SSI: Scalable Software Infrastructure for Scientific Computing

Akira Nishida

Mission

The Scalable Software Infrastructure for Scientific Computing Project was initiated in November 2002, for the purpose of constructing a scalable software infrastructure to replace existing implementations of parallel algorithms in individual scientific fields. The project covered the following four areas: iterative solvers for linear systems, fast integral transforms, their effective implementation for high performance computers of various types, and joint studies with institutes and computer vendors, in order to evaluate the developed libraries for advanced computing environments.

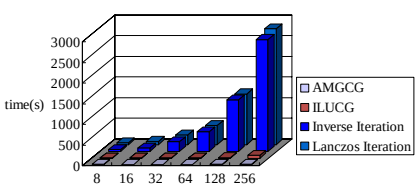
An object-oriented programming model was adopted to enable users to write their parallel codes by just combining elementary mathematical operations. Implemented algorithms are selected from the viewpoint of scalability on massively parallel computing environments. The libraries are freely available via the Internet, and intended to be improved by the feedback from users. Since the first announcement in September 2005, the codes have been downloaded and evaluated by more than 140 organizations around the world.

Lis: A Library of Iterative Solvers for Linear Systems

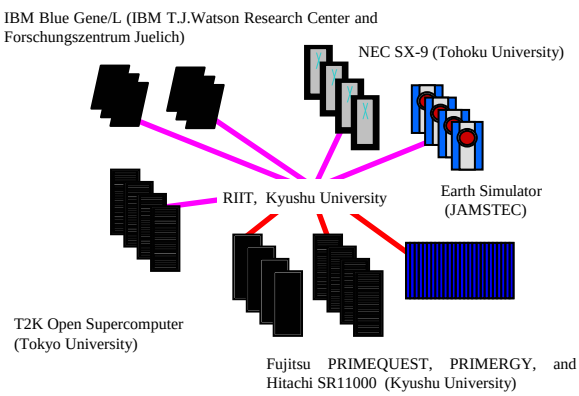
In the fields such as quantum physics and fluid mechanics, we have to solve large scale linear systems to compute numerical solutions of differential equations.
Lis, a Library of Iterative Solvers for linear systems, is an efficient parallel implementation of iterative methods for both linear equations and eigenproblems, with various solvers, preconditioners, and sparse matrix storage formats. Lis is released under the modified BSD license, and supports easy migration from serial environments to massively parallel computing environments with tens on thousands of processors.

```
1: LIS_MATRIX    A;
2: LIS_VECTOR    b, x;
3: LIS_SOLVER    solver;
4: int           iter;
5: double        times, itimes, ptimes;
6:
7: lis_initialize(argc, argv);
8: lis_matrix_create(LIS_COMM_WORLD, &A);
9: lis_vector_create(LIS_COMM_WORLD, &b);
10: lis_vector_create(LIS_COMM_WORLD, &x);
11: lis_solver_create(&solver);
12: lis_input(A, b, x, argv[1]);
13: lis_vector_set_all(1.0, b);
14: lis_solver_set_optionC(solver);
15: lis_solve(A, b, x, solver);
16: lis_solver_get_iters(solver, &iter);
17: lis_solver_get_times(solver, &times, &itimes, &ptimes);
18: printf("iter = %d time = %e (p=%e i=%e)¥n", iter, times,  ptimes,
19: lis_finalize();
```

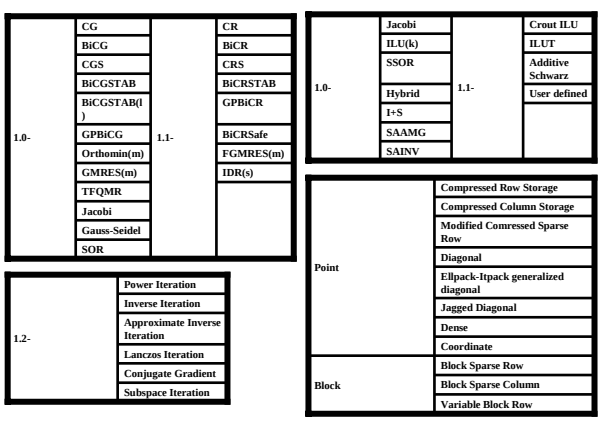
An example code for C



Computation time for smallest eigenvalues of matrices derived from Poisson equation, weak scaling up to order of 16,000,000



Current platforms for our development



Supported solvers and formats

Research Institute of Information Technology
KYUSHU UNIVERSITY
Hakozaki, Higashi, Fukuoka 812-0054 Japan
http://www.cc.kyushu-u.ac.jp/

Scalable Software Infrastructure for Scientific Computing Project
Leader: Akira Nishida
http://www.ssisc.org/