Carnegie Mellon University

OTTERTUNE

Lessons Learned from Automatically Optimizing Databases Using Machine Learning in the Real World

Bohan Zhang

Cofounder @ OtterTune
Engineer@OpenAI

01. Background

02. The Real World

03. Technical Lessons

04. Business Lessons

01. Background

01. Background

Databases are **notoriously** complex to deploy, optimize, and maintain.

- Physical Design (Indexes, Partitioning)
- Knob Configuration
- Query Optimization
- Hardware Provisioning

Human experts are **scant**, **unscalable**, and **expensive**.

U.S. BUREAU OF LABOR STATISTICS

Occupational Employment and Wages, May 2021

15-1242 Database Administrators

Administer, test, and implement computer databases, applying knowledge of database management systems. Coordinate changes to computer databases. Identify, investigate, and resolve database performance issues, database capacity, and database scalability. May plan, coordinate, and implement security measures to safeguard computer databases. Excludes "Information Security Analysts" (15-1212) and "Database Architects" (15-1243).

National estimates for Database Administrators:

Employment estimate and mean wage estimates for Database Administrators:

Employment (1)	Employment RSE (3)	Mean hourly wage	Mean annual wage (2)	Wage RSE (3)
85,870	1.5 %	$ 46.42	$ 96,550	0.9 %

Percentile wage estimates for Database Administrators:

Percentile	10%	25%	50% (Median)	75%	90%
Hourly Wage	$ 23.50	$ 30.36	$ 46.50	$ 59.88	$ 72.79
Annual Wage (2)	$ 48,880	$ 63,160	$ 96,710	$ 124,550	$ 151,400

01. Automated Database Optimization

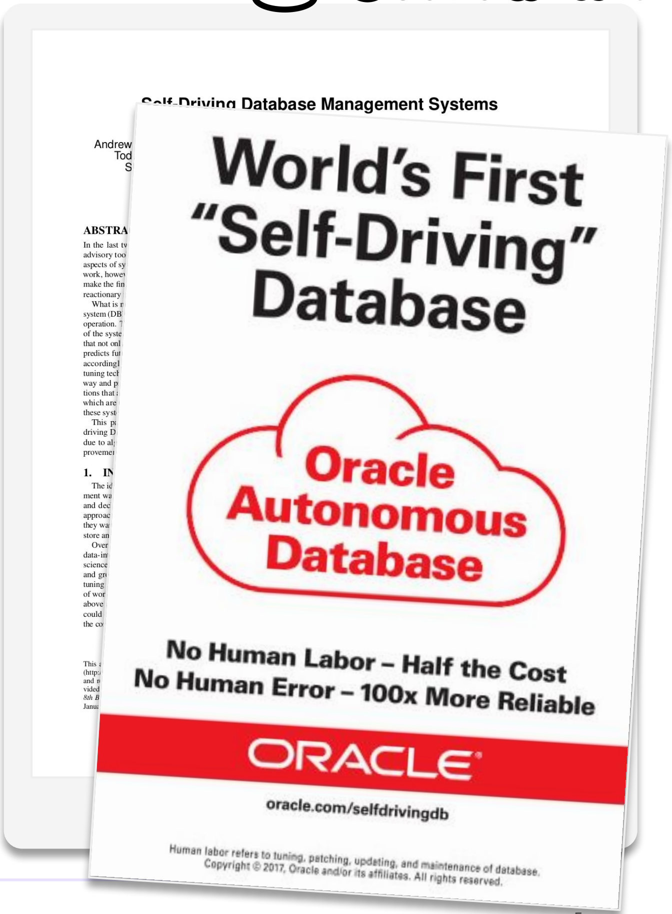
There is a long history of attempts in research to automate database management systems.

- 1970s: Self-Adaptive Databases
- 1990s/2000s: Self-Tuning Databases
- 2010/2020s: Self-Driving Databases

Microsoft AutoAdmin

ORACLE

IBM DB2



Research in the last decade has focused on applying **machine learning** (ML) methods to solve the tuning problem for databases.

01. Machine Learning for Databases

Indexes:

[Azure Auto Indexing](), [Oracle Autonomous Database Service](), [Cornell UDO](), [OpenGauss]()

Partitioning:

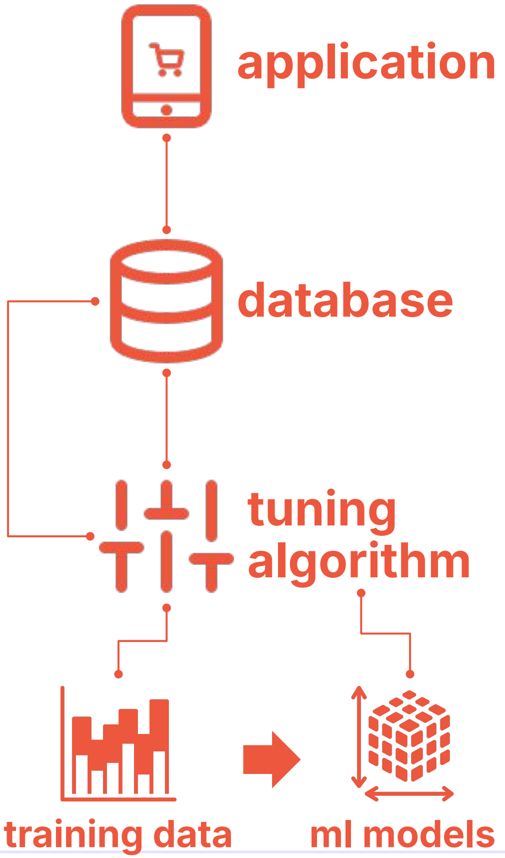
[Cloud Partition Adviser]()

Knob Configuration:

[OtterTune](), [CDBTune](), [Akamas](), [ResTune](), [QTune]()

Query Optimization:

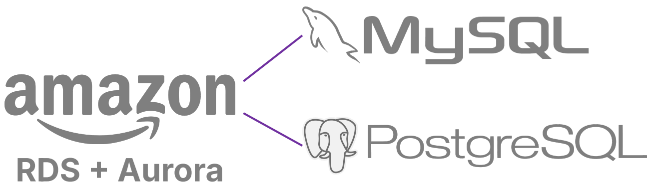
[Bao]() (Join Algos), [Neo]() (Join Ordering), [MySQL Heatwave Autopilot]() (Plan Stitching)



01. OtterTune Overview

OtterTune is an automated database tuning and resource optimization service.

Based on research developed at **Carnegie Mellon University** Database Group.



It uses **machine learning** to automatically optimize the configurations of DBMSs to improve performance, reduce costs, and maintain healthy operations.

Research: Knobs

Commercial: Knobs, Indexes, Queries, Cloud Config

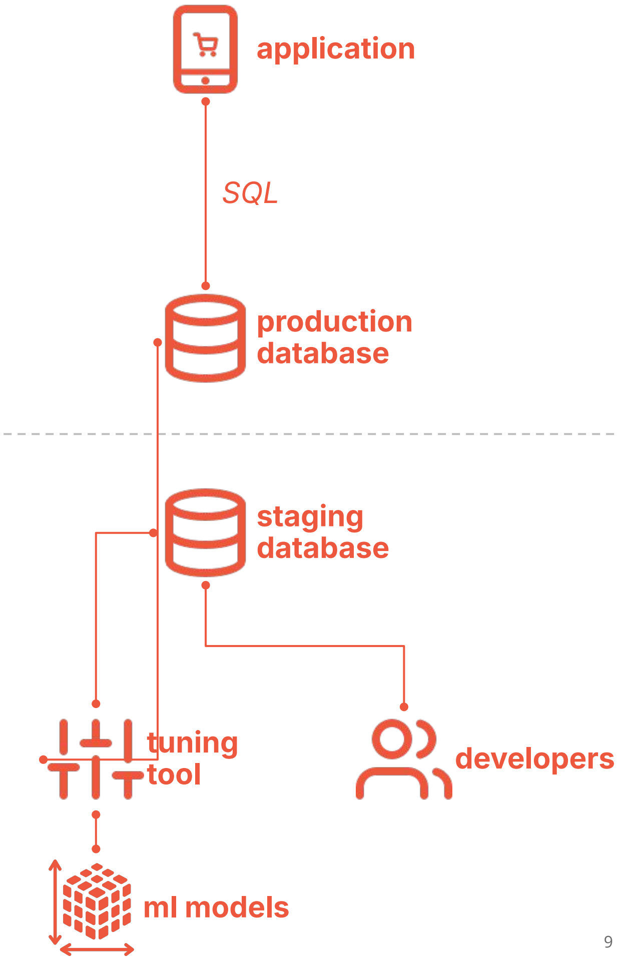
02. The Real World

02. Real-World Databases

Challenge #1: Users do <u>not</u> maintain suitable staging databases.

Training models on staging DBs is **bad** because of inconsistent workloads and hardware.

> Dev/Staging databases run on smaller hardware with a subset of the production databases because of cost.
>
> Dynamic hardware scaling (burst credits) and serverless instances make this worse.

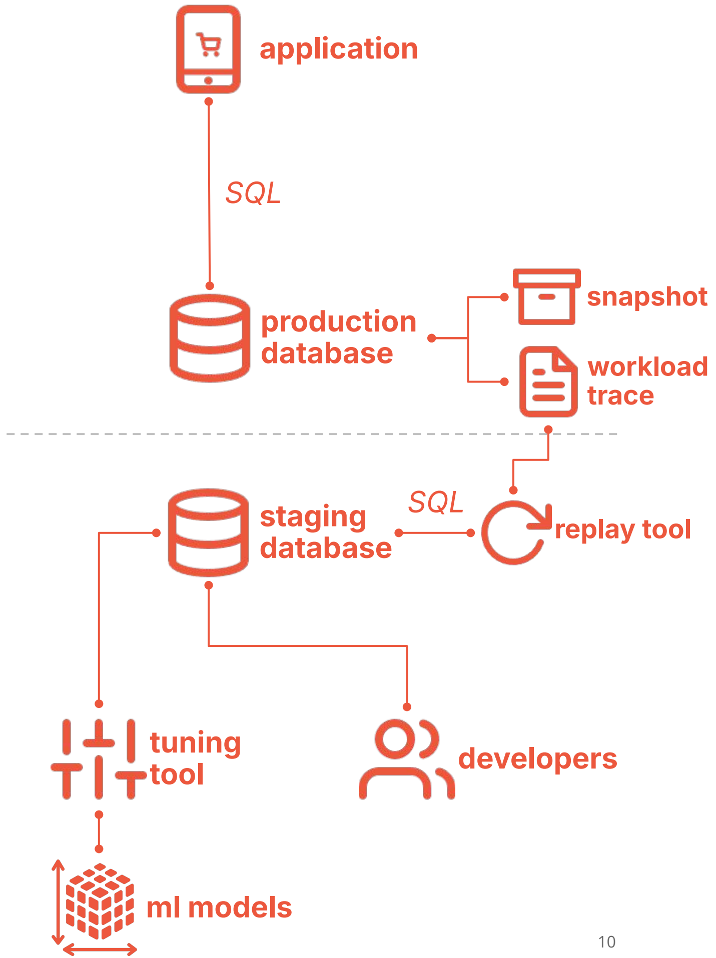


02. Real-World Databases

Challenge #2: Users <u>cannot</u> capture workloads and replay them.

Without a **repeatable workload** as a baseline, it is difficult for the ML models to learn whether they are improving a database.

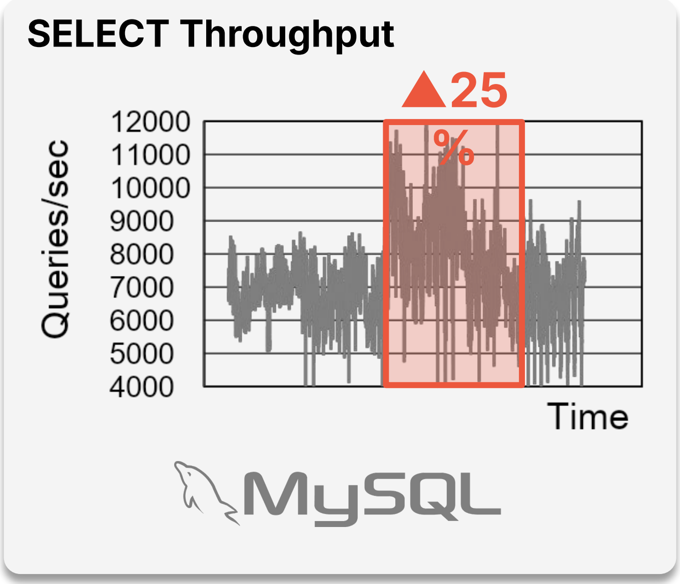
> Tools for open-source DBMSs are less sophisticated than commercial DBMSs.
>
> Existing built-in slow query log methods do not capture transaction boundaries. It's super expensive to log all queries to disk.



02. Real-World Databases

Challenge #3: Users mostly do <u>not</u> know what their database is doing. The production workload is dynamic.

Workload patterns and application **changes** make it difficult to measure whether a tuning tool is making things better or worse.



03. Lessons Learned

03. Production Database Tuning

Tuning a staging database using a replayed workload is impractical in real-world scenarios.

> Users do not maintain suitable staging databases.
>
> Users cannot capture workloads and replay them.

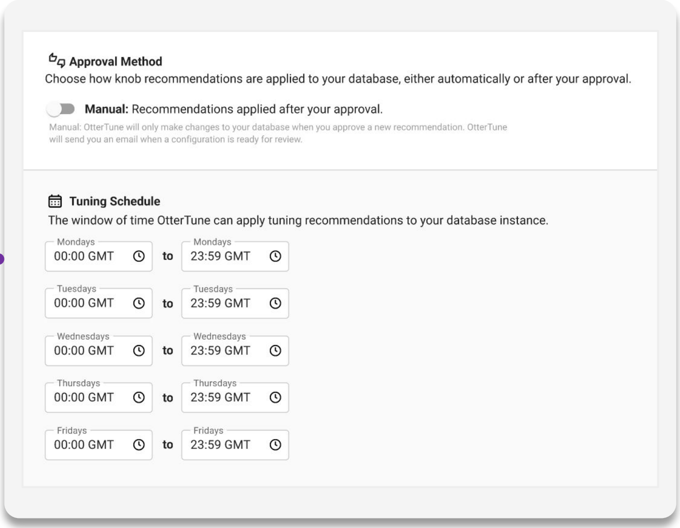
Most customers allow us to **carefully** tune their production databases. But they need to have control over **what**, **when,** and **why** changes are applied.

Customers don’t care about how fancy or novel your approach is. They care about how much **benefits** they can gain and how **quickly** they can see the value

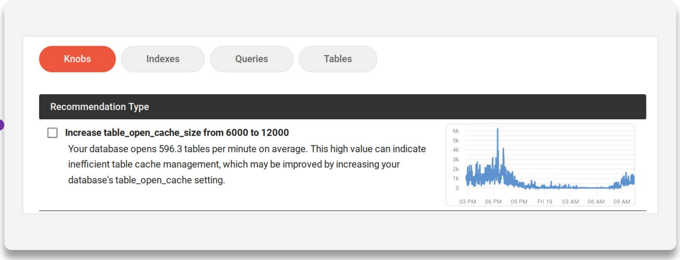
03. Manual Controls & Explanations

To reduce untimely performance degradations or downtime, a tuning service must provide controls to allow humans to specify **what** and **when** the service will optimize the database.

- Tuning Periods
- Knob Bounds
- Restart Tracking & Scheduling
- Human-in-the-loop Approval



To help build trust, provide users with data-driven explanations about recommendations. (**why** they should apply them)



03. Machine Learning & Domain Knowledge

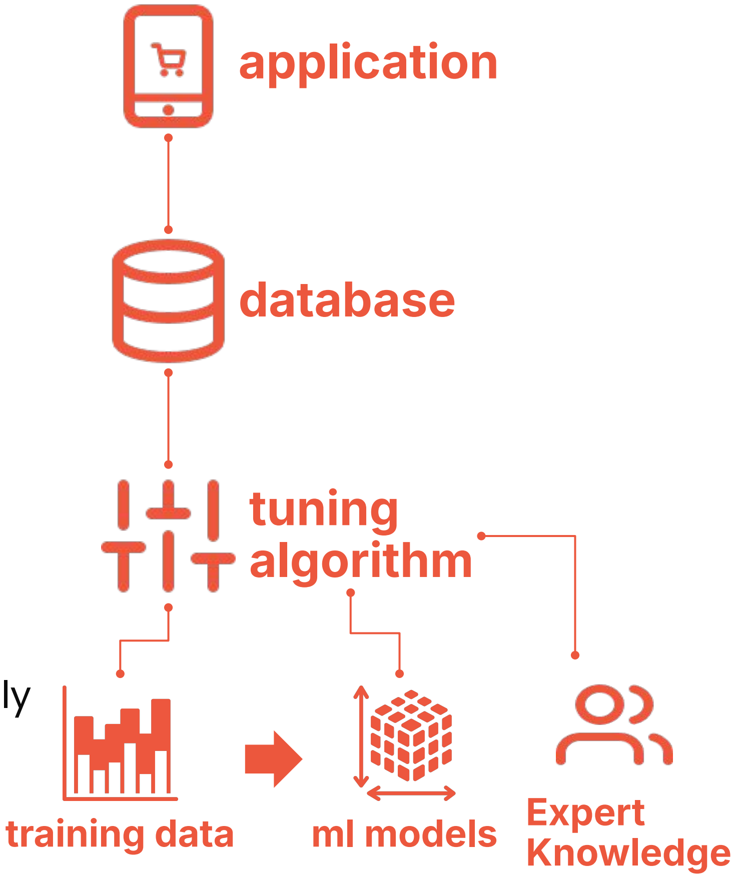
Machine Learning models can help find the (nearly) optimal database configurations.

> No explanations about recommendations
>
> Need time to converge

Heuristic-based approach using domain knowledge may not yield the optimal recommendations.

> Provide explanations about recommendations
>
> Do not need training data, recommend immediately

You don't need to pick just one. Integrate **both** methods for better recommendations.



03. Not Only A Machine Learning Problem

Automated database tuning with ML works **better** in the real-world than in the research lab, but getting the full benefit of these optimizations and tuning production databases **safely** is **not only a ML problem**.

> Knob Bounds
>
> Tuning Periods & Schedule
>
> Explanations about recommendations
>
> …

We extend OtterTune to support three ML models. They all have **similar** performance:

> **Gaussian Process Regression**
>
> Deep Neural Network
>
> Deep Deterministic Policy Gradient

An Inquiry into Machine Learning-based Automatic Configuration Tuning Services on Real-World Database Management Systems
VLDB 2021

04. Business Lessons

04. Business Lessons

- Our biggest challenge is from the business side, rather than the technical side.

- Database tuning is a **nice-to-have** rather than a **must-have**. In VC terms, it's more of a **vitamin** than a **painkiller**.

  Our priority is low. People show interest but often get pulled away by other higher-priority tasks, making it take too long to actually try and buy our product. It's difficult to scale.

- Many users try our product for free in the first month to get most of the benefits and walk away. We are struggling to make the product **sticky**.

  Dynamic workloads cannot save us.

04. Business Lessons

- **Deal sizes** are limited. Even with large enterprise customers, they will not pay us a lot.

  1k MySQL instances -> 5 engineers to manage them

  2k MySQL instances -> still 5 engineers to manage ?

  It doesn't scale linearly with the size of the data.

  Even large enterprise customers are only willing to pay us a single DBA's salary.(~$150k)

- The beauty of data companies like Snowflake, Databricks, and Datadog is that their costs **scale linearly** with the size of the data.

  1k instances -> $X;   2k instances -> $2X

  Large Enterprise with massive data scale can pay lots of $$$. (largest deal can be > $50M per year)

04. The only thing that matters

The only thing that matters

In a great market—a market with lots of real potential customers—the market *pulls* product out of the startup.

Marc Andreessen
https://pmarchive.com/guide_to_startups_part4.html

Does ML for DB really work?

END

bohan@openai.com
@BohanZhangOT