SSI: Overview of Simulation Software Infrastructure for Large Scale Scientific Applications

Akira Nishida

Department of Computer Science, Tokyo University
JST CREST

International Workshops on Advances in Computational Mechanics
Innovative Computational Strategies for Parallel/Grid Environments
Motivation

- Emergence of large scale parallel scientific simulations
- Domestic Efforts for numerical infrastructure
  - Mainly developed in national supercomputing centers (for mainframes and vector computers) in 1980s
  - Some works from industry (E.g. Fujitsu’s SSL II for VPP series)
- Development in US
  - ScaLAPACK (with BLAS and LAPACK), PETSc, Trilinos(former Aztec), etc.
    - Developed and used in national laboratories
    - Standardized and modularized
    - Run on massive parallel computing environments
    - Distributed via WWW (netlib etc.) since 1990s
- Demands for scalable and portable parallel numerical libraries
Brief History of Basic Numerical Libraries

- Projects in US and Europe
  - NATS (National Activity to Test Software) Project by NSF started in 1970
  - EISPACK (1972) and LINPACK (1978)
  - Standardization of level 1 BLAS (Basic Linear Algebra Subprograms) in 1979
  - Development of LAPACK, LAPACK2, and ScaLAPACK by NSF and DARPA during 1987-1995
  - PARASOL (An Integrated Programming Environment for Parallel Sparse Matrix Solvers) since 1996
  - SciDAC (Scientific Discovery through Advanced Computing) Program started in 2001 by DoE
    (Development of hardware/software infrastructure for terascale computing)
Features of the Project

- Started as a $2M and 5-year national project since Nov. 2002
- Complete survey of domestic and overseas research projects
  - Cooperation with other projects
  - Investigate problems with existing libraries
  - Refinement of software specification
- Development
  - Select and evaluate target architectures (need to predict mainstreams in 2007)
  - Fast prototyping of core components (need feedbacks)
  - Start with replacement of original libraries used in real applications
- Primary Targets:
  - Portable object-oriented implementation of the following libraries:
    - Parallel eigensolvers
      - CG type algorithms (selected eigenpairs for physical applications)
      - QR methods (general purpose)
    - Parallel linear solvers
      - Iterative solvers (for FDM and FEM)
      - Direct solvers (general purpose, real/complex, symmetric/non-symmetric, dense/band/sparse)
    - Parallel fast integral transforms
      - Fast Fourier transforms (general purpose)
      - Fast Legendle Transform (climate studies) etc.
- Distribution
  - Distribution via the network
  - Publication of manuals from major publisher
Core Research Fields

- Eigensolvers
  - Akira Nishida (Tokyo Univ.)
    - CG (conjugate gradient) type eigensolvers for large sparse eigenproblems and their parallel implementation.

- Linear solvers
  - Hidehiko Hasegawa (Tsukuba Univ.)
    - Development of iterative linear solvers
  - Kengo Nakajima (RIST)
    - Applicational fields
  - Hisashi Kotakemori (JST)
    - Development
  - Akihiro Fujii (Tokyo Univ. Doctoral candidate)
    - Parallel and vector implementation of AMG preconditioned CG method
  - Tomohiro Sogabe (Tokyo Univ. Doctoral candidate)
    - Studies on iterative solvers. Proposed BiCR type method.
  - Kuniyoshi Abe (Gifu Shotoku Gakuen Univ.)
    - Joint researcher with S. L. Zhang on product type iterative solvers
  - Shao-Liang Zhang (Tokyo Univ.)
    - Studies on iterative solvers. Proposed GPBiCG (product type iterative solver).
  - Shoji Ito (Tsukuba Univ.)
    - Development of direct solvers
  - Koh Hashimoto (Tokyo Univ.)
Core Research Fields (2)

- Fast integral transforms
  - Reiji Suda (Tokyo Univ.)
    - Fast legendre transform for spherical climate analysis
  - Daisuke Takahashi (Tsukuba Univ.)
    - Development of optimized parallel FFT
  - Akira Nukada (JST)
    - Development of optimized parallel FFT
- Parallel and portable implementation
  - Akira Nishida
  - Reiji Suda
  - Hidehiko Hasegawa
  - Kengo Nakajima
  - Tamito Kajiwara (JST)
  - Daisuke Takahashi
  - Akira Nukada
  - Akihiro Fujii
  - Yuichiro Hourai (Tokyo Univ. PhD candidate)
## Schedule

<table>
<thead>
<tr>
<th>Fiscal Year</th>
<th>2002 (5 months)</th>
<th>2003</th>
<th>2004</th>
<th>2005</th>
<th>2006</th>
<th>2007 (7 months)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Facilities</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Survey of</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Applications</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Survey of</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>software</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>engineering</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Survey of</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>hardware</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>technologies</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Algorithms</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Programming</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>model</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Implementation</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>and verification</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Tutorials</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Target (1): Architectures and Systems

- Survey of trends and direction of hardware technologies
  - Trends of computer architectures
    - Higher density and lower power
      - E.g. IBM Blue Gene/L: 130 thousand CPU - 180TFLOPS,
      - E.g. Fujitsu BioServer
    - Symmetric multithreading
      - IBM Power, Sun UltraSPARC, Intel Pentium & Itanium, etc.
  - Higher parallelism in every level of architecture
  - It becoming more important to optimize performance of the libraries, while designing them growing more complex
Current Status:
Architectures and Systems

- Predict computing environment to be available in 5 years
  - Up-to-date facilities to be updated every year
  - Current facilities of SSI Project
    - Shared memory programming environment: SGI Altix 3700
      (Intel Madison 1.3GHz × 32, Linux OS. 32GB main memory)
    - Vector processing environment: NEC SX-6
    - Cluster computing environment: Dual Intel Xeon 2.8GHz server
      x 16, GbE interconnect
    - 10GbE enabled networking environment
      (Cisco C6509)
  - Most of major architectures have been covered

- Portability
  - Portability can be tested easily on the SSI environment by the developers
Current Status:
Architectures and Systems (2)

- SGI Altix 3700
- NEC SX-6i
- Sun Fire 3800
- Sun StorEdge T3
- HyperTransport Interconnected Opteron Cluster
- InfiniBand Interconnected Itanium3 Cluster
- GbE or 10GbE LAN
- To GbE (10GbE) WAN
- Cisco Router C6509
- To Desktops

---

Note: The diagram shows the network architecture with different systems connected through various networks such as GbE, 10GbE, and WAN.
Target (2):
Algorithms

- Promotion of fundamental studies
  - Promotion of fundamental studies by the members (research meetings)
  - Provide up-to-date computing environment for joint researchers
- Support porting of existing libraries written by the members to the new computing environment
  - Planning to develop a new libraries based on a book “Numerical Libraries in Fortran 77” published by Maruzen Co.,Ltd. by Hasegawa et al.
  - NEDO APC automatic parallelizer developed has been implemented on our environment.
    - Automatically add OpenMP adaptives
- Fast release. Get feedbacks from beta users
  - A home page http://ssi.is.s.u-tokyo.ac.jp/ has been opened
  - Cooperation with AIST PHASE project http://phase.hpcc.jp/, etc.
- Lightweight libraries with minimum functions for large scale problems
  - Keep balance with oo overheads and performance
    - OO interface + primitive APIs
- Publish detailed documents
  - Easy to use
Current Status: Algorithms

- Eigensolvers (CG Type)
  - Solve minimum eigenvalue of generalized eigenproblem on real symmetric matrices
    \[ Ax = \lambda Bx \]
  or maximum eigenvalue of the equivalent eigenproblem
    \[ Bx = \lambda Ax, \quad \lambda = 1/\lambda \]
  - Minimize Rayleigh quotient
    \[ \phi(x) = \frac{x^TBx}{x^TAx} \]
    using that the most ascending direction is
    \[ \frac{g(x) + \lambda g(x)}{g(x)^Tg(x)} = \frac{2(Bx - \lambda Ax)}{x^TAx} \]
    by solving conjugate gradient method with the above coefficient as \( i \)
    \[ x_{i+1} = x_i + \alpha_i \beta_i \]
    \[ p_i = -g_i + \alpha_i \beta_i p_{i-1}, \quad \beta_i = \frac{g_i^T \beta_{i-1}}{g_i^T \beta_{i-1}} \]
  - Theoretically \( O(n) \) complexity
Current Status: Algorithms (2)

- Eigensolvers
  - CG type methods
    - AMG preconditioned CG solvers for eigenproblems by Knyazev and Argentati (2003) (See Figures)
    - ILU preconditioned CR solver by Suetomi and Sekimoto (1989)
Current Status: Algorithms (3)

- **Parallel AMG preconditioned CG method**

- **Smoothed Aggregation MG**
  - Solution of $Ax=b$
  - Algebraic multigrid method
    - Generate restricted matrix using vertex sets
    - named aggregates generated the coefficient matrix
  - Iteration number does not depend problem size
  - Robust convergence even with anisotropic problems
  - Cancel the convergence problem with MGCG by Tatebe and Oyanagi

- **Parallelization of direct linear solver**
Current Status: Algorithms (4)

- Linear solvers
  - Iterative solver (Bi-CR type method)
  - S. Li, Zhang, T. Sogabe, Bi-CR method for solving large nonsymmetric linear systems, the 2003 International Conference on Numerical Linear Algebra and Optimization, October 7-10, 2003 (Invited Talk)

\[
\begin{align*}
\mathbf{x}_n &= \mathbf{x}_0 + \mathbf{z}_n, \quad \mathbf{z}_n \in K_n(A; \mathbf{r}_0) \\
\mathbf{r}_n &= \mathbf{r}_0 - A\mathbf{z}_n, \quad \mathbf{r}_n \in K_{n+1}(A; \mathbf{r}_0)
\end{align*}
\]

**CG:** \( \min \| \mathbf{r}_n \|_{A^{-1}} \) \quad **CR:** \( \min \| \mathbf{r}_n \| \)
Current Status: Algorithms (5)

- Replace CG in Bi-CG with more stable CR algorithm
- Tested with Toeplitz matrices and some Matrix Market problems
- Derived CRS, BiCRSTAB, or GPBiCR which corresponds to CGS, BiCGSTAB, and GPBiCG
Current Status: Algorithms (6)

- Fast integral transforms
  - Joint studies with researchers in the field of weather forecast and earth hydrodynamics
- Main results
  - Efficient implementation of parallel FFT algorithms in a (multiprocessor) node
  - In-place FFT algorithm
    - Less memory size
    - Need bit-reverse process
    - Implemented on Itanium server (NEC AzusA)
    - 2.9Gflops with 8PEs (12.4% of peak performance)
  - Radix-8 FFT Kernel for Multiply-add Instructions
Target (3): Software and Implementations

• Provide general-purpose, easy-to-use software infrastructure
• Surveys of status and directions of programming technologies
  • Scalability
    • MPI
      ▪ Standard for message passing on distributed memory architectures
    • Co-Array Fortran
      ▪ Developed by Cray (for T3E)
      ▪ Open64 based implementation available from Rice Univ.
      ▪ Requested for the next version of Fortran
    • HPF(JA)
      ▪ Developed by HPFPC and Earth Simulator Center
  • Global Arrays
    ▪ API based
    ▪ Easy to implement
  • Object oriented programming concepts
    • Access to objects via APIs only
    • OO concepts supported language: C++ and Fortran 9x/200x
Concluding Remarks

- Performance of computers to keep rapid progress
  - Parallel simulation technology is to be used in wider areas with popularization of distributed
- Domestic effort for software infrastructure for massively parallel applications will be helpful to
  - Produce intellectual property
    - Design for long term use at home and overseas
    - Suppose to be used by researchers working at supercomputing centers and research laboratories as a practical components
  - Publish official manual on the algorithms and their usage
  - Target a standard high quality library
- Create new technical infrastructure
  - Distribution of high quality common components for scientific simulation
  - Establishment of reliable designing/evaluating methodologies via feedbacks from users