Validity of Differential Privacy as a Workload Obfuscation Method

David Lee, Kathlyn Sinaga, Noah Picarelli-Kombert

01

Motivation

Why obfuscate workloads?

Consider this...

Workload information affects system performance.

DDoS attacks, sub-optimal performance due to a difference in expected and real workload

But there are many tuning knobs to optimize.

Global data organization, global search algorithm, metadata for searching, local data organization & search algorithm, modification policy, adaptivity

How do we safely interact with tuning services?

Could we give somewhat accurate information? How would that affect privacy and performance?

02

Problem Statement

- Differential privacy is a proven method of retaining privacy while keeping accuracy in data analysis.
- Could a differentially private workload distribution be used to create an effective tuning?

Questions we explored

Q1

Can a workload be made differentially private?

Q2

Could this obfuscated workload be used to create a competitive tuning?

03

Background

Differential Privacy & Robust Tuning with Endure

Differential Privacy

A mechanism that preserves privacy during data analysis

Privacy through plausible deniability

Query: Count the number of point queries in the workload.

Workload A

Output: 12

Workload B

Output: 12

Differential Privacy Terms

$\mathcal{M}$ **Randomized Algorithm**

An algorithm that perturbs the true query result to provide plausible deniability

$\mathcal{S}$ **Range**

A subset of the range of $\mathcal{M}$

x, y **Databases**

Collections of records from a universe $\mathcal{X}$

ε **Privacy Level**

Bounds the privacy guarantees of differential privacy

Differential Privacy Definition

A randomized algorithm $\mathcal{M}$ is ε-differentially private if for all $\mathcal{S} \subseteq \text{Range}(\mathcal{M})$ and for all x, y such that $||x - y||_1 \leq 1$:

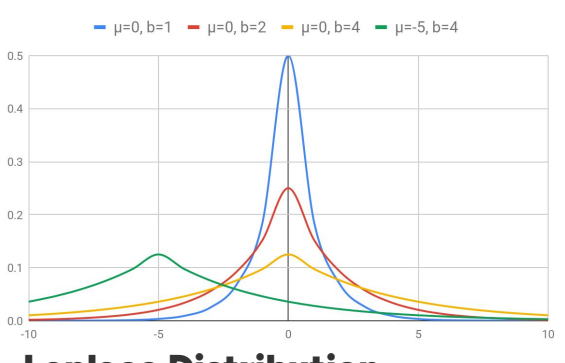
$$e^{-\varepsilon} \leq \frac{\mathbb{P}[\mathcal{M}(y) \in \mathcal{S}]}{\mathbb{P}[\mathcal{M}(x) \in \mathcal{S}]} \leq e^{\varepsilon}$$

The lower the ε, the more private.

Laplace Mechanism

$$\mathcal{M}_L(x, f(\cdot), \varepsilon) = f(x) + (Y_1, Y_2, \ldots, Y_n)$$

$$Y_i : \text{ i.i.d. RV from } Lap\left(\mu = 0, b = \frac{\Delta f}{\varepsilon}\right)$$



Laplace Distribution

$$Lap(x|\mu, b) = \frac{1}{2b}\exp\left(-\frac{|x-\mu|}{b}\right)$$

μ = 0 to not add bias to the noise.

Robust Tuning

Using Endure to create robust tunings

Two Tuning Paradigms

Nominal Tuning

$$\Phi^* = \arg\min_{\Phi} C(\mathrm{w}, \Phi)$$

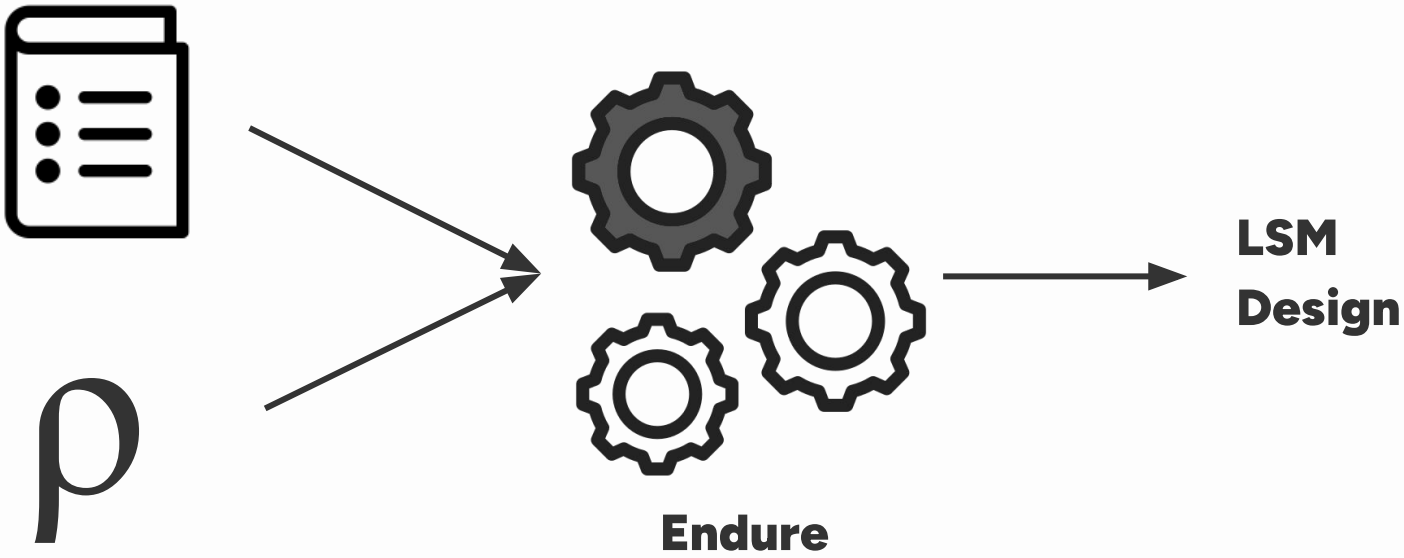
- The submitted workloads is assumed to be the exact real workload.
- The nominal tuning will probably have better average latency.

Robust Tuning

$$\Phi^* = \arg\min_{\Phi} C(\hat{\mathrm{w}}, \Phi)$$
$$s.t.\ \hat{\mathrm{w}} \in \mathcal{U}_{\mathrm{w}}$$

- The submitted workload is within a certain uncertainty region.
- Robust tuning optimizes the worst-case latency for the uncertainty region.

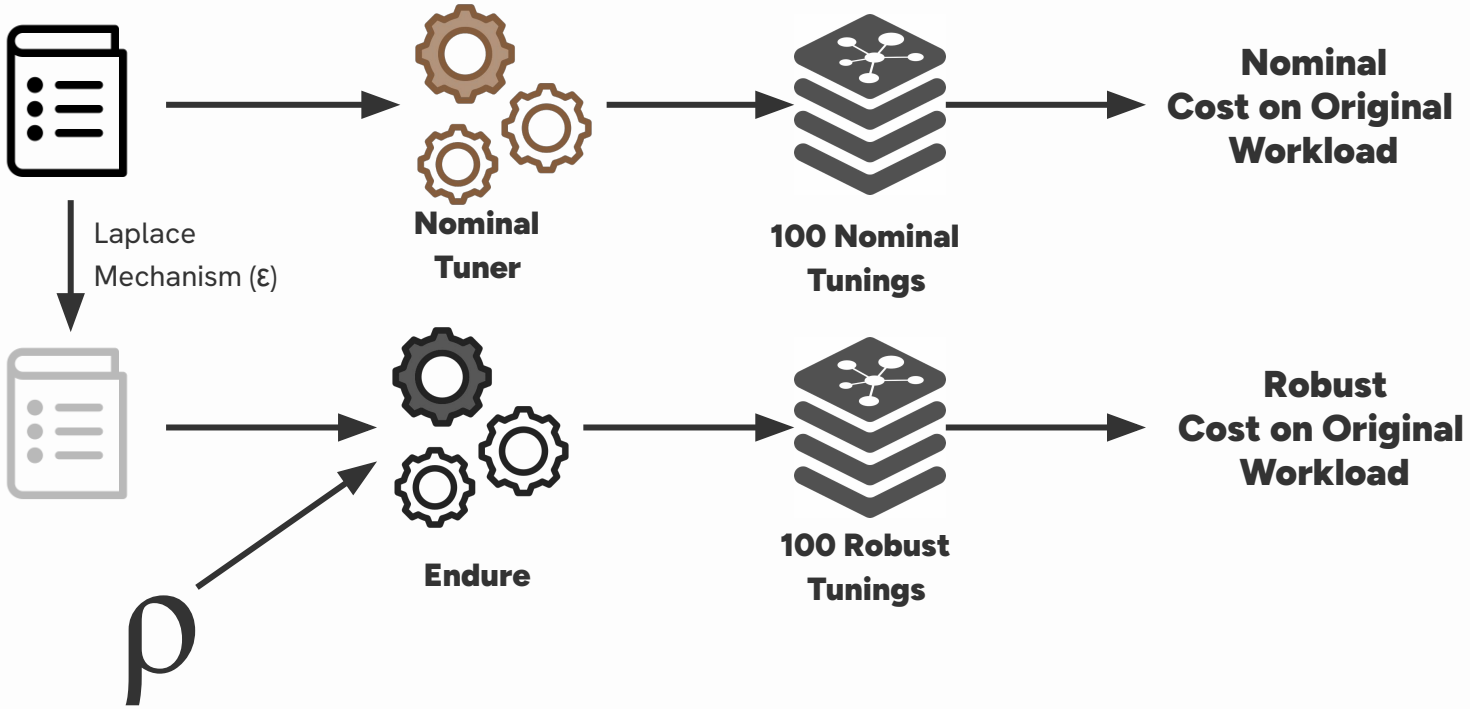
Endure as a black box



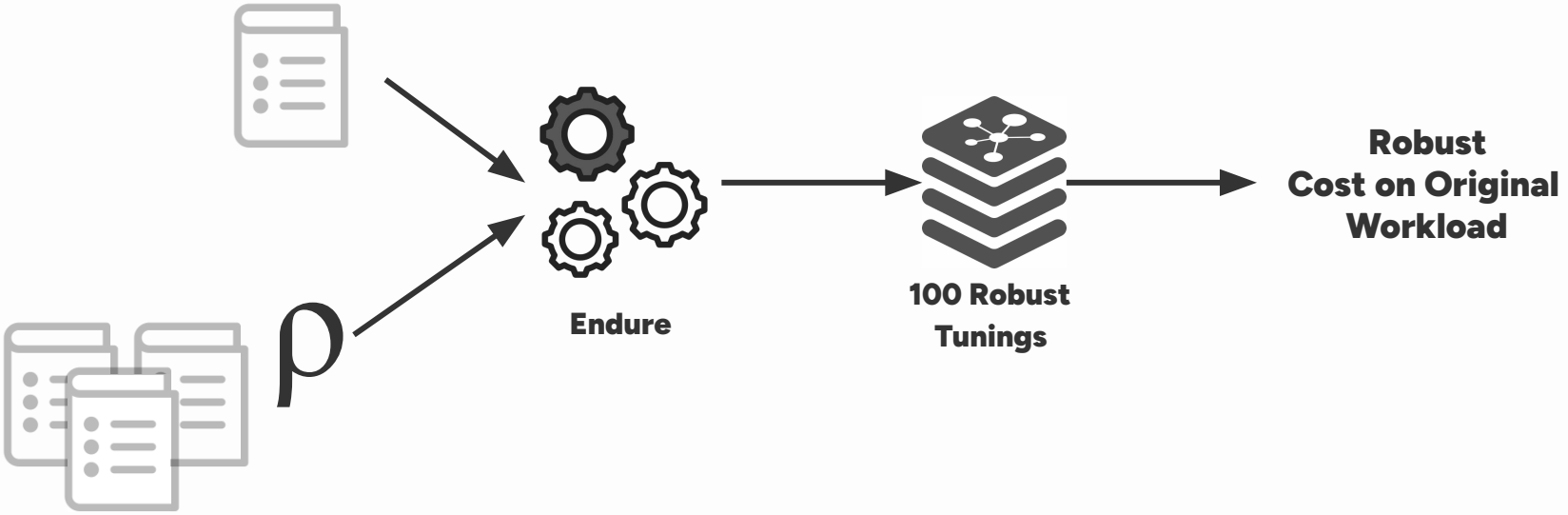
04

Experiment Design

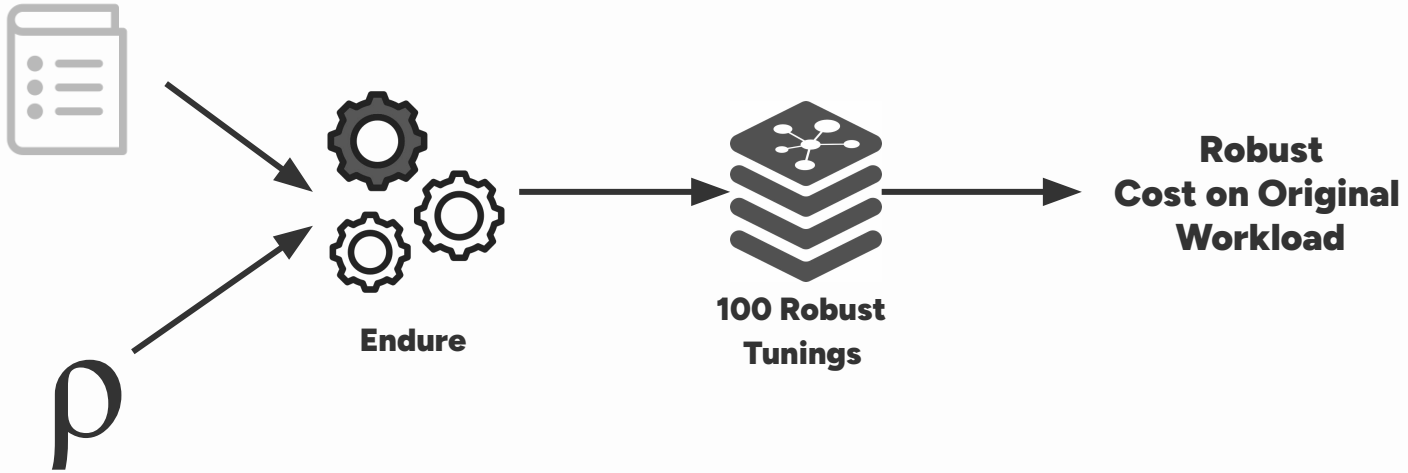
General Experiment Flow



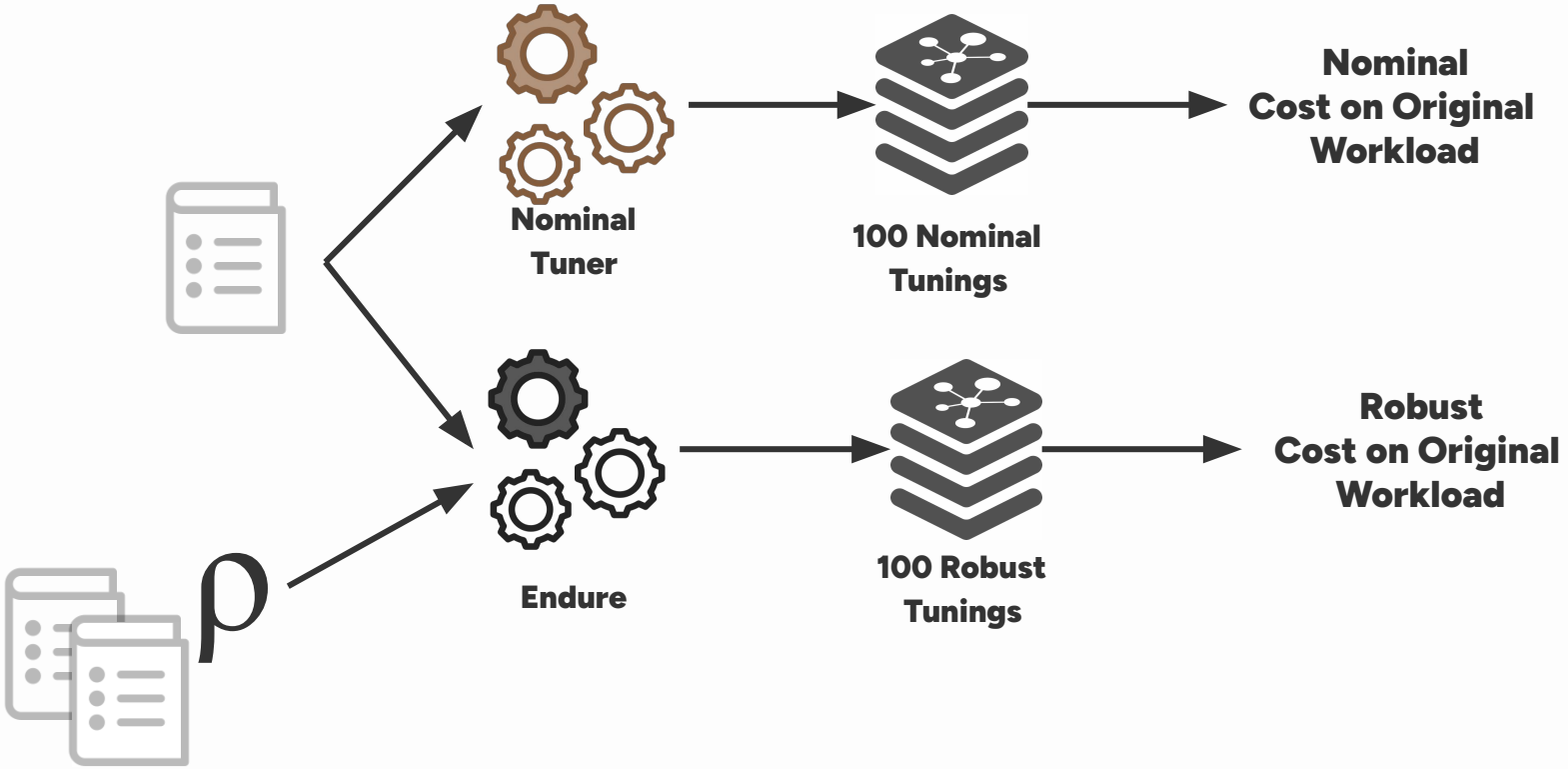
Dynamic Rho


ρ
Endure
100 Robust
Tunings
Robust
Cost on Original
Workload

Static Rho


ρ
Endure
100 Robust
Tunings
Robust
Cost on Original
Workload

Nominal vs. Robust

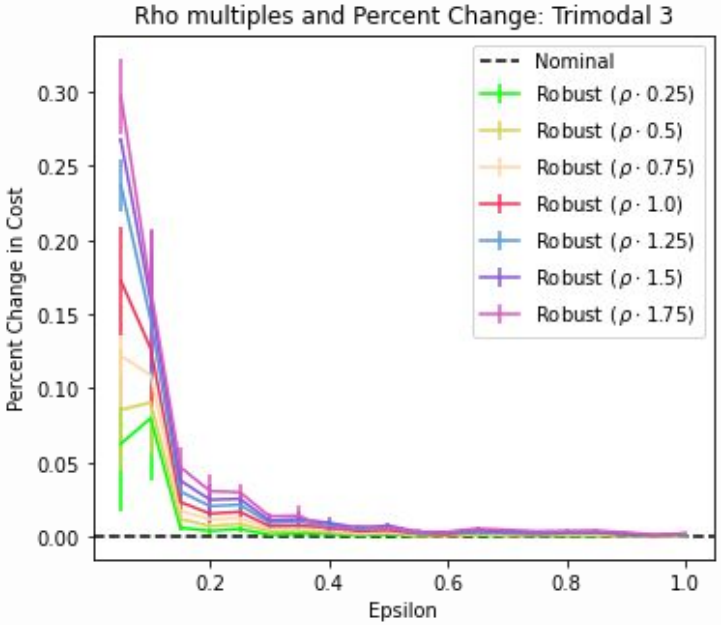


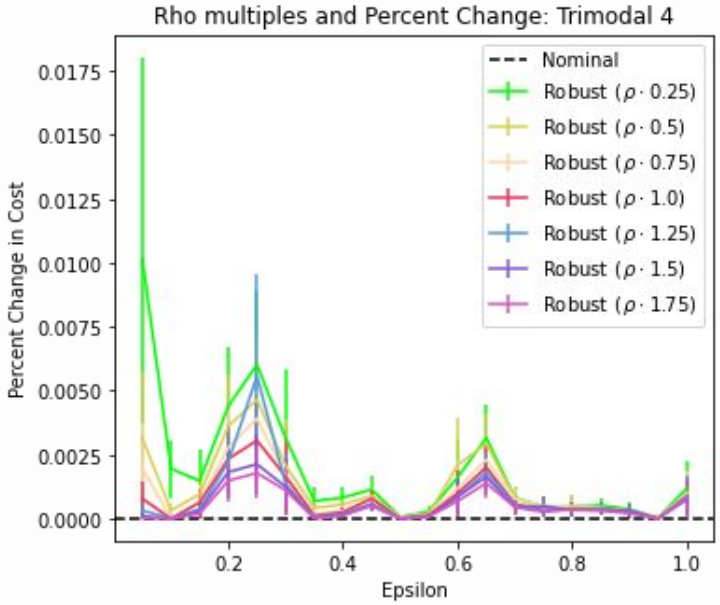
04

Results & Conclusions

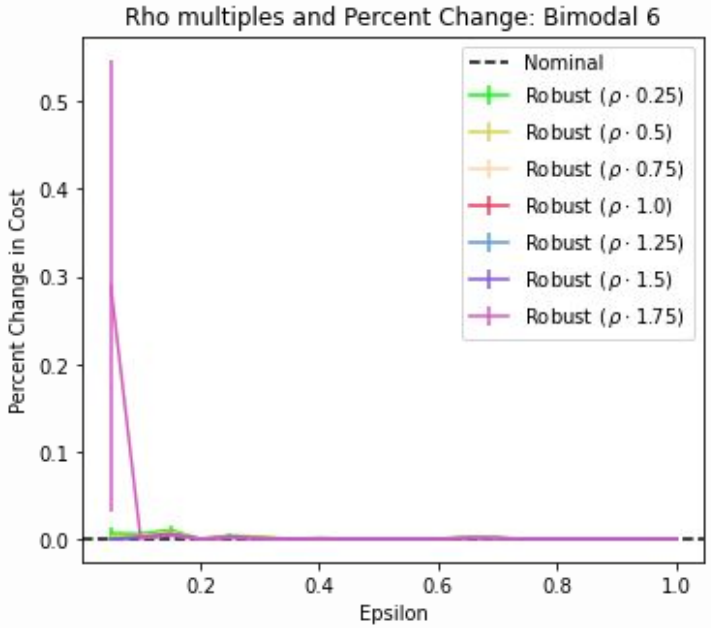
Dynamic Rho

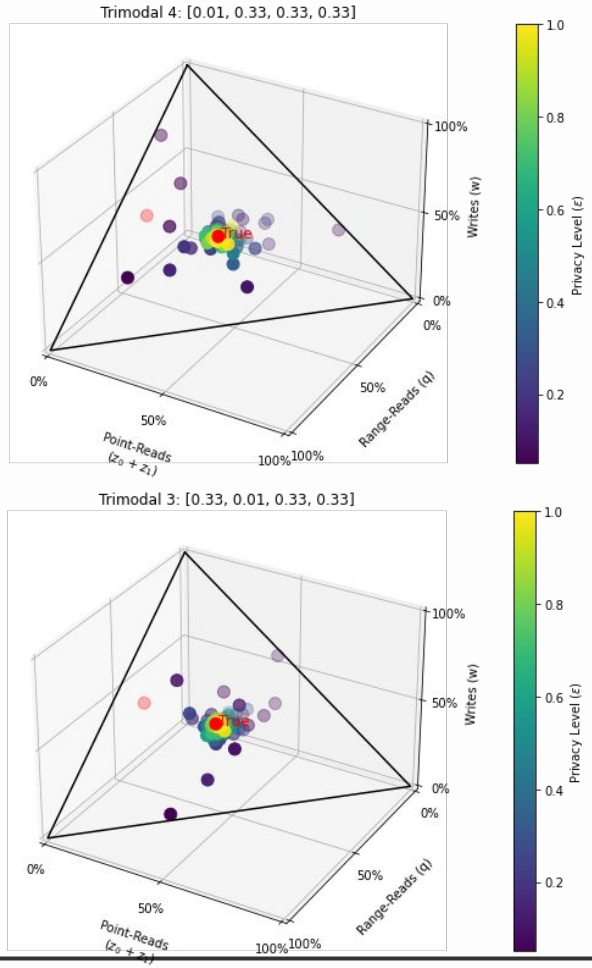
The smaller rho, the better

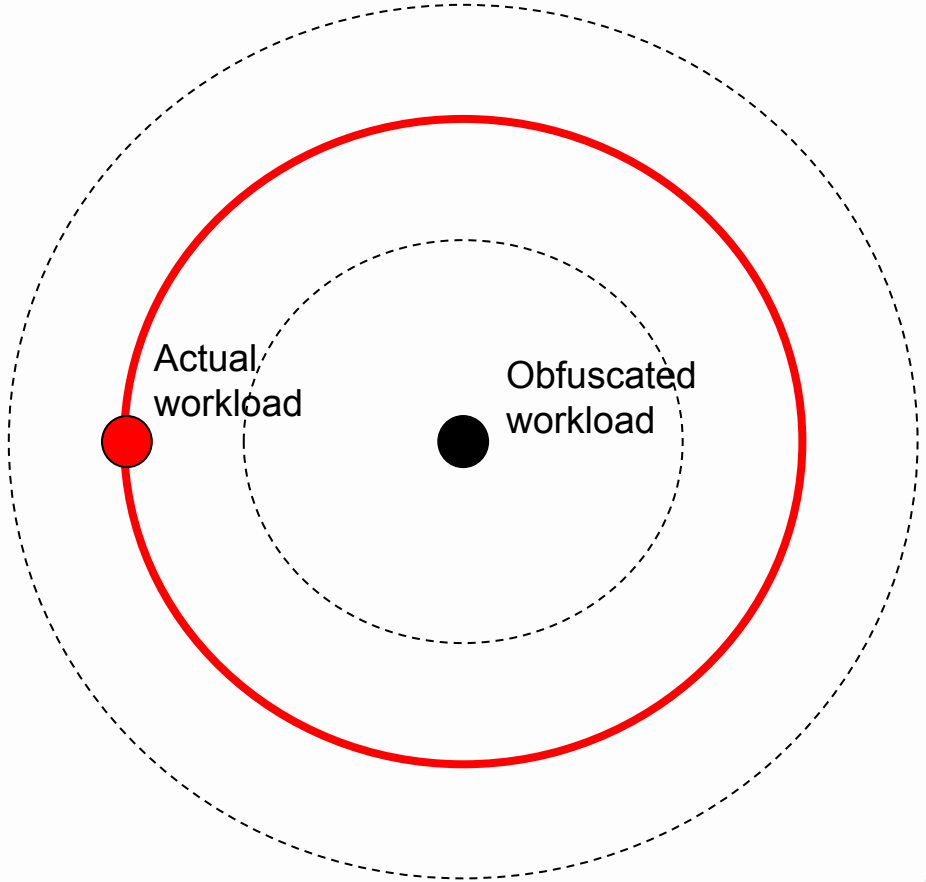




Almost exactly the same as nominal

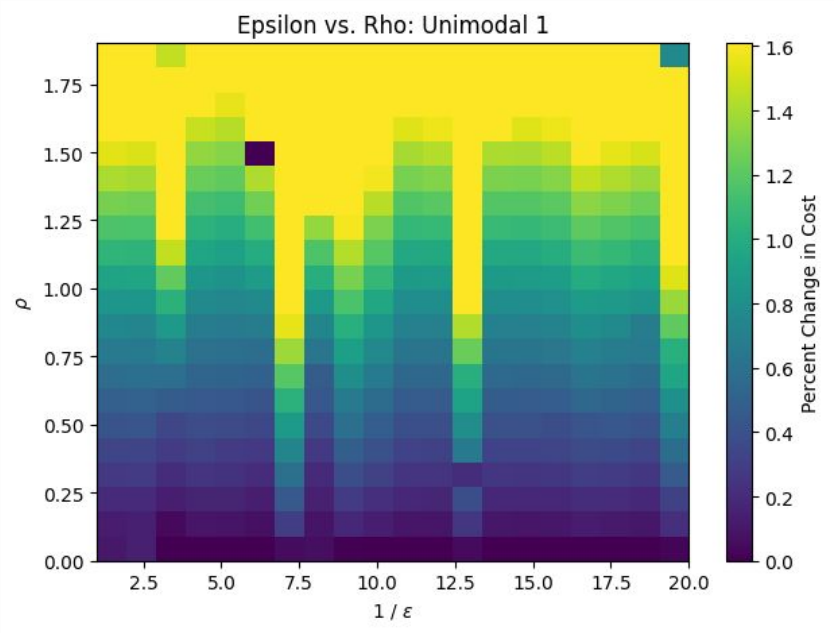


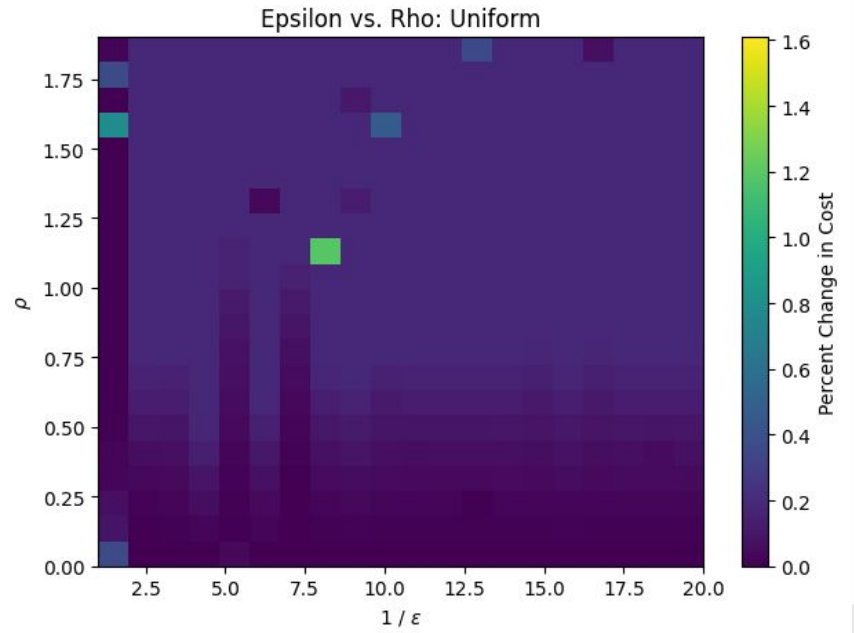

Trimodal 4: [0.01, 0.33, 0.33, 0.33]
True
Writes (w)
Point-Reads ($z_0 + z_1$)
Range-Reads (q)
Privacy Level (ε)
Trimodal 3: [0.33, 0.01, 0.33, 0.33]
True
Writes (w)
Point-Reads ($z_0 + z_1$)
Range-Reads (q)
Privacy Level (ε)


Actual workload
Obfuscated workload

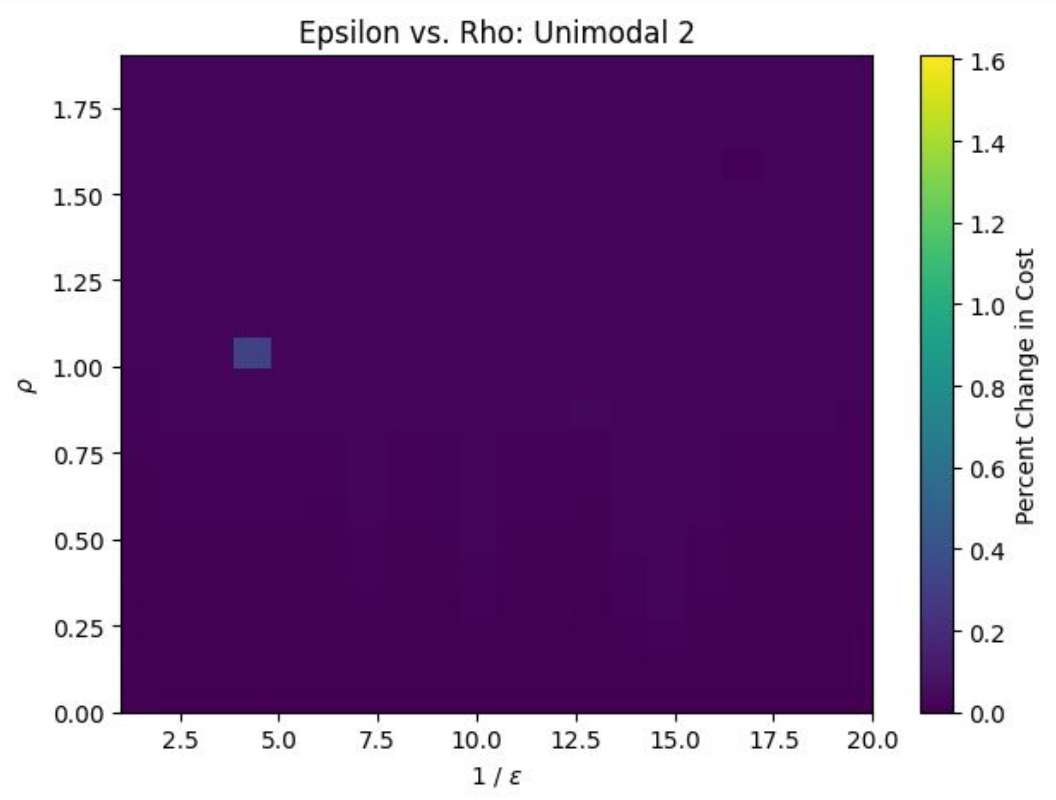
Static Rho

A Gradient


Epsilon vs. Rho: Unimodal 1
ρ
1 / ε
Percent Change in Cost
0.00
0.25
0.50
0.75
1.00
1.25
1.50
1.75
2.5
5.0
7.5
10.0
12.5
15.0
17.5
20.0
0.0
0.2
0.4
0.6
0.8
1.0
1.2
1.4
1.6

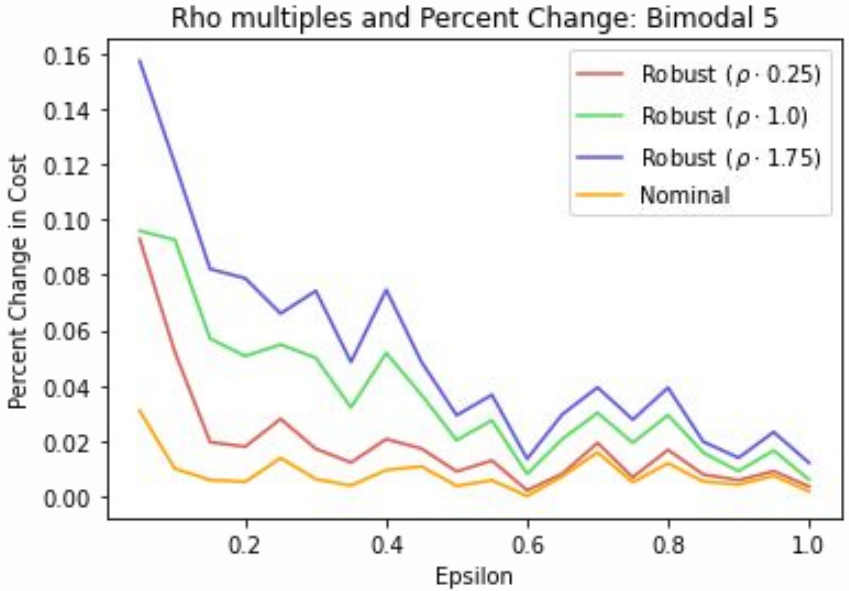

Epsilon vs. Rho: Uniform
ρ
1 / ε
Percent Change in Cost
0.00
0.25
0.50
0.75
1.00
1.25
1.50
1.75
2.5
5.0
7.5
10.0
12.5
15.0
17.5
20.0
0.0
0.2
0.4
0.6
0.8
1.0
1.2
1.4
1.6

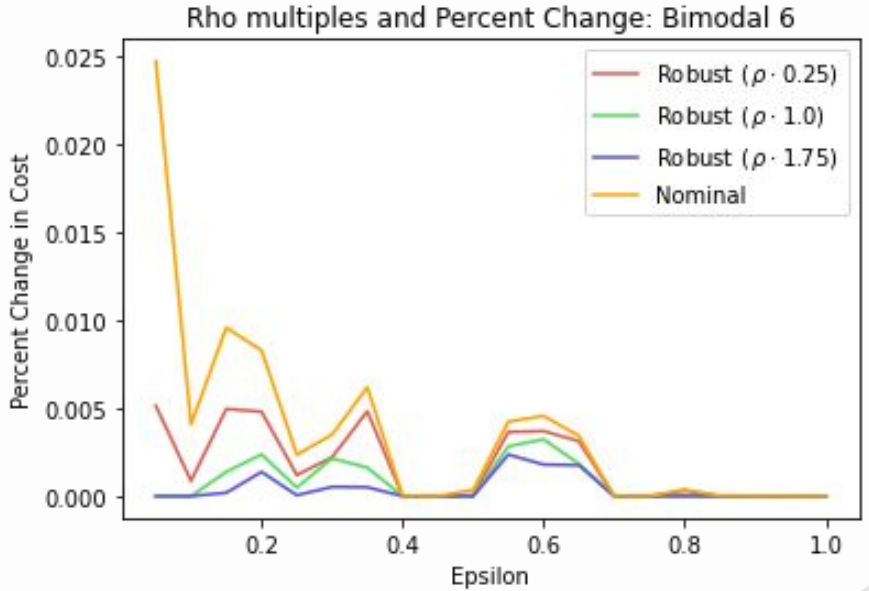
No Gradient


Epsilon vs. Rho: Unimodal 2
ρ
1 / ε
Percent Change in Cost
0.00
0.25
0.50
0.75
1.00
1.25
1.50
1.75
2.5
5.0
7.5
10.0
12.5
15.0
17.5
20.0
0.0
0.2
0.4
0.6
0.8
1.0
1.2
1.4
1.6

Robust vs. Nominal

Results Mirror the Dynamic Rho Experiment





Conclusions

- **Can a workload be made differentially private?**
  - Yes! When treated as a collection of aggregate results from a table of queries identified by type (zero-result point queries, non-zero-result point queries, range queries, writes).
- **Could an obfuscated workload be used to create a competitive tuning?**
  - Yes! With epsilon value 0.2 nominal tunings of an obfuscated workload perform similarly to nominal tunings of the actual workload.
- **Is any of this actually necessary?**
  - No! The noise does not hide general distribution. The “actual” workloads are likely to be just as inaccurate to future workloads as the obfuscated workloads are to them.

Questions we explored

Q1

Can a workload be made differentially private?

Yes! Workloads are represented as aggregate function results.

Q2

Could this obfuscated workload be used to create a competitive tuning?

Yes! With epsilon value 0.2, nominal tuning performs quite well.

Is this necessary?

No! The noise does not hide general distribution.

Future Work

Focus on enhancing robust tuning methods

- Laplace noise does not hide the general distribution.
- Endure is already proven to work with rho values wider than what we've experimented with.
- Focus on predictive tuning (new algorithms, ML, etc)

Work towards in-house tuning services with no third party

- Making open-source tuning software cuts out the need for DB owners to send data to a third party altogether

Future Work

- **Focus on enhancing robust tuning methods**
  - The Laplace noise does not hide the general distribution and the actual determined workload will not perfectly describe future activity on the DB.
  - Endure is already proven to work with rho values wider than what we've experimented with.
  - Using the innate inaccuracy as "natural privacy" and focusing on predictive tuning (new algorithms, ML, etc) might work better
- **Work towards in-house tuning services with no third party**
  - Making open-source tuning software cuts out the need for DB owners to send data to a third party altogether

References

[1] Cynthia Dwork and Aaron Roth. 2014. The Algorithmic Foundations of Differential Privacy. Vol. 9. Now Publishers Inc. 211–407 pages. https://doi.org/10.1561/0400000042

[2] Andy Huynh, Harshal Chaudhari, Evimaria Terzi, and Manos Athanassoulis. 2022. Endure: A Robust Tuning Paradigm for LSM Trees under Workload Uncertainty. Proceedings of the VLDB Endowment 15, 8 (2022), 1605–1616. https://vldb.org/pvldb/vol15/p1605-huynh.pdf

[3] Maurizio Naldi and Giuseppe D'Acquisto. 2015. Differential Privacy: An Estimation Theory-Based Method for Choosing Epsilon. (10 2015), 1–6.