OtterTune
Automatic Database Management System
Tuning Through Large-scale Machine Learning

Dana Van Aken, Andrew Pavlo, Geoffrey J. Gordon, Bohan Zhang
DBMS Tuning

- Tuning a DBMS’s **configuration knobs** knobs for the application’s **workload** and **hardware** is critical for performance
- Tuning even one DBMS deployment is hard...
Challenge #1: Dependencies

MySQL v5.6, YCSB: update-heavy workload
VM - 2GB RAM, 2 vCPUs

99th %-tile latency (sec)
lower is better
Challenge #2: Continuous Settings

MySQL v5.6, YCSB: update-heavy workload
VM - 2GB RAM, 2 vCPUs

99th %-tile latency (sec)
lower is better
Challenge #3: Non-reusable Configs

MySQL v5.6, 3 different YCSB workloads

99th %-tile latency (sec)
lower is better
Challenge #4: Tuning Complexity

Number of configuration knobs in MySQL and Postgres releases over the last 16 years

MySQL

Postgres

2012: 600
2016: 539

2000: 0
2004: 400
2008: 291

↑7x

↑5x
Existing Solutions

Auto-tuners
- Heuristic-based;
- DBMS-specific;

“Smart” Auto-tuners
- Research only;
- no data reuse;
OtterTune

Reuse *historical performance data* from tuning *past* DBMS deployments to tune *new* DBMS deployments

Train ML models to:

1. Identify the most impactful knobs
2. Find past workloads that are similar to the new workload
3. Recommend knob settings that improve a target objective (e.g., latency, throughput)
1. **User specifies the target objective**

2. **Controller starts the first observation period**
   - Measures external metrics, then collects DBMS-specific internal metrics
   - Returns: metrics, current knob configuration

3. **Tuning manager saves the data and computes the next configuration**
   - Returns: next configuration, expected improvement

4. **Controller installs next configuration on DBMS**
Assumptions & Limitations

- All DBMS instances run on the same hardware
- Blacklist for knobs that should NOT be tuned
Machine Learning Pipeline

Workload Characterization

- Metrics
- Samples
- Distinct Metrics

Knob Identification

- Knobs
- Samples
- Importance

Automatic Tuner

- Data
- Graphs
- Analysis
- Tuning
Workload Characterization

• **Goal:** find a set of metrics that best characterize an application’s workload

• **Method:** Use the metrics stored in OtterTune’s data repository
**Internal DBMS Metrics:**

- Directly affected by the knobs’ settings

- **Problem:** redundancy
  - Same but different units
  - Highly correlated

- **Solution:** prune them
  - Factor analysis to capture correlation patterns
  - K-means to group correlated metrics

---

```sql
mysql> SHOW GLOBAL STATUS;
+--------------------------+-----------------+
| METRIC_NAME              | VALUE           |
+--------------------------+-----------------+
| ABORTED_CLIENTS         | 0               |
| ABORTED_CONNECTS        | 0               |
| INNODB_BUFFER_POOL_BYTES_DATA | 129499136     |
| INNODB_BUFFER_POOL_BYTES_DIRTY | 76070912     |
| INNODB_BUFFER_POOL_PAGES_DATA | 7904           |
| INNODB_BUFFER_POOL_PAGES_DIRTY | 4643           |
| INNODB_BUFFER_POOL_PAGES_FLUSHED | 25246         |
| INNODB_BUFFER_POOL_PAGES_FREE | 0             |
| INNODB_BUFFER_POOL_PAGES_MISC | 288            |
| INNODB_BUFFER_POOL_PAGES_TOTAL | 8192           |
| INNODB_BUFFER_POOL_READS | 15327          |
| INNODB_BUFFER_POOL_READ_AHEAD | 0             |
| INNODB_BUFFER_POOL_READ_AHEAD_EVICT | 0   |
| INNODB_BUFFER_POOL_READ_AHEAD_RND | 0          |
| INNODB_BUFFER_POOL_READ_REQUESTS | 2604302     |
| INNODB_BUFFER_POOL_WAIT_FREE | 0             |
| INNODB_BUFFER_POOL_WRITE_REQUESTS | 562763     |
| INNODB_DATA_FSYNCS       | 2836           |
| INNODB_DATA_PENDING_FSYNCS | 1              |
| INNODB_DATA_WRITES       | 28026          |
| UPTIME                   | 5996           |
| UPTIME_SINCE_FLUSH_STATUS | 5996           |
```
Sample Metric Clusters

MySQL (v5.6)

innodb_data_reads
innodb_buffer_pool_reads
innodb_pages_read
...

questions
table_locks_immediate
table_open_cache_hits
...

Postgres (v9.3)

blks_read
heap_blks_read
idx_blks_read
...

buffers_backend
blk_write_time
buffers_clean
...
Knob Identification

• **Goal**: identify which knobs affect the DBMS’s performance

• **Method**: feature selection technique to order the knobs by importance
Configuration Knobs:

- Knobs have varying degrees of impact on the DBMS’s performance
  - Some have high impact
  - Some have no impact
  - For many, it depends on the workload

- Problem: which knobs matter?
- Solution: **Lasso** to determine the order of importance

- Problem: how many to tune?
- Solution: **Incremental knob selection** to gradually increase the number of knobs tuned during a session

```
mysql> SHOW GLOBAL VARIABLES;

+----------------+------------------+
| KNOB_NAME      | KNOB_VALUE       |
+----------------+------------------+
| AUTOCOMMIT     | ON               |
| AUTOMATIC_SP_PRIVILEGES | ON           |
| INNODB_BUFFER_POOL_SIZE | 134217728   |
| INNODB_CHANGE_BUFFERING | all         |
| INNODB_FLUSH_LOG_AT_TRX_COMMIT | 1           |
| INNODB_FLUSH_METHOD | OFF         |
| INNODB_FORCE_LOAD_CORRUPTED | OFF       |
| INNODB_FORCE_RECOVERY | 0           |
| INNODB_LARGE_CAPACITY | 200        |
| INNODB_LARGE_PREFIX | OFF         |
| INNODB_LOCKSUnsafe_FOR_BINLOG | OFF       |
| INNODB_LOCK_WAIT_TIMEOUT | 50         |
| INNODB_LOG_BUFFER_SIZE | 8388608   |
| INNODB_LOG_FILE_SIZE | 5242880   |
| INNODB_LOG_FILES_IN_GROUP | 2           |
| INNODB_LOG_FILE_SIZE | 5242880   |
| INNODB_SORT_BUFFER_SIZE | 2097152   |
| SQL_AUTO_IS_NULL | OFF            |
| TIMED_MUTEXES   | OFF             |
| VERSION_COMPILE_OS | debian-linux-gn |
| WAIT_TIMEOUT   | 28800           |
+----------------+------------------+
```
Top 5 Configuration Knobs

MySQL (v5.6)
1. innodb_buffer_pool_size
2. innodb_flush_method
3. innodb_log_file_size
4. innodb_thread_concurrency
5. innodb_max_dirty_pages_pct_lwm

Postgres (v9.3)
1. shared_buffers
2. checkpoint_segments
3. effective_cache_size
4. bgwriter_lru_maxpages
5. bgwriter_delay
Automated Tuning

• Step 1: Workload Mapping
• Step 2: Configuration Recommendation
Step 1: Workload Mapping

- **Goal**: Match the target DBMS’s workload to the most similar known workload in the data repository

- **Method**: For each known workload, compute a similarity score by comparing the performance measurements
**Similarity score**: average distance in performance measurements over all metrics

- Problem: computing the score with raw metrics would be unfair
- Solution: compute deciles, bin values
- Problem: little overlap in the configurations tried between workloads
- Solution: for each metric, train statistical model to predict performance

<table>
<thead>
<tr>
<th>Workload A</th>
<th>Workload B</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>p99 latency</code></td>
<td><code>throughput</code></td>
</tr>
<tr>
<td>792</td>
<td>512</td>
</tr>
<tr>
<td>4</td>
<td>5</td>
</tr>
<tr>
<td>8</td>
<td>2</td>
</tr>
<tr>
<td>3</td>
<td>5</td>
</tr>
</tbody>
</table>

**Bin metric values**

**Target DBMS’s Workload**

**Known Workloads**
Step 2: Recommendation

• **Goal**: recommend configurations in order to optimize the target metric over the entire tuning session

• **Method**: Reuse data from the similar workload in step #1 to train a statistical model and use it to optimize the next configuration to try
Recommendation Steps

- Use similar historical data & new data to train a Gaussian Process model
- Predict the latency & variance for a sample set of possible configurations
- Optimize the next configuration to try, trading off:
  - **Exploration**: gathering information to improve the model
  - **Exploitation**: greedily trying to do well on the target objective
Experimental Setup

DBMSs: MySQL (v5.6), Postgres (v9.3)

Training data collection:
- 15 YCSB workload mixtures
- ~30k trials per DBMS

Experiments conducted on Amazon EC2
Data vs. No Data

99th %-tile latency (sec)
lower is better

OtterTune
iTuned

MySQL (v5.6)
Postgres (v9.3)

TPC-C
Wikipedia

2 hours of tuning time

46% 40%
30%
40% 42%

lower is better
MySQL: Efficacy Comparison

OtterTune generates a knob configuration comparable to the DBA!
Postgres: Efficacy Comparison

OtterTune generates a knob configuration comparable to the DBA!
Future Work

• Automatically detecting the hardware capabilities of the DBMS’s host machine
• Auto-tuning hierarchical components
• Multi-objective optimization
THE END

@danavanaken