Automatic Database Management System Tuning Through Large-scale Machine Learning

Ike Lage, Samantha Pelletier, Braedon Villano
What is a knob?

- Memory for caches
- Flushing frequency
- Buffer pool size
Difficulties

DBMS 1

\[ X \]

\[ \equiv \]

DBMS 2

\[ Y \]

- Standardization

Application 1

\[ X \]

Application 2

\[ X \]

- Independence

- Universality
Tuning Complexity

Ownership Cost of Large-Scale DBMS

DBA Time Spent

Personnel

Everything Else

Tuning

Other

Number of knobs

Release date

MySQL

Postgres

0

200

400

600

2000

2004

2008

2012

2016
Existing Solutions

- Tuning tool A
- Tuning tool B
- Tuning tool C

System specific

Include manual steps

Tuning session 1

Inefficient
Solution: OtterTune
Step 1

Data Repository

Workload Characterization

Step 2

Workload

Lasso

Step 3

Automatic Tuner

Workload Mapping

Configuration Generator
Step 1: Workload Characterization
- Data Repository
- Workload Characterization

Step 2: Lasso
- Lasso

Step 3: Automatic Tuner
- Workload Mapping
- Configuration Generator
Workload Characterization

All metrics

Relevant metrics
Workload Characterization

All metrics → Representation → Low dimensional → Clustering

Metric used downstream
Workload Characterization

Step 1

Data Repository → Workload Characterization → Workload

Step 2

Lasso

Step 3

Automatic Tuner

Workload Mapping → Configuration Generator
Knobs!

1. fanout

2. optimizer

3. log size

4. buffer pool

// TODO: visualize buffer pool
Knobs?

- USELESS
- LEAFSIZE
- CORRELATED
- FANOUT

BTREE LATENCY
Feature Selection (LASSO)
**Step 1**

- Data Repository
- Workload Characterization
- Workload

**Step 2**

- Lasso

**Step 3**

- Automatic Tuner
- Workload Mapping
- Configuration Generator
Workload Mapping

Previous workloads

Current workload

Dataset

Current

Previous
Configuration Recommendation

Dataset

<table>
<thead>
<tr>
<th>Current</th>
</tr>
</thead>
<tbody>
<tr>
<td>Previous</td>
</tr>
</tbody>
</table>

Gaussian Process Regression

Metric

Configurations

Explore
Exploit

Prediction
Uncertainty
But Does it Work?
Efficacy Comparison

Throughput

Postgres

MySQL

Latency

Better

Better
Number of Knobs
Number of Knobs: Fixed vs. Incremental

Fixed Knobs

Lasso

Incremental Knobs

Lasso + 120 Minutes
Number of Knobs: Incremental Tuning is Best

MySQL

![Graph showing 99th percentile time (ms) vs. tuning time (minutes) for different number of knobs. The graph indicates that incremental tuning has a lower 99th percentile time compared to other methods.]

- Blue: 4 knobs
- Green: 8 knobs
- Red: 16 knobs
- Purple: Max knobs
- Yellow: Incremental
Number of Knobs: Incremental Tuning is Best

MySQL

Postgres

Vector

4 knobs  8 knobs  16 knobs  Max knobs  Incremental
Is there a better way to incrementally add knobs?
Future Work