Automatic Tuning AMG Library for Fluid Analysis Applications

Akihiro Fujii†, Osamu Nakamura‡

†Kogakuin Univ. ‡Sumitomo metal Industries
Automatic tuning of Linear Solver

Fluid analysis code

for(t=0; t<time_step; t++){
    ....
    ....
    call solve(Ax=b)
    ....
    ....
}

Matrix A, vector x,b

Is data well allocated to CPUs?

Linear Solver

Is the parameter setting suited to the problem?

vector x
This talk considers ...

• Matrix distribution subroutines
  – Distributed matrices can be re-distributed using ParMETIS

• Online Automatic tuning of AMG parameters
  – In the flow analysis, similar linear problems are solved repeatedly.
  – Effectiveness of parameter settings is evaluated, and it can choose better parameters, as the time step proceeds.
    • Parameter range for tuning is set by users.
Outline

• Background
• Aim of this talk
• Matrix distribution
• Automatic tuning of AMG parameters
• Numerical tests & Evaluation
• Summary
Matrix distribution

User Code: solve $Ax = b$

- call store(CRS matrix A)
- call distribute()
- call solve(x,b)

- Matrix is stored in the library
- ParMETIS calculates new ordering
- Matrix is distributed to each process
- Vectors are also distributed and AMG solver is called
- Distributed solution vector is reordered to initial ordering.
Mat. Distribution in Flow analysis

User Code:

```c
for(t=0; t<time_step; t++){
    .....  
    if(t==0)then  
        call store(A)  
        call distribute()  
    endif  
    call solve(x,b)  
    .....  
}
```

- Matrix A does not change in this analysis.
  - AMG Setup phase is done only once.
- `distribute()`
  - Mat. A is equally decomposed to all processes
- `solve(x,b)` contains vector distribution and collection process
Outline

• Background
• Aim of this talk
• Matrix distribution
• **Automatic tuning of AMG parameters**
• Numerical tests & Evaluation
• Summary
Parameters of our AMG library

- **Setup Part**
  - Theta: criterion value of Strong connection of SA-AMG method

- **Iterative Part**
  - Preconditioned method: CG, BICGSTAB, ...
  - Types of smoother
    - Gauss Seidel method, Symmetric Gauss Seidel method
    - Multi-color GS, Symmetric Multi-color GS
  - Acceleration coefficient of smoothers
  - Iteration number of smoother
  - Multigrid cycles
    - V, W, F cycle

After setup part parameter is determined, then iterative part’s parameters are optimized.
- Setup part parameter’s adequate value mainly depends on the problem matrix.
Convergence time with various iterative parameters

Cycles (3 types) x Acceleration coefficients (5 values) = 15 patterns

AMG preconditioned method and smoother types (empirically) can mostly determine the order of performance.
Flow chart of AT

- Setup part parameter theta
- Preconditioned method and smoothers
- Smoother’s Acceleration coefficient
- Multigrid cycles

Parameter Evaluation among 10 fastest settings

Solving constant time steps with optimized parameter

Initial residual calculation + Time measurement

Next box is determined by difference between the fastest parameter and previous optimized parameter

No overhead of AT
AT process in Flow analysis

Periodical evaluation of the parameter setting

Trial execution for parameter evaluation

Normal execution with Optimized parameter setting

t=0  t=20  t=120  t=140  t=240

time step
Numerical Tests

• In the tests, flow analysis code is multi-threaded single process code.
  – In order to use matrix distribution, flow analysis code is modified to run in the MPI environment.
    • Only one process is calculating the analysis code.

• Machine environment
  – CPU: X5570(4cores)x2
  – Memory: 6GBx2
Numerical Tests

• AMG-BICGSTAB solver is used.
• Default AMG’s parameter setting is
  – 0.05: SA-AMG’s strong connection criterion value.
  – V cycle
  – Smoother
    • # of threads >1 : 1 iteration of Symmetric multi-color Gauss-Seidel smoother at each level
    • # of threads ==1 : 1 iteration of Symmetric Gauss-Seidel
  – Acceleration coefficient of smoother is
    • 0.8:cavL, cavL_R
    • 0.2:Karman
Problems for Numerical test

- `cav` means cavity flow
- `cavL_R` is cavity flow with high Reynolds number

<table>
<thead>
<tr>
<th>Name</th>
<th>cavL</th>
<th>cavL_R</th>
<th>Karman</th>
</tr>
</thead>
<tbody>
<tr>
<td>Size of problem mat.</td>
<td>200000</td>
<td>200000</td>
<td>200000</td>
</tr>
<tr>
<td>scheme</td>
<td>central diff.</td>
<td>Quick</td>
<td>Quick</td>
</tr>
<tr>
<td>Reynolds num.</td>
<td>1000</td>
<td>20000</td>
<td>1000</td>
</tr>
<tr>
<td>Time steps</td>
<td>1000</td>
<td>1000</td>
<td>1000</td>
</tr>
</tbody>
</table>
cavL\_R and Karman vortex
Effectiveness of Mat.dist.: cavL

# of threads _ # of processes

<table>
<thead>
<tr>
<th></th>
<th>8_1</th>
<th>4_2</th>
<th>2_4</th>
<th>1_8</th>
</tr>
</thead>
<tbody>
<tr>
<td># of iter</td>
<td>4.9</td>
<td>4.99</td>
<td>4.94</td>
<td>2.59</td>
</tr>
<tr>
<td>Vec dist</td>
<td>6.68</td>
<td>7.6</td>
<td>8.7</td>
<td></td>
</tr>
</tbody>
</table>

Total Solver Time [sec]

Time Steps
Effectiveness of AT+Mat.dist: cavL

- \# of Thread _ \# of process

- \theta = 0.05, V, sgs 1 iter, 0.8

- \theta = 0.04, V, gs 2 iter, 0.8
Effectiveness of Mat.dist.: cavL_R

# of threads _ # of processes

<table>
<thead>
<tr>
<th></th>
<th>8_1</th>
<th>4_2</th>
<th>2_4</th>
<th>1_8</th>
</tr>
</thead>
<tbody>
<tr>
<td># of iter</td>
<td>8.04</td>
<td>8.81</td>
<td>8.36</td>
<td>4.68</td>
</tr>
<tr>
<td>Vec dist</td>
<td>6.28</td>
<td>7.29</td>
<td>8.25</td>
<td></td>
</tr>
</tbody>
</table>
Effectiveness of AT+Mat.dist: cavL_R

# of threads _ # of processes

theta=0.05, V, sgs 1 iter, 0.8

theta=0.07, V, gs 2 iter, 0.8
Effectiveness of Mat.dist.: Karman

# of threads _ # of processes

<table>
<thead>
<tr>
<th></th>
<th>8_1</th>
<th>4_2</th>
<th>2_4</th>
<th>1_8</th>
</tr>
</thead>
<tbody>
<tr>
<td># of iter</td>
<td>15.83</td>
<td>14.94</td>
<td>17.57</td>
<td>11.05</td>
</tr>
<tr>
<td>Vec dist</td>
<td>6.82</td>
<td>7.25</td>
<td>8.59</td>
<td></td>
</tr>
</tbody>
</table>
Effectiveness of AT+Mat.dist: Karman

# of threads _ # of processes

theta=0.05, V, sgs 1 iter, 0.2

theta=0.07, V, gs 2 iter, 0.2
Summary and conclusion

• This study introduces and evaluates
  – Matrix distribution
  – Automatic parameter tuning
• Matrix relocation and Automatic tuning reduces the solver time up to 33% and 16% respectively.
Acknowledgements

• A part of this research was supported by Grants-in-Aid for Scientific Research, MEXT.(No. 23240005)