



シームレス高生産・高性能 プログラム環境 チュートリアル

自動チューニング機構付き 数値ライブラリXabclib

片桐孝洋(東大) 黒田久泰(愛媛大/東大)

共同研究者:

櫻井隆雄、直野健(日立中研)、中島研吾(東大)

PCクラスタワークショップin京都2010
2月18日(木)15:30~16:30
キャンパスプラザ京都 4F第3講義室

チュートリアルの流れ

- (講師：片桐)
 - 開発背景
 - Xabclib概要紹介
 - 汎用的自動チューニング用API OpenATLib
 - 自動チューニング機能
 - ユーザポリシー設定機能
 - プログラム例
 - 典型性能
- (講師：黒田)
 - Xabclib利用法
 - OpenATI_LINEAR SOLVEの使い方

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～

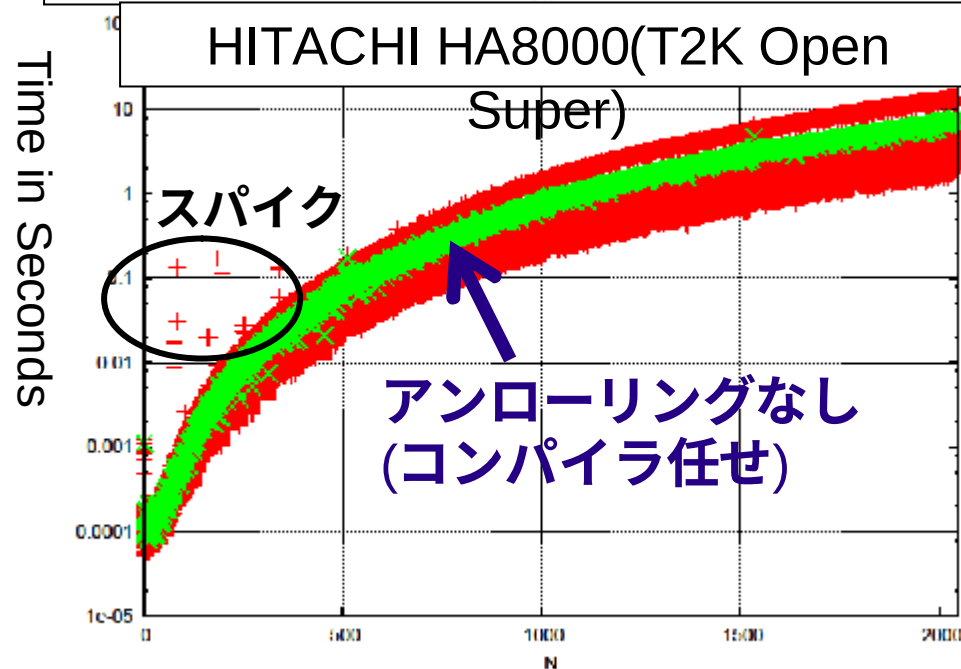
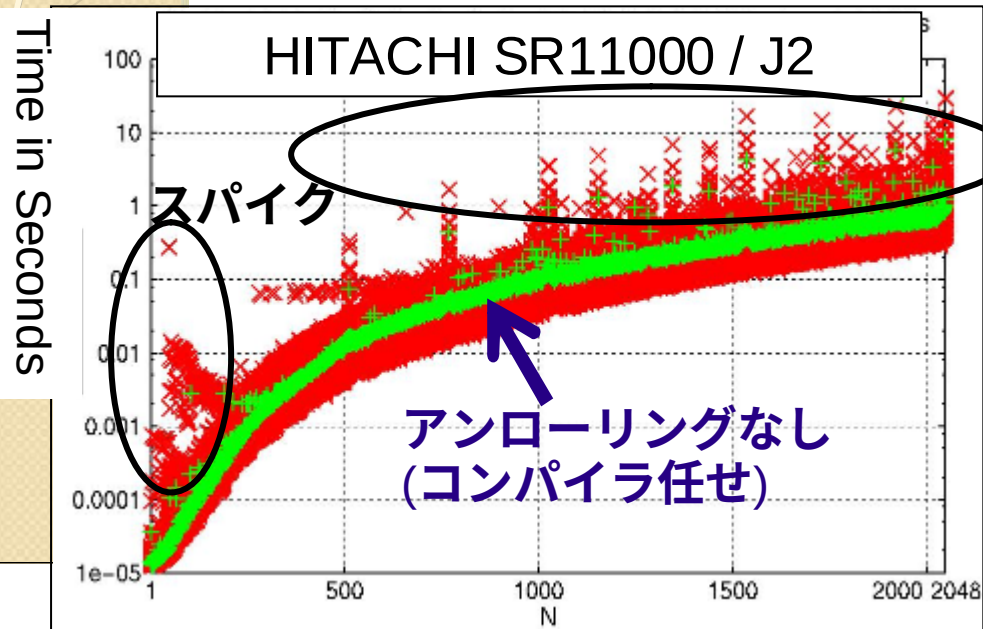
- 並列実行モデルの変化

- 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 SR11000/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

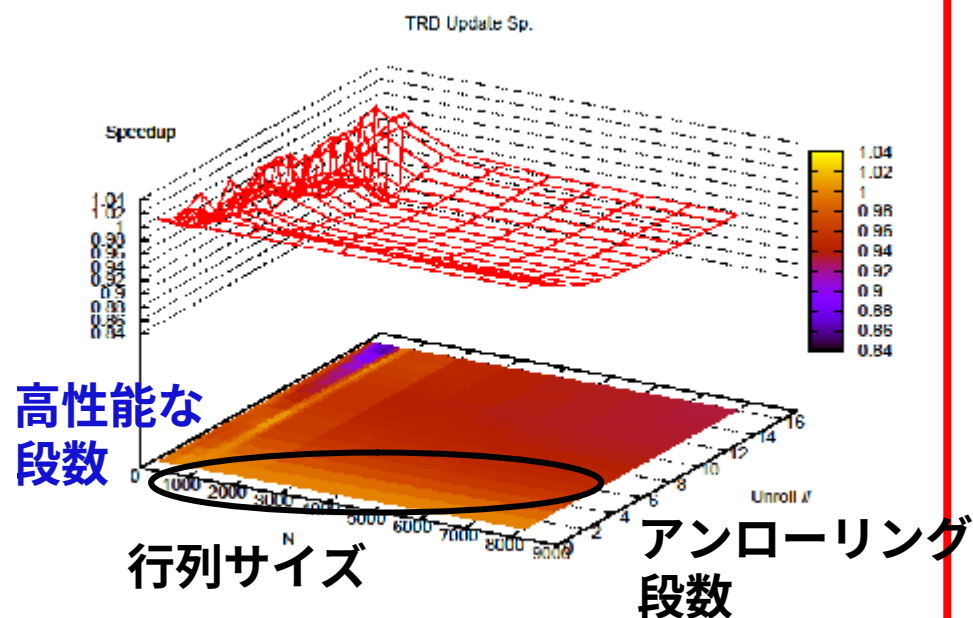
●10倍ぐらい実行時間がぶれる
(実行時間の<不安定性>)

●場合により100倍も遅い!

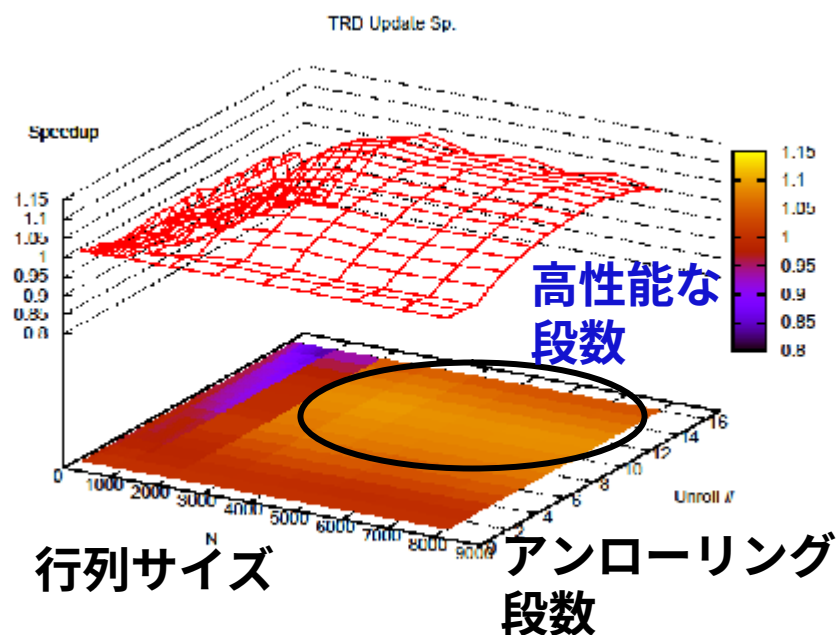
マルチコアでの実行形態の違いによる 性能挙動の劇的な変化例 (T2Kオープンスパコン)

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！



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

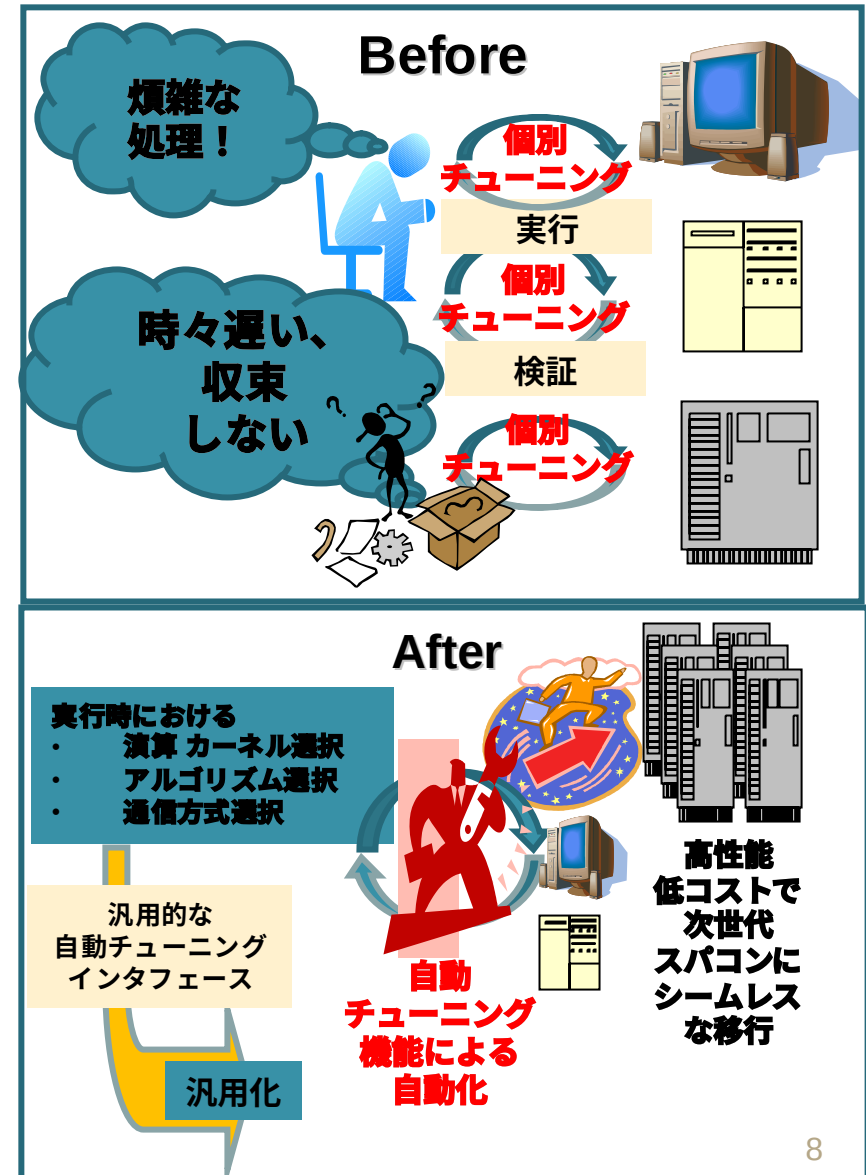
- 目標

- 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)



Xabclib（β版）の特徴

- 汎用的自動チューニングAPIの提供
 - 自動チューニング機能の再利用性を考慮した、汎用的APIのOpenATLibを提供
- 世界に先駆けた先進的AT機能の提供
 1. 自動チューニングポリシー設定機能
 - 速度、メモリ量、演算精度の方針設定機能
 2. マルチコア向きの負荷バランスの改善が可能な疎行列演算方式
 - Branchless Segmented Scan (BSS) 方式
 3. 直交化アルゴリズム動的選択機能

OpenATLibの位置づけ

- 動的選択する自動チューニングインターフェースをATライブラリ開発者に提供
- 性能パラメタ最適化機能の以下の機能
 - 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
 - 直交化方式選択
 - 実装方式選択
 - 疎行列 - ベクトル積 (SpMxV)
 - 負荷バランス改良、ベクトル計算機向け、並列化向き
 - 計算機資源選択
 - 実行時のメモリ量を考慮した SpMxVとリスタート周期
- ユーザポリシー設定機能
 - 速度・メモリ量・演算精度

Nomenclature of OpenA I Lib functions

First Character	The character shows data type. ● S : Single Precision ● D : Double Precision
Second and Third Characters	If the function is auxiliary, it comes "AF". If the function is computation, it comes matrix kinds in the second character, and matrix storage format in the third character. ● The second character: ● S : Symmetric. ● U : Non-symmetric. ● D : Diagonal. ● T : Tridiagonal. ● The third character: ● R : CRS Format. ● C : CCS Format.
Fourth and Fifth Characters	Process Kinds. ● MV : Matrix-vector multiplication.

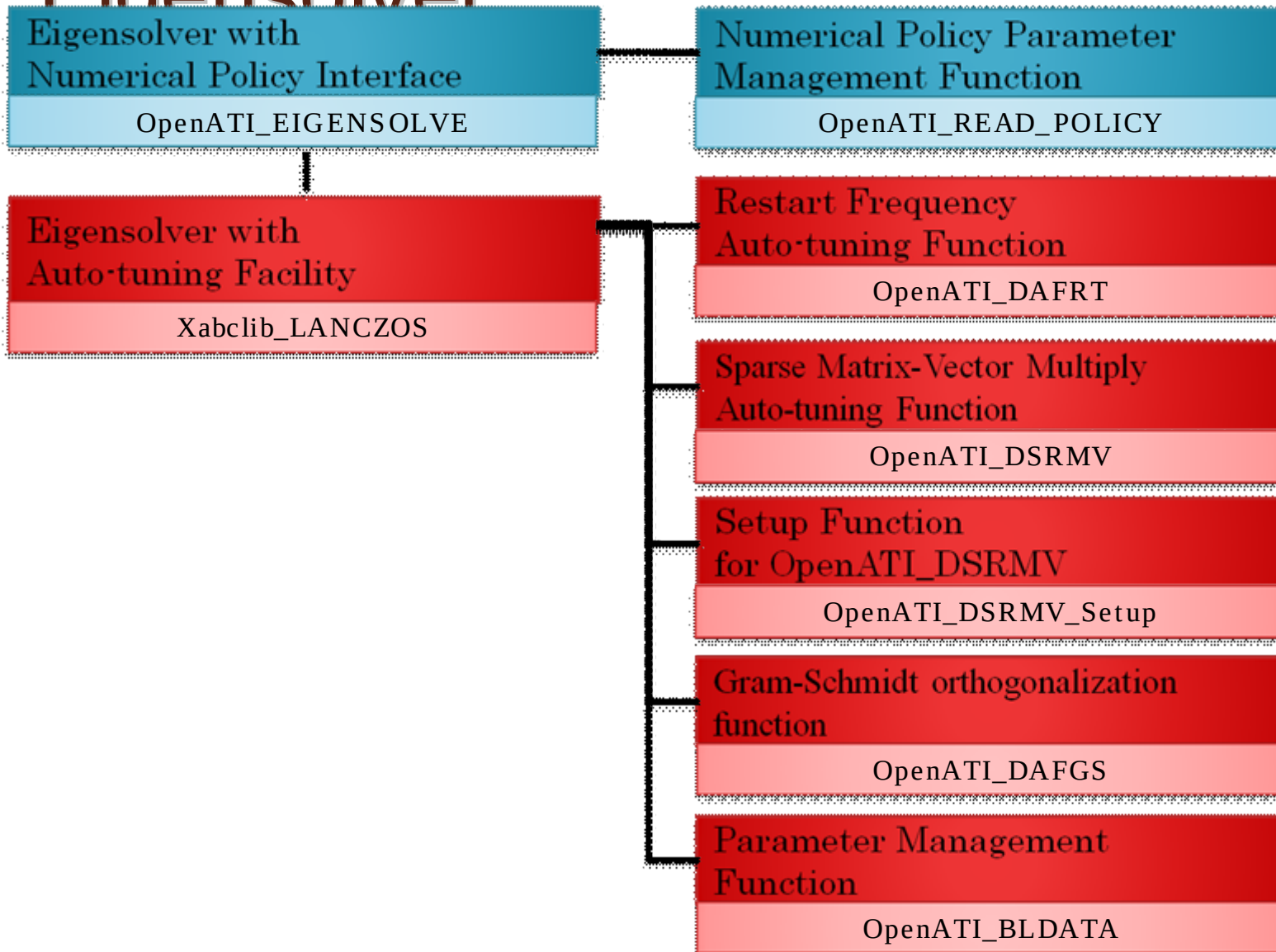
Main Auto-tuning Function Providing OpenATLib (Xabclib ver. Beta)

Function Name	Description
OpenATI_DAFRT	Judge increment for restart frequency on Krylov subspace.
OpenATI_DSRMV	Judge the best implementation for double precision symmetric SpMxV on CRS format.
OpenATI_DURMV	Judge the best implementation for double precision non-symmetric SpMxV on CRS format.
OpenATI_DSRMV_Setup	Setup function for OpenATI_DSRMV.
OpenATI_DURMV_Setup	Setup function for OpenATI_DURMV.
OpenATI_DAFGS	Gram-Schmidt orthogonalization function with 4 implementations.
OpenATI_BLDATA	Set default parameters. (Block data format for Fortran.)
OpenATI_LINEARsolve	Over-Linear Solver with numerical policy interface.
OpenATI_EIGENSolve	Over-Eigen Solver with numerical policy interface.

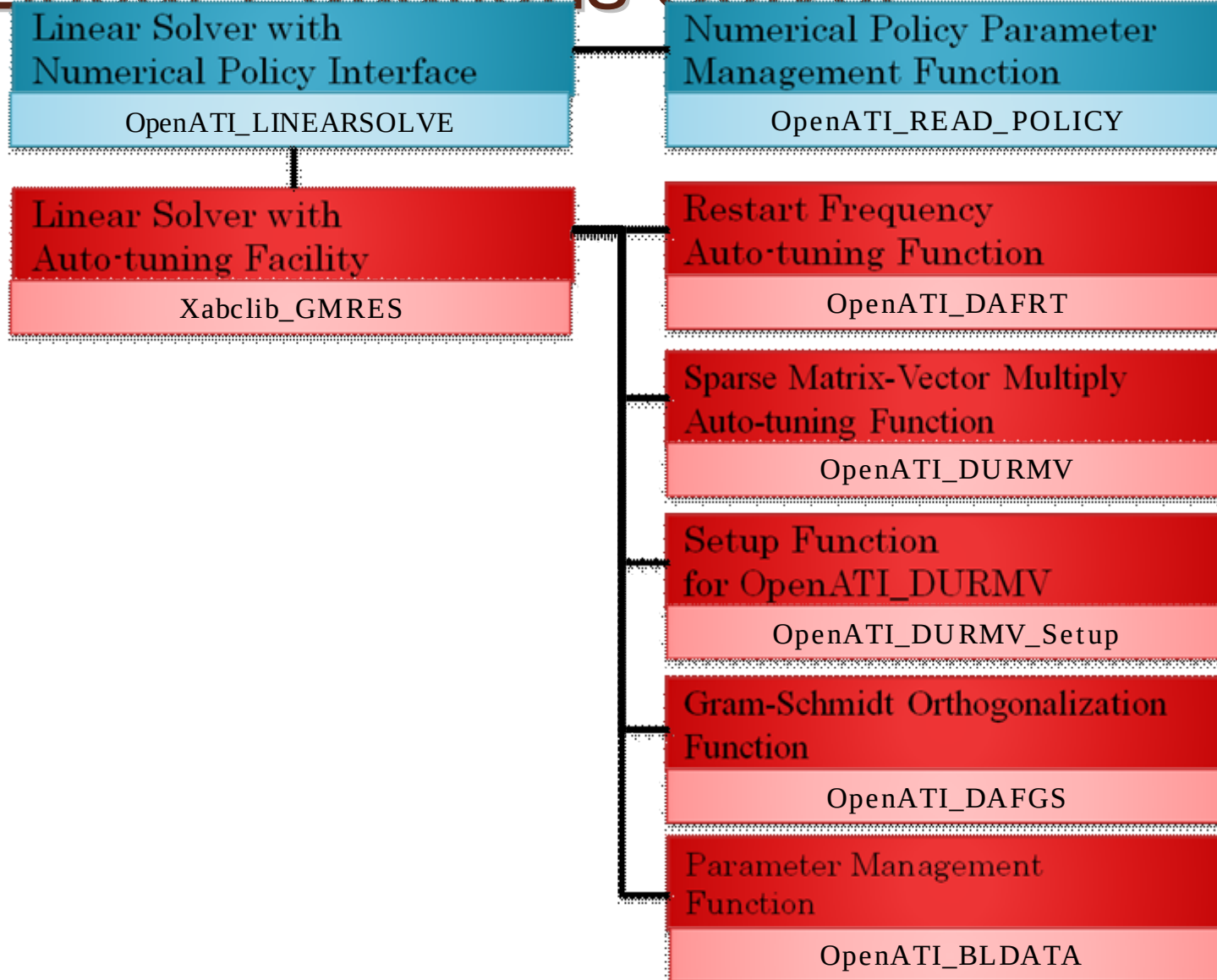
Red functions are newly developed in the version Beta.

Blue functions are modified API from the version Alpha.

OVERVIEW OF LIBRARY Construction: Eigensolver



OVERVIEW OF Library Construction: Linear Equations Solver



OpenATlib: Restart Frequency Interface

- **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.

An Example of OpenATI_DAFRT

```
INCLUDE "OpenAT.inc" // Include OpenAT.inc
MSIZE=1              // Initial restart frequency.
I=5                  // Judgment frequency.
DO ITER=1, MAX_ITER
  ~ omission ~
  IF RSDID < TOL RETURN // Convergence Test
  SAMP (K)=RSDID //Set residual to SAMP(K).
  IF (mod (K, I) .eq. 0) THEN //Call DAFRT per I times.
    IRT=0
    CALL OpenATI_DAFRT (I, SAMP,IRT,INFO)
    IF IRT= 1 MSIZE=MSIZE+1 //Increase restart
frequency.
    K=0
  END IF
  K=K+1
  ~ omission ~
```

ENDDO

OpenATlib: SpMxV Interface

- **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**
 - INF(LINF) : Groups of rows processed each thread
 - LINF : The Size of INF
 - INFO : Error Code

Red arguments are Auto-tuning facility functions.

Blue arguments are newly added from the version Alpha.



Overview of Auto-tuning Method for SpMxV

- **OpenATI_DSRMV**
 - **S1)** Row Decomposition Method.
 - **S2)** Normalized NZ Method.
 - **S3)** Normalized NZ Method, with vector reduction parallelization.
- **OpenATI_DURMV**
 - **U1)** Row Decomposition Method.
 - **U2)** Normalized NZ Method
(for scalar multi-core processors).
 - **U3)** Modified Segmented Scan
(Branchless Segmented Scan (BSS))
(for scalar multi-core processors).
 - **U4)** Original Segmented Scan
(for vector processors).

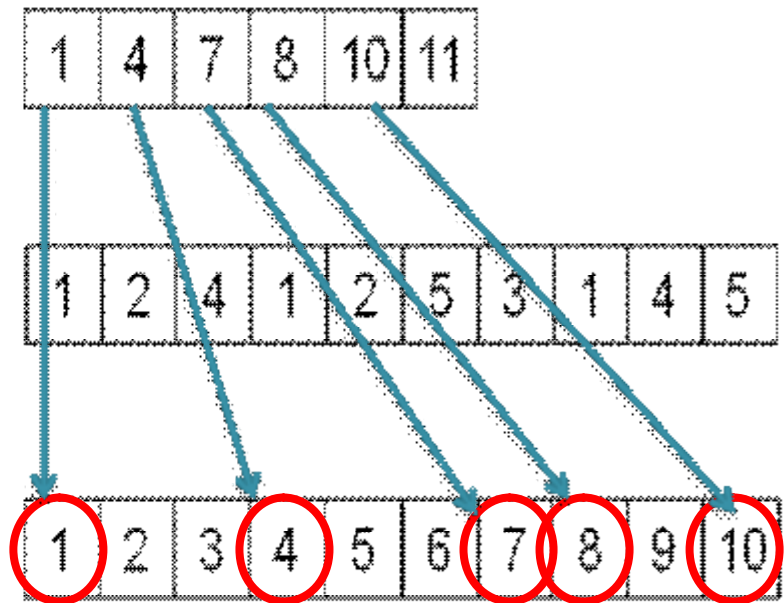
Compressed Row Storage (CRS) for Non-symmetric Matrices

1	2	0	3	0
4	5	0	0	6
0	0	7	0	0
8	0	0	9	0
0	0	0	0	10

Pointers to first row elements.

= Row indexes for non-zero elements.

Values for non-zero elements.



○ first element in each row

Row Decomposition Method and Normalized NZ Method on CRS Format (Both Symmetric and Unsymmetric)

* : Non-Zero Elements

Owned

#Elements

Thread 1

*	*			*		3
—	—			—		

Thread 2

	*		*	*		
	—		—	—		
		*		*	*	6
		—		—	—	

Thread 3

			*			1
			—			

Thread 4

				*		
				—		
					*	2
					—	

Row Decomposition Method
(Conventional)

(SW: S1, U1)

Input Matrix is divided into the number of threads blocks for balancing the number of row processed by each thread

Owned #Elements

Thread 1

*	*			*		3
—	—			—		

Thread 2

	*		*	*		3
	—		—	—		

Thread 3

		*		*	*	3
		—		—	—	

Thread 4

			*			
			—			
				*		
				—		
					*	3
					—	

Normalized NZ Method
(SW: S2, S3, U2)

Input matrix is divided into the number of threads blocks for normalizing the number of non-zero element processed by each

Segmented Scan and BSS (Unsymmetric Only)

input matrix

1	0	0	2	0	0	0	0
0	3	0	0	0	4	5	6
7	8	0	9	0	10	0	0
11	0	12	13	0	0	14	0
0	0	0	0	15	0	16	17
18	19	20	21	22	23	24	25
0	26	0	27	0	0	0	0
0	0	0	0	0	0	28	0
29	0	0	0	0	30	0	0

row pointer (IRP)

1	3	7	11	15	18	26	28	29	31
---	---	---	----	----	----	----	----	----	----

Each 6 element is divided.
Not row-base
management.

segment-vector #3

Modified Segmented Scan (BSS)

MFLAG

1
3
7
11
13
15
18
19
25
26
28
29
31

JFSTART

0
2
4
7
8
12

JFSTART
holds
first elements
information.

Needs no IF-sentence
in the most inner

Access Direction

○ first element
in each row

Original Segmented Scan

FLAG

T	T	F	F	F
F	F	F	F	T
T	F	T	F	F
F	F	F	F	T
F	T	F	F	T
F	F	T	F	F

"T" means
first element
in each row.

Access Direction

SpMxV Reduction Optimization

(Sw: S3, Symmetric Only)

- There is computation with “zero” values.
- To reduce the zero computation, we have information for indexes from start location to end location for non-zero values.

- The computation columns:
From Second
To Third.

WK

(a Workspace array for parallelization)

*	0	0	0
*	0	0	0
*	*	0	0
*	*	0	0
0	*	*	0
0	*	*	0
0	*	*	*
0	*	*	*

An Example of OpenATI_DSRMV

```
INCLUDE "OpenAT.inc"           // Include OpenAT.inc
OpenATI_DSRMV_IPARM_1=3        //Initialize DSRMV parameter.
ICASE=0                        //Initialize DSRMV parameter.
LSINF= N+NUM_SMP+3
ALLOCATE(SINF(LSINF))
    ~ omission ~
DO ITER=1, MAXITER
    //The first SpMxV.
    CALL OpenATI_DSRMV (N, NNZ, IRP, ICOL, VAL, X, Y, ICASE,
NUM_SMP,
        WK, SIN, LSINF, INFO)
    OpenATI_DSRMV_IPARM_1=1 //Hereafter, we select the best one.
    ~ omission ~
    // SpMxV after run-time searching.
    // We can use the best implantation based on previous information.
    CALL OpenATI_DSRMV (N, NNZ, IRP, ICOL, VAL, X, Y,
ICASE,NUM_SMP, WK,
        SIN,LSINF,INFO)
    ~ omission ~
ENDDO
```

OpenATlib: Re-orthogonalization Selection

- **Re-orthogonalization Selection Interface**

OpenATI_DAFGS

- **Arguments:**

- NORMALFLG : Normalization of Output vector
 - 0 : not normalized
 - 1 : normalized
- N : Vector Length
- X(N) : The vector to re-orthogonalize
- Q(LQ,MM) : Orthogonalized vectors Q(1:N,MM)
- LQ : Leading Dimension of Q
- MM : The number of vector of Q
- HR(MM) : Inner product X by Q(1:N,M)
- IDGKS : Iterative refinement of DGKS
 - 0 : no Iterative refinement
 - 1 : Iterative refinement
- INFO : Error Code



Overview of Auto-tuning Method for Re-orthogonalization

- Classical Gram-Schmidt method (CGS)
 - Accuracy of reorthonormalization is lower.
 - Parallel performance is excellent.
- DGKS method
 - Supplies improved accuracy by running CGS 2 times.
- Modified Gram-Schmidt method (MGS)
: Default Algorithm
 - Most effective performance and accuracy in many cases.
- Blocked Classical Gram-Schmidt method (BCGS)
 - Orthonormalized by intra-block with CGS, and by inter-block with MGS.



OpenATlib: Eigensolver Selection

- **Eigensolver Selection Interface**
OpenATI_EIGENSOLVE

- **Arguments:**

- N : The number of dimension for the matrix
- NNZ : The number of non-zero elements for the upper triangle part
- IRP(N+1) : Pointes to diagonal elements on each row
- ICOL(NNZ) :
The row indexes for non-zero elements on the upper triangle part
- VAL(NNZ) :
The values for non-zero elements on the upper triangle part
- IORDER : Option parameter for eigensolve
 - 1 : Compute eigenvalues and eigenvectors from the raw value, that means including minus.
 - 2 : Compute eigenvalues and eigenvectors from the absolute value
- NEV : The number of eigenvalues you need
- EV(NEV) : The eigenvalues
- EVEC (N,NEV) : The eigenvectors. The k-the eigenvector corresponding to the eigenvalue EV(k) is set to the k-th column.
- INFO : Error Code

OpenATlib: Linear Equations Solver

Selection

• **Linear Equations Solver Selection Interface**

OpenATI_LINEARSOLVE

◦ Arguments:

- N : The number of dimension for the matrix
- NNZ : The number of non-zero elements for the matrix
- IRP(N+1) : Pointes to diagonal elements on each row
- ICOL(NNZ) :
The row indexes for non-zero elements on the matrix
- VAL(NNZ) :
The values for non-zero elements on the matrix
- B(N) : The elements for right hand size vector b
- X(N) :
 - INPUT : Set the elements of initial guess for solution vector x_0
 - OUTPUT : Return the elements of solution vector x
- INFO : Error Code

User Policy Description

- The Overall Format

`<keyword> = <value>`

`<keyword> : = POLICY / CPU / RESIDUAL / MAXMEMORY /
MAXTIME / PRECONDITIONER`

- The POLICY Functions

`POLICY = <value>`

`<value> := TIME / ACCURACY / MEMORY`

“TIME” is selected by default.

- If `POLICY = TIME`

➤ Over-Solvers preference for execution time over accuracy and saving memory. Therefore, algorithms for high performance are positively selected.

- If `POLICY = ACCURACY`

➤ Over-Solvers recalculation solution of solvers. If false convergence occurs, Over-Solvers continue to re-execute with more exact convergence test until true convergence.

- If `POLICY = MEMORY`

User Policy Description

- The CPU Function

CPU = <value>

<value> := entry OMP_NUM_THREADS at run-time.

OMP_GET_NUM_THREADS is selected by default.

Note) $1 \leq \text{<value>} \leq \text{OMP_GET_MAX_THREADS}()$

- The RESIDUAL Function

RESIDUAL = <value>

<value> := entry require accuracy by real value.

The default value is 1.0D-8.

In case of “POLICY = ACCURACY” is set and false convergence occur, solver continue to re-execute with more exact convergence test until true convergence.

- The MAXMEMORY Function

MAXMEMORY = <value>

<value> := entry require memory usage in [Gbyte].

The default value is “memfree” in /proc/meminfo (Linux).

If fails to get property in /proc/meminfo, search and allocate free memory dynamically. Note) The maximum limit of MAXMEMORY is 16Gbyte

User Policy Description

- The MAXTIME Function

MAXTIME = <value>

<value> := entry time tolerance in [sec].

The default value is infinite.

When execution time exceeds time tolerance, computation is stopped.

- The PRECONDITIONER Function

PRECONDITIONER = <value>

<value> := NO / JACOBI / SSOR / ILU0

ILU0 is selected by default.

This keyword is used by only OpenATI_LINEARSOLVE.

- PRECONDITIONER = NO

: No preconditioner

- PRECONDITIONER = JACOBI

: JACOBI

- PRECONDITIONER = SSOR

: SSOR

- PRECONDITIONER = ILU0

: ILU(0)

User Policy Settings

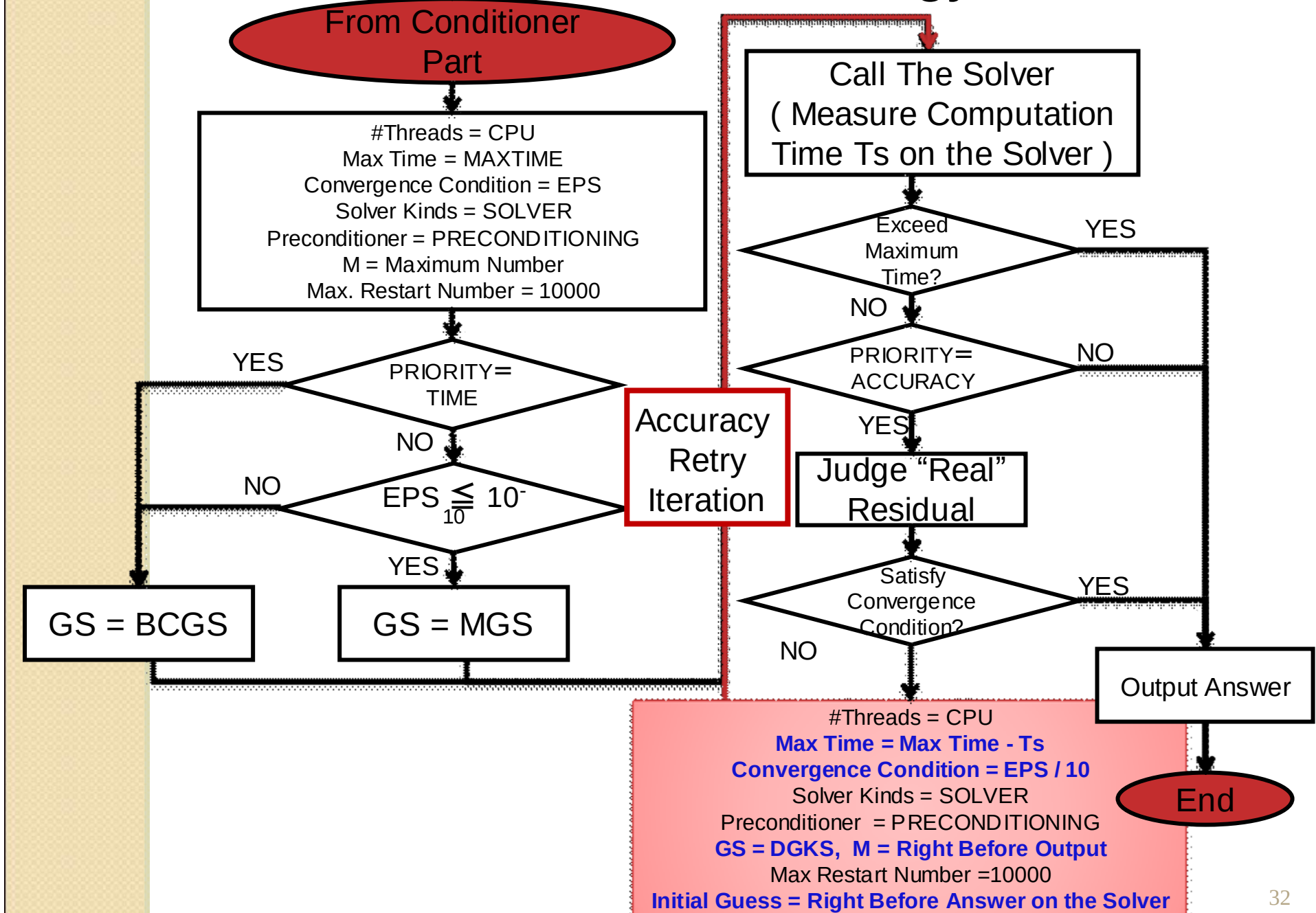
- Prepare “Policy File”
 - Example

```
POLICY = ACCURACY
RESIDUAL = 1.0D-10
CPU = 16
PRECONDITIONER = ILU0
MAXMEMORY = 1.0
MAXTIME = 500.0
```

- OPENATI_POLICY_REPORT.txt

```
*****
**** OpenATI LINEAR SOLVER POLICY REPORT ****
**** 2010.0114 11:30 ****
date / time
*****
[Environment variables]
OPENATI_DEBUG =
OPENATI_POLICY = ./input_policy.dat
[Policy Definitions]
POLICY = ACCURACY
SMPs = 16
SOLVER = XABCLIB_GMRES
PRECONDITIONER = ILU0
REQUIREMENT WORKING MEMORY = 16.00000000000000
<<< Upper Bound 16GBYTE >>>
REQUIREMENT RESIDUAL = 1.000000000000000E-008
REQUIREMENT MAX. TIME = 500.0000000000000
MAX. SUBSPACE SIZE = 14214
RUNTIME MEMORY USE = 3.24 [GBYTE]
KRYLOV SUBSPACE EXPAND AT = 1 MATVEC AT = 1
Initial Gram-Schmidt Strategy = BCGS
===== OPENATI LINEARSOLVE SUCCESSFULLY ENDED =====
[OPENATI LINEARSOLVE RESULT]
MATRIX DATA : N= 14214 NNZ= 259688
FASTEST MATVEC NO. = 11
case
FINAL KRYLOV SUBSPACE SIZE = 42
FINAL Gram-Schmidt Strategy = DGKS
2-Norm of RHS = 25.2388589282479
NUMBER OF RETRYED GMRES = 6
TOTAL RESTARTS of GMRES = 197
RESIDUAL NORM = 3.005885687924543E-010
SET-UP TIME = 1.126790046691895E-002 [SEC]
SOLVER TIME = 1.32032704353333 [SEC]
TOTAL TIME = 1.33159494400024 [SEC]
```


POLICY=ACCURACY Strategy





性能評価

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

- OpenATI_EIGENSOLV

Convergence Test Value	1.0E-08
Calculated Eigenvalues	10
Max Execution Time	1000 seconds

- OpenATI_LINEAR SOLVE

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)
Max Execution Time	1000 seconds

- Default Restart Frequency

30 : Same as PETSc default value

Test matrices for OpenATI_EIGENSOLVE

- From University Florida Sparse Matrix Collection (21 kinds)

Matrix	N	NNZ	Field
vibrobox	12328	177578	Acoustics
Lin	256000	1011200	Chemistry
cf1	70656	949510	Fluid Dynamics
cf2	123440	1605669	
gyro	17361	519260	Model Reduction
t3d1	20360	265113	
c-71	76638	468096	Optimization
c-73	169442	724348	
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 OpenATI_LINEAR SOLVE

● From University Florida Sparse Matrix Collection

(22 kinds)

Matrix	N	NNZ	Field
chipcool0	20082	281150	2D/3D
chem_master1	40401	201201	
torso1	116158	8516500	
torso2	115067	1033473	
torso3	259156	4429042	
memplus	17758	126150	Electric circuit
ex19	12005	259879	Fluid dynamics
poisson3Da	13514	352762	
poisson3Db	85623	2374949	
airfoil_2d	14214	259688	
viscoplastic2	32769	381326	Materials

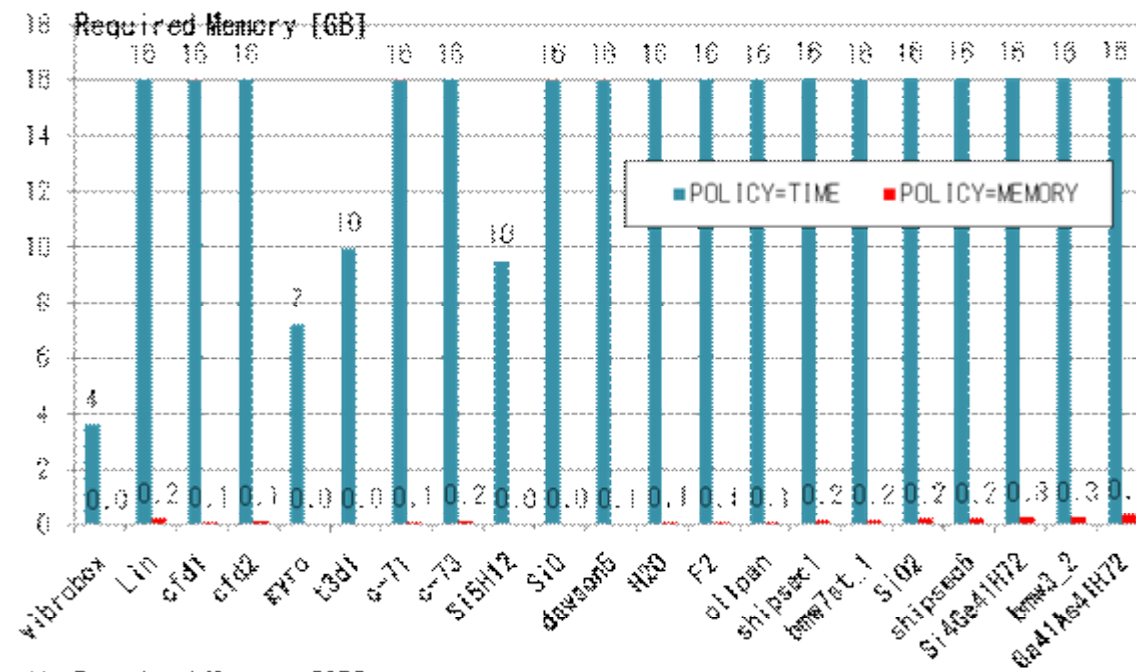
Matrix	N	NNZ	Field
xenon1	48600	1181120	Materials
xenon2	157464	3866688	
wang3	26064	177168	Semiconductor device
wang4	26068	177196	
ec132	51993	380415	
sme3Da	12504	874887	Structural
sme3Db	29067	2081063	
sme3Dc	42930	3148656	
epb1	14734	95053	Thermal
epb2	25228	175027	
epb3	84617	463625	



Policy Memory Optimization Condition

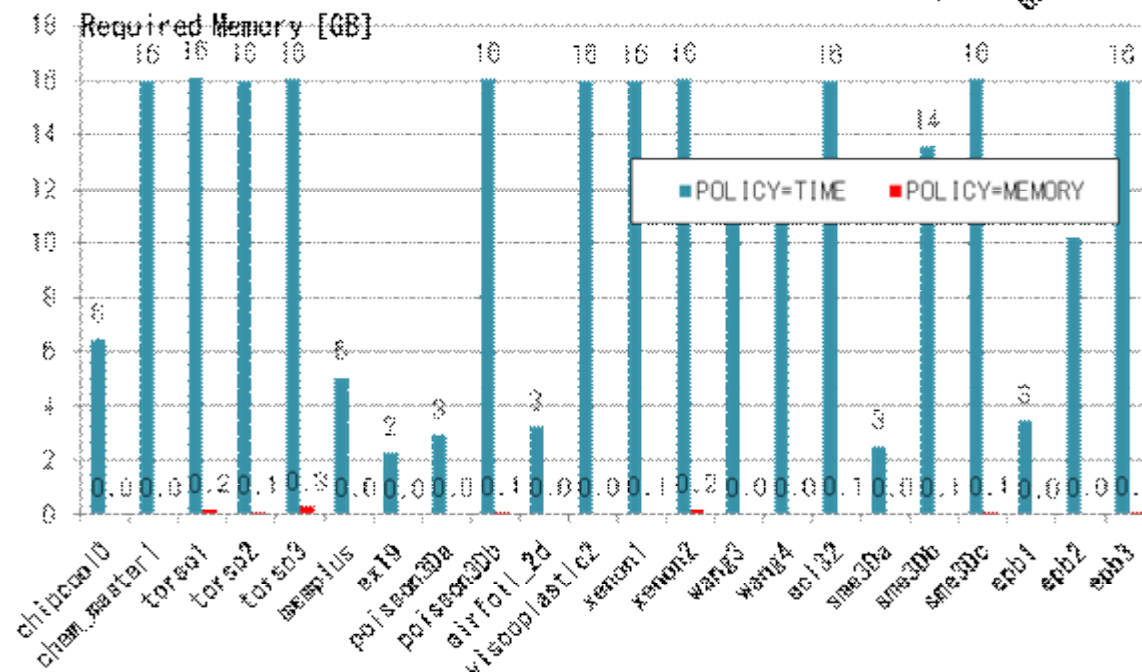
- Max Restart Frequency
 - POLICY=MEMORY : 100 (Constant Value)
 - POLICY=TIME : The maximum value up to 16GByte. The number of matrix size is theoretical maximum.
- GS Method is fixed as BCGS
- SpMxV Selection
 - Symmetric
 - All kinds of SpMxV are checked.
 - Unsymmetric
 - Two kinds of SpMxV without Segmented Scan are checked.

Memory Policy Effect



OpenATI_EIGENSOLVE

Averaged Speed-down:
1.24x
Worst Speed-down:
6.12x (Ga41As41H72)



OpenATI_LINEARsolve

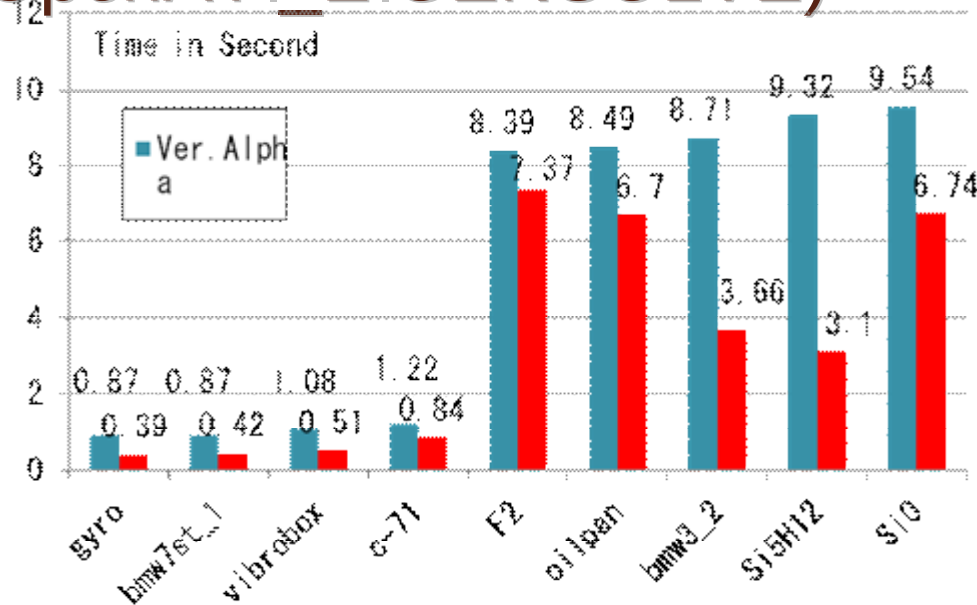
Averaged Speed-down:
1.09x
Worst Speed-down:
2.04x (ex19)



Policy Accuracy Optimization Condition

- Stop Criterion and Ver.Beta Function
 - The 2-norm of “Real” residual vector (“Real Residual Error”) is set to **less than 1.0D-8**.
 - The computation of the “real” residual error is computed in every converged stop on the solver.
 - If the “real” residual error is not satisfied to inputted user policy, the inner convergence condition on the solver, say EPS, is reduced to the factor of 1/10.
 - Then, the solver is called again with the new EPS until satisfying the “real” residual error. (This is called “Retry Iteration”.)
 - GS methods are auto-tuned. (MGS, BCGS, and DGKS)
- Ver. Alpha Details
 - There is no function for the Accuracy Policy.
 - The execution time with same solver inner convergence condition EPS for same number of Retry Iteration to that of Ver. Beta is measured.
 - GS method is fixed as MGS.
(Only provided one is MGS for Ver. Alpha)

Accuracy Policy Effect (OpenATI_EIGENSOLVE)



The following matrices for Ver. Alpha are not converged.

●T3dl :

Residual: 2.24E-01

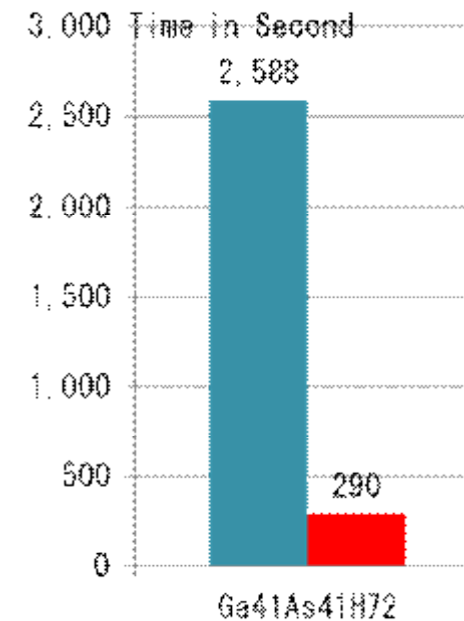
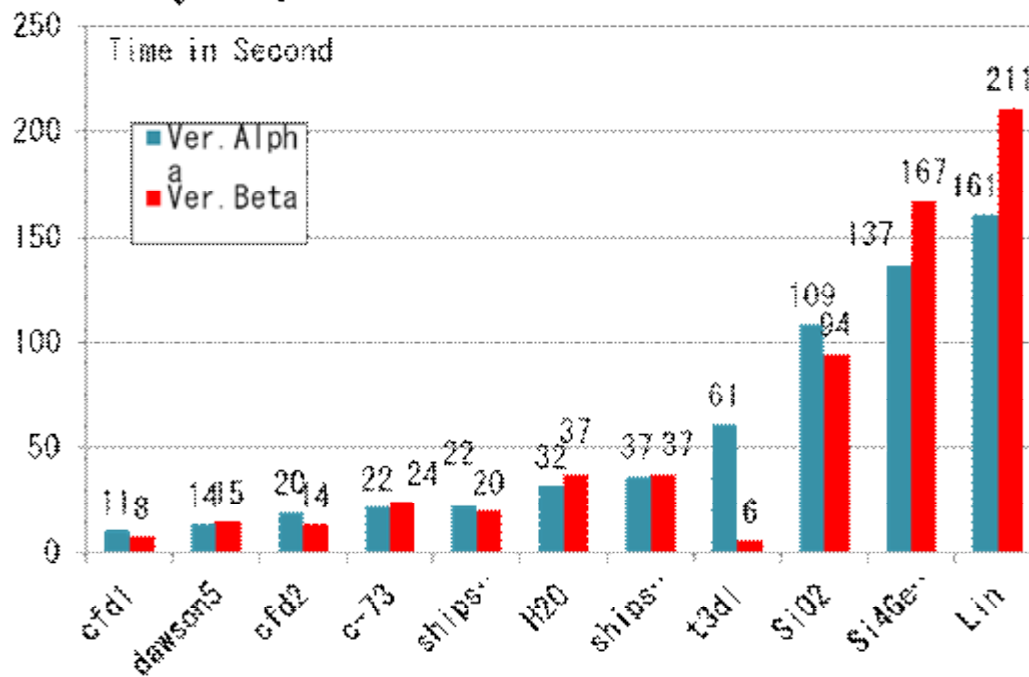
Orthogonality: 1.27E-02

●Ga41As41H72

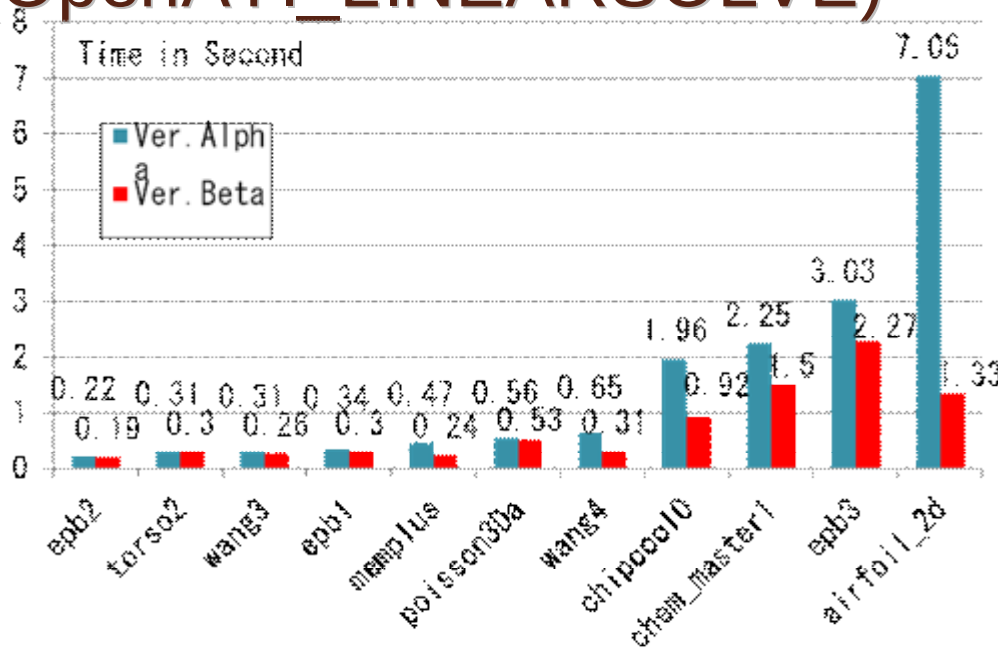
Residual : 2.52E-08

Orthogonality : 4.81E-09

Averaged Speedup: 2.24x
Best Speedup: 8.92x
(Ga41As41H72)



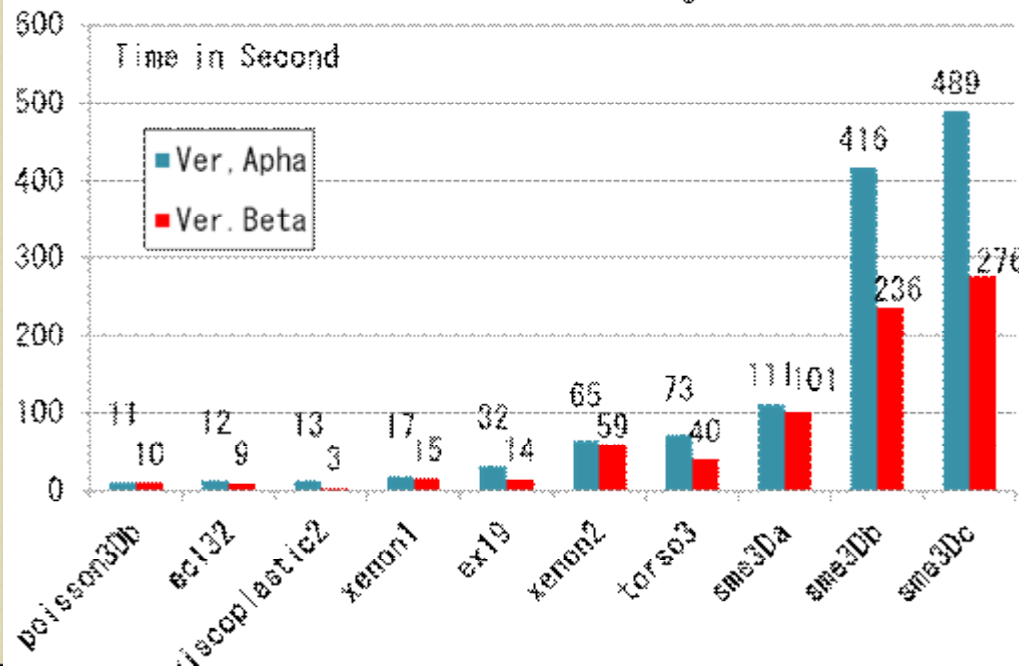
Accuracy Policy Effect (OpenATI_LINEARsolve)



torso1 is not converged for both versions.

The following matrices for Ver. Alpha are not converged.

- torso3 : 2.28E-08
- memplus : 1.92E-08
- airfoil_2d : 5.17E-05
- viscoplastic2 : 3.67E-05
- wang4 : 4.35E-08



Averaged Speedup:
1.69x

Best Speedup:
5.29x (airfoil_2d)

This is due to highly accuracy for computation with DGKS orthogonalization, provided Ver.Beta.

おわりに

- 汎用的自動チューニングAPIのOpenATLibの新機能

- ソルバレベルの解法選択機能API

- OpenATI_EIGENSOLVE および OpenATI_LINEARISOLVE の提供

- 数値計算ポリシー設定機能API

- GS直交化アルゴリズム選択機能API

- 世界に先駆けた先進的AT機能

- 自動チューニングポリシー設定機能

- **メモリ量最適化**：平均1.2倍の速度低下で
平均200倍のメモリ量削減が可能
- **演算精度最適化**：旧バージョン（α版）に対し，
 - 平均1.6倍～2.4倍の速度向上
 - **実誤差を満たさない場合（偽収束）を改善**
 - **要求精度改善工程の自動化**

- マルチコア向きSpMxVの負荷バランス改善手法

- （非対称用）Branchless Segmented Scan (BSS) 法の実装
- 極めて偏った要素をもつ疎行列、高マルチコア実行での効果を期待
- （対称・非対称用）負荷バランス改善機能（NZ法）、
リダクション削減法の実装
- **コア数の増加に応じた速度向上を達成**

- 直交化アルゴリズム動的選択機能

- 精度改善、および、収束性の向上による高速化に貢献



Xabclib利用法

- インストール方法
- プログラム開発の方法
- デモ

実行環境

東京大学情報基盤センター

T2Kオープンスパコン(東大版)

HITACHI HA8000クラスタシステム

Xabclibのインストール

- インストール手順

<http://www.abc-lib.org/Xabclib/> から
Xabclib-beta.tar.gz をダウンロードする

```
$ tar zxvf Xabclib-beta.tar.gz    ← 解凍  
$ ./make.all                      ← ライブラリ生成
```

libOpenAT.a ができる

ただし、Intel Fortran コンパイラの利用を想定している。
他のコンパイラを使用するときは、下記の箇所の修正が必要。

- コンパイラまたはコンパイラオプションを変更する場合

下記3つの makefile のF90の定義を修正する

./OpenAT/makefile 内の

F90 = ifort -O3 -m64 -openmp -mcmodel=medium

./Xabclib_GMRES/makefile 内の

F90 = ifort -O3 -m64 -parallel -openmp -par_report3 -mcmodel=medium

./Xabclib_LANCZOS/makefile 内の

F90 = ifort -O3 -m64 -openmp -mcmodel=medium

Xabclibの基本手順

- プログラム作成に必要なもの

次の2つのファイルがあれば良い

```
OpenAT.inc  
libOpenAT.a
```

- ソースコードの記述

OpenATI関数のパラメータ設定を行う場合、下記のように指定する。

```
INCLUDE "OpenAT.inc"
```

OpenAT.inc には、common文による共通領域の定義が記載されている

- コンパイル方法

次のようにlibOpenAT.a をリンクする

```
$ ifort プログラム名 libOpenAT.a
```


OpenAT.inc の内容

- `OpenATI_DAFRT_IPARM_1`
MM比による自動チューニングを行うかどうかのフラグ (1=有効)
- `OpenATI_DAFRT_RPARM_1`
MM比(残差Max-Min比)の閾値 (…… 100.0D0 ……)
- `OpenATI_DSRMV_IPARM_1`
対称行列ベクトル積の自動チューニングを検索範囲(1, 2, 3)
- `OpenATI_DURMV_IPARM_1`
非対称行列ベクトル積の自動チューニングを検索範囲(1, 2, 3)
- `OpenATI_DURMV_IPARM_2`
非対称行列ベクトル積の性能評価時の反復回数(1, ……)
- `OpenATI_DAFGS_IPARM_1`
Gram-Schmidt直交化の実装方式 (0, 1, 2, 3)
※下線はデフォルト値

詳細は Developer's Manual または OpenAT.inc 内のコメント欄に記載

Xabclib_GMRES

● サブルーチンの呼び出し方

```
CALL Xabclib_GMRES(N,NNZ,IRP,ICOL,VAL,B,X,KIND_PRECOND,  
$ PRECOND,NPRE,IPCPARM,RPCPARM,MSIZE,IGRPARM,RGRPARM,  
$ IAT,WK,LWK,INFO)
```

N	:係数行列の次元数
NNZ	:係数行列の非零要素の個数
IRP	:係数行列の各行の先頭要素へのポインタ
ICOL	:係数行列の非零要素の行番号
VAL	:係数行列の非零要素の値
B	:右辺ベクトル
X	:初期解ベクトル&解ベクトル
KIND_PRECOND	:前処理の種類
PRECOND	:前処理行列
NPRE	:配列PRECONDのサイズ
IPCPARM	:前処理行列のオプションパラメータ1
RPCPARM	:前処理行列のオプションパラメータ2
MSIZE	:リスタート周期
IGRPARM	:GMRES法のオプションパラメータ
RGRPARM	:GMRES法のオプションパラメータ
IAT	:自動チューニングのオプションパラメータ
WK	:作業領域
LWK	:作業領域WKのサイズ
INFO	:エラーコード

OpenATI_LINEARsolve
を使えばこの部分は不要

Xabclib_LANCZOS

● サブルーチンの呼び出し方

```
CALL Xabclib_LANCZOS(N,NNZ,IRP,ICOL,VAL,NEV,EV,EVEC,LDE,  
$      MSIZE,IPARM,RPARM,IAT,WK,LWK,IWK,LIWK,INFO)
```

N :係数行列の次元数
NNZ :係数行列の上三角部分の非零要素の個数
IRP :係数行列の上三角部分の各行の先頭要素へのポインタ
ICOL :係数行列の上三角部分の非零要素の行番号
VAL :係数行列の上三角部分の非零要素の値
NEV :求めたい固有値の個数
EV :固有値
EVEC :各固有値に対応する固有ベクトル
LDE :EVECの整合寸法
MSIZE :リスタート周期
IPARM :Lanczos法のオプションパラメータ
RPARM :Lanczos法のオプションパラメータ
IAT :自動チューニングのオプションパラメータ
WK :作業領域
LWK :作業領域WKのサイズ
IWK :作業領域
LIWK :作業領域IWKのサイズ
INFO :エラーコード

OpenATI_EIGENSOLVE
を使えばこの部分は不要

OpenATI_LINEARsolve

• サブルーチンの呼び出し方

```
CALL OpenATI_LINEARsolve(N,NZ,IRP,ICOL,VAL,B,X,INFO)
```

N :係数行列の次元数
NZ :係数行列の非零要素の個数
IRP :係数行列の各行の先頭要素へのポインタ
ICOL :係数行列の非零要素の行番号
VAL :係数行列の非零要素の値
B :右辺ベクトル
X :初期解ベクトル&解ベクトル
INFO :エラーコード

OpenATI_EIGENSOLVE

• サブルーチンの呼び出し方

```
CALL OpenATI_EIGENSOLVE(N, NNZ, IRP, ICOL, VAL, IORDER, NEV, EV,  
$                        EVEC, INFO)
```

N : 係数行列の次元数
NNZ : 係数行列の上三角部分の非零要素の個数
IRP : 係数行列の上三角部分の各行の先頭要素へのポインタ
ICOL : 係数行列の上三角部分の非零要素の行番号
VAL : 係数行列の上三角部分の非零要素の値
IORDER: 非零要素の値を絶対値としてみるかどうかのフラグ
NEV : 求めたい固有値の個数
EV : 固有値
EVEC : 各固有値に対応する固有ベクトル
INFO : エラーコード

自動チューニングポリシー設定

- 自動チューニングポリシー設定ファイル

```
$ export OPENATI_POLICY=input_policy.dat ← ファイル名を指定
```

input_policy.dat の中身

POLICY = TIME	← 他には ACCURACY, MEMORY
CPU = 16	← 利用スレッド数
RESIDUAL = 1.0D-10	← 許容残差
MAXMEMORY = 16	← 単位は GB
MAXTIME = 600.0	← 単位は 秒
PRECONDITIONER = NO	← あるいは JACOBI, SSOR, ILU0

6つの自動チューニングポリシーの設定が可能

OpenATI_LINEARsolve() または OpenATI_EIGENSolve が呼ばれたときにこれらのポリシーに基づいて自動チューニングが行われる。

HA8000システムでの実行

ジョブスクリプトとジョブ投入

ジョブスクリプト名：go

```
#!/bin/bash
#@$-q debug
#@$-N 1
#@$-J T16
#@$-lM 28GB
#@$-lT 0:05:00
#@$-o OUT
#@$-e ERR
cd $PBS_O_WORKDIR ← qsubを実行したディレクトリに移動
export OMP_NUM_THREADS=16
export OPENATI_POLICY=./input_policy.dat
./a.out > go.txt
```

ジョブ投入

```
$ qsub go
```

結果は go.txt に出力される