ENDURE: A Robust Tuning Paradigm for LSM Trees Under Workload Uncertainty

BOSTON UNIVERSITY

Andy Huynh, Harshal A. Chaudhari, Evimaria Terzi, Manos Athanassoulis

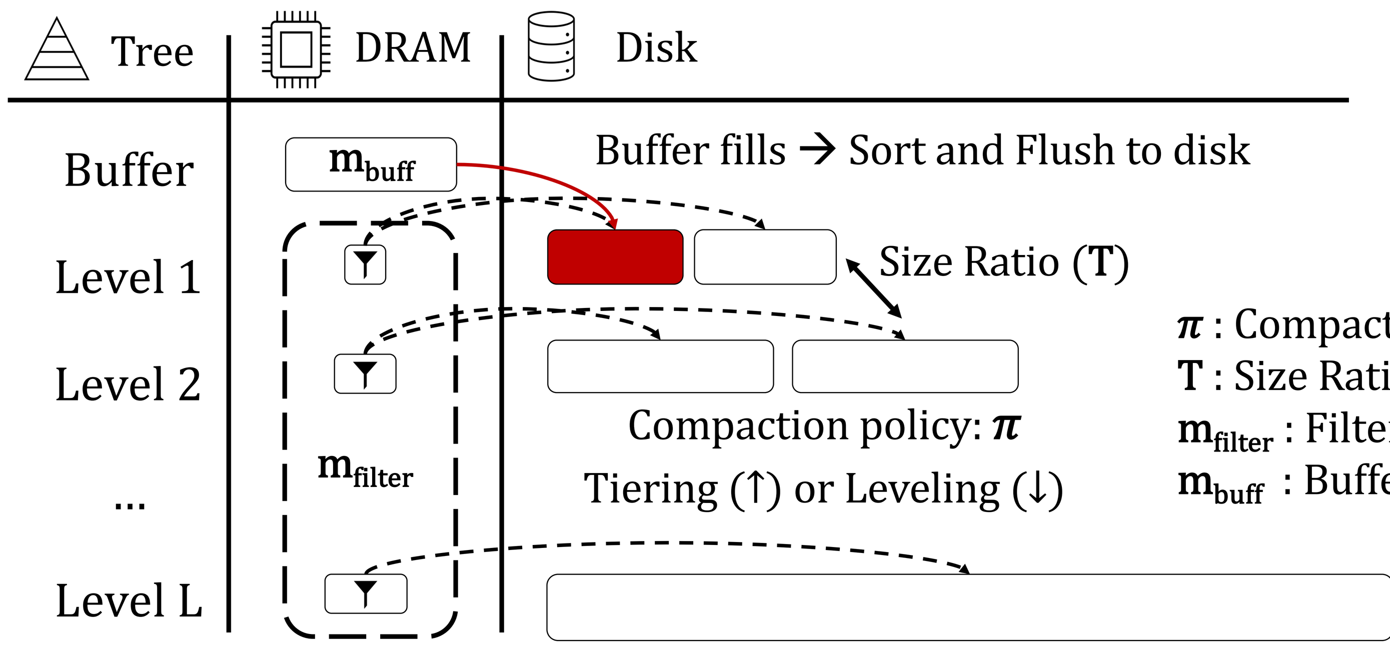
Age of LSM trees

RocksDB, WT, levelDB, cassandra

Flexibility for applications

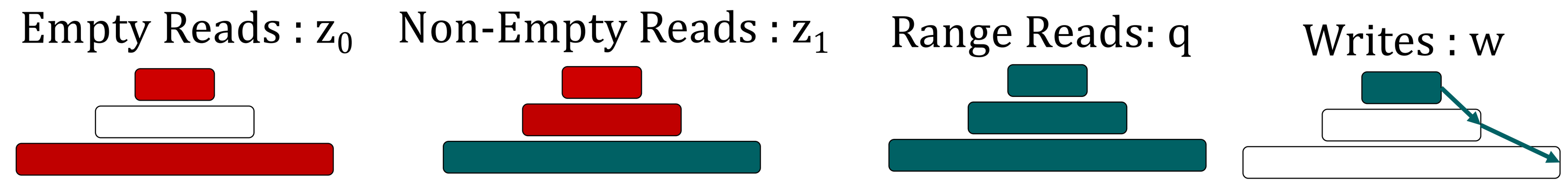
Compaction | Buffer Size | Size Ratio

High impact tuning knobs



The LSM Tuning Problem

Cost is sum of expected I/Os per query type, weight by frequency



Both problems are finding the design Φ that minimizes C.
Nominal bases decision on w, while **Robust** considers all workloads in U_w^ρ

$\mathbf{w}$: Workload (z_0, z_1, q, w)

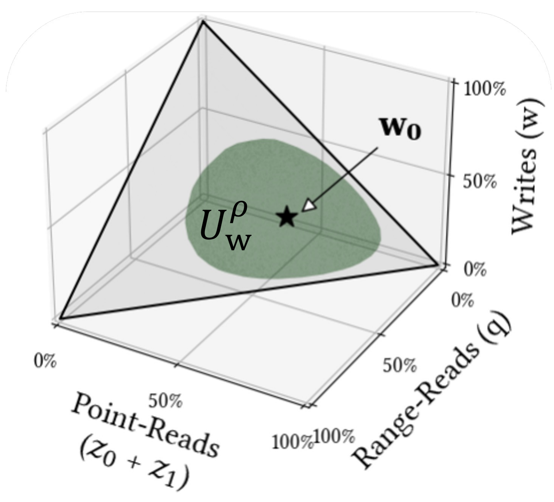
Φ : LSM Tree Design $(m_{buff}, m_{filter}, T, \pi)$

C : Cost (I/O)

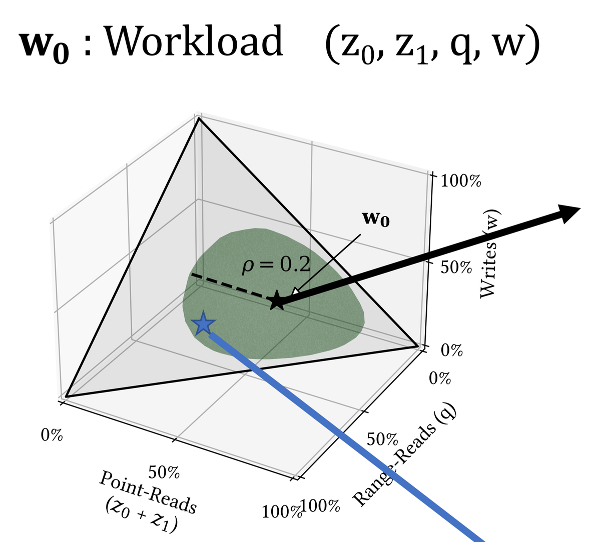
$$\Phi^* = \operatorname{argmin}_\Phi C(\mathbf{w}, \Phi) \qquad \text{Nominal}$$

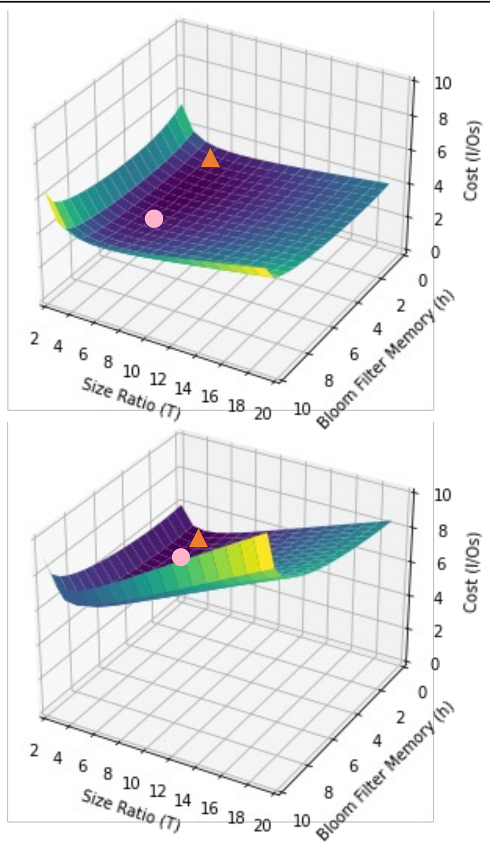
U_w^ρ: Uncertainty neighborhood of workloads
ρ : Size of this neighborhood

$$\Phi^* = \operatorname{argmin}_\Phi C(\hat{\mathbf{w}}, \Phi)$$
$$s.t., \quad \hat{\mathbf{w}} \in U_w^\rho \qquad \text{Robust}$$



The LSM Tuning Problem





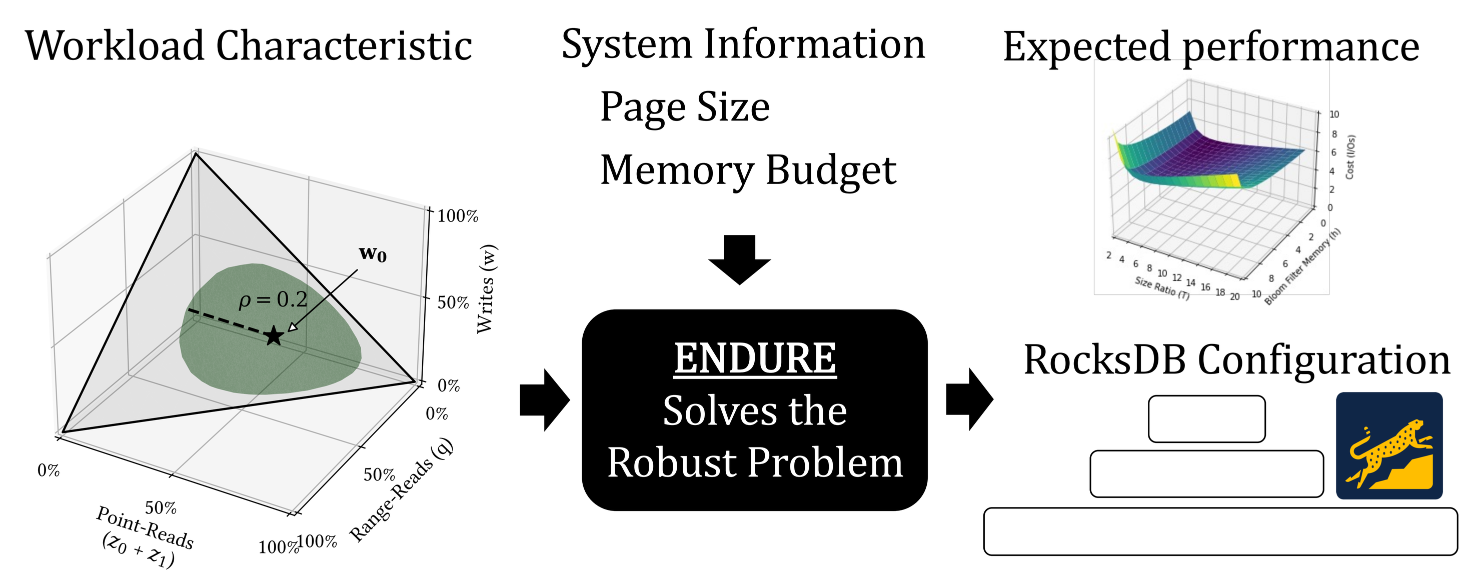
- Optimal configuration for the workload
- Robust configuration for the workload neighborhood

Nominal tuning may lead to suboptimal tunings if observed workloads and expected workloads are far

Robust tuning solution minimizes highest value among any workload in our uncertainty neighborhood

Uncertainty neighborhood defines which workloads to consider for robust tuning

ENDURE: Robust Tuning

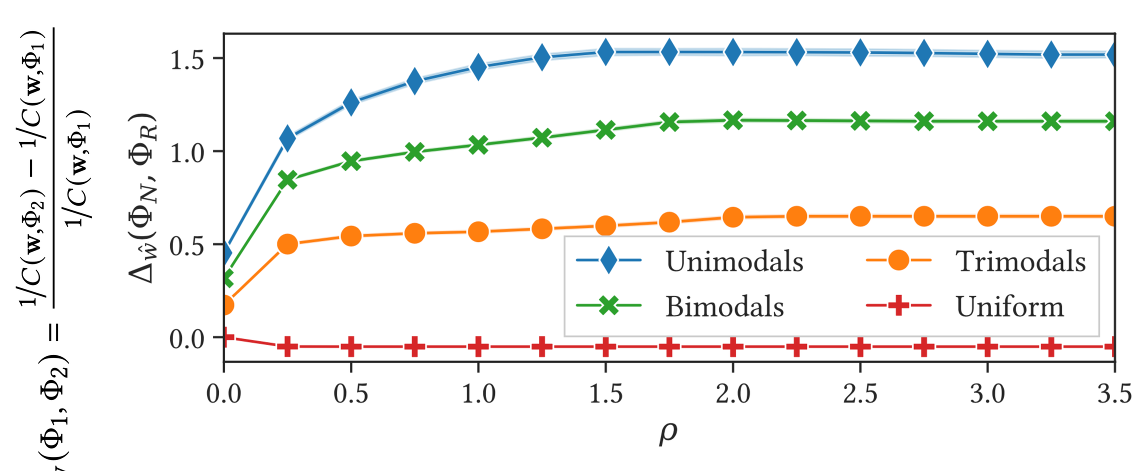


ENDURE implemented in Python **alongside RocksDB**

Users provide workload characteristics: expected workload and uncertainty

Selected Results (More can be found in our full paper)

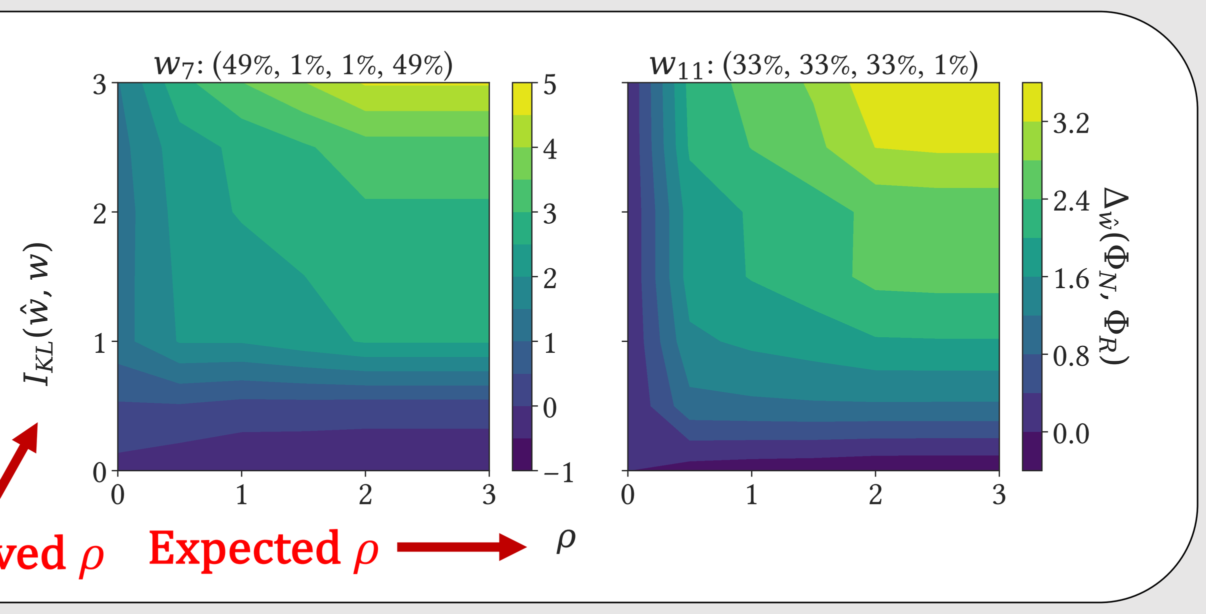
Idx	z_0	z_1	q	w	Type
0	25%	25%	25%	25%	Uniform
1	97%	1%	1%	1%	Unimodal
2	1%	97%	1%	1%	
3	1%	1%	97%	1%	
4	1%	1%	1%	97%	
5	49%	49%	1%	1%	Bimodal
6	49%	49%	1%	1%	
7	49%	49%	1%	1%	
8	49%	49%	1%	1%	
9	49%	49%	1%	1%	
10	49%	49%	1%	1%	
11	33%	33%	33%	1%	Trimodal
12	33%	33%	33%	1%	
13	33%	33%	33%	1%	
14	33%	33%	33%	1%	



Unbalanced workloads = overfitted nominal tunings

Highest throughput when observed and expected ρ **match**

Lowest throughput when ρ is **mismatched**



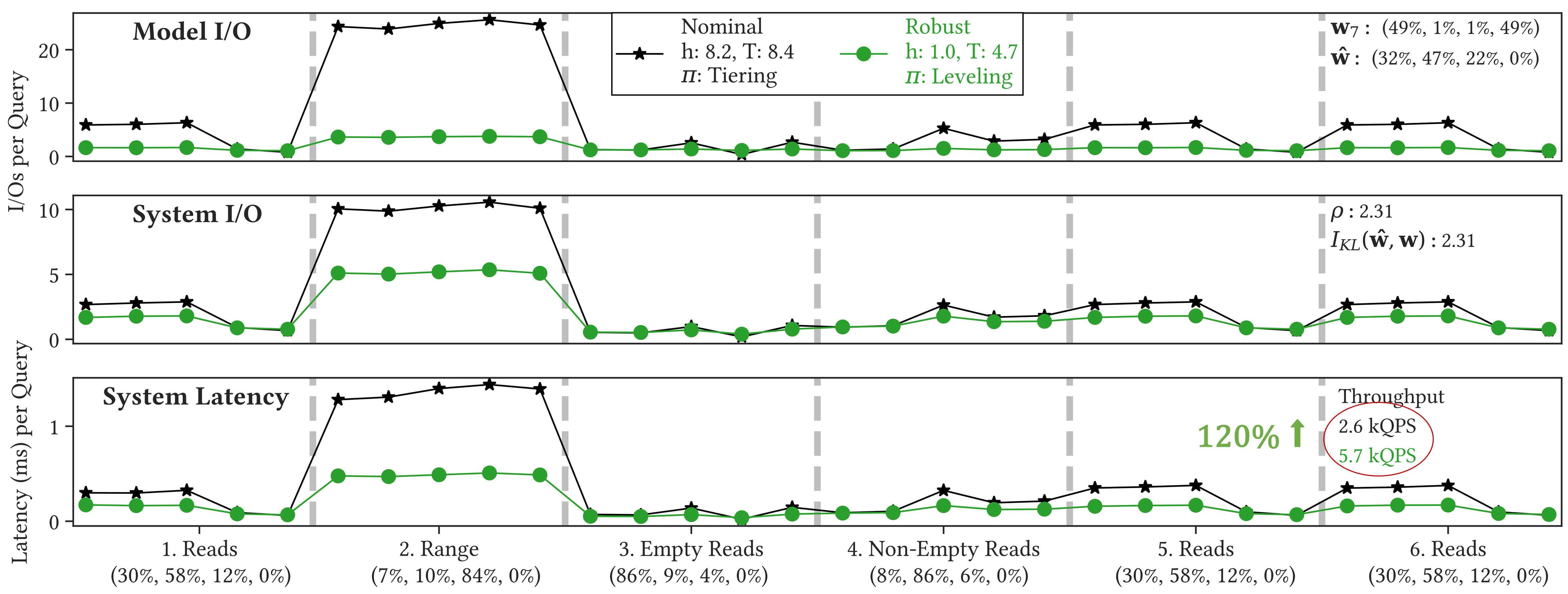
Selected Results

RocksDB instance
10 million unique key-value pairs of size **1KB**

6 sessions with **5 observation periods** per session
Observation period: 200K queries
Overall **6million** queries

Writes are unique
Range queries are short range queries (1-2 pages)

Small subset of results! Take a look at our paper for a more detailed analysis



BOSTON UNIVERSITY · IBM · NSF · MiDAS

github.com/BU-DiSC/endure