SMAT：
An Input Adaptive Auto-Tuner for Sparse Matrix-Vector Multiplication

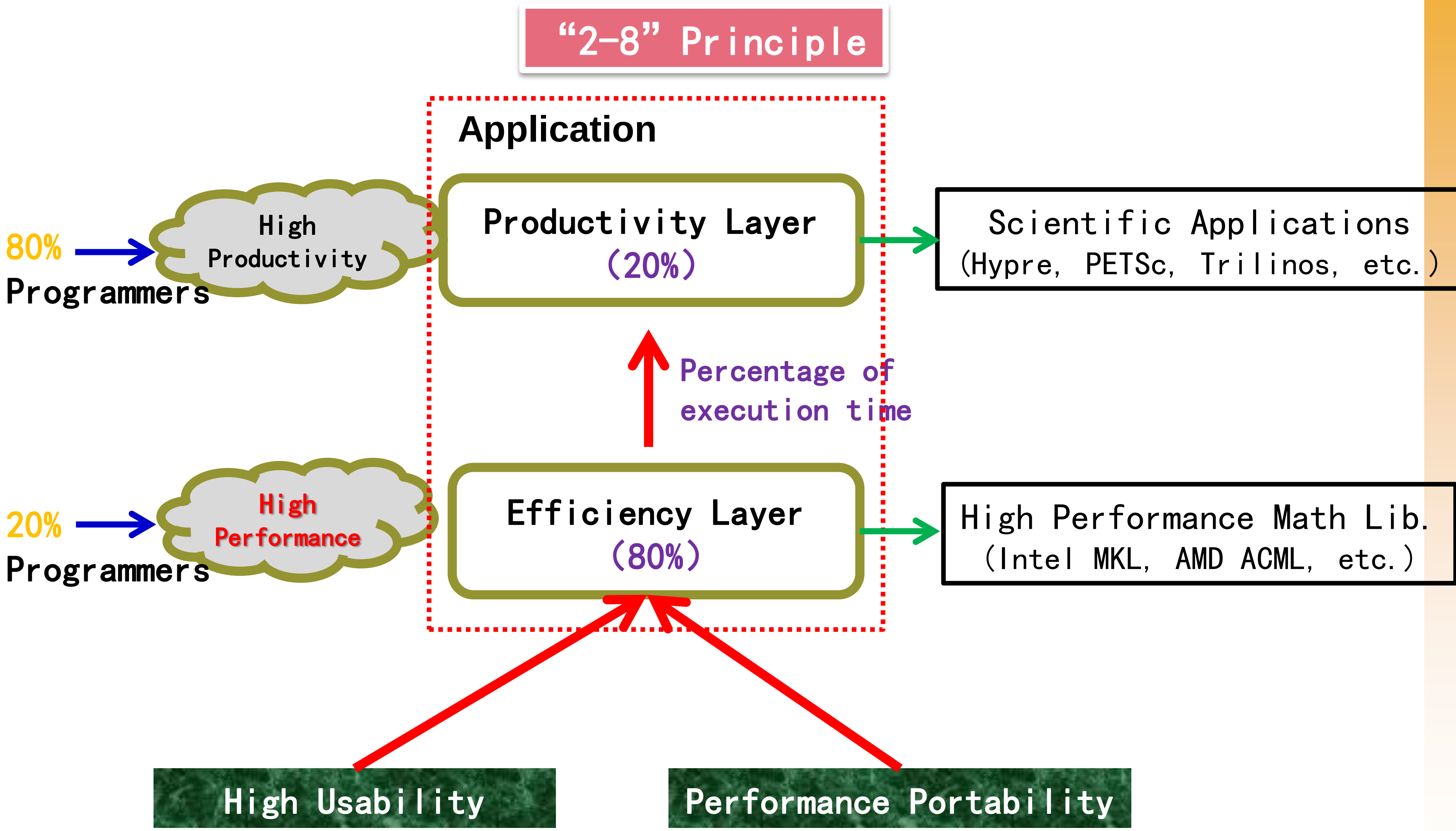
Jiajia Li, Guangming Tan, Mingyu Chen, Ninghui Sun

Institute of Computing Technology,
Chinese Academy of Sciences

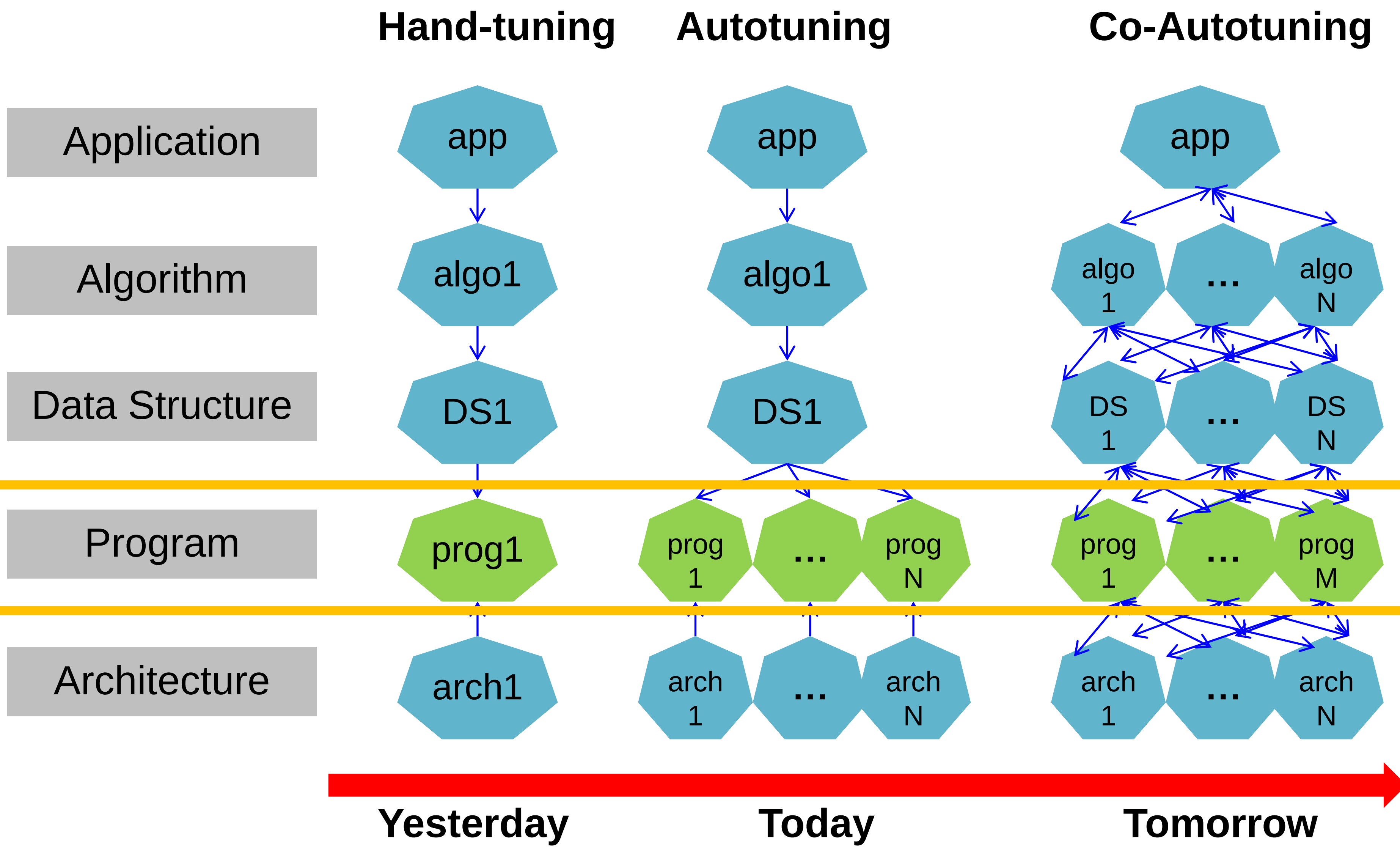
中国科学院计算技术研究所
Institute of Computing Technology,Chinese Academy of Sciences

Contents

High Performance Computational Software Development-1

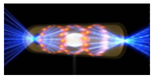


High Performance Computational Software Development-2



Killer Applications

Computational Science

ITER


Climate

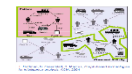

Oil


Exascale

Big Data

Data Science

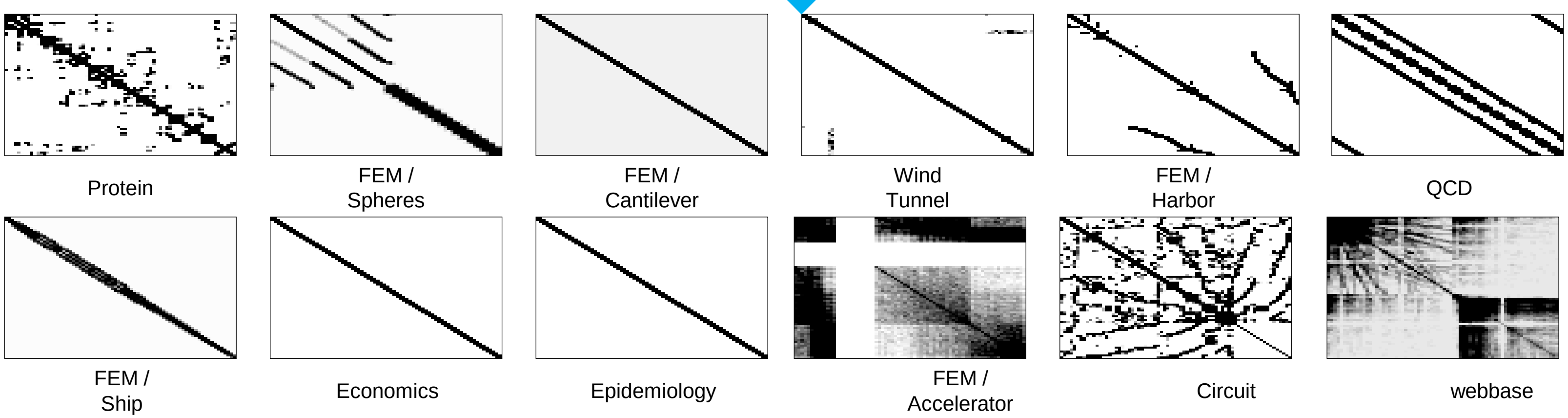
Social network

National Security


System Biology


Sparse Linear System

Sparse Matrices



Sparse Matrix

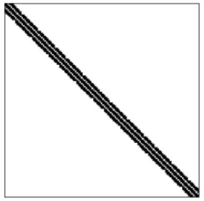
- Diverse Application Background
- Different Solving Methods

} ⇨

Diff. Nonzero Distribution Structure

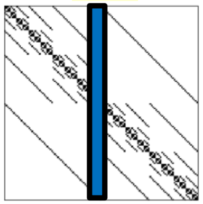
⇩

- Kinds of Sparse Matrices
  - Diagonal Matrix
  - "Slim" Matrix
  - "Fat" Matrix
  - Power-Law Matrix
  - Matrix with Dense Blocks
  - …

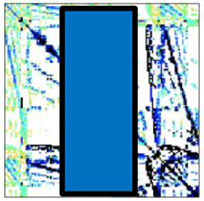


Diagonal
"pcrystk02"

2

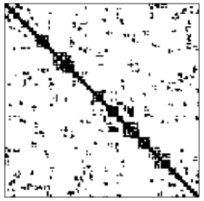


"Slim"
"Bfly"



"Fat"
"crankseg_2"

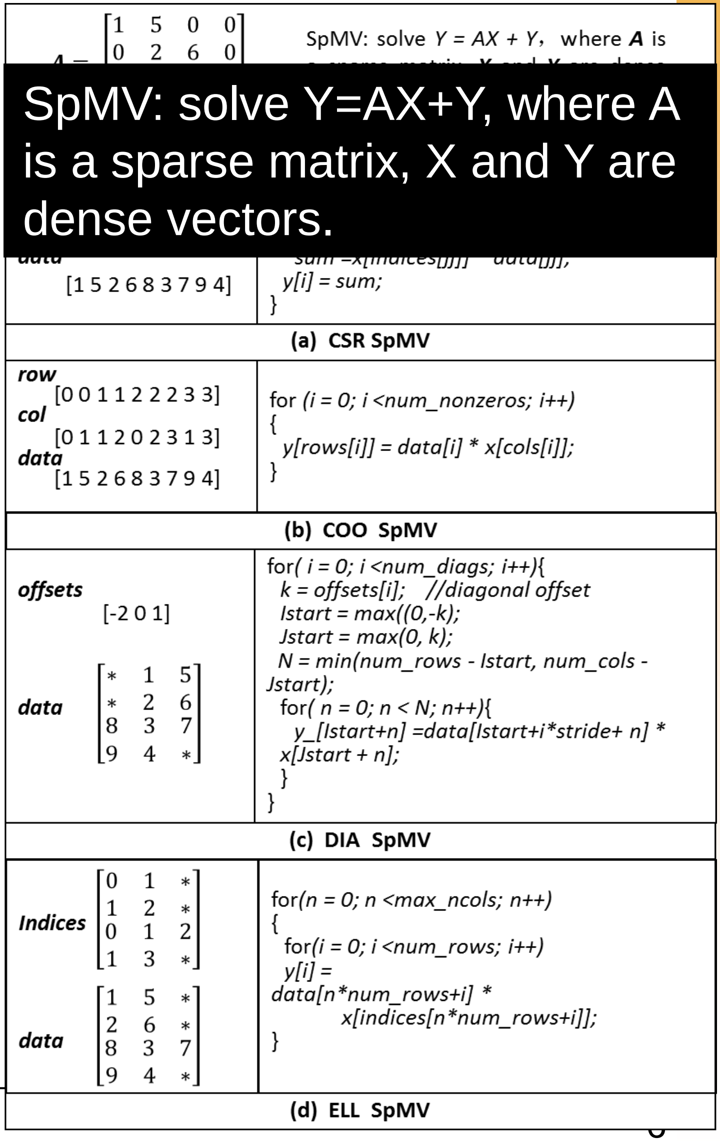
222



Power-Law
"roadNet-CA"

number of nonzeros per row

Storage Formats



SpMV: solve Y=AX+Y, where A is a sparse matrix, X and Y are dense vectors.

Motivation

- **Sparse solvers mainly use ONE storage format CSR**
  - Hypre (LLNL), PETSc (ANL), Trilinos (SNL)

Low Performance

GAP!

(*csr* is the only one exposed to users)

smat_<precision > <csr><operation> ( )

mkl_<precision > <format> <operation> ( )

(variants of *format*)

High Performance
High Productivity

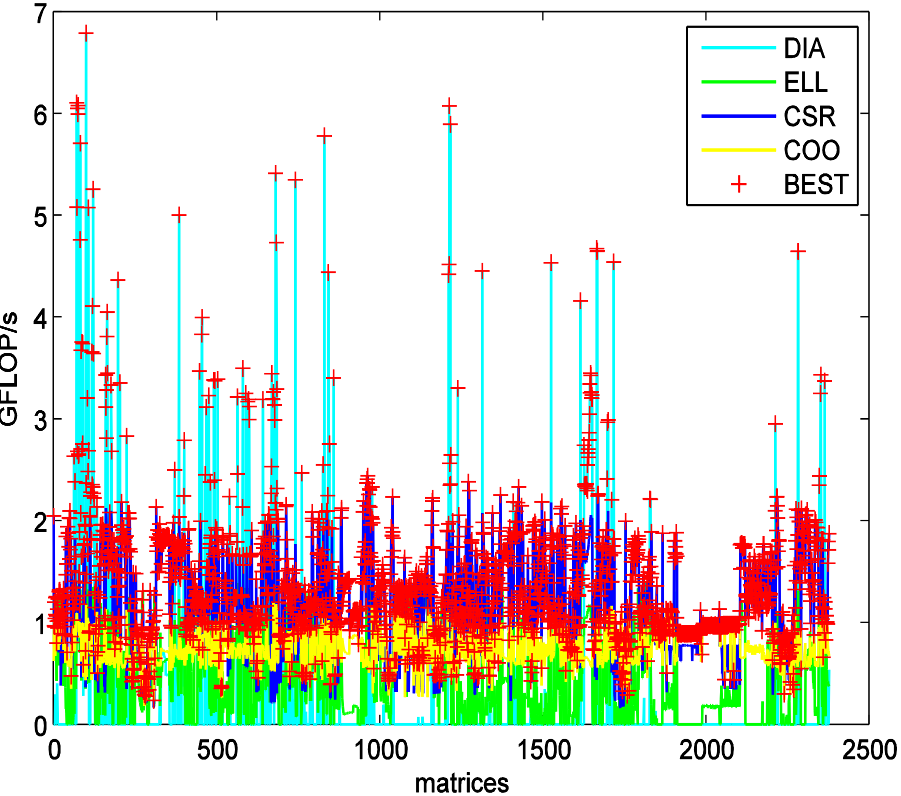
- **Libraries provide complicated interfaces to users**
  - MKL (Intel), OSKI (UCB), SpBLAS (UTK)

Low Productivity

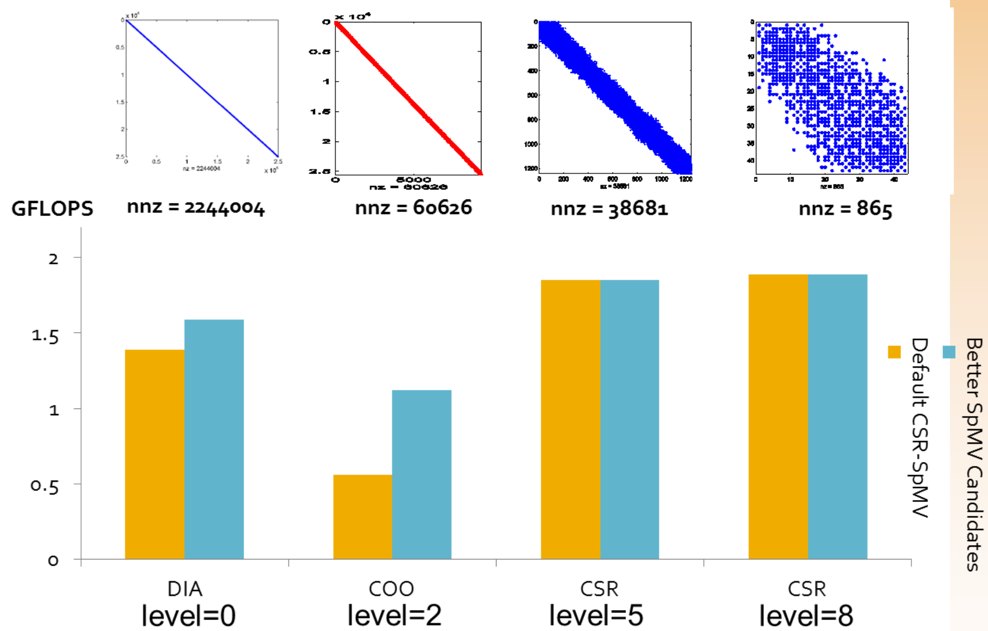
Motivation cont.

Observation 1: The optimal formats are diverse for sparse matrices from different application areas.

Observation 2: Different formats are needed in different stages of one application during runtime.



Algebraic Multigrid (AMG) Solver



Performance Gap: 10x!

Application-Architecture Aware Auto-tuner Design

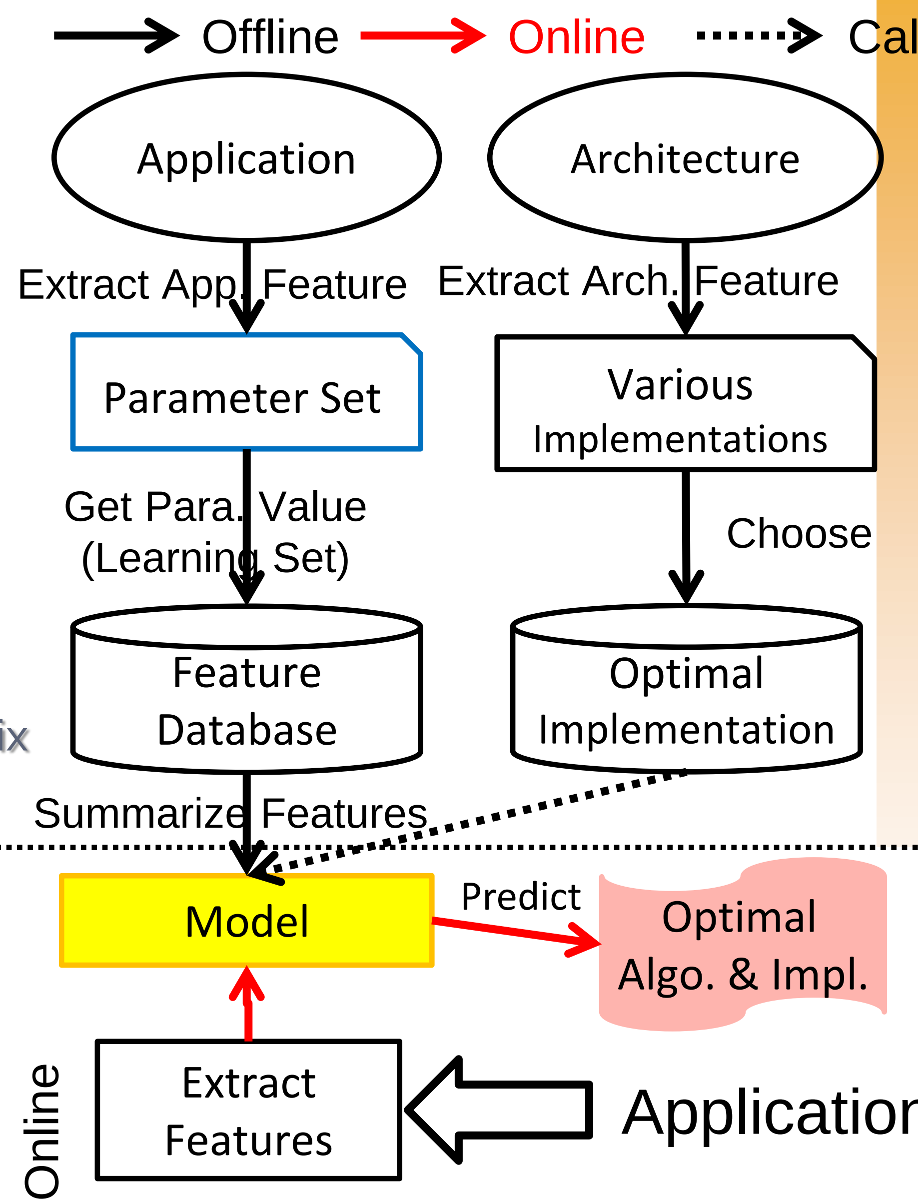
- **Offline**
  - Extract application
  - Determine feature
  - Build feature databa (including the optima implementation)
  - Summarize the rules
  - Choose the optimal ir architecture characte
- **Online**
  - Extract parameter values of the input matrix
  - Predict the optimal algo. & impl. based on model

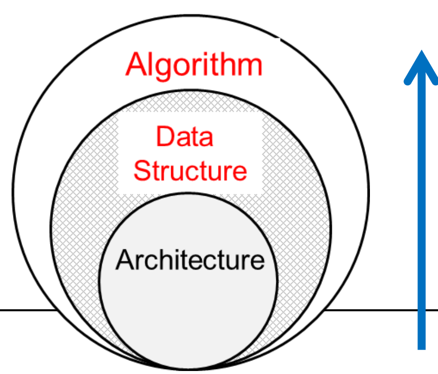
Sparse App.

- Matrix Dimension
- Diagonal Situation
- Nonzero Distribution
- Nonzero Fill Ratio

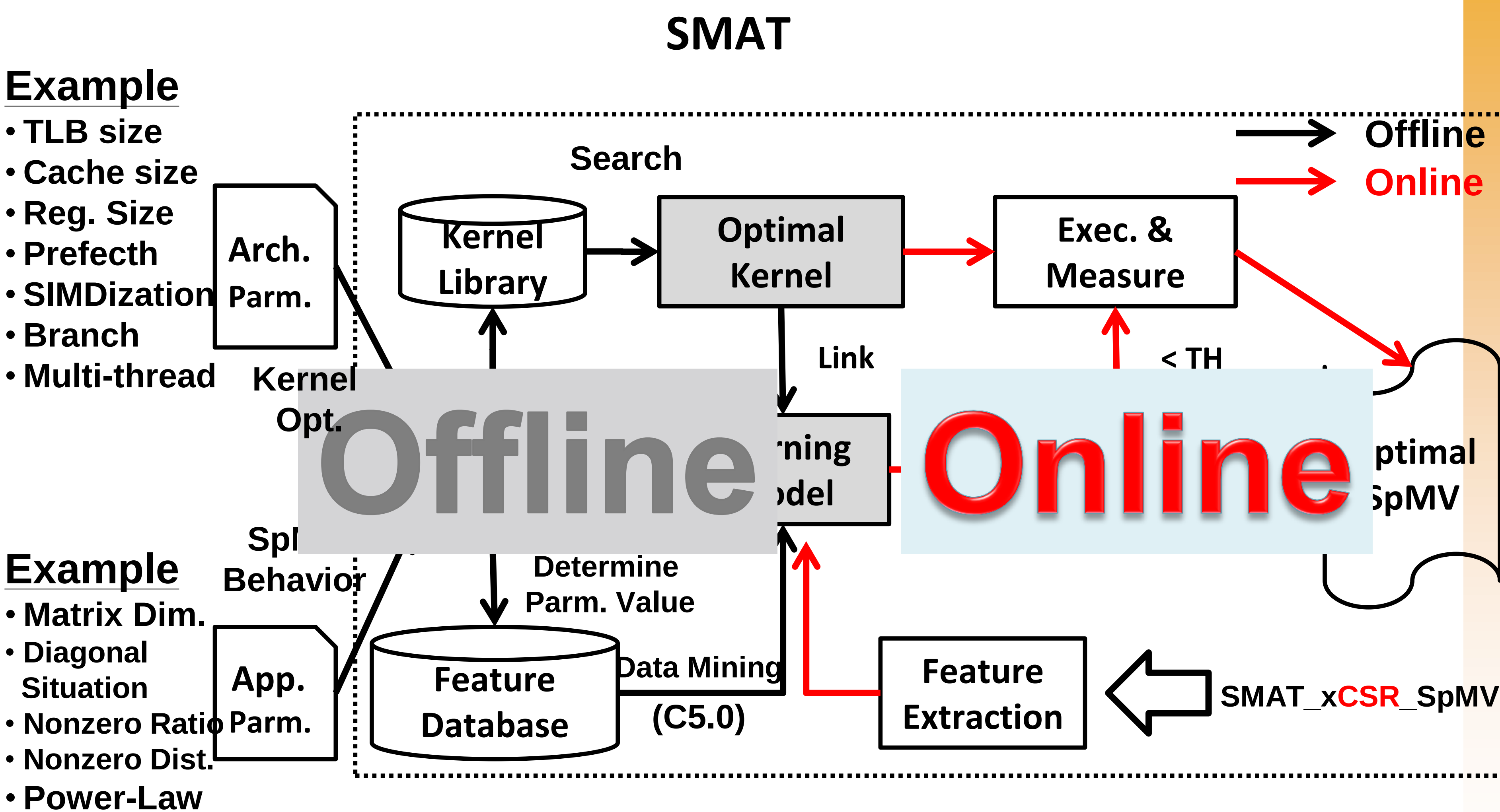
Graph App.

- Dimension
- Degree Distribution
- Diameter
- Un-/directed
- Power-Law
- Connectivity





SMAT Framework

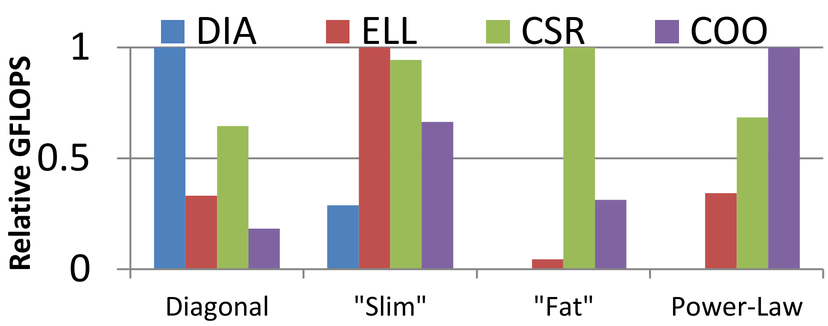


SMAT API

Intel MKL		SMAT
mkl_xcsrgemv mkl_xdiagemv mkl_xbsrgemv mkl_xcscmv mkl_xcoogemv mkl_xskymv	→	SMAT_xCSR_SpMV

So Easy!

Matrix Feature Parameters



Generally:

- Diagonal Matrix ↔ DIA Format
- "Slim" Matrix ↔ ELL Format
- "Fat" Matrix ↔ CSR Format
- Power-Law Matrix ↔ COO Format

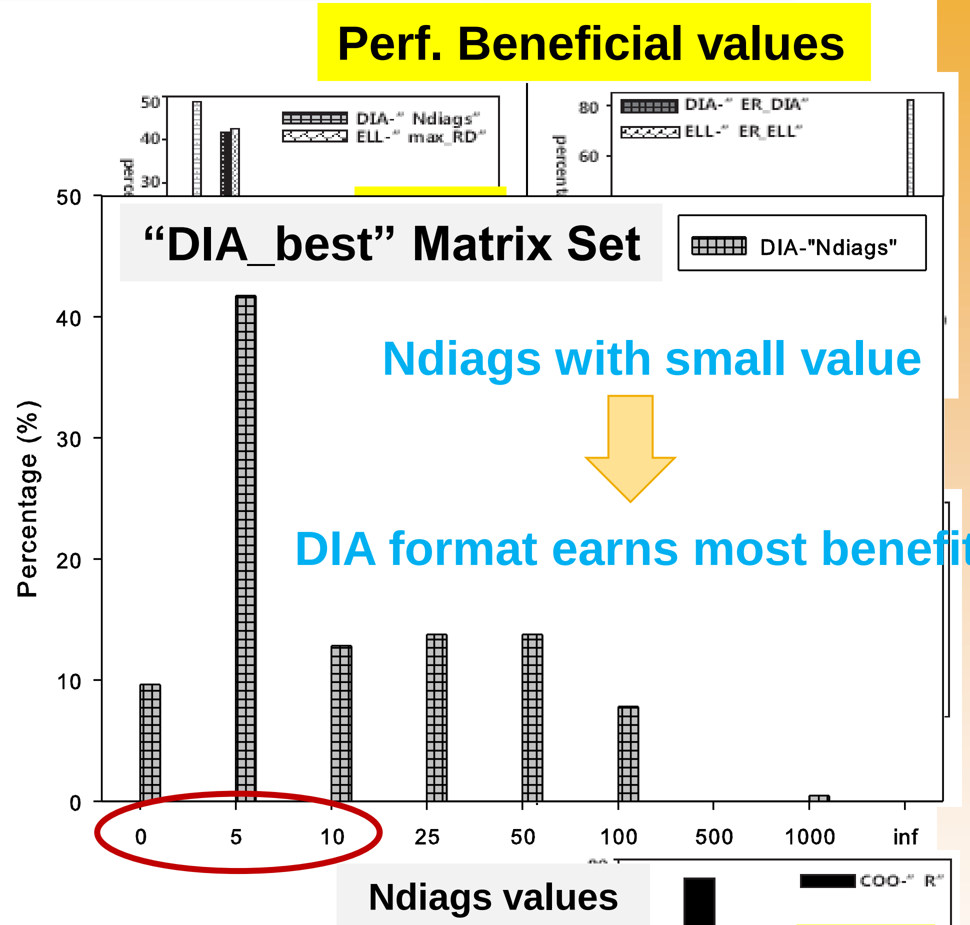
Parameter		Meaning	Formula	DIA	ELL	CSR	COO
Matrix Dimension	M	the number of rows	-	√	√	√	√
	N	the number of columns	-	√	√	√	√
Diagonal Situation	Ndiags	the number of diagonals	-	↓			
	NTdiags_ratio	the ratio of "true" diagonals to total diagonals	$NTdiags\_ratio = \frac{number\ of\ \text{"}true\ diagonals\text{"}}{Ndiags}$	↑			
Nonzero Distribution	NNZ	the number of nonzeros	-	√	√	√	√
	aver_RD	the number of nonzeros per row	$aver\_RD = \frac{NNZ}{M}$	√	√	√	√
	max_RD	the maximum number of nonzeros per row	$max\_RD = \max_1^M \{number\ of\ nonzeros\ per\ row\}$		↓		
	var_RD	the variation of the number of nonzeros per row	$var\_RD = \frac{\Sigma_1^M \lvert row\_degree - aver\_RD \rvert^2}{M}$			↓	
Nonzero Ratio	ER_DIA	the ratio of nonzeros in DIA data structure	$ER\_DIA = \frac{NNZ}{Ndiags \times M}$	↑			
	ER_ELL	the ratio of nonzeros in ELL data structure	$ER\_ELL = \frac{NNZ}{max\_RD \times M}$		↑		
Power-Law	R	a factor of power-law distribution	$P(k) \sim k^{-R}$				[1, 4]

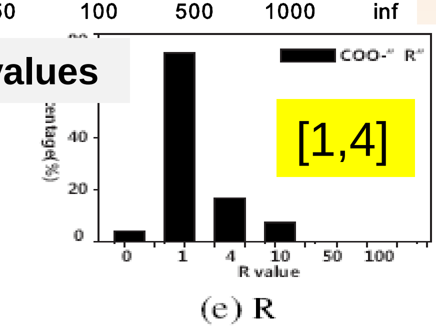
"√" shows the parameter is useful for all formats.

"↑/↓" indicates the larger (smaller) the parameter value is, the format shows more benefit.

Feature Extraction Process

- Divide the matrix set based on the optimal storage format, and measure SpMV performance on them
- Draw the value distribution graph to each parameter, and find the regulations.



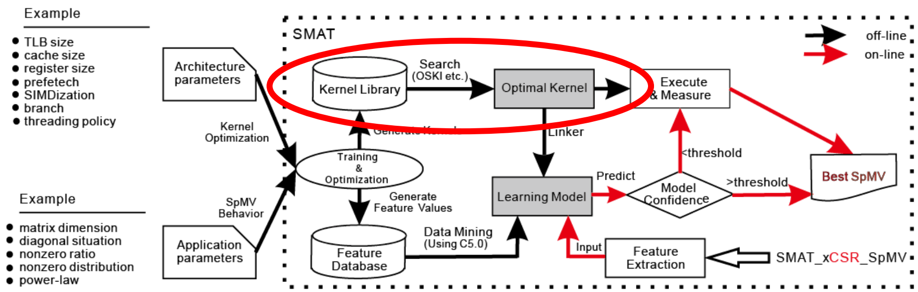


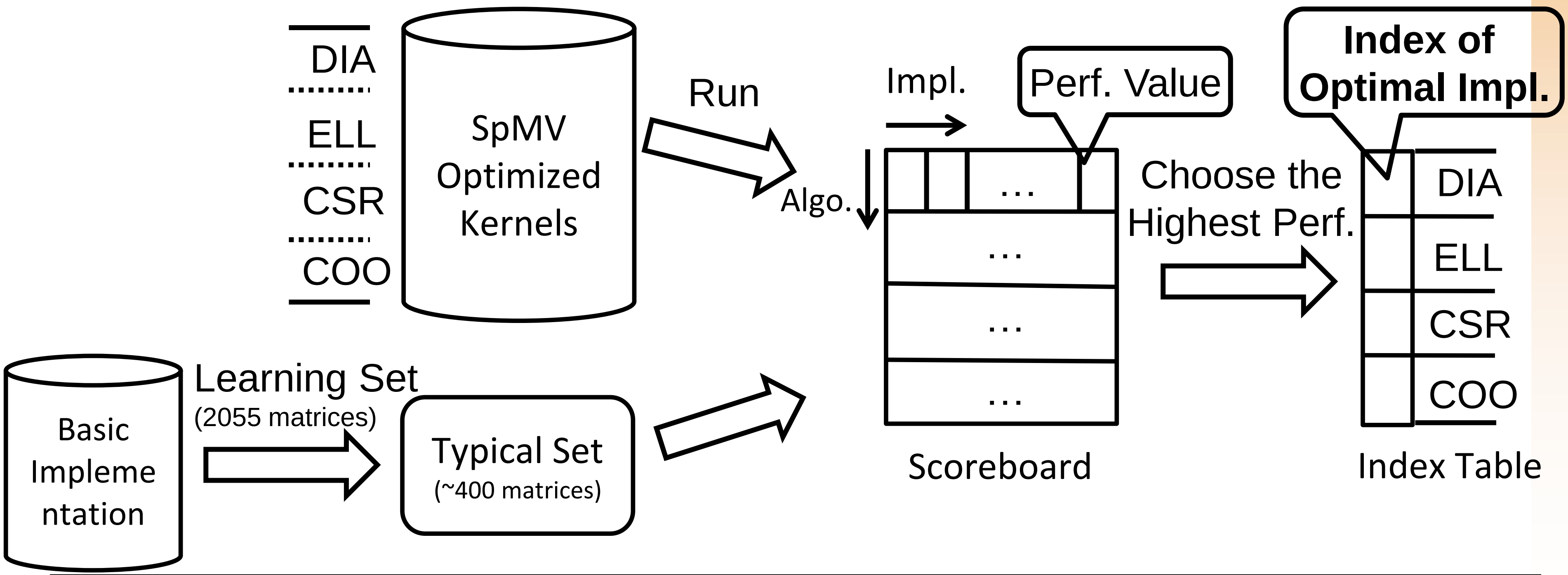
Matrix Partition	DIA_best	ELL_best	CSR_best	COO_best
Size of Sub-set	206	169	1496	507

SMAT—Choosing Optimal Implementation

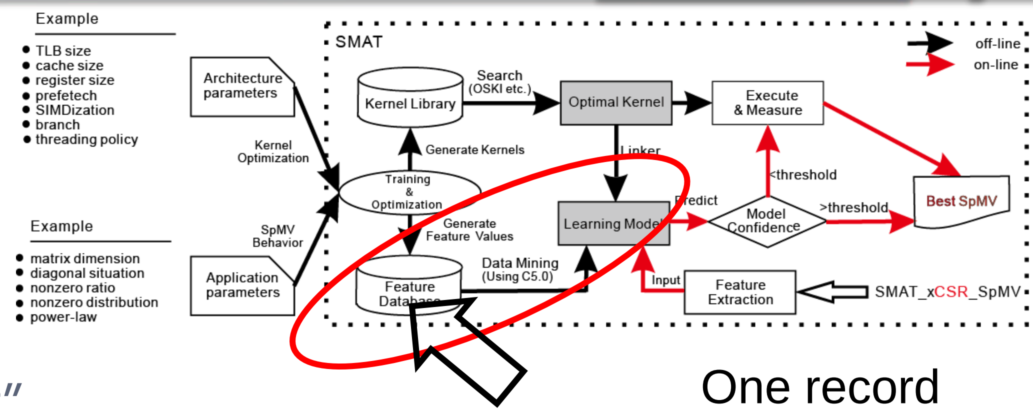
Scoreboard Strategy

- Choose typical matrix set to test each algo. and impl.
- Record the performance value on scoreboard
- The optimal impl. for each algo. are recorded in index table





SMAT—Data Mining



One record

{ M, N, Ndiags, Ntdiags_ratio, NNZ, max_RD, var_RD, ER_DIA, ER_ELL, R, Best_Format }

◆Belong to data mining problems

- Classification problem
- Target Attribute: "Best_format"

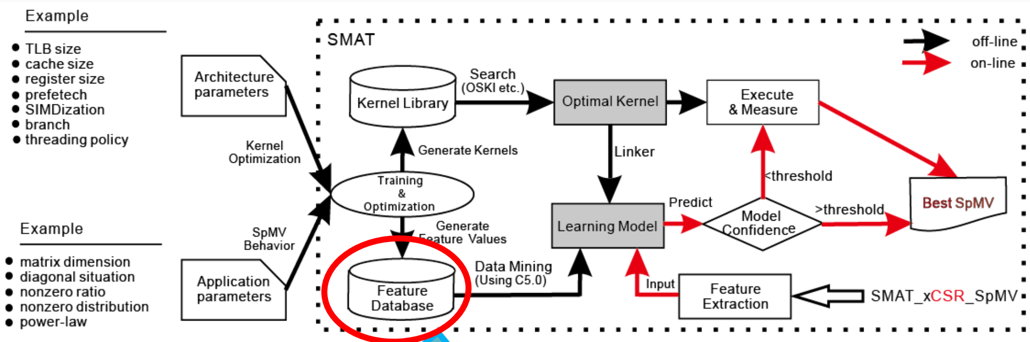
$$f(\overrightarrow{x_1}, \overrightarrow{x_2}, \cdots, \overrightarrow{x_n}, \overrightarrow{TH}) \rightarrow C_n(DIA, ELL, CSR, COO)$$

$\overrightarrow{x_i}$: value of each record, $\overrightarrow{\mathrm{TH}}$: threshold of each parameter,
$C_n(DIA, ELL, CSR, COO)$: one of the four formats. `

◆Build model

- Use ruleset to represent model
- Add rule confidence

SMAT—Data Mining



Data Mining Process

Training Data

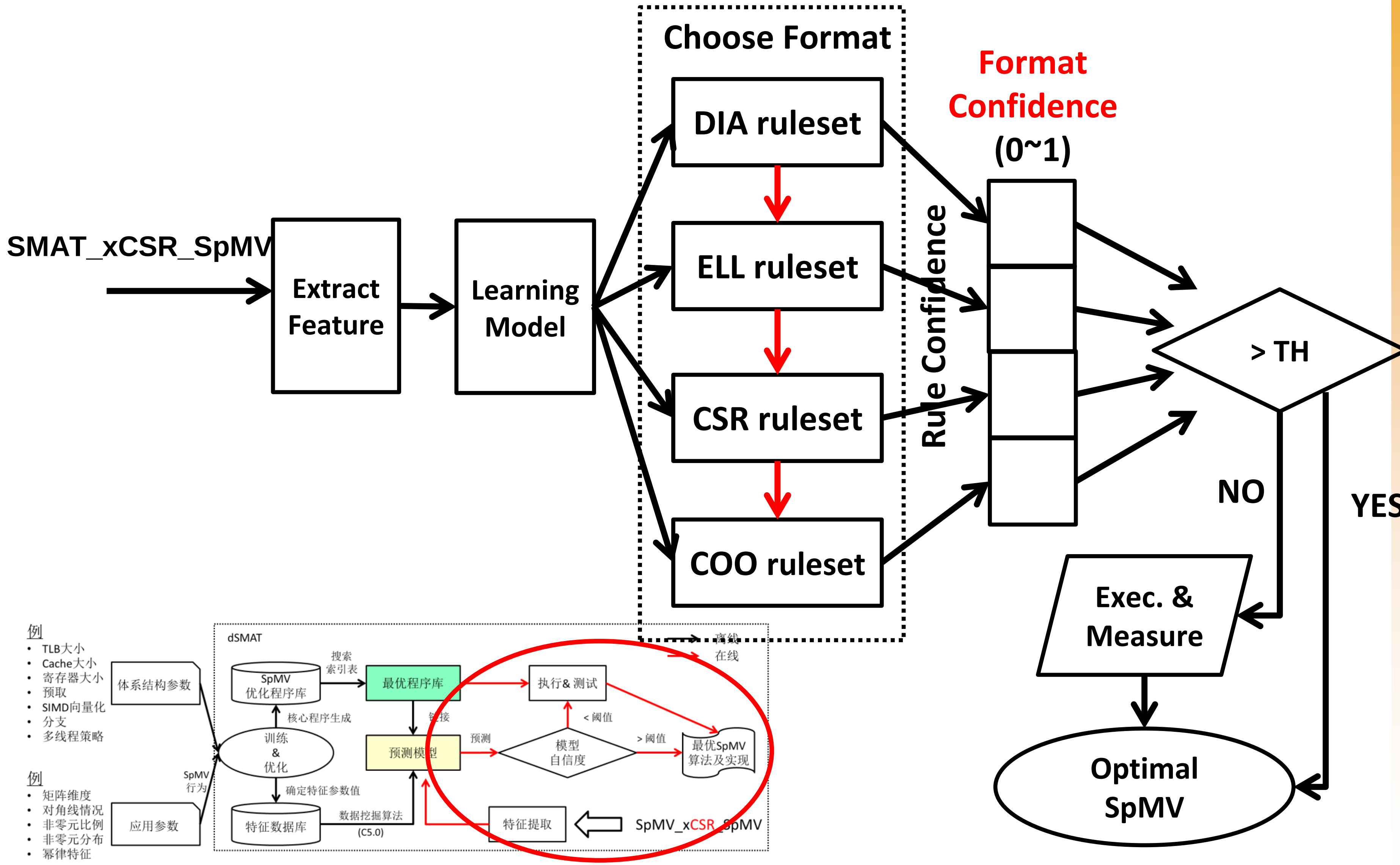
M	N	Ndiags	Ntdiags _ratio	...	Best _format
36476	36476	1328	0.001		ELL
23560	23560	33	0.515		DIA
12M	12M	1.8M	~0		COO
...	...	...	...		...

Classification Algorithm

Rules

IF Ndiags <= 39
AND ER_DIA > 0.3
AND ER_ELL <= 0.9
THEN
Best_format = DIA

SMAT -- Online Procedure



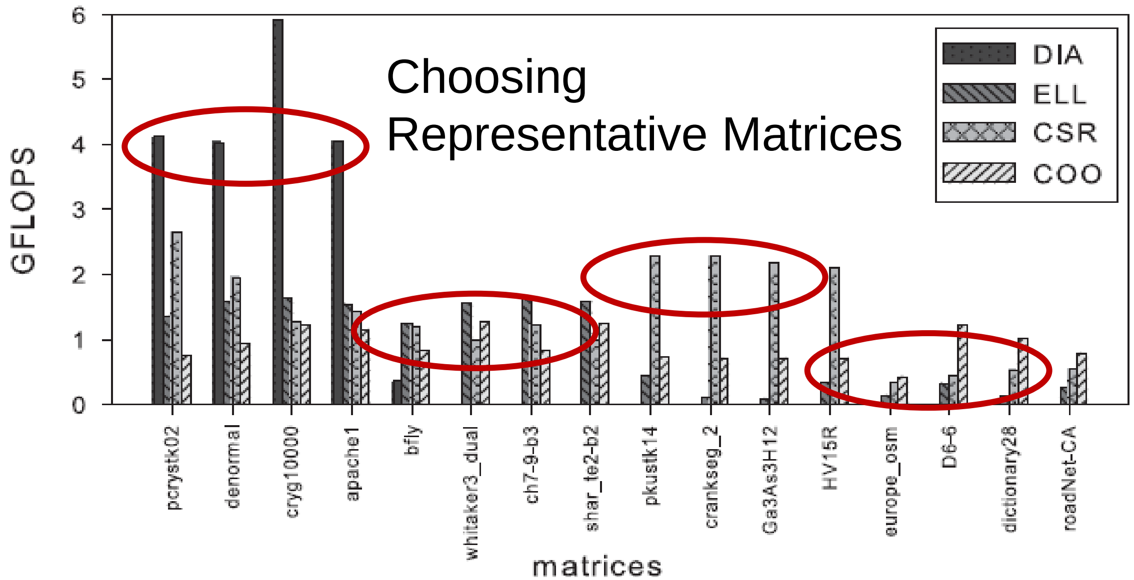
Platform and Matrix Set

◆ Platforms:

- Intel Xeon X5680
- AMD Opteron 6168

◆ Matrix set: The University of Florida Sparse Matrix Collection

- Learning Set: 2055
- Testing Set: 331, represented by 16 matrices



Representative Matrices

No.	Graph	Name	Dimensions	Nonzeros (NNZ / M)	Application area
1		pcrystk02	14K×14K	491K (35)	duplicate materials problem
2		d[illegible]	[illegible]	[illegible]	[illegible]-example
3		cryg10000	10K×10K	[illegible] (5)	materials problem
4		apache1	81K×81K	311K (4)	structural problem
5		bfly	49K×49K	98K (2)	undirected graph sequence
6		whita[illegible] _dual	[illegible]	[illegible]	[illegible]/3D problem
7		ch7-9-b3	106K×18K	[illegible] (4)	combinatorial problem
8		shar_te2-b2	200K×17K	601K (3)	combinatorial problem
9		pkustk14	152K×152K	15M (98)	structural problem
10		crankse[illegible]	[illegible]	[illegible]	[illegible]uctural problem
11		Ga3As3H12	61K×61K	[illegible] (97)	theoretical/quantum chemistry
12		HV15R	2M×2M	283M (140)	computational fluid dynamics
13		europe_osm	51M×51M	108M (2)	undirected graph
14		[illegible]	[illegible]	[illegible]	[illegible] problem
15		dictionary28	53K×53K	178K (3)	undirected graph
16		roadNet-CA	2M×2M	6M (3)	undirected graph

Diagonal Mat.

"Slim" Mat.

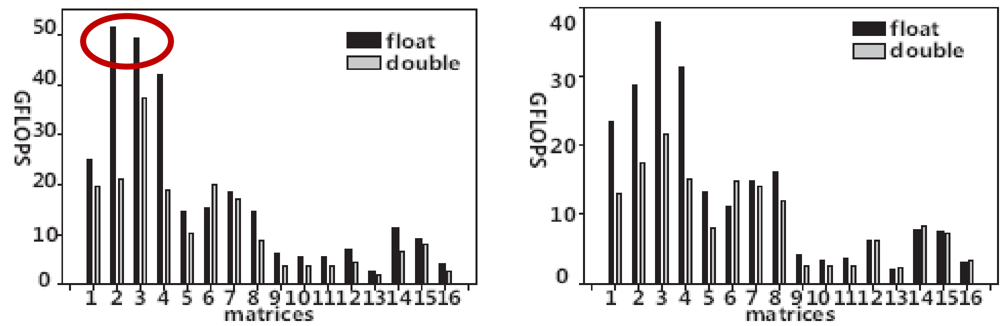
"Fat" Mat.

Power-Law Mat.

Performance

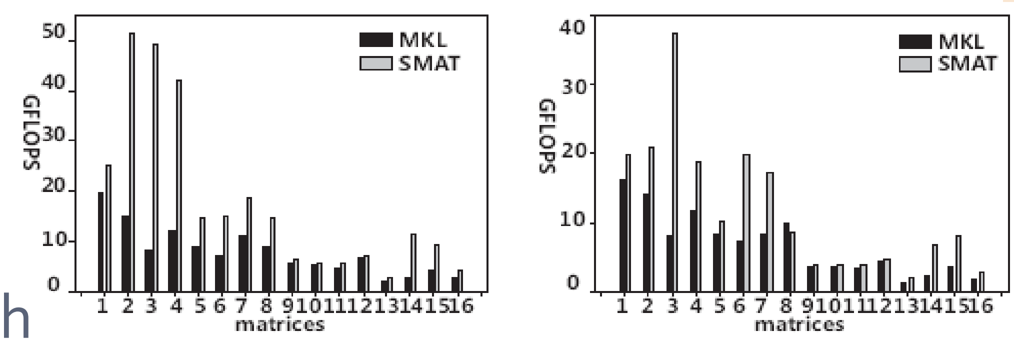
- **SMAT Auto-tuning**
- **Optimized SpMV kernels**
  - Assembling opt.
    - Loop unrolling
    - SIMDization
  - Multi-threading on task level
    - Allocate a sub-block to each thread
    - Independently choose the optimal algo. & impl. on each sub-block

Performance



(a) Intel (b) AMD

Compared with MKL



(a) float (b) double

~3X Speedup on average

Analysis on Accuracy and Overhead

- **Analyze the prediction procedure and accuracy on 16 representative matrices**
- **Overhead**
  - When the model predicts right, small overhead (about **2** CSR-SpMV)
  - Wrong predict, execute & measure mudule used, the overhead is more than **10** CSR-SpMV (OSKI:40+; clSpMV: ~10）
  - When a matrix is used hundreds of times in an iterative method, the overhead can be overlapped.

Matrix Number	Matrix Name	Model Prediction Format	Execution	SMAT Prediction Format	Actual Best Format	Model Accuracy	SMAT Overhead (times of CSR-SpMV)
1	pcrystk02	DIA	-	DIA	DIA	R	2.28
2	denormal	DIA	-	DIA	DIA	R	2.09
3	cryg10000	DIA	-	DIA	DIA	R	2.11
4	apache1	DIA	-	DIA	DIA	R	1.94
5	bfly	ELL	-	ELL	ELL	R	1.18
6	whitaker3_dual	ELL	-	ELL	ELL	R	4.89
7	ch7-9-b3	ELL	-	ELL	ELL	R	2.25
8	shar_te2-b2	ELL	-	ELL	ELL	R	2.24
9	pkustk14	$confidence < TH$	CSR+COO	CSR	CSR	W	16.39
10	crankseg_2	$confidence < TH$	CSR+COO	CSR	CSR	W	16.28
11	Ga3As3H12	$confidence < TH$	CSR+COO	CSR	CSR	W	16.2
12	HV15R	$confidence < TH$	CSR+COO	CSR	CSR	W	15.43
13	europe_osm	COO	-	COO	COO	R	2.3
14	D6-6	COO	-	COO	COO	R	5.79
15	dictionary28	COO	-	COO	COO	R	2.05
16	roadNet-CA	COO	-	COO	COO	R	2.38

"R" and "W" represent Right and Wrong prediction respectively.

Application Usability—Applied to Numerical Solver

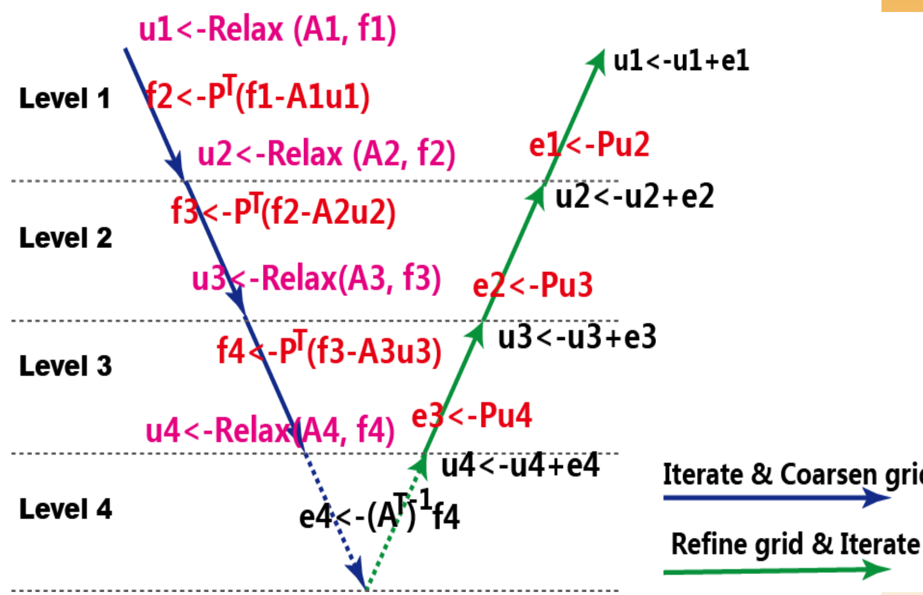
Algebraic Multi-grid Algorithm

- An iterative algo. to solve linear equations Au=f , where A is sparse matrix, u, f are dense vectors
- As a pre-conditioner applied in applications such as laser fusion and climate modeling

SpMV the critical operation of AMG, takes 90% execution time.

Coarsen	Rows	Hypre AMG	SMAT AMG	Speedup
cljp_7pt_50	125k	3034	2487	**1.22**
rugeL_9pt_500	250k	388	300	**1.29**

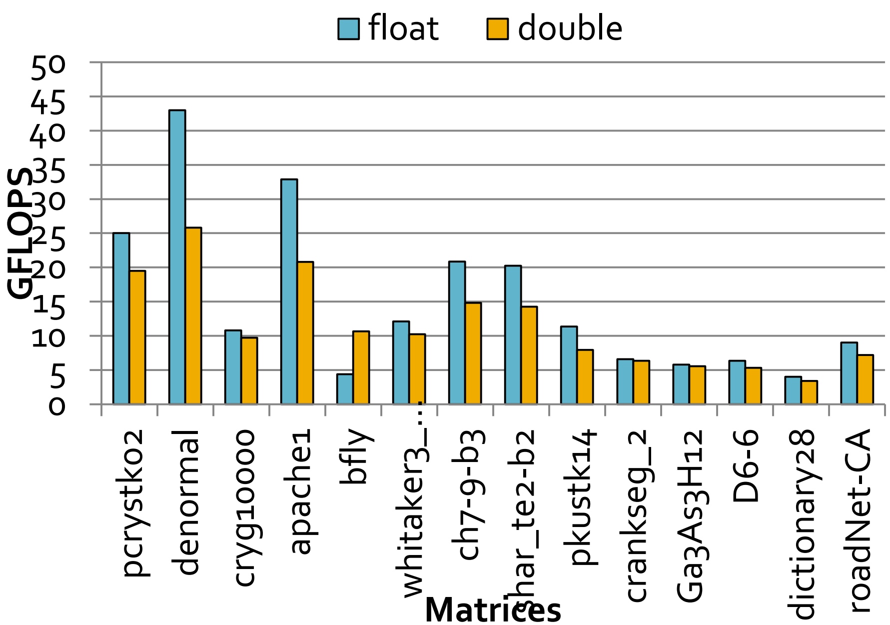
AMG Execution Process



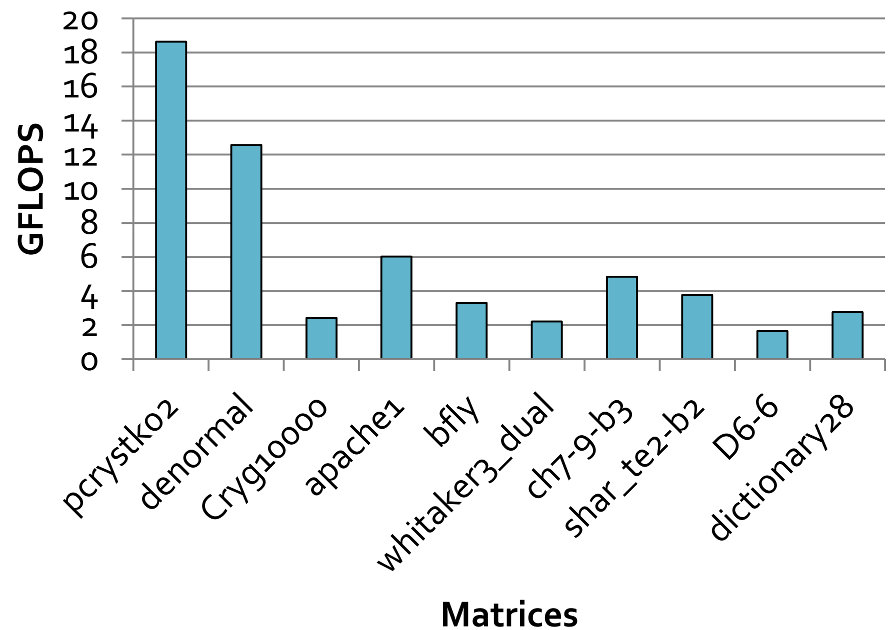
- Relax is relaxation algotithm, such as Jacobi、G-S iterative methods
- A, P are sparse matrices
- U, f, e are dense vectors

SMAT Many-core

Performance



NVIDIA K20



Xeon Phi

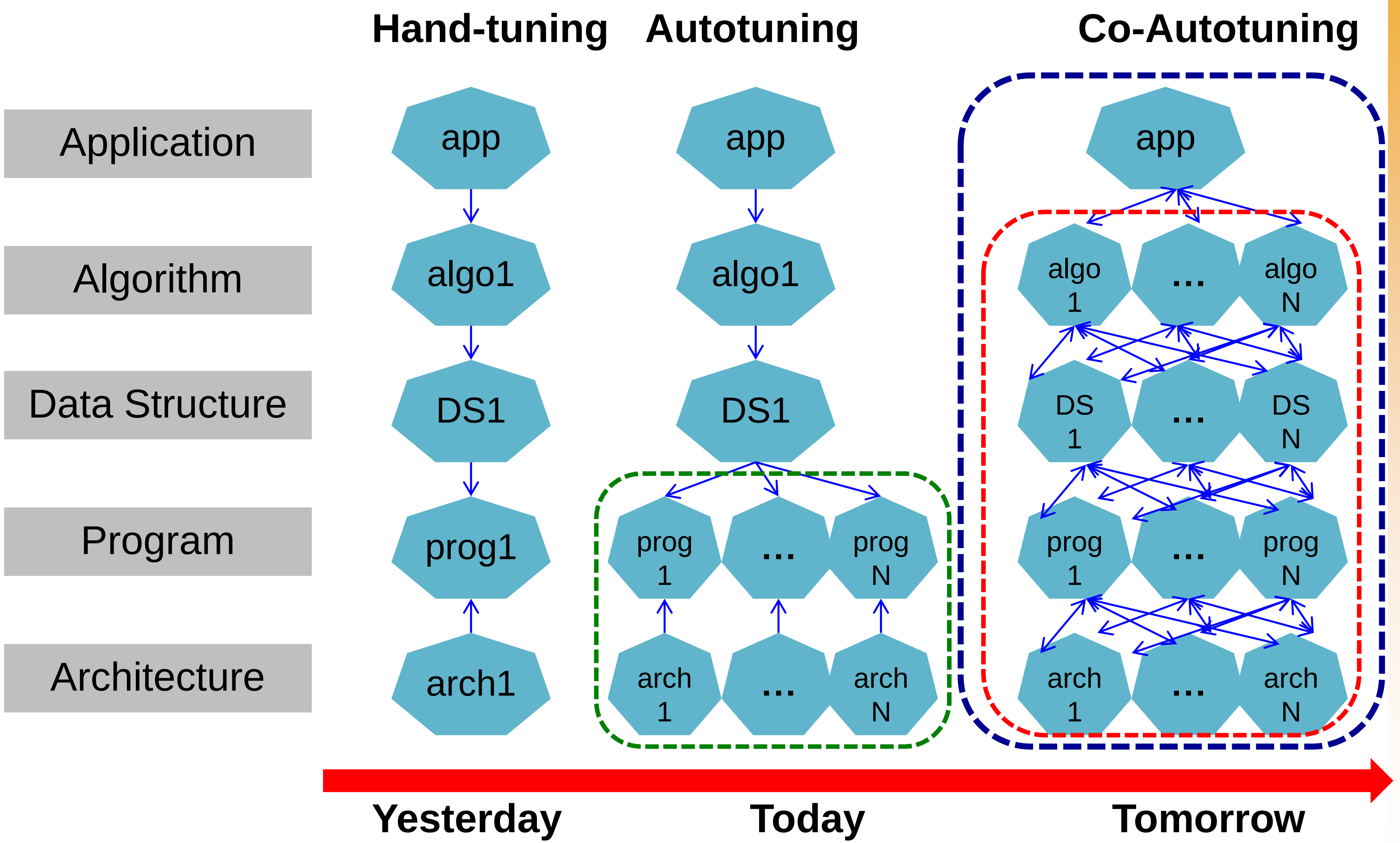
Accuracy

- For 289 testing matrices: 89%(single), 95%(double)

SMAT Summary

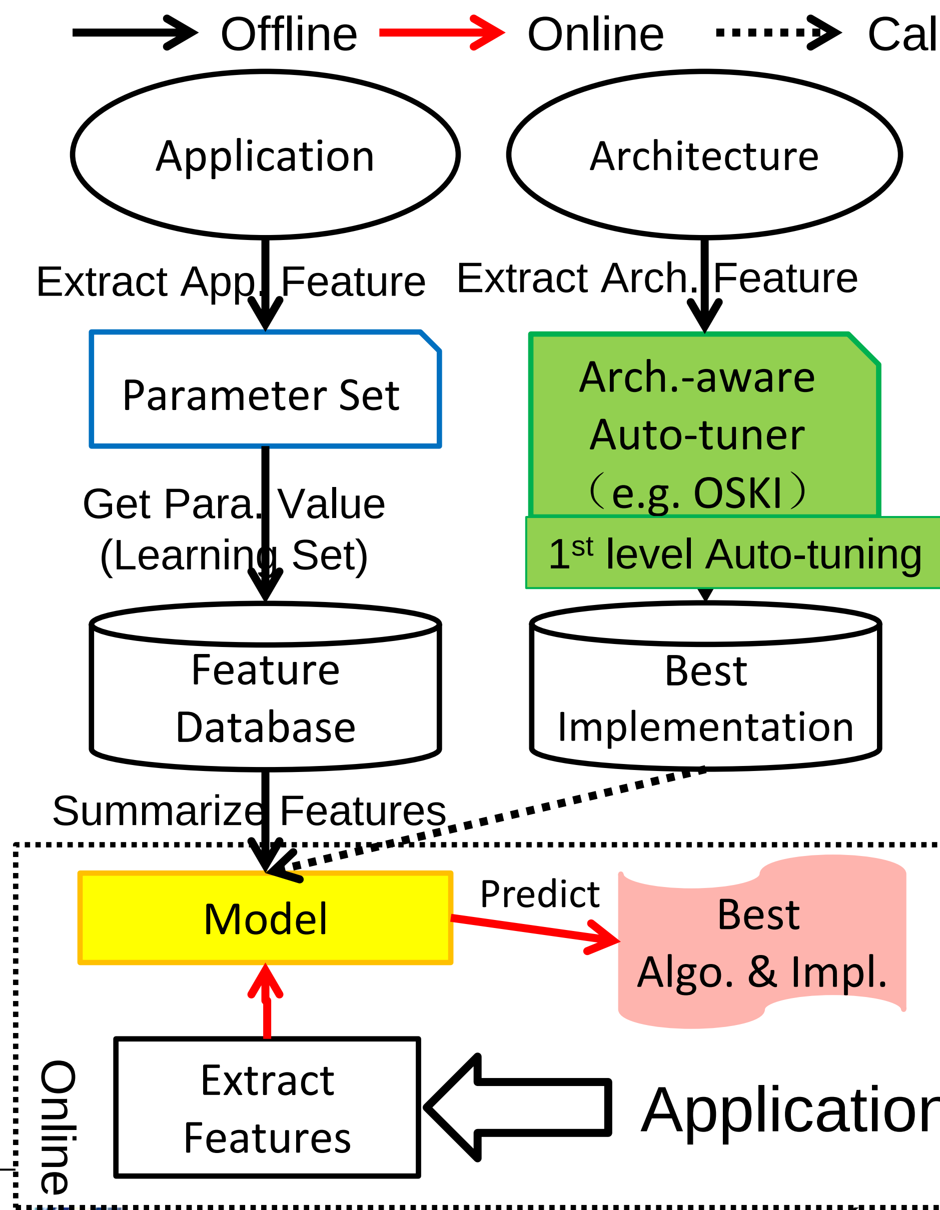
- **Develop auto-tuning method**
  - Design application-architecture aware SpMV auto-tuner.
  - Develop auto-tuning method to algorithm level
- **Introduce data mining to auto-tuning method**
  - Reinforce its usability and extensibility
- **API Easy-to-use**
  - Unify the interface
- **Increase SpMV performance using SMAT**
- **Apply SMAT to AMG, and extend it to many-core architecture**

High Performance Computational Software Development



Future Work

- ◆Extend storage formats
- ◆Combine other auto-tuners



Thank You !

Question?

SMAT: An Input Adaptive Auto-Tuner for Sparse Matrix-Vector Multiplication.

Jiajia Li, Guangming Tan, Mingyu Chen, Ninghui Sun

中国科学院计算技术研究所
Institute of Computing Technology,Chinese Academy of Sciences