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

Sami Alabed Eiko Yoneki

Department of Computer Science and Technology
University of Cambridge

General purpose key-value store with complex parameters

- RocksDB an embedded versatile key-value store.

General purpose key-value store with complex parameters

- RocksDB an embedded versatile key-value store.
- Many use cases: embedded store for reliable messaging in stream processors, low latency, or high throughput systems.

General purpose key-value store with complex parameters

- RocksDB an embedded versatile key-value store.
- Many use cases: embedded store for reliable messaging in stream processors, low latency, or high throughput systems.
- It achieves that by exposing a large number of configurations to the end-user.

General purpose key-value store with complex parameters

- RocksDB an embedded versatile key-value store.
- Many use cases: embedded store for reliable messaging in stream processors, low latency, or high throughput systems.
- It achieves that by exposing a large number of configurations to the end-user.
- Parameters are influenced by the hardware specifications, service objectives, and workload characteristics.

General purpose key-value store with complex parameters

- RocksDB an embedded versatile key-value store.
- Many use cases: embedded store for reliable messaging in stream processors, low latency, or high throughput systems.
- It achieves that by exposing a large number of configurations to the end-user.
- Parameters are influenced by the hardware specifications, service objectives, and workload characteristics.
- Finding optimal parameters might require many experiments.

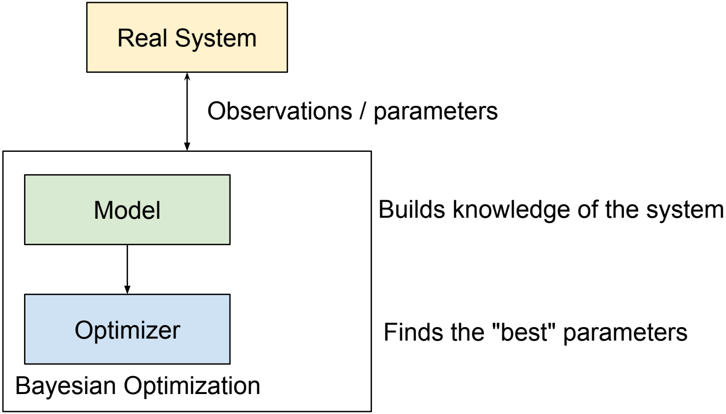
General purpose key-value store with complex parameters

- RocksDB an embedded versatile key-value store.
- Many use cases: embedded store for reliable messaging in stream processors, low latency, or high throughput systems.
- It achieves that by exposing a large number of configurations to the end-user.
- Parameters are influenced by the hardware specifications, service objectives, and workload characteristics.
- Finding optimal parameters might require many experiments.

Research question

How do we select the optimal configurations as quickly as possible?

Results with tuning using Bayesian Optimization

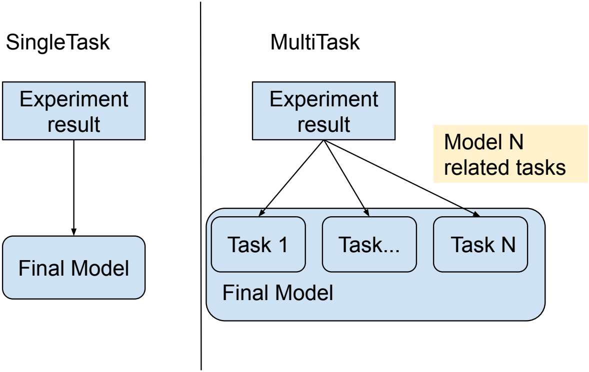


BO reduces the number of communications with the real system

Opportunities to optimize the Bayesian optimization loop

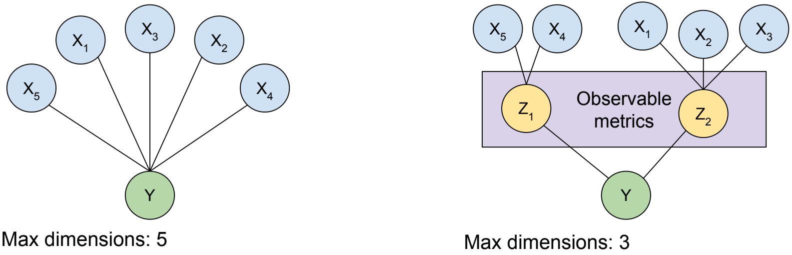
- Navigating the *curse of dimensionality* by providing a wider context with multi-task learning.
- Use the expert knowledge to reduce the dimensions of the model through parameters clustering.

Optimizing other tasks provide more data per iteration



Multi-task modeling learns the wider context of the system.

Decomposability reduces dimensions



Structural modeling reduces the dimensional space of the problem. On the right the maximum effective dimension is three for the the metric with largest number of parameters:$p(z_2|x_1, x_2, x_3)$.

Modeling with Gaussian Process

- Popular model used in BO.
- Powerful non-parametric model that captures the relation between variables in a multivariate normal distribution.
- Defined as $GP(\mu, K)$ where μ is the mean function, and K is a covariance kernel.
- Still suffers from the *curse of dimensionality* in high dimensional settings.

Multi-task optimization

- Mitigating the *curse of dimensionality* by providing context about the system interactions.
- Replace the kernel function in the GP with an Intrinsic Coregionalization Model (ICM).
- Has two parts: parameter covariance kernel, and a task similarity kernel.

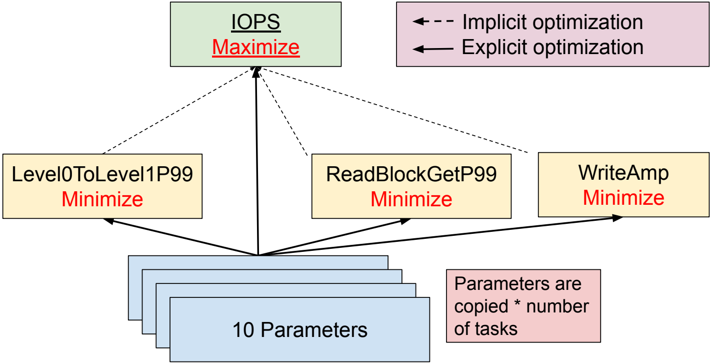
Multi-task optimization

- Mitigating the *curse of dimensionality* by providing context about the system interactions.
- Replace the kernel function in the GP with an Intrinsic Coregionalization Model (ICM).
- Has two parts: parameter covariance kernel, and a task similarity kernel.
- Each task is a multivariate Gaussian distribution optimized later in the BO loop.

Multi-task optimization

- Mitigating the *curse of dimensionality* by providing context about the system interactions.
- Replace the kernel function in the GP with an Intrinsic Coregionalization Model (ICM).
- Has two parts: parameter covariance kernel, and a task similarity kernel.
- Each task is a multivariate Gaussian distribution optimized later in the BO loop.
- Creates a copy of every parameter and a task observation.

Multi-task in RocksDB IOPS optimization



Optimizing the complimentary tasks provide implicit optimization to the primary goal.

Multi-task scales poorly

- Multi-task provides more information per training sample.
- Downside: it duplicates the training samples in the process.
- A standard GP inference scales linearly with number of tasks $O(Tn^3)$.

Decomposability through clustering

- Computer systems have a natural decomposability where certain parameters only influence a subset of the metrics.

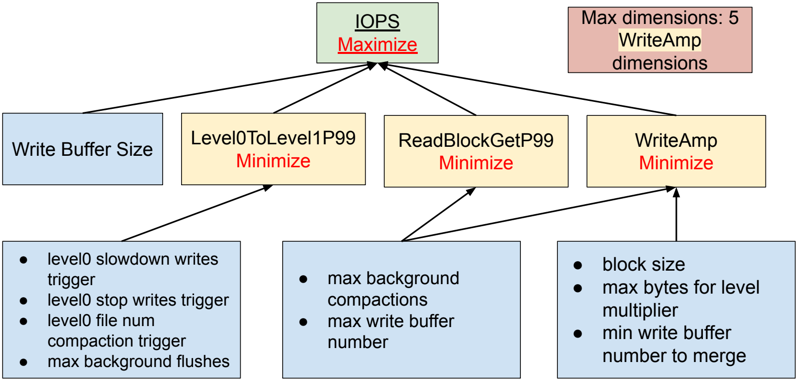
Decomposability through clustering

- Computer systems have a natural decomposability where certain parameters only influence a subset of the metrics.
- Decomposing the parameter space reduces the max dimensions by assigning a subset of parameters to each task.

Decomposability through clustering

- Computer systems have a natural decomposability where certain parameters only influence a subset of the metrics.
- Decomposing the parameter space reduces the max dimensions by assigning a subset of parameters to each task.
- The assignment can be found through a combination of expert knowledge and unsupervised learning methods.

RocksDB structured multi-task learning



Reducing the maximum effective dimension. The assignment was done by picking association with the highest covariance values.

Evaluation Goals

- Maximize RocksDB's IO throughput by tuning ten parameters.
- Success criteria for the tuner is to converge faster and find the most performant IO throughput.
- Highlight the efficiency of cluster-based multi-task approach in exploiting system decomposability.

Mix of discrete parameters with large possible values.

RocksDB tuned parameters, every parameter is a discrete variables.

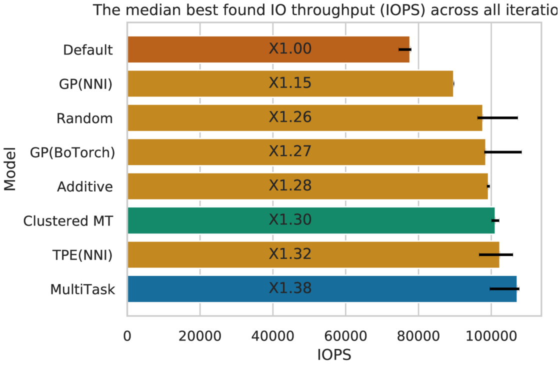
Parameter	Range	Default
max_background_compactions	$[1, 2^8]$	1
max_background_flushes	$[110]$	1
write_buffer_size	$[1, 15 * 10^7]$	2^{26}
max_write_buffer_number	$[1, 2^7]$	2
min_write_buffer_number_to_merge	$[1, 2^5]$	1
max_bytes_for_level_multiplier	$[5, 15]$	10
block_size	$[1, 5 * 10^5]$	2^{12}
level0_file_num_compaction_trigger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

Workload has a dynamic changing read and write behavior.

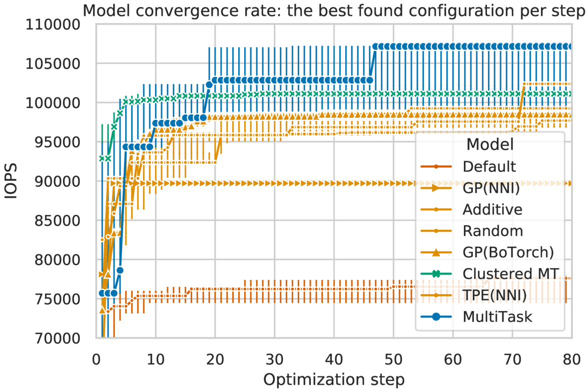
- Used RocksDB's workload generator *db_bench* to simulate a social graph workload.
- It runs 50 million queries in fifteen-minutes.
- The workload has a mixture of all RocksDB operations: 78% GET, 13% PUT, 6% DELETE, and 3% Iterate.
- The pattern change every 5000 operation reflecting real workload.

Performance improvement over the default.

The best IO throughput found in 100 steps with the median of five runs reported with the minimum and maximum achieved IOPS.



Steps to find optimal configuration maximizing IOPS.



Best found configuration per training step.

Summary and opportunities

- We presented a mechanism to find optimal configurations in RocksDB using Bayesian optimization.
- Achieved faster convergence by utilizing multi-task learning that provides more information per execution run.
- Decomposability of parameter space through manual assignment to a specific task led to even faster and stable convergence while reducing runtime complexity.
- Future work to use GMM or probabilistic clustering methods can automate the decomposability process.