Databases Are Blind to Energy

Given the hardware, which software choices change database energy?

Tobias Rahn (ELCA), **Andres Navarro Pedregal** (ETH Zurich),
Michal Friedman (ETH Zurich)

Databases Are Everywhere

Data centers use about 1.5% of the world's electricity today, and that could double by 2030.[1]

[1] International Energy Agency. 2025. Energy and AI. Technical Report. IEA. https://www.iea.org/reports/energy-and-ai

Databases Are Everywhere

Data centers use about 1.5% of the world's electricity today, and that could double by 2030.[1]

Usually we want more efficient hardware to reduce the energy consumption, but what about the software effects?

[1] International Energy Agency. 2025. Energy and AI. Technical Report. IEA. https://www.iea.org/reports/energy-and-ai

Problems Energy Measurements

- **Coarse meters.** Benchmarks and dashboards report **whole-machine** power, never **which query** caused the spike.

Problems Energy Measurements

- **Coarse meters.** Benchmarks and dashboards report **whole-machine** power, never **which query** caused the spike.
- **Default configurations.** Database configurations are chosen for **safety and compatibility**, not energy. No one measures how much energy those defaults cost.

Problems Energy Measurements

- **Coarse meters.** Benchmarks and dashboards report **whole-machine** pow

What is the energy cost of different software choices?

- **Defa** **afety and** those defaults cost.

Methodology

We fix **one machine** and change **one setting at a time**:

Methodology

We fix **one machine** and change **one setting at a time**:

Database engine	Durability	Indexes	Query plan

Methodology

We fix **one machine** and change **one setting at a time**:

Database engine	Durability	Indexes	Query plan

Our metric, energy per transaction:

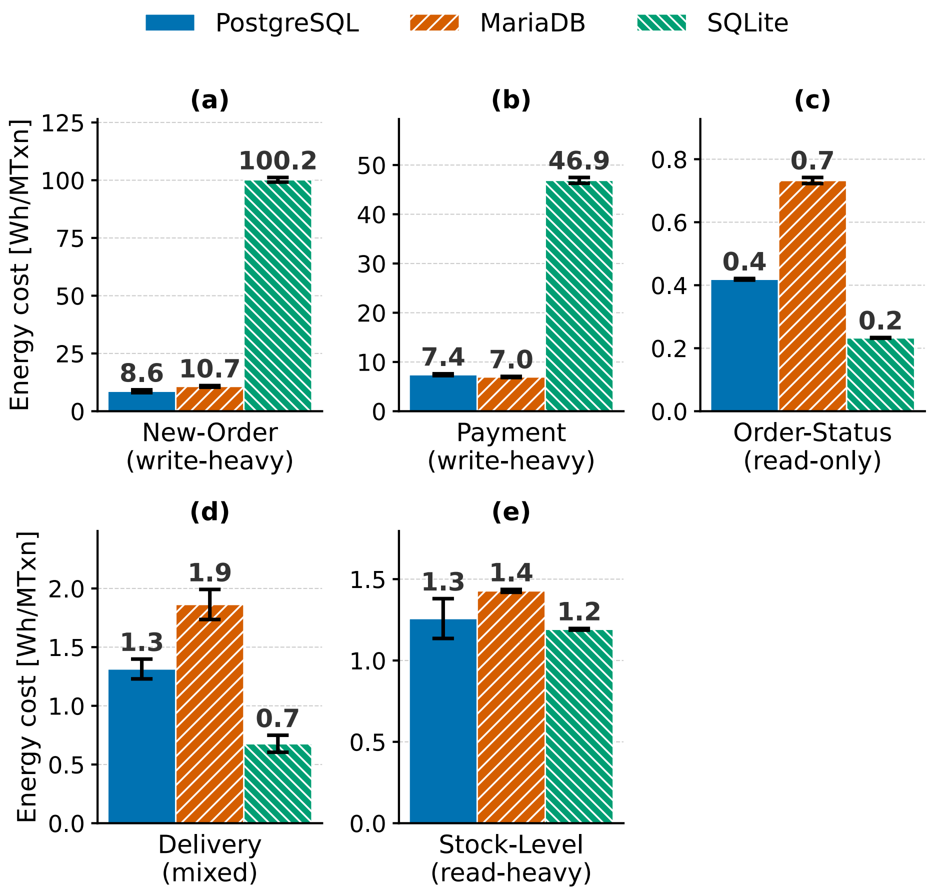
$$\mathrm{Wh/Txn} = \frac{\text{total energy}}{\text{transactions completed}} = \frac{\text{CPU + DRAM + disk}}{\text{transactions completed}}$$

It makes software choices comparable regardless of run duration.

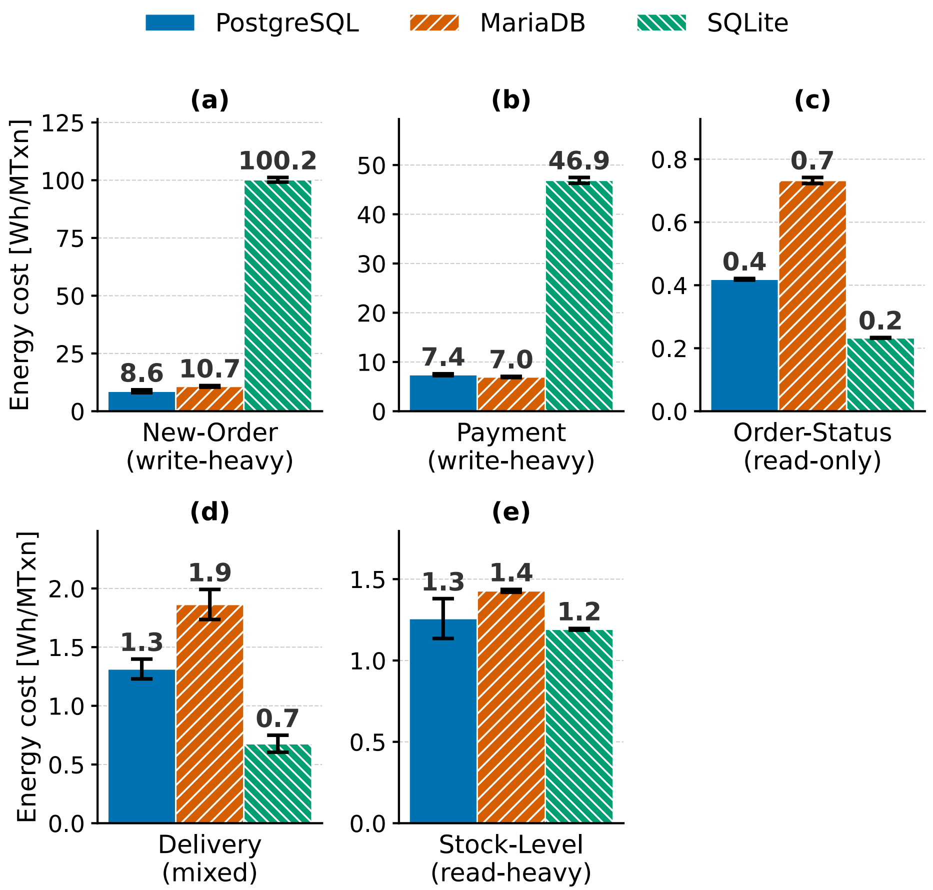
Setup

- Machines. A desktop with HDD and a server with a cached disk.
- Engines. **PostgreSQL**, **MariaDB** (client-server), and **SQLite** (embedded).
- Workloads. TPC-C: many small writes. TPC-H: big analytical queries.
- Energy. Seven RAPL-based tools + a disk model, all validated against a wall-power meter. Consistent trends across all tools.

Different Database Architectures

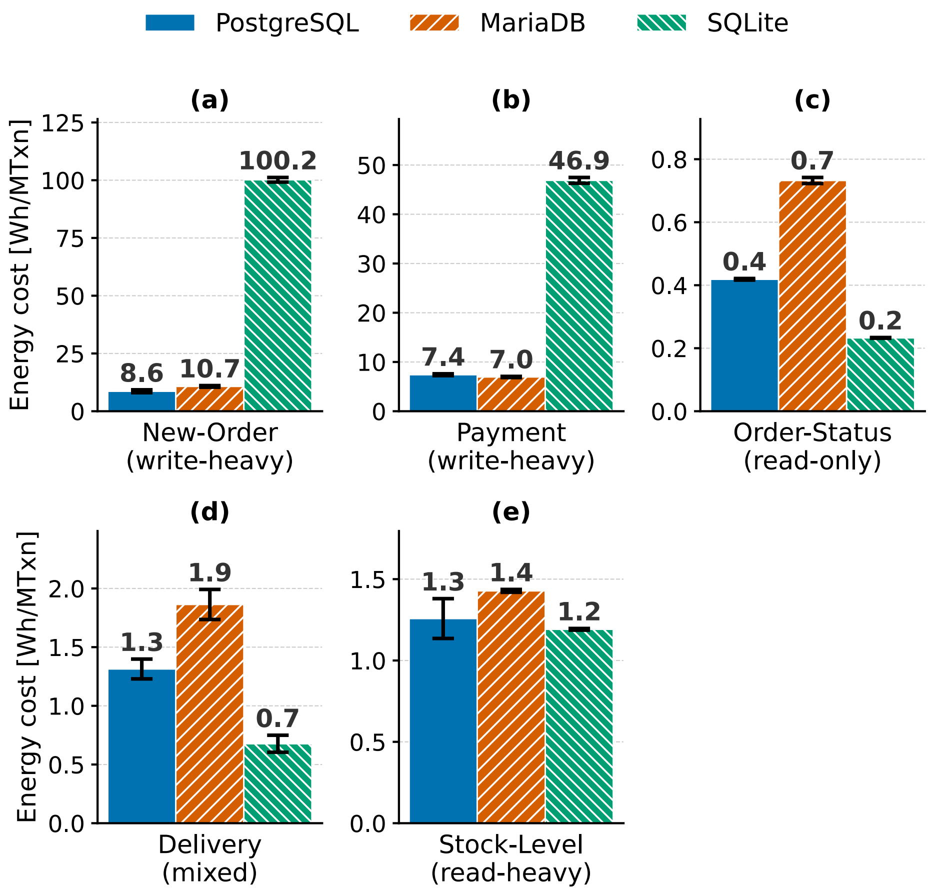


Different Database Architectures



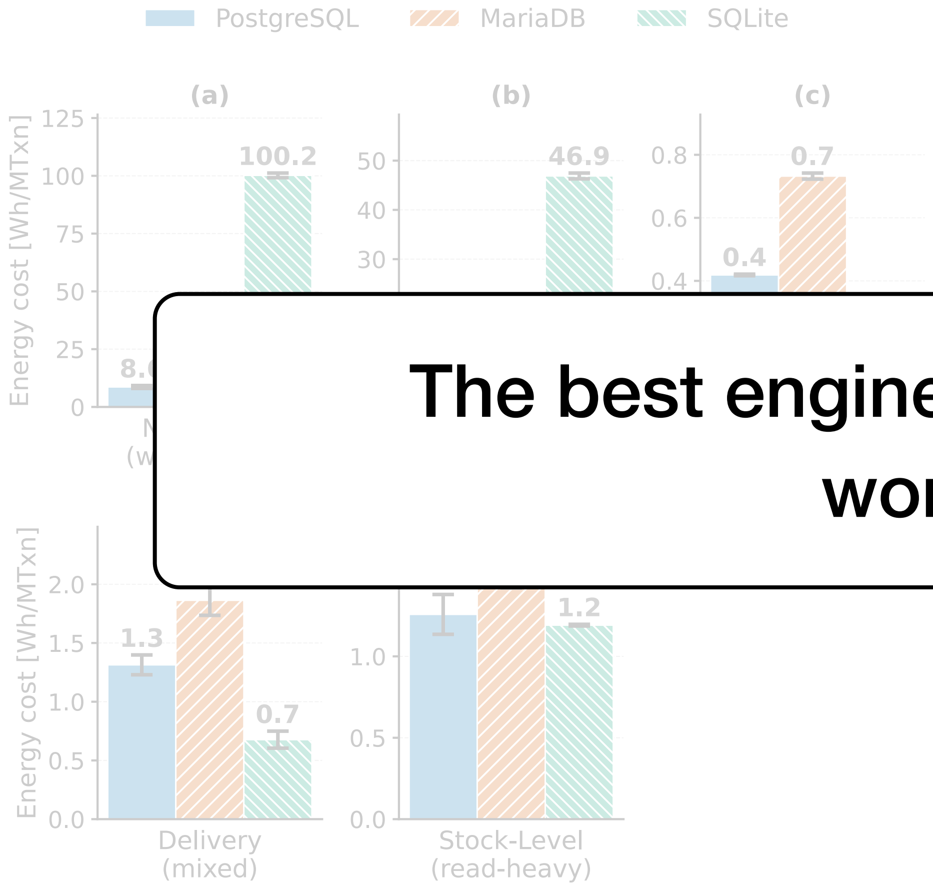
- Reads: **SQLite** uses up to **3x less** energy than **MariaDB**. No server overhead, no IPC.

Different Database Architectures



- Reads: **SQLite** uses up to **3x less** energy than **MariaDB**. No server overhead, no IPC.
- Writes: **SQLite** uses up to **11.6x more** energy than **PostgreSQL**. It syncs every single commit to disk.

Different Database Architectures



- Reads: SQLite uses up to 3x less energy than MariaDB. No

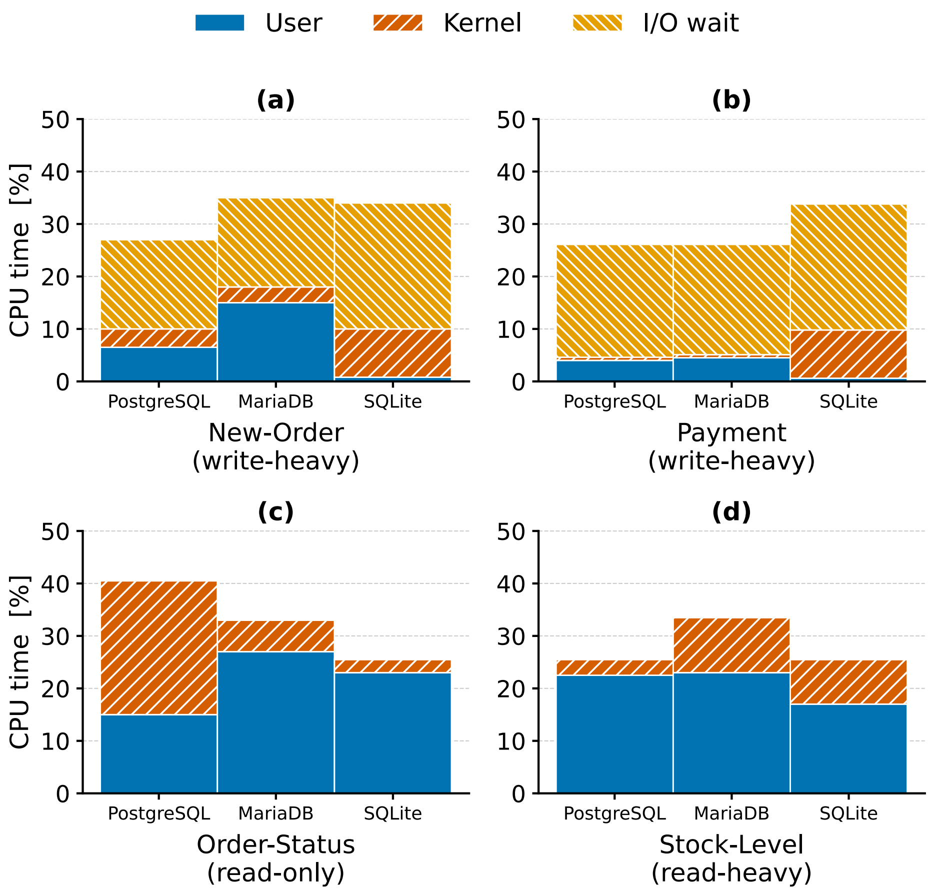
11.6x

eSQL.

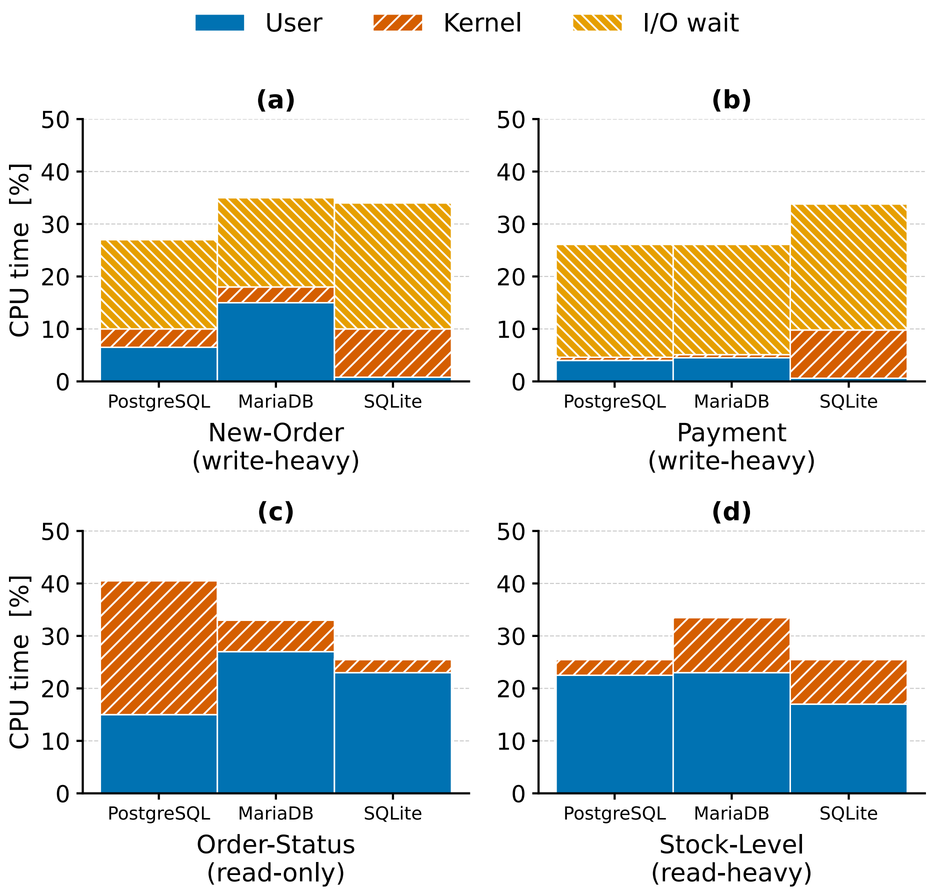
It syncs every single commit to disk.

The best engine changes with the workload.

Why Writes Cost More?

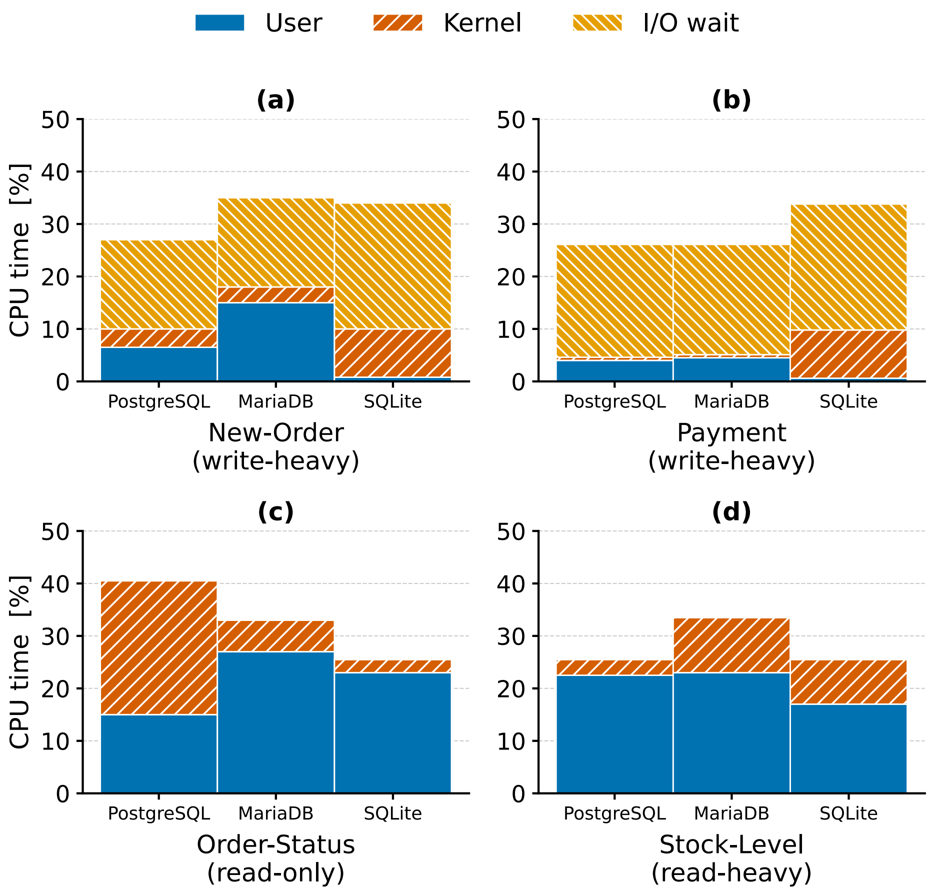


Why Writes Cost More?



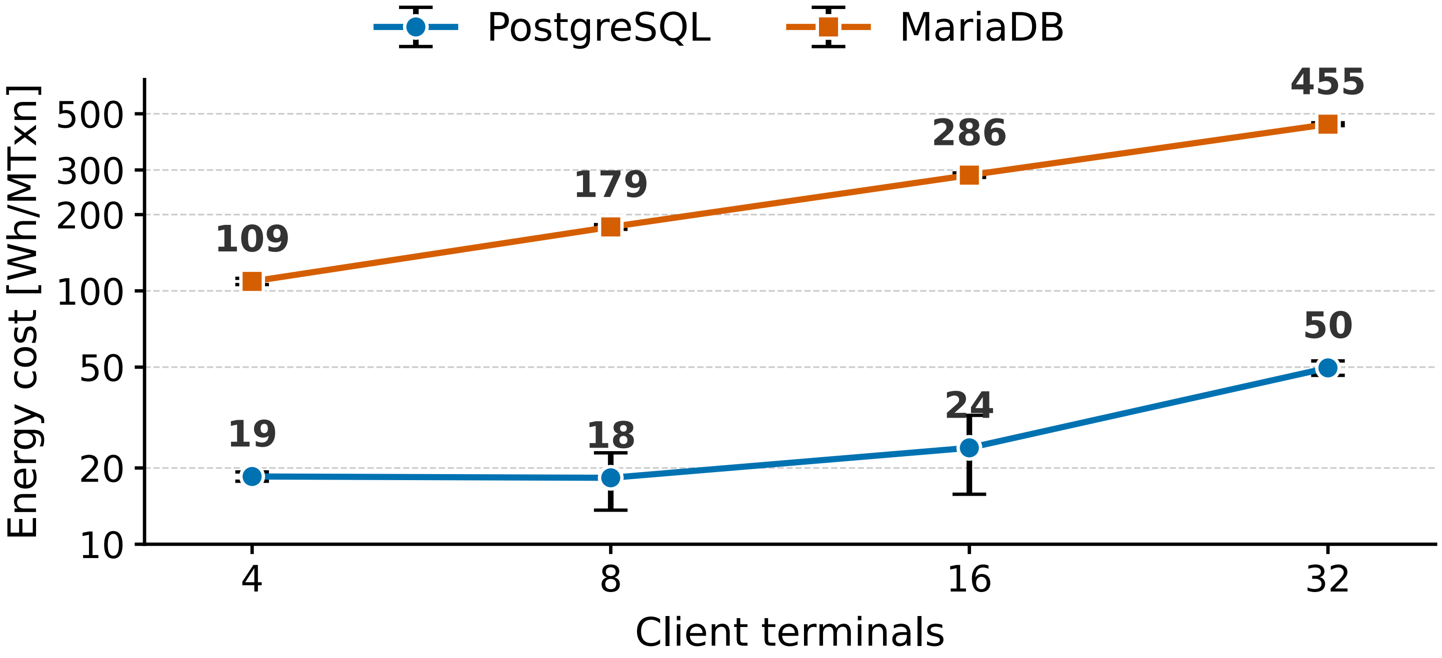
- On writes, **SQLite** spends up to **24%** of the time in I/O wait: CPU draws power while waiting for `fsync`, with under 1% useful computation.

Why Writes Cost More?



- On writes, **SQLite** spends up to **24%** of the time in I/O wait: CPU draws power while waiting for `fsync`, with under 1% useful computation.
- **PostgreSQL** and **MariaDB** batch many commits into one disk sync (group commit), reducing stalls.

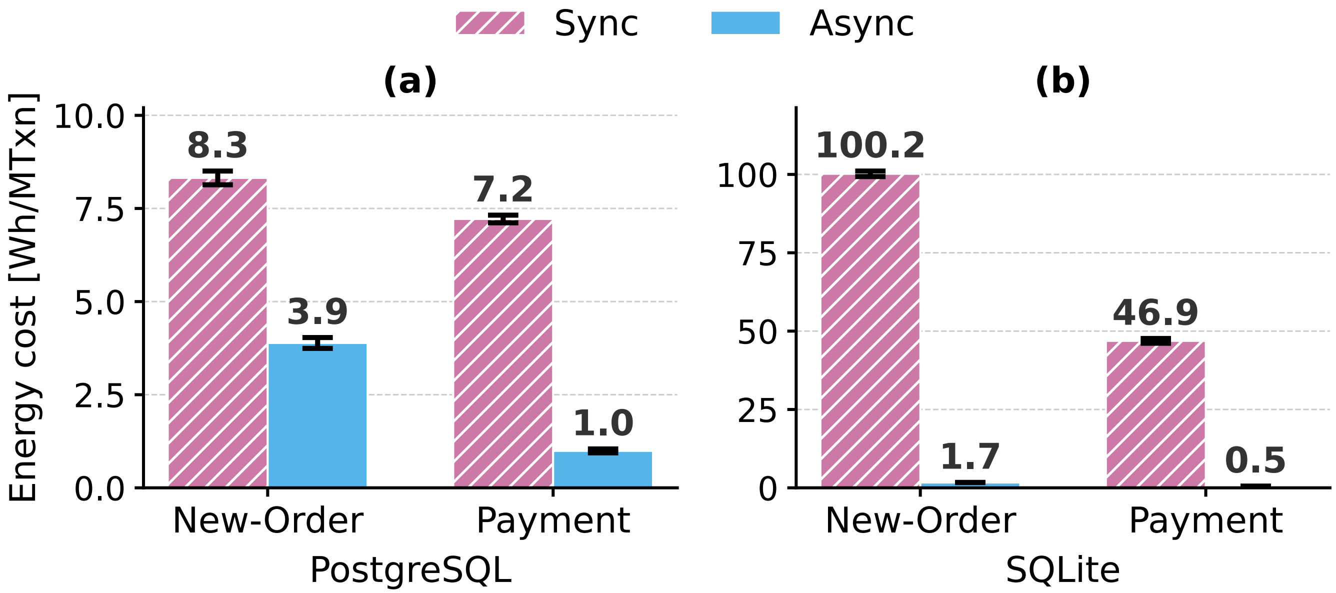
Does This Gap Hold Under Bigger Loads?



TPC-C, 64 warehouses, 4–32 terminals, server (cached disk)

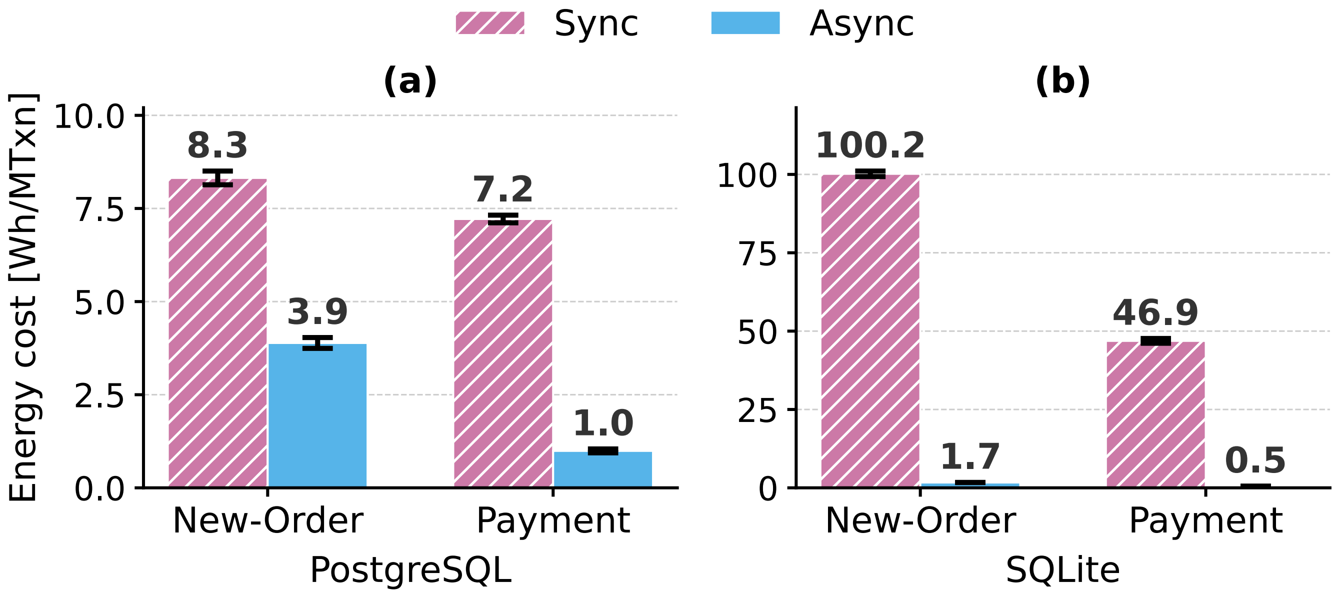
- With more users, **MariaDB** stays **6–9x** above **PostgreSQL**. The gap grows with concurrency.

Durability And Energy Trade-offs



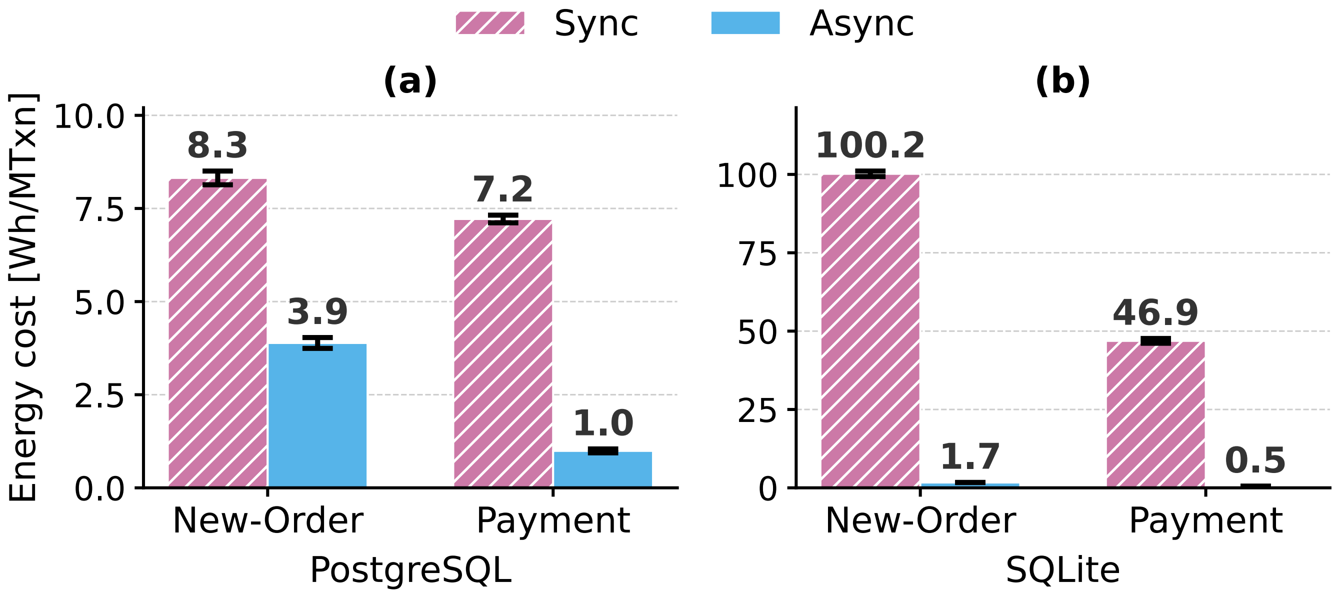
- **Synchronous commit:** transactions waits for `fsync`, durable on crash.

Durability And Energy Trade-offs



- **Synchronous commit:** transactions waits for `fsync`, durable on crash.
- **Asynchronous commit:** skip the `fsync`, recent commits may be lost.

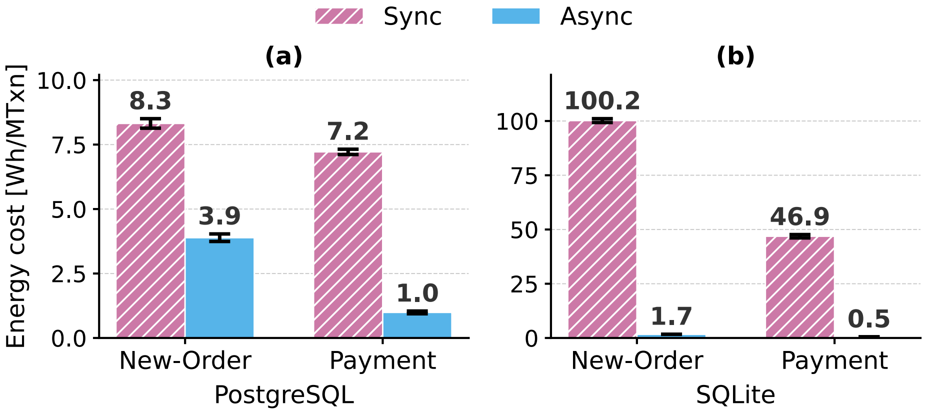
Durability And Energy Trade-offs



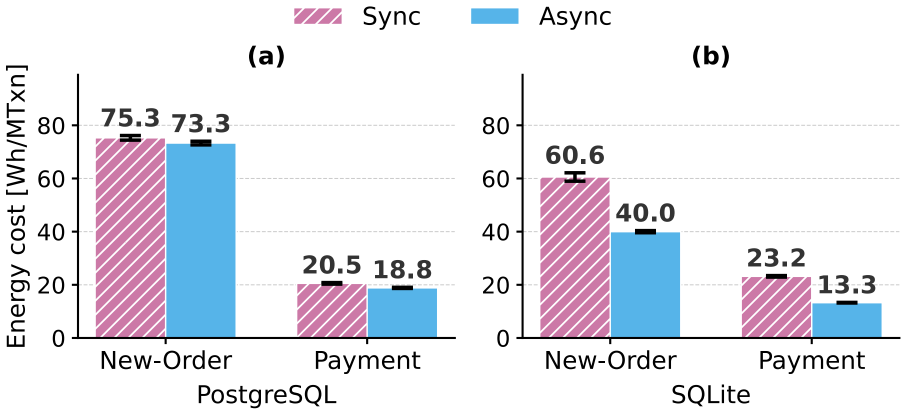
- **Synchronous commit:** transactions waits for `fsync`, durable on crash.
- **Asynchronous commit:** skip the `fsync`, recent commits may be lost.
- On a **slow disk**, disabling durability saves **2–90x** energy.

What If The Disk Is Cached?

Slow disk (HDD)

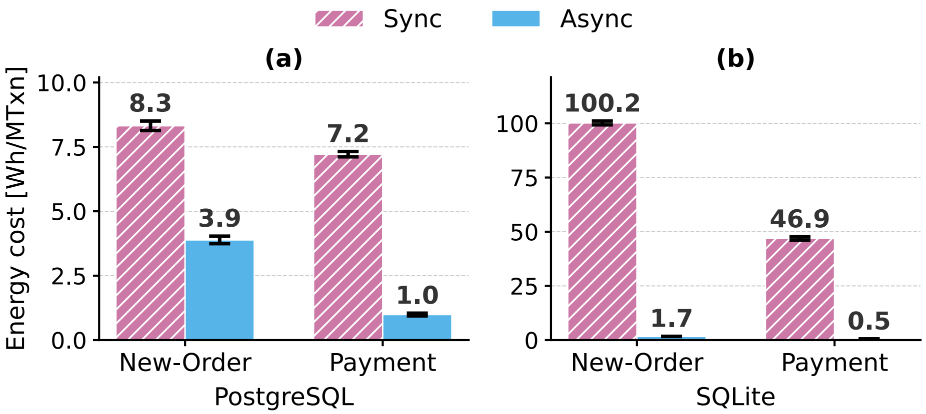


Cached disk (RAID write-back)

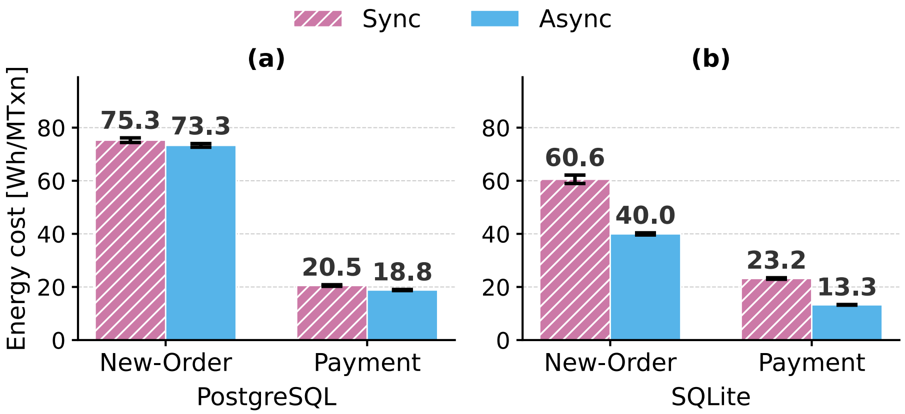


What If The Disk Is Cached?

Slow disk (HDD)



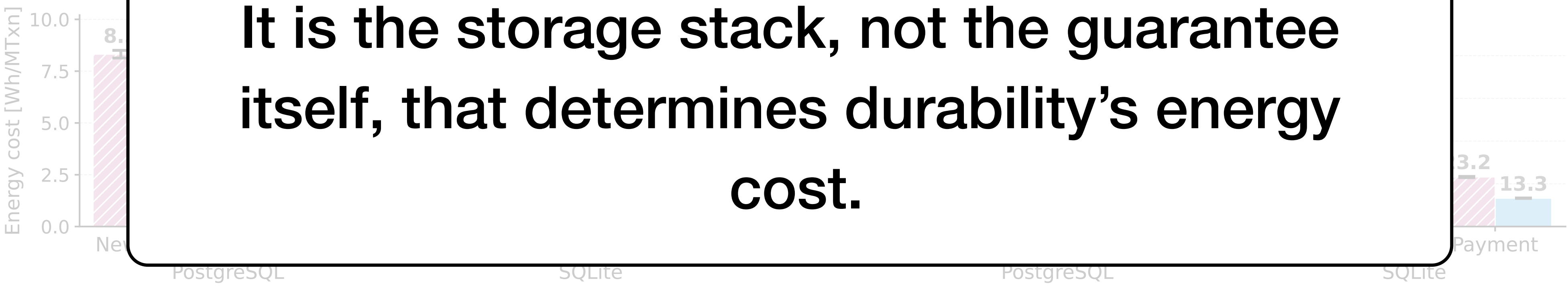
Cached disk (RAID write-back)



- On a **cached disk**, the same change saves only **1.5x**.

What If The Disk Is Cached?

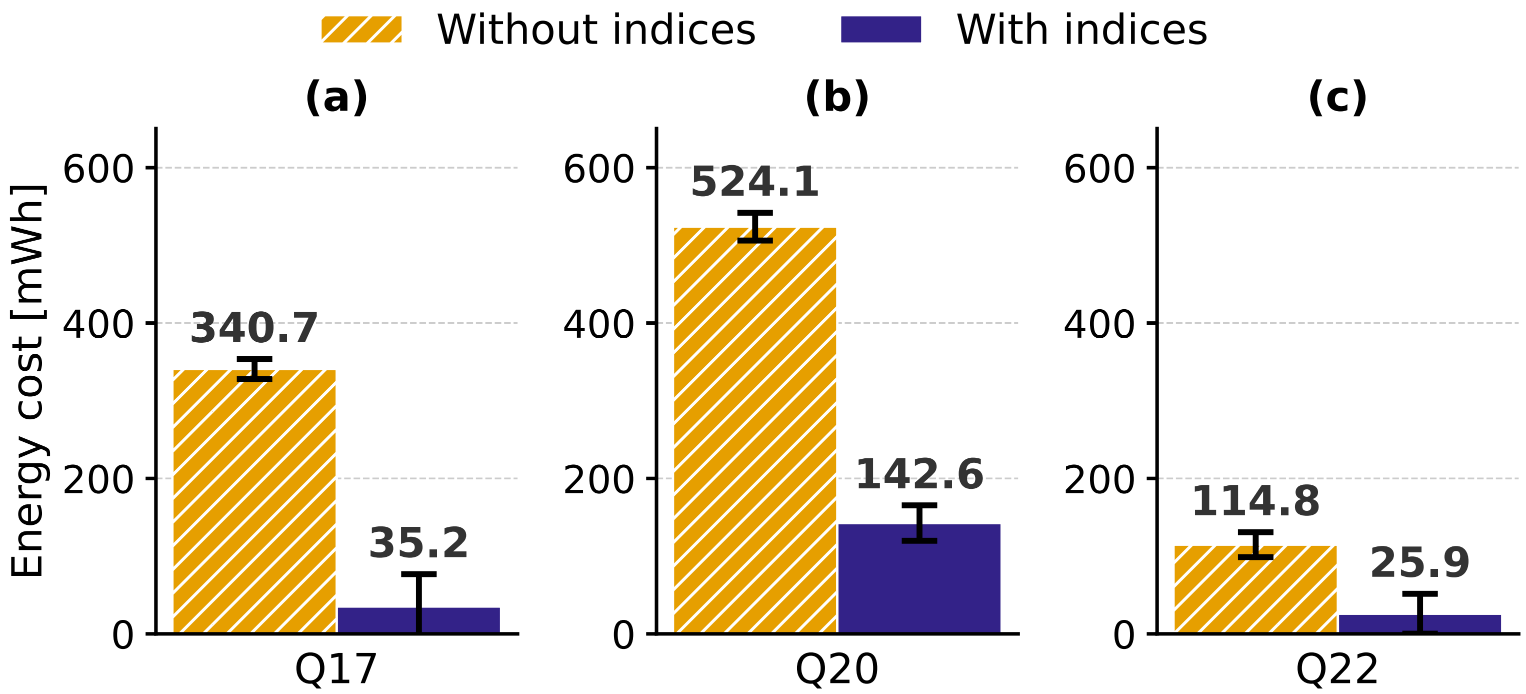
Slow disk (HDD) | Cached disk (RAID write-back)



It is the storage stack, not the guarantee itself, that determines durability's energy cost.

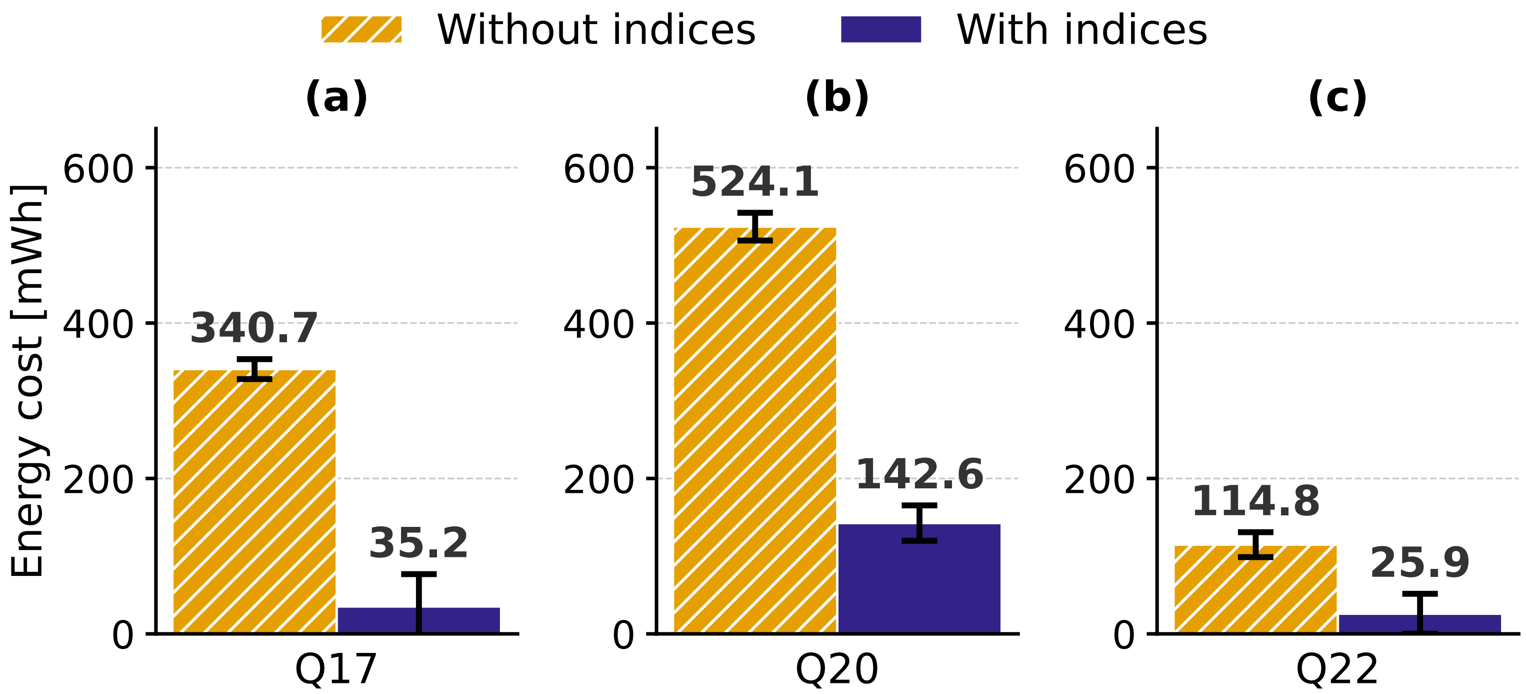
- On a **cached disk**, the same change saves only **1.5x**.

Indexing Vs. No Indexing



TPC-H on SQLite, desktop (HDD)

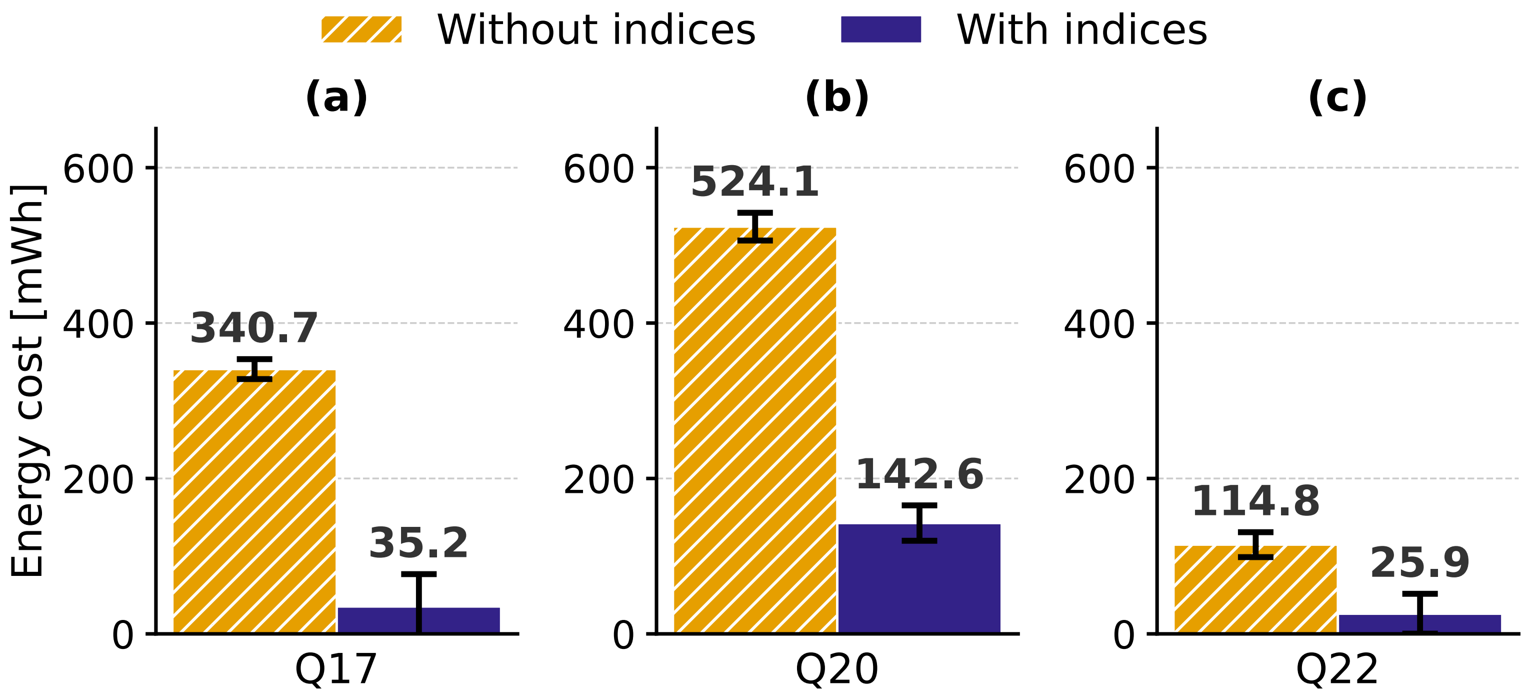
Indexing Vs. No Indexing



TPC-H on SQLite, desktop (HDD)

- Adding an index cuts **SQLite**'s energy up to **9.7x** on selective queries.

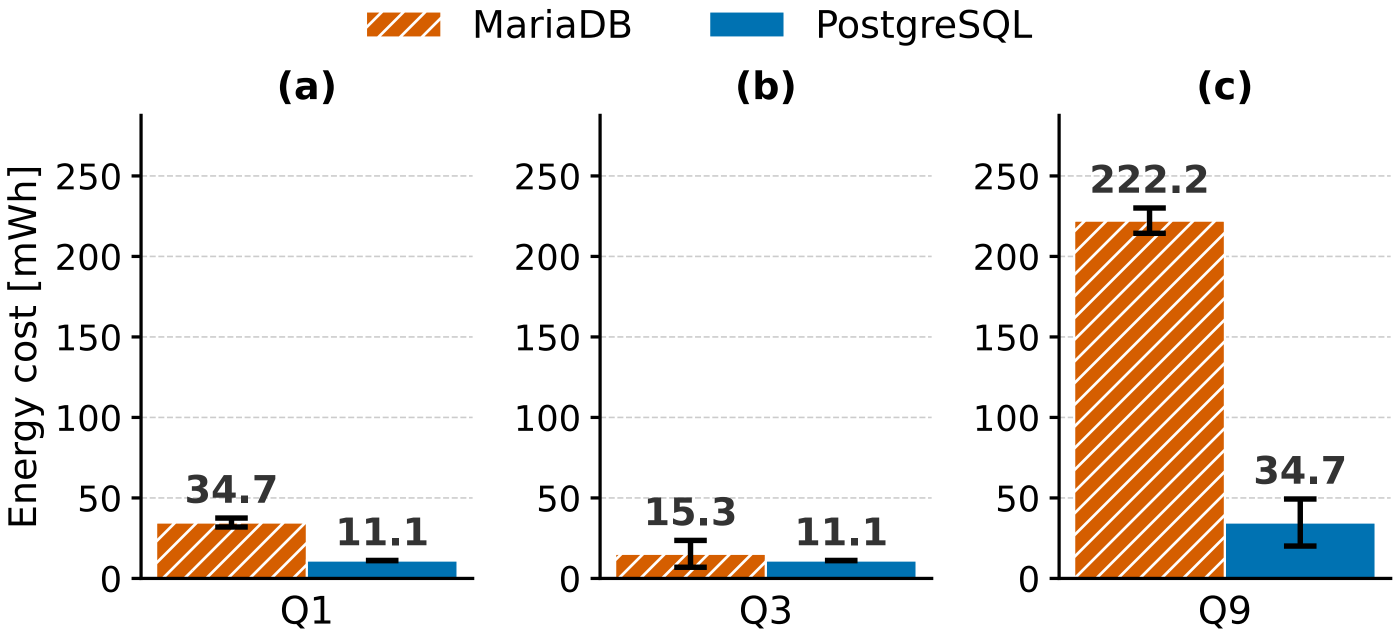
Indexing Vs. No Indexing



TPC-H on SQLite, desktop (HDD)

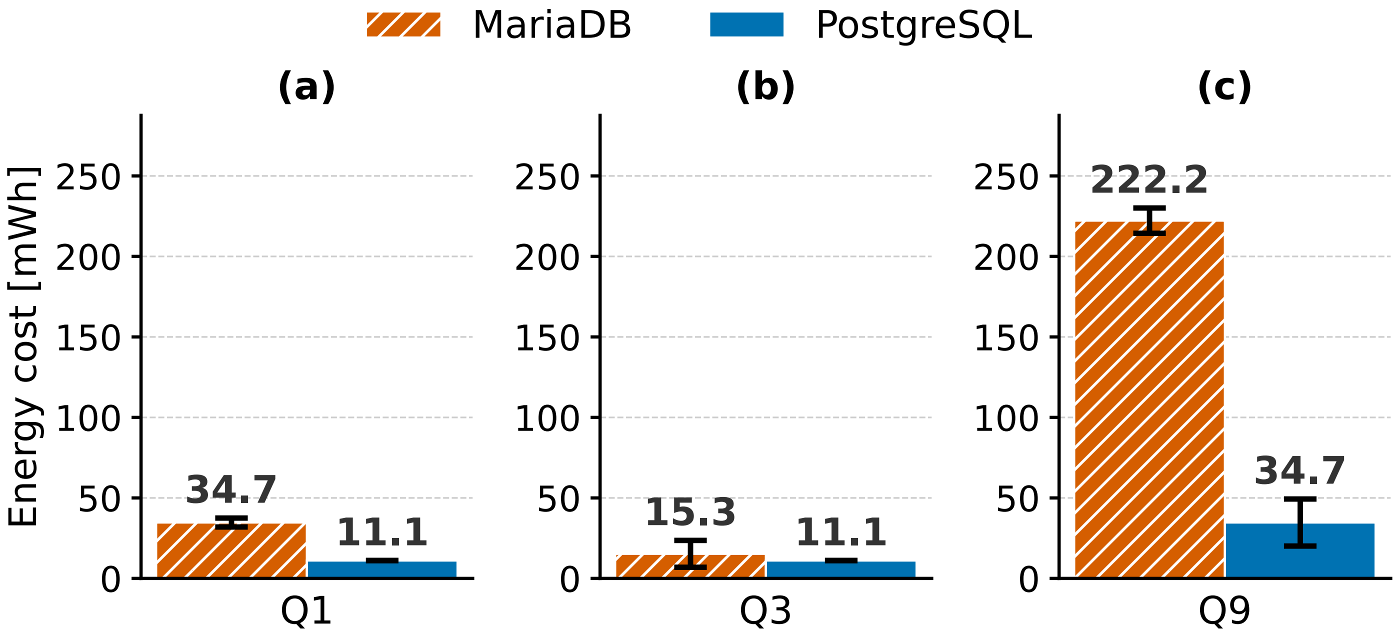
- Adding an index cuts **SQLite**'s energy up to **9.7x** on selective queries.
- Building all indices costs only 20% of the energy of one unindexed query.

Different Query Plans



TPC-H at SF 0.1, desktop (HDD)

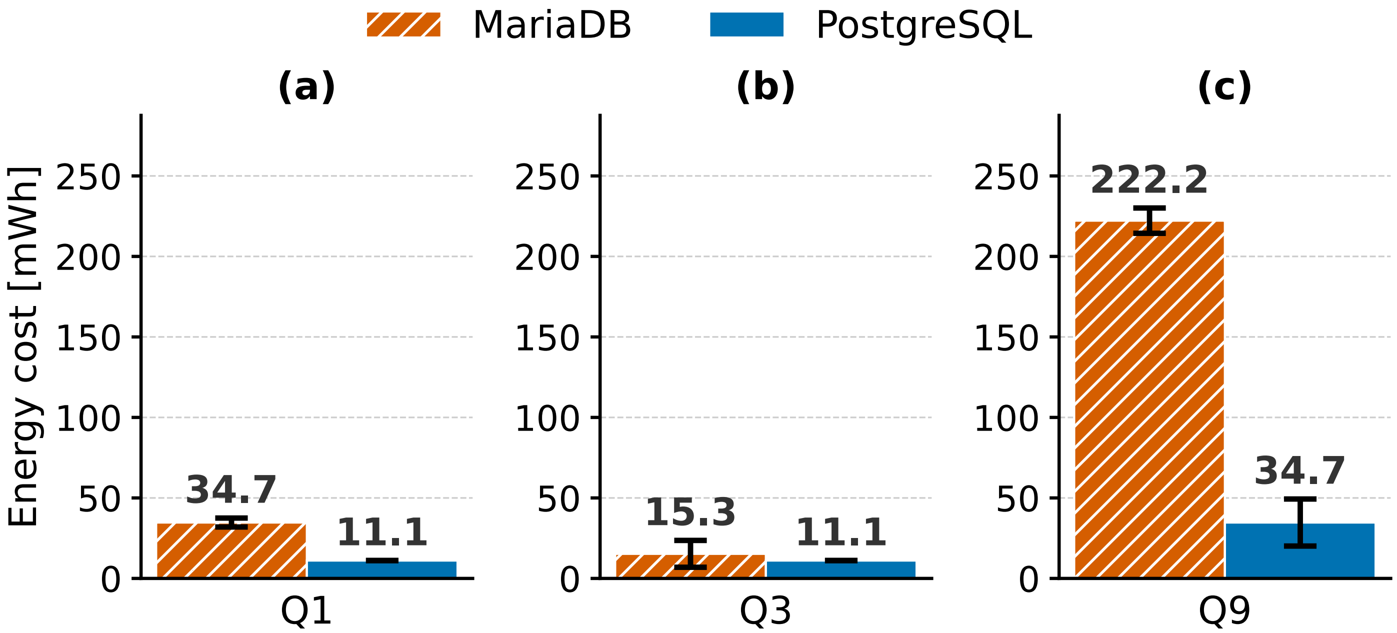
Different Query Plans



TPC-H at SF 0.1, desktop (HDD)

- Same query, same data, same hardware. **MariaDB**'s plan uses up to **6.4x** more energy than **PostgreSQL**'s.

Different Query Plans



TPC-H at SF 0.1, desktop (HDD)

- Same query, same data, same hardware. **MariaDB**'s plan uses up to **6.4x** more energy than **PostgreSQL**'s.
- At larger scale (SF 1), this gap grows up to **620x** on Q9.

Every gap is a software choice on the same hardware. All measurable today.

[2] Xu et al. 2013. [3] Radovanovic et al. 2021.

Every gap is a software choice on the same hardware. All measurable today.

Once energy per transaction is visible, we can act on it:

- **Energy-aware query planning:** add an energy term to the optimizer's cost model (prior work shows 11–22% savings[2]).

[2] Xu et al. 2013. [3] Radovanovic et al. 2021.

Every gap is a software choice on the same hardware. All measurable today.

Once energy per transaction is visible, we can act on it:

- **Energy-aware query planning:** add an energy term to the optimizer's cost model (prior work shows 11–22% savings[2]).
- **Carbon-aware scheduling:** defer or reroute **individual queries** to low-carbon hours[3].

[2] Xu et al. 2013. [3] Radovanovic et al. 2021.

Every gap is a software choice on the same hardware. All measurable today.

Once energy per transaction is visible, we can act on it:

- **Energy-aware query planning:** add an energy term to the optimizer's cost model (prior work shows 11–22% savings[2]).
- **Carbon-aware scheduling:** defer or reroute **individual queries** to low-carbon hours[3].
- **New hardware:** CXL-attached memory, NVMe — different stall profiles change which plan wins.

[2] Xu et al. 2013. [3] Radovanovic et al. 2021.

Conclusion

- On the **same machine**, four software knobs changed energy by **2–90x**: engine **11.6x**, durability **2–90x**, index **9.7x**, plan **6.4x**.
- **Default configurations** are not the most energy efficient.
- **Report Wh/Txn alongside throughput and latency** to make these differences actionable.

Databases Are Blind to Energy

Tobias Rahn (ELCA), **Andres Navarro Pedregal*** (ETH Zurich), Michal Friedman (ETH Zurich)

*anavarro@ethz.ch

Full paper