dbtune

Lessons learned from autotuning PostgreSQL

A 5-year long journey

Swiss PGDay

June 28, 2024

Luigi Nardi, Ph.D.

Founder & CEO, DBtune

About me

 @luinardi

 @nardiluigi

B.Sc and M.Sc. Computer Engineering at La Sapienza — Rome (Italy)

- 2006 M.Sc. thesis at LAAS-CNRS — Toulouse (France)
- 2007 Ph.D. Computer Science at Université Pierre et Marie Curie — Paris (France)
- 2011 Research Engineer at Murex SAS — Paris (France)
- 2014 Postdoc Imperial College London (UK)
- 2017 Research Staff at Stanford University (USA)
- 2019 Assistant Professor in Machine Learning at Lund University (Sweden)
- 2021 Founder & CEO at DBtune — Malmö (Sweden)
- 2024 Associate Professor in Machine Learning at Lund University (Sweden)

The DBtune team

dbtune

Leadership

Founder & CEO
Dr. Luigi Nardi
Stanford & Lund

Strategy Advisor
Moritz Zimmermann
42Cap, Hybris, SAP

Advisor
Dr. Kunle Olukotun
Stanford & SambaNova

Senior ML Engineer
Dr. Erik Hellsten
Chalmers, DTU, LTH, Volvo

Frontend Engineer
Aiman Mohsin
Diya, Sia Smtech

Special Consultant
Magnus Hagander
Redpill, PG Core Team

Strategy Advisor
Kingston Duffie
Serial Entrepreneur

Dev

Senior SW Engineer
Muhammad Umair
Freie, Heidelberg, SAP

Backend Engineer
Tahir Masood
FAST, Ibex Global

Senior DB Engineer
Dr. Curt Kolovson
Berkeley, VMware, MariaDB

Technology Advisor
Peter Zaitsev
Co-founder & CTO Percona

Tech Lead
Costa Alexoglou
Co-founder VisualEyes, Neo4j

Senior DevOps
Mohsin Ejaz
EDB

Senior ML Engineer
Raffaello Baluyot
Volvo, Autoliv

Technology Advisor
Johan Svensson
Co-founder & CTO Neo4j

M & S

Marketing Coordinator
Ellyne Phneah
LTH, ZDNet, Symantec

Director of Sales
Tom Howcroft
VoltDB

Marketing Advisor
Mark Jennings
Techstars, Notch

Sales Advisor
Alan Facey
B2B Sales Leader

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, DBMS parameter, index, OS parameter, etc.

Why does it matter?

Technical perspective

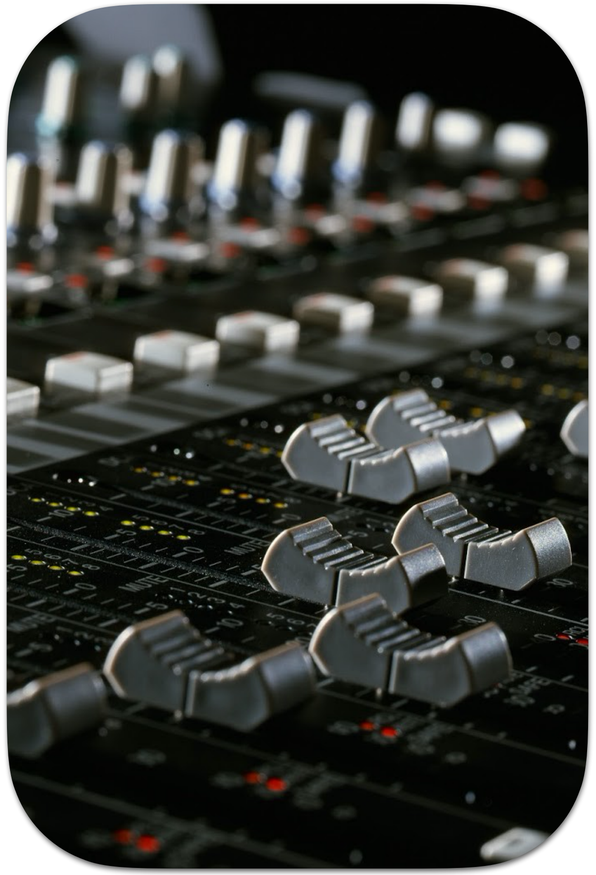
- Directly impacts system performance
  - Transactions per second — Throughput
  - Average query runtime — Latency
- Improves scalability / stability / reliability
- SLA

Business perspective

- Decreases cloud / infrastructure spend
- Higher end-user satisfaction
- Reduces downtime
- Increases productivity
- Sustainability

Database system parameter tuning

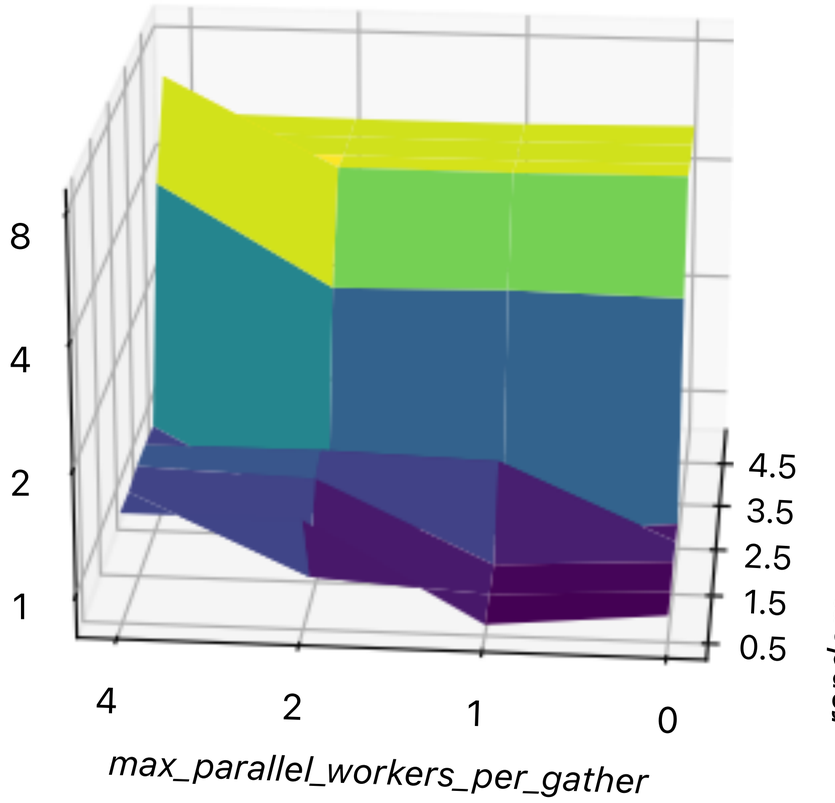
- Adjusting knobs to best fit the workload
- PostgreSQL parameters that are typically important: *work_mem*, *shared_buffers*, *max_wal_size*, etc.
- Example *max_parallel_workers_per_gather*:
  Max # of workers started by a Gather or Gather Merge node
- Example *random_page_cost*:
  Planner's cost of 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*

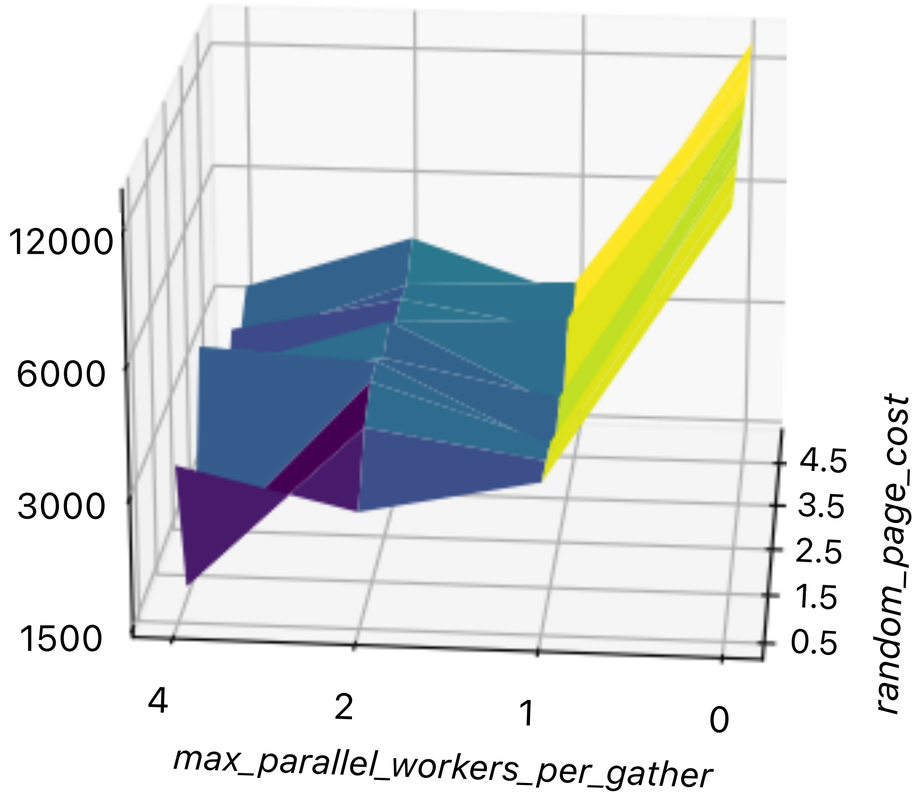
Epinions



Query runtime in ms

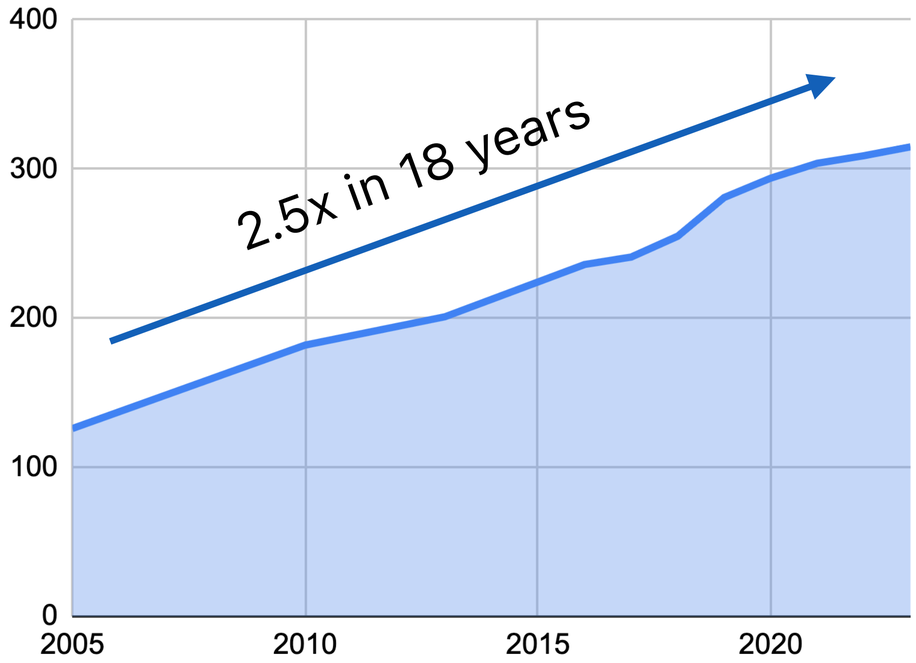
Lower the better

TPC-H



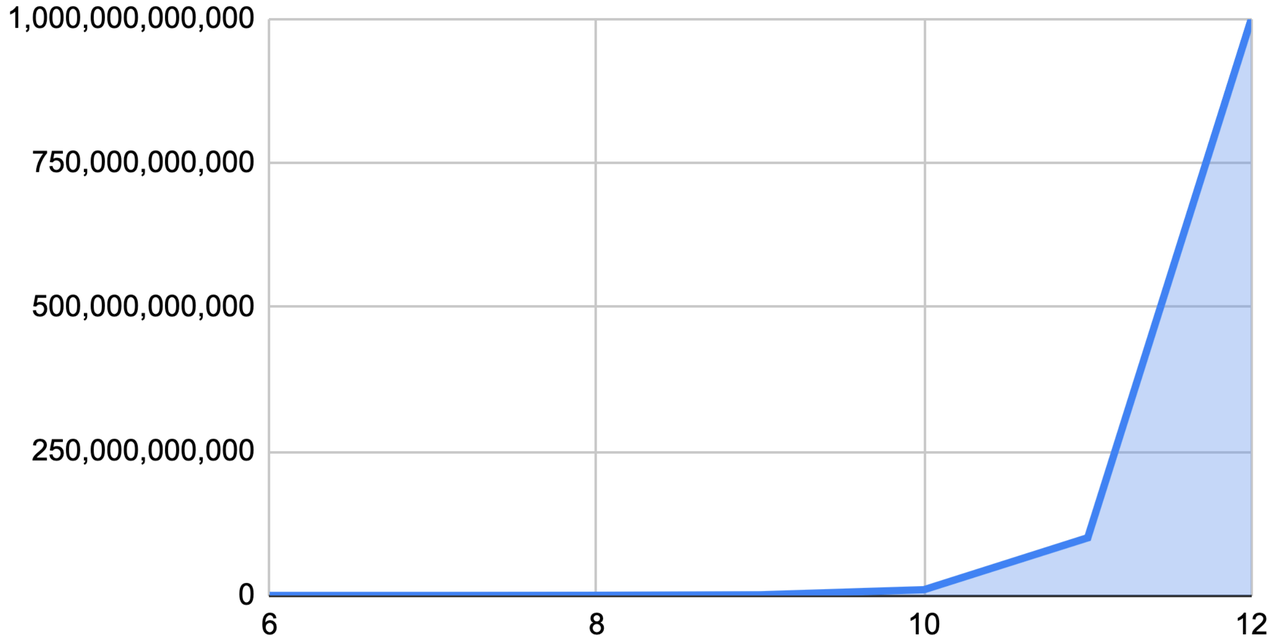
Complexity is growing over time

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

PGTune

POSTGRESQL CONFIGURATOR

One-size-fits-all
Uses generic rules

Workload agnostic
Not bespoke

Ineffective
Tune again in a week

Inadequate
Seasonal workload

New approach

dbtune

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 and changes
- You scale your cloud instance — Up or down

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

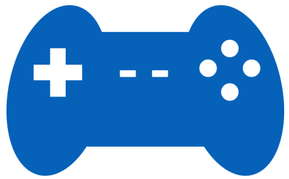
- Tuning is typically **reactive** to something going wrong — Not **proactive**
- Maybe looked at once or twice a year
- Often engage expensive external resources / experts
- Different workloads are not treated differently
- Modus operandi: Throw more hardware / compute at any issue ($$$)

We introduce DBtune

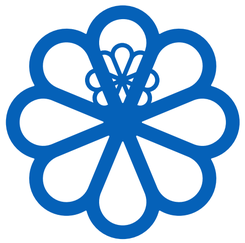
A unique ML-powered database tuning cloud hosted service

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 ML or database tuning



Highly scaleable → DBtune can tune multiple databases in heterogeneous environments

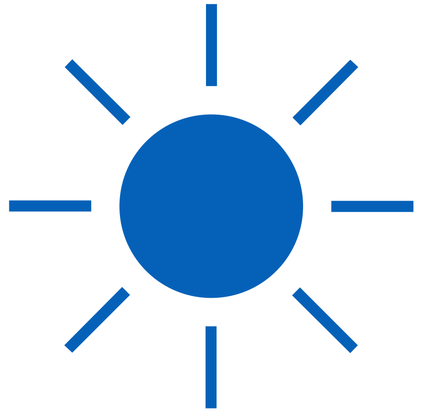
User value propositions

DBtune boosts service performance / improves business margins

Reduce cloud / infrastructure costs

Make your service radically faster

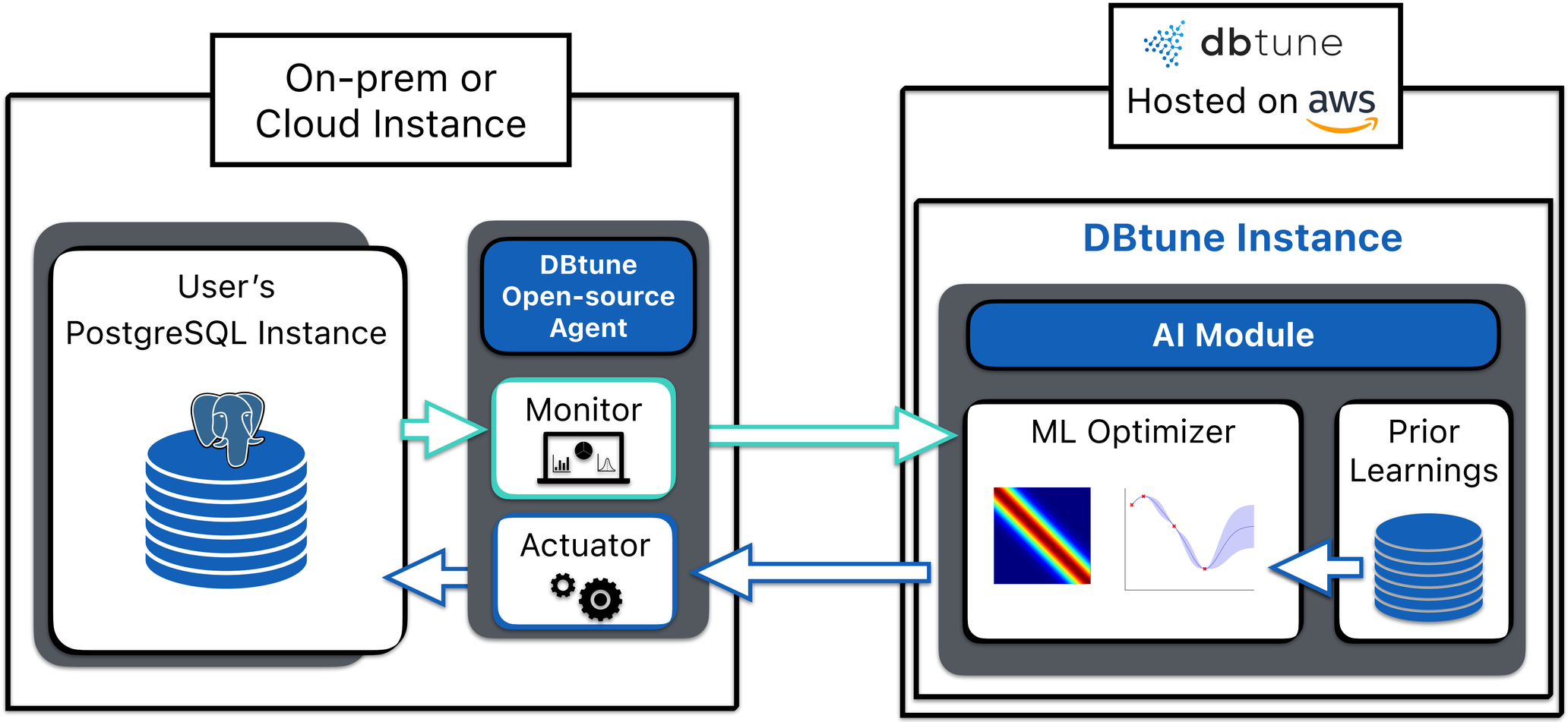
Free up your DBAs



Reduce energy consumption

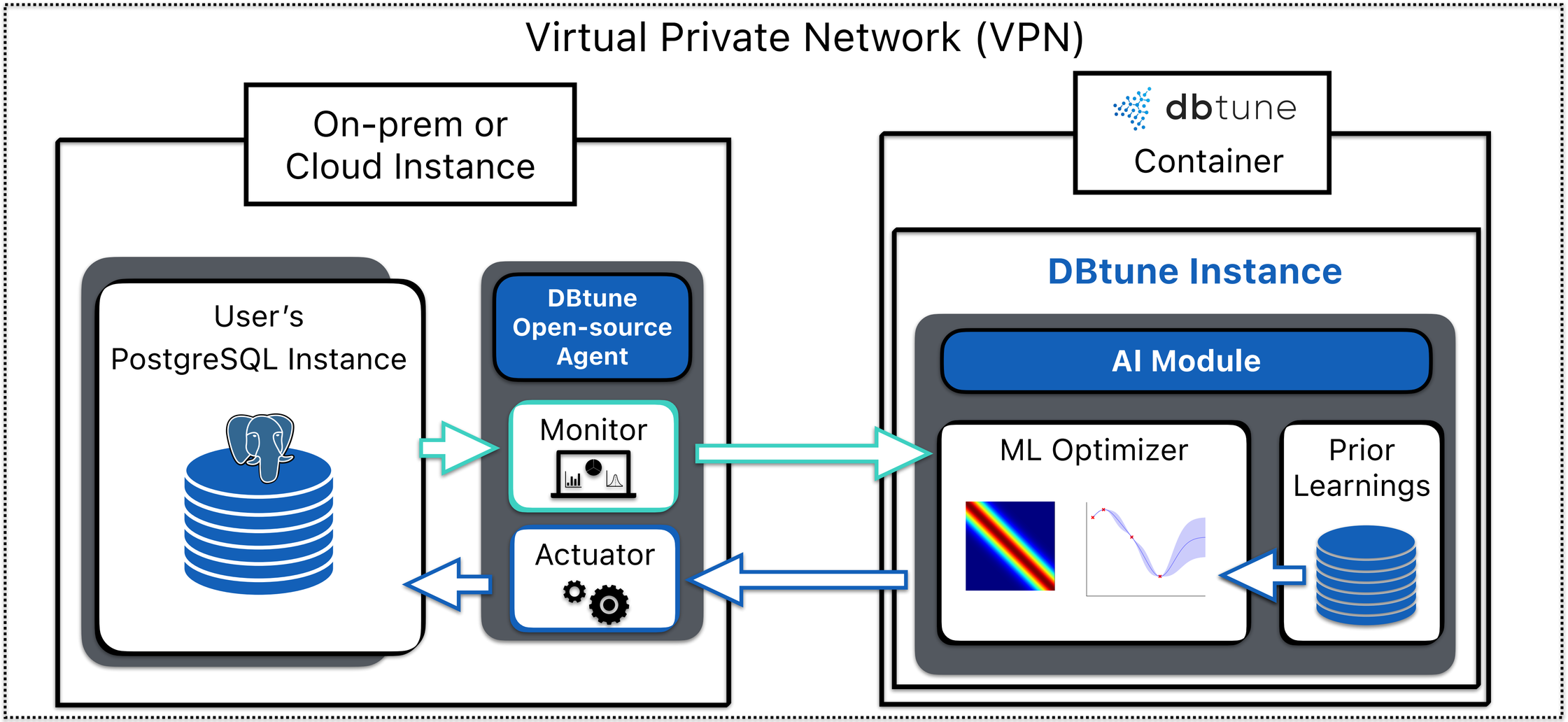
DBtune architecture for self-managed PostgreSQL (1)

High-level view



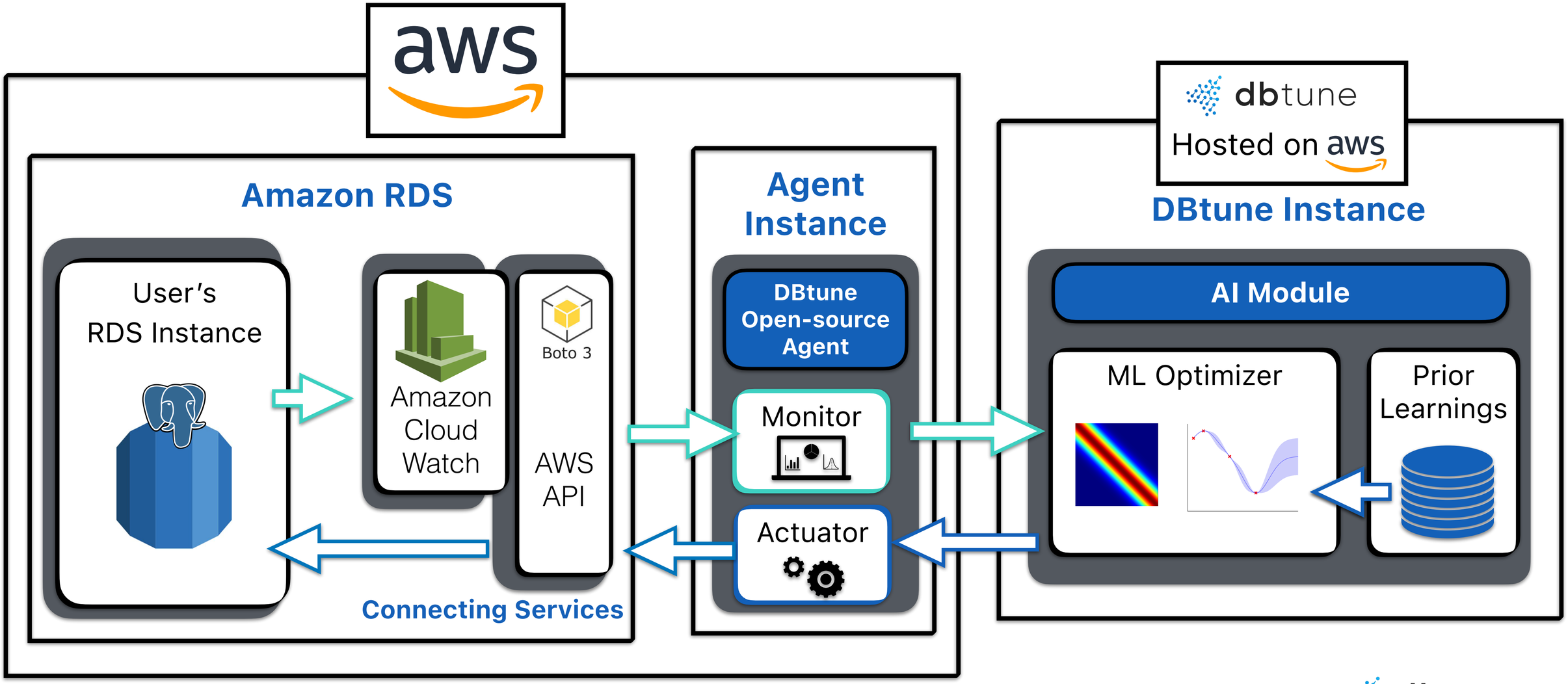
DBtune architecture for instances that are offline (2)

High-level view



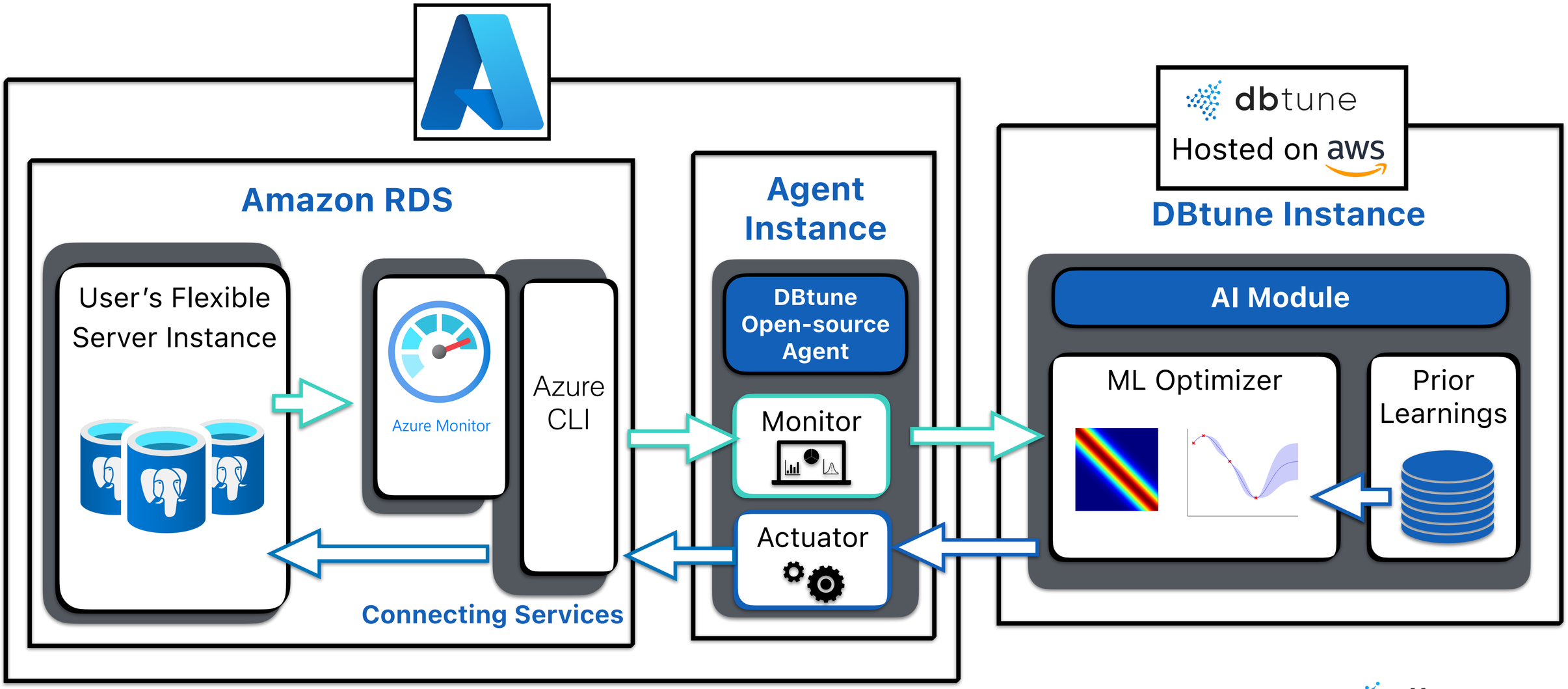
DBtune architecture for Database as a Service (DBaaS) (3)

High-level view RDS PostgreSQL/Aurora



DBtune architecture for Database as a Service (DBaaS) (4)

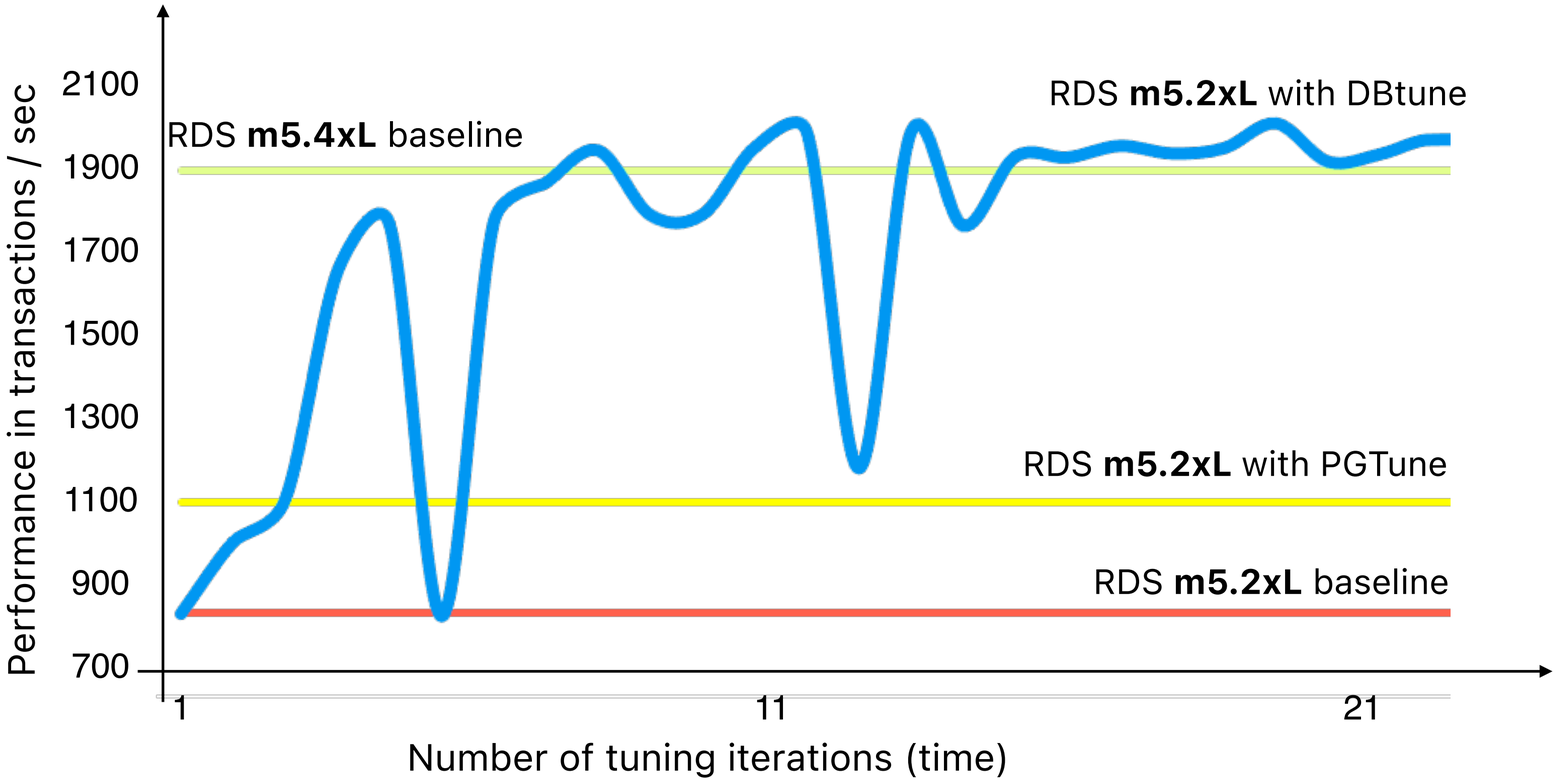
High-level view Azure Flexible Server



Performance tuning results

DBtune doubles the performance of PostgreSQL Amazon RDS

Performance impact of tuning RDS m5.2xLarge cloud instance on the TPC-C 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"

PostgreSQL parameters tuned by DBtune

Database reload (11 parameters)

- work_mem
- *max_parallel_workers*
- *max_parallel_workers_per_gather*
- *effective_io_concurrency*
- *bgwriter_lru_maxpages*
- *random_page_cost*
- *sequential_page_cost*
- *bgwriter_delay*
- *max_wal_size*
- *min_wal_size*
- *checkpoint_completion_taget*

0 or 1 restart with good defaults

- shared_buffers = 25%
- *max_worker_processes ~ vCPU*

With many database restarts

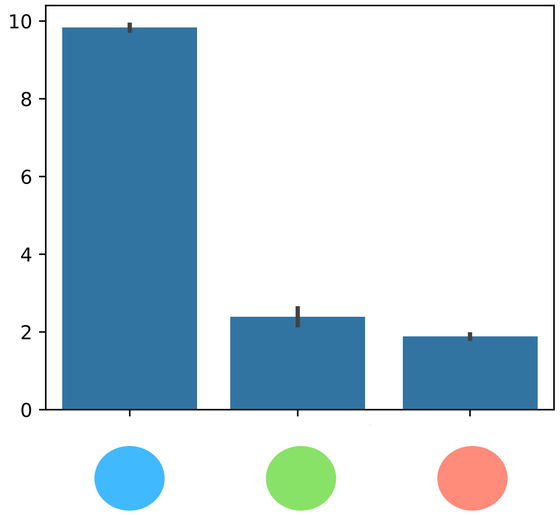
- shared_buffers
- *max_worker_processes*

http://docs.dbtune.com/Supported%20databases/PostgreSQL

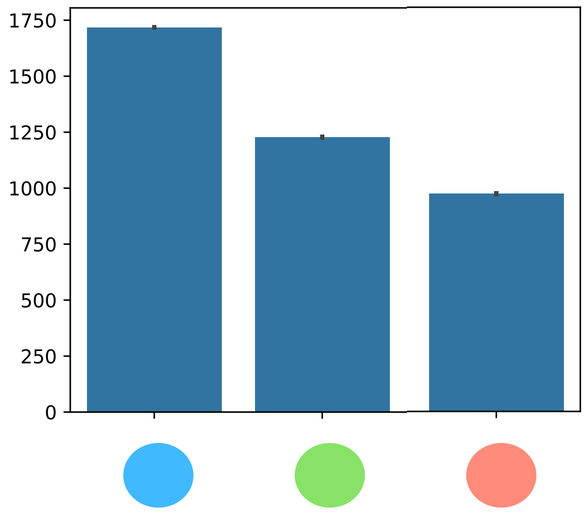
Performance downside of non-restart (reload-only) strategy

Average query runtime

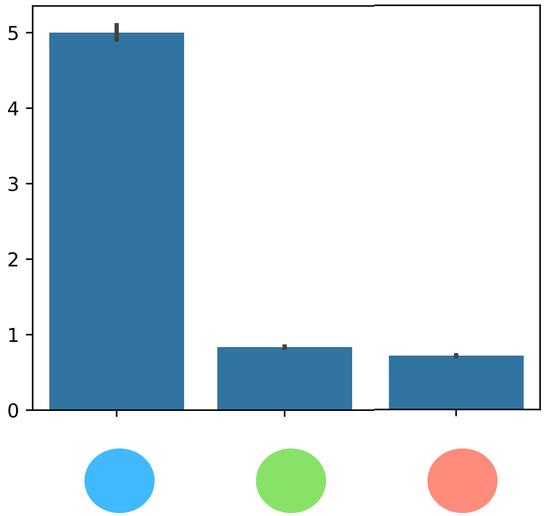
ResourceStresser



TPC-H



Epinions



Default PostgreSQL configuration & no autotuning

shared_buffers=25% & reload-only & autotuning

Restarts allowed & autotuning

Safe tuning 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

The sustainability angle — Out-of-the-envelop calculation

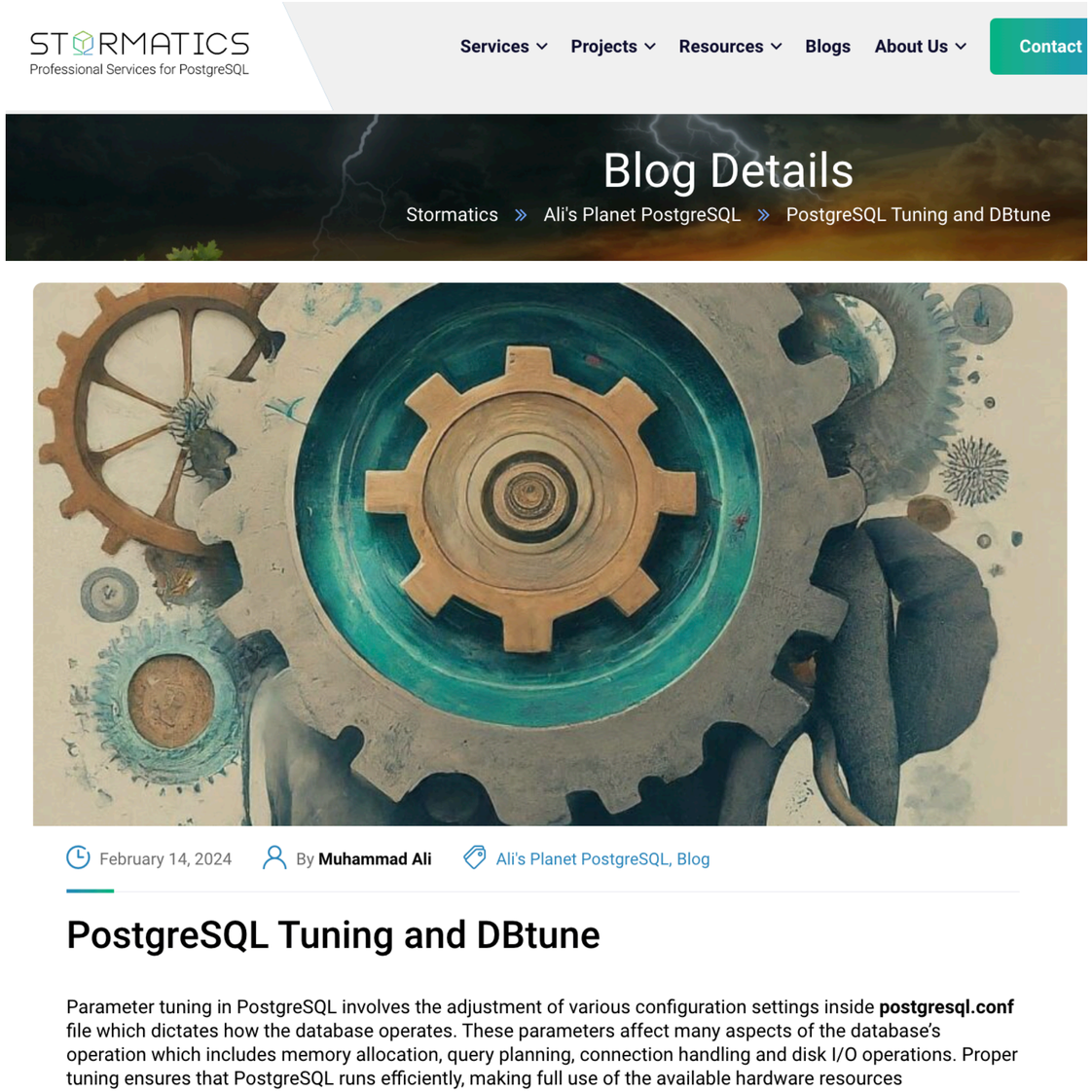
How much CO2 savings is at stake?

Database instance size largely impacts data center emissions

Impact on a single database	Impact on average data center	Impact on the EU data center landscape
-50%	**-32%**	**-7Mt**
CO2 emissions	CO2 emissions	CO2 savings p.a.
for a single database with DBtune[1]	for an average data center with DBtune[2]	across the EU with DBtune[3,4]

1) DBtune/Teads 2) Borderstep 3) Statista 4) EU digital strategy

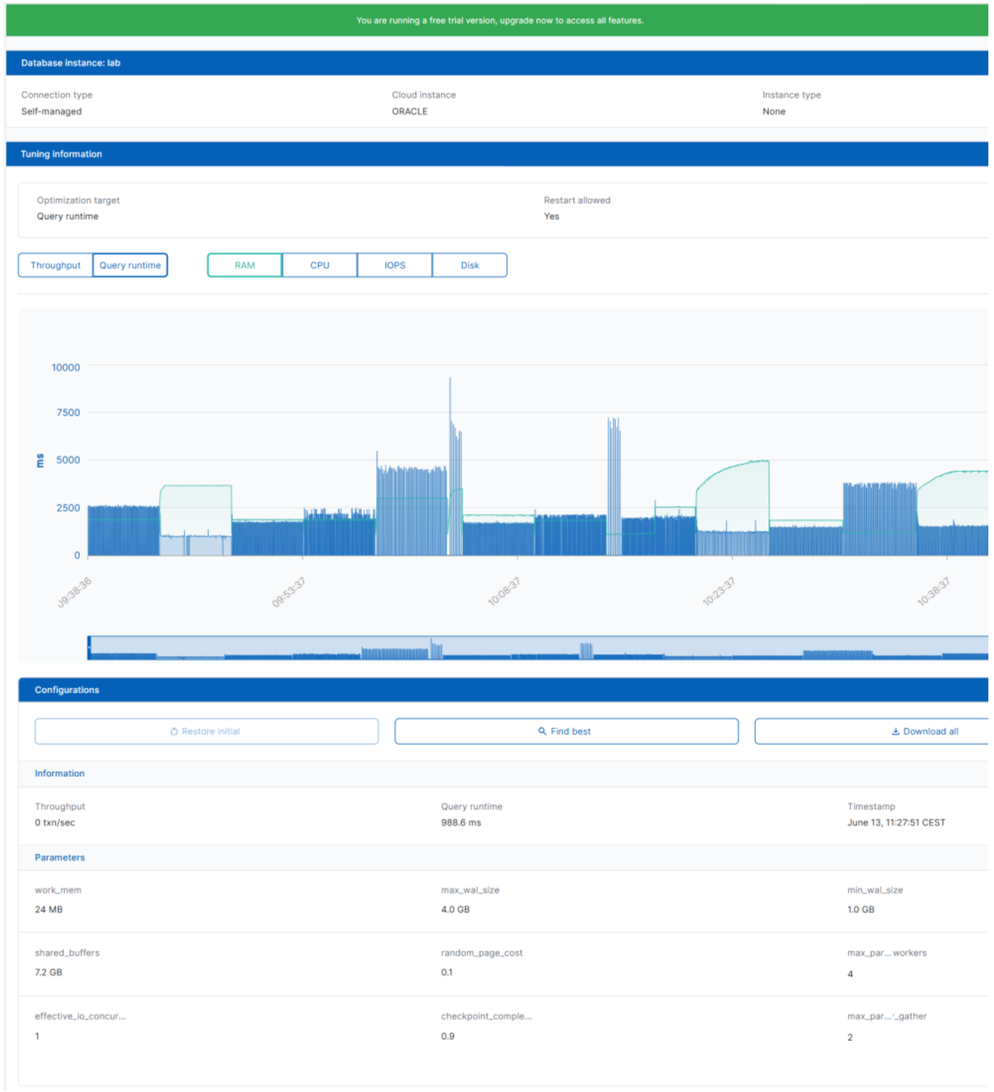
Independent evaluation by Stormatics



- Across all tests cases DBtune delivered improvement in performance up to 13.6x
- Compared to manual tuning DBtune achieved 2.2x speedup

Blog: https://stormatics.tech/alis-planet-postgresql/postgresql-tuning-and-dbtune

Independent technical analysis by Franck Pachot

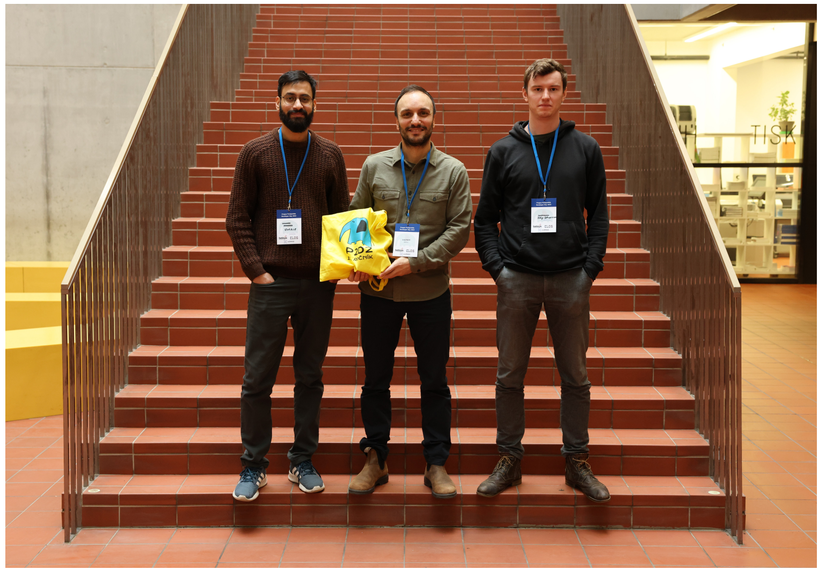


- DBtune proposes to optimize parameters you wouldn't think of
- Using DBtune doesn't mean that the DBA goes on vacation — DBtune does the boring stuff
- Humans prefer stable configuration, AI is more aggressive

Blog: https://www.linkedin.com/pulse/testing-dbtune-showing-postgresql-double-buffering-some-franck-pachot-voyhe

PG Developer Day Prague DBtune training

Live DBtune tuning with 25 attendees (January 31st, 2023)



On the left, a photo of our training session. On the top right three members of the DBtune team, Umair, Luigi and Filip, who delivered the training, and bottom the full event.

Useful links -> *github.com/dbtuneai/synthetic_workload*
Handout to try DBtune on a synthetic workload

luigi@dbtune.com

dbtune