High-Dimensional Bayesian Optimization with Multi-Task Learning for RocksDB

동국대학교 통계학과 조성운

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

과제번호: 2017-0-00477

목차

Abstract

Maximizing the **throughput of RocksDB IO operations** by **auto-tuning then parameters of var ying ranges**

High-Dimensional Problem

- multi-task modeling
- dimensionality reduction through clustering

Introduction

Auto-tuning method 필요한 이유

A high-dimensional optimization space is a common phenomenon in general-purpose systems as they have many **parameters** and **objectives**

파라미터와 측정 결과값을 모두 확인하여 결정하는 것은 어렵다.

Introduction

Bayesian Optimization(BO)

- A sample efficent tuner, has received considerable attention in recent years due to its versatility and efficiency
- The framework, first builds a system model and then uses the model to find an optimal config-uraiton

Introduction

BO’s drawback is its inability to handle heigh-dimensional spaces

- Curse of dimensionality
- A computationally expensive operation in the surrogate model

➡ Optimizing over **multiple targets** to increase learning mileage per training sample

➡ Decomposes into a subset of system components that influence the primary target.

Background

- RocksDB is a key-value store based on LevelDB that provides efficient concurrent **reads and writes**
- RocksDB stores new data in a Log-Structured Merge-Tree(LSM-Tree) format in memory
    - Once the memtable is full, RocksDB flushes it sequentially into a Sorted Sequenc e Tree(SST) file on disk
    - Rocks DB still processes concurrent writes during the flushing process
- SST organizes the data in levels starting from **level-0 to level-n**
    - When a level is full, it performs a housekeeping operaion(compaction) that merges and re moves tombstones
    - During compaction, RocksDB stalls new writes, leading to a reduced IO throughput and inc reased latency

Background

Bayesian Optimization

BO is a sample efficient optimization framework that solves the problem of **block-box functio n** optimization

Surrogate Model
$\boldsymbol{f}$

- contains a belief of the system and updates at every now observation

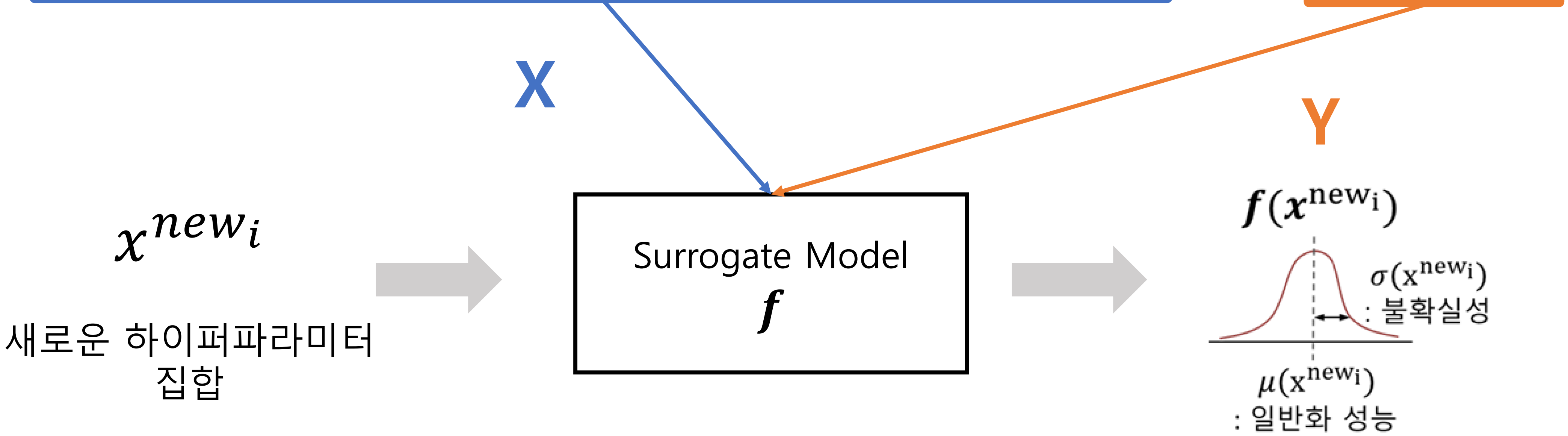
argmax Acquisition function

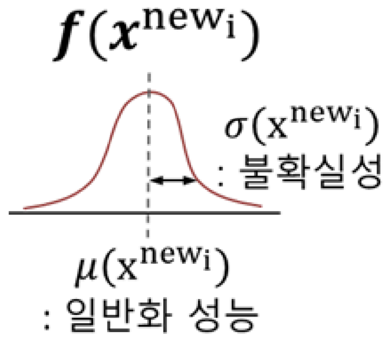
- performs numerical optimization operations over the model to find configurations to test next on the objective function

Background

Learning Rate	# of iterations	# of hidden layers	...	Mini batch size
0.001	100	3	...	128
0.001	100	3	...	128

Performance
100
200





argmax Acquisition function

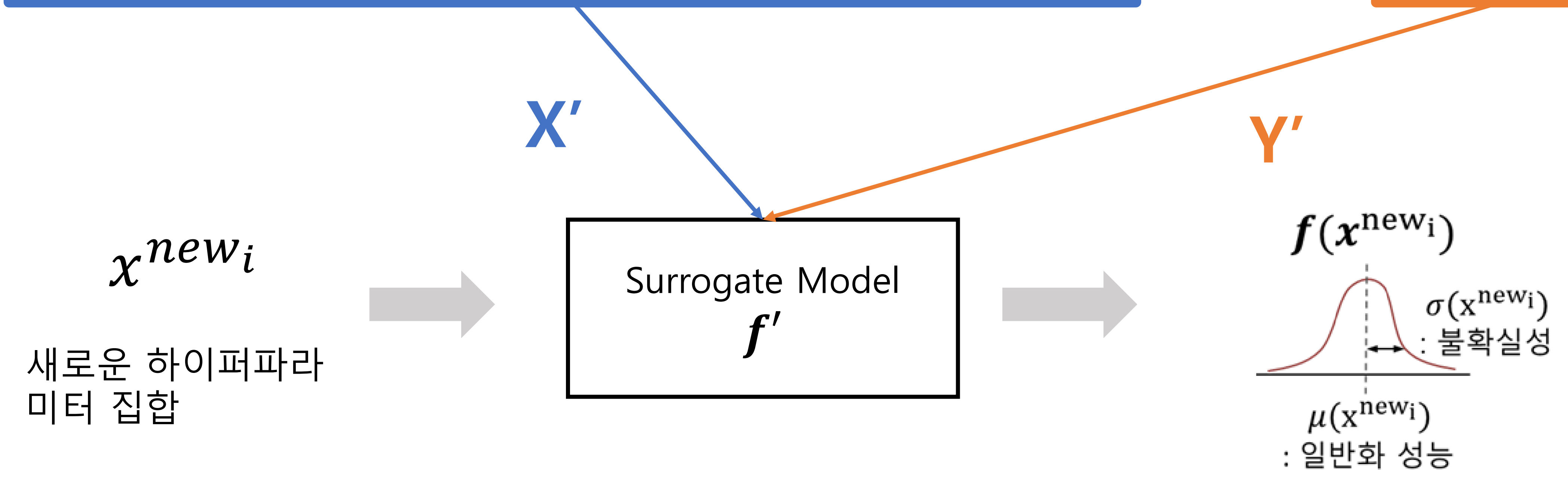
x^*

Learning Rate	# of iterations	# of hidden layers	…	Mini batch size
0.001	100	3	…	128
0.001	100	3	…	128
		X*		

Performance
100
200

Learning Rate	# of iterations	# of hidden layers	…	Mini batch size
0.001	100	3	…	128
0.001	100	3	…	128

Performance
100
200



Structured multi-task optimization

overveiw

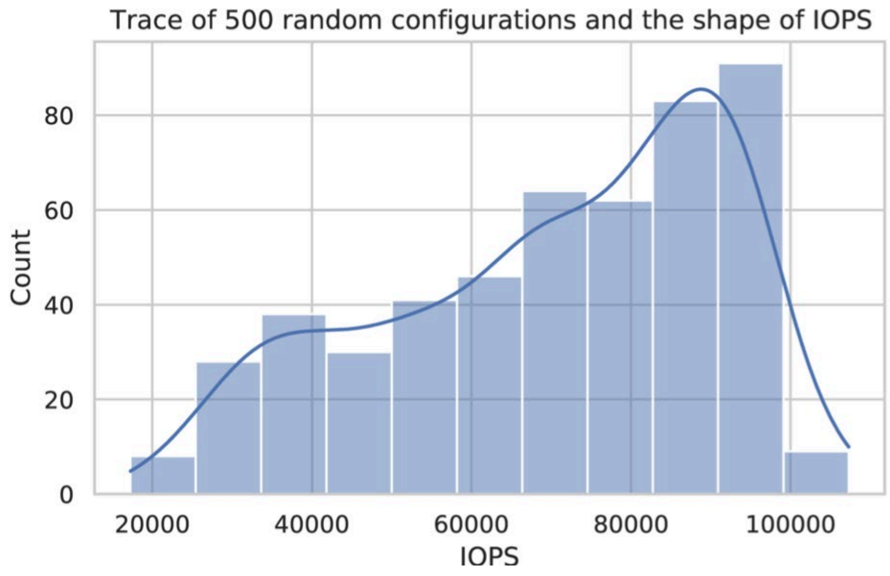
- Used multi-task learning to capture the intersection between system components
- Learn more from every sample, reducing the observation needed for convergence
- Reduces the dimensions through a manual grouping of parameters to speed up the convergence

Problem space and assumptions

The fundamental assumption in using GP

- the function is differentiable at every point
- the modeling space is a multivariate Gaussian distribution

We performed **500 independent experime nts** where we randomly sampled the model ing space(then parameters)



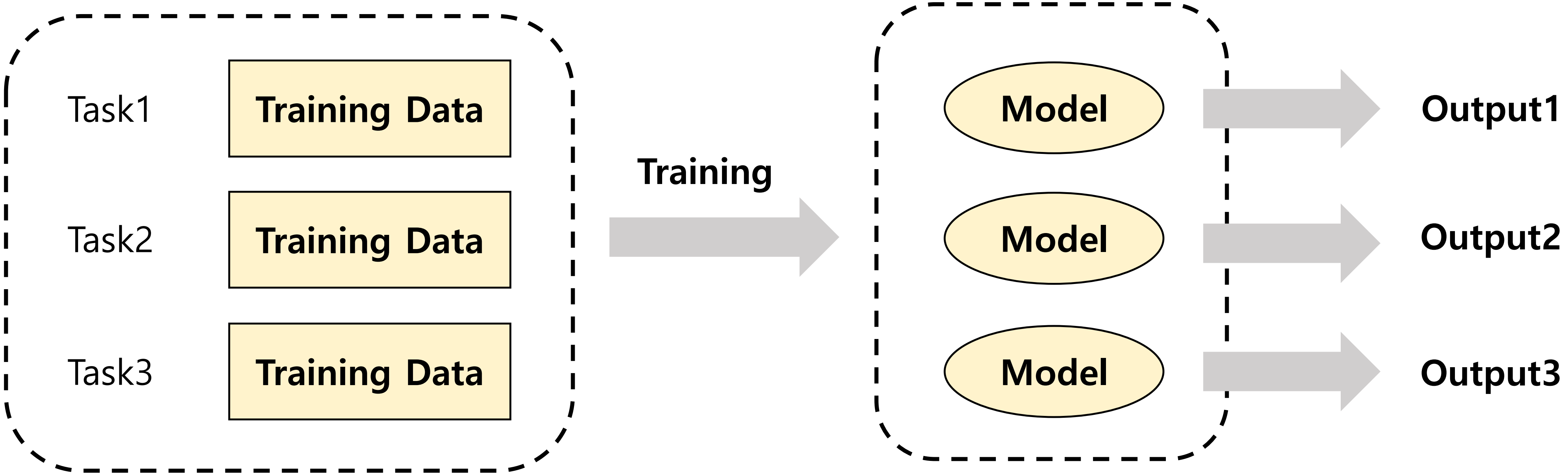
Multi-task learning

Chose three additional objectives based on our understanding of Rocks DB architecture

- **WriteAmplification** : the ratio of bytes written to storage to the bytes written to the backend.
- **ReadBlockGetP99** : The 99^{th} percentile latency to read a block of data.
- **Level0Tolevel1P99** : The 99^{th} percentile time it takes to compact blocks stored in level0 to level1.

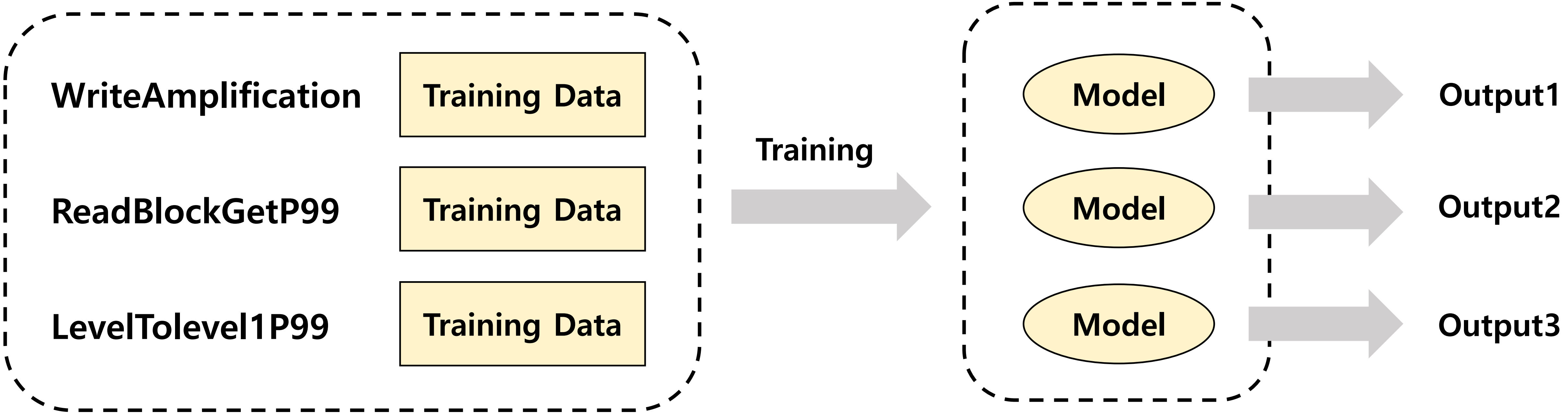
Multi-task learning

A machine learning method based on shared representations, which uses task multit asking for learning



Multi-task learning

A machine learning method based on shared representations, which uses task multit asking for learning



Multi-Task learning in GP

Intrinsic Coregionalization Model (ICM kernel)

$$\boldsymbol{k}\big((\boldsymbol{x},\boldsymbol{m}),(\boldsymbol{x}',\boldsymbol{m}')\big) = \boldsymbol{k_x}(\boldsymbol{x},\boldsymbol{x}')\boldsymbol{k_T}(\boldsymbol{m},\boldsymbol{m}')$$

$\boldsymbol{k_x}(\boldsymbol{x}, \boldsymbol{x}')$ → The parameter covariance kernel

$\boldsymbol{k_T}(\boldsymbol{m},\boldsymbol{m}')$ → The task similarity kernel

ICM challenges

- ICM method provides a neat trick to get more mileage out of the few sample
- A standard GP inference is $O(Tn^3)$, duplicating the data to the number of tasks scales this to

Curse of dimensionality
문제 해결되지 않음!!

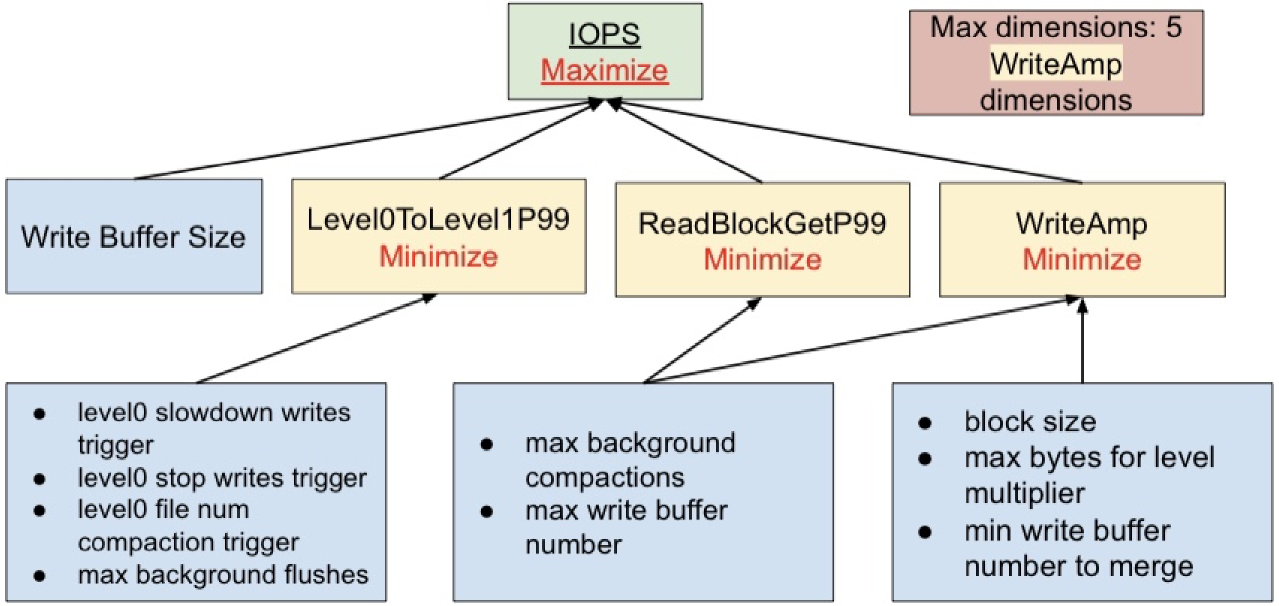
Decomposability through clustering

The decomposability refers to the smallest unit of obseravable RocksDB's performace metric and a corre sponding set of parameters in this context

Using the 500 random configurations thrace, we calculated **the correlations betw een IOPS and the 517 observable metric** from RocksDB and **the correlations be tween them to the parameters**

30개 군집으로 clustering!!

Decomposability through clustering



각 architecture와 관련있는 cluster와 parameter들을 입력값으로
Bayesian Optimization 진행

Decomposability through clustering

Table 1. RocksDB parameters and their impact. All reported parameters are discrete ordinal variables.

Parameter	Range	Default
max_background_compactions	$[1, 2^8]$	1
max_background_flushes	[110]	1
write_buffer_size	$[1, 15 * 10^7]$	2^{26}
max_write_buffer_number	$[1, 2^7]$	2
min_write_buffer_number_to_merge	$[1, 2^5]$	1
max_bytes_for_level_multiplier	[5, 15]	10
block_size	$[1, 5 * 10^5]$	2^{12}
level0_file_num_compaction_trigger	$[1, 2^8]$	2^2
level0_slowdown_writes_trigger	$[1, 2^{10}]$	0
level0_stop_writes_trigger	$[1, 2^{10}]$	36

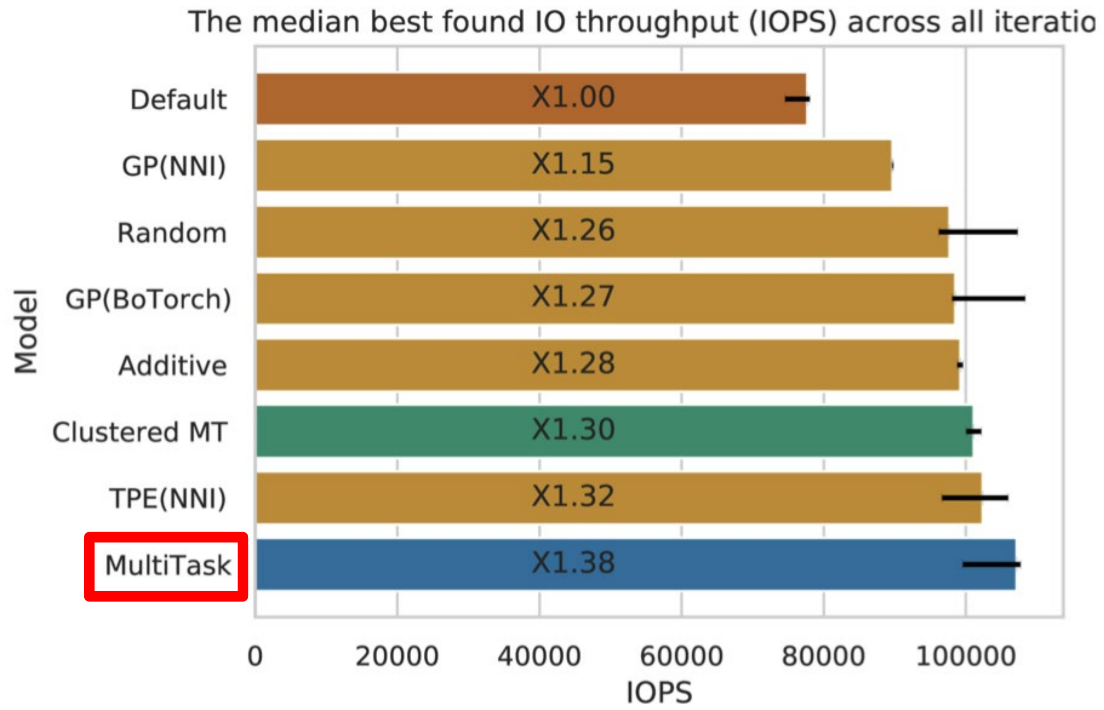
Evaluation

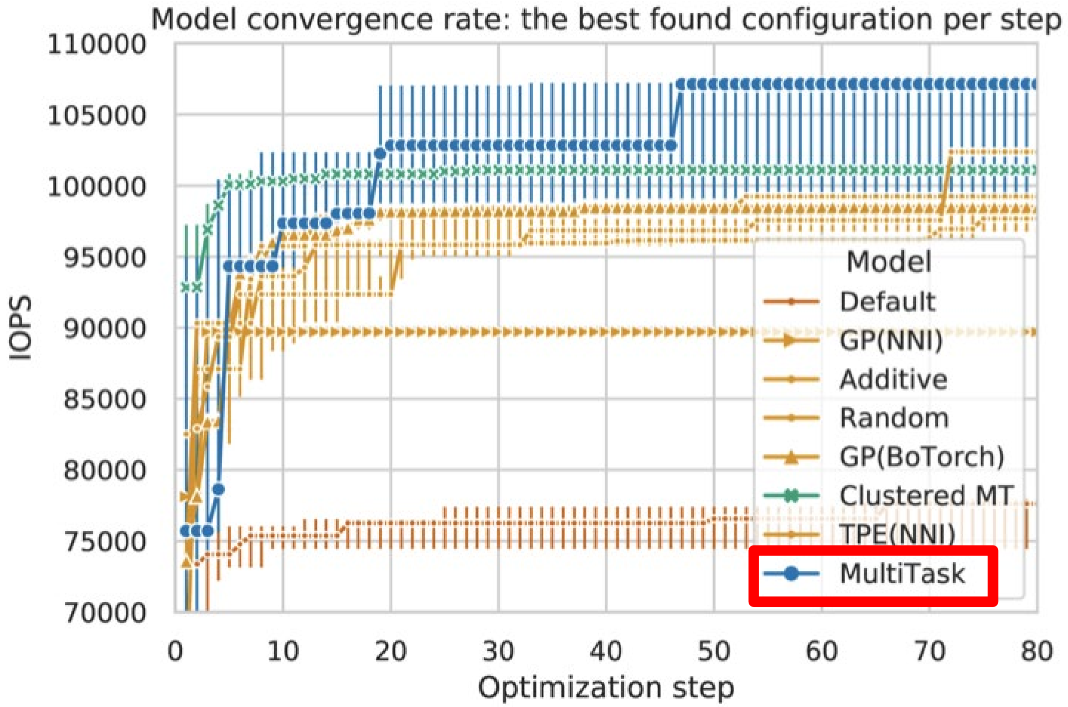
- used RocksDB's benchmark tool **db_bench**
- Set a budget of 100 optimization steps

Table 2. Alternative surrogate models as baselines. The background has a short introduction to these methods 2.3.

Method	**Use case**
TPE [4]	Handles discrete parameters.
GP (NNI) [29]	Standard $O(n^3)$ implementation.
Random [5]	Low effective search dimensions.
Additive kernel [15]	Low-dimensions decomposability.
Default	RocksDB v6.17 default settings.
BoTorch [3]	Efficient GPyTorch $O(n^2)$ GP.

Evaluation





IO Troughput이 가장 높다

가장 빠르게 증가한다

Conclusion

- The tuner exploits alternative observable metrics and structural decomposability to converge f aster and reduce the dimensional space
- Utilize multi-task learning to provide an accessible mechanism for expressing structure in the model
- Tuner outperformed the default configuration by x1.35 in 10 iterations, compared to the other state-of-the-art methods requiring 60 iterations