Speedup your Analytics: Automatic Parameter Tuning for Databases and Big Data Systems

Jiaheng Lu, University of Helsinki
Yuxing Chen, University of Helsinki
Herodotos Herodotou, Cyprus University of Technology
Shivnath Babu, Duke University / Unravel Data Systems
Outline

Motivation and Background

History and Classification

Parameter Tuning on Databases

Parameter Tuning on Big Data Systems

Applications of Automatic Parameter Tuning

Open Challenges and Discussion
What and How to Tune?

➢ What to configure?
  ❖ Which parameters (knobs)?
  ❖ Which are most important?

➢ How to tune (to best throughput)?
  ❖ Increase buffer size?
  ❖ More parallelism on writing?

Figure. Tuning guitar knobs to right notes (frequencies)
### What to Tune – Some Important Knobs for throughput

<table>
<thead>
<tr>
<th>Parameter Name</th>
<th>Brief Description and Use</th>
<th>Default</th>
</tr>
</thead>
<tbody>
<tr>
<td>bgwriter_delay</td>
<td>Background writer’s delay between activity rounds</td>
<td>200ms</td>
</tr>
<tr>
<td>bgwriter_lru_maxpages</td>
<td>Max number of buffers written by the background writer</td>
<td>100</td>
</tr>
<tr>
<td>checkpoint_segments</td>
<td>Max number of log file segments between WAL checkpoints</td>
<td>3</td>
</tr>
<tr>
<td>checkpoint_timeout</td>
<td>Max time between automatic WAL checkpoints</td>
<td>5min</td>
</tr>
<tr>
<td>deadlock_timeout</td>
<td>Waiting time on locks for checking for deadlocks</td>
<td>1s</td>
</tr>
<tr>
<td>default_statistics_target</td>
<td>Default statistics target for table columns</td>
<td>100</td>
</tr>
<tr>
<td>effective_cache_size</td>
<td>Effective size of the disk cache accessible to one query</td>
<td>4GB</td>
</tr>
<tr>
<td>shared_buffers</td>
<td>Memory size for shared memory buffers</td>
<td>128MB</td>
</tr>
</tbody>
</table>
What are the Important Parameters and How to Choose

➢ Affect the performance most (manually)
  ❖ Based on expert experiences
  ❖ Default documentation

Parameters have strong correlation to performance are important!

Performance-sensitive parameters are important!

If you want higher throughput, better tuning memory-related parameters
What are the Important Parameters and How to Choose

- Affect the performance most
- Strongest correlation between parameters and objective function (model)
  - Linear regression model for independent parameters:
    - Regularized version of least squares – Lasso (*OtterTune 2017*)
      - Interpretable, stable, and computationally efficient with higher dimensions
  - Deep learning model (*CBDTune 2019*)
    - The important input parameters will gain higher weights in training

\[ Y = \beta_0 + \beta_1 X_1 + \ldots + \beta_N X_N \]
How to Tune – Key Tuning Goals

➢ **Avoidance**: to identify and avoid error-prone configuration settings

➢ **Ranking**: to rank parameters according to the performance impact

➢ **Profiling**: to classify and store useful log information from previous runs

➢ **Prediction**: to predict the database or workload performance under hypothetical resource or parameter changes

➢ **Tuning**: to recommend parameter values to achieve objective goals
### How to Tune – Tuning Methods

<table>
<thead>
<tr>
<th>Methods</th>
<th>Approach</th>
<th>Methodology</th>
<th>Target Level</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rule-based</td>
<td>SPEX (2013)</td>
<td>Constraint inference</td>
<td>Avoidance</td>
</tr>
<tr>
<td></td>
<td>Xu (2015)</td>
<td>Configuration navigation</td>
<td>Ranking</td>
</tr>
<tr>
<td>Cost-model</td>
<td>STMM (2006)</td>
<td>Cost model</td>
<td>Tuning</td>
</tr>
<tr>
<td>Simulation-based</td>
<td>Dushyanth (2005)</td>
<td>Trace-based simulation</td>
<td>Prediction</td>
</tr>
<tr>
<td></td>
<td>ADDM (2005)</td>
<td>DAG model &amp; simulation</td>
<td>Profiling, tuning</td>
</tr>
<tr>
<td>Experiment driven</td>
<td>SARD (2008)</td>
<td>P&amp;B statistical design</td>
<td>Ranking</td>
</tr>
<tr>
<td></td>
<td>iTuned (2009)</td>
<td>LHS &amp; Guassian Process</td>
<td>Profiling, tuning</td>
</tr>
<tr>
<td>Machine Learning</td>
<td>Rodd (2016)</td>
<td>Neural Networks</td>
<td>Tuning</td>
</tr>
<tr>
<td></td>
<td>OtterTune (2017)</td>
<td>Guassian Process</td>
<td>Ranking, tuning</td>
</tr>
<tr>
<td></td>
<td>CDBTune (2019)</td>
<td>Deep RL</td>
<td>Tuning</td>
</tr>
</tbody>
</table>
Relational Database Tuning Methods

 Covered #knobs
 Training cost

 2005  2017

 Time

 Rule-based
 Machine Learning
 Experiment-driven
 Simulation-based
 Cost Modeling
 Rule-based

 Figure. Developing trend: putting more training cost to uncover more knobs

 Rule-based
 Cost Modeling
 Adaptive
 Simulation-based
 Experiment-driven
 Machine Learning

 Figure. Required expert knowledge on system
Tuning Method: Rule-based

- Tuning based on rules derived from DBAs’ expertise, experience, and knowledge, or Rule of Thumb default recommendation

- Guarantee cache memory to accelerate queries ...
- Better not change the deadlock timeout if ...
- Default settings work most of the cases ...

Rules → Expert

Trusted Parties → Rule of Thumb → Documents
A cost model establishes a performance model by cost functions based on the deep understanding of system components.
Tuning Method: Cost Modeling (STMM)

- **STMM: Adaptive Self-Tuning Memory in DB2 (2006)**
  - Reallocates memory for several critical components (e.g., compiled statement cache, sort, and buffer pools)
Tuning Method: Simulation-based

A simulation-based approach simulates workloads in one environment and learns experience or builds models to predict the performance in another.

- Running job here is (1) expensive or (2) slowdown concurrent jobs or (3)...
- Simulate it in small environment with tiny portion of data …

- Often product environment
- Often test environment
Tuning Method: Experiment-driven

- An experiment-driven approach relies on repeated executions of the same workload under different configuration settings towards tuning parameter values.

![Diagram of tuning process]

Classic paper: Tuning Database Configuration Parameters with iTuned. 2009
Tuning Method: Machine Learning

Machine Learning (ML) approaches aim to tune parameters automatically by taking advantages of ML methods.

Input knobs ➔ ML Training ➔ ML Model ➔ Recommended knobs

Goal ➔ Training logs ➔ Actual run logs
Tuning Method: Machine Learning (OtterTune 2017)

- **Factor Analysis**: transform high dimension parameters to few factors
- **Kmeans**: Cluster distinct metrics
- **Lasso**: Rank parameters
- **Gaussian Process**: Predict and tune performance

Figure. OtterTune system architecture
Tuning Method: Machine Learning (CDBTune 2019)

- **Reinforcement learning**
  - **State:** knobs and metrics
  - **Reward:** performance change
  - **Action:** recommended knobs
  - **Policy:** Deep Neural network

- **Key idea**
  - Feedback: try-and-error method
    - Recommend -> good/bad
  - Deep deterministic policy gradient
    - Actor critic algorithm

**Reward:** Throughput and latency performance change $\Delta$ from time $t - 1$ and the initial time to time $t$
Tuning Method: Adaptive

- An adaptive approach changes parameter configurations online as the environment or query workload changes.

![Diagram](image)

Figure. CLOT (2006) strategy
The Differences of Tuning Database & Big Data Systems in research papers

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Relational Database</th>
<th>Big Data System</th>
</tr>
</thead>
<tbody>
<tr>
<td>More parameters on</td>
<td>memory</td>
<td>More parameters on vcores</td>
</tr>
<tr>
<td>Resource</td>
<td>Often fixed resources</td>
<td>Now more varying resources</td>
</tr>
<tr>
<td>Scalability</td>
<td>Often single machine</td>
<td>Often many machines in a distributed environment</td>
</tr>
<tr>
<td>Metrics</td>
<td>Throughput, latency</td>
<td>Time, resource cost</td>
</tr>
</tbody>
</table>
References (1/1)


➢ Dias, K., Ramacher, M., Shaft, U., Venkataramani, V., & Wood, G.. Automatic Performance Diagnosis and Tuning in Oracle. In CIDR (pp. 84-94), 2005


