

Xabclib : 汎用な自動チューニング インターフェースOpenATLib を利用した疎行列反復解法ライブラリ

片桐孝洋(東大) 黒田久泰(愛媛大/東大)
中島研吾(東大)

共同研究者:
櫻井隆雄、直野健(日立中研)

第5回ASE研究会
東京大学情報基盤センター大会議室
2009年8月21日 16時~17時

話の流れ

- 第一部
 - 背景
 - AT実装再利用性
 - ATインターフェースが持つべき機能
 - ATポリシー設定機能：先進的なAT機能
 - OpenATLib：
汎用的なATインターフェースライブラリ
- 第二部
 - Xabclib：OpenATLibを利用した
疎行列反復解法ライブラリ
 - 性能評価
 - 既存アプローチとの違い
- おわりに



背景

ソフトウェア性能工学の必要性

- IT企業の重役にむけた調査[1]では、その半数が製品の性能問題に直面し、少なくとも20%は再開発している。
- 再開発には、遅延、コスト増加、開発失敗、のリスクが付きまとう。
- 性能を考慮したソフトウェア工学体系が必要である。
- **ソフトウェア性能工学**
(Software Performance Engineering, SPE) [2]
 - 定義：ソフトウェア工学の全般と、ソフトウェア開発サイクルを通して直面する性能要求に見合う解析 の工学体系

[1] Compuware, Applied Performance Management Survey, Oct. 2006.

[2] C. U. Smith, L. G. Williams: Software Performance Engineering:
A Case Study Including Performance Comparison with Design Alternatives,
IEEE Transactions on Software Engineering archive Volume 19 , Issue 7 (July 1993)

ソフトウェア自動チューニング 工学の全体像

コード生成

コンパイルと実行

3. 最適化フェーズ

実行結果の解析

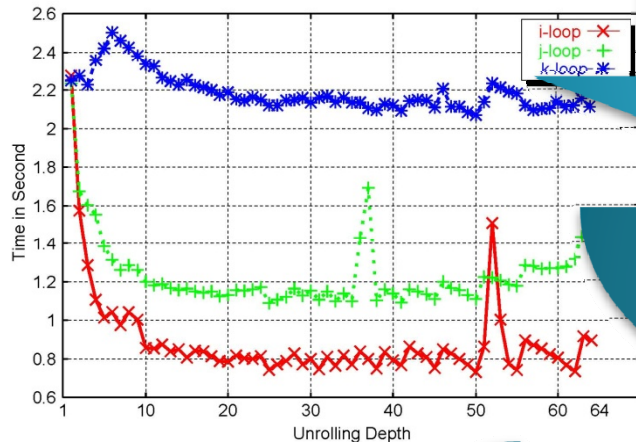
2. プログラミング
フェーズ

4. データベース化
とチューニング
知識探索
フェーズ

チューニング知識
データベース

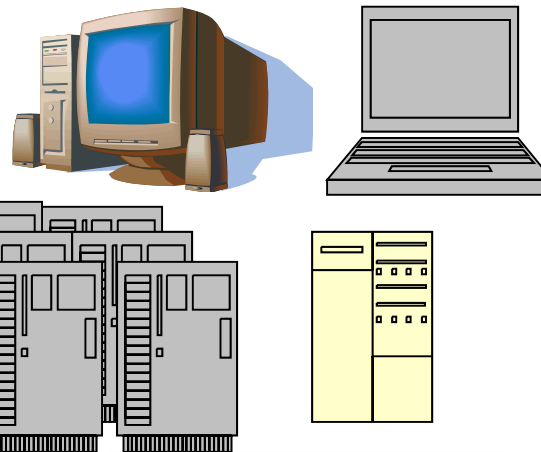
対象計算機

1. 仕様策定フェーズ



```
do i=1, n
  do j=1, n
    do k=1, n
      C(i, j) = C(i, j) + A(i, k) * B(k, j)
    enddo
  enddo
enddo
```

```
do i=1, n, 2
  do j=1, n
    do k=1, n, 2
      Btmp1 = B(k, j)
      Btmp2 = B(k+1, j)
      Ctmp1 = C(i, k) * Btmp1
      Ctmp1 = Ctmp1 + A(i, k+1) * Btmp2
      Ctmp2 = Ctmp2 + A(i+1, k) * Btmp1
      Ctmp2 = Ctmp2 + A(i+1, k+1) * Btmp2
    enddo
    C(i, j) = Ctmp1
    C(i+1, j) = Ctmp2
  enddo
enddo
```



```
!ABCLib$ install unroll (i,k) region start
!ABCLib$ name MyMatMul
!ABCLib$ varied (i,k) from 1 to 8
do i=1, n
  do j=1, n
    do k=1, n
      C(i, j) = C(i, j) + A(i, k) * B(k, j)
    enddo
  enddo
enddo
!ABCLib$ install unroll (i,k) region end
```

計算機環境の変化

- マルチコアの浸透

- 非均質メモリアクセス(ccNUMA)
- 多階層化されたキャッシュ構造
 - L1、L2は局所、L3は共有
- チップ内のコア数の増大
 - ハイエンド：32コア～、ローエンド：8コア～

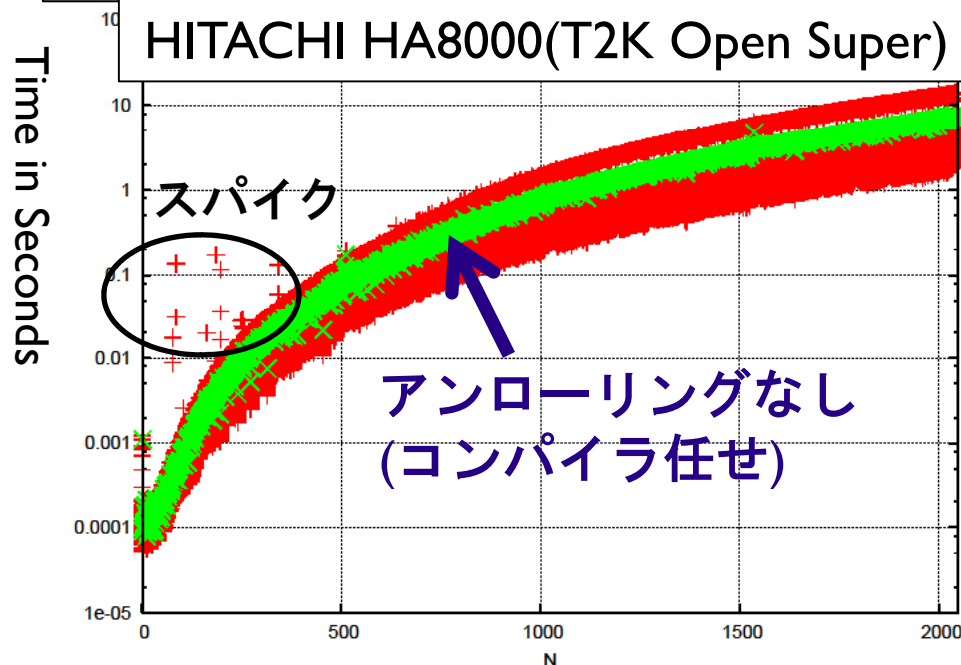
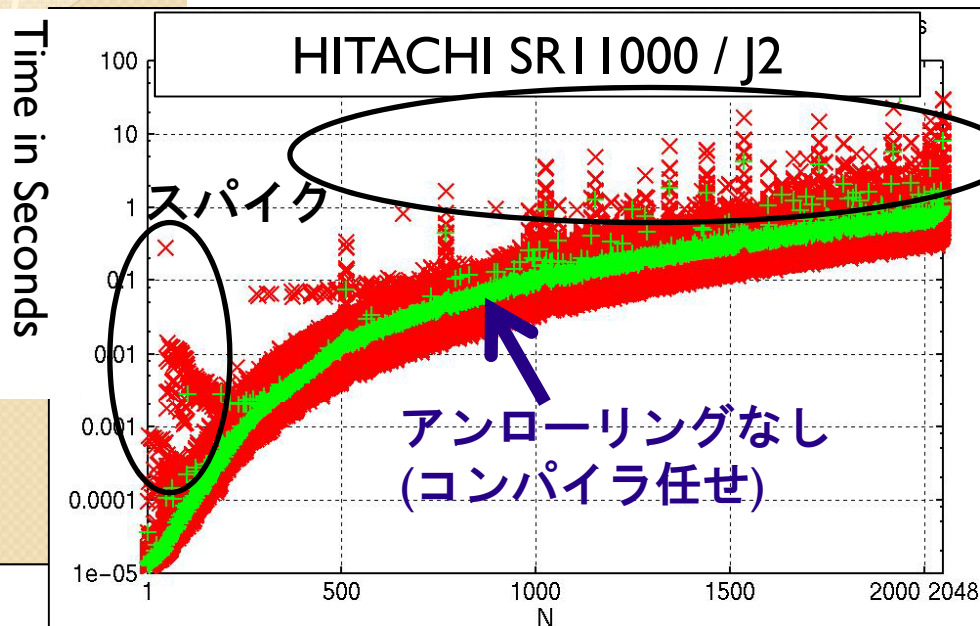
- 並列実行モデルの変化

- ピュアMPI実行
 - 低～中並列実行（～1,000コア）時
- ハイブリッドMPI（OpenMP + MPI）実行
 - 超並列実行（10,000+コア実行）時

実行時自動チューニングの必要性

- 現実の性能データはモデル化できない
 - 予測できない「スパイク」がある
 - 入力データ依存性がある
 - 実行時に定まる、疎行列形状 や 数値特性
 - ・ 最適な 前処理方式 や 利用アルゴリズムが 実行時まで不明
 - OSのリソースマネージメント依存
 - 割り当てられる ノード や ソケットが実行時まで不明
 - ・ ローカルメモリがつかえるか
 - ・ マスター・スレーブ形態で、理想的にコア割り当てされるか
 - 割り当てノード群に対する、物理ネットワーク形状（通信性能）が実行時まで不明！
- もうコンパイラでは手に負えない
 - 手動チューニングはコスト高
 - 手動でできないこともある
 - 予測できない極端な性能差が不確定におこる場合
 - 実用的観点から性能自動チューニング技術へ期待

性能不安定性：行列-行列積



●密行列の行列-行列積

BLASを用いていない単純コード

●3重ループ (i,j,k), アンローリング1段~4段

●次元Nに関し $4*4*4=64$ 種類の実装

●1 から2048次元まで1刻みのデータ.

●コンパイラ HITACHI Optimized Fortran90.

オプション: -Oss

●自動並列化 (ノード内)

●計算機構成:

●HITACHI SRI1000/J2

●HA8000 (T2K Open Supercomputer
(Todai Combined Cluster))

●Installed in Information Technology
Center, The University of Tokyo.

●16コア/ノード.

●コンパイラ任せは遅い

●常時<特定の実装>が
速くない

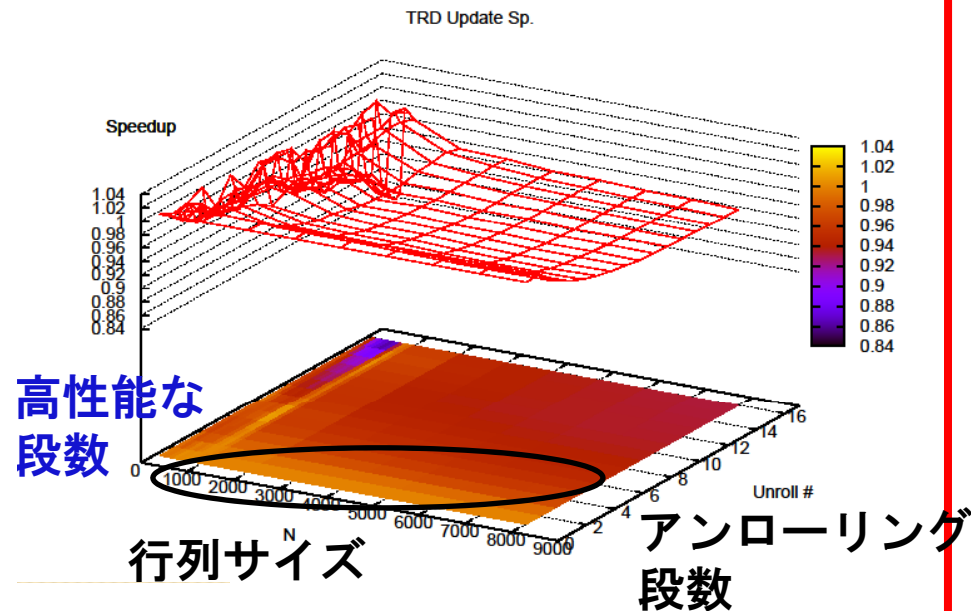
●10倍ぐらい実行時間がぶれる
(実行時間の<不安定性>)

●場合により100倍も遅い!

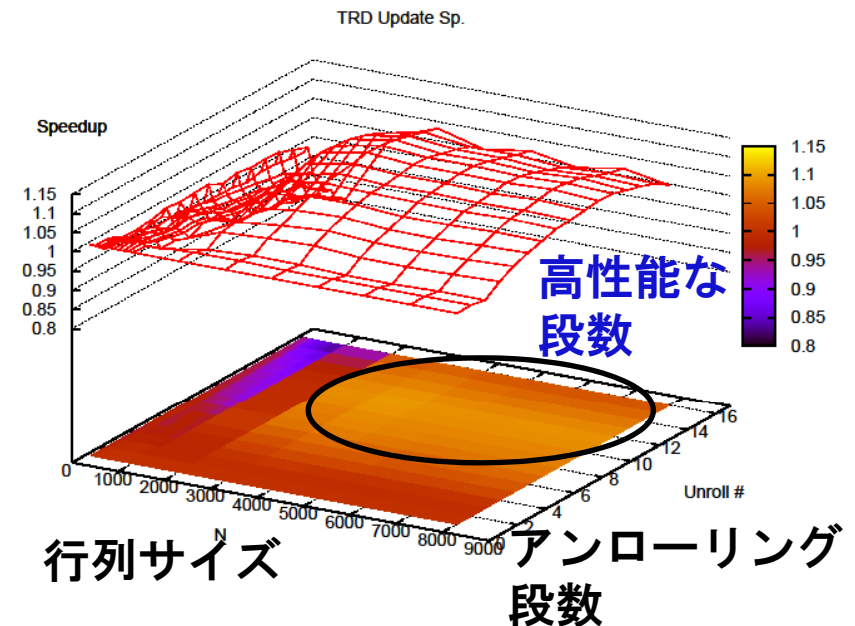
マルチコアでの実行形態の違いによる 性能挙動の劇的変化例 (T2Kオープンスパコン)

対称固有値ソルバの三重対角化 (行列更新部分)

4コア (4コア / 1ソケット実行)



4コア (1コア / 1ソケット実行)



ソケットごとの共有L3キャッシュの影響

→ (動的な) 実行コア数により最適実装が変わる！

自動チューニング（AT）機構とは

- ・計算機アーキテクチャ
- ・計算機システム

- ・プログラム
- ・アルゴリズム

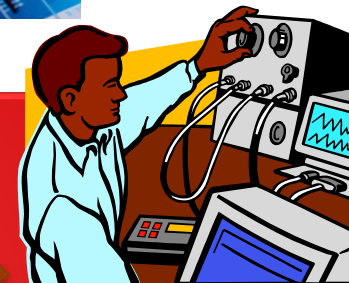


性能調整つまみ
(性能パラメタ)



調整機構

- ・最適化
- ・パラメタ探索
- ・学習／自己適応

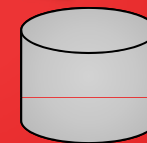


性能モニタ
機構

- ・プログラム
- ・アルゴリズム



つまみ
自動生成
機構

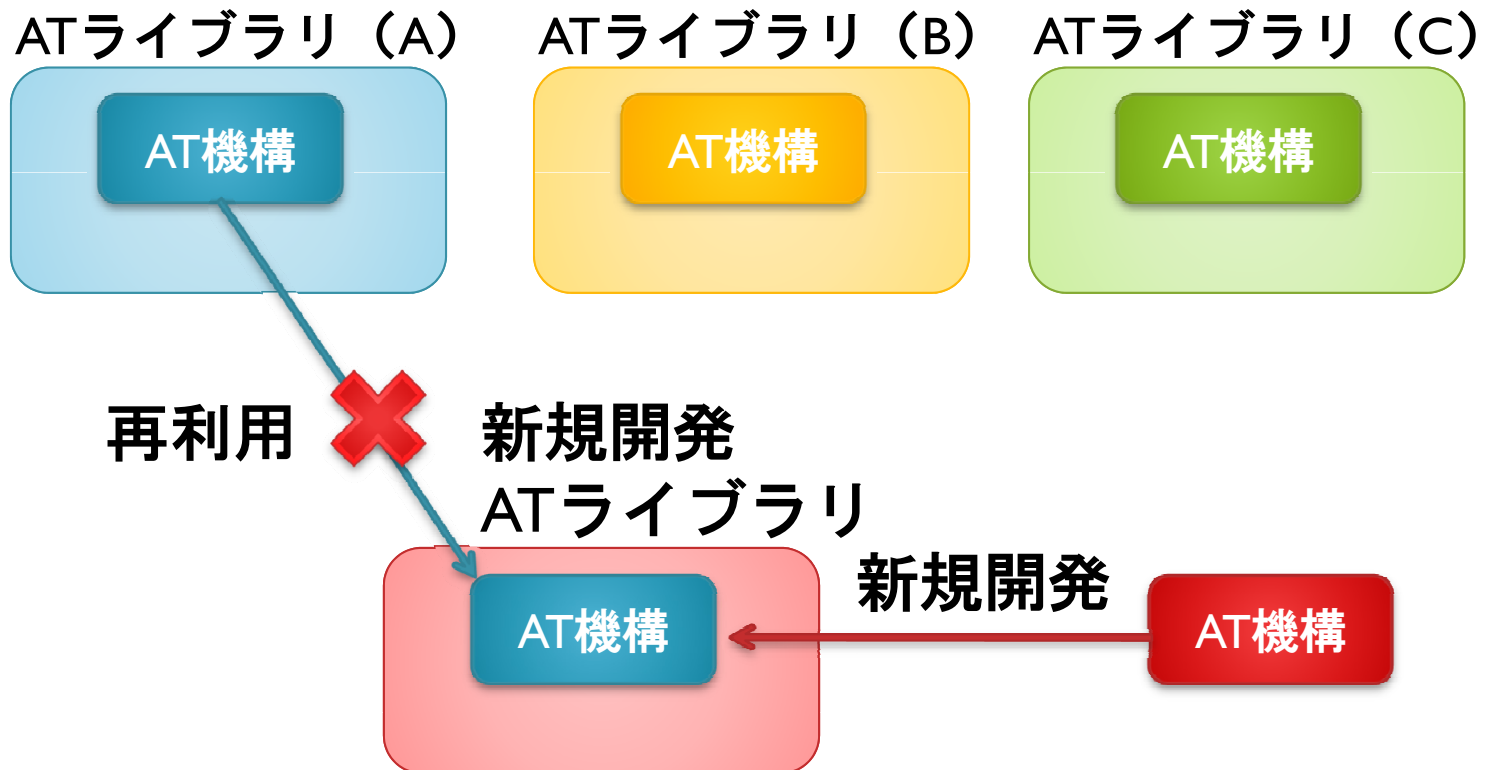


AT性能データベース

自動チューニング機構

従来ATライブラリの問題

- 個々に、AT機構が開発されている
- 個々の、AT機構は再利用できない
 - “AT実装再利用性” がない



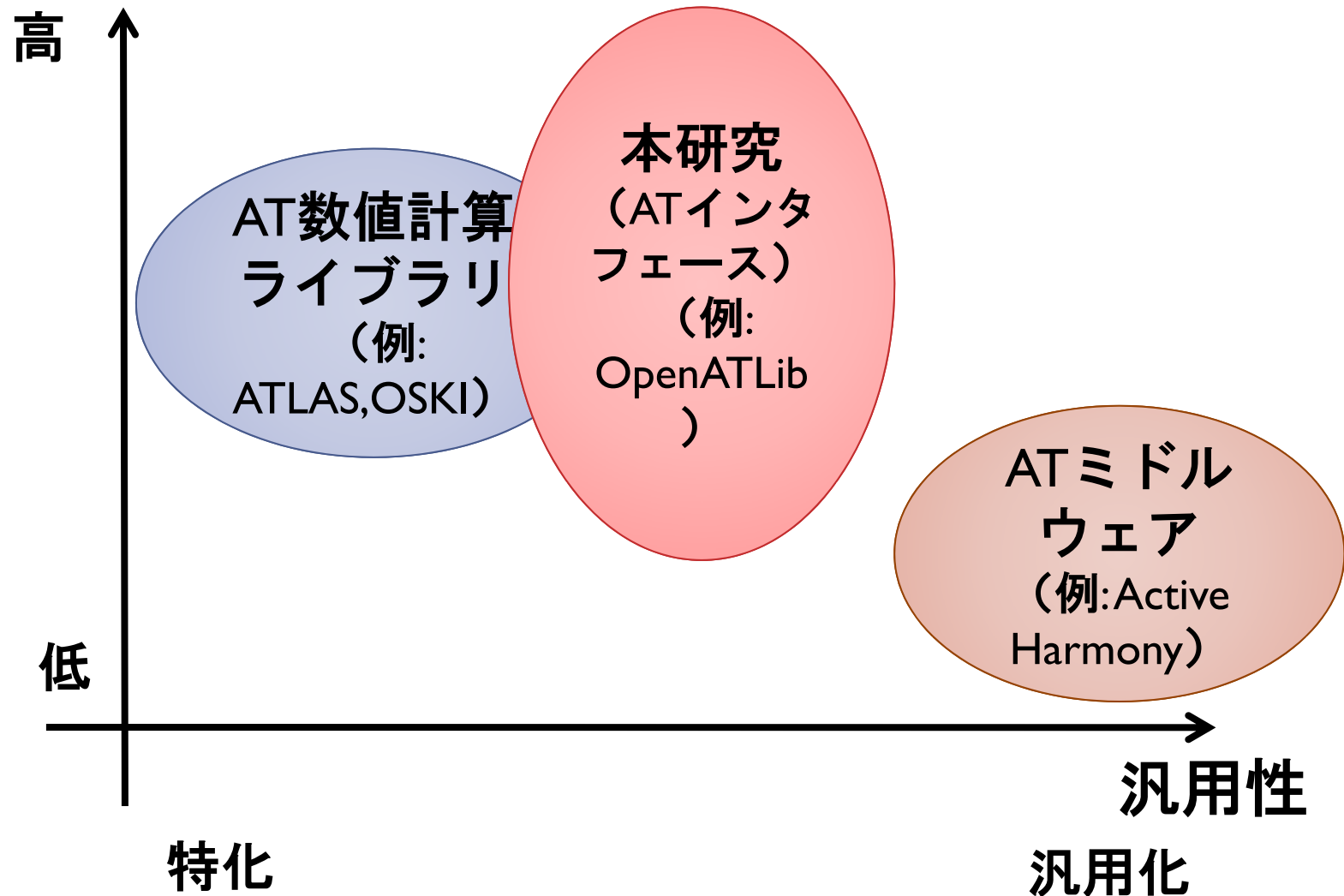
本研究の目的

- AT実装再利用性を達成する枠組み
 1. AT機構で必要となる機能を定義
 2. 1を基に汎用的なAPIを定義
 3. 2をライブラリ化して公開
 4. 3のライブラリのAPIを利用し、ATライブラリを個々の開発者が作成
- 適用対象
 - 数値計算ライブラリ
- 汎用化の度合い
 - 基本演算レベル（BLAS）以上の高度なAT機能を提供する
 - 多くの数値計算ライブラリの要求に対応
 - BLASレベルでは、従来のATライブラリでAT実装再利用性を達成（例：ATLAS）

本研究のアプローチの位置づけ

ソフトウェア階層

(高：アプリケーション階層、低：ハードウェア階層)





ATインターフェースが 持つべき機能

想定ユーザとAT要求仕様の違い

- **ATライブラリ開発者**

- ATライブラリを開発するユーザ
- 要求AT機能：
高度なものが必要
- インターフェース：
複雑でもよい

- **エンドユーザ**

- ATライブラリを利用するユーザ
- 要求AT機能：
ATを意識させない、もしくは、限定する
- インタフェース：
完全隠ぺい、もしくは、単純化

エンドユーザの観点での AT 2 要因

- 静的要因（S要因）
 - AT実行が、エンドユーザによる
ライブラリ利用前に行える
- 動的要因（D要因）
 - AT実行が、エンドユーザによる
ライブラリ利用時、もしくは、
少なくとも 1 回の利用後に行える

2要因による要求機能の違い

要因	ATタイミング と所要時間	ATデータ 採取分類	ATデータベー ス 連結性
静的要因 (S要因)	●インストール時 ●数時間～数日	●試行 ●性能モデル化のためのお試し実行が主	●ATデータベースへのデータ蓄積が主
動的要因 (D要因)	●実行起動前時、および、実行時 ●数秒～数分	●実施 ●性能モデルからの予測パラメタによる実行が主	●ATデータベースからのデータ読み込みが主 ●少数のデータの蓄積 ●S要因のDBから連結性が必要

ATインターフェース機能の分類

- 従来方式

機能分類

提供先

- 性能パラメータ最適化機能

- アルゴリズム選択
- 実装方式選択
- 計算資源選択

- ユーザ知識設定機能

- ATを行う位置
- 疎行列の形状情報
 - ランダム、帯、ブロック

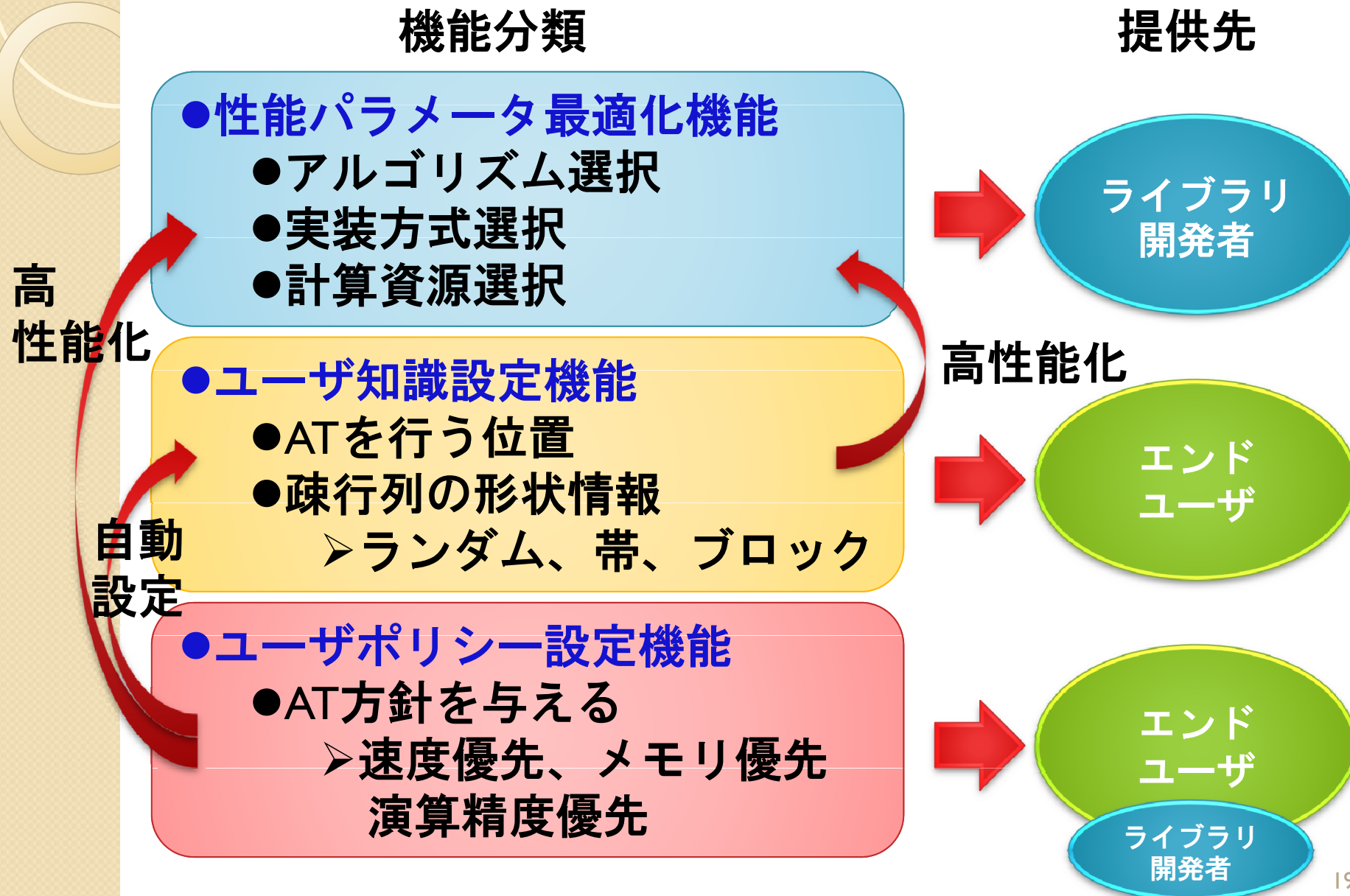
ライブラリ
開発者

高性能化

エンド
ユーザ

ATインターフェース機能の分類

● 提案方式



ATポリシー記述の一例

- [直野ら, 2005]

```
1 $POLICY libname = SOLVER-A
2 $POLICY RCP = CPU, CPU_limit = 100, CPU_best = 20
3 $POLICY UAF = time, time_limit = 1000 sec, time_best = 100 sec
4 $POLICY selection = min; time / time_best + CPU / CPU_best
5 $POLICY tune ICP
6 CALL SOLVER-A (UCP, ICP)
```

CPU数：100個以下、20個以上

実行時間：1000秒以下、100秒以上

最適化の目的関数：コスト定義関数を最小化

コスト定義関数を定義



反復解法ソルバ XABCLIB (OPENMP並列化版)

※愛媛大学 黒田久泰 先生
東京大学 中島研吾 先生
日立製作所 直野健 博士
日立製作所 櫻井隆雄 氏
との共同研究

e-サイエンス実現のためのシステム統合・連携ソフトウェアの研究開発
高生産・高性能計算機環境実現のためのシステムソフトウェアの研究開発
「シームレス高生産・高性能プログラミング環境」高性能高可搬性ライブラリ
(平成20年度～平成23年度) : 代表 東京大学 石川裕 教授

- 問題

- 最適化が職人芸に依存、生産性がない、可搬性がない、煩雑な処理、コストがかかる
- 理論的に範囲外のパラメタ指定による収束失敗

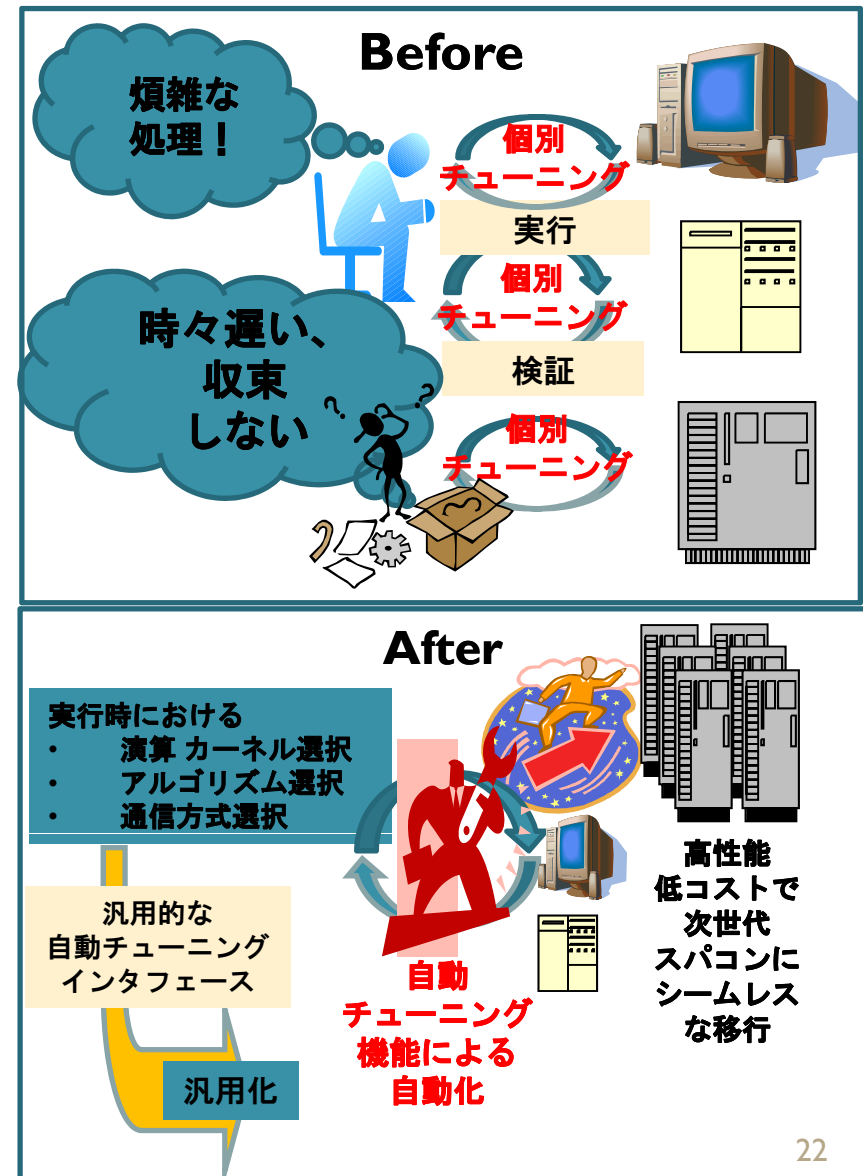
- 目標

- 高生産性／性能移植性をもつ数値計算ライブラリの提供

- 疎行列の非ゼロ成分の分布を考慮し

実行時における以下の自動チューニング機能を提供

1. 演算カーネル選択
2. 数値アルゴリズム選択
3. 並列実装方式選択
4. 汎用的なAPI (OpenATLib)



OpenATLibの位置づけ

- D要因に起因するATインターフェースを、ATライブラリ開発者に提供
- 性能パラメータ最適化機能のうち、以下の機能の実装
 - アルゴリズム選択処理
 - Krylov部分空間のリスタート周期選択
 - 実装方式選択
 - 疎行列 - ベクトル積
 - キャッシュ向け最適化、アンローリング段数、並列化向き実装
 - 計算機資源選択
 - 実行時のメモリ量を考慮した疎行列 - ベクトル積
- ユーザポリシー指定機能
 - 本年度、新規開発を予定

E-Science Development Library

- Development of 2009 (**Development From Scratch**)
 - **Xabclib_LANCZOS**
 - Eigensolver with the Restart LANCZOS for Standard Eigenproblem
 - **Xabclib_GMRES**
 - Linear Equations Iterative Solver with GMRES(m)
 - **OpenATlib**
 - **Common AT Interface Library**
- The following AT Facilities are implemented:
 1. General Auto-tuning Facility for Re-start Frequency
 2. Run-time Selection for SpMxV Implementations
 3. Optimization for Multicore Processors
 - Optimized for invoked number of processes
 4. **Run-time Implementation Selection Based on Run-time Memory Restriction**
 5. Common AT Interface to Establish The above AT Functions
- Fortran and OpenMP Parallelization, CRS format

OpenATlib命名規則

- OpenATLibが提供する機能は以下の3種類に分類
 - a) 演算機能 (例：行列-ベクトル積など)
 - b) 補助機能 (例：特定パラメータの調整など)
 - c) パラメータ管理機能 (OpenATI_BLDATA)
- a), b)の各機能名の "OpenATI_" 後の 5文字の命名規則
 - 1文字目 型を示す文字が入る。
 - S : Single Precision
 - D : Double Precision
 - 2-3文字目
 - 補助機能のとき : AF が入る。
 - 演算機能のとき : 2文字目に行列の種類, 3文字目に行列格納形式を示す文字が入る。
 - 2文字目
 - S : Symmetry
 - U : Unsymmetry
 - D : Diagonal
 - T : Tridiagonal
 - 3文字目
 - R : CRS形式
 - C : CCS形式
 - 4-5文字目 処理の種類を示す文字入る。
 - MV : 行列-ベクトル積
 - RT : リスタート周期の判定

OpenATlib: A General AT Interface for Library Developers

- **Restart Frequency Interface**

OpenATI_DAFRT(NSAMP, SAMP, IRT, INFO)

- NSAMP: The Number of Sampling Points
- SAMP: Sampling Data (double)
- IRT: Judgment Flag
 - 0 : Do not need Increase
 - 1 : Need Increase

- **Judgment Function : MM ratio [T.Sakurai,2008]**

$$R_i(s, t) = \frac{\max_z \{ r_i(z); z=s-t+1, \dots, s \}}{\min_z \{ r_i(z); z=s-t+1, \dots, s \}}$$

Is a ratio for the max per min on residuals r_i from $s-t+1$ to s .

Extension to ILIB: Last S samples.

OpenATI_DAFRTの利用例

```
INCLUDE "OpenAT.inc" // OpenAT.incのインクルード  
～略～
```

```
DO ITER=1, MAXITER //反復解法のループ  
  MSIZE=1 //最初のリスタート周期  
  I=5 //判定の頻度
```

～略～

```
IF (RSDID < TOL) RETURN // 収束判定  
SAMP (K)=RSDID // SAMP(K)に残差を入力  
IF (ITER .eq. I) THEN // I回にI回DAFRTをCALL  
  IRT=0
```

```
  CALL OpenATI_DAFRT (I, SAMP, IRT, INFO)
```

```
  IF (IRT .eq. I) MSIZE=MSIZE+1 //リスタート周期を増加  
  K=0  
END IF
```

～略～

```
ENDDO
```

OpenATlib: A General AT Interface for Library Developers

- **SpMxV Interface ($y = Ax$)**
 - OpenATI_DSRMV() : Symmetric**
 - OepnATI_DURMV() : Unsymmetric**
 - Arguments:
 - N : Matrix Size
 - NNZ : Number of Nonzero Elements
 - IRP(N+1) : Diagonal Index Pointer
 - ICOL(NNZ) : Row Index Pointer
 - VAL(NNZ) : Element Value
 - X(N) : RHS Vector
 - Y(N) : LHS vector
 - **ICASE : AT Implement Switch No**
 - **NUM_SMP: Reduction Number**
 - **WK(N, NUM_SMP) : Work Space**
 - **INFO : Error Code**

OpenATI_DSRMVの利用例

```
INCLUDE "OpenAT.inc"           // OpenAT.incのインクルード
OpenATI_DSRMV_IPARM_I=3      // DSRMVのパラメータ
ICASE=0                      // DSRMVのパラメータ

DO ITER=1, MAXITER           //反復解法のループ
                                ~略~

    IF (ITER .eq. 2) THEN
        OpenATI_DSRMV_IPARM_I=1 //以後は判定された実行方式を用いる
    ENDIF

                                ~略~

    // 行列 - ベクトル積の呼び出し
    CALL OpenATI_DSRMV
        (N, NNZ, IRP, ICOL, VAL, VEC, JPARM, IPARM, RPARM, INFO)
                                ~略~

    IF (解が収束した) RETURN

ENDDO
```




性能評価

T 2 KオープンスパコンCPU緒元

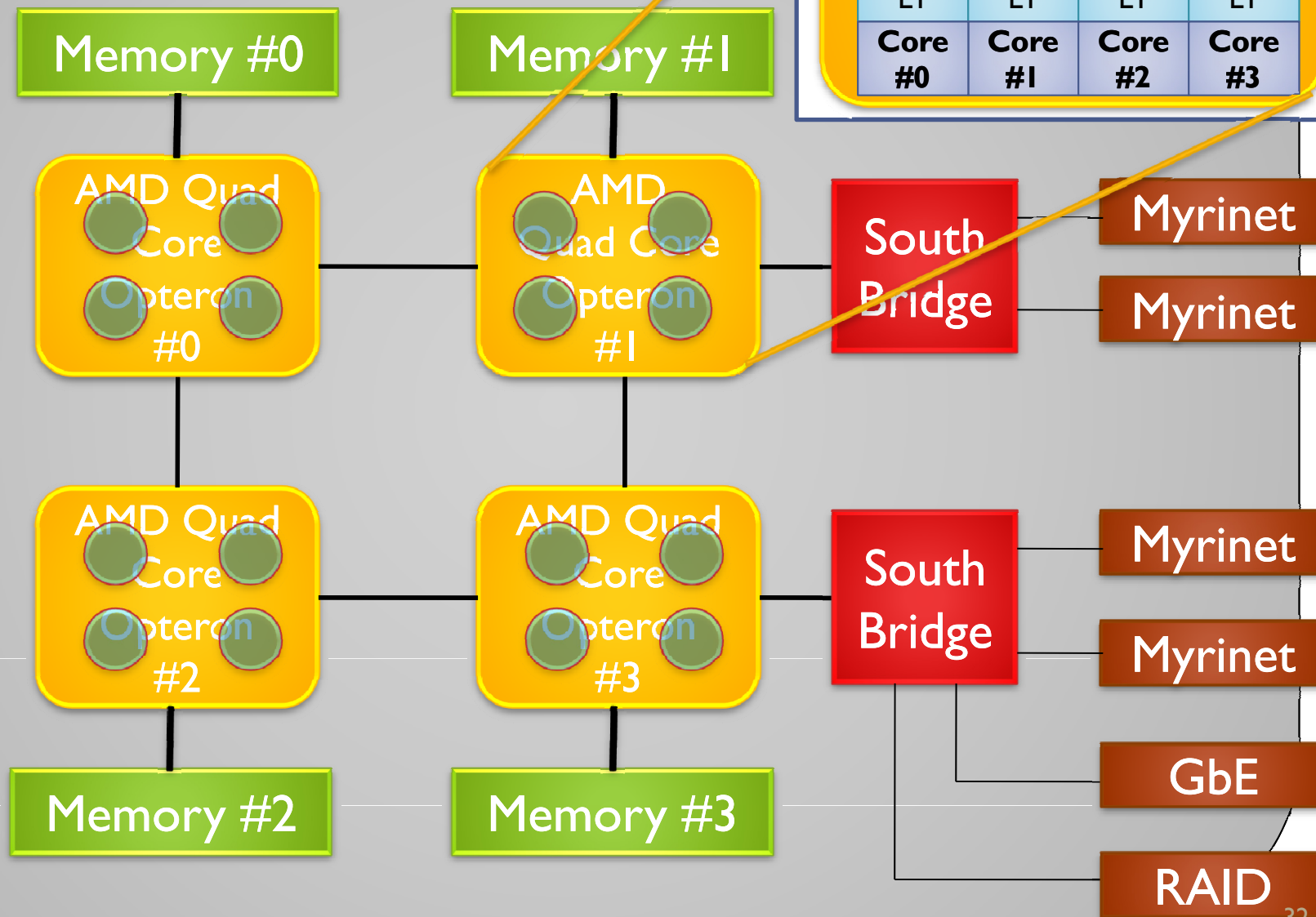
(AMD Quad Core Opteron 2.3GHz)

項目	値
L1キャッシュ	64 Kbytes (命令、データ、双方は分離)
	2 Way Associativity (ライトバック、3サイクル)
	キャッシュライン: 64 bytes、LRU置換
L1データTLB	Full Associativity
	2Mbyte Page:8エントリ, 4Kbyte Page:32エントリ
L2キャッシュ	512 Kbytes (2300 MHz)
L3キャッシュ	2048 Kbytes
命令デコード	3 Way
演算実行	3 Way (整数、アドレス生成、浮動小数点)
SIMD命令実行	MMX, SSE, SSE2
命令実行	乱発行乱終了 (Out-of-order) 整数、浮動小数点
レジスタ	● レガシーモード: 汎用32bit: 8個、128bit-XMM:8個、64bit MMX:8個 (x87互換用:8個と同一) ● 64bit モード: 汎用 64bit:16 個、128bit-XMM:16 個、64bit MMX:8個 (x87互換用:8個と同一)
システムバス	1000 MHz

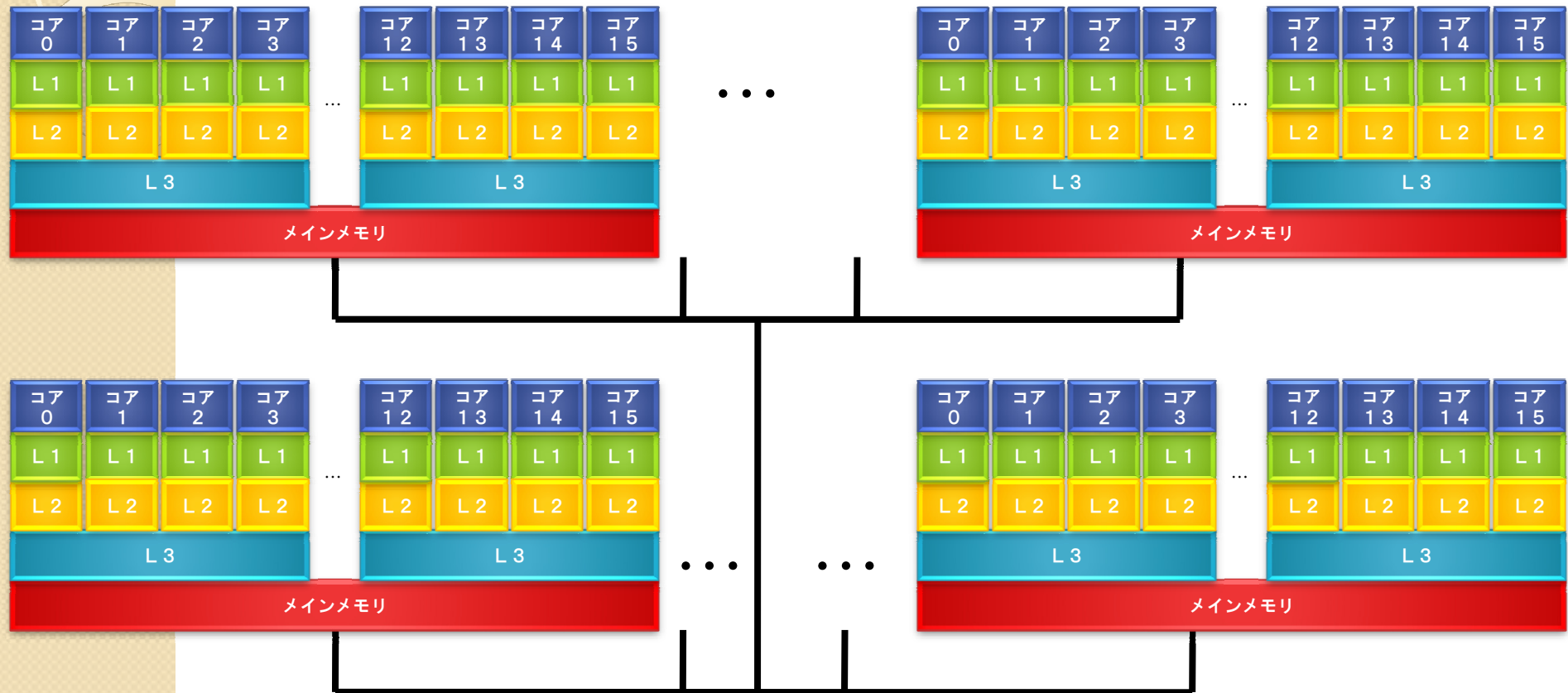
ノード構成（東大、タイプA群）

各CPUの内部構成

L3			
L2	L2	L2	L2
L1	L1	L1	L1
Core #0	Core #1	Core #2	Core #3



スパコンでの全体メモリ構成図



メモリが多段に階層化
(L1、L2、L3、分散メモリ)

高速ネットワーク
(5 Gバイト/秒
×双方向)
(タイプA群)

性能評価環境

- T2K オープンスパコン(東大版)
- OpenMPによる1ノード実行
(最大1ノードで16コア)
- コンパイラ
 - Intel Fortran Compiler Professional Version 11.0
- コンパイラオプション:
 - -O3 -m64 -openmp -mcmmodel=medium
- **自動チューニング機能**
 - リスタート周期自動選択
 - SpMxVの実装選択
 - a) 通常; b) 8-2 アンロール; c) ベクトル化;
 - 実行時メモリ制約による実装選択
- アルゴリズムに特有な事項
 - LANCZOS: 10個の固有値を計算
 - GMRES : Jacobi前処理
 - その他の選択: Scaling, ILU, SSOR

Computer Environments

CPU	Quad-Core AMD Opteron(tm) Processor 8356 2.3GHz 16core / node
L2 Cache Size	2MByte / 4core
Main Memory	32GByte (8GBytes / Socket)
OS	Red Hat Enterprise Linux 5
Compiler	Intel Fortran Compiler Professional Version 11.0
Compiler Option	-O3 -m64 -openmp -mcmodel=medium

Computation Condition

●Xabclib_LANCZOS

Convergence Test Value	1.0E-08
Calculated Eigenvalues	10

●Xabclib_GMRES

Convergence Test Value	1.0E-08
RHS vector x	All elements are set to 1.
Initial guess	All elements are set to 0.
Preconditioner	ILU(0)

●Default Restart Frequency

30 : Same as PETSc default value

Test matrices for Xabclib_LANCZOS

- From University Florida Sparse Matrix Collection (20 kinds)

Matrix	N	NNZ	Field
vibrobox	12328	177578	acoustics
Lin	256000	1011200	chemistry
cfdl	70656	949510	Fluid dynamics
cfdl2	123440	1605669	
gyro	17361	519260	Model reduction
t3dl	20360	265113	
c-71	76638	468096	optimization
Si5H12	19896	379247	structural
SiO	33401	675528	
dawson5	51537	531157	

Matrix	N	NNZ	Field
H2O	67024	1141880	structural
F2	71505	2682895	
oilpan	73752	1835470	
shipsec1	140874	3977139	
bmw7st_1	141347	3740507	
SiO2	155331	5719417	
shipsec5	179860	5146478	
Si41Ge41H72	185639	7598452	
bmw3_2	227362	5757996	
Ga41As41H72	268096	9378286	

Test matrices for Xabclib_GMRES

- From University Florida Sparse Matrix Collection (20 kinds)

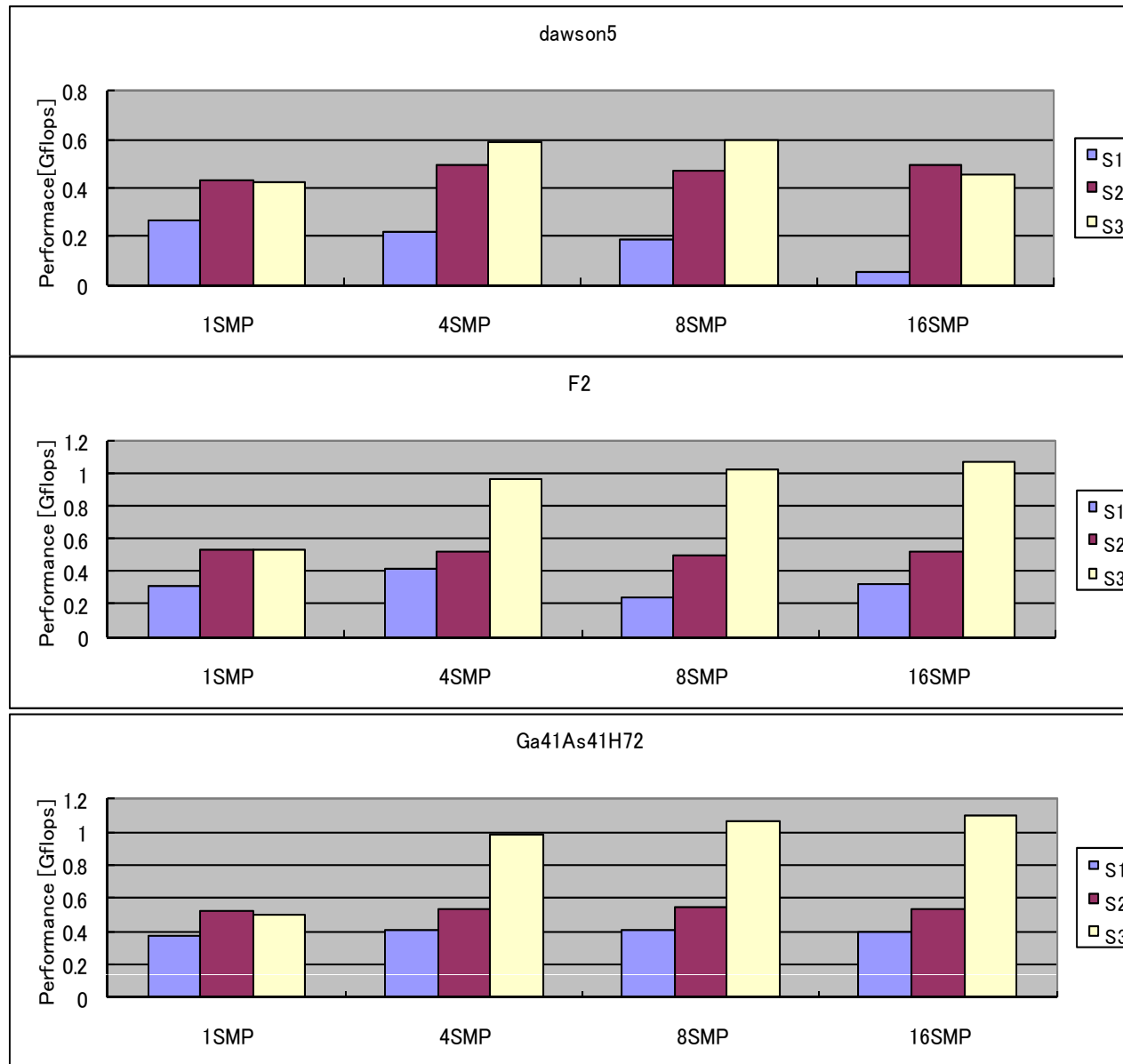
Matrix	N	NNZ	Field
chem_master1	40401	201201	2D/3D
torso2	115967	1033473	
torso1	116158	8516500	
torso3	259156	4429042	
memplus	17758	126150	electric circuit
ex19	12005	259879	fluid dynamics
poisson3Da	13514	352762	
airfoil_2d	14214	259688	
poisson3Db	85623	2374949	
viscoplastic2	32769	381326	materials

Matrix	N	NNZ	Field
xenon1	48600	1181120	materials
xenon2	157464	3866688	
wang4	26068	177196	semiconductor device
ec132	51993	380415	
sme3Da	12504	874887	structural
sme3Db	29067	2081063	
sme3Dc	42930	3148656	
epb1	14734	95053	thermal
epb2	25228	175027	
epb3	84617	463625	

Restart Frequency AT Effect: Xabclib_LANCZOS

Matrices	Fixed Restart Frequency			Auto-tuning			Speedup with AT
	M	restart	time(sec)	M	restart	time(sec)	
vibrobox	30	24	1.13	35	20	1.04	1.09
Lin	30	1047	601.65	150	54	214.49	2.81
cfdl	30	55	12.42	80	24	11.16	1.11
cf2	30	45	18.69	70	28	21.31	0.88
gyro	30	10	1.13	35	16	1.43	0.79
t3dl	30	1878	125.62	90	33	6.69	18.79
c-71	30	4	0.70	25	17	1.38	0.51
Si5H12	30	192	17.99	115	34	9.34	1.93
SiO	30	161	24.32	100	37	13.72	1.77
dawson5	30	1052	135.75	105	34	14.79	9.18
H2O	30	623	168.46	130	40	42.01	4.01
F2	30	27	15.04	50	21	15.42	0.97
oilpan	30	28	10.63	45	24	11.97	0.89
shipsec1	30	26	20.89	50	30	30.58	0.68
bmw7st_1	30	1	1.06	15	5	1.27	0.83
SiO2	30	699	805.68	145	45	207.74	3.88
shipsec5	30	53	59.31	75	27	60.41	0.98
Si4Ge4IH72	30	411	679.91	150	40	274.68	2.48
bmw3_2	30	3	4.20	25	19	12.93	0.33
Ga4IAs4IH72	30	10000	NOT CONV.	150	44	204.05	Infinity

SpMxV AT Effect : Xabclib_LANCZOS



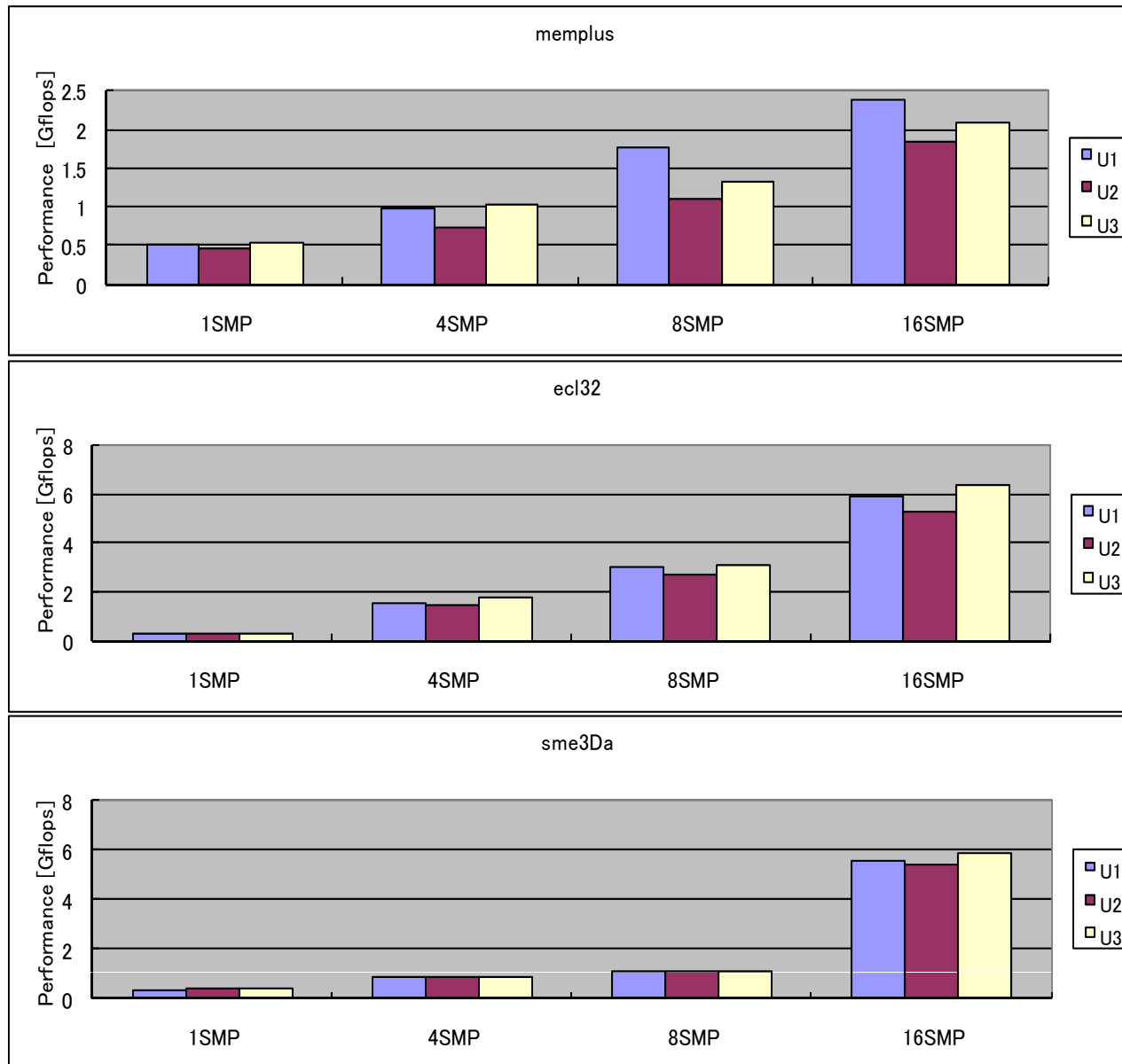
All AT Effect: Xabclib_LANCZOS

Matrices	Non AT			AT			Speedup with AT
	M	restart	time(sec)	M	restart	time(sec)	
vibrobox	30	24	1.13	35	20	1.12	1.01
Lin	30	1047	601.65	150	54	190.48	3.16
cfdl	30	55	12.42	80	24	11.75	1.06
cf2	30	45	18.69	70	28	21.35	0.88
gyro	30	10	1.13	35	16	0.83	1.36
t3dl	30	1878	125.62	90	33	5.59	22.46
c-71	30	4	0.70	25	17	1.17	0.60
Si5H12	30	192	17.99	115	34	8.10	2.22
SiO	30	161	24.32	100	37	14.48	1.68
dawson5	30	1052	135.75	105	34	14.68	9.25
H2O	30	623	168.46	130	40	35.34	4.77
F2	30	27	15.04	50	21	8.74	1.72
oilpan	30	28	10.63	45	24	9.72	1.09
shipsec1	30	26	20.89	50	30	29.73	0.70
bmw7st_1	30	1	1.06	15	5	0.85	1.25
SiO2	30	699	805.68	145	45	137.32	5.87
shipsec5	30	53	59.31	75	27	36.35	1.63
Si41Ge41H72	30	411	679.91	150	40	160.70	4.23
bmw3_2	30	3	4.20	25	19	8.63	0.49
Ga41As41H72	30	10000	NOT CONV.	150	44	144.12	Infinity

Restart Frequency AT Effect : Xabclib_GMRES

Matrices	Fixed Restart Frequency			Auto-tuning			Speedup with AT
	M	restart	time(sec)	M	restart	time(sec)	
chem_master1	30	22	2.55	52	14	2.25	1.13
torso2	30	1	0.68	7	2	0.31	2.19
torso1	30	1	2.54	2	1	0.72	3.53
torso3	30	12	33.57	32	14	34.11	0.98
memplus	30	5	0.25	22	10	0.20	1.25
ex19	30	1000	NOT CONV.	100	60	26.23	Infinity
poisson3Da	30	3	0.48	17	7	0.54	0.89
airfoil_2d	30	7	0.73	22	14	0.83	0.88
poisson3Db	30	7	10.38	17	14	11.03	0.94
viscoplastic2	30	19	2.93	37	15	1.70	1.72
xenon1	30	30	20.16	62	19	16.18	1.25
xenon2	30	40	92.96	72	20	64.29	1.45
wang4	30	5	0.37	17	9	0.29	1.28
ec132	30	1000	NOT CONV.	92	22	11.61	Infinity
sme3Da	30	670	215.49	100	90	101.33	2.13
sme3Db	30	1000	NOT CONV.	100	120	377.29	Infinity
sme3Dc	30	1000	NOT CONV.	100	122	575.63	Infinity
epb1	30	11	0.38	32	14	0.35	1.09
epb2	30	3	0.22	12	9	0.21	1.05
epb3	30	11	3.22	42	14	3.02	1.07

SpMxV AT Effect: Xabclib_GMRES

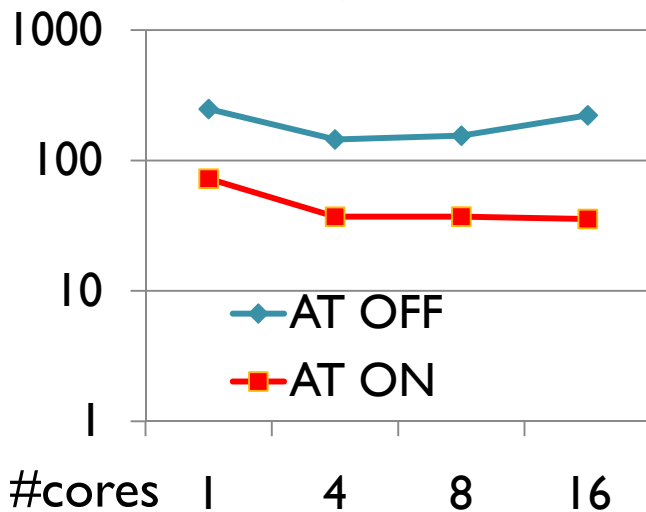


All AT Effect : Xabclib_GMRES

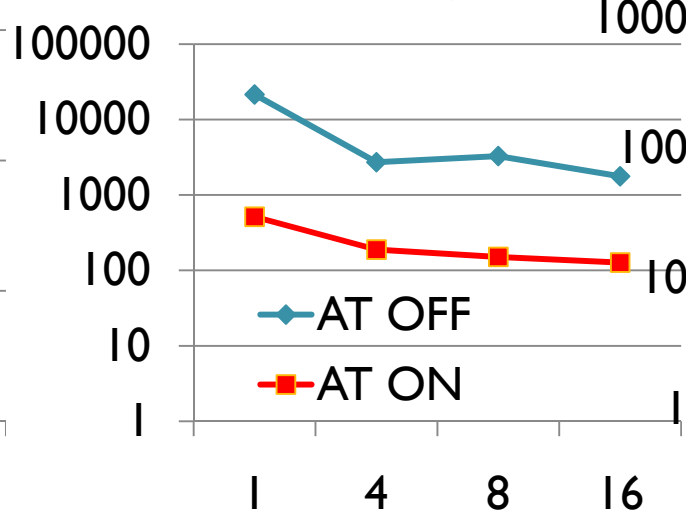
Matrices	Non AT			AT			Speedup with AT
	M	restart	time(sec)	M	restart	time(sec)	
chem_master1	30	22	2.55	52	14	2.22	1.15
torso2	30	1	0.68	7	2	0.30	2.27
torso1	30	1	2.54	2	1	0.72	3.52
torso3	30	12	33.57	32	14	33.76	0.99
memplus	30	5	0.25	22	10	0.20	1.25
ex19	30	1000	NOT CONV.	100	54	21.28	Infinity
poisson3Da	30	3	0.48	17	7	0.54	0.89
airfoil_2d	30	7	0.73	22	14	0.83	0.88
poisson3Db	30	7	10.38	17	14	11.04	0.94
viscoplastic2	30	19	2.93	37	15	1.74	1.68
xenon1	30	30	20.16	62	19	16.47	1.22
xenon2	30	40	92.96	72	20	64.31	1.45
wang4	30	5	0.37	17	9	0.29	1.28
ec132	30	1000	NOT CONV.	92	22	11.66	Infinity
sme3Da	30	670	215.49	100	87	97.48	2.21
sme3Db	30	1000	NOT CONV.	100	120	380.39	Infinity
sme3Dc	30	1000	NOT CONV.	100	98	531.37	Infinity
epb1	30	11	0.38	32	14	0.34	1.12
epb2	30	3	0.22	12	9	0.21	1.05
epb3	30	11	3.22	42	14	3.00	1.07

Parallel Performance (LANCZOS)

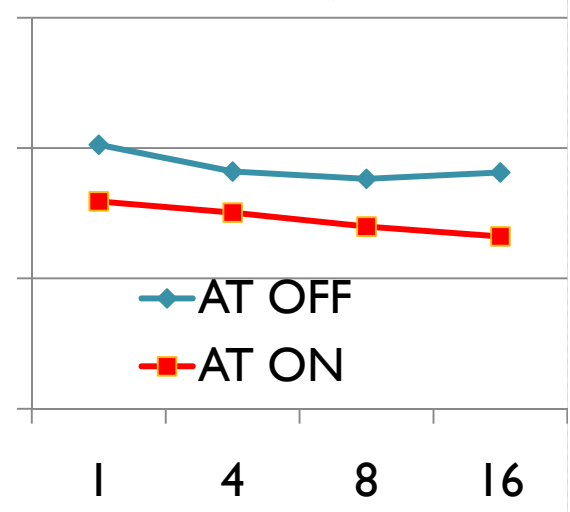
H2O.rsa, M=30



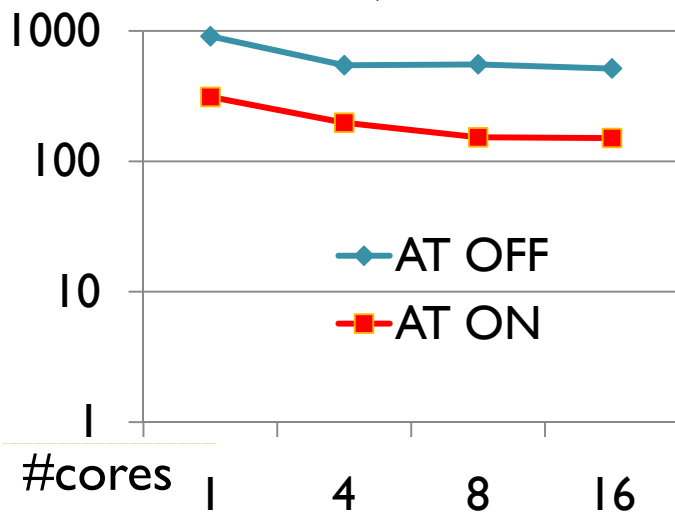
Ga4As4H72.rsa, M=30



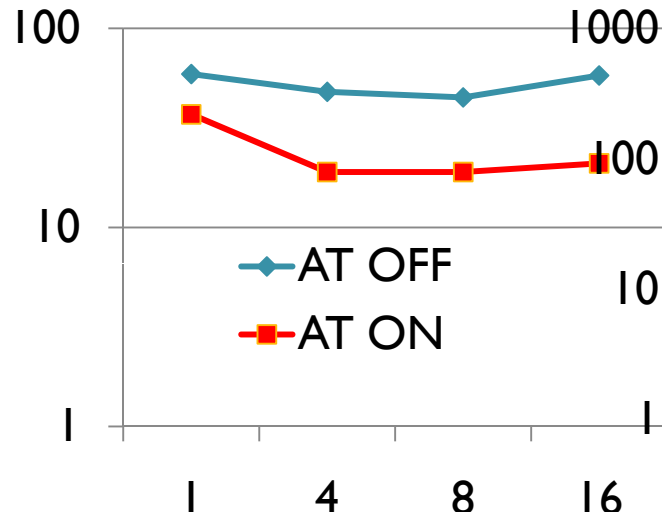
SHIPSEC1.rsa, M=11



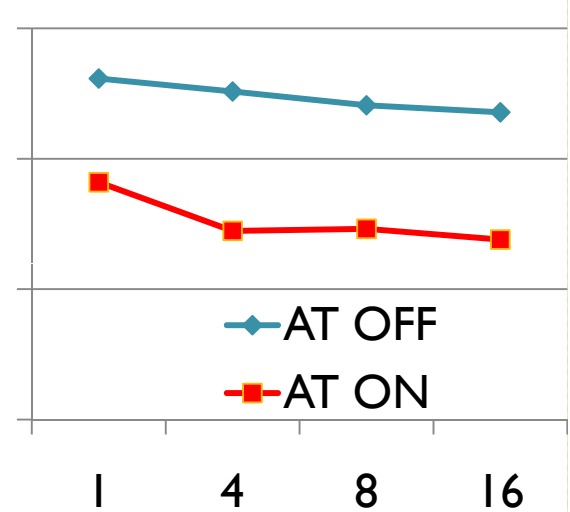
SiO2.rsa, M=30



cfd2.rsa, M=11



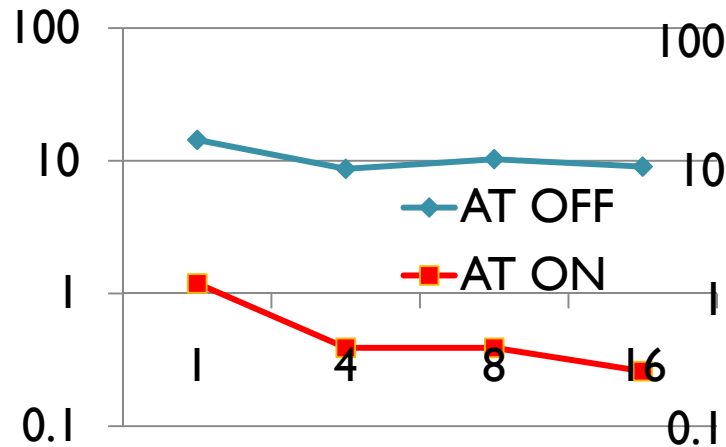
C-73.rsa, M=20



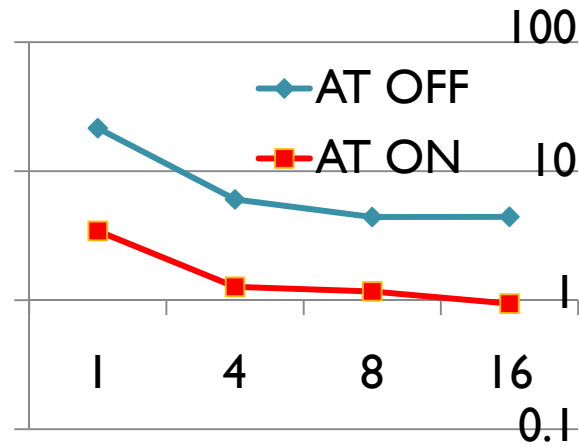
Averaged Sp: 1.98x (ideal: 16x)

Parallel Performance (GMRES, Jacobi Preconditioner)

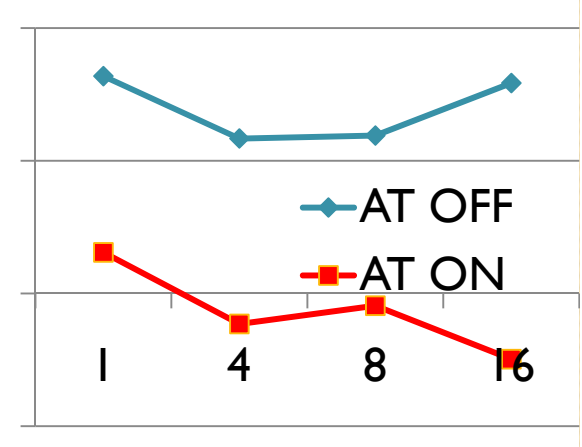
Memplus.rcs, M=1



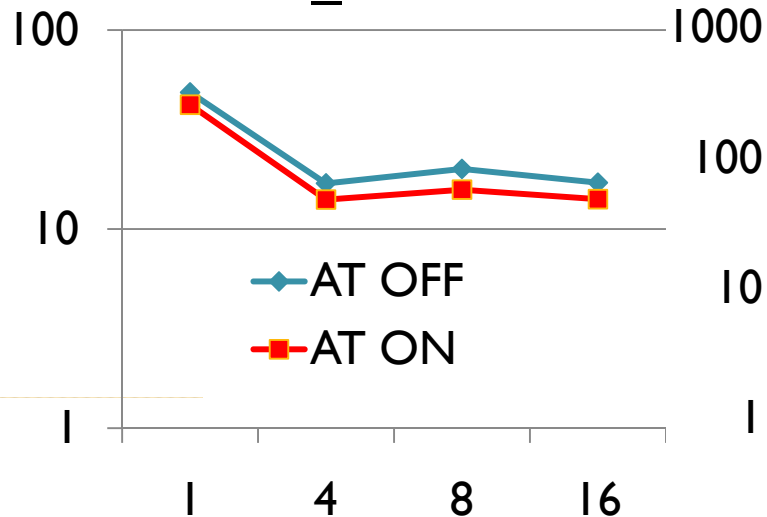
chipccol0.rcs, M=2



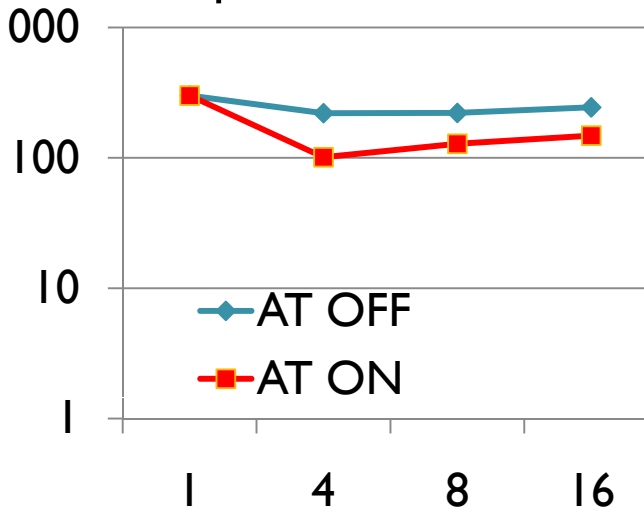
wang3.rcs, M=2



chem_master1.rcs, M=2



epb3.rss, M=60



**Averaged Sp:
2.59x
(ideal: 16x)**



既存アプローチとの違い

従来方式との違い（数値ライブラリ）

ソフトウェア	AT実装 再利用性	汎 用 性	A T 機 能	適 用 局 面	対応 AT要因	備考
Open AT Lib	○	○	高	中 ～ 広	S要因 D要因	•ATポリシー機能で適用局面増加 •対象処理が特化： ATオーバーヘッド低、AT品質が高い •ユーザレベル実現 •静的ライブラリ提供
ATLAS, OSKI	× BLAS のみ	×	中	狭	S要因用 D要因用 と 分離	•メモリ、精度のAT未対応 •S要因用（ATLAS） •D要因用（OSKI）

他のアプローチとの比較

ソフトウェア	数値計算への適用	A T 機能	対応 AT 要因	備考
Open AT Lib	○	高	S要因 D要因	•ATポリシー機能で適用局面増加 •対象処理が特化： ATオーバーヘッド低、 AT品質が高い •ユーザレベル実現 •静的ライブラリ提供
ABCLib Script / ROSE / 反復解法可 視化ツール	○ ※数値 ライブラリ に限定 しない	高	S要因 D要因	•ATオーバーヘッド高 (知識がない人が使う場合) •汎用的性能モデル構築困難： AT品質が場合により低下 •可視化ツール： 数値計算ライブラリ接続性
Active Harmony	△	低	D要因 のみ	•OSカーネル限定 •ユーザレベルで実現できない？ •ATオーバーヘッド高

OpenATLibの今後の実装課題

- 疎行列-ベクトル積のA T
 - 負荷分散の改良：
 - 例. Segmented Scan形式の採用
： 普通の CRS形式よりメモリ空間が必要
 - A Tの候補を増やす
 - アンローリング段数、レジスタブロッキング
- アルゴリズム選択
 - Krylov部分空間法の直交化アルゴリズム
 - 例. Modified G-S, Classical G-S, IRCG-S, Householder
： 演算精度と実行時間とのトレードオフ
- 並列実装方式自動選択 (MPI)
 - コレクティブ通信を使う実装
 - 1対1通信を使う実装
- OpenATLibの提供機能の充実化

おわりに

- AT機構を再利用し、ATソフトウェアの生産性を高める概念（**AT実装再利用性**）を提案した
- AT機構に必要とされる機能は
 1. 性能パラメータ最適化機能
 2. ユーザ知識設定機能
 3. **ユーザポリシー設定機能**
であり、ユーザポリシー設定機能は先進的、かつ、AT適用の局面増大に資する
- AT機構に必要とされる機能の一部を実装したOpenATLibを開発
 - Xabclibは、OpenATLibを利用して開発
- OpenATLib、Xabclibのソースコード（α版，ユーザポリシー機能実装なし）は、PCクラスタコンソーシアム経由で本年11月ごろ配布予定である

謝辞

- 自動チューニング研究会（順不同）

1. 主査：須田礼仁（東大）
2. 幹事（総務）：片桐孝洋（東大）
3. 幹事（連絡）：伊藤祥司（理研）
4. 顧問：弓場敏嗣（電通大名誉教授）
5. 今村俊幸（電通大）
6. 山本有作（名大）
7. 直野健（（株）日立製作所中央研究所）
8. 清水謙多郎（東大）
9. 佐藤周行（東大）
10. 岩下武史（京大）
11. 寺内和也（日本ビジュアルニューメリックス）
12. 恵木正史（（株）日立製作所中央研究所）
13. 櫻井隆雄（（株）日立製作所中央研究所）
14. 黒田久泰（愛媛大）
15. 中島研吾（東大）
16. 滝沢寛之（東北大）
17. 高橋大介（筑波大）
18. 八杉昌宏（京大）

