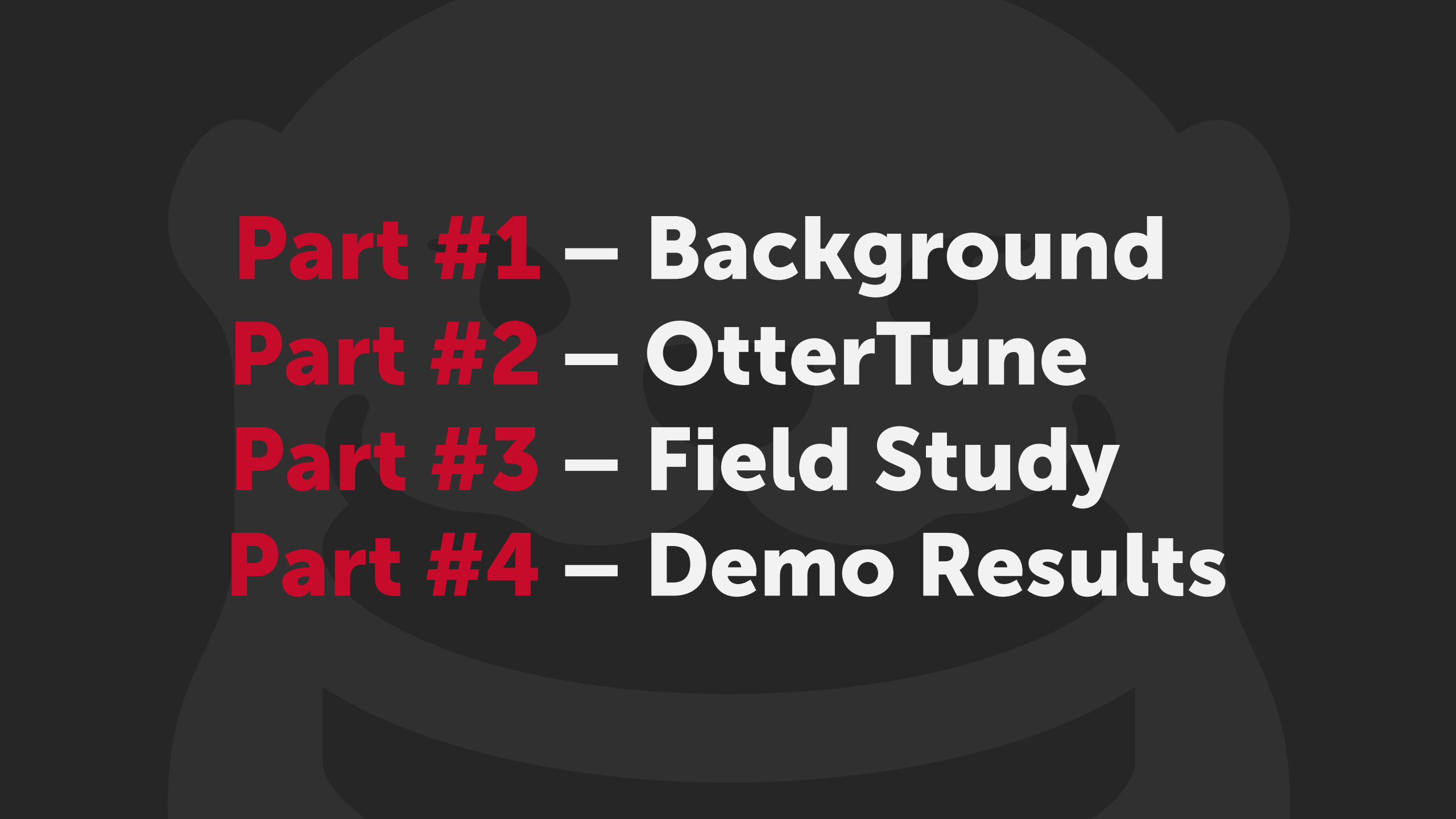
OTTERTUNE

Automated Database Tuning Service

https://ottertune.com

@andy_pavlo


Part #1 – Background
Part #2 – OtterTune
Part #3 – Field Study
Part #4 – Demo Results

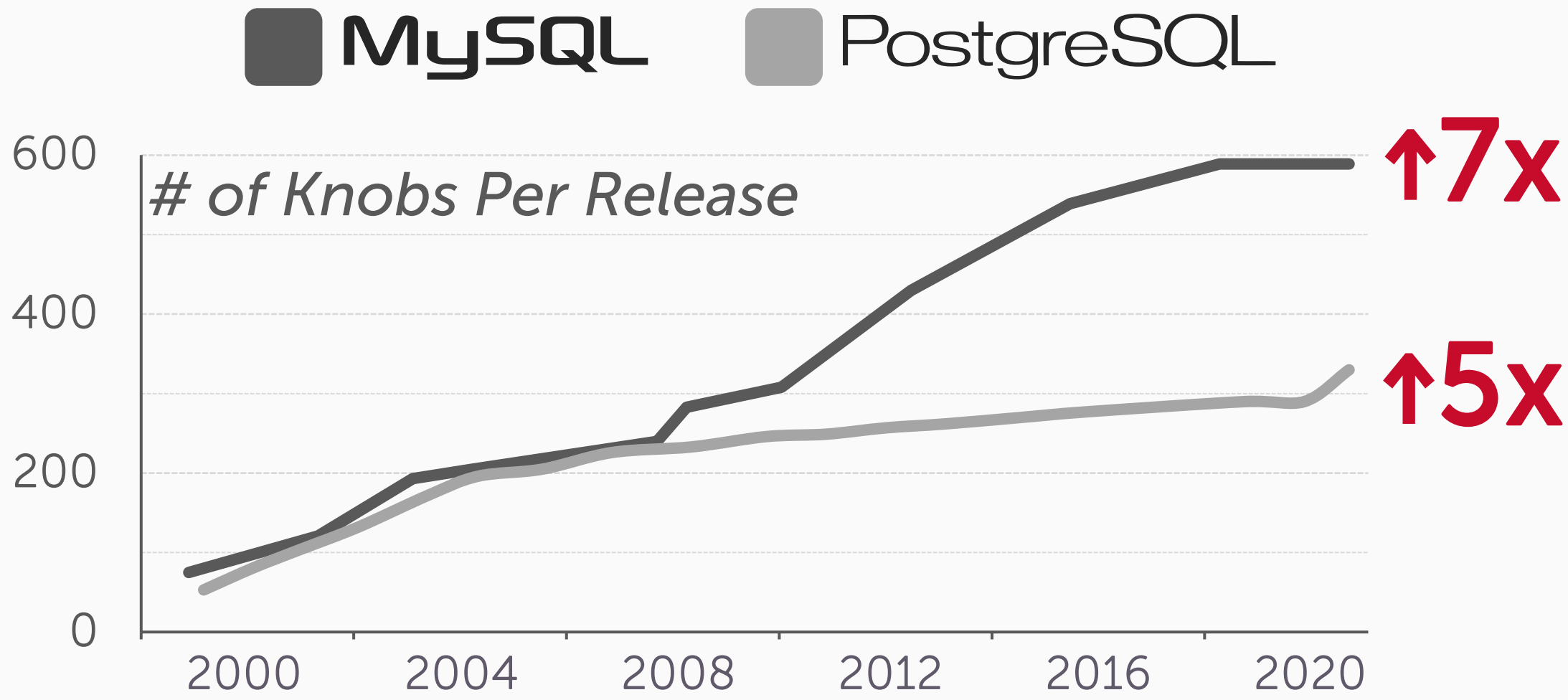
KNOB TUNING

Every DBMS exposes knobs that control the runtime behavior of the system.

Tuning the knobs for a workload improves the DBMS's performance & efficiency.

Managing many knobs on many instances exceeds the abilities of humans.

KNOB TUNING



Source: https://db.cs.cmu.edu/papers/2017/p1009-van-aken.pdf

OtterTune

OVERVIEW

OtterTune is a DBMS configuration monitoring and tuning service.

It uses machine learning to automatically optimize the knob configuration of DBMSs to improve their performance and reduce hardware/software costs.

Automatic Database Management System Tuning Through Large-Scale Machine Learning
SIGMOD 2017

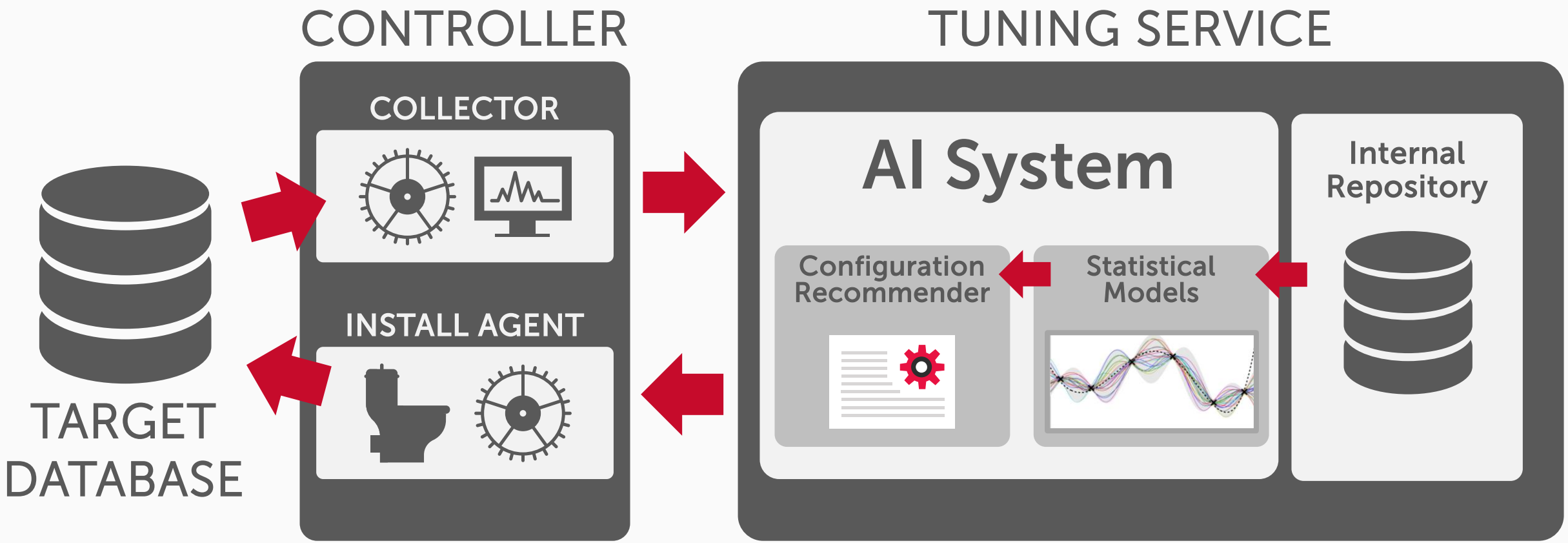
A Demonstration of the OtterTune Automatic Database Management System Tuning Service
VLDB 2018

Demonstration

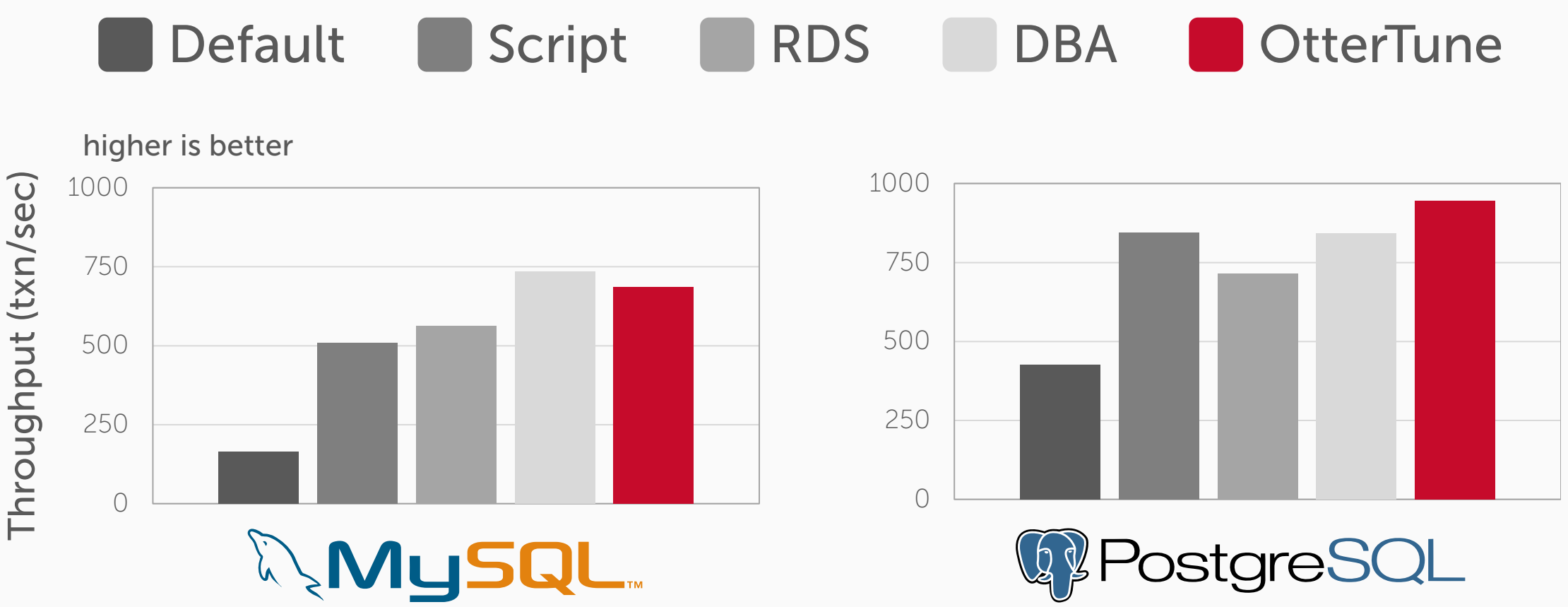
Postgres v11 • Amazon RDS

TPC-C Benchmark

ARCHITECTURE



TPC-C EVALUATION



Source: https://db.cs.cmu.edu/papers/2017/p1009-van-aken.pdf

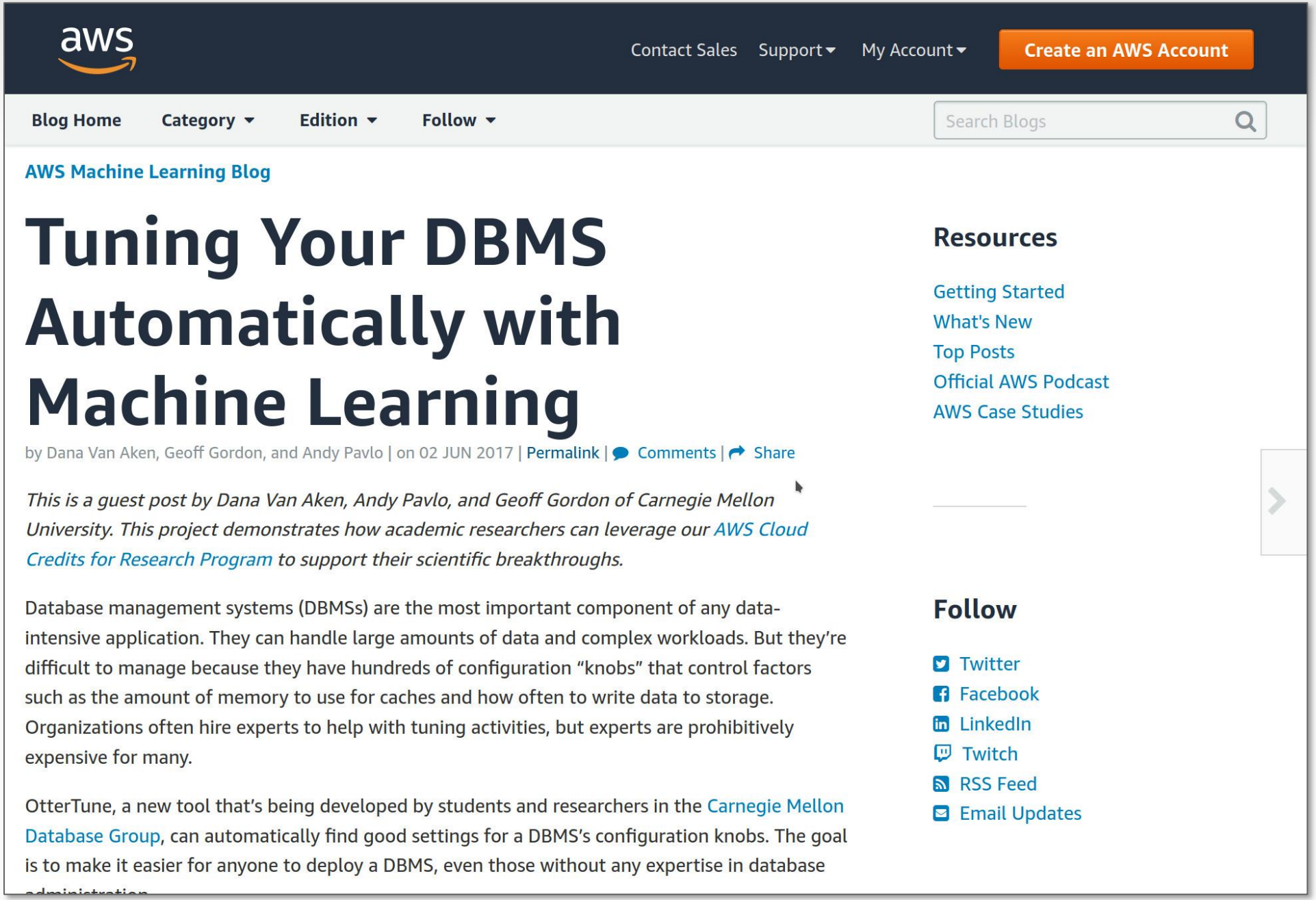
WHAT NEXT?

We investigated several directions in enhancing OtterTune's ML algorithms:

- → Hardware Context
- → Sharing Data Across Versions
- → Improved Workload Mapping

Nothing made a difference because synthetic workloads like TPC-C did not require more complex methods...

Field Study



Source: https://aws.amazon.com/blogs/machine-learning/tuning-your-dbms-automatically-with-machine-learning/

FIELD STUDY: SOCIETE GENERALE

A French bank asked us in 2018 to setup **OtterTune** to tune 1000 Postgres instances.

Docker-based, self-service deployment architecture that provided pre-configured knob tuning.

FIELD STUDY: SOCIETE GENERALE

A French bank asked us in 2018 to setup **OtterTune** to tune ~~1000~~ One Postgres instances.

Docker-based, self-service deployment architecture that provided pre-configured knob tuning.

We collaborated for eight months…

Attempt #2

FIELD STUDY: SOCIETE GENERALE

A French bank asked us in 2019 to setup **OtterTune** to tune an Oracle v12 instance.

→ 1TB Database, read-heavy OLTP workload.

→ Custom ticket tracking application for IT support.

Configuration already tuned by expert DBA.

Target Objective: Oracle DB Time

Attempt #2

FIELD STUDY: SOCIETE GENERALE

A French bar

OtterTune to

→ 1TB Database,

→ Custom ticket

Configuratio

Target Obje

DB Time Definition

- Total time in database calls by foreground sessions
- Includes
  - CPU time
  - IO time
  - non-idle wait time
- DB Time is not called response time on purpose
- Common currency for Oracle performance analysis

EVALUATION

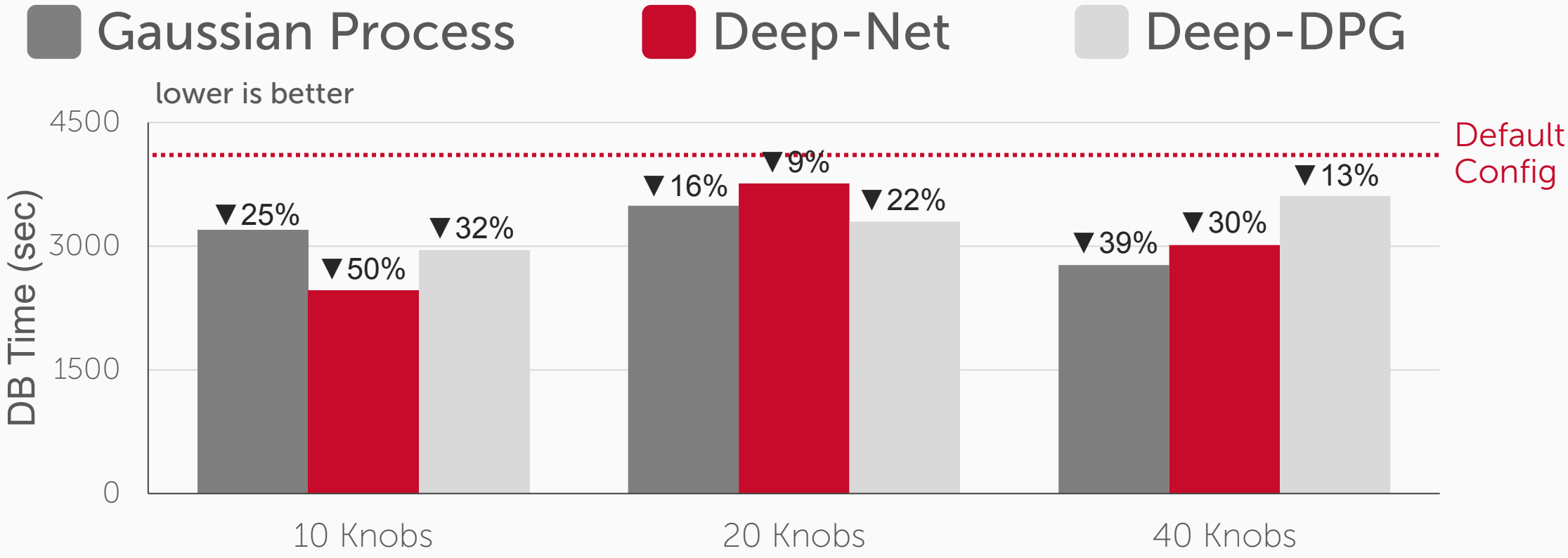
We extended OtterTune to support three ML tuning algorithms:

- → Gaussian Process Models (OtterTune 2017)
- → Deep Neural Network (OtterTune 2019)
- → Deep Deterministic Policy Gradient (CDBTune 2019)

Evaluate their ability to tune the DBMS for 10, 20, 40 configuration knobs.

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

EVALUATION



Average of Three Sessions per Algorithm/#Knobs

Total Compute Time: 81 Days

Lessons Learned

LESSONS LEARNED

ML Algorithm Does Not Matter (For Now)

Tuning Data is Not Privacy Concern (For Now)

Replaying Workload Traces

Handling Bad Configurations

Restarting

REPLAYING WORKLOAD TRACES

Workload replay time depends on the configuration, but bad configurations could run for hours.

→ We need to keep the workload (almost) the same in each iteration to avoid metrics falsely reporting lower results.

Solution: Avoid long-running iterations by aborting the workload replay early.

BAD CONFIGURATIONS

When there is little prior data, algorithms will choose bad configurations that cause failures.

- → Scenario #1: Slow Execution
- → Scenario #2: DBMS Fails to Start
- → Scenario #3: DBMS Fails After Delay

Need to provide feedback to the algorithms that a configuration was bad.

BAD CONFIGURATIONS

How to identify when there is a failure?

Solution: Scrape log for explanation/indicators of DBMS status.

How to incorporate results from bad configuration in the algorithm's training data?

Solution: Set the objective function to the worst configuration seen so far.

SYSTEM RESTARTS

Some DBMS knobs do not take effect until you restart the system. But nobody likes to restart their DBMS unless they really have to.

The time it takes to bring the DBMS back online after restart can be non-deterministic based on the configuration changes.

→ Example: MySQL log file size

SYSTEM RESTARTS

How to know whether the service is allowed to restart DBMS?

Solution: A human must tell us. They can also specify # of restarts per day and time window.

How to estimate how long it will take the DBMS to complete restart?

Solution: Open problem. Lots of factors…

WHAT'S NEXT

As of August 2020, we have spun out a **OtterTune** from Carnegie Mellon as a start-up.

Currently deploying **OtterTune** at companies to tune their DBMSs.

→ Only MySQL and Postgres for now.

https://ottertune.com

Demo Results

CONCLUSION

ML can solve some of the problems with automatic tuning of DBMSs.

→ Requires non-ML infrastructure to correctly capture workload traces and database snapshots.

Using ML effectively for DBMS optimization requires knowledge of both ML and systems.

END

@andy_pavlo