「シミュレーション技術の革新と
実用化基盤の構築」研究領域

大規模シミュレーション向け基盤ソフトウェアの開発

平成14年度－19年度

中央大学理工学研究所/
東京大学大学院情報理工学系研究科
西田 晃

研究実施の概要

- 大規模シミュレーションの基礎となる数値計算ライブラリ
  - インターネット登場以前 ～
    - それぞれの応用分野で個別に開発・実装
    - 大型計算機センター内での公開，商用ライブラリ
  - インターネットの普及
    - コンピュータの処理能力は18ヶ月で2倍に（Moore の法則）
    - 通信網の帯域幅は6ヶ月で2倍に（Gilder の法則）
    - デスクトップPCからスーパーコンピュータまで，多様な資源がネットワーク上で共有できるようになった
    - オープンな環境で開発が可能に （GNU, Linux, グリッドコンピューティング…）
  - 並列処理技術の進展
    - 大規模化が予想される今後の計算環境に対応したスケーラブルなソフトウェア共通基盤の整備が急務

研究実施の概要

- 平成14年度より JST CREST プロジェクトとして採択
- 広く使ってもらうには
  - 今後の普及が予想される計算機環境を想定した開発
  - 移植性の確保
  - 実装手法に関する研究（コンピュータサイエンス）の重要性
    - オブジェクト指向に基づくプログラミングインタフェースの採用
  - 各分野についてソースコードを含むソフトウェアをインターネット上に無償公開，広範なユーザの要望を取り入れつつ更新
- まずは主要な分野から国産ライブラリを
  - 反復解法，高速関数変換，数値ライブラリを呼び出すためのミドルウェアや，これらの効果的な計算機上への実装手法を対象に
  - 並列化に適したアルゴリズムを用いたスケーラブルなライブラリを実現
- 研究機関や計算機ベンダとの共同研究
  - 平成16年度より IBM Watson 研究所と，また平成18年度より地球シミュレータセンター共同プロジェクトの一環として，外部機関と共同研究
  - 多様な計算機アーキテクチャに対応

研究実施の概要

- 実装手法
  - 高性能計算のためのハードウェア技術
    - 今後の普及が予想される計算機技術とその性能評価に関する研究
    - 研究開発環境の整備
    - 研究機関や計算機ベンダとの共同研究
  - その上で高い性能を発揮することのできるソフトウェア技術
    - 可搬性・記述性に関する研究
    - ライブラリの記述手法，記述言語の双方について研究
  - ライブラリ設計への応用
    - 並列数値ライブラリの利用を容易にするためのミドルウェア SILC の開発
    - オブジェクト指向型インタフェースの採用
    - 性能自動最適化技術の採用

研究実施の概要

- 反復解法
  - 流体シミュレーションや構造解析など，広汎な需要
  - 反復解法及びその前処理手法についての研究
    - 高並列な環境での使用に耐えるスケーラブルなアルゴリズムの設計
    - マルチレベルな解法として，代数的マルチグリッド（AMG）法を反復解法の前処理として研究・実装
  - 多様な反復解法，前処理，及び疎行列格納形式に対応した並列反復解法ライブラリ Lis（A Library of Iterative Solvers for Linear Systems）を開発し，ソフトウェアを公開
    - 逐次環境から大規模並列環境まで，オブジェクト指向型インタフェースの採用により単一のプログラムで対応可能

研究実施の概要

- 高速関数変換
  - 高速フーリエ変換ライブラリを中心に，実際の計算環境において高い実効性能を得ることのできるソフトウェアを開発
  - 高速フーリエ変換ライブラリ FFTSS を公開
    - 多様な FFT カーネルルーチンを搭載
    - 計算環境に応じて最適なカーネルセットを選択する自動最適化機能を搭載
    - 商用版を含む既存のライブラリと比較して高速な計算を実現
    - 並列環境への対応
  - Blue Gene/L，地球シミュレータ上での性能最適化

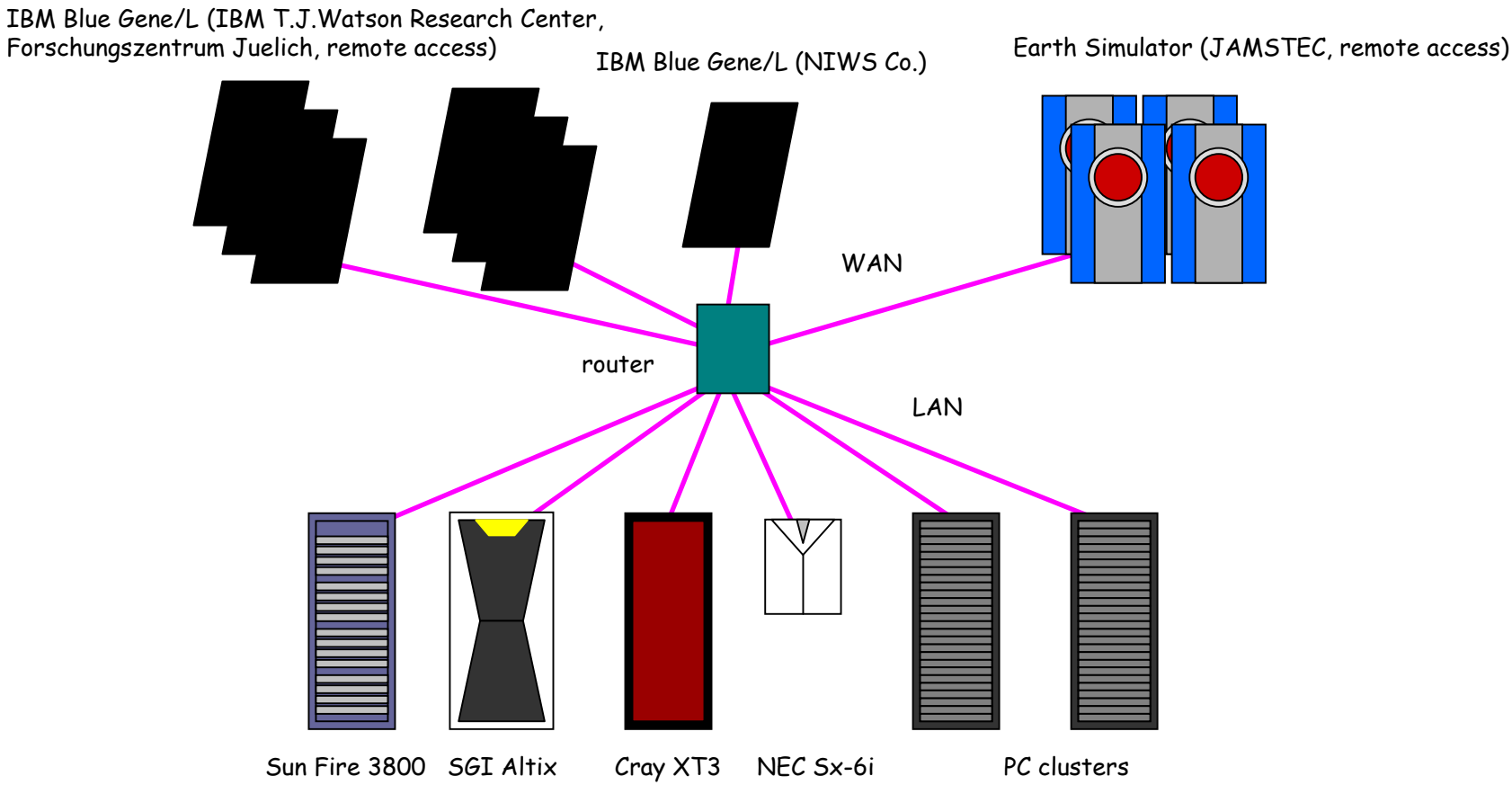
参加メンバー

- **研究代表者**
  - 西田 晃 (東京大学／中央大学)
- **実装手法研究グループ**
  - 西田 晃 (東京大学／中央大学)
  - 長谷川 秀彦 (筑波大学)
  - 須田 礼仁 (東京大学)
  - 中島 研吾 (東京大学)
  - 高橋 大介 (筑波大学)
  - 小武守 恒 (科学技術振興機構／東京大学)
  - 梶山 民人 (科学技術振興機構／東京大学)
  - 額田 彰 (科学技術振興機構／東京大学)
  - 藤井 昭宏 (工学院大学)
- **反復解法研究グループ**
  - 西田 晃 (東京大学／中央大学)
  - 長谷川 秀彦 (筑波大学)
  - 張 紹良 (名古屋大学)
  - 中島 研吾 (東京大学)
  - 阿部 邦美 (岐阜聖徳学園大学)
  - 伊藤 祥司 (理化学研究所)
  - 藤井 昭宏 (工学院大学)
  - 小武守 恒 (科学技術振興機構／東京大学)
  - 曽我部 知広 (名古屋大学)
- **高速関数変換グループ**
  - 須田 礼仁 (東京大学)
  - 高橋 大介 (筑波大学)
  - 額田 彰 (科学技術振興機構／東京大学)

実装手法

- 目標
  - 今後の普及が予想されるハードウェア技術を予測
  - その上で高い性能を発揮することのできるソフトウェアを開発
- 開発環境
  - Linux OS の積極採用
    - 移植性，オープン性を評価
      - ≫ SGI Altix, Cray XT3, IBM Blue Gene, IBM OpenPower, Sony PS3
      - ≫ PC クラスタ
  - コストパフォーマンスの重視
    - 将来的な普及率を念頭に
      - ≫ SGI Altix（2003年7月導入．Intel Itanium ベース．SGI はその後 MIPS から完全移行．）
      - ≫ Cray XT3（2005年3月導入．AMD Opteron ベース．Red Storm の商用化．Cray はその後 XT4, XT5 を発表．）
    - 消費電力
      - ≫ IBM Blue Gene（2003年11月 IBM Watson 研究所と共同研究開始． IBM PowerPC 440 搭載．）
      - ≫ マルチコア化の進展（SGI Altix で共有メモリ環境に対応済み）
  - 疎行列処理等，局所性の小さい計算への対応
    - 開発用ベクトル計算機（NEC SX-6i）を導入
    - 平成18年度より地球シミュレータ共同プロジェクト（海洋研究開発機構）の一環として，大規模並列環境に対応

実装手法


IBM Blue Gene/L (IBM T.J.Watson Research Center, Forschungszentrum Juelich, remote access)
IBM Blue Gene/L (NIWS Co.)
Earth Simulator (JAMSTEC, remote access)
WAN
router
LAN
Sun Fire 3800
SGI Altix
Cray XT3
NEC Sx-6i
PC clusters

実装手法

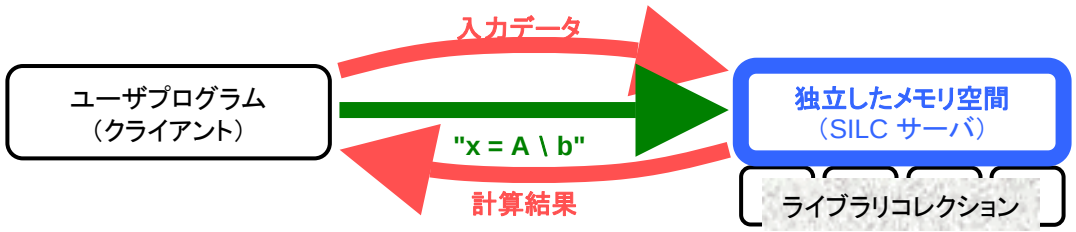
- 実装方針
  - 可搬性とともに，利用者が効率的に処理を記述できるような枠組みを確保する必要
    - 実装の多相性をユーザに意識させない仕組みの採用
      - ライブラリの記述手法，記述言語の双方について研究し，各分野で反映
    - オブジェクト指向型インタフェースの採用
    - 性能の自動最適化
      - 高速関数変換においては性能の自動最適化機能を備えた FFT ライブラリを実装
    - 多言語インタフェースの実装
      - ライブラリは C 言語で記述
      - Fortran インタフェースの実装．他言語インタフェースにも対応可能（C++など）
  - 数値計算ライブラリを簡易な記法により利用するためのフレームワーク
    - SILC（Simple Interface for Library Collections）を提案
    - スクリプト言語への拡張を含め，2件の特許出願

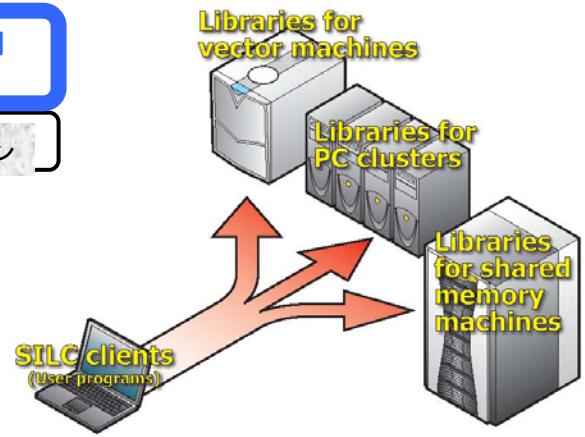
実装手法

- SILC (Simple Interface for Library Collections)
  - 行列計算ライブラリは科学技術計算プログラムの作成に不可欠
  - 個々のライブラリは提供するインタフェース（API）を通じて利用される
    - 連立一次方程式Ax = b の求解の場合，ユーザはまず行列A とベクトルb をライブラリ固有のデータ構造で準備
    - 特定のライブラリ関数（求解ルーチン）を所定の引数の順序に従って呼び出す
    - 作成したユーザプログラムが特定のライブラリに依存
    - ライブラリ間のインタフェースには互換性がないことが多い
  - 計算環境に依存しない行列計算ライブラリインタフェースSILC（Simple Interface for Library Collections）を提案

実装手法

- SILC（概要）
  - 行列やベクトルなどの入力データは，ユーザプログラムから独立したメモリ空間に転送
  - 数式の形で与えられた計算指示は，適当なライブラリ関数の呼び出しに翻訳，独立したメモリ空間内で実行
  - ユーザプログラムからの要求に応じて計算結果がユーザプログラムのメモリ空間に戻される





実装手法

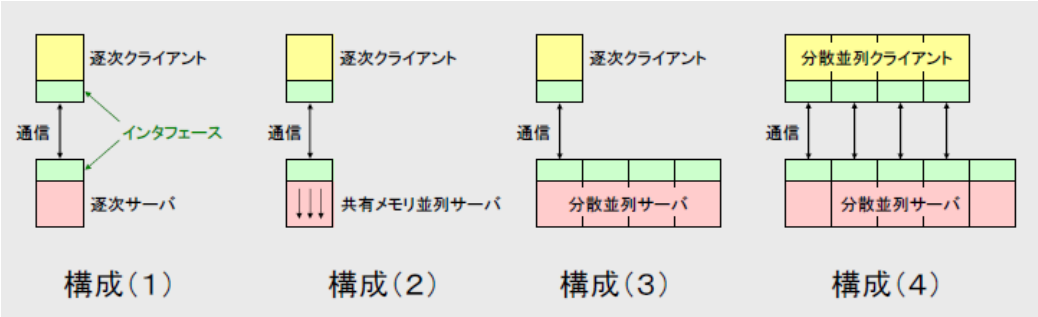
- SILC（インタフェース）
  - SILC のインタフェースを介して LAPACK の連立一次方程式求解ルーチンを呼び出すC プログラムの例

```
silc_envelope_t A, b, x;
/* 行列A とベクトルb の作成*/
SILC_PUT("A", &A);
SILC_PUT("b", &b);
SILC_EXEC("x = A ¥¥ b"); /* 求解*/
SILC_GET(&x, "x");
```

  - LAPACK のデータ構造を用いて行列A とベクトルb を作成した後, SILC が提供するSILC_PUT, SILC_EXEC, SILC_GET の3つのルーチンを介して求解ルーチンを呼び出す

実装手法

- SILC（実装）
  - 以下の4種のシステム構成を想定
    - (1) 逐次クライアント＋逐次サーバ
    - (2) 逐次クライアント＋共有メモリ並列サーバ
    - (3) 逐次クライアント＋分散並列サーバ
    - (4) 分散並列クライアント＋分散並列サーバ
  - ユーザプログラムはSILC サーバを構成する1つもしくは複数のプロセスに接続してデータ転送と計算指示
  - ユーザプログラムからサーバに転送されたデータは，サーバが備えるデータ再分散機構によりライブラリの要求するデータ分散方式に変換，複数のサーバプロセスに分散して保持
  - ユーザプログラムに返される計算結果は，データ再分散機構により転送前にユーザプログラムの要求するデータ分散方式に戻される



実装手法

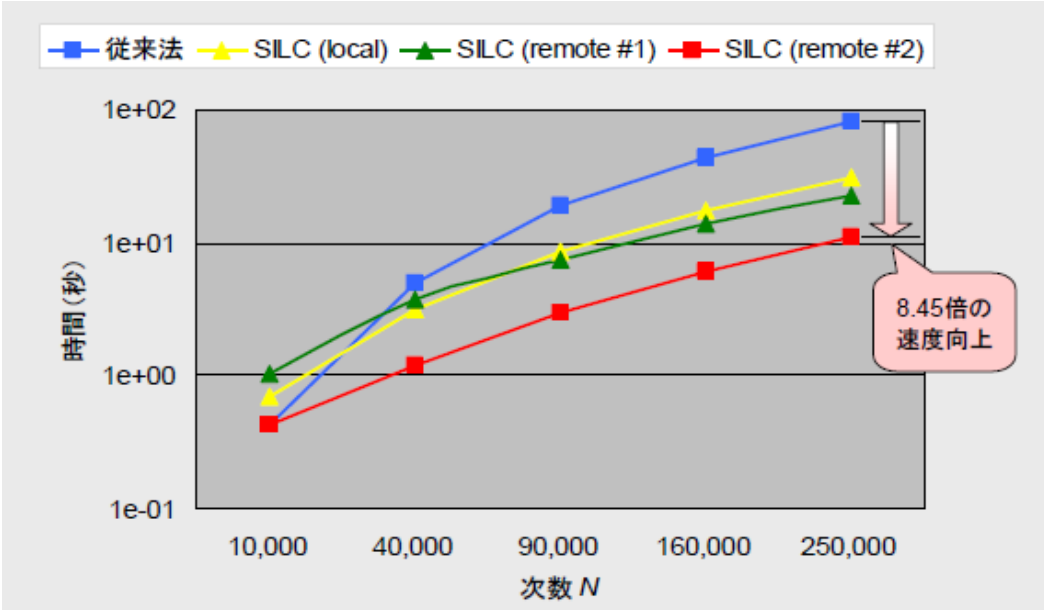
- SILC(性能評価)
  - 実現したシステムを通じて遠隔の高速な分散並列環境を利用することにより，従来法で書かれたユーザプログラムと比較して性能向上
  - 複数の計算環境の組み合わせについて，2次元拡散方程式の初期値問題を差分法で解いた結果を示す
  - 連立一次方程式の求解には反復解法ライブラリLis の前処理なし共役勾配法を用い，サーバ上で連立一次方程式の求解演算子をLis のMPI 版の呼び出しに翻訳するように設定

	ユーザプログラム	SILC サーバ
従来法	Xeon4 (1 PE)	–
SILC (local)	Xeon4 (1 PE)	Xeon4 (4 PEs)
SILC (remote #1)	Xeon4 (1 PE)	Xeon8 (8 PEs)
SILC (remote #2)	Xeon4 (1 PE)	Altix (16 PEs)

ホスト名	仕様
Xeon4	IBM eServer xSeries 335 (dual Intel Xeon 2.8 GHz, 1.0 GB RAM) ×4, Red Hat Linux 8.0, LAM/MPI 7.0
Xeon8	Xeon4 と同じ PC クラスタの別の8ノード
Altix	Intel Itanium2 1.3 GHz × 32基, メモリ 32 GB, Red Hat Linux Advanced Server 2.1, SGI MPI 4.4 (MPT 1.9.1)

実装手法

- SILC(性能評価)
  - 次数をN，反復回数をαとすると，通信量はO(N)，計算量はO(αN) であるため，反復回数のかかる問題であれは通信コストを加味しても遠隔の高速な並列計算機を利用した方が有利



実装手法

- SILC（スクリプト言語としての利用）
  - 当初既存の言語処理系と組み合わせた利用を想定
  - 単独では条件分岐や反復などの制御構文を含む計算処理を記述できない制約
  - SILCの命令記述言語を拡張し，制御構文を加えたスクリプト言語を定義
  - これを用いて書かれたユーザプログラムを解析し，演算処理と制御処理を分離して従来のSILC の枠組みで処理することにより，スクリプト化を実現
  - 以下のような対話型処理/バッチ処理が可能に

実装手法

```
# 三重対角行列 A とベクトル b を作る
n = 400
A = diag(2.0 * ones(n, 1)) - diag(ones(n-1, 1), 1) - diag(ones(n-1, 1), -1)
b = A * (-ones(n, 1))

# 連立一次方程式 Ax=b を CG 法で解く
rho_old = 1.0
p = zeros(n, 1)
x = zeros(n, 1)
r = b
bnrm2 = 1.0 / norm2(b)
iter = 1
while (iter <= n) {
  rho = r' * r
  beta = rho / rho_old
  p = r + beta * p
  q = A * p
  alpha = rho / (p' * q)
  r = r - alpha * q
  nrm2 = norm2(r) * bnrm2
  x = x + alpha * p
  if (nrm2 <= 1.0e-12) {
    break
  }
  rho_old = rho
  iter += 1
}

# 解 x をファイルに保存する
save "sol.mtx", x

# 反復回数を表示する
message "number of iterations:“
pprint iter
```

実装手法

- SILC（関連研究）
  - Trilinos
    - 行列計算ライブラリを C++ のクラスライブラリとして統合するための枠組みを提案
    - 各ライブラリのAPI は，(1)共通の行列とベクトルのデータ構造を用いること，(2)ライブラリの開発言語として C++ を採用し，解法のインタフェースを規定する共通の抽象クラスを継承することの2 点において統一
    - (3)共通のディレクトリ構造とコンパイル方法を採用し，開発者の異なるライブラリを同じ方法で構築可能なパッケージとしてまとめる
    - API の詳細はライブラリ毎に大きく異なる
  - Ninf-G
    - 遠隔の計算環境を利用するという点で分散型 SILC と関連
    - グリッド環境において遠隔手続き呼び出し(RPC)を実現するためのミドルウェア
    - 従来の関数呼び出しに似た記法で RPC を記述
  - Matlab
    - ライブラリプログラムを簡便に実行するためのスクリプト言語を提供
    - 可視化や GUI 作成のための機能を備え、対話利用を想定
    - 並列化については MIT，Interactive Supercomputing 社 による Star-P などの研究開発が進行中

反復解法

- 概要
  - 共役勾配法，共役残差法系の線形反復解法，固有値解法とその前処理手法について，多くの国際･国内会議で成果を発表
  - これらの解法及び多数の疎行列格納形式に対応したオブジェクト指向型の並列反復解法ライブラリ Lis（A Library of Iterative Solvers for Linear Systems）を開発･公開
  - 一般的な SMP PCクラスタ環境の他，NEC SX（地球シミュレータを含む），IBM Blue Gene，Cray XT などの特殊な計算環境にも対応

反復解法

- 反復解法ライブラリ Lis
  - 以下に，Lis で対応している解法(20種)，前処理手法(11種)，及び行列格納形式(11種)の一覧を示す

1.0.x	CG	1.1.0で追加	CR
	BiCG		BiCR
	CGS		CRS
	BiCGSTAB		BiCRSTAB
	BiCGSTAB(l)		GPBiCR
	GPBiCG		BiCRSafe
	Orthomin(m)		FGMRES(m)
	GMRES(m)		
	TFQMR		
	Jacobi		
	Gauss-Seidel		
	SOR		

1.0.x	Jacobi	1.1.0で追加	Crout版ILU
	ILU(k)		ILUT
	SSOR		Additive schwarz
	Hybrid		ユーザ定義前処理
	I+S		
	SA-AMG		
	SAINV		

Point	Compressed Row Storage
	Compressed Column Storage
	Modified Compressed Sparse Row
	Diagonal
	Ellpack-Itpack generalized diagonal
	Jagged Diagonal Storage
	Dense
	Coordinate
Block	Block Sparse Row
	Block Sparse Column
	Variable Block Row

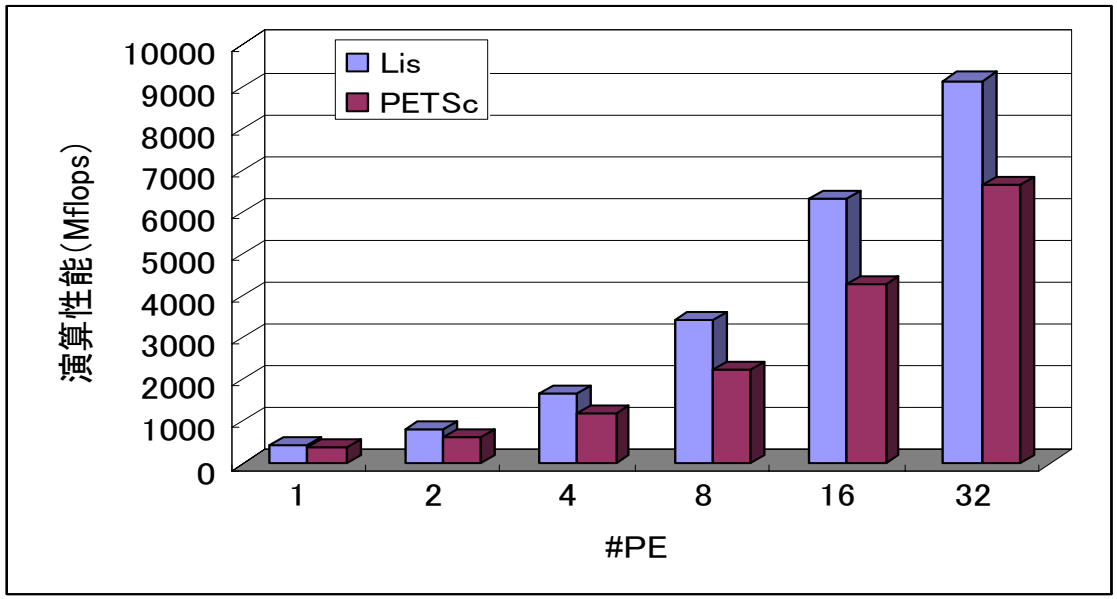
反復解法

- Lis を用いたプログラミング例

```
 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,
18: printf("iter = %d time = %e (p=%e
19: lis_finalize();
```

反復解法

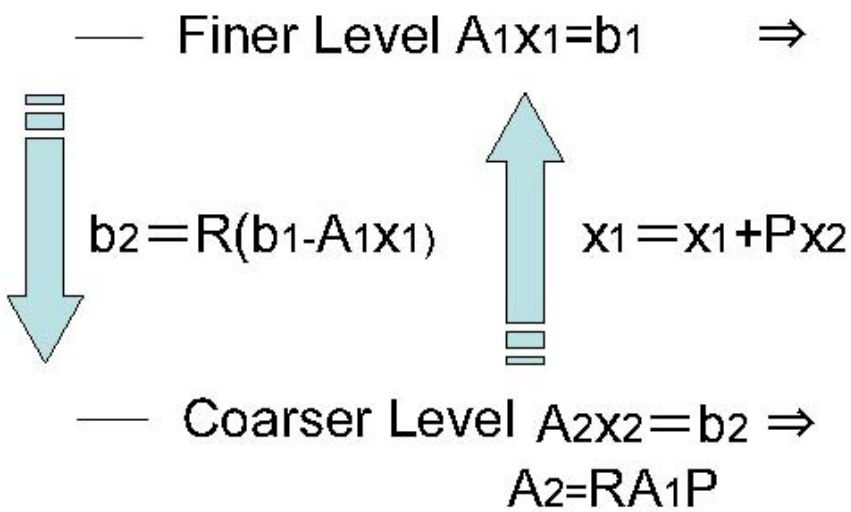
- SGI Altix（32ノード搭載）上での3次元 Poisson 方程式（次数100万元，非零要素数26,207,180）を用いた MPI 版共役勾配法での性能
- 比較対象はPETSc

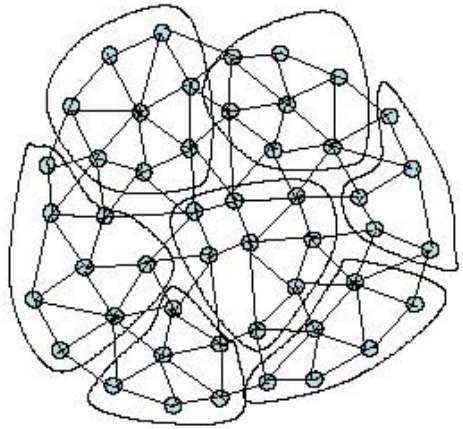


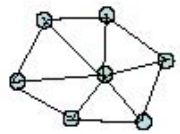
反復解法

- 代数的マルチグリッド前処理
  - 流体シミュレーションや構造解析などの応用分野から派生する連立一次方程式の反復解法及びその前処理手法を中心に，高並列な環境での使用にも耐え得るスケーラブルかつ信頼性の高いアルゴリズムについて研究
  - 大規模連立一次方程式の反復解法として，マルチレベルな解法が前処理として有効
  - 一般にマルチグリッド法は
    - 並列性に富む
    - 逐次計算の場合，問題行列の大きさを $n \times n$ とすると，うまく機能した場合収束までの計算量は $O(n)$
  - 代数的マルチグリッド（Algebraic Multigrid, AMG）法は問題行列を点枝接続行列と見なす
  - 各要素とその位置関係に関する情報のみを利用して誤差の高周波数成分の取り除かれた粗いレベルの行列を生成
  - 幾何的マルチグリッド法と同等の計算量で高速に問題を解くことができ，
    - 不規則な疎行列に対しても適用可能
    - 異方性問題に対しても有効
  - といった利点を持つ

反復解法

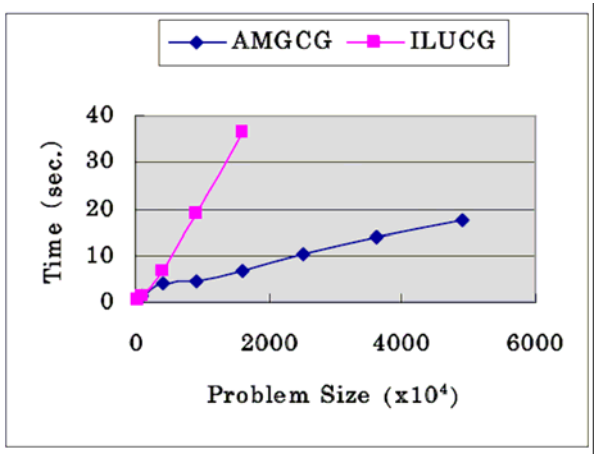

Finer Level $A_1x_1=b_1$ ⇒
$b_2=R(b_1-A_1x_1)$
$x_1=x_1+Px_2$
Coarser Level $A_2x_2=b_2$ ⇒
$A_2=RA_1P$





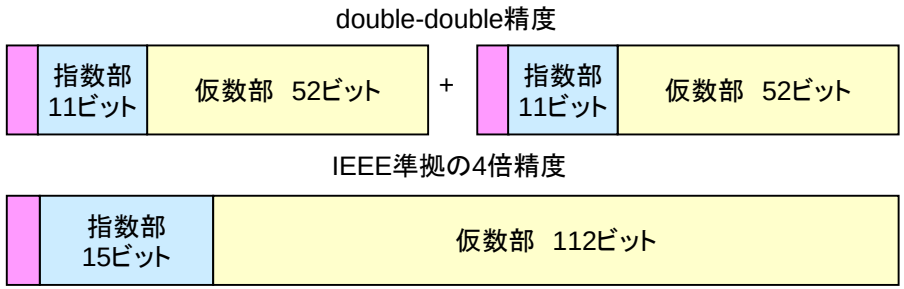
反復解法

- 代数的マルチグリッド前処理
  - 本研究では，algebraic multigrid 法を前処理として用いた共役勾配法（SA-AMGCG 法）とその効率的な並列実装手法を提案
  - ICCG 法（Localized ILU 前処理付共役勾配法）との比較から，大規模な問題になるほどより高速に解けることを示した
  - 以下に IBM Blue Gene 1024ノードでの実行結果を示す（2次元 Poisson 方程式，次数4,900万元）
  - 反復解法ライブラリ Lis 上に対称問題・非対称問題版とも実装済み



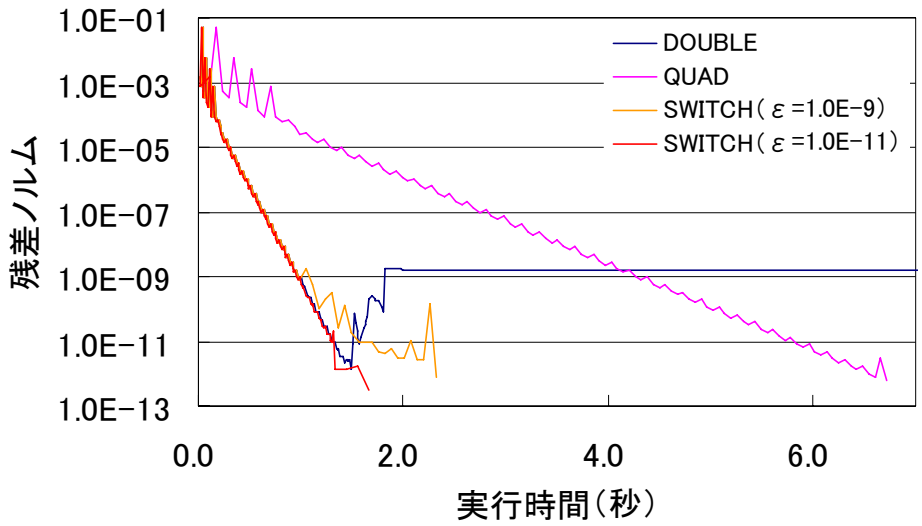
反復解法

- 高精度計算(実装)
  - 共役勾配法などのクリロフ部分空間法の収束性は丸め誤差に大きく影響されるが，高精度演算を用いた場合，通常は計算時間が多くかかってしまう.
    - 倍精度浮動小数を2個用いた double-double 型4倍精度演算の Lis への実装
    - SIMD 命令を用いた高速化手法
    - 倍精度と4倍精度の混合型反復法を提案
  - Intel Fortran コンパイラを用いた完全4倍精度演算に比べて約5.8倍の性能向上を得た
  - また，これを用いた4倍精度のみの計算と比較して最良で2.8倍の性能向上を得た



反復解法

- 高精度計算(性能評価)
  - 以下に Toeplitz係数行列を用いた倍精度(DOUBLE), 4倍精度(QUAD), 収束の停滞を検知して4倍精度に切り替える混合精度演算での計算例を示す



反復解法

- 双共役残差（BiCR）法
  - 対称行列用の共役残差 (CR, conjugate residual) 法を非対称行列用へ拡張
  - 双共役勾配 (BiCG) 法よりも多くの場合に安定した収束特性を示す
  - また，積型反復解法にこれを応用した GPBiCR 法を提案

set $\mathbf{x}_0$ is an initial guess, $\mathbf{r}_0 = \mathbf{b} - \mathbf{A}\mathbf{x}_0$, $\beta_{-1} = 0$,

$\mathbf{r}_0^*$ is an arbitrary vector, such that $(\mathbf{r}_0^*, \mathbf{r}_0) \neq 0$., e.g., $\mathbf{r}_0^* = \mathbf{r}_0$,

for $n = 0, \cdots$ until $\| \mathbf{r}_n \| \leq \varepsilon \| \mathbf{r}_0 \|$ do :

begin

$$\mathbf{p}_n = \mathbf{r}_n + \beta_{n-1}\mathbf{p}_{n-1}, \qquad \mathbf{p}_n^* = \mathbf{r}_n^* + \beta_{n-1}\mathbf{p}_{n-1}^*,$$

$$(\mathbf{A}\mathbf{p}_n = \mathbf{A}\mathbf{r}_n + \beta_{n-1}\mathbf{A}\mathbf{p}_{n-1}),$$

$$\alpha_n = \frac{(\mathbf{r}_n^*, \mathbf{A}\,\mathbf{r}_n)}{(\mathbf{A}^{\mathrm{T}}\mathbf{p}_n^*, \mathbf{A}\mathbf{p}_n)},$$

$$\mathbf{r}_{n+1} = \mathbf{r}_n - \alpha_n \mathbf{A}\mathbf{p}_n, \qquad \mathbf{r}_{n+1}^* = \mathbf{r}_n^* - \alpha_n \mathbf{A}^{\mathrm{T}}\mathbf{p}_n^*,$$

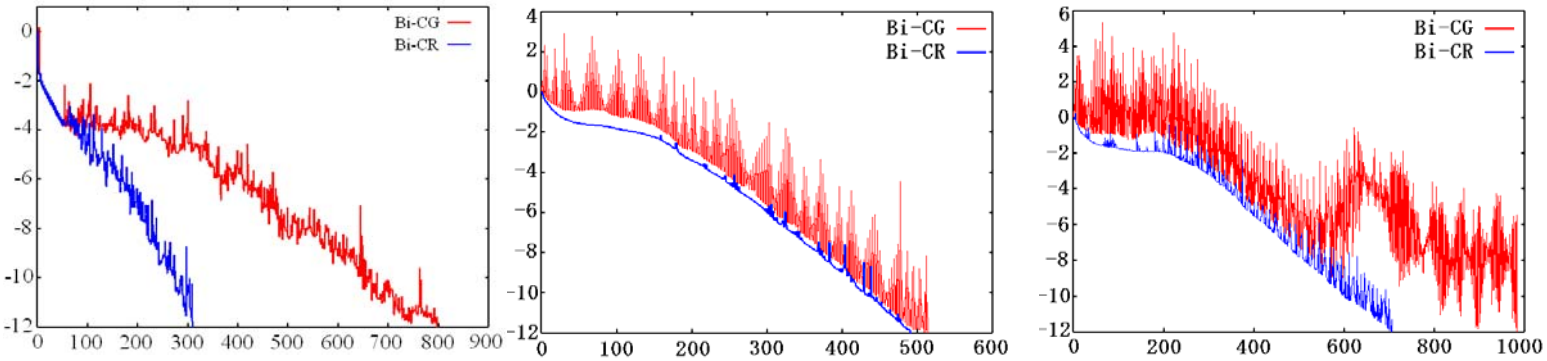
$$\mathbf{x}_{n+1} = \mathbf{x}_n + \alpha_n \mathbf{p}_n,$$

$$\beta_n = \frac{(\mathbf{r}_{n+1}^*, \mathbf{A}\,\mathbf{r}_{n+1})}{(\mathbf{r}_n^*, \mathbf{A}\,\mathbf{r}_n)},$$

end

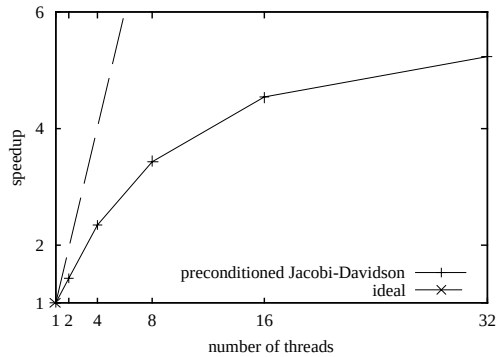
反復解法

- 双共役残差法(計算例)
  - Toeplitz 系(γ=1.5), Matrix Market から抽出した問題(WATT1, WATT2, petroleum engineering)での収束特性を示す
  - BiCR 系の各解法は Lis 1.1.0 より並列版を実装済み



反復解法

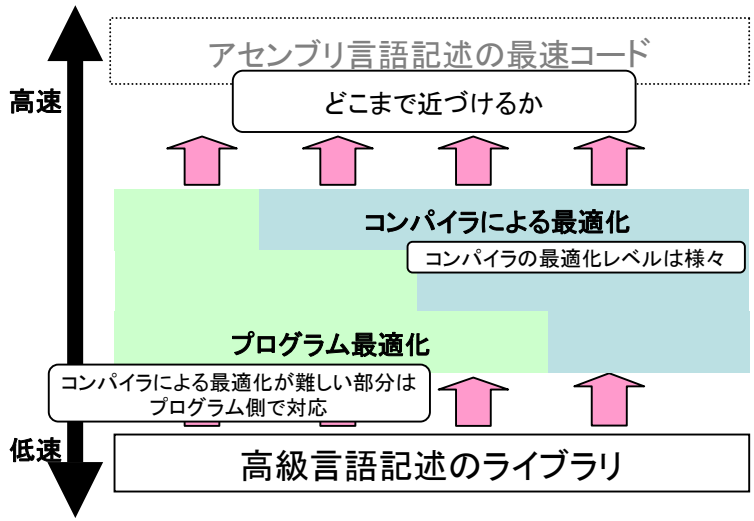
- 固有値解法
  - 少数の固有値・固有ベクトルを効率的に求める必要
  - Jacobi-Davidson 法の並列化と性能評価



  - 一般化固有値問題が Rayleigh 商の非線形最適化問題に帰着して共役勾配法により解けることに着目
    - 高いスケーラビリティ
    - 共役勾配法系の固有値解法では，代数的マルチグリッド法などの高性能な前処理と組み合わせることにより，Lanczos 法系の解法と比較してより高速に固有値を計算可能
    - Lis をベースとして並列ライブラリとして実装
    - 関連研究
      - Lawrence Berkeley 米国立研究所による反復解法ライブラリ Hypre への組み込みに関する研究（Knyazev）

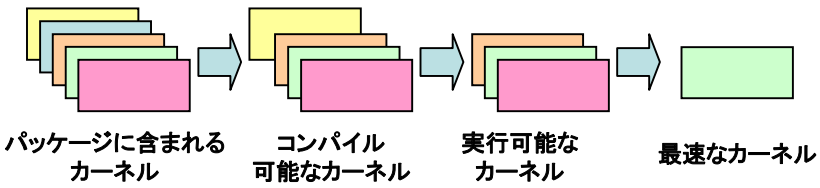
高速関数変換

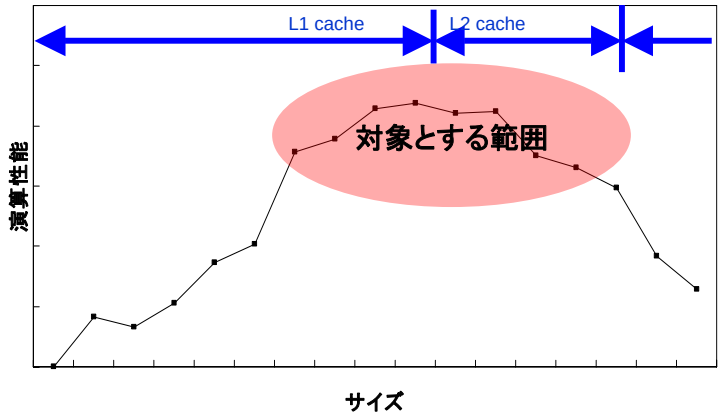
- 高速フーリエ変換
  - 大規模な科学技術計算から信号処理まで，多くの分野で利用
  - プロセッサアーキテクチャの急激な進歩に伴い，それに適した新しい FFT カーネルの開発が必要
  - カーネルの自動最適化機能を備えた高速フーリエ変換ライブラリ FFTSS を公開



高速関数変換

- 高速フーリエ変換（自動最適化）
  - 各種拡張命令の使用／不使用
  - FFTカーネルの基底の優先度
  - 各基底のカーネルの計算順序
  - 階層メモリへの最適化（多次元FFT）
  - 最適なMPI関数を選択（MPI版）





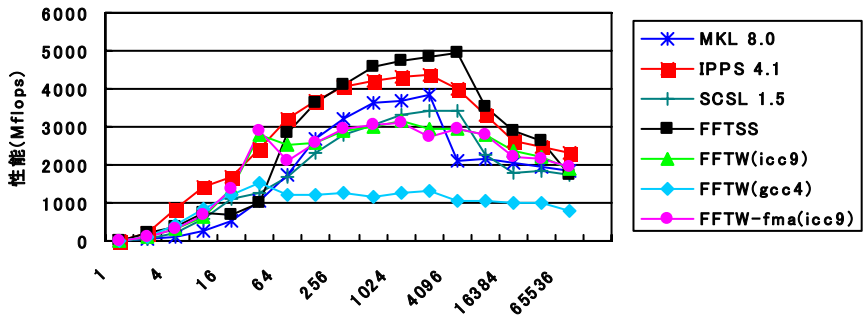
高速関数変換

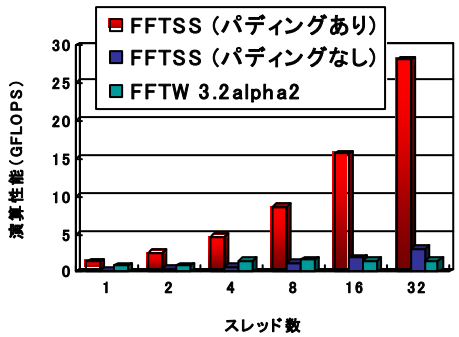
- 高速フーリエ変換(プログラミング例)

```
max_threads = omp_get_num_procs();
fftss_plan_with_nthreads(max_threads);
plan = fftss_plan_dft_2d(nx, ny, py, vin, vout,
FFTSS_FORWARD, FFTSS_MEASURE);
{ /* 配列の初期化. */ }
for (nthreads = 1; nthreads <= max_threads; nthreads ++) {
fftss_plan_with_nthreads(nthreads);
t = fftss_get_wtime();
fftss_execute(plan);
t = fftss_get_wtime() - t;
printf("%d スレッドで実行した場合%lf 秒.¥n", nthreads, t);
}
```

高速関数変換

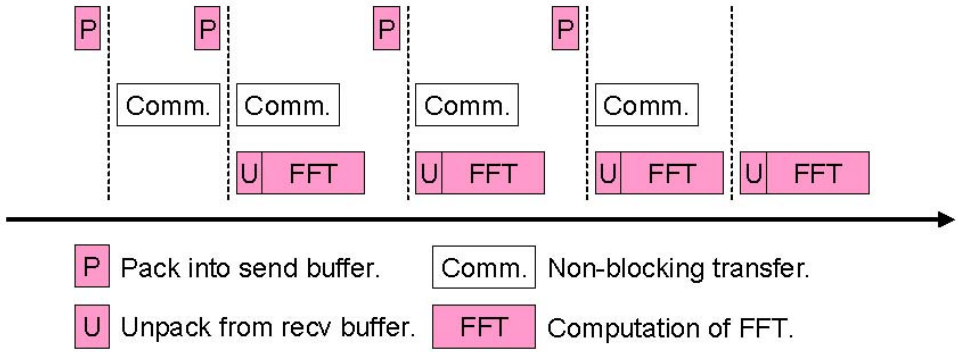
- 高速フーリエ変換
  - 商用ライブラリ等と比較した IBM POWER5 1.65GHz 上での1次元 FFT，パディングを実装したOpenMP 版2次元 FFT の SGI Altix 上での評価結果を示す





高速関数変換

- 高速フーリエ変換（並列環境への対応）
  - 大量のデータを用いた大規模な FFT 計算が必要
  - MPI 並列環境向けライブラリの開発
  - 地球シミュレータ共同プロジェクトの一環としてベクトル計算機環境への最適化
    - メモリアクセス　⇒　ベクトル256GB／s
    - ノード間通信　⇒　高速12．3GB／s
    - 通信待ち　⇒　ソフトウェアパイプライン化して隠蔽
  - 倍精度2次元並列 FFT において，地球シミュレータ上で 16TFLOPS を達成



高速関数変換

- 高速ルジャンドル変換
  - 気象予報・気候予測などの分野では，球面上の球座標において直交関数系となる球面調和関数を用いることが多い
  - 緯度・経度格子上の関数値と球面調和関数展開の係数との間の変換である球面調和関数変換の高速化が重要
  - 球面調和関数は東西方向の三角関数と南北方向のルジャンドル陪関数との積に分解できる
  - 前者はFFTで処理可能
  - 後者を高速に解くための一般化高速多重極子展開法を提案，$O(N^2 \log N)$ での安定な計算が可能となることを示した

公開状況

- ソフトウェア
  - 成果物配布サイト http://www.ssisc.org/ (Scalable Software Infrastructure Project HP) にて公開
  - 平成17年9月より年1回の割合でメジャーアップデート(ACT-JSTベースのライセンスで公開開始)
  - 平成19年10月より修正BSDライセンス(修正・複製・再頒布を許可)にて公開開始
  - 国内外の主要メーリングリストにて広報
  - 平成19年11月までの間に, Lis, FFTSS, SILCの国内外からのダウンロード件数は, それぞれ436件, 433件, 169件

成果発表

- 論文発表 … 海外20件，国内13件
- 口頭発表 … 海外15件，国内37件
- 特許出願 … PCT出願1件，国内1件

第1回SSIプロジェクトワークショップ風景（平成18年12月27日撮影）

まとめ

- 実装手法
  - 計算機科学分野で得られたハードウェア技術，ソフトウェア技術に関する知見を迅速に科学技術計算に応用するためのサイクル作り
  - 数値ライブラリの利用を容易にするためのミドルウェア SILC の開発
    - 並列環境への対応
    - スクリプト言語としての利用
- 反復解法
  - 反復解法ライブラリ Lis
    - 逐次，OpenMP, MPI, OpenMP・MPIハイブリッド並列環境に対応
    - さまざまな解法・前処理・行列格納形式に対応
  - 計算機科学，計算科学両分野の知見を反映したソフトウェア基盤として，十分な波及効果
- 高速関数変換
  - 高速フーリエ変換ライブラリ FFTSS
    - オートチューニング機能により，計算機ベンダにより提供されているカーネルと比較しても優れた性能
    - 信号処理，X線CT等での利用事例．大規模並列処理への利用に期待

むすび

- メンバーの意見を取り入れつつ，一貫した研究方針を保つことに腐心
- 研究費に関しては，プロジェクトの方針に沿った使い方ができた
- 若手研究者については，1名の研究補助員，3名のポスドク研究員を受け入れ．各人とも高性能計算分野において優秀な人材に
- 戦略的創造研究推進事業を担当する機会を与えていただいたことに感謝致します．