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

Akira Nishida
Department of Computer Science, University of Tokyo
JST CREST

98th IPSJ SIGHPC Meeting

Motivation

- Emergence of large scale scientific simulations in various fields
- Development of numerical libraries in Japan
  - Mainly developed in supercomputing centers (on mainframes and vector supercomputers) in 1980s
  - Cooperation with vendors (E.g. Fujitsu SSL II)
- Development in US
  - ScaLAPACK (with BLAS and LAPACK), PETSc, Aztec, etc.
    - Developed and used in national laboratories
    - Standardized and modularized
    - Run on parallel computing environments
    - Distributed via WWW (netlib etc.) since 1990s (also mirrored in Japan)
- Demands for reliable and portable parallel numerical libraries as social infrastructure

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)

Brief History of Basic Numerical Libraries (2)

- Projects in Japan
  - Basic numerical libraries
    - Internal use in national supercomputing centers

      Program System for Statistiacal Analysis with Least-Squares Fitting (T. Nakagawa and Y. Oyanagi et al., 1976-1982)
    - Offline distribution

      A series of books by K. Murata, T. Oguni, H. Hasegawa published from Maruzen Co.,Ltd. with floppy disks
    - <u>No major national projects for development of parallel numerical libraries</u>
  - Parallel processing
    - Real World Computing (RWC) Project by MITI (M. Sato, Y. Ishikawa, T. Kudo et al.)
      - OBPLib: Object oriented library for scientific computing on distributed memory architectures
      - Omni OpenMP Compiler: Free OpenMP compiler for shared memory parallel architectures

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:
    - Parallel eigensolvers
      - QR algorithms (general purpose, real/complex, symmetric/non-symmetric)
      - Lanczos/Arnoldi, Davidson methods (selected eigenpairs for physical applications)
    - Parallel linear solvers
      - Direct solvers (general purpose, real/complex, symmetric/non-symmetric, dense/band/sparse)
      - Iterative solvers (for FDM and FEM)
    - Parallel fast integral transforms
      - Fast Fourier transforms (general purpose)
      - Fast Legendle Transform (climate studies) etc.
    - Portable object-oriented implementation
- Distribution
  - Distribution via network
  - Publication of manuals from major publishers

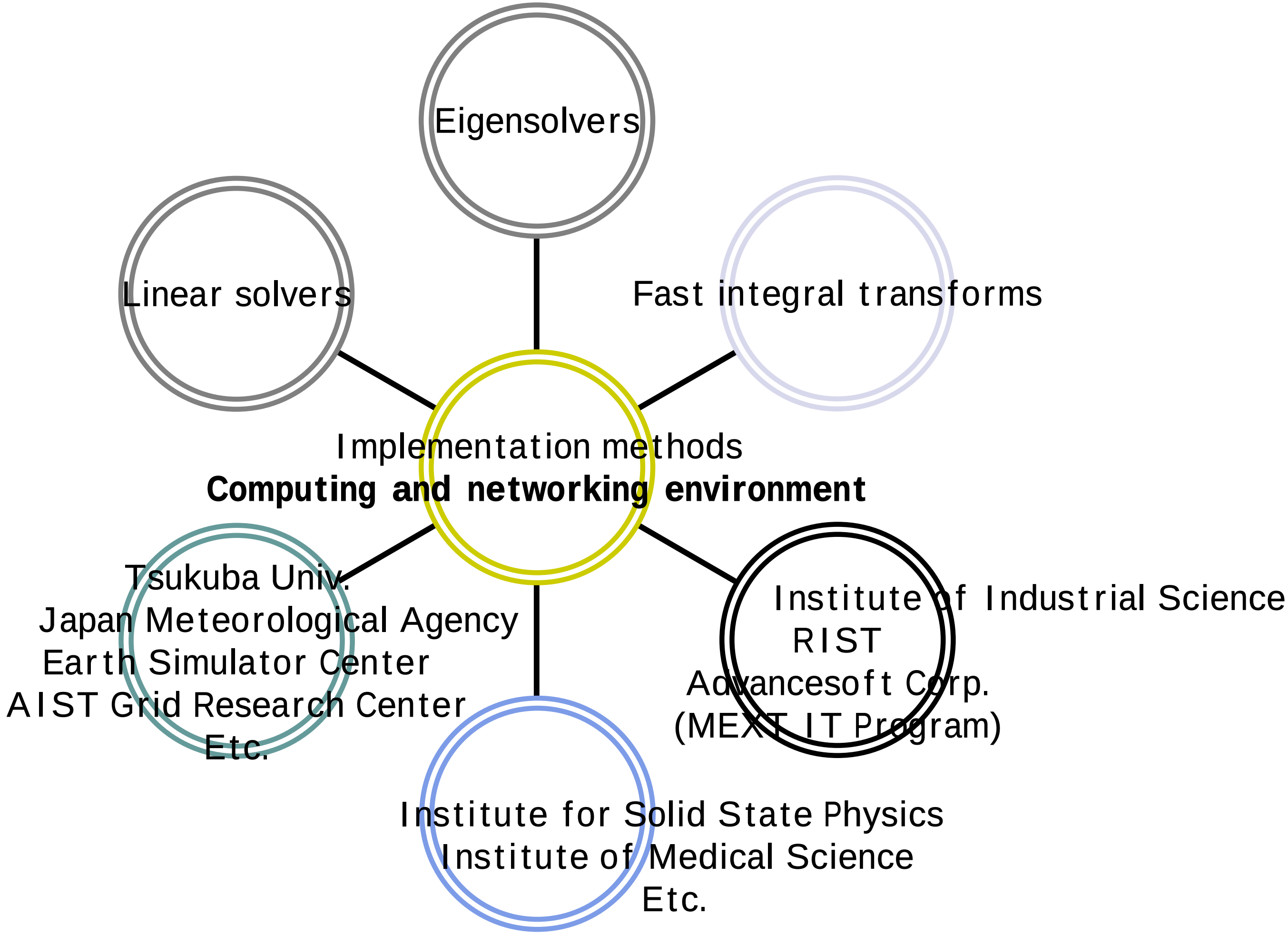
Core Research Fields

- Eigensolvers
  - Akira Nishida (Tokyo Univ.)
    - Eigensolvers for large sparse eigenproblems and their parallelization.
- Linear solvers
  - Hidehiko Hasegawa (Tsukuba Univ.)
    - Development of direct/iterative linear solvers
  - Shao-Liang Zhang (Tokyo Univ.)
    - Studies on iterative solvers. Proposed GPBiCG (product type iterative solver).
  - Kengo Nakajima (RIST)
    - General purpose solver for finite element problems
  - **Kuniyoshi Abe** (Gifu Shotoku Gakuen Univ.)
    - Joint researcher with S. L. Zhang on product type iterative solvers
  - Shoji Ito (Tsukuba Univ.)
    - Development of direct solvers
  - Koh Hashimoto (Tokyo Univ.)
    - Joint researches with S. L. Zhang. Studies on machanical systems.
  - 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.

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 (Tokyo Univ. Doctoral candidate)
    - Development of optimized parallel FFT
- Parallel and distributed portable implementation
  - Akira Nishida
  - Reiji Suda
  - Hidehiko Hasegawa
  - Kengo Nakajima
  - Akira Nukada
  - Akihiro Fujii
  - Yuichiro Hourai (Tokyo Univ. Doctoral candidate)
    - Parallel distributed computation, optimization of broadcast communications on tree-structured networks

Organization



Schedule

Fiscal Year	2002 (5 months)	2003	2004	2005	2006	2007 (7 months)
Facilities						
Survey of Applications						
Survey of software engineering						
Survey of hardware technologies						
Algorithms						
Programming model						
Implementation and verification						
Tutorials						

Target (1): Architectures and Systems

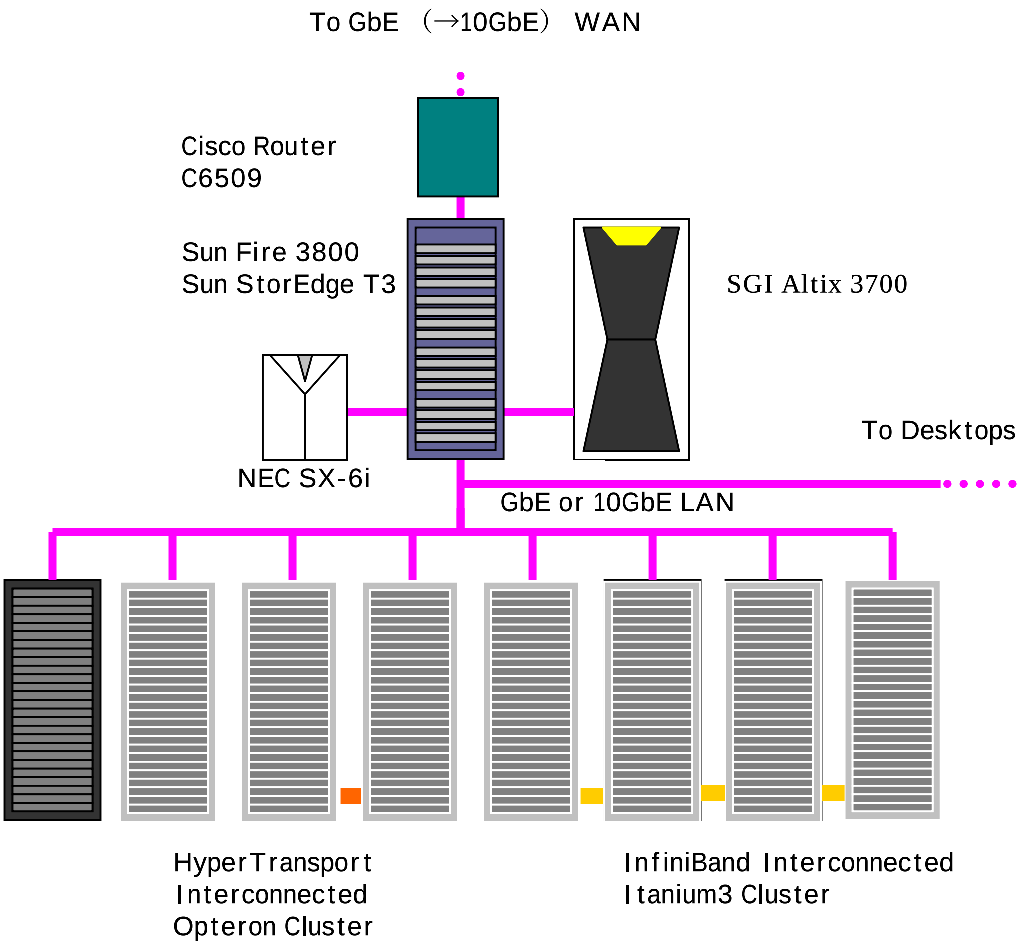
- Survey of trends and direction of hardware technologies
  - Trends of computer architectures
    - Higher density and lower power
      - E.g. IBM Blue Gene/L: 130 thousand CPU - 180TFLOPS,
      - E.g. Fujitsu BioServer
    - Symmetric multithreading
      - IBM Power, Sun UltraSPARC, Intel Pentium & 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

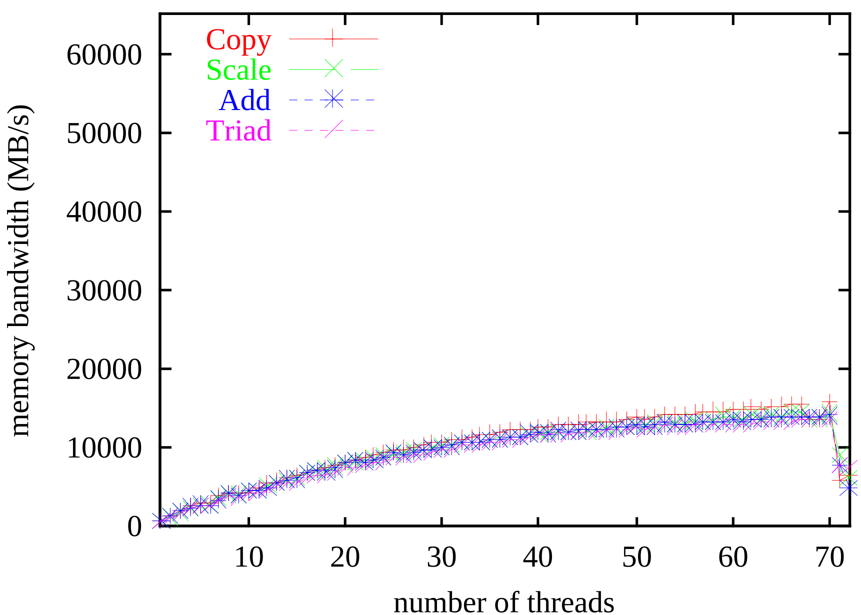
- Predict computing environment to be available in 5 years
  - Up-to-date facilities to be updated every year
  - Current facilities of SSI Project
    - Shared memory programming environment: SGI Altix 3700 (Intel Madison 1.3GHz × 32, Linux OS. 32GB main memory)
    - Vector processing environment: NEC SX-6i
    - 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

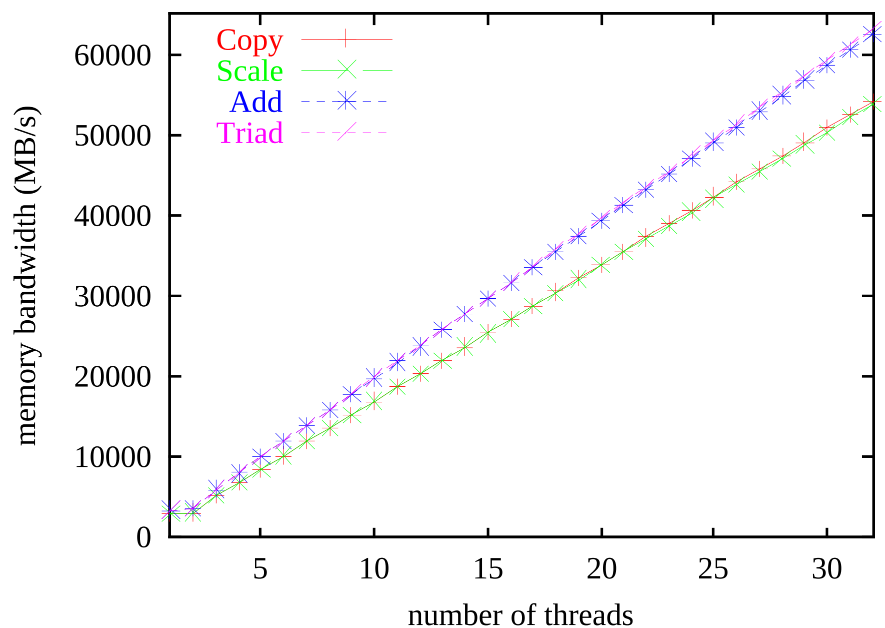
Current Status: Architectures and Systems (2)



Current Status: Architectures and Systems (3)

- Shared memory computer SGI Altix 3700
  - Memory bandwidth performance compared with Sun Fire 15k of UltraSPARC III 900MHz x 72, Solaris 8 , with STREAM benchmark, 1.8GB data





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 pallelizer 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 mimimum functions for large scale problems
  - Keep balance with oo overheads and performance
    - OO interface + primitive APIs
- Publish detailed documents
  - Easy to use

Current Status: Algorighms

- Eigensolvers (CG Type)
  - Sove mimimum eigenvalue of generalized eigenproblem on real symmetric matrices

    $Ax = \lambda Bx$

    or maximum eigenvalue of the equivalent eigenproblem

    $Bx = \mu Ax, \ \mu = 1/\lambda$
  - Minimize Rayleigh quotient

    $\mu(x) = x^T Bx \ / \ x^T Ax$

    using that the most ascending direction is

    $\nabla \mu(x) \equiv g(x) = 2(Bx - \mu Ax) \ / \ x^T Ax$

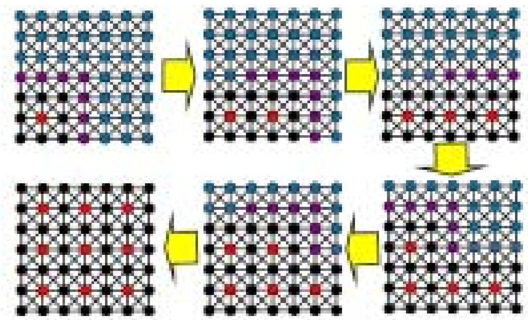
    by solving conjugate gradient method with the above coefficient as α_i

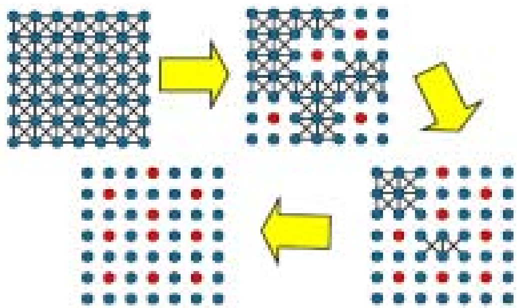
    $x_{i+1} = x_i + \alpha_i p_i,$

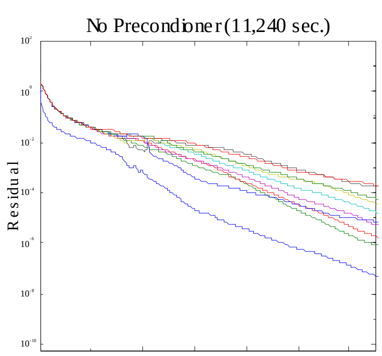
    $p_i = -g_i + \beta_{i-1} p_{i-1}, \ \beta_{i-1} = g^T_i g_i \ / \ g^T_{i-1} g_{i-1}$
  - Theoretically O(n) complexity

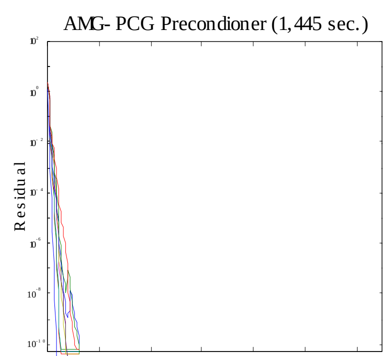
Current Status: Algorighms (2)

- 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)

- Linear solvers
  - Iterative solver (Bi-CR type method)
    - S. -L. 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)

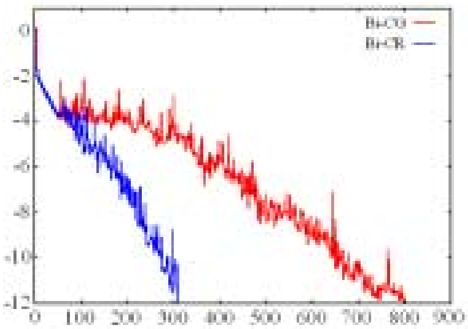
$$\mathbf{x}_n = \mathbf{x}_0 + \mathbf{z}_n, \quad \mathbf{z}_n \in K_n(\mathbf{A}; \mathbf{r}_0)$$

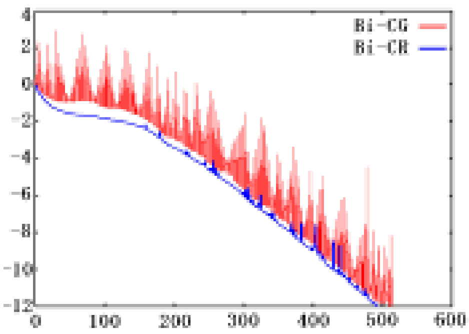
$$\mathbf{r}_n = \mathbf{r}_0 - \mathbf{A}\mathbf{z}_n, \quad \mathbf{r}_n \in K_{n+1}(\mathbf{A}; \mathbf{r}_0)$$

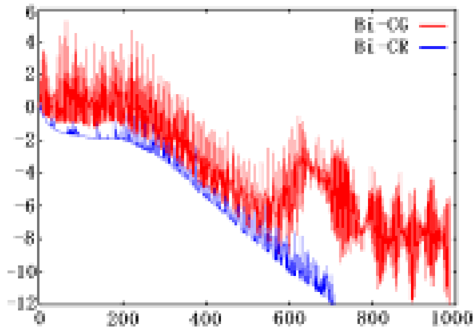
CG: $\min \|\mathbf{r}_n\|_{\mathbf{A}^{-1}}$ CR: $\min \|\mathbf{r}_n\|$

Current Status: Algorithms (4)

- 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

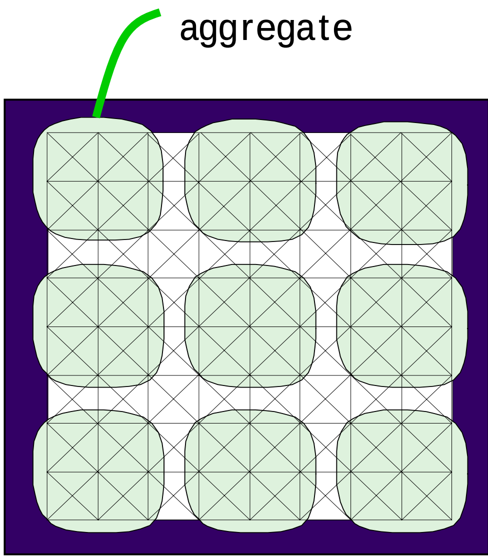






Current Status: Algorithms (5)

- Parallel AMG preconditioned CG method
  - . Fujii, A. Nishida and Y. Oyanagi. Improvement and Evaluation of Smoothed Aggregation MG for Anisotropic Problems. In *Proceedings of Symposium on Advanced Computing Systems and Infrastructures*, pp.137-144, 2003.
  - A. Fujii, A. Nishida and Y. Oyanagi. Parallel AMG Algorithm by Domain Decomposition. *IPSJ Transactions on Advanced Computing Systems*, Vol. 44, No.SIG 6 (ACS 1), pp.9-17, 2003.
- 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
  - H. Hasegawa, Parallelization of Direct Liear Solver for Banded Matrices using OpenMP. *IPSJ Transactions on Advanced Computing Systems*, to appear.

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
    - A. Nukada, A. Nishida and Y. Oyanagi. New Radix-8 FFT Kernel for Multiply-add Instructions. In *Proceedings of High Performance Computing Symposium 2004*, pp.17-24.
    - A. Nukada, A. Nishida and Y. Oyanagi. Parallel Implementation of FFT Algorithm on Distributed Shared Memory Architecture and its Optimization. *IPSJ Transactions on Advanced Computing Systems* Vol. 44, No. SIG 6 (ACS 1), pp.1-8, 2003.
    - 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

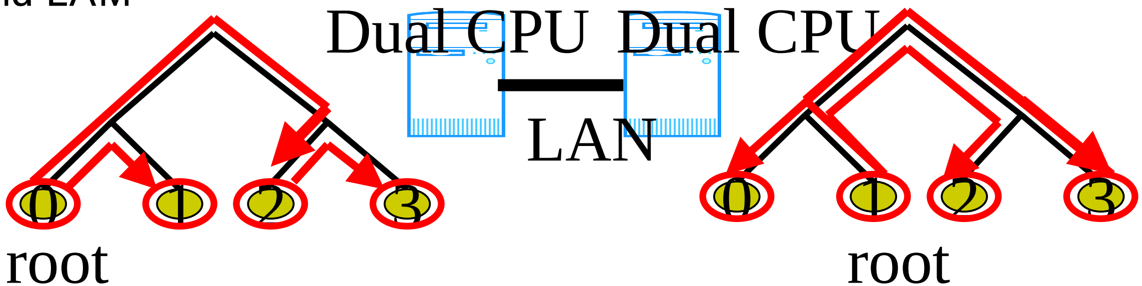
Target (3): Software and Implementations

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

Current Status: Software and Implementations

- Parallel Implementation
  - Joint research with Tokyou Univ. COE project "Information Science and Technology Strategic Core"
    - **Optimization of communication on cluster/grid environments**
      - Y. Hourai, A. Nishida, and Y. Oyanagi, "Optimal Broadcast Scheduling on Tree-structured Networks", IPSJ Transactions on Advanced Computing Systems, to appear.
  - Traditional implementation of broadcast communications (MPICH-Score and LAM)
    - Fix or ignore network topology
    - (Most implementations just shift the schedule of process ID 0 for other processes)

    E.g. Performance significantly changes when altering broadcasting root with naïve implementation of binary tree based algorithms
  - Optimization considering parameterized bandwidth and latency ... NP hard
  - Reduction of redundancy using isomorphism ... Faster broadcast than MPICH-Score and LAM



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