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

Dana Van Aken, Dongsheng Yang, Sebastien Brillard, Ari Fiorino, Bohan Zhang, Christian Bilien, Andrew Pavlo
Motivation

Problem
Modern DBMS have hundreds of tunable configuration knobs

Existing Approaches
DBA
Static rules
Machine learning

Limitations
Open-source DBMSs
Synthetic benchmarks
Dedicated local storage

Source: Aken et al., 2021
Machine Learning Based Knob Tuning

**Algorithms**

- Gaussian Process Regression (GPR)
- Deep Neural Network (DNN)
- Deep Deterministic Policy Gradient (DDPG++)

**Real-world case study**

- Real-world ticketing application

- Oracle database

- Shared storage

Source: Aken et al., 2021
OtterTune Architecture (Aken et al., 2017)

Source: Aken et al., 2021
GPR/DNN Tuning Pipeline (OtterTune)

Source: Aken et al., 2021
DDPG++ Tuning Pipeline (CDBTune)

Source: Aken et al., 2021
Machine Learning Based Knob Tuning

Algorithms

Gaussian Process Regression (GPR)

Deep Neural Network (DNN)

Deep Deterministic Policy Gradient (DDPG++)

Real-world case study

Oracle database

Real-world ticketing application

Source: Aken et al., 2021
Tuning Knobs Selected by DBA

Source: Aken et al., 2021
Tuning Knobs Ranked by OtterTune

Source: Aken et al., 2021
Adaptability to Different Workloads

Source: Aken et al., 2021
Summary

➢ OtterTune extension: DNN and DDPG++
➢ Real-world case study: ticketing application running on Oracle database with shared storage
➢ All approaches significantly outperform the baseline, however, there is no clear ranking between the different approaches
➢ Hybrid approach combining DBA-selected and ML-tuned knobs outperforms fully automated approach
➢ Latin hypercube sampling performs (surprisingly) well for small knob sets
➢ The underlying hardware can have a significant impact on the performance

Source: Aken et al., 2021
Some Thoughts on the Paper

➢ Original OtterTune paper ✓, follow-up paper ❌
➢ Paper contains errors

“As an algorithm learns more, it is less likely to select poor configurations. Thus, the number of long-running replays decreases as the algorithm nears convergence. Table 4 shows that GPR has the fewest canceled replays. DDPG++ has fewer canceled replays than DDPG due to its improved convergence rate (see Section 4.3). LHS has the highest workload execution time and percentage of canceled replays because it is a sampling technique and never converges.” [p. 11]

<table>
<thead>
<tr>
<th></th>
<th>GPR</th>
<th>DNN</th>
<th>DDPG</th>
<th>DDPG++</th>
<th>LHS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Execute (sec)</td>
<td>762</td>
<td>1006</td>
<td>1021</td>
<td>1274</td>
<td>1311</td>
</tr>
<tr>
<td>% Canceled</td>
<td>1.8%</td>
<td>8.7%</td>
<td>12.9%</td>
<td>26.8%</td>
<td>32.4%</td>
</tr>
</tbody>
</table>

Table 4: Workload Replay Time per Algorithm – The median workload execution time and the percentage of replays canceled for the algorithms.

Source: Aken et al., 2021
Some Thoughts on the Paper

- Industry case-study, few novel contributions to the field
- Issues raised in the introduction are only partially addressed
  - Open-source vs. enterprise DBMS
  - Synthetic vs. real-world workloads
  - Dedicated vs. shared storage
- Results provide limited insights for readers and practitioners
- Extend evaluation to other real-world applications (e.g., OLAP workloads)
- Extend comparison of DBA-tuned knobs to DNN and DDPG++ approaches
OtterTune in 2022

Commercial Service

Record Label 😊
Questions / Discussion
