LINFENG ZHU | ASIANNA HAUGHTON | ROHIT SAHA

BOSTON UNIVERSITY

PRIVACY-PRESERVING DATABASE TUNING

BALANCING PRIVACY & PERFORMANCE WITH ENDURE

PAPER OVERVIEW

MOTIVATION

PRIVACY RISKS IN LSM-TREE DATABASES

- **LSM-Tree Databases**: Power data-driven apps like e-commerce platforms with efficient key-value operations
- **Privacy Issue**: Workload statistics are used to tune these systems, which can leak sensitive user information
- **Example**: In an e-commerce platform, a spike in write operations to a key range might reveal a user frequently buying medical supplies, hinting at a health condition
- **Consequence**: Attackers can infer personal details without accessing the data itself, just by analyzing workload patterns

DIFFERENTIAL PRIVACY AS A SOLUTION

- **Differential Privacy (DP)**: Adds noise to workload statistics to protect individual user data
- **How It Works**: In the e-commerce example, DP might adjust the write operation statistic (e.g. 0.2→ 0.22), masking individual actions
- **Benefit**: Preserves aggregate trends for system tuning while making it statistically impossible to infer personal details
- **Our Goal**: Quantify DP's impact on tuning, optimize noise levels, and enhance Endure to balance privacy and performance

BACKGROUND

CORE COMPONENTS

- **LSM Trees**
  - Storage engine in systems like RocksDB, Cassandra → **known for high write throughput**
  - Structure: **Data organized in levels, with merging** (compaction) of sorted runs to optimize reads
  - Key Parameters: Size ratio, merge policy (tiering vs. leveling), memory for Bloom filters
  - Tuning Need: Performance depends on workload-based tuning (e.g. Bloom filter allocation)
- **Endure's Robust Tuning**
  - Addresses **workload uncertainty** by modeling a neighborhood of probable workloads
  - Goal: Compute configurations **maximizing worst-case throughput** across the neighborhood
  - Traditional Assumption: **Expected workload is known**, with potential deviations
- **Differential Privacy (DP)**
  - Framework to release aggregate statistics without revealing individual data
  - Mechanism: **Add noise to workload proportions**
  - Privacy Parameter (ε): Smaller ε increases privacy by adding more noise, **perturbing the workload**
  - Goal in LSM Tuning: Prevent inference of individual user operations from the workload distribution

EXPERIMENT DESIGN

PROJECT WORKFLOW

Ω	$\rightarrow$	ρ	$\rightarrow$	Φ	$\rightarrow$	$C(\Omega, \Phi)$	$\rightarrow$	Results
Workload Proportion Vector		ENDURE Uncertainty Parameter		Optimal DB Configuration		Cost Function		
$\downarrow$								
Ω_ε	$\rightarrow$	ρ	$\rightarrow$	Φ_ε	$\rightarrow$	$C(\Omega_\varepsilon, \Phi_\varepsilon)$	$\rightarrow$	Results
Perturbed Workload Proportion Vector		ENDURE Uncertainty Parameter		Optimal DB Configuration For Perturbed Workload		Cost Function		

WORKLOAD PERTURBATION

- A perturb_workload function applies differential privacy to the workload vector **Ω** to produce $\boldsymbol{\Omega_\epsilon}$
- **Ω** = $[z_0, z_1, q, w]$
  - $\mathbf{z_0}$ = % empty point lookups | $\mathbf{z_1}$ = % point lookups | **q** = % range queries | **w** = % writes
  - $z_0 + z_1 + q + w = 1$
- $\boldsymbol{\Omega_\epsilon} = [z'_0, z'_1, q', w']$
  - A Laplace distribution is used to perturb(add noise) the workload proportions

$$\boxed{\hat{w}_i = w_i + \text{Laplace}\left(\frac{\Delta}{\epsilon}\right)}$$

  - Δ = sensitivity
  - ε = noise parameter
  - The vector is then normalized to ensure $z'_0 + z'_1 + q' + w' = 1$

PROJECT WORKFLOW

$\Omega \rightarrow \rho \rightarrow \Phi \rightarrow C(\Omega, \Phi) \rightarrow$ Results

Workload Proportion Vector

ENDURE Uncertainty Parameter

Optimal DB Configuration

Cost Function

$\downarrow$

$\Omega_\varepsilon \rightarrow \rho \rightarrow \Phi_\varepsilon \rightarrow C(\Omega_\varepsilon, \Phi_\varepsilon) \rightarrow$ Results

Perturbed Workload Proportion Vector

ENDURE Uncertainty Parameter

Optimal DB Configuration For Perturbed Workload

Cost Function

ENDURE'S UNCERTAINTY PARAMETER

- **What is ρ?**
  - ρ is the uncertainty parameter in Endure and is what separates nominal and robust tuning
- **How ρ Improves Robustness**
  - By tuning for a range of possible workloads, ρ ensures the configuration (Φ) can adjust to variation in workload during tuning
  - A larger ρ prepares the system for greater uncertainty
- **ρ and Nominal Tuning**
  - When ρ approaches 0, the uncertainty region shrinks, and Endure's tuning becomes equivalent to nominal tuning (optimizing for the expected workload only)
- **Our Experiment with ρ**
  - We test how different ρ values (0.1, 0.3, 0.5, etc.) impact performance

PROJECT WORKFLOW

$\Omega \rightarrow \rho \rightarrow \Phi \rightarrow C(\Omega, \Phi) \rightarrow$ Results

Workload Proportion Vector

ENDURE Uncertainty Parameter

Optimal DB Configuration

Cost Function

$\downarrow$

$\Omega_\varepsilon \rightarrow \rho \rightarrow \Phi_\varepsilon \rightarrow C(\Omega_\varepsilon, \Phi_\varepsilon) \rightarrow$ Results

Perturbed Workload Proportion Vector

ENDURE Uncertainty Parameter

Optimal DB Configuration For Perturbed Workload

Cost Function

ENDURE’S OPTIMAL LSM-TREE CONFIGURATION

- **What is Φ?**
  - Φ is the optimized LSM tree configuration for the expected workload
- **Inputs to Φ**
  - Workload: Either the original workload (Ω) or the perturbed workload (Ω_ε)
  - Uncertainty Parameter (ρ)
- **Outputs of Φ**
  - Optimal size ratio (T), memory allocation for Bloom filters (m_{filt}), and compaction policy (π) based on workload proportions

PROJECT WORKFLOW

$\Omega \rightarrow \rho \rightarrow \Phi \rightarrow C(\Omega, \Phi) \rightarrow \text{Results}$

Workload Proportion Vector → ENDURE Uncertainty Parameter → Optimal DB Configuration → **Cost Function** → Results

$\downarrow$

$\Omega_\varepsilon \rightarrow \rho \rightarrow \Phi_\varepsilon \rightarrow C(\Omega_\varepsilon, \Phi_\varepsilon) \rightarrow \text{Results}$

Perturbed Workload Proportion Vector → ENDURE Uncertainty Parameter → Optimal DB Configuration For Perturbed Workload → **Cost Function** → Results

COST FUNCTION

- **What is the Cost Function?**
  - The cost function $C(\Omega, \Phi)$ measures the expected cost of a workload under a given Φ

$$\boxed{C(\mathbf{w}, \Phi) = z_0 Z_0(\Phi) + z_1 Z_1(\Phi) + qQ(\Phi) + wW(\Phi)}$$

- **Inputs**
  - Original workload ($w = \Omega$) or perturbed workload ($w = \Omega_\varepsilon$)
  - Baseline configuration (Φ) or private configuration (Φ_ε)
- **Outputs of the Cost Function**
  - A scalar cost value representing the performance of the configuration on the workload
- **How We Used It to Compare Performance**
  - Compared baseline performance with private performance to assess the impact of differential privacy
  - Analyzed how different ρ values affect the performance gap between the two costs

PROJECT WORKFLOW

$\Omega \rightarrow \rho \rightarrow \Phi \rightarrow C(\Omega, \Phi) \rightarrow$ **Results**

Workload Proportion Vector

ENDURE Uncertainty Parameter

Optimal DB Configuration

Cost Function

$\downarrow$

$\Omega_\varepsilon \rightarrow \rho \rightarrow \Phi_\varepsilon \rightarrow C(\Omega_\varepsilon, \Phi_\varepsilon) \rightarrow$ **Results**

Perturbed Workload Proportion Vector

ENDURE Uncertainty Parameter

Optimal DB Configuration For Perturbed Workload

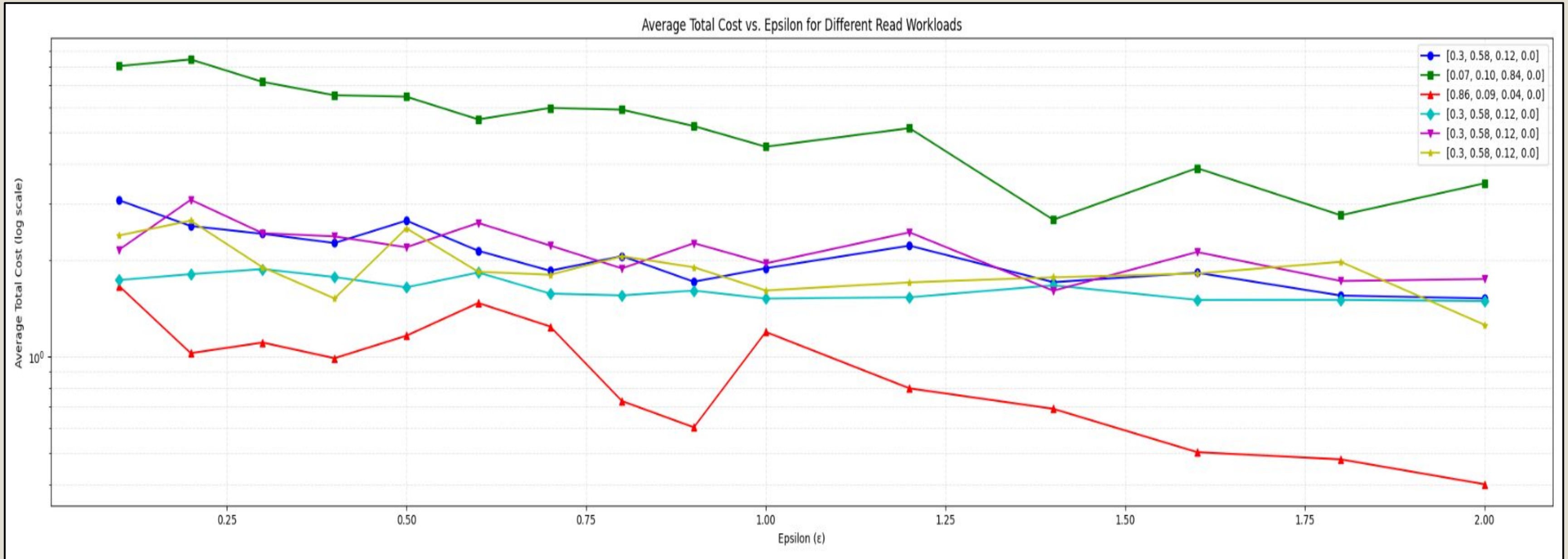
Cost Function

RESULTS

READ WORKLOADS

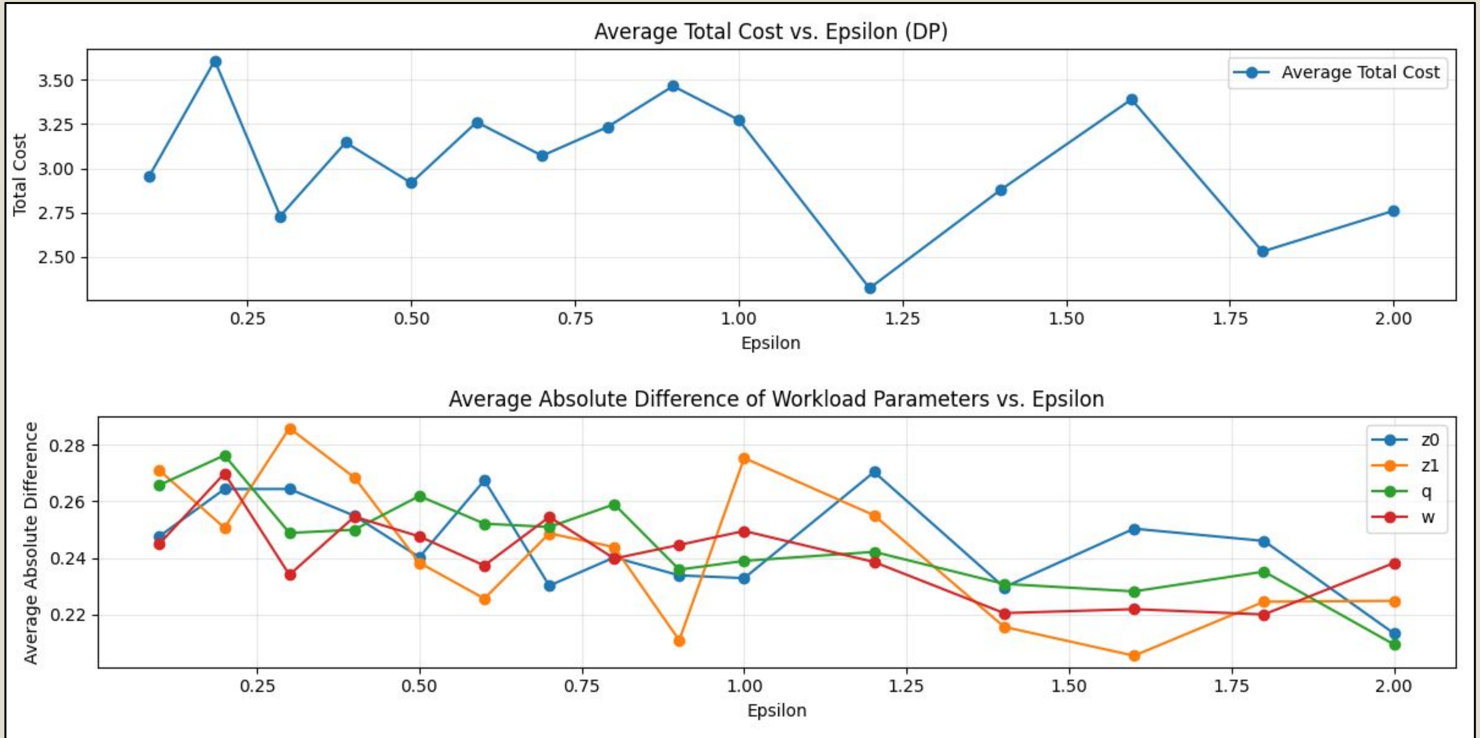
- **Figure: Average Total Cost vs. Epsilon across Different Read Workloads.**

  Each line represents a different query mix profile (e.g., range-heavy, point-heavy, etc.). Results are averaged across 100 trials, with log-scaled Y-axis to highlight performance differences.



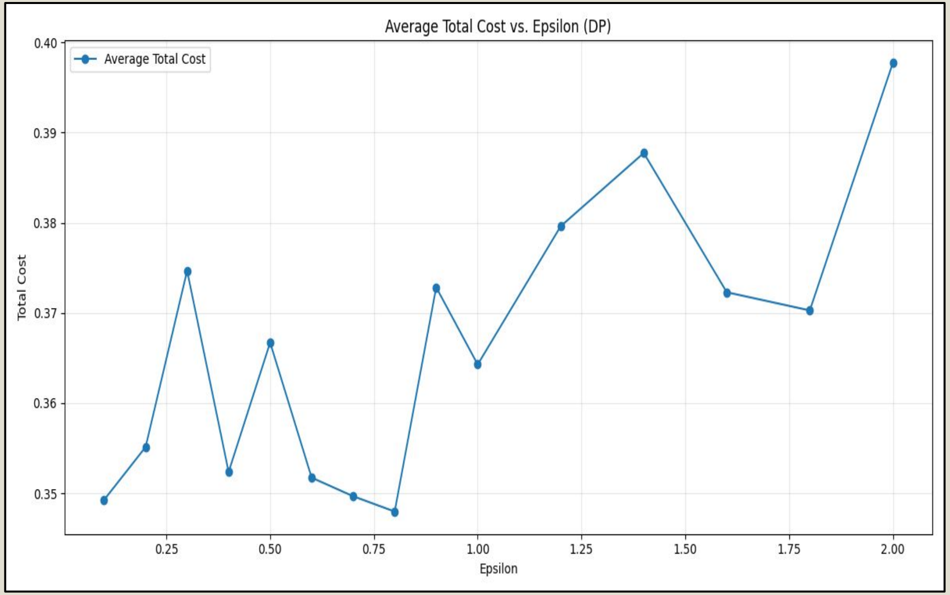
UNIFORM WORKLOADS

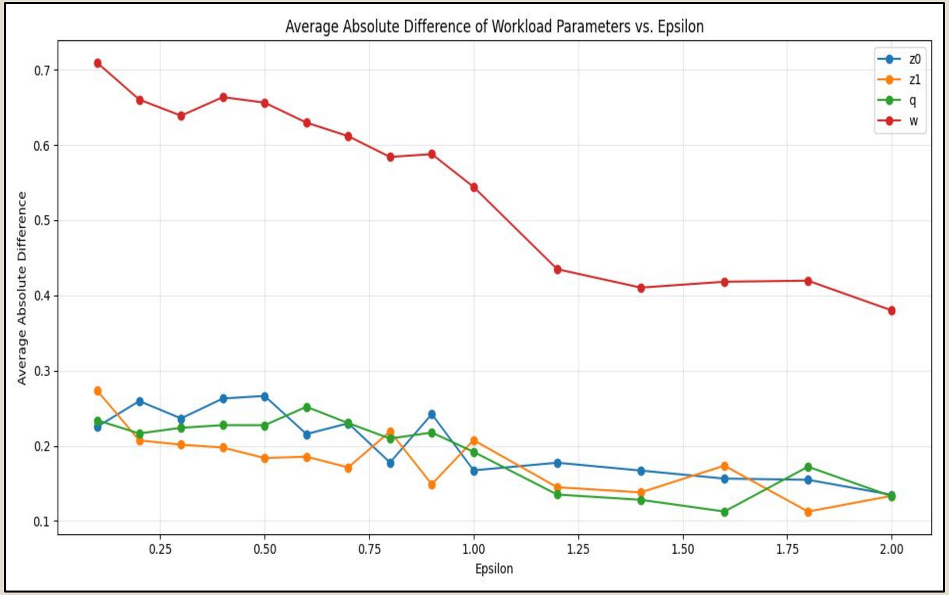
- Average Total Cost vs. Epsilon under uniform workload (equal z0, z1, q, w)



WRITE WORKLOADS

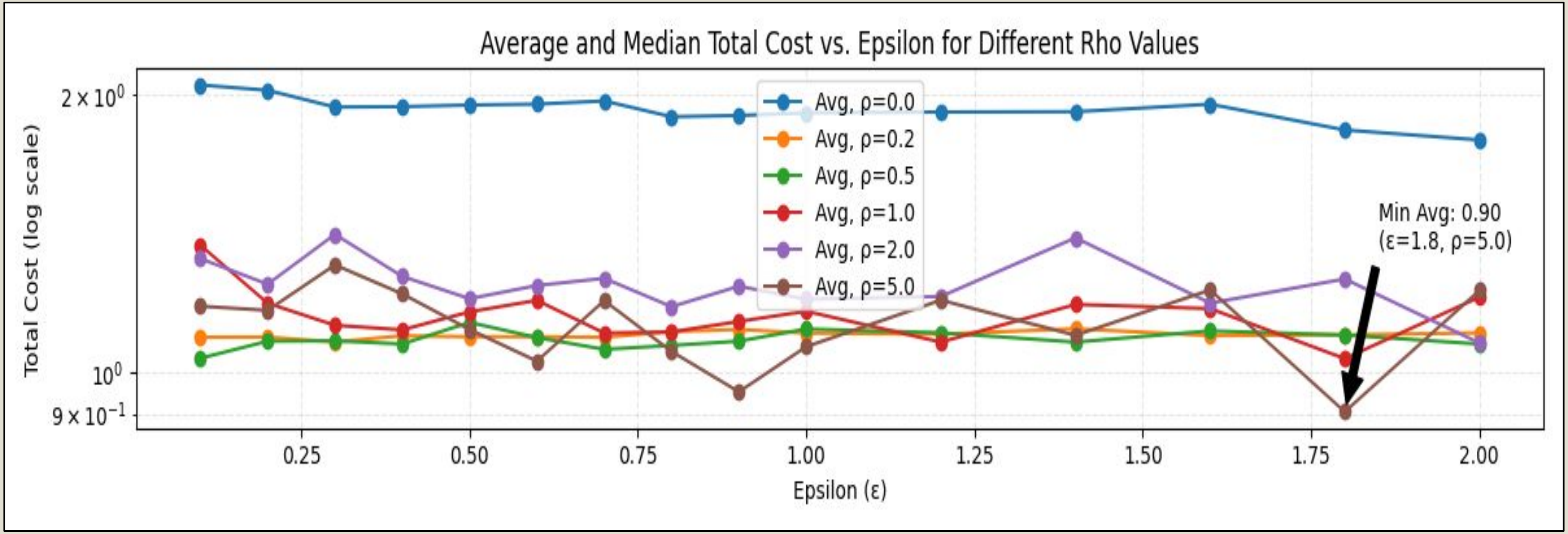
- **Workload = [z0=0.01, z1=0.01, q=0.01, w=0.97]** — Write-Intensive Profile





ρ PARAMETERIZATION

- Evaluate impact of ρ (robust tuning parameter) on total cost under DP
- Workload = Uniform [0.25, 0.25, 0.25, 0.25]



CONCLUSION

Conclusion

- **Differential privacy introduces real cost** in LSM-tree tuning — especially under strong privacy (small ε).
- **Workload sensitivity varies**:
  - Range queries (`q`) and empty lookups (`z0`) are **most impacted** by noise.
  - Write-heavy workloads are **more stable**, but still require robust modeling.
- **Robust tuning via ρ improves reliability**:
  - High ρ guards against workload distortion and improves total cost under DP.
- **No one-size-fits-all strategy**:
  - Tuning depends on workload type, privacy budget, and tolerance for risk.
- **Endure remains effective under DP** when paired with careful parameterization.

THANK YOU