RTune: A RocksDB Tuning System with Deep Genetic Algorithm

연세대학교 컴퓨터과학과 JIN HUIJUN

2022년 8월

과제명: IoT 환경을 위한 고성능 플래시 메모리 스토리지 기반 인메모리 분산 DBMS 연구개발

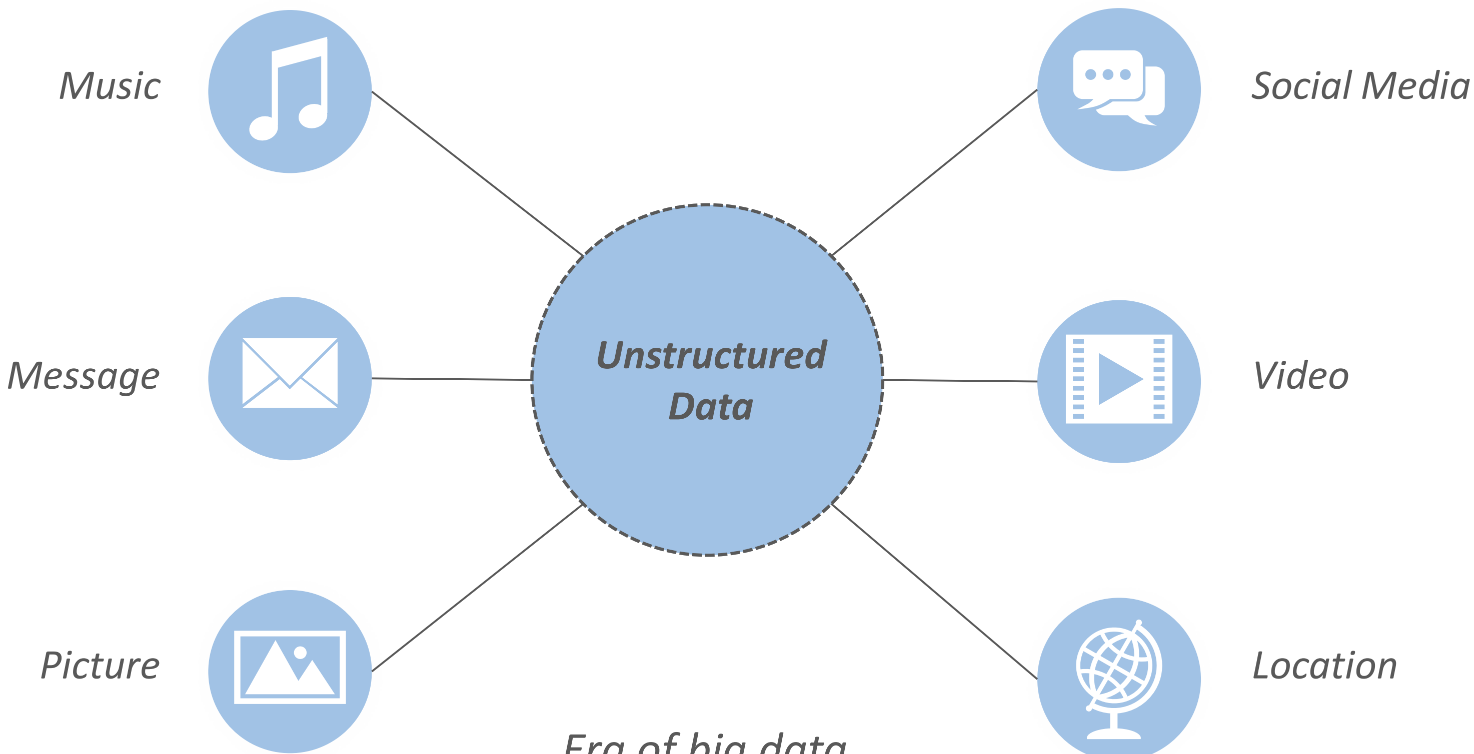
과제번호: 2017-0-00477

CONTENTS

YONSEI UNIVERSITY

1 Introduction

Introduction



Era of big data

Key-value databases have been proposed

Introduction

RocksDB

- *Disk-based key-value database*
- *Use Log-structured Merge-tree (**LSM-tree**)*

LSM-Tree

- ***Write amplification (WA)***
  - *Decrease data processing performance*
  - *Decline the lifespan*
- ***Space amplification (SA)***
  - *Increasing space usage*

Reduce WA and SA by tuning RocksDB knobs

- *Too many factors for performance tuning*
  - *Knobs, workload, hardware*

RTune : RocksDB tuning system

- ***Contributions:***
  - *Generated **RocksDB data repository**.*
  - ***New workload representation** for dimension reduction.*
  - *Created **combined workloads** that are as close to the target workload as possible.*
  - ***Novel score function** to train a DNN model.*
  - *Use a **genetic algorithm** with a trained **DNN** model to find the best solutions.*

YONSEI UNIVERSITY

2 Related Work

Related Work

Model	Optimization Target	Total Tuning time	Data Repository Dependency	Main Techniques	Target Database	Workload Mapping
OtterTune (2017)	Throughput Latency	60 min	O	Lasso repression GP	MySQL Postgres Vector	Euclidean distance
BestConfig (2017)	Throughput	-	X	DDS RBS	MySQL Cassandra Hive	X
CDBTune (2019)	Throughput Latency	Offline: 2.3 h Online: 25 min	X	DDPG	MySQL Postgres MongoDB	X
Multi-Task (2021)	IOPS	10 iterations	X	Multitask Clustering	RocksDB	X
RTune	TIME, RATE, WAF, SA	15 min	X	DNN, GA	RocksDB	Combined Workload

3

Background

1. LSM-Tree, WA and SA

2. Mahalanobis Distance

LSM-Tree, WA and SA

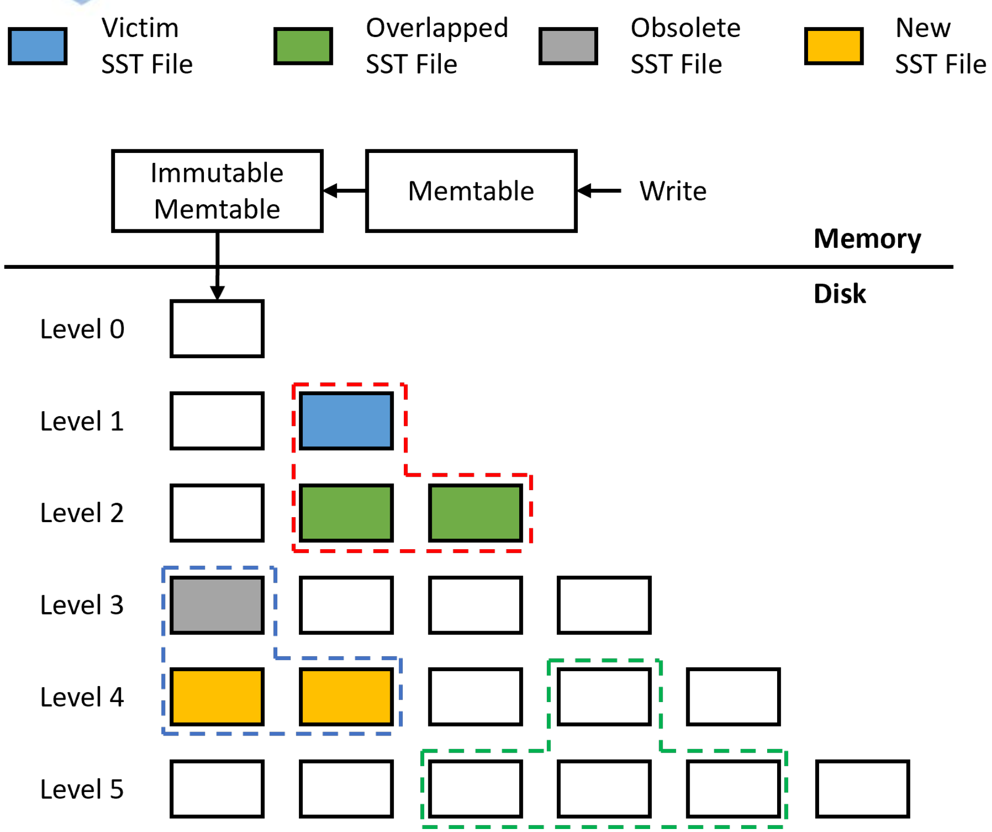
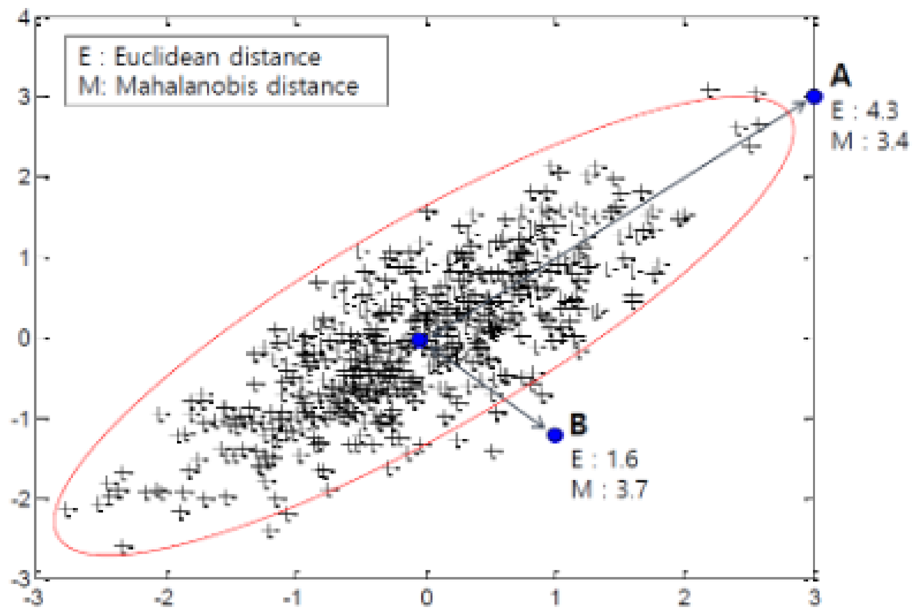


Figure 1. LSM-Tree and compaction

- ***Write Amplification (WA)***
  - *Additional write operations*
  - ***Overlapped SST File***
- ***Space Amplification (SA)***
  - *Additional space occupancy*
  - ***Obsolete SST File***
- ***Issues***
  - *Multi-threaded*
  - *Important to reduce **WA** and **SA***

Mahalanobis Distance



- ***Euclidean Distance***
  - $D_E(\vec{x}) = (\vec{x} - \vec{\mu})^{\mathrm{T}} (\vec{x} - \vec{\mu})$
- ***Mahalanobis Distance (MD)***
  - *Consider the* ***variance*** *between data*
  - $D_M(\vec{x}) = \sqrt{(\vec{x} - \vec{\mu})^{\mathrm{T}} \mathbf{S}^{-1} (\vec{x} - \vec{\mu})}$

Source: https://wikidocs.net/32057

4

Method

1. Architecture

2. Data Generation

3. Design

Architecture of proposed model

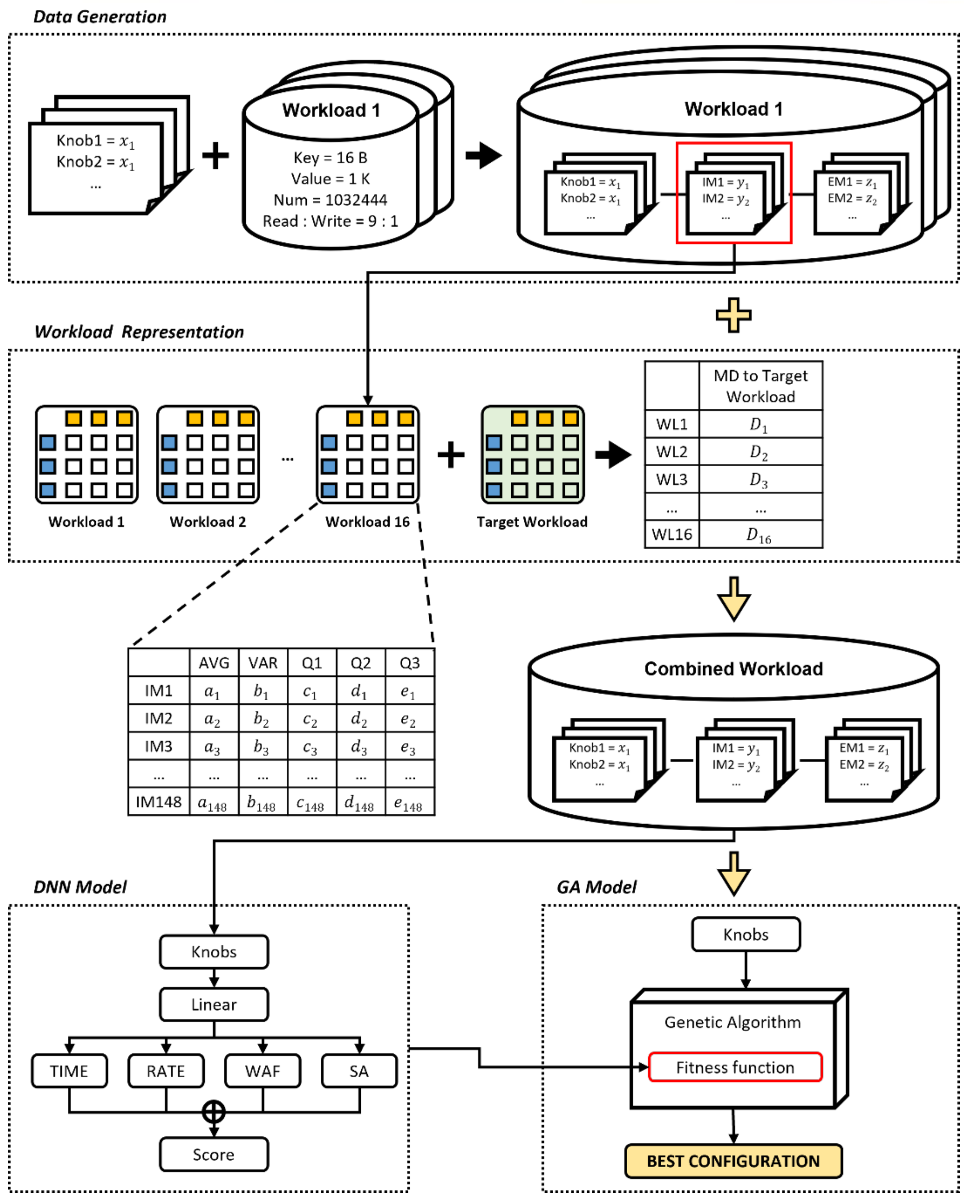


Figure 2. Overview of proposed model architecture

Data Generation

- ***DB_Bench*** *: RocksDB benchmarking tool.*
  - *Internal Metrics(IM) : Configuration, **workload**, I/O status, etc.*
  - *External Metrics(EM) : TIME, RATE, WAF, SA*

- ***Knobs***
  - *Selected **22 knobs** refer to official sites.*
  - *Generated 20,000 random **configurations** (set of knobs).*

- ***Workload***
  - ***16** workloads with size of **1 GB** each.*
  - *Different value sizes, # of entries, read-write ratio, update.*
  - *Key size : **16 B***

Data Generation

Table 1. Basic workloads.

Workload Index	Value Size (B), # of Entry	Read : Write	Update
0	1024, 1032444	9 : 1	-
1	1024, 1032444	1 : 1	-
2	1024, 1032444	1 : 9	-
3	1024, 1032444	-	TRUE
4	4096, 261124	9 : 1	-
5	4096, 261124	1 : 1	-
6	4096, 261124	1 : 9	-
7	4096, 261124	-	TRUE
8	16384, 65472	9 : 1	-
9	16384, 65472	1 : 1	-
10	16384, 65472	1 : 9	-
11	16384, 65472	-	TRUE
12	65536, 16380	9 : 1	-
13	65536, 16380	1 : 1	-
14	65536, 16380	1 : 9	-
15	65536, 16380	-	TRUE

Workload Representation

- *20,000* ***[Configuration – IM – EM]*** *pairs for each workload.*
- ***IM*** *include information for a workload.*
- *Represent a workload by IM.*

- ***Disadvantages*** *:*
  - ***Huge*** *size of table.*
  - ***Expensive*** *to proceed with various calculations.*

Table 2. Original workload representation.

Configuration #	1	2	…	20000
IM 1	12965	13040	…	14586
IM 2	1239	837	…	297
…	…	…	…	…
IM n	44	63	78	208

Workload Representation

- *Use 5 statistics **[Average, Variance, 1st Quartile, 2nd Quartile, 3rd Quartile]** of 20,000 data of each internal metrics.*

- ***Advantages** :*
  - ***Dimension reduction***
  - ***Easy** to proceed with various calculations.*

Table 3. New workload representation.

	Average	Variance	1st Quartile	2nd Quartile	3rd Quartile
IM 1	13544	55615	10513	13448	15798
IM 2	834	2315	564	912	1132
…	…	…	…	…	…
IM n	80	1213	68	81	121

Combined Workload

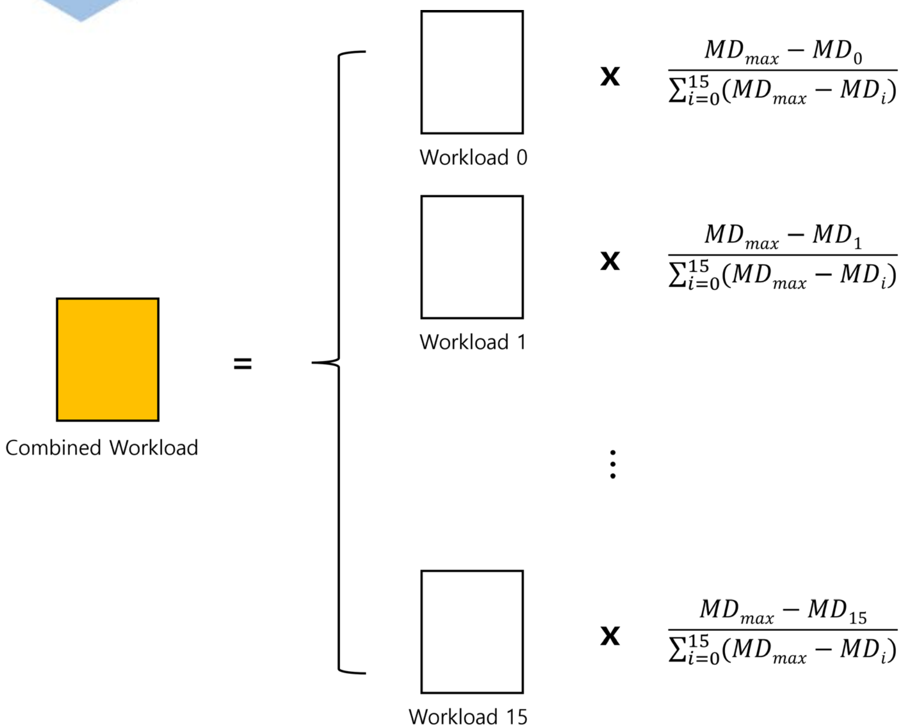


Figure 3. Combined workload calculation process

- *Distance → **similarity***
- *Calculate the **proportion** of each basic workload data to be included in CW*
- *20,000 **[Configuration – IM – EM]** pairs in CW*

Deep Neural Network Model

- *Train **DNN** model with CW*
  - *Input : **Configurations***
  - *Output : Prediction for **4 EM**.*

- ***Score function***
  - $Score = \alpha_1 \frac{TIME_D}{TIME_P} + \alpha_2 \frac{RATE_P}{RATE_D} + \alpha_3 \frac{WAF_D}{WAF_P} + \alpha_4 \frac{SA_D}{SA_P}$
  - $\alpha_1 = \alpha_2 = \alpha_3 = \alpha_4 = 0.25$

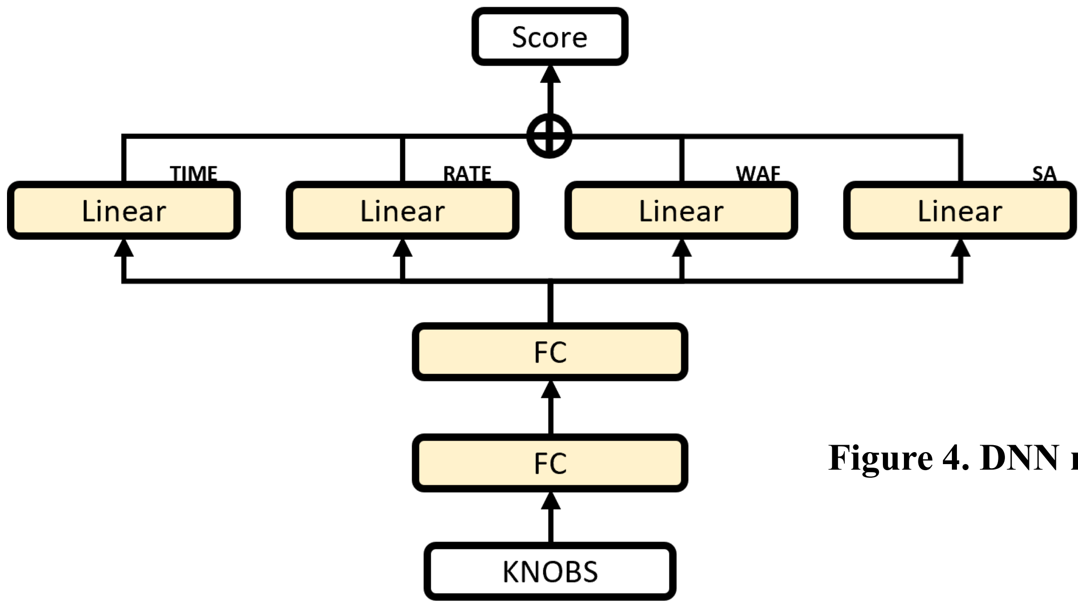


Figure 4. DNN model structure

Deep Neural Network Model

Table 4. Hyperparameters of DNN model.

Optimizer	Adamw
Learning Rate	0.0002
Epoch	300
Loss Function	MSE
X Scaler	MinMaxScaler
Y Scaler	StandardScaler
Layer	(22, 64, 16, 1)
Activation Function	ReLU

Genetic Algorithm

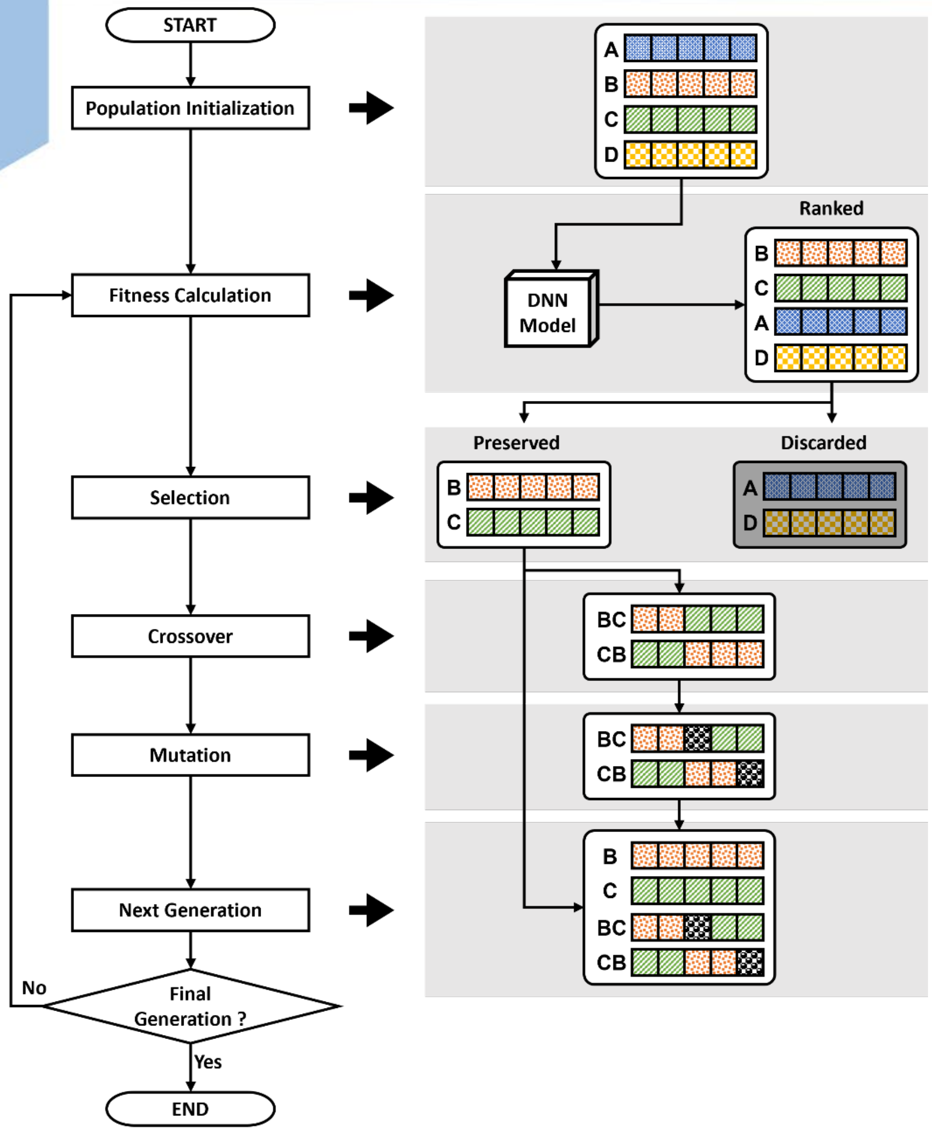


Figure 5. Genetic Algorithm

- *DNN model* → ***Fitness calculation***
- *Mutant knob* → ***New knob***
- *Highest fitness score* → ***Optimal solution***

Genetic Algorithm

Table 5. Hyperparameters of GA model.

Mutation Ratio	0.4
Crossover Ratio	0.5
Population Size	128
Generation	1000
Selection Algorithm	Rank Selection
Selection Size	64

5

Evaluation

1. Experimental Setup

2. Results

Target Workload Information

- *Generated **6 target workloads***
  - ***16** workloads with size of **1 GB** each.*
  - *Different value sizes, # of entries, read-write ratio, update.*
  - *Key size : **16 B***

- *Generate **20** data pair for target workloads.*

Table 6. Target workloads.

Workload Index	Value Size (B), # of Entry	Read : Write	Update
16	8192, 130816	7 : 3	-
17	8192, 130816	3 : 7	-
18	8192, 130816	-	True
19	32768, 32752	7 : 3	-
20	32768, 32752	3 : 7	-
21	32768, 32752	-	True

External Metrics

- *External Metrics:* ***TIME, RATE, WAF, SA***
  - *TIME (s):* ***Total execution time***
    - *Time internal from the start of the data recording to the end.*
  - *RATE (MB/s):* ***Data processing rate***
    - *The number of operations processed by RocksDB per second.*
  - *WAF :* ***WA factor***
    - *Ratio of* ***physical data size*** *and* ***logical data size*** *written to the storage.*
    - $WAF = \frac{Physical\ data\ size}{Logical\ data\ size}$
  - *SA (MB):* ***Space amplification***
    - *The size of the data recorded in the actual* ***LSM-Tree****.*

Results

- *Apply **geometric mean** to the EM.*
- $Score = \sqrt[4]{\frac{TIME_D}{TIME_A} \times \frac{RATE_A}{RATE_D} \times \frac{WAF_D}{WAF_A} \times \frac{SA_D}{SA_A}}$
- *Performance of default setting is described as a **red line pointing to 1**.*

Overall Comparison

- *Overall comparison among **default settings**, **Facebook** recommended configuration, database administrator (**DBA**), and **RTune**.*

- *Best performance among the **5 target workloads**.*
- *Slightly lower than DBA in the **17th** workload.*

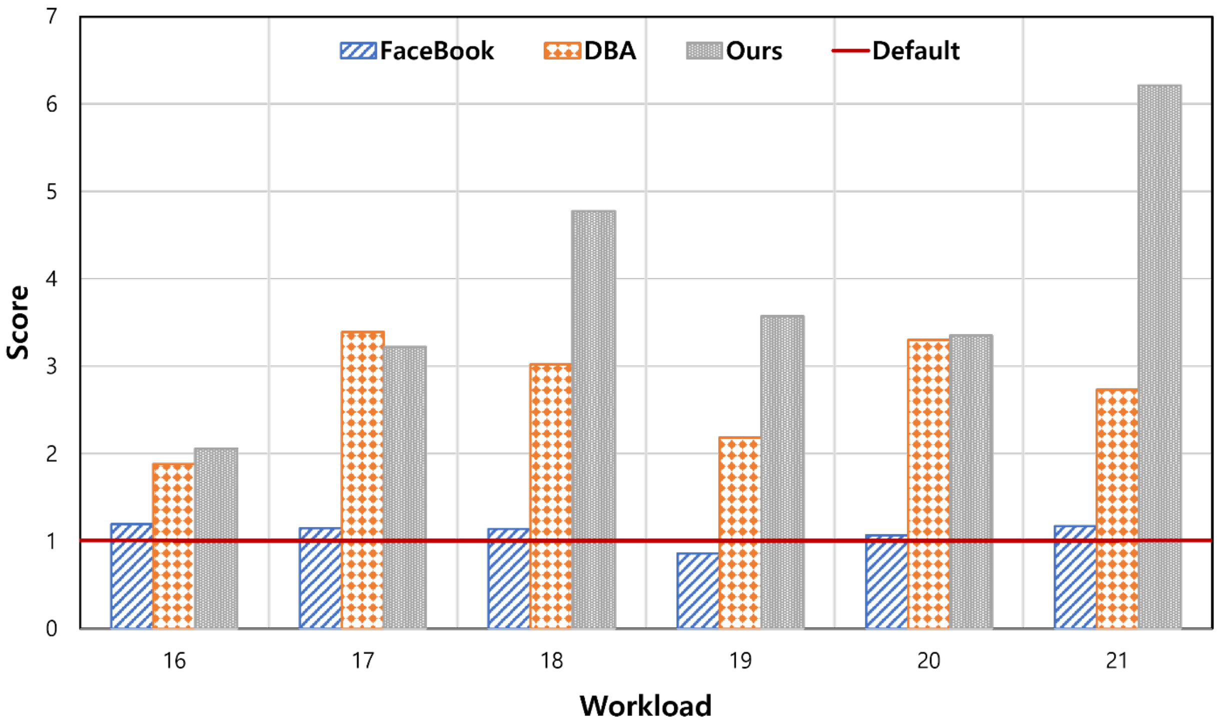


Figure 6. Overall performance comparison

Combined Workload & Pruning Internal Metrics

- ***"Not Combined"*** *: Use the* ***closest*** *workload as the CW.*
- ***"IM Clustering"*** *: Use pruned IM by k-means clustering.*

- *Best performance in* ***all target workloads****.*
- *Better to describe a workload using* ***CW*** *and* ***full IM****.*

Figure 7. Workload combining and internal metrics pruning comparison

Number of Knobs

- *Difference in the **number of knobs**.*
- *Pruning knobs with **3, 5, and 7 knobs** with a random forest.*

- *Best performance in **all target workloads**.*
- *Top 7 model achieved good performance in workload 16, 17, and 19, but it is **not stable**.*

Figure 8. Comparison of number of knobs

Weight Comparison

- *Comparison of using **4 different weight pairs** to the score function.*

- *Best performance in **all target workloads**.*
- *The rest of 4 models hard to reach the default setting.*

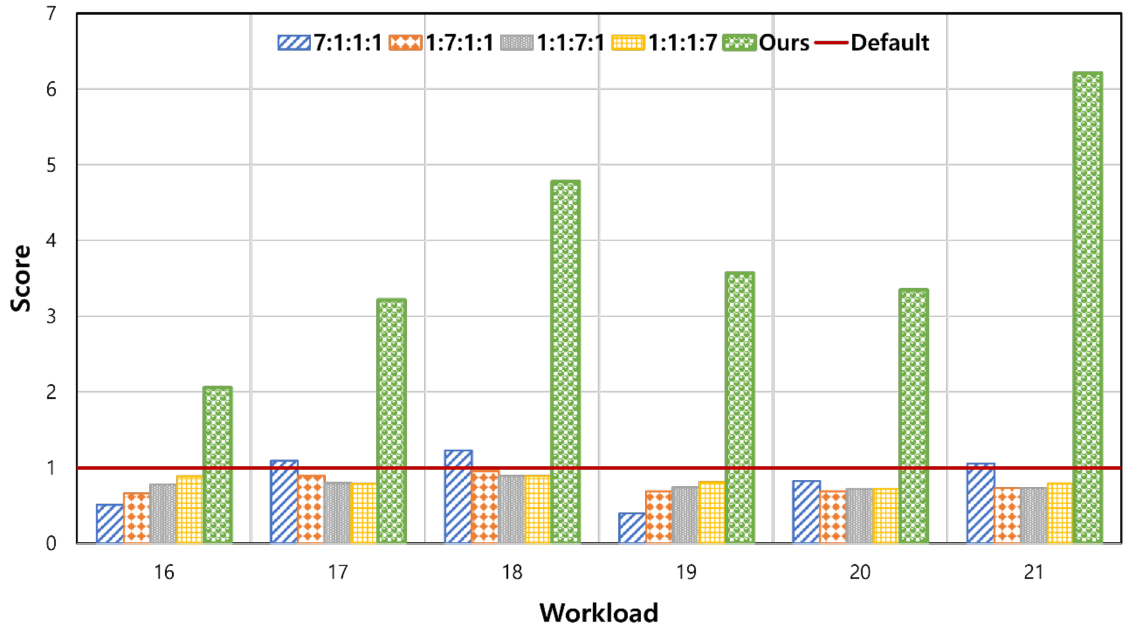
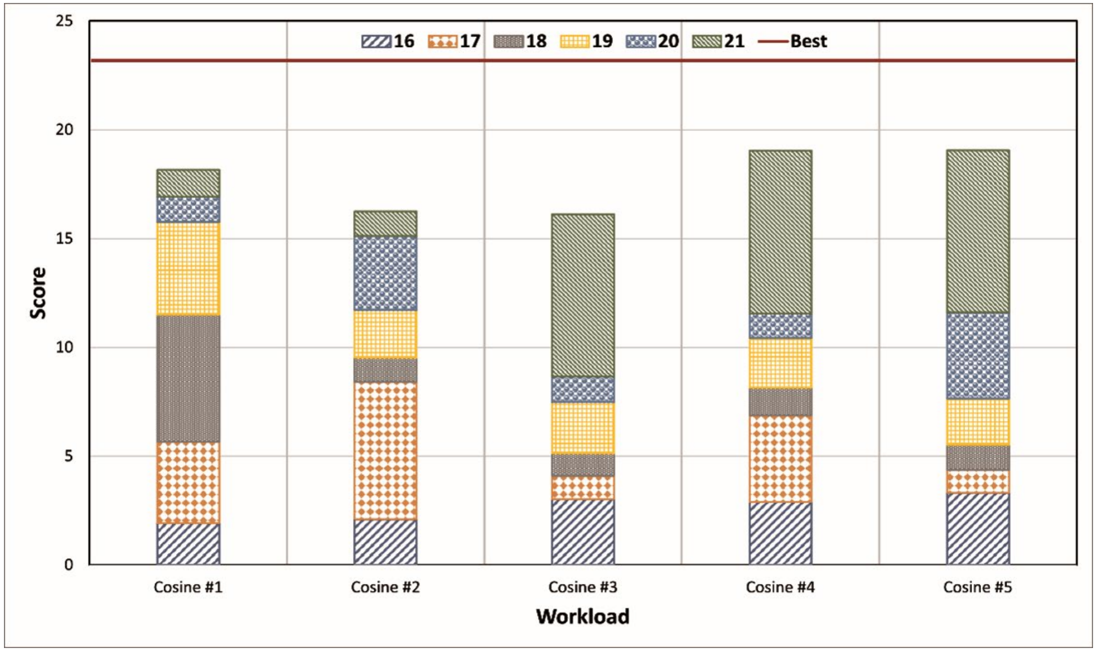


Figure 9. Different weight comparison

Cosine similarity & Mahalanobis distance

- *Comparison of cosine similarity and Mahalanobis distance.*

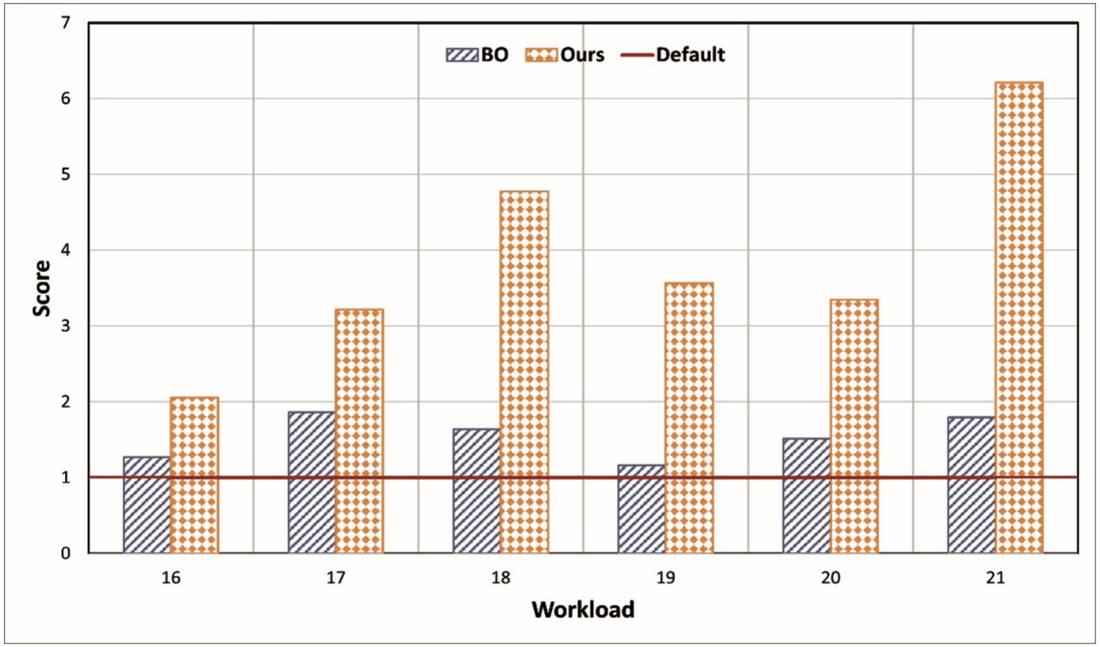
- *The sum of the performance was good but not the best.*
- *Performance is not stable especially in the case of Cosine #3.*



Bayesian optimization comparison

- *Comparison of Bayesian optimization with GA.*

- *BO cannot overperform our model.*
- *The score barely exceeds the default line in workload 19.*



6 Conclusion

Conclusion and Future Works

- ***Conclusion***
  - *Generated **RocksDB data repository**.*
  - *Applied **MD** and **new workload representation** to create **CW**.*
  - ***Novel score function** to train DNN model with 4 EM **simultaneously**.*
  - *Use **GA** with **DNN** model to find the **optimal solutions**.*
  - *Proved the optimal solutions yields the **best performance** through comparative experiments.*

THANK YOU

Contact: jinhuijun@yonsei.ac.kr