Xabclib: A Sparse Iterative Solver with a Generalized Auto-tuning Interface, and Overview of Auto-tuning Studies in Japan

Takahiro Katagiri
Information Technology Center, The University of Tokyo

Joint work with:
Prof. H. Kuroda (Ehime U.)
Prof. K. Nakajima (U. Tokyo)
Mr. T. Sakurai and Dr. K. Naono (HITACHI Ltd.)
Outline of This Talk

PART1: Quick Overview of Current Auto-tuning Studies in Japan

PART2: Developing OpenATLib
- E-Science Project granted by MEXT Japan
  - OpenATLib
  - Xabclib_LANCZOS, Xabclib_GMRES
- Evaluation with The T2K Open Supercomputer (Todai)

Conclusion Remarks
Current Auto-tuning Studies in Japan

Prof. Suda (U. Tokyo)
Prof. Yamamoto (Nagoya U.)
Prof. Imamura (U. of Electro-communication)
Dr. Shoji Itoh (RIKEN)
Auto-tuning Research Group in Japan (Since 2002)

1. Chair: Reiji Suda (U. Tokyo)
2. Vice Chairs: Takahiro Katagiri (U. Tokyo) and Shoji Itoh (RIKEN)
3. Advisory: Toshitsugu Yuba (U. of Electro-Communications)
4. Toshiyuki Imamura (U. of Electro-Communications)
5. Yusaku Yamamoto (Nagoya U.)
6. Ken Naono (HITACHI Ltd.)
7. Kentaro Shimizu (U. of Tokyo)
8. Hiroyuki Sato (U. of Tokyo)
9. Takeshi Iwashita (Kyoto U.)
10. Kazuya Terauchi (Visual Numerics Japan)
11. Masahi Egi (HITACHI Ltd.)
12. Takao Sakurai (HITACHI Ltd.)
13. Hisayasu Kuroda (Ehime U.)
14. Kengo Nakajima (U. of Tokyo)
15. Hiroyuki Takizawa (Tohoku U.)
16. Daisuke Takahashi (Tsukuba U.)
17. Masahiro Yasugi (Kyoto U.)
Mathematical Core for Automatic Tuning

Automatic Tuning Abstraction

- Design
- Measurement
- Analysis
- Decision

Automatic Tuning Math Toolset

- Performance Modeling Toolbox
- Experimental Design Toolbox
- Data Analysis Toolbox
- Optimization Toolbox

Key Ideas of Our Methods

1. Online Automatic Tuning
2. Bayesian Data Analysis
3. Sequential Experimental Design
4. Stability and Efficiency Guarantee

Balancing...

- Prior Knowledge
- Empirical Info
Optimizing the Blocking Strategy for Linear Algebra Algorithms via Dynamic Programming

Yusaku YAMAMOTO (Nagoya University)

Motivation

- Blocking is the key to achieving high performance in linear algebra routines
- As hardware evolves, more and more elaborate blocking strategy is needed
- Difficult to find the optimal blocking strategy by hand-tuning

Our approach

- Represent all possible blocking strategies using binary trees
- Estimate the exec. time of each tree based on BLAS performance models
- Find the near-optimal tree using dynamic programming

Optimal Tree for Opteron, 6000 x 6000 QR decomposition
High performance and highly scalable eigensolver project \texttt{<eigen_s & eigen_sx>}

- **Purpose**
  - High performance eigensolver for quantum physics code.
  - Highly-scalable eigensolver on a peta-scale super computer.
    - **Multicore, multiprocessor**
      - Beyond billion cores!!
    - **MPI and OpenMP hybrid**
      - available on multiple programming model
    - **Narrow memory bandwidth**

- **Algorithm**
  - **Narrow-band-reduction**
  - **Block version divide and conquer**
    - Replace Level 2 to 2.5 or 3BLAS!!

- **New features**
  - Development of AT-facilitated numerical library
    - Algorithm selection of bandsize and the number of block
  - **DD/QD high precision version**

---

**Performance scalability of our eigensolver in weak-scaling**

![Performance Graph]

- **Performance [GFLOPS]**
- **Number of nodes (matrix dimension is 9600-times the number of nodes)**
Survey and Evaluation System for Numerical Algorithms
Shoji Itoh (RIKEN)

Systematic performance evaluation and characteristic analysis of numerical algorithms

We analyze actual characteristics and so on, by using some data analyzing methods or data evaluation methods, QC, statistics, data mining or visualization technique. Here, we pay attention to various data generated by several solution algorithms.

Cells are arranged in like this. They indicate performance information.

Grouping by Solver or Preconditioning

<table>
<thead>
<tr>
<th>Solver</th>
<th>Precond</th>
<th>Matrix</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Matrices (Combination of Solver and Preconditioning)
Unrolling Depth: Developer specifies using directive

Example: Matrix-matrix multiplication code

```
!ABCLib$ install unroll (i,j,k) region start
!ABCLib$ name MyMatMul
!ABCLib$ varied (i,j,k) from 1 to 4
!ABCLib$ debug (pp)

do i=1, N
  do j=1, N
    da1 = A(i, j)
    do k=1, N
      dc = C(k, j)
      da1 = da1 + B(i, k) * dc
    enddo
    A(i, j) = da1
  enddo
enddo
!ABCLib$ install unroll (i,j,k) region end
```

Install-time optimization;
Unrolling process;

Unrolling Depth

Target Region (Auto-tuning Region)
Developing OpenATLib

Joint work with:
Prof. H. Kuroda (Ehime U.)
Prof. K. Nakajima (U. Tokyo)
Mr. T. Sakurai and Dr. K. Naono (HITACHI Ltd.)
“Seamless and Highly-Productive Parallel Programming Environment for High-Performance Computing”, High Performance Library, MEXT, JAPAN (2008 ~ 2011)

- **Problem**
  - Heavily relies on artisan techniques.
  - Non-productive and non-portability.
  - Time-consuming and not cost-effective.
  - Sometimes fails in convergence caused by out of range on parameters.

- **Goal**
  - To provide highly productive and performance portable numerical library.

- With respect to sparse matrix non-zero structures, we supply an **On-the-fly AT Facility for**:
  1. Computation Kernel Selection
  2. Numerical Algorithm Selection
  3. Parallel Implementation Selection
  4. General AT API (OpenATLib)
Architecture and Programming Trend

- **Pervasive Multicore Architectures**
  - Non-uniform Memory Access (ccNUMA)
  - Multi layer memories:
    - L1 and L2 caches are localized. L3 cache is shared.
  - Number of cores is getting increased
    - High-end: 32+, Low-end: 8+

- **Changing Parallel Programming Model**
  - Pure MPI
    - Low ~ middle parallel execution (up to 1,000 cores)
  - Hybrid MPI (OpenMP + MPI)
    - Massively Parallel Execution (10,000+ cores)

- **Hard to manage the performance on compilers**
  - High cost for hand tuning
  - To reduce cost, auto-tuning technologies are spotted out.
What is “Auto-tuning Facility”? 

- Computer Architecture
- Computer System Software
- Program
- Algorithm

Performance Parameters

Tuning Facility
- Optimization
- Parameter Search
- Machine Learning / Self Adaptive

Parameterization, auto-tuner and code generation Facility

Performance Monitor Facility
: AT Performance Database

Auto-tuning Facility
Conventional AT Library Problem

- Dedicated AT facilities are developed.
- Not considered to be re-using:
  - **No Reusability for AT Facility**
Aim of This Study

- Framework to establish AT Facility Reusability
  1. Define general functions for AT on numerical libraries.
  2. Define general APIs based on 1.
  3. Open the library which is implemented for 2.
  4. Developers use the library of 3 to develop their new AT libraries.

- Target
  - Numerical libraries.

- Coverage of AT
  - Higher parameter AT requests.
    - Conventional AT libraries only supply BLAS level.
    - Ex. ATLAS and OSKI.
    - Algorithm specific parameters on numerical libraries.
  - Policy request from end-users should be supported.
    - Ex. Speed, Memory Space, Numerical Accuracy.
New AT Function Request

Function

- Parameter Optimization (conventional)
  - Algorithm Selection
  - Implementation Selection
  - Computer Resource Selection

- User Knowledge Setting (conventional)
  - Location of AT
  - Sparse Matrix Distribution
    - Band, Block Size, Random

- User Policy Setting (Proposal)
  - AT Policy
    - Priority Speed
    - Priority Memory Space
    - Priority Numerical Accuracy
    - Both of Them

To

- Library Developer
- End User

Optimization Information

Auto Setting
An Example of AT Policy Description for end-users

[Naono et.al., 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)

Number of Cores: from 20 to 100.
Execution time: from 100 sec. to 1000 sec.

Object function: minimize of cost definition function

The cost definition function
A Release Implementation of AT

OpenATLib: AT API Library

ABC Lib_GMRES and ABC Lib_LANCZOS:
OpenMP Parallelization Approach Using OpenATLib
Current Aim of OpenATLib

- On the fly AT functions.
- To AT Library Developers.
- Limited Implementations on Performance Parameter Optimization Function.
  - Algorithm Selection Function
    - Restart frequency selection on the Krylov subspace method.
  - Implementation Selection Function
    - SpMxV (Sparse Matrix vector multiplication)
      - Optimization for (very limited now):
        - Cache aware, unrolling depth, OpenMP parallelization
  - Computer Resource Selection
    - Selection for SpMxV kernel with on the fly information of memory space.

- User Policy Setting Function
  - To be developed in this year.
Development of 2009 (Development From Scratch)

- **OpenATlib**
  - Common AT Interface Library

- **ABC Lib_LANCZOS**
  - Eigensolver with the Restart LANCZOS for Standard Eigenproblem

- **ABC Lib_G MRES**
  - Linear Equations Iterative Solver with GMRES(m)

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 Memory Restriction
5. Common AT Interface to Establish The above AT Functions

**Target:** Fortran90 and OpenMP Parallelization, CRS format
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]

\[ R_i (s, t) = \frac{\max_z \{ r_i (z); z=s-t+1,\ldots,s\} }{\min_z \{ f_i (z); z=s-t+1,\ldots,s\} } \]

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

Note: Last \( S \) samples.
```
INCLUDE "OpenAT.inc"  // Including OpenAT.inc

~ omission ~

K = 1

DO ITER=1, MAXITER  // Main loop of iterative method.
  MSIZE=1  // First restart frequency.
  I=5     // Checking frequency.

~ omission ~

IF (RSDID < TOL) RETURN  // Convergence Test
SAMP (K)=RSDID // Set residual to SAMP(K).
IF (mod(ITER, I) .eq. 0) THEN // Call DAFRT per I times.
  IRT=0
  CALL OpenATI_ DAFRT (I, SAMP, IRT, INFO)
  IF (IRT .eq. 1) MSIZE=MSIZE+1  // Increase restart frequency.
  K=1
END IF

~ omission ~

ENDDO
```
OpenATlib: A General AT Interface for Library Developers

- **SpMxV Interface** $(y = Ax)$
  - `OpenATI_DSRMV()` : Symmetric
  - `OpenATI_DURMV()` : Unsymmetric
  - **Data Format:** CRS (Compressed Row Storage)
  - **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 Implementation Switch Number
    - $NUM_SMP$ : Reduction Number
    - $WK(N, NUM_SMP)$ : Work Space
    - $INFO$ : Error Code
INCLUDE “OpenAT.inc”  // Including OpenAT.inc
OpenATI_DSRMV_IPARM_1=3  // Initialize DSRMV AT function
I CASE=0  // DSRMV argument

DO I TER=1, MAXI TER  // Main loop of iterative method.
  ~ omission ~
  IF (ITER .eq. 2) THEN
    OpenATI_DSRMV_IPARM_1=1
    // After this, use the best implementation.
  ENDIF
  // Call sparse matrix vector routine
  // If ITER = 1, then check all candidates of kernel implementations.
  CALL OpenATI_DSRMV
      (N, NNZ, IRP, I COL, VAL, VEC, J PARM, I PARM, R PARM, I NFO)
  ~ omission ~
  IF (answer is converged.) RETURN
ENDDO
Evaluation Environment

- T2K Open Supercomputer (Todai)
- One Node Execution with OpenMP Parallelization (Maximum 16 cores on one node)
- Compiler
  - Intel Fortran Compiler Professional Version 11.0
- Compiler Option:
  - -O3 -m64 -openmp -mcmodel=medium
- AT Facility
  - Restart Frequency
  - SpMxV implementation selection
    - a) Normal; b) 8-2 unroll; c) Vectorized;
    - Implementation Selection According to Memory Usage
- Algorithm Specialized Matters
  - LANCZOS: 10 eigenvalues computations
  - GMRES: The Jacobi Preconditioner
    - Other Choices: Scaling, ILU, SSOR
T2k Open Supercomputer (Todai)
Physical Node Construction
(Type A Cluster)

CPU Construction

<table>
<thead>
<tr>
<th></th>
<th>L3</th>
<th>L2</th>
<th>L2</th>
<th>L2</th>
<th>L2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Core #0</td>
<td>L1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Core #1</td>
<td>L1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Core #2</td>
<td>L1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Core #3</td>
<td>L1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Memory #0
Memory #1
Memory #2
Memory #3

AMD Quad Core Opteron #0
AMD Quad Core Opteron #1
AMD Quad Core Opteron #2
AMD Quad Core Opteron #3

South Bridge

Myrinet
Myrinet
Myrinet
Myrinet

GbE
RAID
<table>
<thead>
<tr>
<th>Computer Environments</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>CPU</strong></td>
</tr>
<tr>
<td><strong>L2 Cache Size</strong></td>
</tr>
<tr>
<td><strong>Main Memory</strong></td>
</tr>
<tr>
<td><strong>OS</strong></td>
</tr>
<tr>
<td><strong>Compiler</strong></td>
</tr>
<tr>
<td><strong>Compiler Option</strong></td>
</tr>
</tbody>
</table>
### Computation Condition

**Xabclib_LANCZOS**

<table>
<thead>
<tr>
<th>Convergence Test Value</th>
<th>1.0E-08</th>
</tr>
</thead>
<tbody>
<tr>
<td>Calculated Eigenvalues</td>
<td>10</td>
</tr>
</tbody>
</table>

**Xabclib_GMRES**

<table>
<thead>
<tr>
<th>Convergence Test Value</th>
<th>1.0E-08</th>
</tr>
</thead>
<tbody>
<tr>
<td>RHS vector x</td>
<td>All elements are set to 1.</td>
</tr>
<tr>
<td>Initial guess</td>
<td>All elements are set to 0.</td>
</tr>
<tr>
<td>Preconditioner</td>
<td>ILU(0)</td>
</tr>
</tbody>
</table>

**Default Restart Frequency**

**30 : Same as PETSc Default value**
## Test matrices for Xabclib_LANCZOS

- From University Florida Sparse Matrix Collection (20 kinds)

<table>
<thead>
<tr>
<th>Matrix</th>
<th>N</th>
<th>NNZ</th>
<th>Field</th>
</tr>
</thead>
<tbody>
<tr>
<td>vibrobox</td>
<td>12328</td>
<td>177578</td>
<td>acoustics</td>
</tr>
<tr>
<td>Lin</td>
<td>25600</td>
<td>101120</td>
<td>chemistry</td>
</tr>
<tr>
<td>cfd1</td>
<td>70656</td>
<td>949510</td>
<td>Fluid dynamics</td>
</tr>
<tr>
<td>cfd2</td>
<td>12344</td>
<td>160566</td>
<td></td>
</tr>
<tr>
<td>gyro</td>
<td>17361</td>
<td>519260</td>
<td>Model reduction</td>
</tr>
<tr>
<td>t3dl</td>
<td>20360</td>
<td>265113</td>
<td></td>
</tr>
<tr>
<td>c-71</td>
<td>76638</td>
<td>468096</td>
<td></td>
</tr>
<tr>
<td>Si5H12</td>
<td>19896</td>
<td>379247</td>
<td>structural</td>
</tr>
<tr>
<td>SiO</td>
<td>33401</td>
<td>675528</td>
<td></td>
</tr>
<tr>
<td>dawson5</td>
<td>51537</td>
<td>531157</td>
<td></td>
</tr>
<tr>
<td>H2O</td>
<td>67024</td>
<td>1141880</td>
<td></td>
</tr>
<tr>
<td>F2</td>
<td>71505</td>
<td>2682895</td>
<td></td>
</tr>
<tr>
<td>oilpan</td>
<td>73752</td>
<td>1835470</td>
<td></td>
</tr>
<tr>
<td>shipsec1</td>
<td>140874</td>
<td>3977139</td>
<td></td>
</tr>
<tr>
<td>bmw7st_1</td>
<td>141347</td>
<td>3740507</td>
<td></td>
</tr>
<tr>
<td>SiO2</td>
<td>155331</td>
<td>5719417</td>
<td></td>
</tr>
<tr>
<td>shipsec5</td>
<td>179860</td>
<td>5146478</td>
<td></td>
</tr>
<tr>
<td>Si41Ge41H72</td>
<td>185639</td>
<td>7598452</td>
<td></td>
</tr>
<tr>
<td>bmw3_2</td>
<td>227362</td>
<td>5757996</td>
<td></td>
</tr>
<tr>
<td>Ga41As41H72</td>
<td>268096</td>
<td>9378286</td>
<td></td>
</tr>
</tbody>
</table>
## Test matrices for Xabclib_GMRES

- From University Florida Sparse Matrix Collection (20 kinds)

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

<table>
<thead>
<tr>
<th>Matrix</th>
<th>N</th>
<th>NNZ</th>
<th>Field</th>
</tr>
</thead>
<tbody>
<tr>
<td>xenon1</td>
<td>48600</td>
<td>1181120</td>
<td>materials</td>
</tr>
<tr>
<td>xenon2</td>
<td>15746</td>
<td>3866688</td>
<td></td>
</tr>
<tr>
<td>wang4</td>
<td>26068</td>
<td>177196</td>
<td>semiconductor device</td>
</tr>
<tr>
<td>ecl32</td>
<td>51993</td>
<td>380415</td>
<td></td>
</tr>
<tr>
<td>sme3Da</td>
<td>12504</td>
<td>874887</td>
<td>structural</td>
</tr>
<tr>
<td>sme3Db</td>
<td>29067</td>
<td>2081063</td>
<td></td>
</tr>
<tr>
<td>sme3Dc</td>
<td>42930</td>
<td>3148656</td>
<td></td>
</tr>
<tr>
<td>epb1</td>
<td>14734</td>
<td>95053</td>
<td>thermal</td>
</tr>
<tr>
<td>epb2</td>
<td>25228</td>
<td>175027</td>
<td></td>
</tr>
<tr>
<td>epb3</td>
<td>84617</td>
<td>463625</td>
<td></td>
</tr>
</tbody>
</table>
## Restart Frequency AT Effect: Xabclib_LANCZOS

<table>
<thead>
<tr>
<th>Matrices</th>
<th>Fixed Restart Frequency</th>
<th>Auto-tuning</th>
<th>Speedup with AT</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>M</td>
<td>restart</td>
<td>time(sec)</td>
</tr>
<tr>
<td>vibrobox</td>
<td>30</td>
<td>24</td>
<td>1.13</td>
</tr>
<tr>
<td>Lin</td>
<td>30</td>
<td>1047</td>
<td>601.65</td>
</tr>
<tr>
<td>cfd1</td>
<td>30</td>
<td>55</td>
<td>12.42</td>
</tr>
<tr>
<td>cfd2</td>
<td>30</td>
<td>45</td>
<td>18.69</td>
</tr>
<tr>
<td>gyro</td>
<td>30</td>
<td>10</td>
<td>1.13</td>
</tr>
<tr>
<td>t3dl</td>
<td>30</td>
<td>1878</td>
<td>125.62</td>
</tr>
<tr>
<td>c-71</td>
<td>30</td>
<td>4</td>
<td>0.70</td>
</tr>
<tr>
<td>Si5H12</td>
<td>30</td>
<td>192</td>
<td>17.99</td>
</tr>
<tr>
<td>SiO</td>
<td>30</td>
<td>161</td>
<td>24.32</td>
</tr>
<tr>
<td>dawson5</td>
<td>30</td>
<td>1052</td>
<td>135.75</td>
</tr>
<tr>
<td>H2O</td>
<td>30</td>
<td>623</td>
<td>168.46</td>
</tr>
<tr>
<td>F2</td>
<td>30</td>
<td>27</td>
<td>15.04</td>
</tr>
<tr>
<td>oilpan</td>
<td>30</td>
<td>28</td>
<td>10.63</td>
</tr>
<tr>
<td>shipsec1</td>
<td>30</td>
<td>26</td>
<td>20.89</td>
</tr>
<tr>
<td>bmw7st_1</td>
<td>30</td>
<td>1</td>
<td>1.06</td>
</tr>
<tr>
<td>SiO2</td>
<td>30</td>
<td>699</td>
<td>805.68</td>
</tr>
<tr>
<td>shipsec5</td>
<td>30</td>
<td>53</td>
<td>59.31</td>
</tr>
<tr>
<td>Si41Ge41H72</td>
<td>30</td>
<td>411</td>
<td>679.91</td>
</tr>
<tr>
<td>bmw3_2</td>
<td>30</td>
<td>3</td>
<td>4.20</td>
</tr>
<tr>
<td>Ga41As41H72</td>
<td>30</td>
<td>10000</td>
<td>-</td>
</tr>
</tbody>
</table>
### All AT Effect: Xabclib_LANCZOS

<table>
<thead>
<tr>
<th>Matrices</th>
<th>Non AT</th>
<th>AT</th>
<th>Speedup with AT</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>M</td>
<td>restart</td>
<td>time (sec)</td>
</tr>
<tr>
<td>vibrobox</td>
<td>30</td>
<td>24</td>
<td>1.13</td>
</tr>
<tr>
<td>Lin</td>
<td>30</td>
<td>1047</td>
<td>601.65</td>
</tr>
<tr>
<td>cfd1</td>
<td>30</td>
<td>55</td>
<td>12.42</td>
</tr>
<tr>
<td>cfd2</td>
<td>30</td>
<td>45</td>
<td>18.69</td>
</tr>
<tr>
<td>gyro</td>
<td>30</td>
<td>10</td>
<td>1.13</td>
</tr>
<tr>
<td>t3dl</td>
<td>30</td>
<td>1878</td>
<td>125.62</td>
</tr>
<tr>
<td>c-71</td>
<td>30</td>
<td>4</td>
<td>0.70</td>
</tr>
<tr>
<td>Si5H12</td>
<td>30</td>
<td>192</td>
<td>17.99</td>
</tr>
<tr>
<td>SiO</td>
<td>30</td>
<td>161</td>
<td>24.32</td>
</tr>
<tr>
<td>dawson5</td>
<td>30</td>
<td>1052</td>
<td>135.75</td>
</tr>
<tr>
<td>H2O</td>
<td>30</td>
<td>623</td>
<td>168.46</td>
</tr>
<tr>
<td>F2</td>
<td>30</td>
<td>27</td>
<td>15.04</td>
</tr>
<tr>
<td>oilpan</td>
<td>30</td>
<td>28</td>
<td>10.63</td>
</tr>
<tr>
<td>shipsec1</td>
<td>30</td>
<td>26</td>
<td>20.89</td>
</tr>
<tr>
<td>bmw7st_1</td>
<td>30</td>
<td>1</td>
<td>1.06</td>
</tr>
<tr>
<td>SiO2</td>
<td>30</td>
<td>699</td>
<td>805.68</td>
</tr>
<tr>
<td>shipsec5</td>
<td>30</td>
<td>53</td>
<td>59.31</td>
</tr>
<tr>
<td>Si41Ge41H72</td>
<td>30</td>
<td>411</td>
<td>679.91</td>
</tr>
<tr>
<td>bmw3_2</td>
<td>30</td>
<td>3</td>
<td>4.20</td>
</tr>
<tr>
<td>Ga41As41H72</td>
<td>30</td>
<td>10000</td>
<td>-</td>
</tr>
<tr>
<td>Matrices</td>
<td>Fixed Restart Frequency</td>
<td>Auto-tuning</td>
<td>Speedup with AT</td>
</tr>
<tr>
<td>-------------------</td>
<td>-------------------------</td>
<td>---------------------------</td>
<td>-----------------</td>
</tr>
<tr>
<td></td>
<td>M</td>
<td>restart</td>
<td>time (sec)</td>
</tr>
<tr>
<td>chem_master1</td>
<td>30</td>
<td>22</td>
<td>2.55</td>
</tr>
<tr>
<td>torso2</td>
<td>30</td>
<td>1</td>
<td>0.68</td>
</tr>
<tr>
<td>torso1</td>
<td>30</td>
<td>1</td>
<td>2.54</td>
</tr>
<tr>
<td>torso3</td>
<td>30</td>
<td>12</td>
<td>33.57</td>
</tr>
<tr>
<td>memplus</td>
<td>30</td>
<td>5</td>
<td>0.25</td>
</tr>
<tr>
<td>ex19</td>
<td>30</td>
<td>1000</td>
<td>-</td>
</tr>
<tr>
<td>poisson3Da</td>
<td>30</td>
<td>3</td>
<td>0.48</td>
</tr>
<tr>
<td>airfoil_2d</td>
<td>30</td>
<td>7</td>
<td>0.73</td>
</tr>
<tr>
<td>poisson3Db</td>
<td>30</td>
<td>7</td>
<td>10.38</td>
</tr>
<tr>
<td>viscousplastic2</td>
<td>30</td>
<td>19</td>
<td>2.93</td>
</tr>
<tr>
<td>xenon1</td>
<td>30</td>
<td>30</td>
<td>20.16</td>
</tr>
<tr>
<td>xenon2</td>
<td>30</td>
<td>40</td>
<td>92.96</td>
</tr>
<tr>
<td>wang4</td>
<td>30</td>
<td>5</td>
<td>0.37</td>
</tr>
<tr>
<td>eci32</td>
<td>30</td>
<td>1000</td>
<td>-</td>
</tr>
<tr>
<td>sme3Da</td>
<td>30</td>
<td>670</td>
<td>215.49</td>
</tr>
<tr>
<td>sme3Db</td>
<td>30</td>
<td>1000</td>
<td>-</td>
</tr>
<tr>
<td>sme3Dc</td>
<td>30</td>
<td>1000</td>
<td>-</td>
</tr>
<tr>
<td>epb1</td>
<td>30</td>
<td>11</td>
<td>0.38</td>
</tr>
<tr>
<td>epb2</td>
<td>30</td>
<td>3</td>
<td>0.22</td>
</tr>
<tr>
<td>epb3</td>
<td>30</td>
<td>11</td>
<td>3.22</td>
</tr>
</tbody>
</table>
SpMxV AT Effect: Xabclib_GMRES
<table>
<thead>
<tr>
<th>Matrices</th>
<th>Non AT</th>
<th>AT</th>
<th>Speedup with AT</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>M</td>
<td>restart</td>
<td>time (sec)</td>
</tr>
<tr>
<td>chem_master1</td>
<td>30</td>
<td>22</td>
<td>2.55</td>
</tr>
<tr>
<td>torso2</td>
<td>30</td>
<td>1</td>
<td>0.68</td>
</tr>
<tr>
<td>torso1</td>
<td>30</td>
<td>1</td>
<td>2.54</td>
</tr>
<tr>
<td>torso3</td>
<td>30</td>
<td>12</td>
<td>33.57</td>
</tr>
<tr>
<td>memplus</td>
<td>30</td>
<td>5</td>
<td>0.25</td>
</tr>
<tr>
<td>ex19</td>
<td>30</td>
<td>1000</td>
<td>-</td>
</tr>
<tr>
<td>poisson3Da</td>
<td>30</td>
<td>3</td>
<td>0.48</td>
</tr>
<tr>
<td>airfoil_2d</td>
<td>30</td>
<td>7</td>
<td>0.73</td>
</tr>
<tr>
<td>poisson3Db</td>
<td>30</td>
<td>7</td>
<td>10.38</td>
</tr>
<tr>
<td>viscoplastic2</td>
<td>30</td>
<td>19</td>
<td>2.93</td>
</tr>
<tr>
<td>xenon1</td>
<td>30</td>
<td>30</td>
<td>20.16</td>
</tr>
<tr>
<td>xenon2</td>
<td>30</td>
<td>40</td>
<td>92.96</td>
</tr>
<tr>
<td>wang4</td>
<td>30</td>
<td>5</td>
<td>0.37</td>
</tr>
<tr>
<td>ecl32</td>
<td>30</td>
<td>1000</td>
<td>-</td>
</tr>
<tr>
<td>sme3Da</td>
<td>30</td>
<td>670</td>
<td>215.49</td>
</tr>
<tr>
<td>sme3Db</td>
<td>30</td>
<td>1000</td>
<td>-</td>
</tr>
<tr>
<td>sme3Dc</td>
<td>30</td>
<td>1000</td>
<td>-</td>
</tr>
<tr>
<td>epb1</td>
<td>30</td>
<td>11</td>
<td>0.38</td>
</tr>
<tr>
<td>epb2</td>
<td>30</td>
<td>3</td>
<td>0.22</td>
</tr>
<tr>
<td>epb3</td>
<td>30</td>
<td>11</td>
<td>3.22</td>
</tr>
</tbody>
</table>
Run-time Auto-tuning for Numerical Library is Crucial in Petascale Computing Era

- It is impossible for hand-tuning on 10,000+ cores.
- Algorithm level run-time AT with respect to input sparse matrix will be a candidate to establish dramatically performance improvement.
- AT effect:
  - From 1.1x to 22x: depends on matrix
  - Restart Frequency AT is crucial: it converges compared to default execution.

User policy Setting function is next technical challenges in numerical library with AT.

OpenATLib, xabcLib { LANCZOS | GMRES }

- Alpha version (without AT policy function)
- It will be distributed in SC2009 as free software via PC Cluster Consortium.
Future Implementation Direction

- **SpMxV AT Function**
  - To obtain better load balancing:
    - Ex. Segmented Scan: *Needs memory space to normal CRS.*
  - Various Kernel Selection Candidate
    - Unrolling Depth Search

- **Algorithm Selection**
  - Orthogonalization Algorithms in Krylov Subspace method

- **Parallel Implementation Selection (MPI)**
  - Use collective functions
  - Use 1-to-1 communication functions

- **More Rich AT Interfaces on OpenATLib**
Run-time Auto-tuning Facility of ILIB_GMRES

Run-time Auto-tuning Contents:

1. Sparse Data Format (corresponding to register blocking)
2. Loop Unrolled Codes for SpMxV
3. Communication Implementations for SpMxV
   - (1) MPI_Allgather
   - (2) MPI_Bcast
   - (3) MPI_Send / Recv (1-to-1 blocking)
   - (4) MPI_Isend / Irecv (1-to-1 non-blocking)
   - (5) MPI_Irecv / Isend (1-to-1 non-blocking)

4. Re-start Frequency

5. Re-Orthogonalization Algorithm (MGS or CGS)
6. Preconditioners (Scaling, Block ILU, Matrix Polynomial)

Default: No Loop Unrolling, Communication: 1-to-1 Blocking, Scaling, CGS Ort., Restart Frequency: 30 fixed.