared-memory, UNIX single- or multi-processor system, including super-scalar and vector supercomputers. Sisal also runs on Macintosh and PC systems. Currently, we are developing Sisal systems for multithreaded and distributed memory systems. The latter will be built on top of Split-C and active messages. Sisal systems include a high-performance compiler; high-performance, micro-tasking runtime system; debugger; and programming tools. Recently, we released a new language definition—Sisal90. The new language manual is now available. Interested parties can get the Sisal software, manuals, and technical reports via anonymous ftp from **sisal.llnl.gov:pub/sisal**. Information about Sisal is also available from the Lawrence Livermore National Laboratory (LLNL) World-Wide Web home page.

Technology Transfer

LLNL, IBM Corporation, and BioNumerik Pharmaceuticals signed a Cooperative Research and Development Agreement (CRADA) to develop a high-performance, Sisal–Fortran 90 programming environment for IBM systems, and to write *ab initio* quantum mechanical and statistical mechanical, molecular dynamics codes to facilitate the rational design of cancer drugs. The project is a three-year, $6 million effort funded by the Department of Energy Technology Transfer Initiative with in-kind contributions from IBM Corporation and BioNumerik Pharmaceuticals. The project is in its second year.

LLNL and TERA Computer Company also signed a CRADA to develop a high-performance, Sisal–Fortran 90 programming environment for TERA's MTA system and to port important scientific applications to MTA using the mixed-language programming environment. Target applications are DYNA3D, *ab initio* quantum mechanical and statistical mechanical molecular dynamics codes for rational drug design, and black oil reservoir simulation. The project is a three-year, $2.2 million effort funded by the Department of Energy Technology Transfer Initiative with in-kind contributions from TERA Computer Company. The project is in its first year.

Sisal Scientific Computing Initiative

The Computing Research Group is accepting proposals for the Sisal Scientific Computing Initiative. This initiative, sponsored by LLNL and the Department of Energy, is intended to promote the use of Sisal and educate the scientific community in the art of functional programming. Members of the Computing Research Group will provide free Cray computer time, educational material, training, consulting, and user services to scientific applications programmers interested in writing Sisal applications.

For more information, contact John Feo, 510-422-6389, feo@diego.llnl.gov.

Work was performed under the auspices of the U.S. DOE by LLNL under contract No. W-7405-ENG-48.