Adaptive Query Scheduling for Mixed Database Workloads with Multiple Objectives

Stefan Krompass\textsuperscript{TUM}, Harumi Kuno\textsuperscript{HPL},
Kevin Wilkinson\textsuperscript{HPL}, Umeshwar Dayal\textsuperscript{HPL}, Alfons Kemper\textsuperscript{TUM}

\textsuperscript{TUM}Technische Universität München
Munich, Germany

\textsuperscript{HPL}Hewlett-Packard Laboratories
Palo Alto, CA, USA
Problem statement

- $n$ service classes (i.e., a set of queries)
- $n \cdot m$ objectives (multiple objectives per service class)
- $n \cdot k$ control knobs (to control service per class, e.g., MPL)

Search problem

Find control knobs settings to achieve objectives for all service classes
Problem statement

- $n$ service classes (i.e., a set of queries)
- $n \cdot m$ objectives (multiple objectives per service class)
- $n \cdot k$ control knobs (to control service per class, e.g., MPL)

Search problem

Find control knobs settings to achieve objectives for all service classes
Difficulties

- **Large search space**
- Queries have different characteristics (resource requirements, variance in resource requirements)
- Service classes have different characteristics (start time, arrival rate, objectives)
- Contention among the queries unknown
- Non-linear relationships between objectives and the control parameters
Difficulties

- **LARGE SEARCH SPACE**
- Queries have different characteristics (resource requirements, variance in resource requirements)
- Service classes have different characteristics (start time, arrival rate, objectives)
- Contention among the queries unknown
- Non-linear relationships between objectives and control parameters

In this presentation: Present framework and experiments with algorithm to tackle the search problem
Solution approach

- Base: algorithm devised by Niu et. al: “Adapting Mixed Workloads to Meet SLOs in Autonomic DBMSs”
  ⇒ Multi-class, single objective
- Extension: assume relationship between objectives is known in order to solve our problem
Workload Adaptation-Maximize Single Objective

- Goal: maximize overall utility (measure to quantify how well the system meets the objectives)
- Service classes \( s_1, \ldots, s_n \), each with single objective
- Idea: assign system resources to service classes by controlling the number of queries a service class may run
- Service class \( s_i \) has control knob \( x_i \)
- Assumption: \( \exists \) “system cost limit” \( X \) where performance is maximized
Workload Adaptation-Maximize Single Objective

\[
\text{maximize } \prod_{i=1}^{n} u_i \left( h_i(x_i) \right) + \cdots + u_n \left( h_n(x_n) \right)
\]

subject to \( x_1 + \cdots + x_n = X \)

- **Estimation model:** control knob setting \((x_i) \rightarrow \) estimated performance
- **Utility function:** performance value \( \rightarrow \) utility (positive if performance \( \geq \) objective, negative otherwise; utility decrease faster for lower performance, utility increase slower with better performance)
Dominance

**Definition**

Objective $o$ is *dominant* for a service class if a set of conditions satisfying $o$ implies that the other objectives of this service class are satisfied as well.

**Note**

- Dominance holds only for a specified range of control knob settings
- Dominance applies to objectives of a single service class only

**Example**

If average response time requirement is satisfied, throughput is also
Framework

- Workload objectives
- Workload manager
  - Admission controller
  - Scheduler
  - Execution controller
- Policy controller
- Policy control loop
- DBMS

- Queries
- Service level objectives
Framework

- Workload objectives
- Workload manager
  - Admission controller
  - Scheduler
  - Execution controller

Policy control loop

Simulated DBMS

- Policy controller
- Workload
  - Queries
    - Service level objectives
Why a simulator?

- Deterministic
- Repeatable results
- Experiment with varying workloads with varying characteristics
- Easily change system configuration
- Speedup
Experiments

Purpose of the experiments

- Two service classes, each with throughput and average response time objectives
- Control knobs: vary MPL for each service class
- Goal: find MPL settings where each objective is met
Experiments

Experimental setup

- Database engine models a parallel, shared-nothing architecture; four nodes with eight disks
- Data is partitioned across the disks
- More details on simulated engine in the paper
- Multiple streams per service class, each stream sends queries one after the other with no wait time between two queries
- OLTP-style queries; a query accesses data on a single partition only
# Experiment 1

<table>
<thead>
<tr>
<th></th>
<th>Service class 1</th>
<th>Service class 2</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Average response time (sec)</strong></td>
<td>0.25</td>
<td>1.0</td>
</tr>
<tr>
<td><strong>Throughput (q/sec)</strong></td>
<td>130</td>
<td>80</td>
</tr>
<tr>
<td><strong>Dominant objective</strong></td>
<td>throughput</td>
<td>throughput</td>
</tr>
<tr>
<td><strong>Algorithm optimizes for</strong></td>
<td>throughput</td>
<td>throughput</td>
</tr>
</tbody>
</table>
Results

overall search space
Results

Overall search space

Objectives of class 1 met, objectives of class 2 violated

Objectives of class 2 met, objectives of class 1 violated

Objectives of classes 1 and 2 met
Results
search space considered by workload adaptation-MSO
Results
workload adaptation-MSO

[Diagram showing time vs. MPL for different workloads]
Results

workload adaptation-MSO
## Experiment 2

<table>
<thead>
<tr>
<th></th>
<th>Service class 1</th>
<th>Service class 2</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Average response time (sec)</strong></td>
<td>0.25</td>
<td>0.6</td>
</tr>
<tr>
<td><strong>Throughput (q/sec)</strong></td>
<td>100</td>
<td>80</td>
</tr>
<tr>
<td><strong>Dominant objective</strong></td>
<td>average response time</td>
<td>throughput</td>
</tr>
<tr>
<td><strong>Algorithm optimizes for</strong></td>
<td>throughput</td>
<td>throughput</td>
</tr>
</tbody>
</table>
Results

daïve
Results

search space considered by workload adaptation-MSO
Results

search space considered by workload adaptation-MSO

Cannot find setting in operating envelope with MPL sum = 32
⇒ increase to 48
Results

search space considered by workload adaptation-MSO
Results
workload adaptation-MSO
Results

workload adaptation-MSO

MPL service class 1

MPL service class 2
Results
workload adaptation-MSO

Solution exists in the search space but algorithm does NOT find it
Results

workload adaptation-MSO

Throughput (in queries per second)

Time (in seconds)

throughput okay
Results
workload adaptation-MSO

![Graph showing average response time of service class $s_2$ violated.](image)
Conclusion and ongoing work

- Presentation of test framework
- Comprehensive search solves the search problem, and gives additional information: Does a solution exist? How many settings satisfy the constraints? → prohibitively expensive
  ⇒ Need heuristic approach
- Solutions found by *workload adaptation-MSO* are “fragile”
  ⇒ Need different set of algorithms to solve the search problem