Automatic Performance Tuning for the Multi-section with Multiple Eigenvalues Method for the Symmetric Eigenproblem

Takahiro Katagiri[1,2], Christof Voemel[2],
James Demmel[2]

[1]The University of Electro-communications, Japan
[2]The University of California at Berkeley, USA

PARA'06, Umea, Sweden
CP4, Monday, June19, 2006, 14:00-14:20

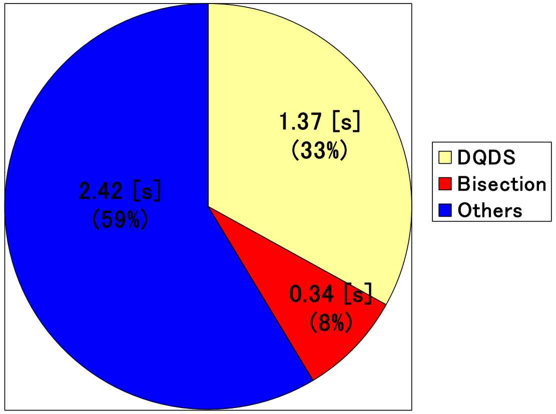
Outline

- Background
  - The bisection routine is bottle-neck in the current implementation of MRRR in LAPACK.
  - Multi-section with Multiple Eigenvalues (MME) Method
- Propose An Run-time Auto-tuning Function for MME
- Performance Evaluation Using the HITACHI SR8000
- Conclusion

Background

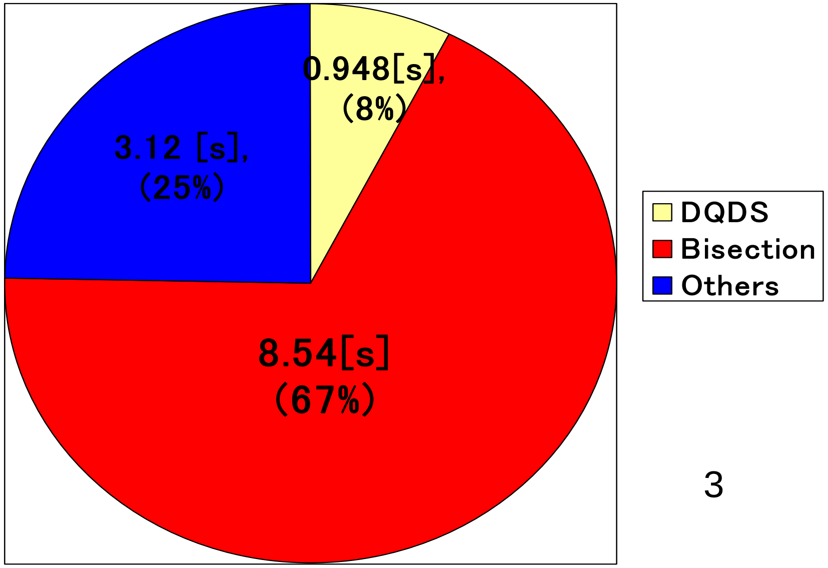
- In the current implementation of LAPACK4.0 MRRR routine, the most heaviest part is bisection routine, if the eigenvalues are tightly clustered.

T=(-1,2,-1) Matrix



Glued Wilkinson +21 Matrix

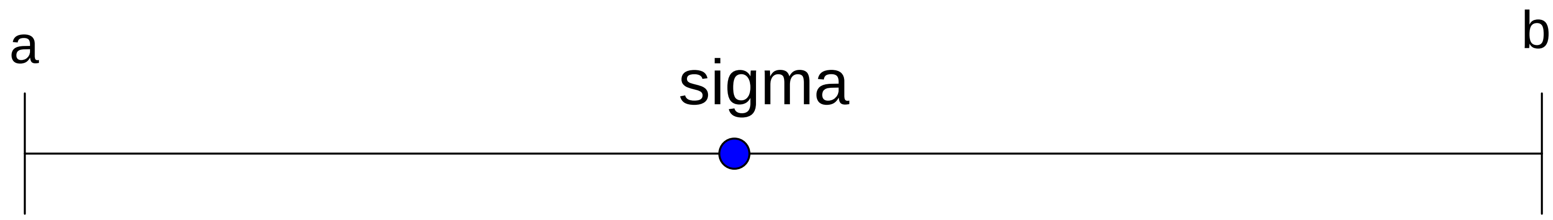
Eigenvalues are very clustered.



HITACHI SR8000
1node/8PE DATA

Bisection Method

- Target: Tridiagonal Symmetric Matrix
- The interval for all eigenvalues is given.
- The sigma is the search point to count the number of eigenvalues to narrow the interval.



sigma = a+(b-a)/2

- Recently algorithm is used.
  – The count is correct except for floating point calculation error. [Demmel et.al, 1994]

The Bisection Kernel

```
S=0; NEG=0;

do J=1, N-1

   T = S - SIGMA

   DPLUS = D(J) + T

   S = T * LLD(J) / DPLUS

   if (DPLUS<0) NEG=NEG+1

enddo
```

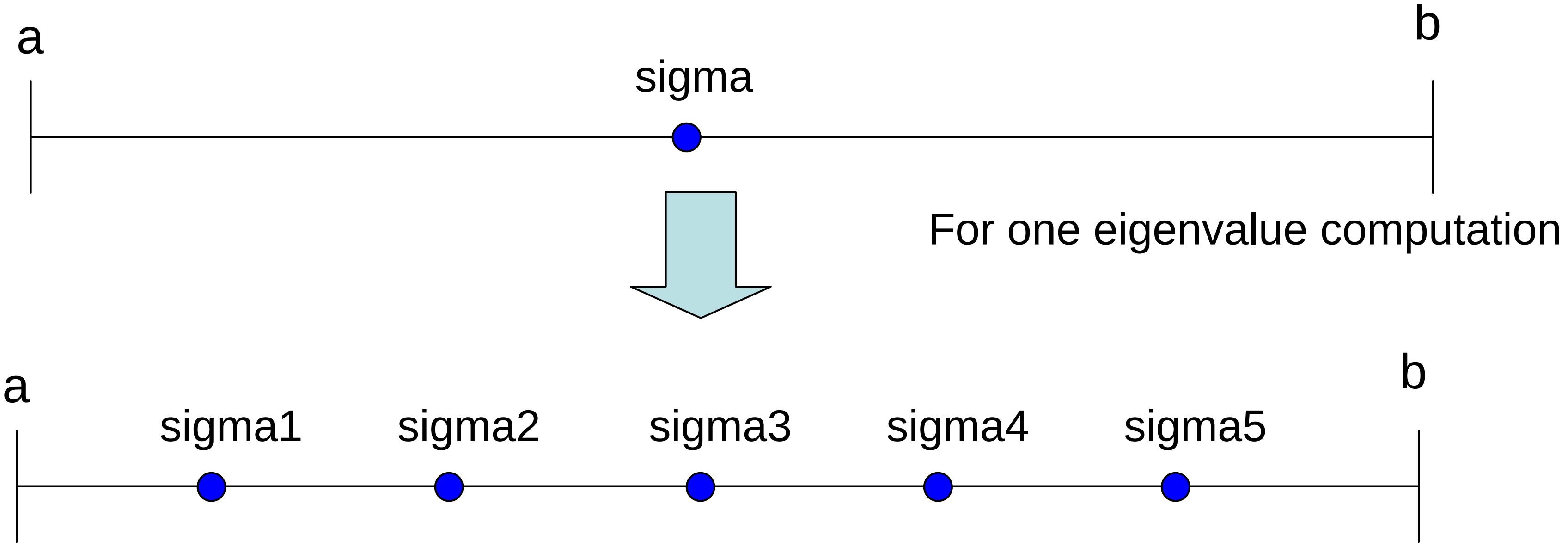
:Loop Carried Flow Dependency

This cannot be vectorized and parallelized.

Multi-section

- Multi-section [Lo et.al.,1987][Simon,1989]

Bisection:



- Merit: The kernel can be parallelized and vectorized.
- Drawback: The parallelism strongly depends on the distribution of eigenvalue. (Guess that the bisection find the eigenvalue in early iteration time.)

The multi-section Kernel

```
S(1:ML)=0; NEG(1:ML)=0;

do I=1, ML
  do J=1, N-1
    T(I) = S(I) - SIGMA(I)
    DPLUS(I) = D(J) + T(I)
    S(I)=T(I)*LLD(J) / DPLUS(I)
    if (DPLUS(I)<0)  NEG(I) = NEG(I) + 1
  enddo
enddo
```

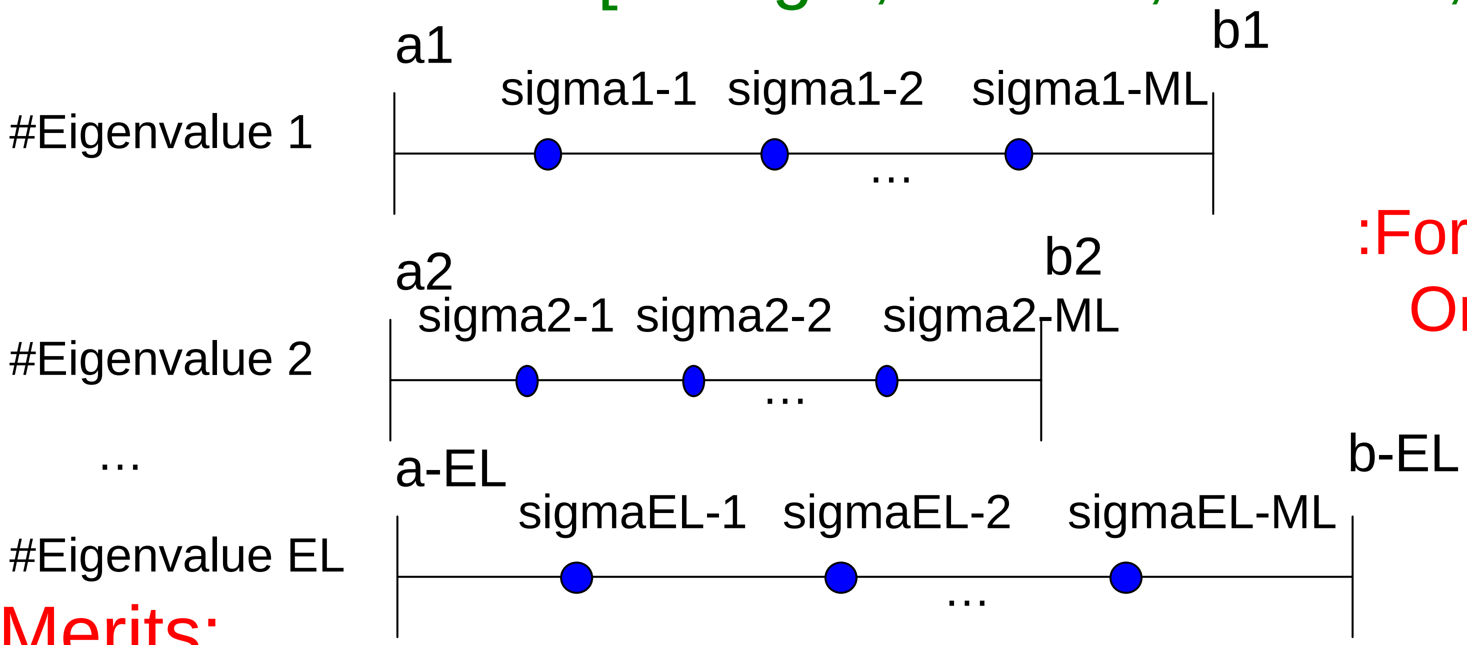
ML: The number of multi-section points.

:There is no dependency for T(I), DLPAS(I), and S(I).

The loop I can be vectorized & parallelized.

Multi-section with Multiple Eigenvalues (MME) Method

- MME Method [Katagiri,Voemel,Demmel,2006]



:For The One Kernel

- Merits:
  - The kernel can be parallelized and vectorized.
  - The outer loop length can keep long, even if we take small multi-section points (ML) --- EL-times to normal multi-section.

⇒ The search efficiency & parallelism keep high compared to multi-section.

- Drawback:

  There is no merit for no multiple eigenvalue case.

The MME Kernel

```
S(1:EL*ML)=0; NEG(1:EL*ML)=0;
do I=1, EL*ML
  do J=1, N-1
    T(I) = S(I) - SIGMA(I)
    DPLUS(I) = D(J) + T(I)
    S(I) = T(I)*LLD(J) / DPLUS(I)
    if (DPLUS(I)<0) NEG(I) = NEG(I) + 1
  enddo
enddo
```

EL: The number of Eigenvalues.
ML: The number of multi-section points.

(1)The loop I can be vectorized & parallelized.
(2) The loop length of I is longer than multi-section.

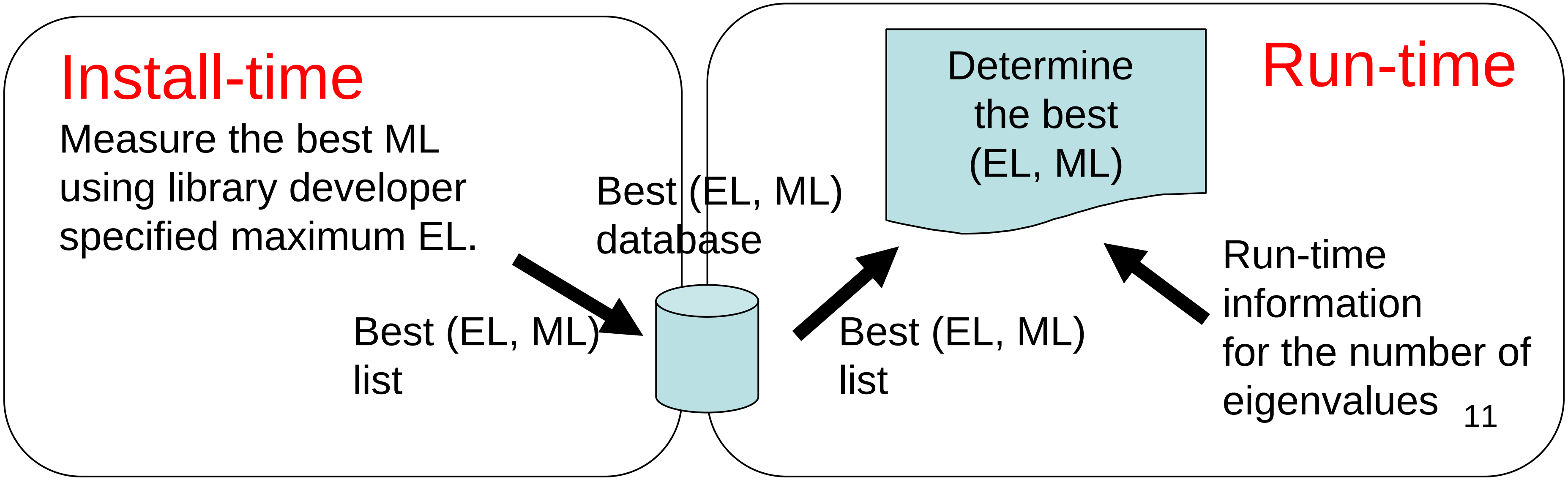
The Parameter Setting Problem of MME

- Parameter Setting Problem: How should we set the two parameters for EL and ML?
- Conventional Problem: Multi-section points (ML) depends on the computer architecture. Hence, we can determine the ML before the routine is called.
- New Problem: However, the number of multiple eigenvalues (EL) depends on the numerical characteristics for input matrices.

⇒ A run-time tuning function is needed to tune MME.

Overall of the New Run-time Auto-tuning Function

- Type: Run-time parameter setting with (1)run-time information for the number of eigenvalues using (2)tuned parameters (Install-time Optimization) in install-time.
- Method: Using an empirical auto-tuning method for install-time auto-tuning to the eigenvalue calculation routine (*DLARRB*) using the MME kernel with a random tri-diagonal matrix.



Overall of the Install-time Optimization Method

1. Measure the normal multi-section kernel time, then find the best ML. Thus, find the best of ml in [1,..,MAXML] with EL=1.
2. Let the best multi-section point be ML*. : Check multi-section time
3. do el=2, MAXEL
    1. Find the best ML using the routine using the MME kernel with el for ml in [1,…,MLE], where MLE in [1,…,MAXML] and MLE el <=ML*. If (el>ML*), then ml=1 is only measured. Let the best ML be ML*_el, and the time be T_el. : Check main problem time
    2. do co-el=1, el-1 : Check co-problem time
        1. Find the best set (EL*_co-best, ML*_co-best) using the routine with MME kernel. The parameters is fixed as (EL(co-el), ML(co-el)) with el. Let the best time be T_co-best.
        2. If (T_co-best < T_el) then
           (EL(el), ML(el)) = (EL*_co-best, ML_co-best);:Co-problem fast
           else
           (EL(el), ML(el)) = (el, ML*_el); :Main problem fast

Note: The comparison is done by the time per eigenvalues.
For example, if the time is *t* with *el* and *ml*, the comparison is done by *t / el.*

Performance Evaluation (1/3)

- Machine: The HITACHI SR8000 1node/8PEs (A SMP for 8PEs)
- Theoretical Peak: 8GFLOPS/node
- Compiler: HITACHI Fortran90 V01-04-/B
- Compiler Option: -opt=4 -parallel=4
- Test Matrices:
  - #1: T=(-1, 2, -1) dimensioned 2100
  - #2: T=(Rnd[0:1]) dimensioned 2100

  } Eigenvalue Distribution : Sparse

  - #3: T= W+ dimensioned 2100
  - #4: T= Glued W+ dimensioned 21, 100times. The glue value is 0.1.Total dimension of the matrix is 2100. ← Very Clustered

Performance Evaluation (2/3)

- Target Process: All eigenvalues and all eigenvalues for the tri-diagonal matrix
- Target routine:
  Total execution time for bisection and MME routines in LAPACK4.0 *DSTEGR* (Hereafter, *GR*) .
  There are two implementations in *GR*:
  - DQDS mode: DQDS for full accuracy of eigenvalues. Using one bisection part.
    - *DLARRV* (Modifying the eigenvector accuracy)
  - Aggressive bisection mode: Using two bisection parts.
    - *DRARRE* (Roughly eigenvalue calculated)
    - *DLARRV* (Modifying the eigenvector accuracy)

Performance Evaluation (3/3)

- Static Parameter Tuning (Hand-tuning)
  - Using a fixed parameter set until the GR routine ends.
  - EL=(1, 2, 3, 4, 8, 16, 32) : 7 kinds
  - ML=(1, 2, 4, 8, 16, 24) : 6 kinds
  - The best time of EL*ML= 42 kinds of combinations is obtained.
- The Proposed Run-time Auto-tuning
  - Using different parameter set according to run-time information.
  - Install-time Optimization
    - Parameters:

      EL=(1, 2,…, 32) : 32 kinds

      ML=(1, 2,…, 24) : 24 kinds
    - Searching space is EL*ML=768. By using the empirical method, the searching space is reduced.
  - Benchmark Matrix in Install-time Optimization
    - The eigenvalue calculation routine with the MME kernel for a tri-diagonal matrix:
    - Dimension: 10,500 : <- L1 Cache Limitation
    - A uniform random matrix with [0:1]
    - One execution time is measured for the target routine.

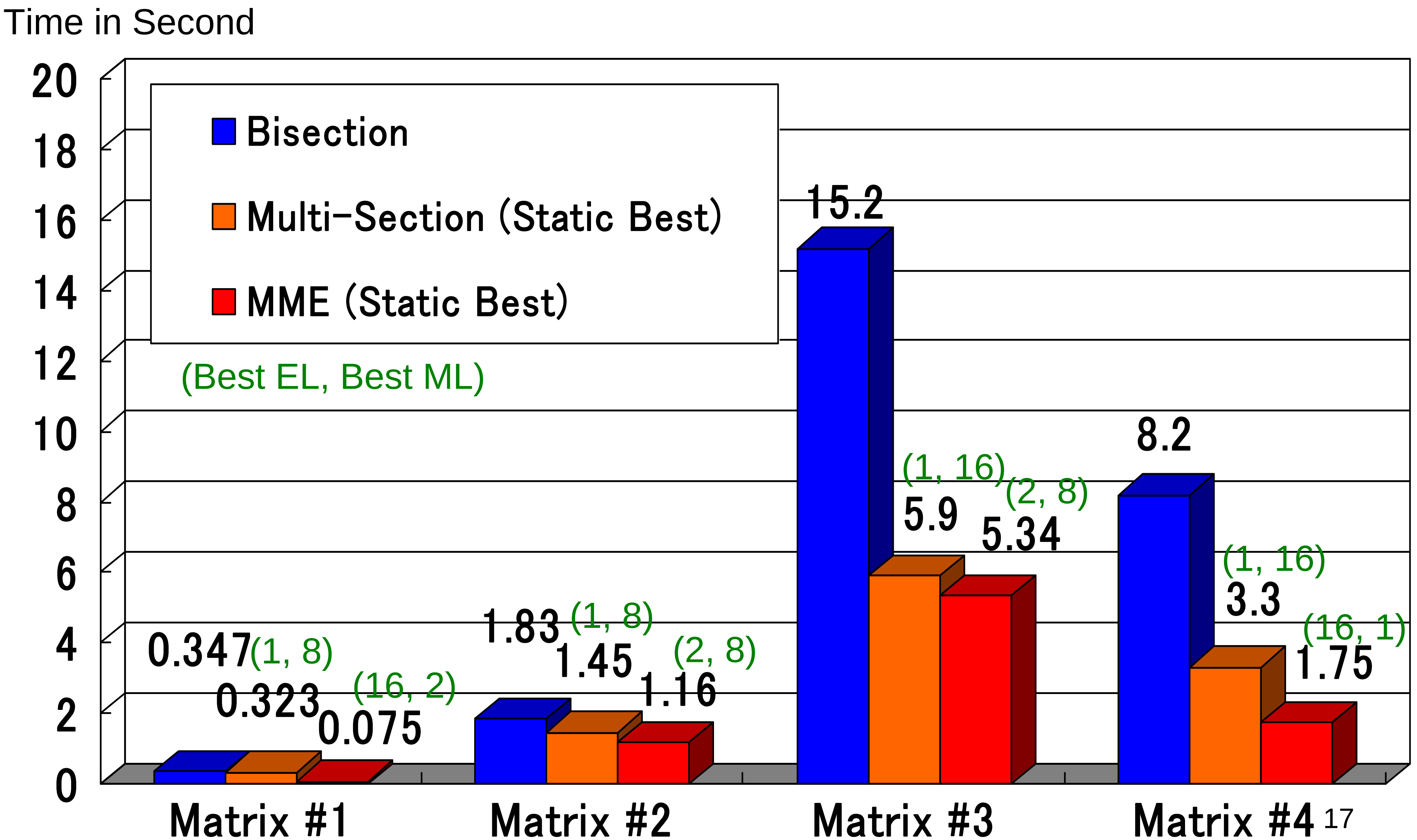
The Auto-Tuned Parameter

Auto-Tuning Time: 129 [second]

#Eig.	EL	ML	
1	1	16	
2	2	8	
3	3	4	
4	4	4	
5	5	3	
6	5	3	:Co-Prob. Fast
7	7	2	
8	8	2	
9	9	1	
10	10	1	
11	11	1	
12	12	1	
13	13	1	
14	14	1	
15	15	1	
16	16	1	
17	16	1	:Co-Prob. Fast
18	16	1	:Co-Prob. Fast
19	16	1	:Co-Prob. Fast
20	16	1	:Co-Prob. Fast

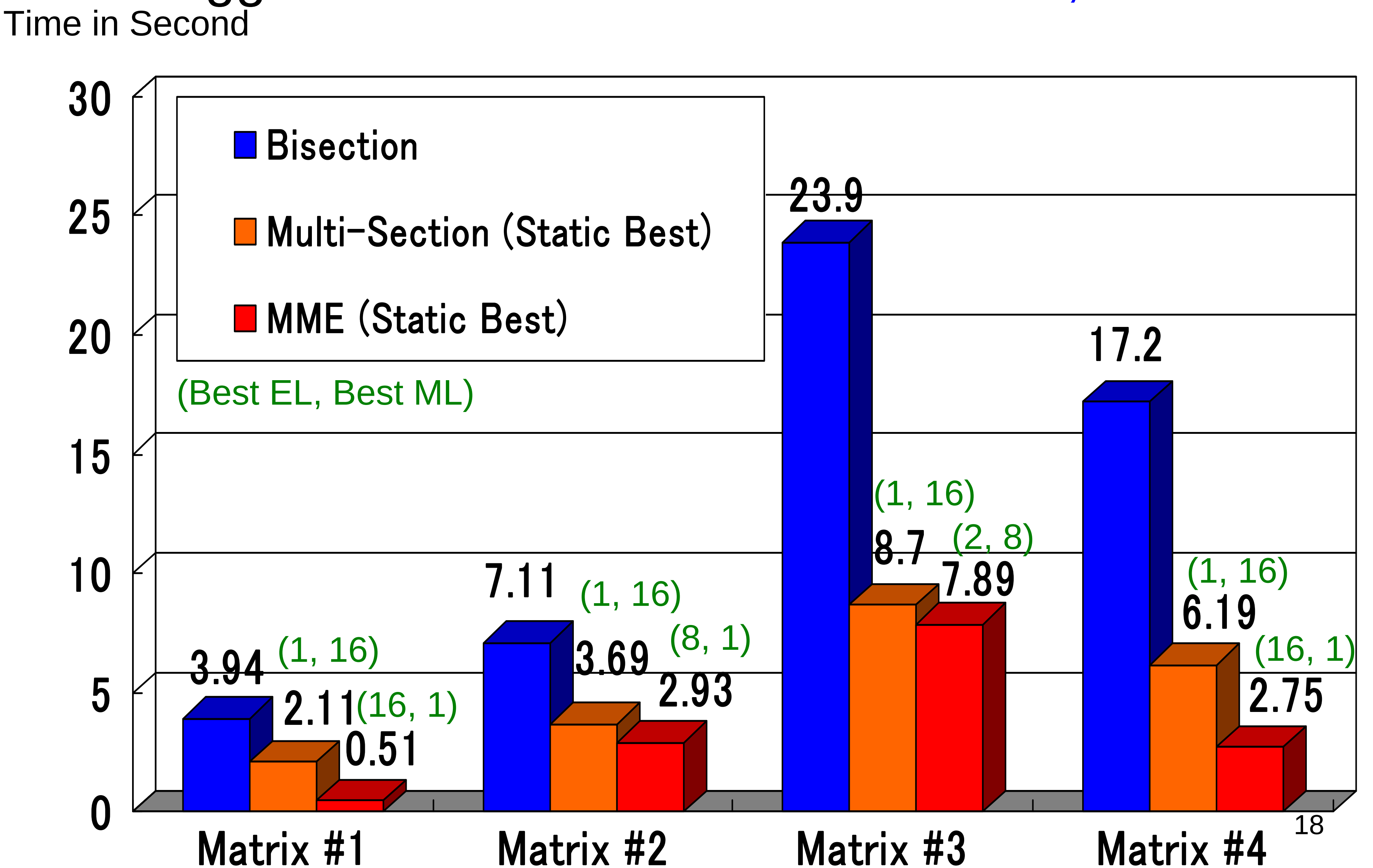
#Eig.	EL	ML	
21	21	1	
22	22	1	
23	23	1	
24	24	1	
25	24	1	:Co-Prob. Fast
26	13	1	:Co-Prob. Fast
27	27	1	
28	14	1	:Co-Prob. Fast
29	29	1	
30	15	1	:Co-Prob. Fast
31	15	1	:Co-Prob. Fast
32	16	1	:Co-Prob. Fast

Effect on Static Tuned MME (1/2) – DQDS Mode : SR8000, N=2100

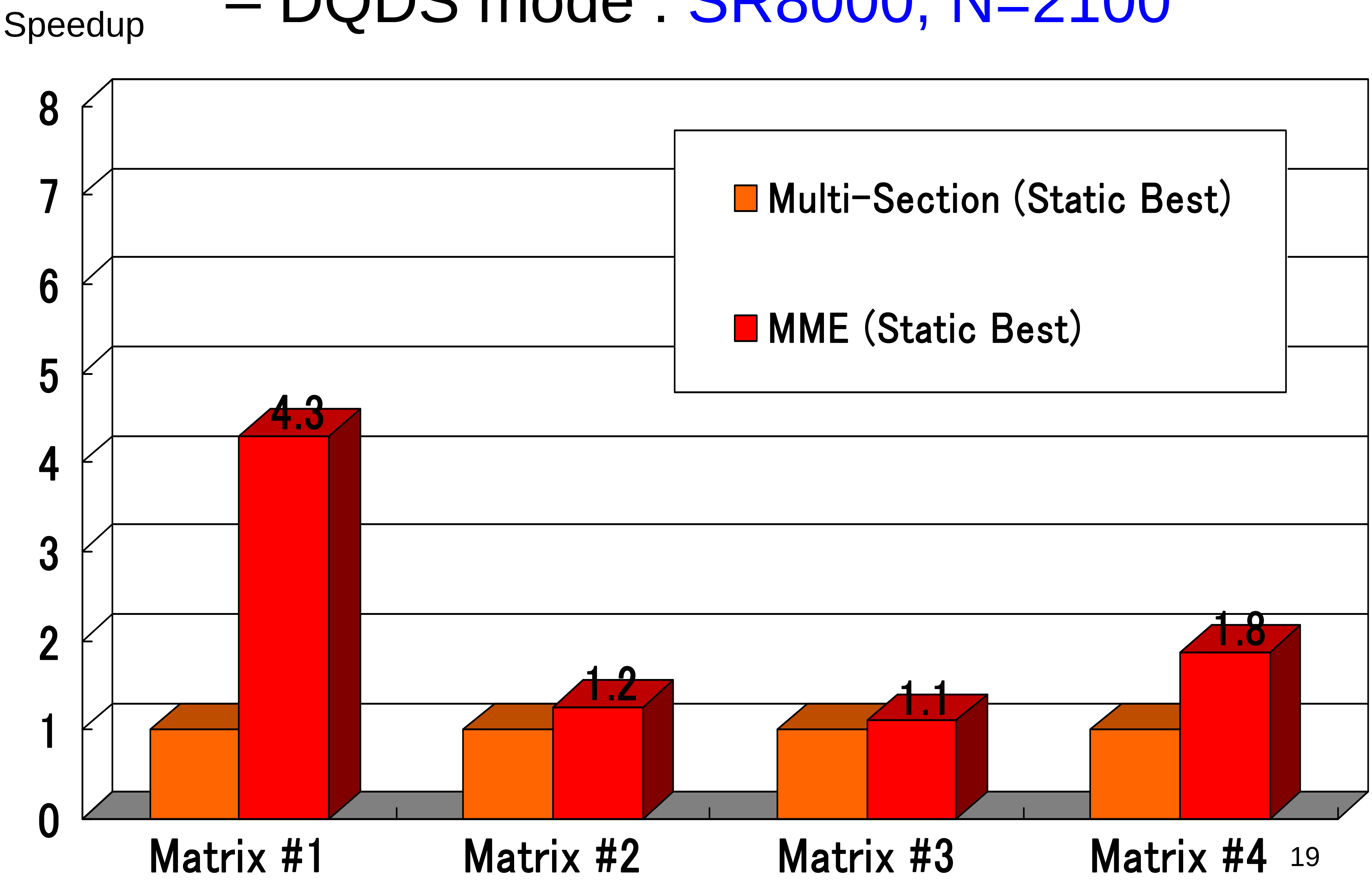


Effect on Static Tuned MME (2/2)

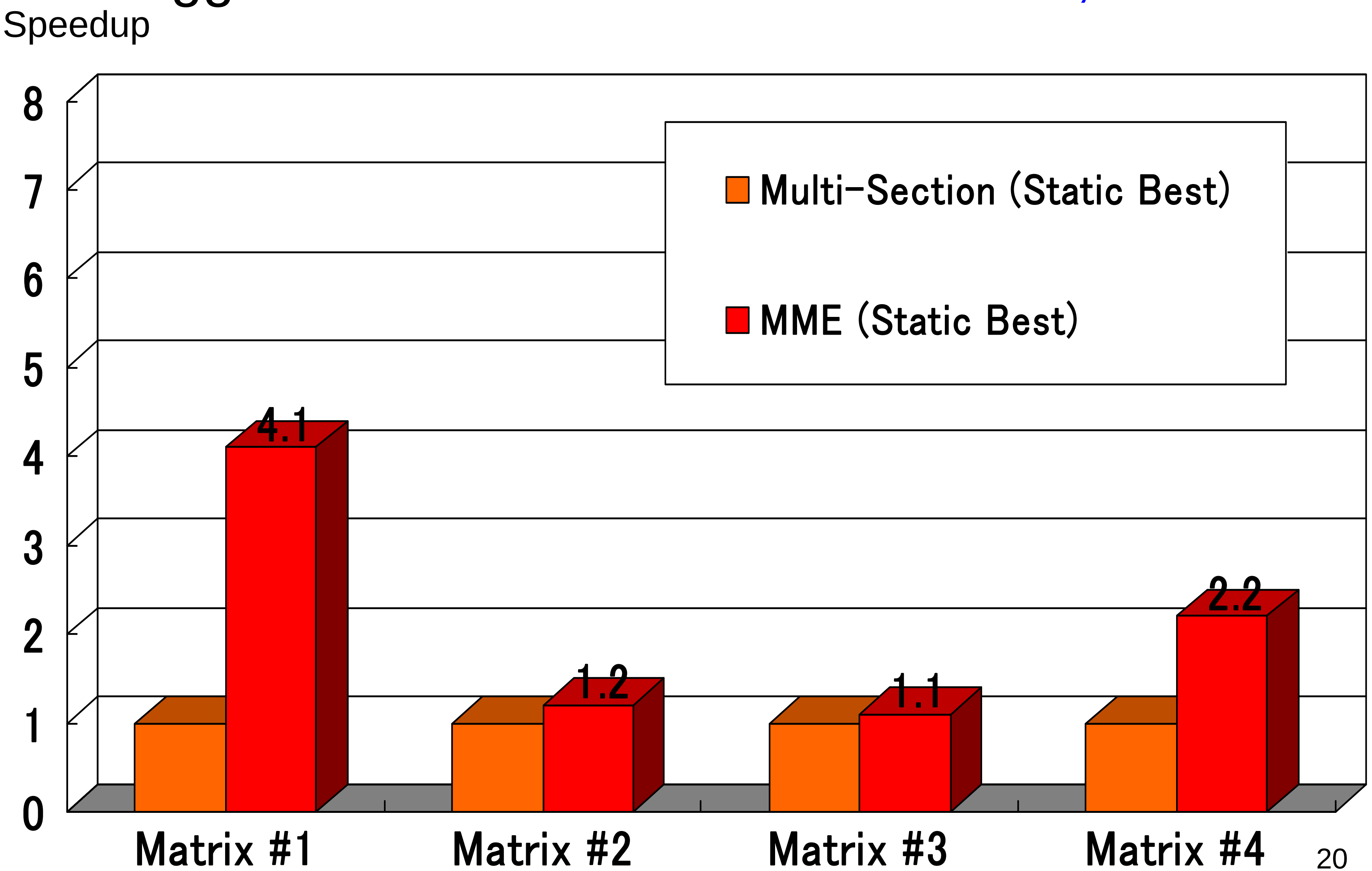
– Aggressive Bisection Mode: SR8000,N=2100



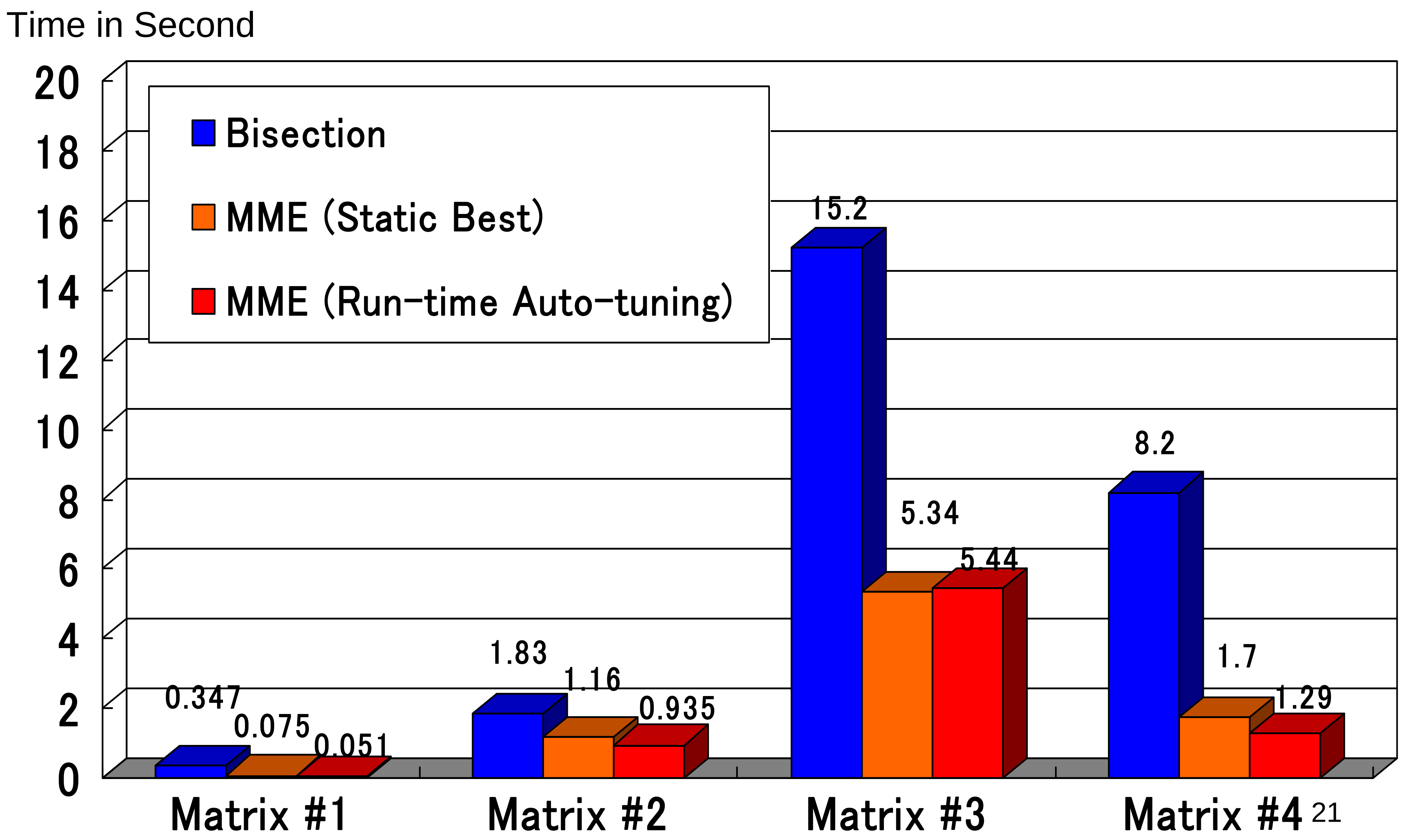
Speedup of Static Tuned MME (1/2) – DQDS mode : SR8000, N=2100



Speedup of Static Tuned MME (2/2)
– Aggressive bisection mode : SR8000, N=2100

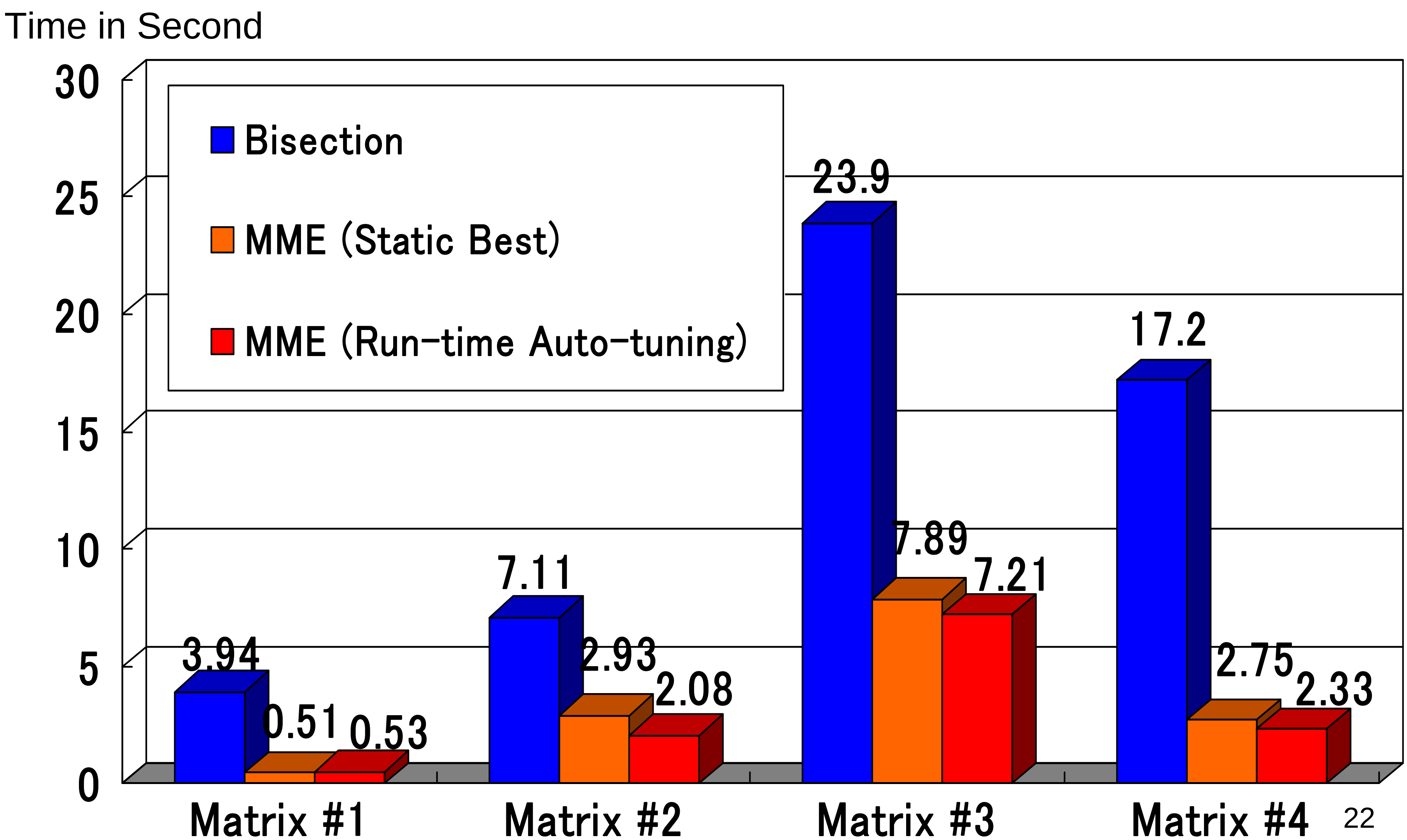


Effect on Run-time Auto-Tuning to MME (1/2) – DQDS mode : SR8000, N=2100

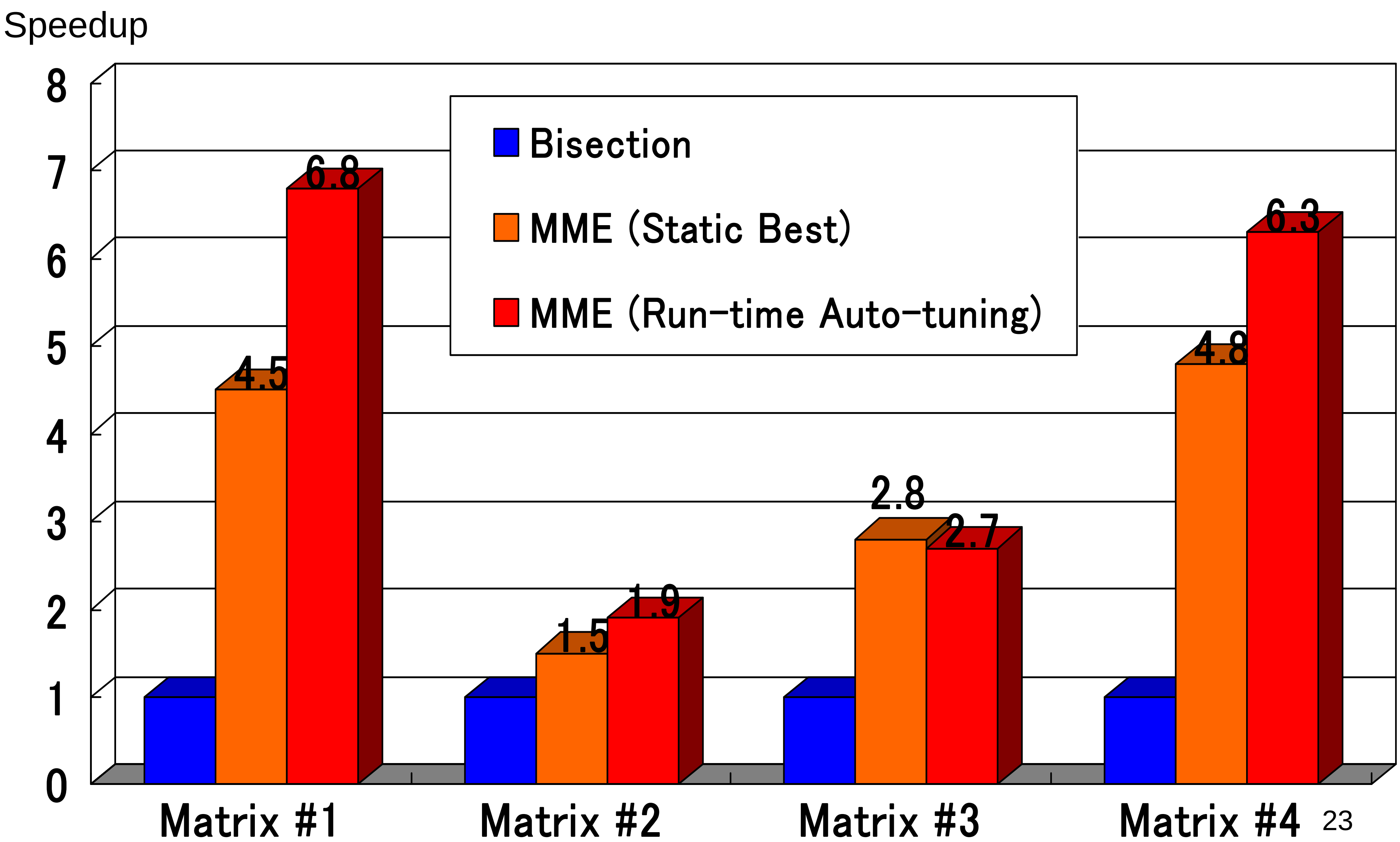


Effect on Run-time Auto-Tuning to MME (2/2)

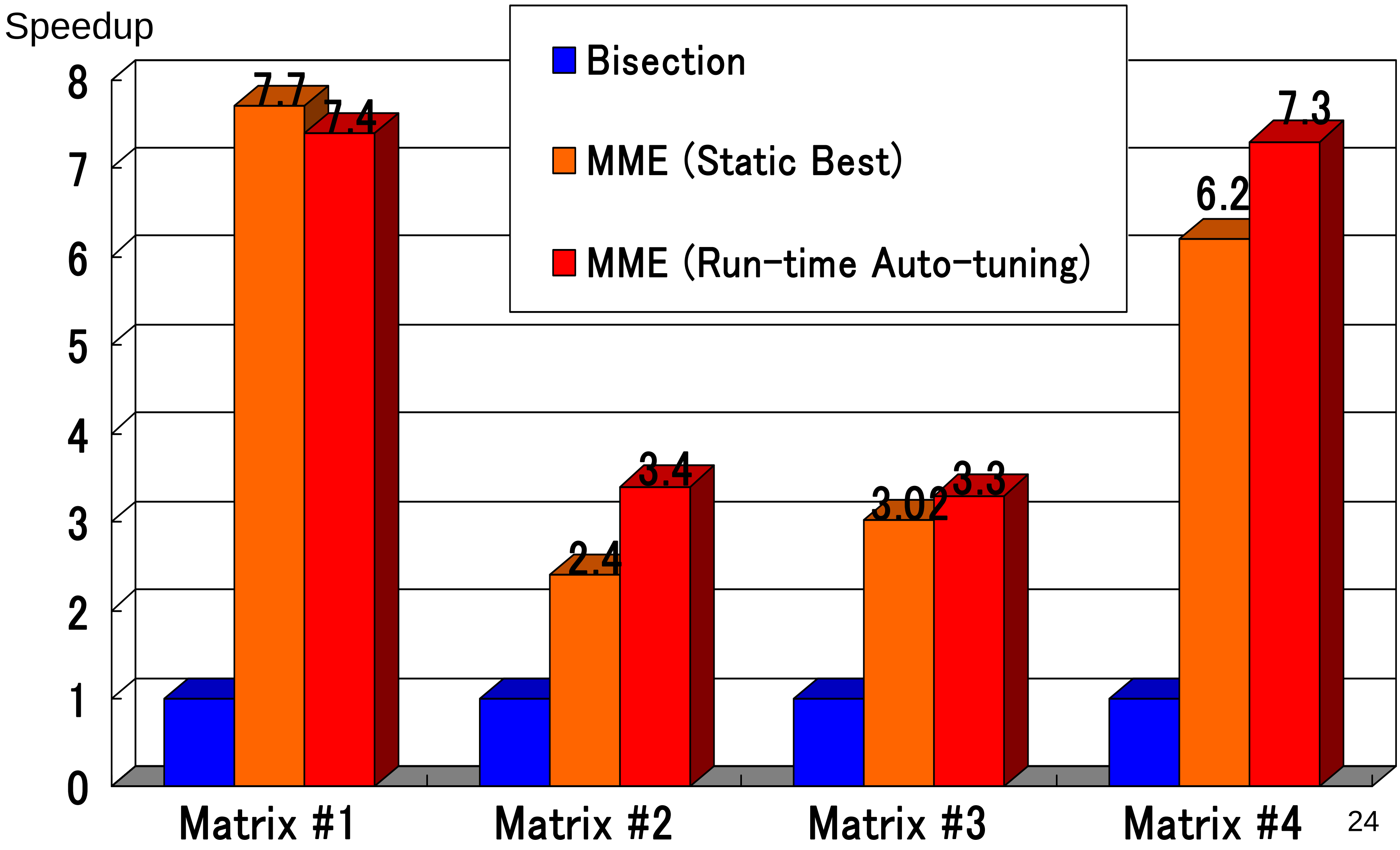
– Aggressive bisection mode : SR8000, N=2100



Speedup of Run-time Auto-Tuning to MME (1/2) – DQDS mode : SR8000, N=2100



Speedup of Run-time Auto-Tuning to MME (2/2)
– Aggressive bisection mode : SR8000, N=2100



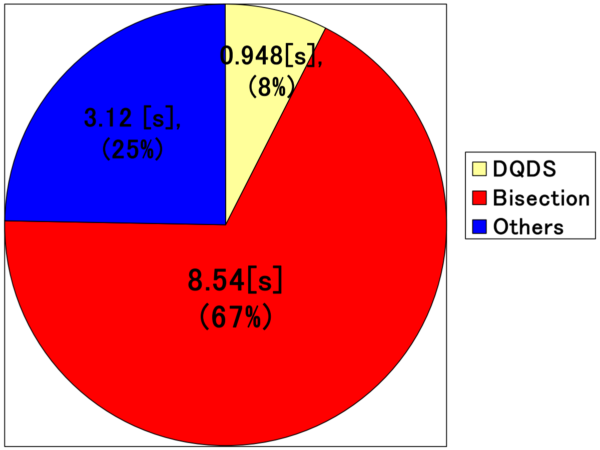
Conclusion (1/2)

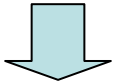
- Most of performance using the run-time auto-tuning was almost same as the static best case of MME.
  - Some cases of the run-time auto-tuning were faster than the static tuned cases. There was a case of 1.7x speedup to the static best MME.
  - Static tuning is impossible to use actual numerical libraries: The best parameter strongly depends on input matrix.
  - So, the proposed run-time auto-tuning method is crucial function for numerical libraries.

Conclusion (2/2)

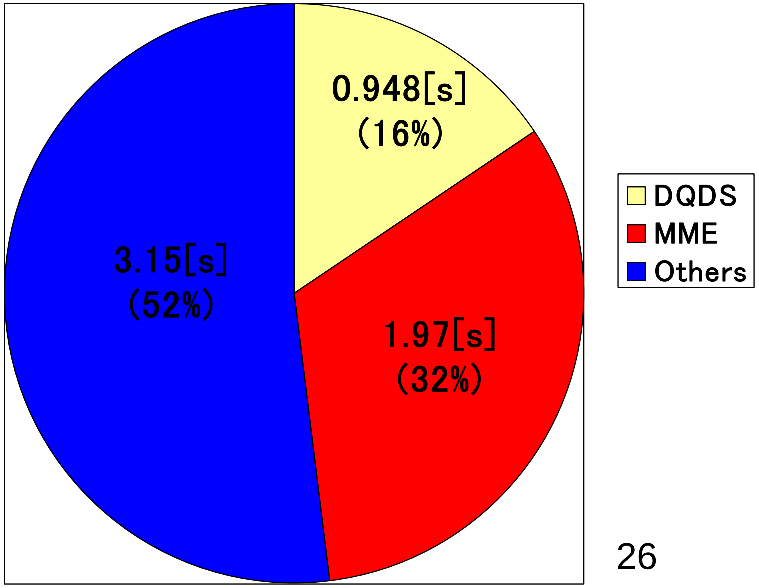
- The bisection routine is not bottle-neck any more on the HITACHI SR8000.
- The routines of DQDS and other part (may be DLARRF, which is computing of RRR of child cluster) will be bottle-neck on the HITACHI SR8000.
- We need to parallelize them.

Glued Wilkinson +21 Matrix





Glued Wilkinson +21 Matrix
With MME on the 1node/8PEs



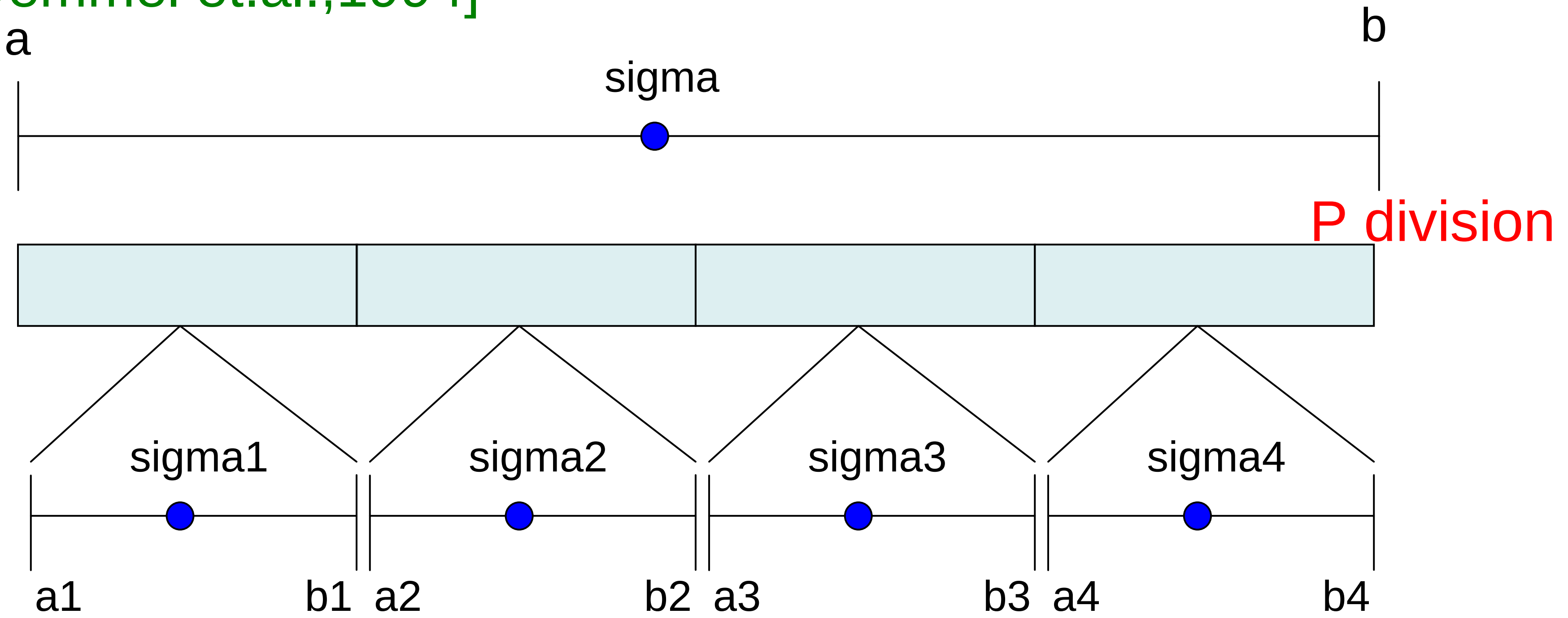
HITACHI SR8000
1node/8PE DATA

Future work

- Evaluation of performance for the empirical run-time auto-tuning method using several SMP parallel environments.
- Implementation of the run-time auto-tuning method using LAPACK API.
- Adapt and evaluate the run-time auto-tuning framework to the other numerical kernels.

Parallelizing The Bisection Kernel (1 / 2)

- Method 1 : Dividing The Interval for Bisection [Demmel et.al.,1994]



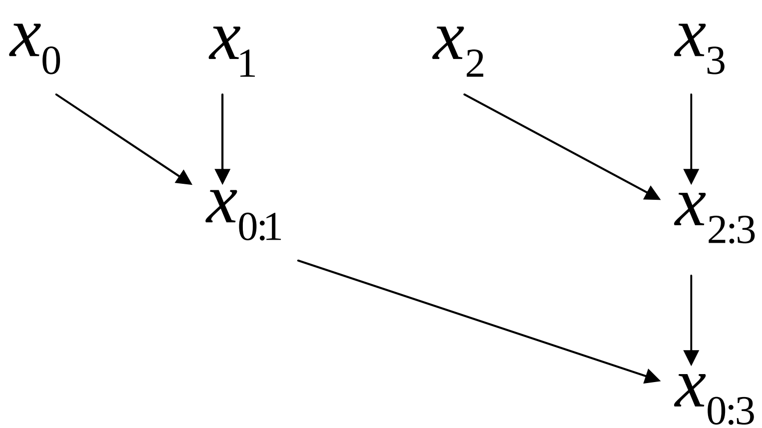
- Drawbacks:
  - The kernel can not be vectorized.
  - The parallel efficiency will be down, if eigenvalues are clustered.

Parallelizing The Bisection Kernel (2 / 2)

- Method 2 : Cyclic Reduction Method (Parallel Prefix Method) for the polynomias of

$$p_k(x) = \det(T_k - xI)$$ [Ren,1996]

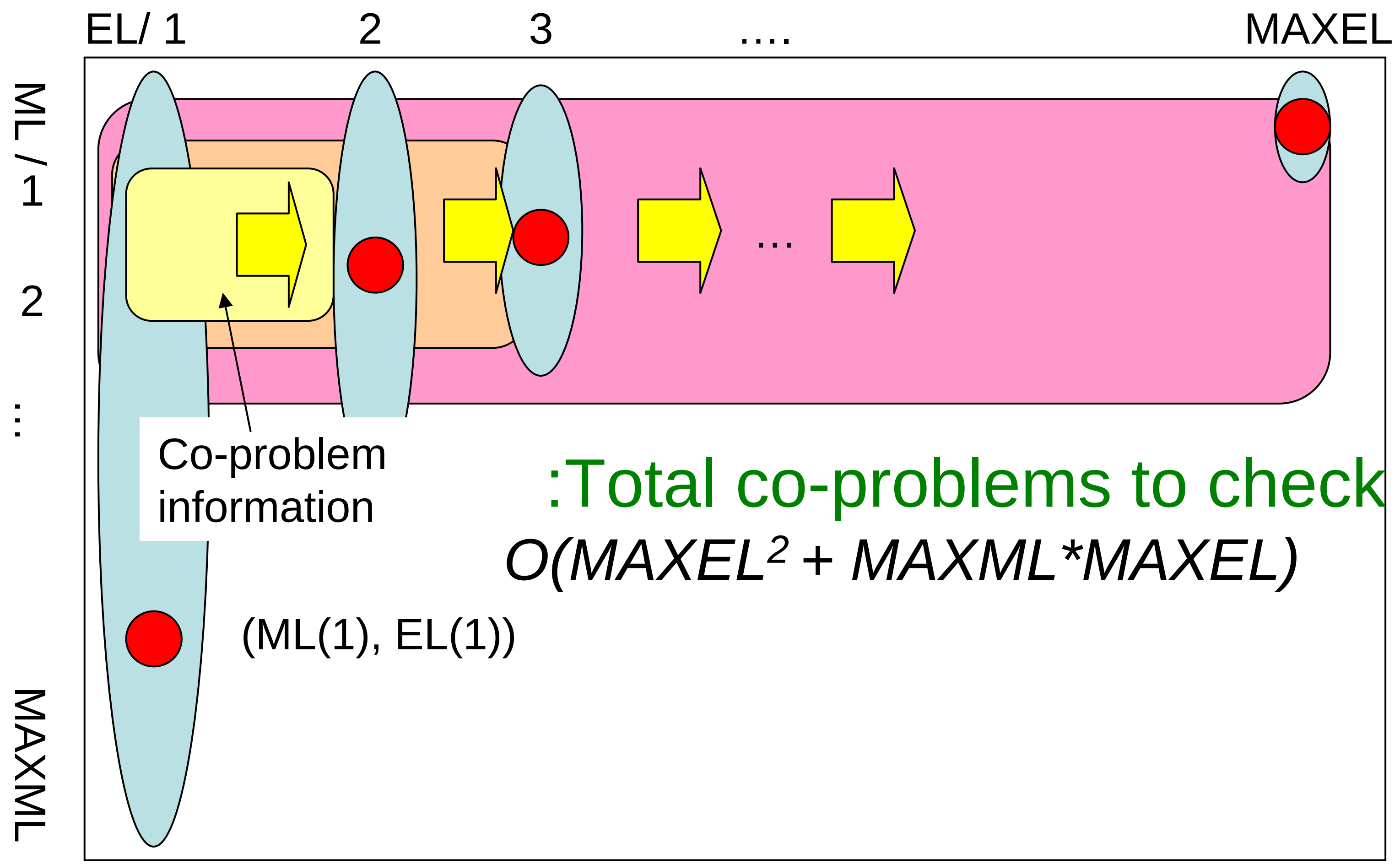
$$\rightarrow \quad p_k(x) = \det(a_k - x)\, p_{k-1}(x) - b_{k-1}^2\, p_{k-2}(x)$$



O(log k) Parallelism

- Merit: The kernel can be parallelized and vectorized.
- Drawback: The method has numerical instability.

Process Flow of the Install-time Optimization Method

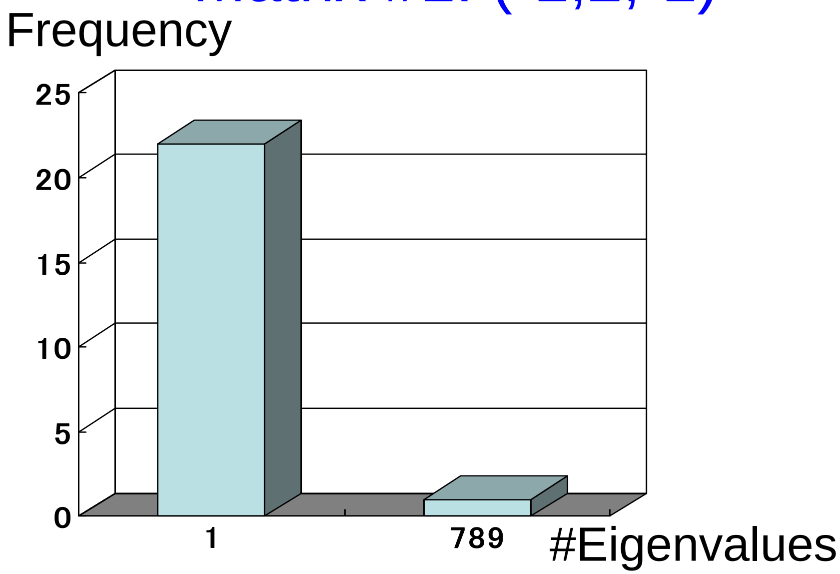


Run-time Auto-tuning Details

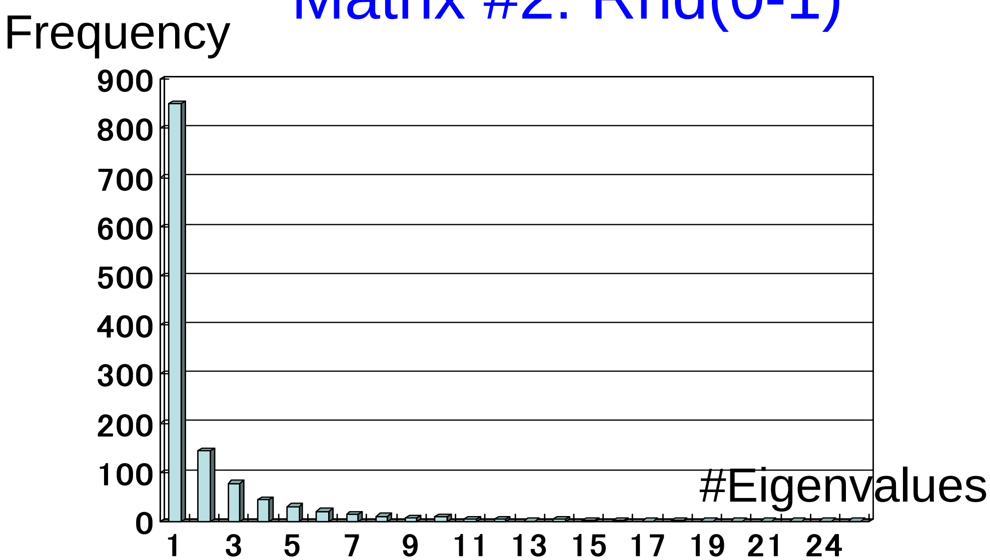
1. Check the required number of eigenvalues. Let the number be el.
2. Search the list of (EL(el), ML(el)) for el in [1,…,MAXEL].
3. If (el<=MAXEL), then (EL(el), ML(el)) are the best parameter set.
4. If (el>MAXEL), then (EL(MAXEL),ML(MAXEL)) are the predicted best parameter set.

The Distribution of The Number of Eigenvalues in *dlarrb* routine (DQDS Mode)

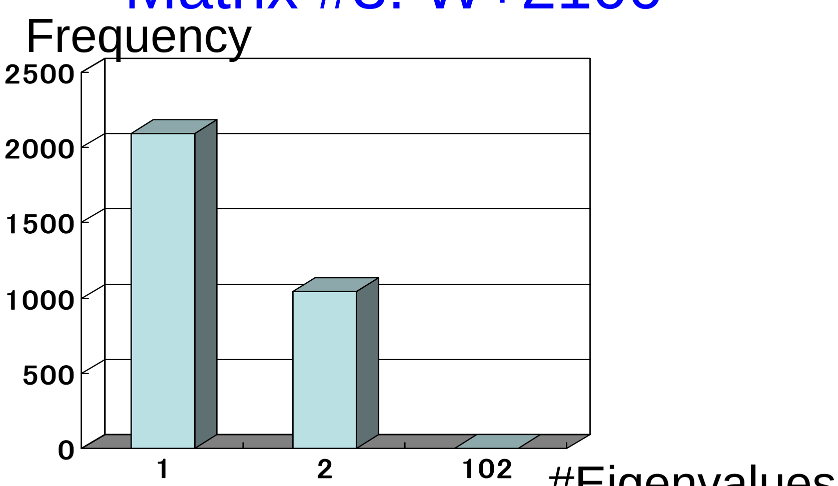
Matrix #1: (-1,2,-1)



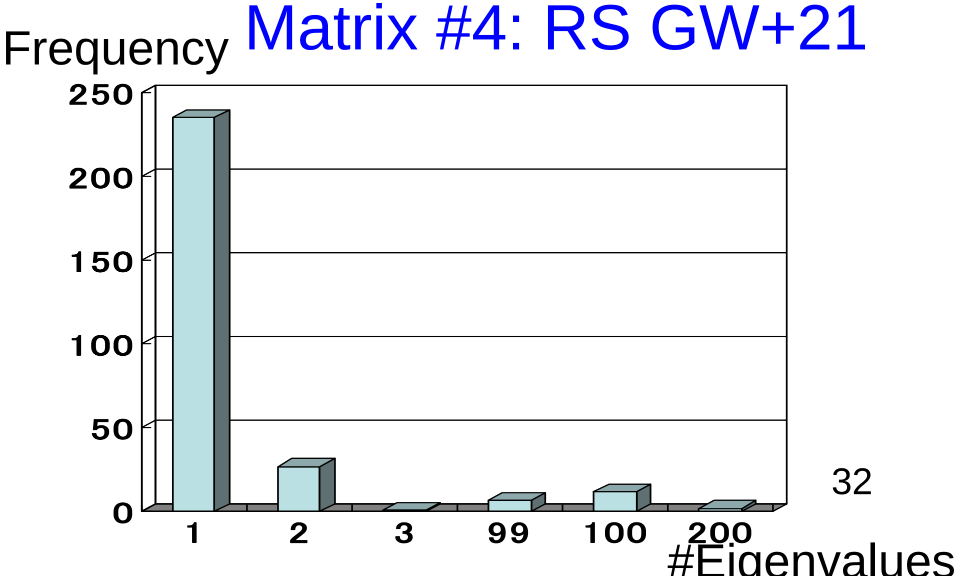
Matrix #2: Rnd(0-1)



Matrix #3: W+2100



Matrix #4: RS GW+21



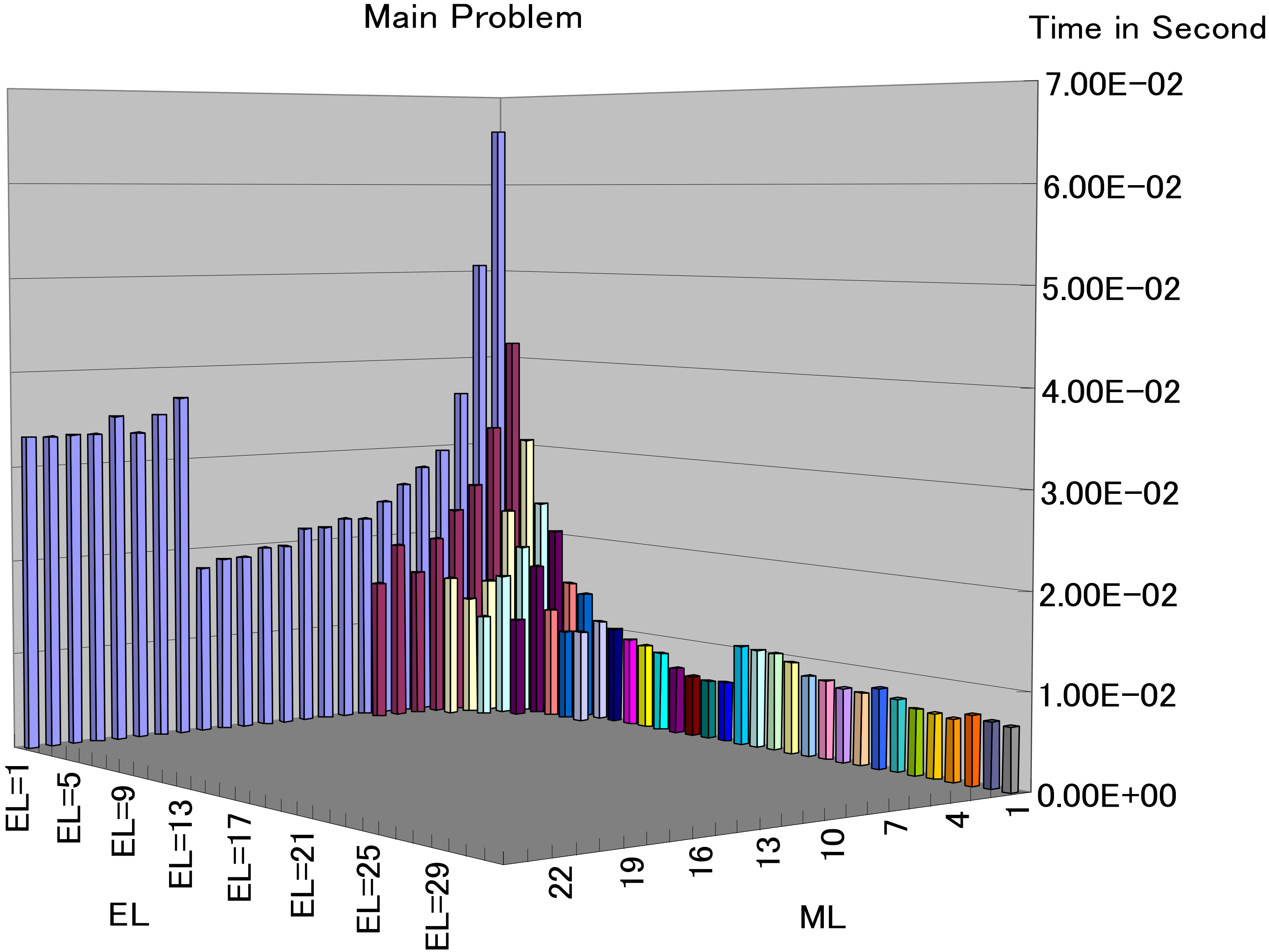
Appendix A: A Static Tuning Log

Matrix #4, Using dqds case

EL/ ML/	1 (MS)	2	3	4	8	16	32
1 /	8.200	5.418	4.127	3.432	2.361	**1.758**	1.760
2 /	6.348	4.249	3.261	6.357	1.961	1.968	1.962
4 /	4.674	3.034	2.392	2.010	2.075	2.098	2.114
8 /	3.532	2.376	3.200	8.200	2.662	2.703	2.797
16 /	3.305	3.975	4.183	4.236	4.355	4.427	4.569
24 /	5.488	5.183	6.035	5.616	5.774	5.971	6.106

EL*ML=16 is an empirical condition on the HITACHI SR8000

Appendix B: Auto-tuning Log – Main Problem



Appendix B: Auto-tuning Log – Co-Problem