



DBtune: AI-driven performance for all PostgreSQL flavors

PGConf EU

October 21, 2025



Dr. Luigi Nardi

Founder & CEO, DBtune

About me

Mixed background in industry and academia

Among other things:

Ph.D. CS at Sorbonne, Research Staff at Stanford, and Associate Professor in AI at Lund

Since 2020, Founder & CEO at DBtune



DBtune in the top 20 PostgreSQL sponsors



<https://speakerdeck.com/clairegiordano/whats-in-a-postgres-major-release-an-analysis-of-contributions-in-the-v17-timeframe-claire-giordano-pgconf-eu-2024>

Malmö PostgreSQL User Group (M-PUG)

M-PUG organizers



Ellyne Phneah
DBtune



Dr. Luigi Nardi
DBtune



Daniel Gustafsson
Microsoft



Dennis Rilorin
Redpill Linpro



- The group is officially recognized by PostgreSQL Europe
- Regular meetups every 4-8 weeks in Malmö — Top speakers
- We are building a vibrant PostgreSQL community in the region

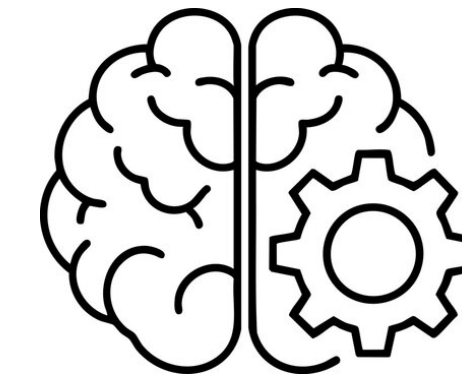


dbtune



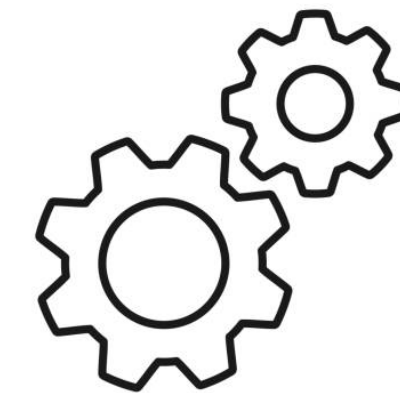
What

DBtune is an AI-powered database tuning service



Where

Spun out of research at Stanford University



How

Tunes for a specific workload, use case and machine

What is database performance tuning?

What is database tuning?

Keeping the database fit and responsive

- ✓ Databases change, grow and slow down
- ✓ Not all workloads and machines are the same
- ✓ **Tuning adapts a database to its current use-case, load and machine**
- ✓ It is a 'dark-art' yet an integral part of any DBA and developer's job
- ✓ Tuning includes query, **server parameters***, index, OS parameter, etc.

*DBtune focuses solely on automating PostgreSQL parameter tuning

Why does it matter?

Technical perspective

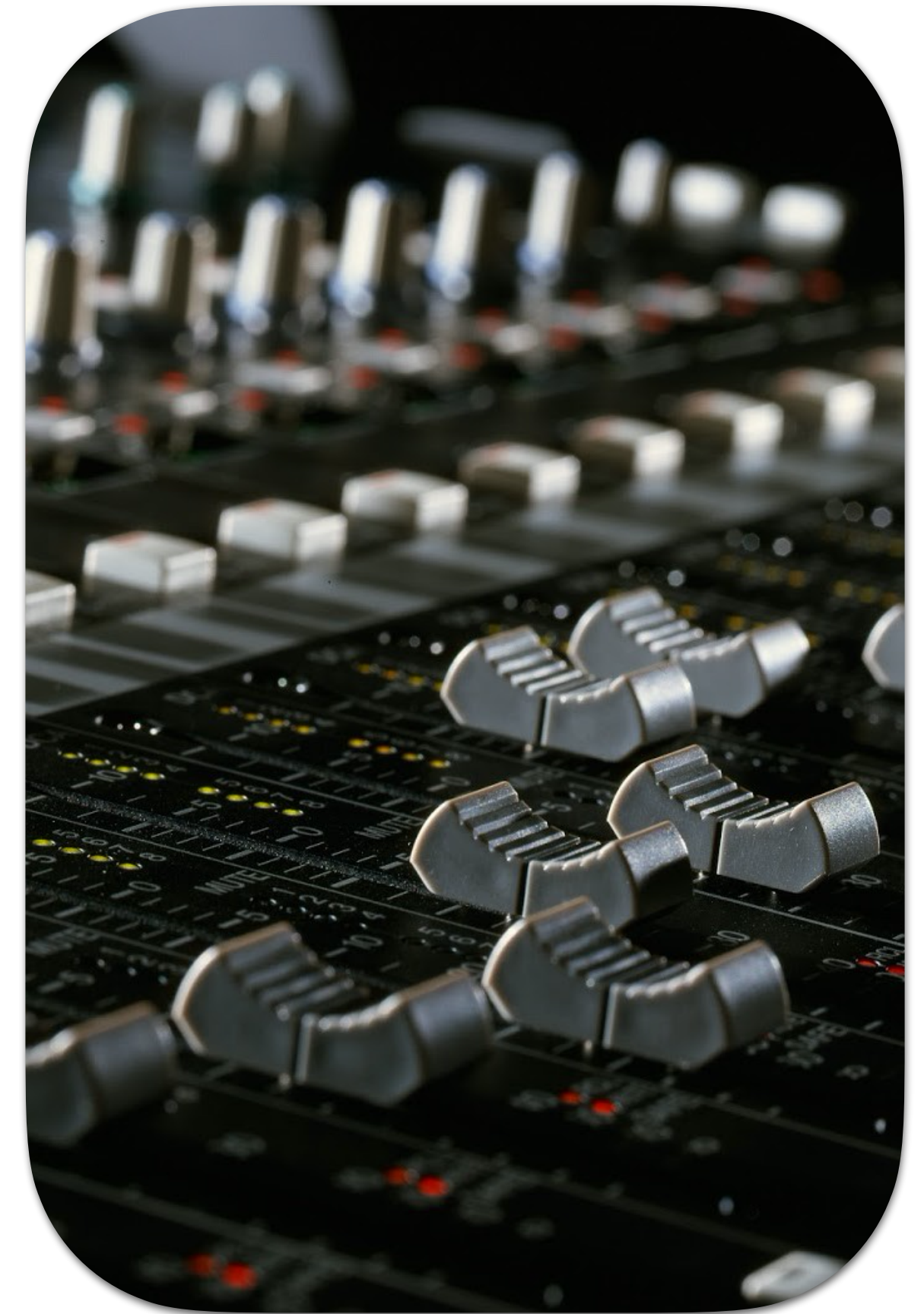
- Impacts system performance
 - Throughput and latency
- Improves scalability / stability / SLA

Business perspective

- Higher end-user satisfaction
- Optimizes infrastructure spend
- Reduces downtime
- Increases productivity
- Saves energy (ESG)

Database system parameter tuning

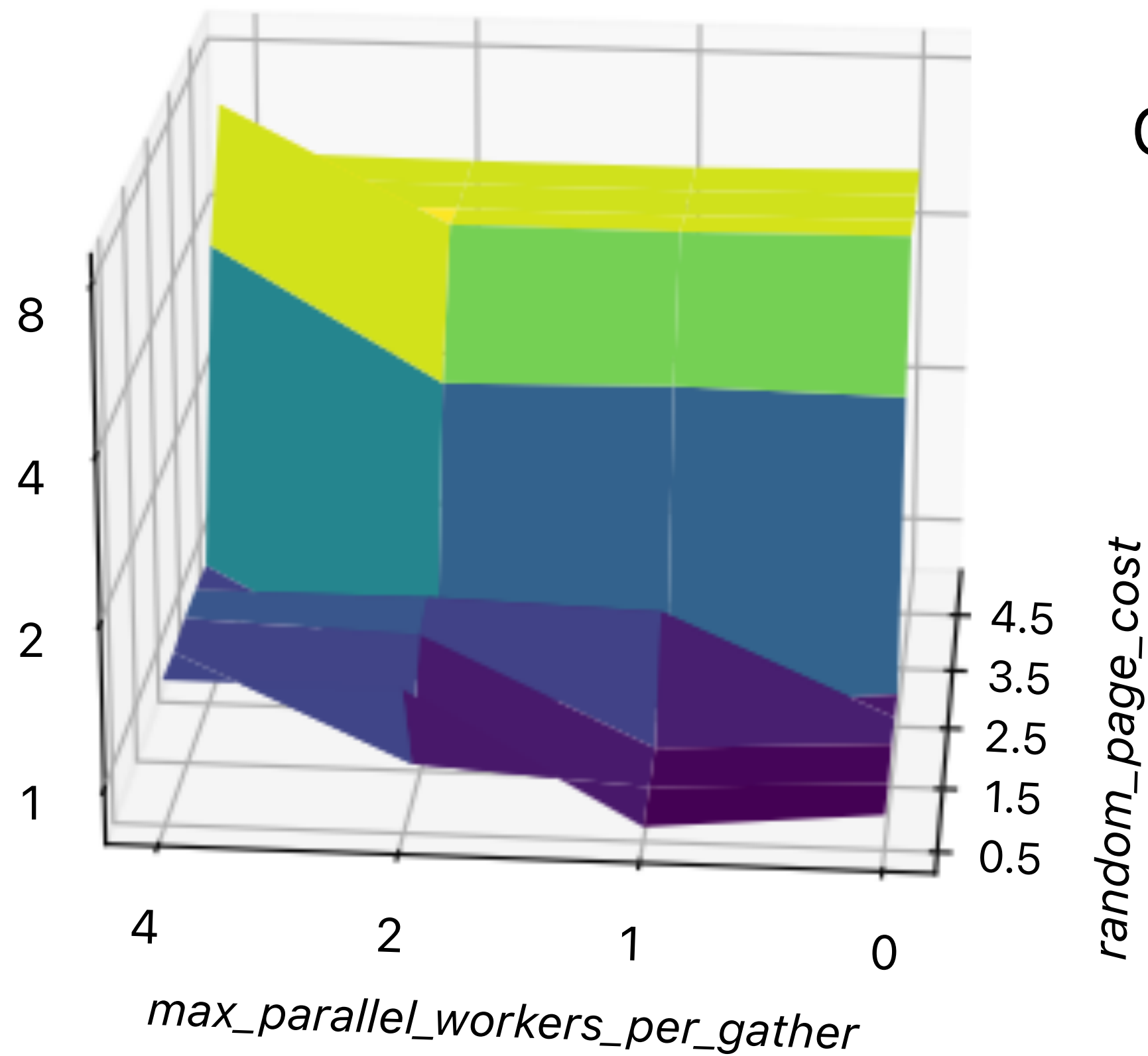
- ✓ Adjusting knobs to best fit the workload
- ✓ PostgreSQL parameters that are typically important: *work_mem*, *shared_buffers*, *max_wal_size*, etc.
- ✓ Example *max_parallel_workers_per_gather*:
Max # of workers started by a Gather or Gather Merge node
- ✓ Example *random_page_cost*:
Planner's cost of a non-sequentially fetched disk page
- ✓ These parameters highly depend on the application



Average query runtime tuning

for *max_parallel_workers_per_gather* and *random_page_cost*

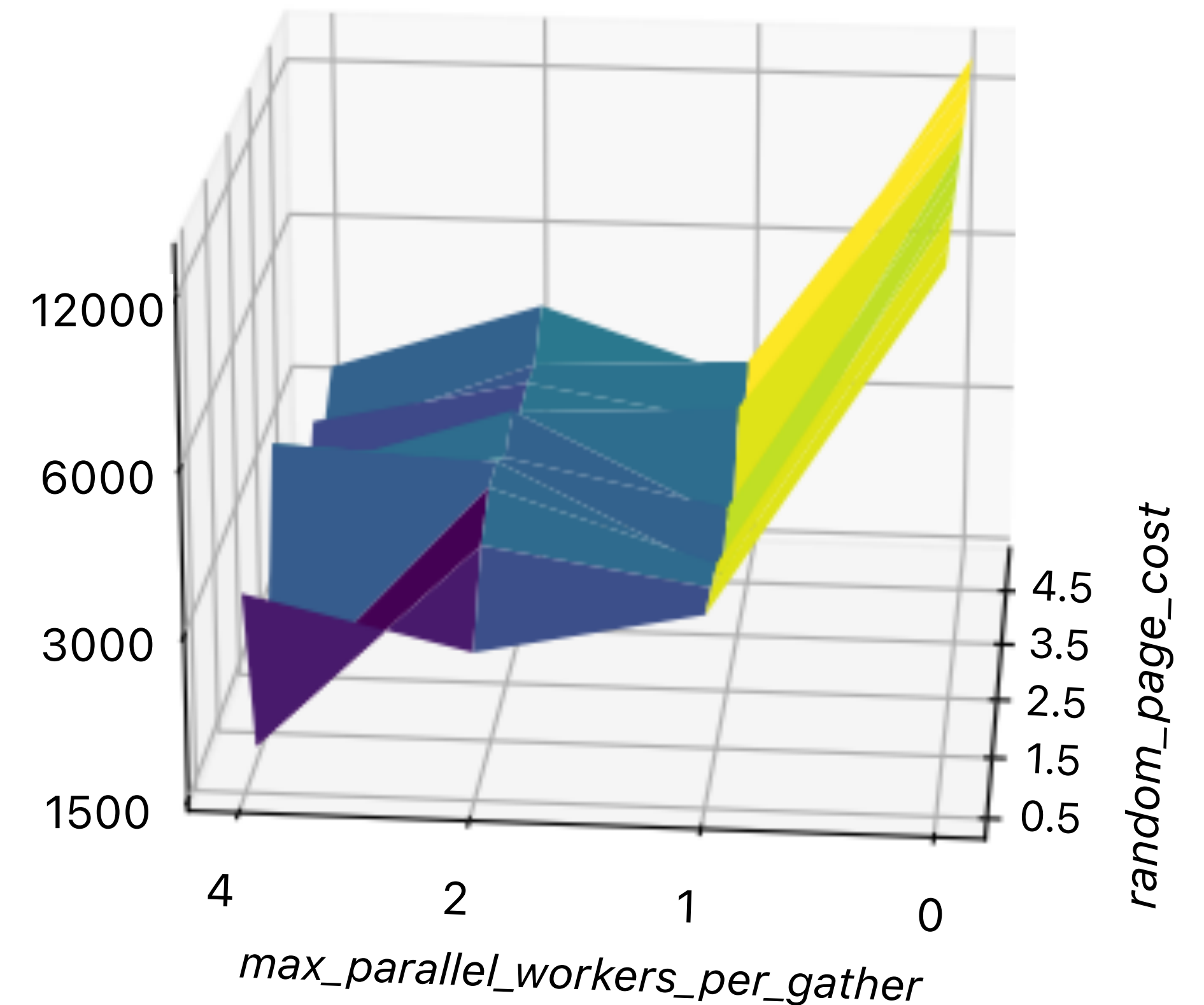
Epinions



Query runtime in ms
Lower the better

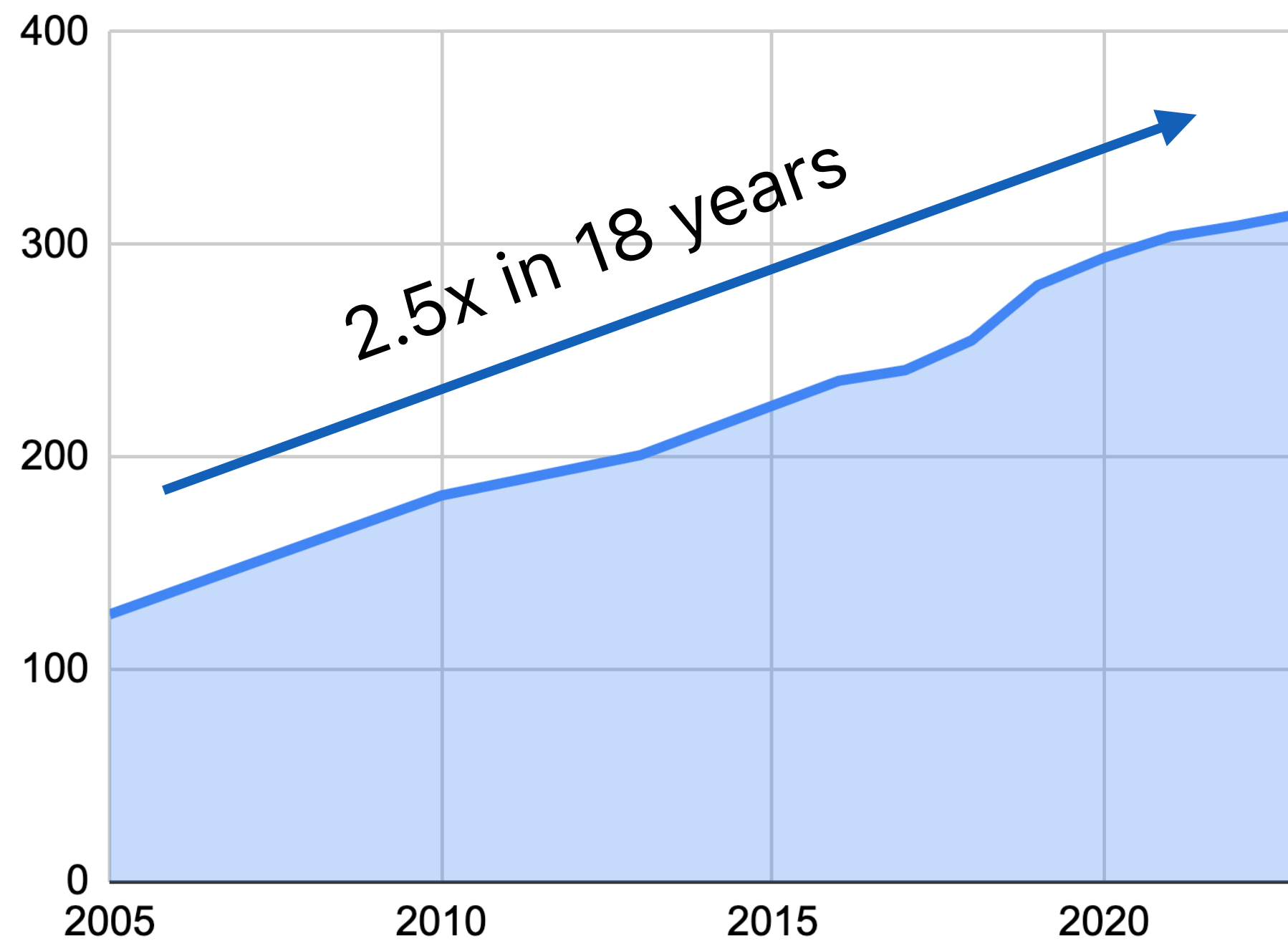


TPC-H



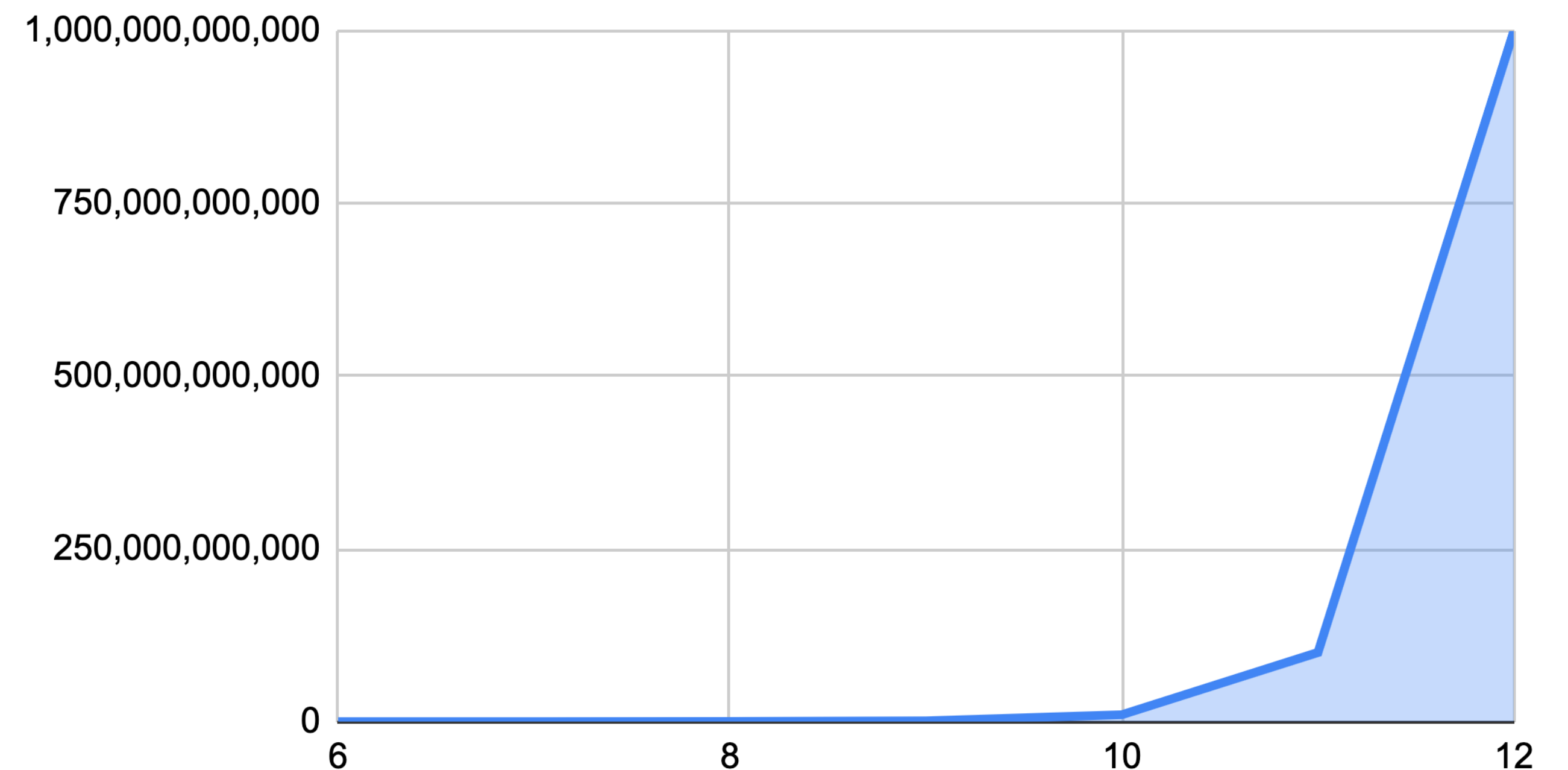
Complexity is growing over time

The number of parameters
is growing **linearly**



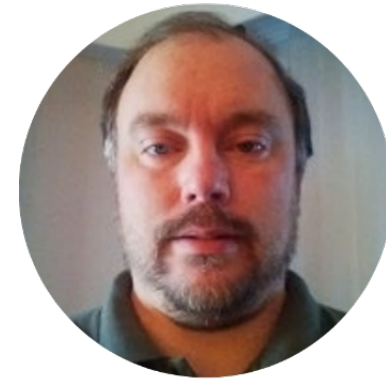
PostgreSQL number of parameters

The number of configurations
is growing **exponentially**



Example of complexity with 12 parameters

How is parameter tuning tackled today by DBAs and developers?



Tuning
guru

Manual

Slow

Takes days

Painstaking

Needs high expertise

Ineffective

Tune again in a week

Inadequate

Seasonal workload

Heuristics

One-size-fits-all

Uses generic rules

Workload agnostic

Not bespoke

Ineffective

Tune again in a week

Inadequate

Seasonal workload



AI approach

Ideally a solution that **learns** by **observation** and **autotunes**

A solution that **adapts** to changing workloads

How often do you tune?

Frequent

- ✓ Your workload changes — Change queries and application
- ✓ Your database grows
- ✓ You scale your cloud instance — Up or down
- ✓ When the data distribution changes it affects the query planner

Infrequent

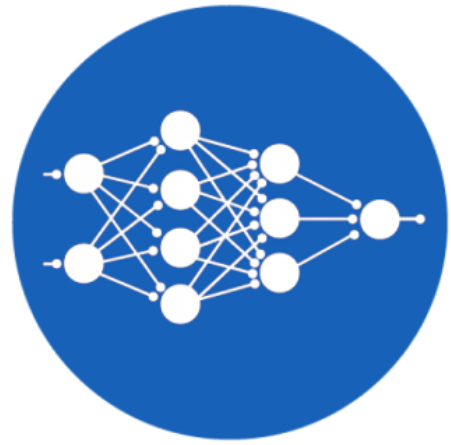
- ✓ You migrate from on-prem to the cloud — Or vice-versa
- ✓ You migrate DBMS — E.g., from Oracle to PostgreSQL
- ✓ You upgrade your version of PostgreSQL

The reality of how most enterprises treat manual parameter tuning today

- ✓ Tuning is typically **reactive** to something going wrong — Not **proactive**
- ✓ Maybe looked at once or twice a year
- ✓ Often engage expensive external resources / experts
- ✓ Different workloads are not treated differently
- ✓ Modus operandi: Throw more hardware / compute at any issue (\$\$\$)

We introduce DBtune

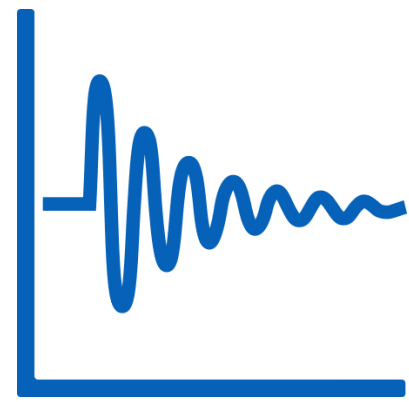
A unique AI-powered database tuning cloud hosted service



Machine learning approach



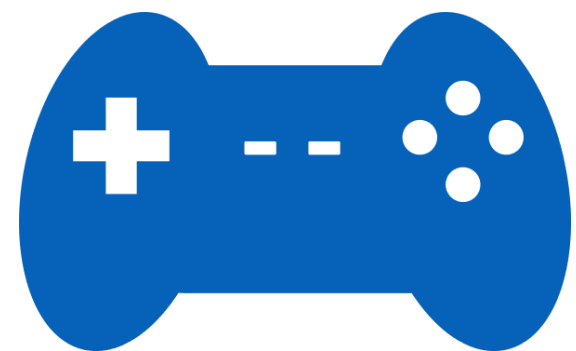
DBtune learns how to solve optimization challenges



Dynamic adaptation



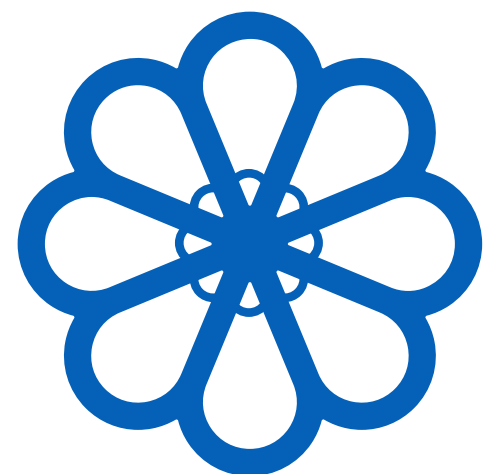
DBtune can tune a database irrespective of its size and complexity



Easy to use



No need for background in AI or database tuning



Highly scaleable

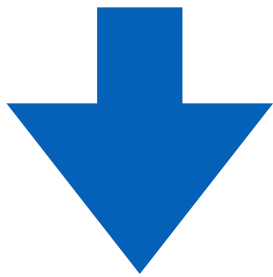


DBtune can tune multiple databases in heterogeneous environments

The ROI that our customers can expect using DBtune



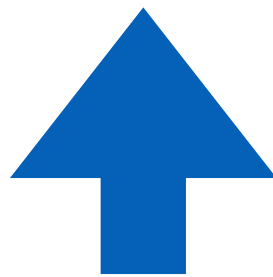
Database spend



Up to 50%



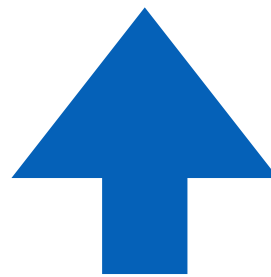
Faster application



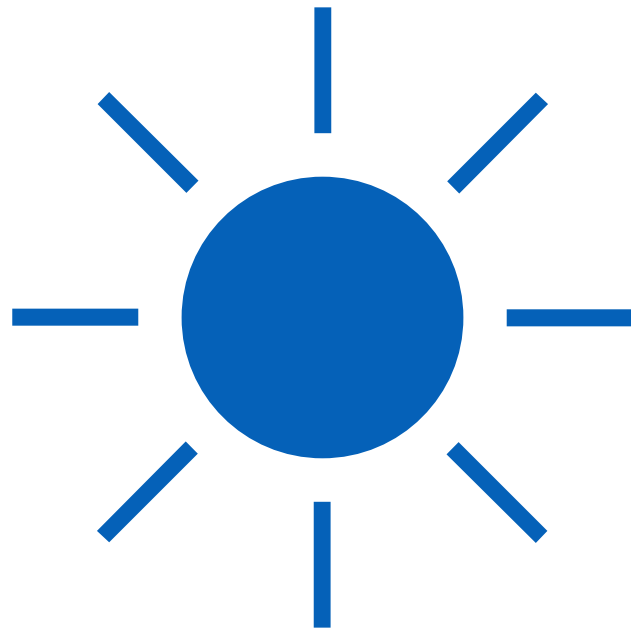
Up to 10x



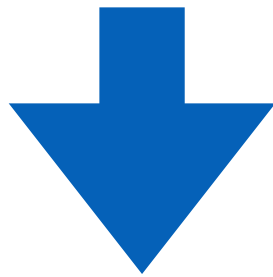
Increase productivity



Up to 25%



Reduction in CO2

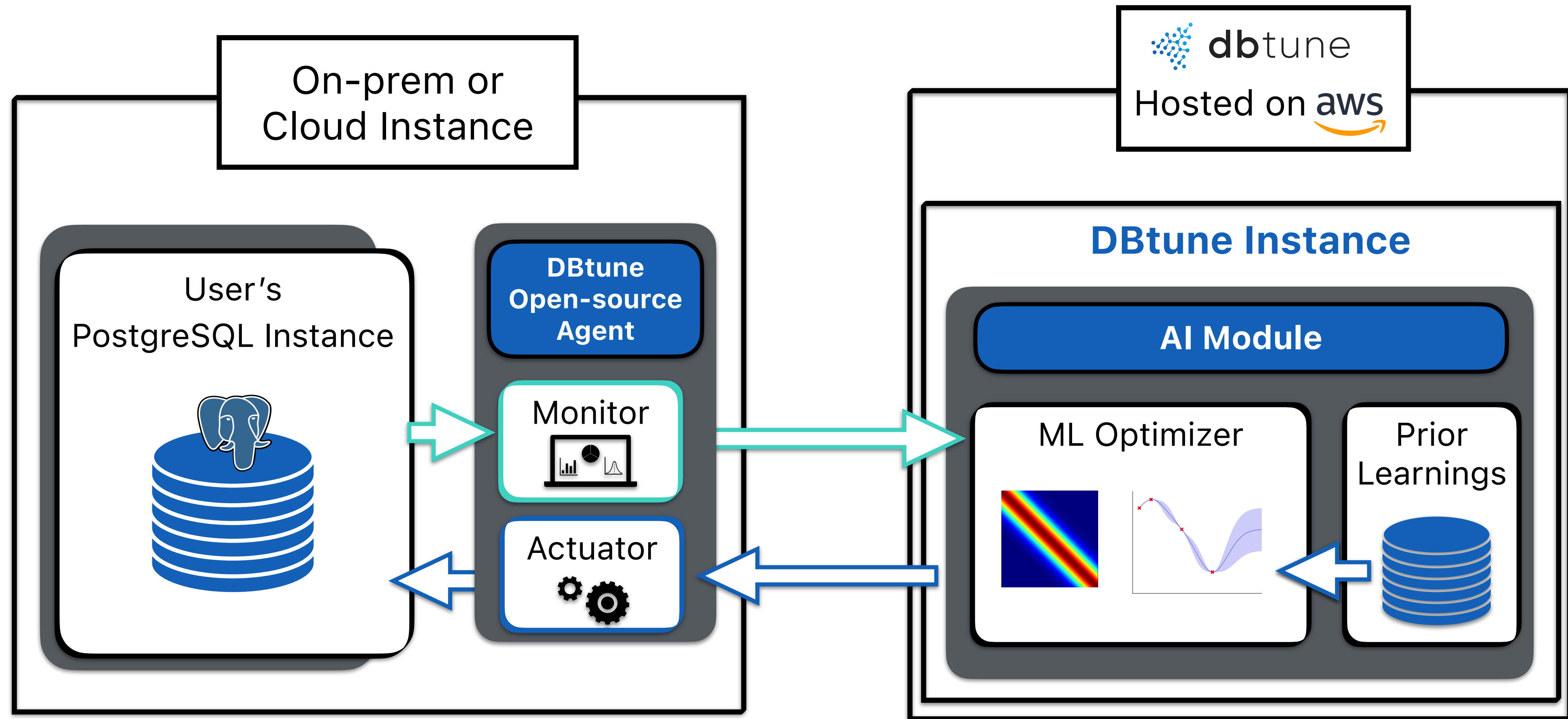


Up to 50%

Business outcomes using DBtune

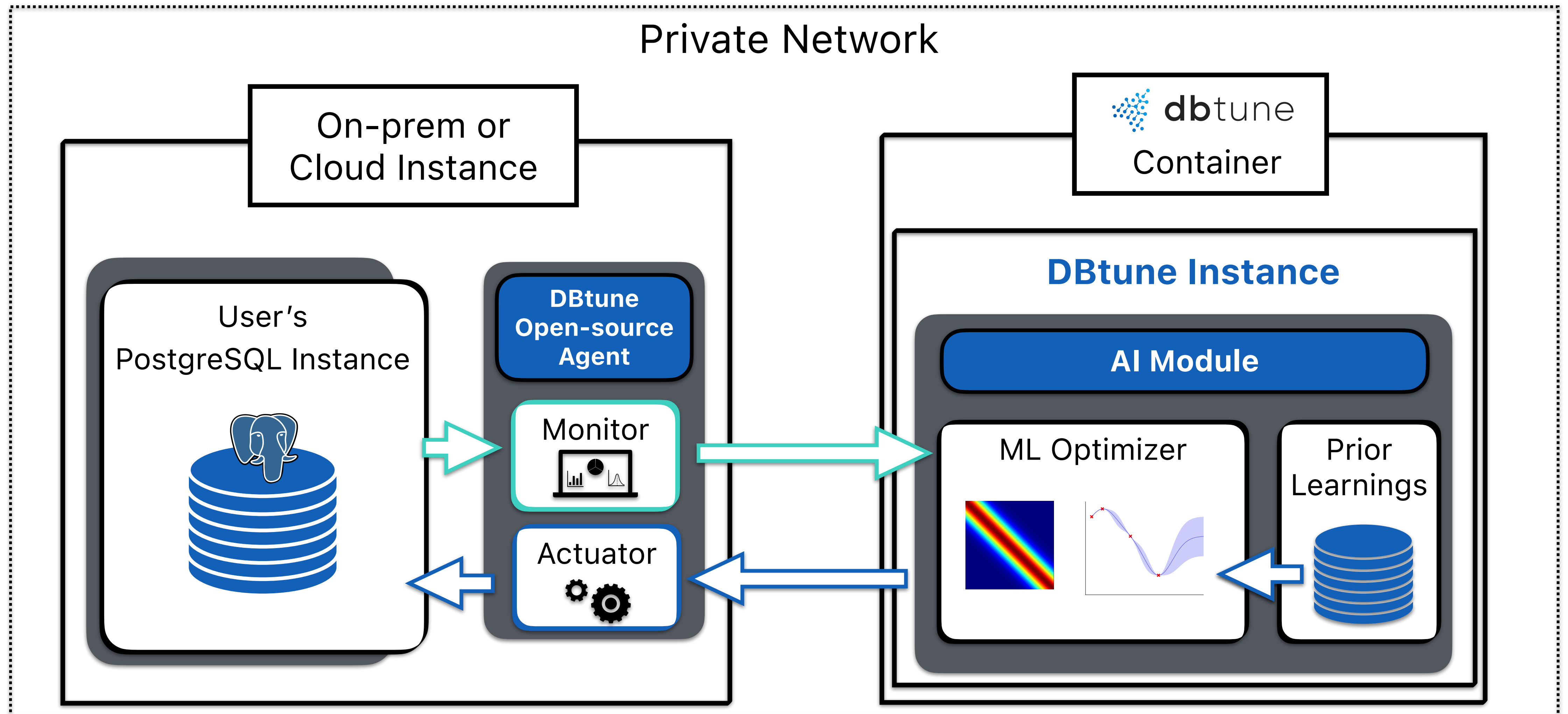
DBtune architecture for self-managed PostgreSQL

High-level view

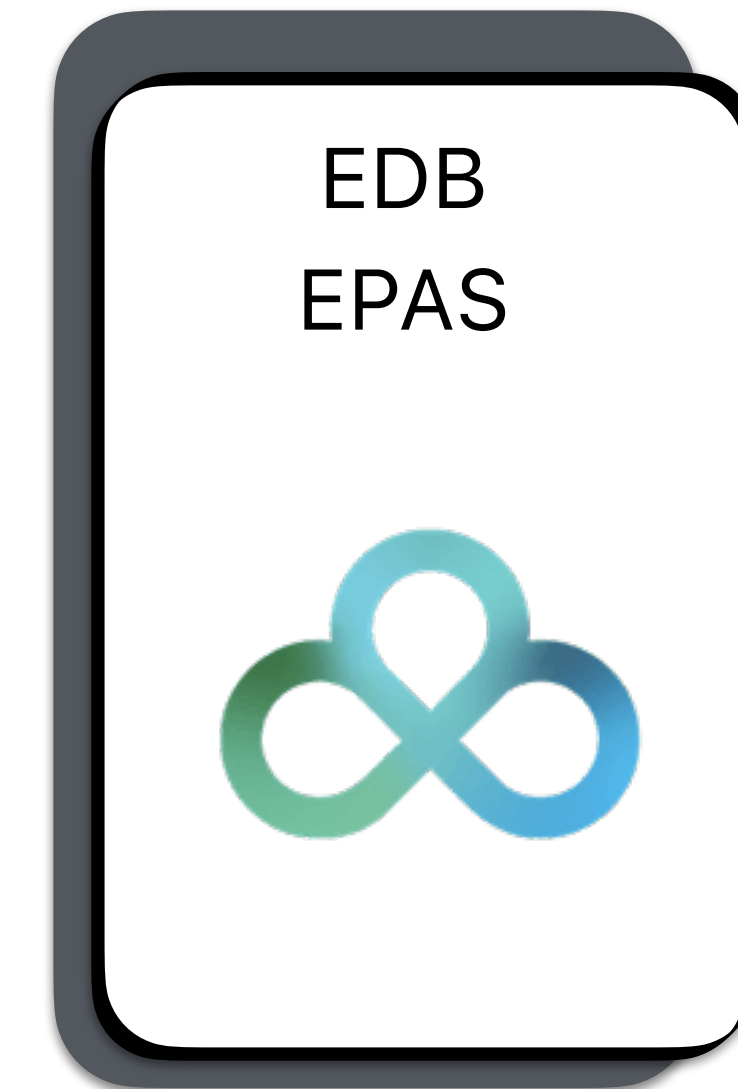
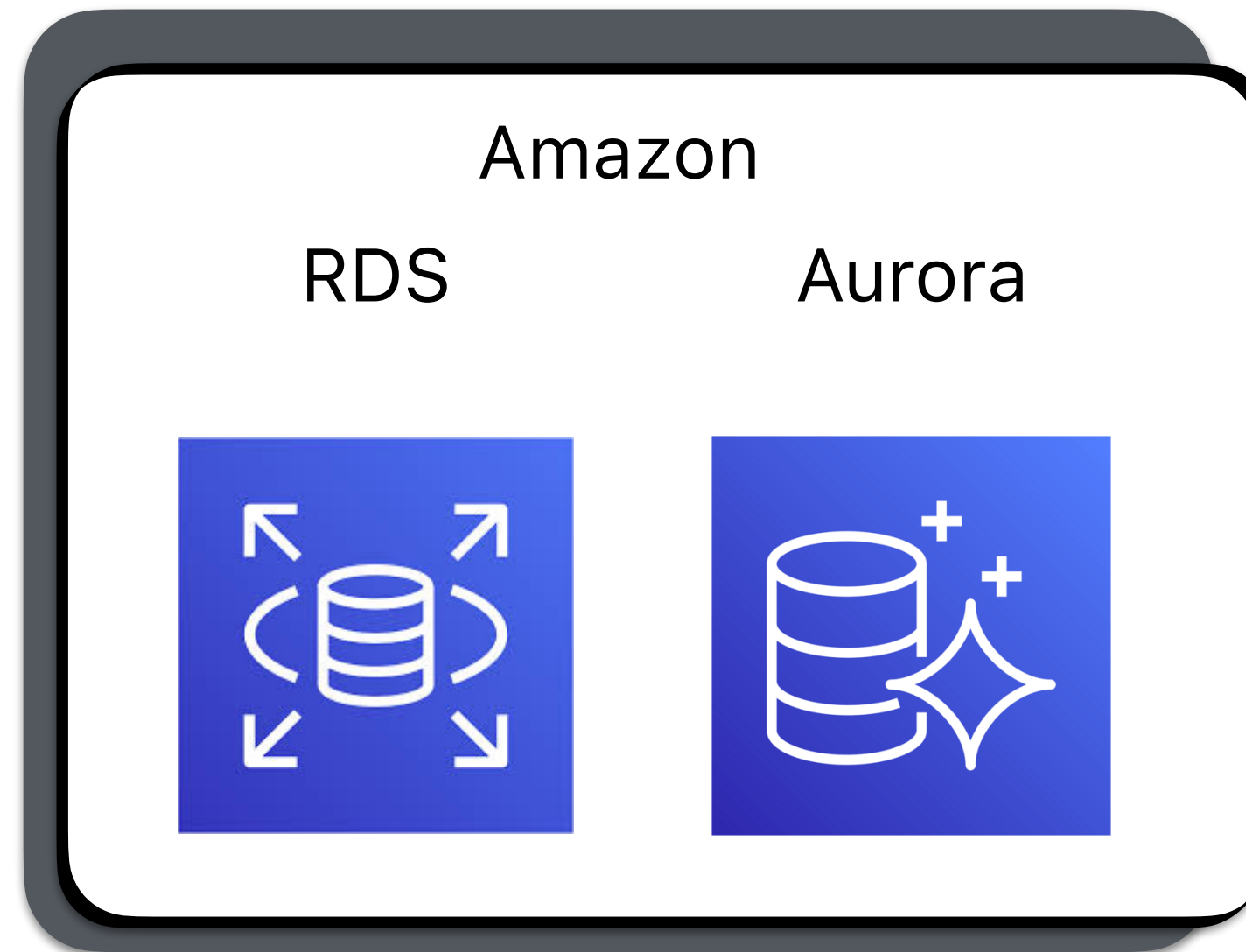


DBtune architecture for instances that are offline

High-level view

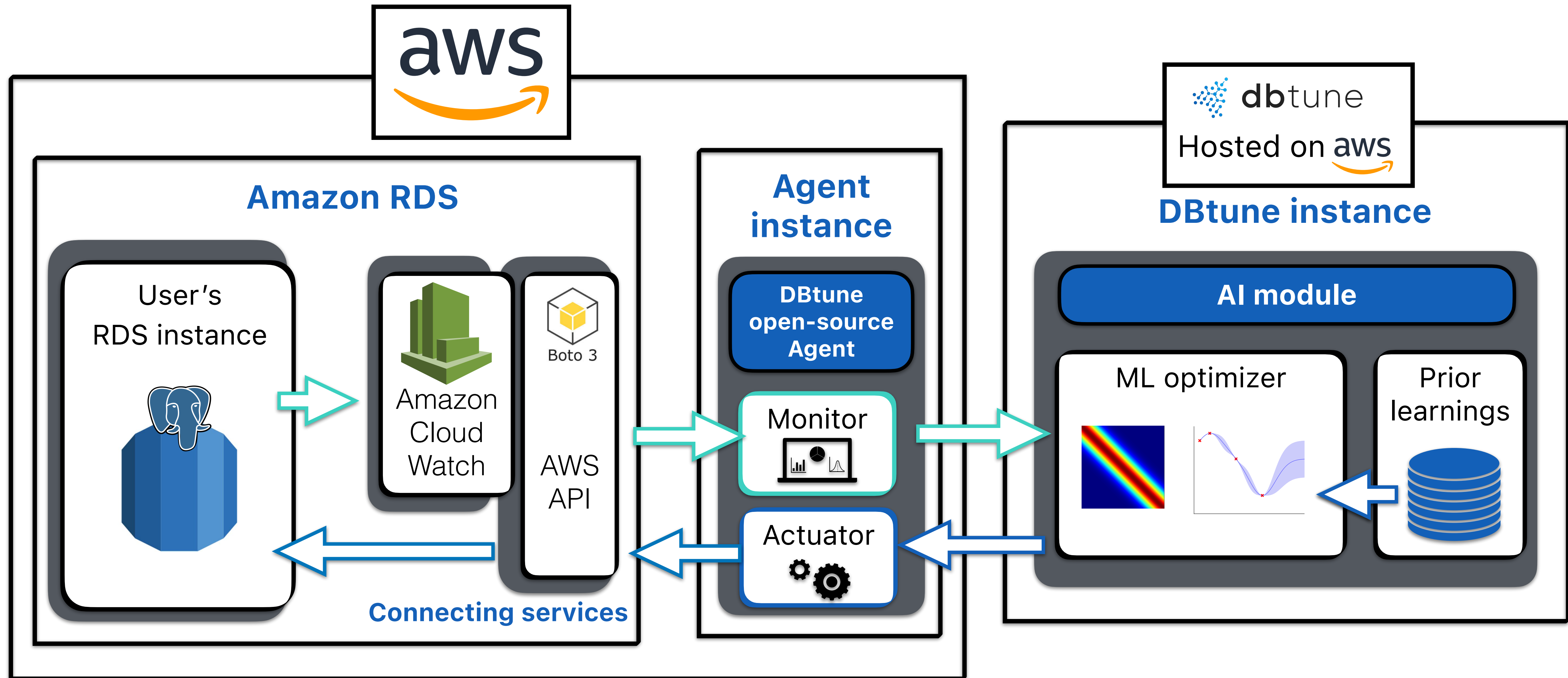


AI-driven performance for all PostgreSQL flavors



DBtune architecture for Database As a Service (DBaaS)

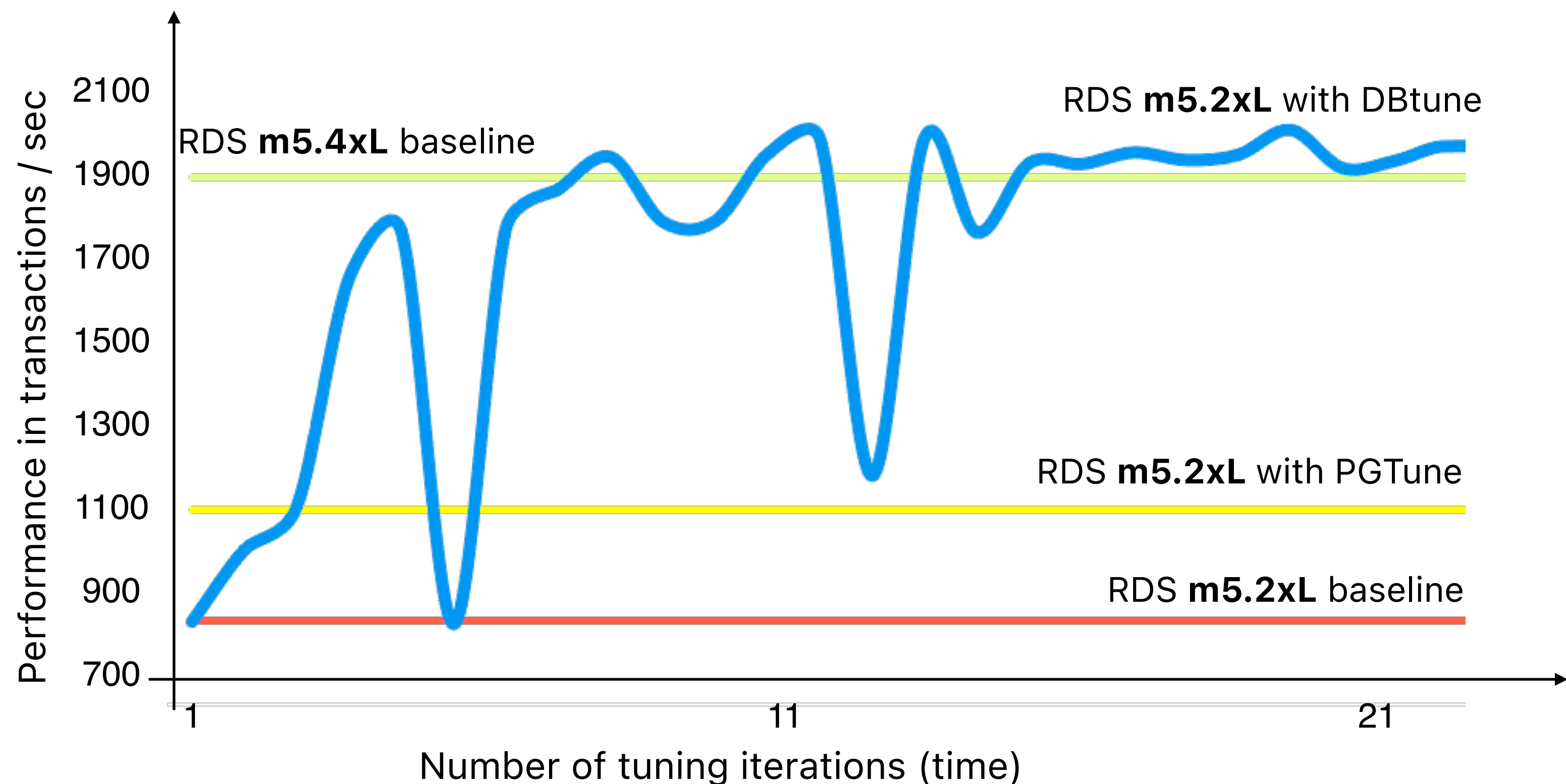
High-level view RDS PostgreSQL/Aurora



Proof of cost reduction: Comprehensive test results

DBtune doubles the performance of PostgreSQL Amazon RDS

Performance impact of tuning RDS m5.2xLarge cloud instance on the TPCCC benchmark



DBtune on the smaller instance type achieves a level performance in excess of that achieved by an instance twice the size

Proof of cost reduction: Detailed cost analysis

DBtune doubles the performance of PostgreSQL Amazon RDS

Hardware				Cost / Year		
AWS RDS Instance Type	Cores	RAM	IOPS	Instance	EBS	Total
db.m5.4xlarge	8	64 GBs	4000	\$12,475	\$4,800	\$17,275
db.m5.2xlarge	4	32 GBs	2000	\$6,237	\$2,400	\$8,637

Per instance savings: \$8,638

- ✔ DBtune halves RDS cost (50% saving)
- ✔ Matches 4xLarge performance on a 2xLarge instance
- ✔ Medium and large companies use hundreds* of RDS instances

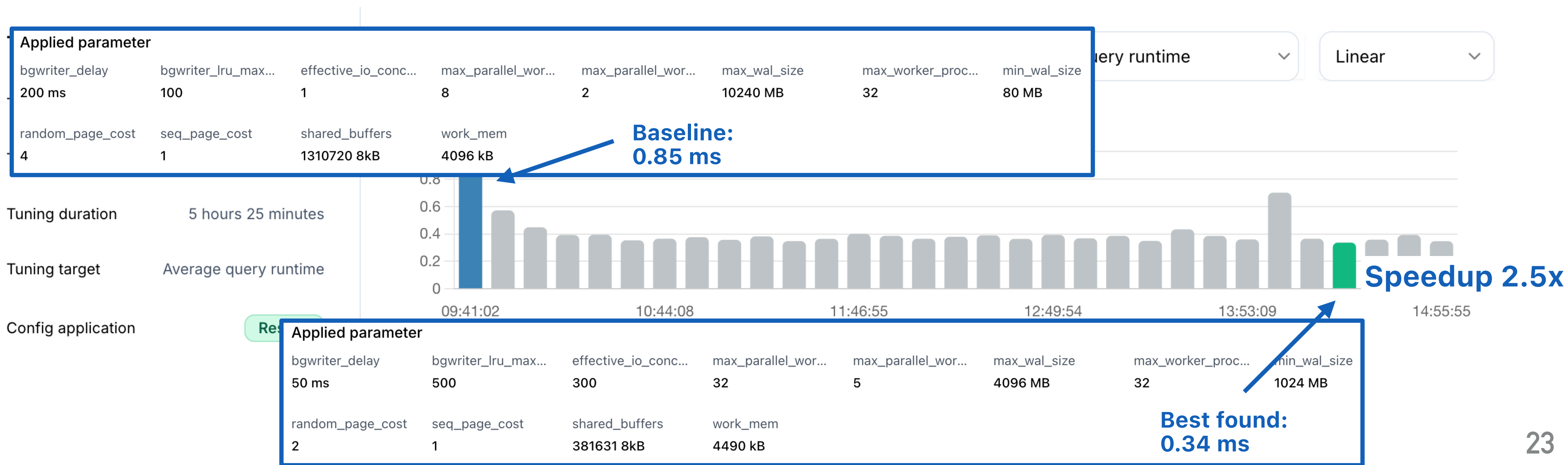
*A16z article: "The Cost of Cloud, a Trillion Dollar Paradox"

Insurance application use case study

Environment: 16 vCPU, 32 GB RAM, on-prem, primary instance, PG 15

Manually tuned baseline by expert DBA

Automated tuning with DBtune



DBtune deployment options

1. Managed: DBtune cloud

- DBtune optimizer as a service (OaaS)
- Private data is not shared outside the database instance

2. Self-hosted: Container image

- Can be hosted on premise or in virtual private network / cloud
- Container which can be run within customer network
- Offline — No external internet connectivity

Safety in production environments

System guardrails to avoid unsafe configurations



Constrained optimization

Parameters have safe upper / lower limits in place



Memory monitoring guardrail

Real-time system memory monitoring to revert from potentially unsafe configurations

E.g. configuration that uses too much RAM — Triggered at 90% of RAM

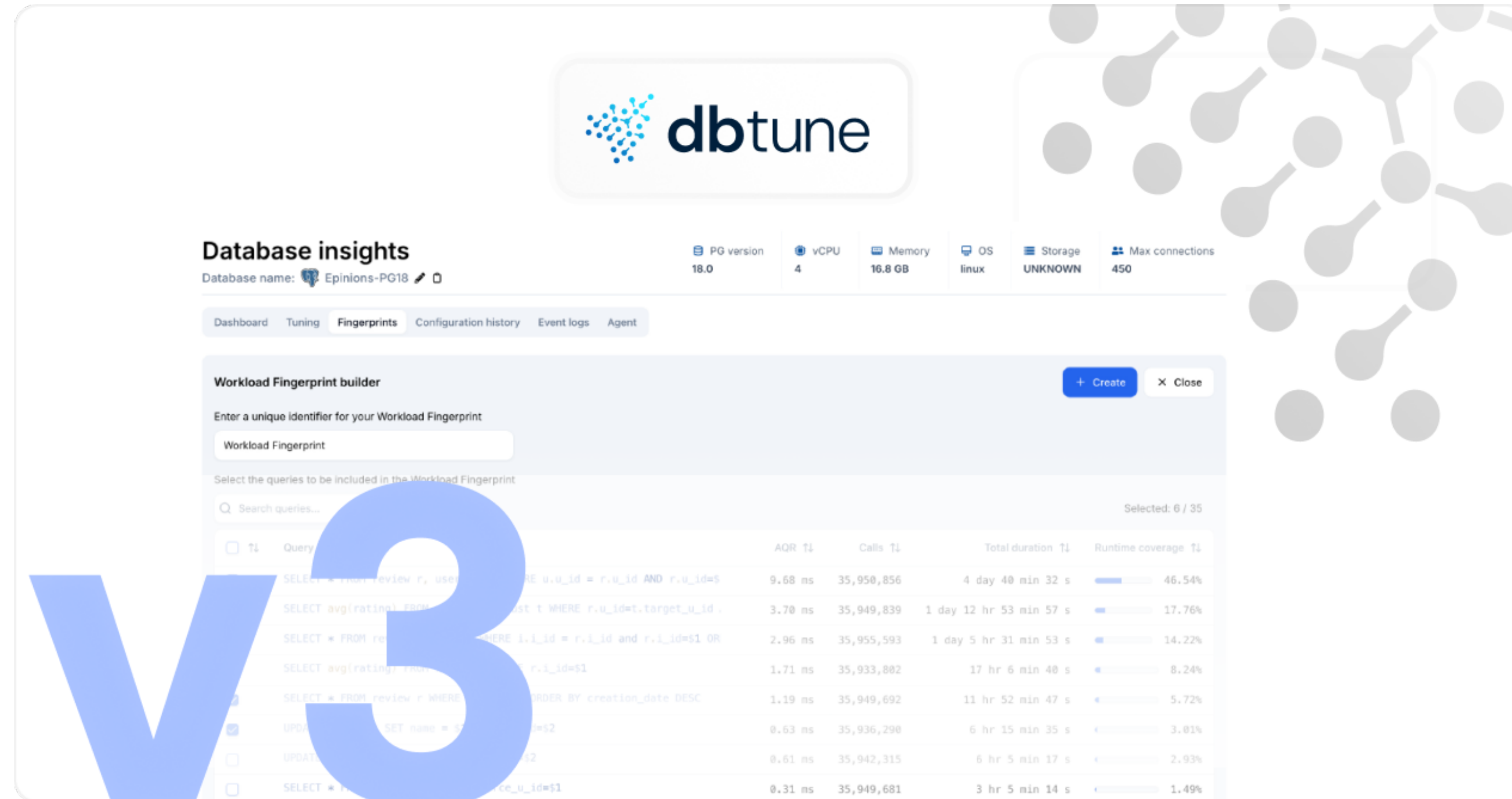


Early exit condition

Optimization space may result in configuration with worse performance than default

This triggers early exit from existing configuration and move to next iteration

DBtune demo



- Workload Fingerprint: Check [my talk at PGConf NYC](#) for a technical overview of the method
- DBtune now tunes a total of [18 PostgreSQL server parameters](#)
- Support for [Google Cloud SQL](#) for PostgreSQL
- Support for PG18
- Major GUI enhancements

Sign up today
app.dbtune.com

Or request a demo
info@dbtune.com



dbtune

