Performance Management for Cloud Databases via Machine Learning

Olga Papaemmanouil
Brandeis University
Outline

Motivation

Offline Learning

Online Learning

Conclusions

- Cloud Databases
- Challenges
- State-of-the-Art
- Why Machine Learning?
- WiSeDB Advisor
Cloud Computing

Paradigm shift: infrastructure, data processing

- economies of scale
- capital expenditure
- pay-as-you-go
Cloud Databases Landscape

- **Database-as-a-Service**
  - Managed DBMS
  - Relational & NoSQL DBs

- **IaaS-deployed DBMSs**
  - Non managed DBMS
  - DIY model

**Cloud**

- Amazon Aurora
- Microsoft SQL Azure
- Oracle Cloud Database
- Amazon RDS

**Infrastructure as a Service (IaaS)**

- Rackspace
- Amazon Web Services
- Azure
- Google Compute Engine
IaaS-deployed Databases

Management Tools
- Monitoring resources, performance, cost
- Event-driven scaling

Data Management Application
- Microsoft SQL Server
- MySQL
- PostgreSQL
- Oracle

IaaS Provider
- VM
- VM
- VM
- VM

OpsWorks

Trusted Advisor
AWS Cloud Optimization Expert

StackDriver Monitoring

CloudWatch
Deployment Challenges

Data Management Application

Custom-built application management tools

IaaS Provider

VM

- SQL Server
- MySQL
- PostgreSQL
- Oracle
Deployment Challenges

SLO (objective metric)
- Query-level: response time
- Workload level: average, total, max, percentile

SLA Fees
- Violation penalties

Data Management Application
- Cost Management
- SLA Management

Pay-as-you-go Model

IaaS Provider

SQL Server, MySQL, PostgreSQL, Oracle
Deployment Challenges

Data Management Application
- Cost Management
- SLA Management
- Resource Provisioning
- Workload Scheduling

Beyond monitoring & alerts
- Automation: scale up & down
- Query routing & scheduling
- Cost-driven management
- SLA-awareness

NP-hard problem
## State-of-the-art

<table>
<thead>
<tr>
<th>Placement</th>
<th>Provisioning</th>
<th>Scheduling</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>PMAX</strong> (Liu et al.)</td>
<td><strong>Auto</strong> (Rogers et al.)</td>
<td><strong>Dolly</strong> (Cecchet et all)</td>
</tr>
<tr>
<td><strong>SLATree</strong> (Chi et al.)</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Multi-tenant SLOs</strong> (Lang et al.)</td>
<td></td>
<td><strong>iCBS</strong> (Chi et al.)</td>
</tr>
<tr>
<td><strong>Delphi / Pythia</strong> (Elmore et al.)</td>
<td><strong>Hypergraph</strong> (Çatalyürek et al.)</td>
<td></td>
</tr>
<tr>
<td><strong>SCOPE</strong> (Chaiken et al.)</td>
<td><strong>Bazaar</strong> (Jalaparti et al.)</td>
<td><strong>many traditional methods ...</strong></td>
</tr>
</tbody>
</table>
## State-of-the-art

<table>
<thead>
<tr>
<th>Placement</th>
<th>Provisioning</th>
<th>Scheduling</th>
</tr>
</thead>
<tbody>
<tr>
<td>PMAX (Liu et al.)</td>
<td>Auto (Rogers et al.)</td>
<td>Dolly (Cecchet et all)</td>
</tr>
<tr>
<td>SLATree (Chi et al.)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Multi-tenant SLOs (Lang et al.)</td>
<td>iCBS (Chi et al.)</td>
<td></td>
</tr>
<tr>
<td>Delphi / Pythia (Elmore et al.)</td>
<td>Hypergraph (Çatalyürek et al.)</td>
<td></td>
</tr>
<tr>
<td>SCOPE (Chaiken et al.)</td>
<td>Bazaar (Jalaparti et al.)</td>
<td>many traditional methods ...</td>
</tr>
</tbody>
</table>
Wish List

Requirements

End-to-end cost-aware service
(resource provisioning, workload scheduling)

Application-defined performance goals
(per query deadline, percentile, average latency, max latency)

Agnostic to workload semantics

Why ML?

complex interactions
arbitrary goals
arbitrary workloads
WiSeDB Advisor

**Offline Learning**
- batch scheduling
- performance vs cost exploration

**Online Learning**
- online scheduling
- performance model free
Outline

- Motivation
- Offline Learning
- Online Learning
- Conclusions

- System Overview
- Supervised Learning
- Adaptive Learning
Offline Learning

**Data Management Application**

- (Offline) Training
- Model Generator

**Workload Spec**
- Q1: 2min
- Q2: 0.5min

**SLO Spec (Deadline)**
- Q1: 3min
- Q2: 1min

**IaaS Provider**

**Base**
- Microsoft SQL Server
- MySQL®
- PostgreSQL
- Oracle®
Offline Learning

Original SLO
- Q1: 3min, $0.12/Q1
- Q2: 1min, $0.2/Q2

Relaxed SLO
- Q2: 4min, $0.05/Q1
- Q2: 2min, $0.1/Q2

Stricter SLO
- Q1: 2.5min, $0.15/Q1
- Q2: 0.7min, $0.13/Q2

Data Management Application

- (Offline) Training
  - Model Generator
  - Strategy Recommendations

IaaS Provider

- Microsoft SQL Server
- MySQL
- PostgreSQL
- Oracle
Offline Learning

Relaxed SLO
- Q1: 4min, $0.05/Q1
- Q2: 2min, $0.1/Q2

Data Management Application

(OFFLINE) Training
- Model Generator
- Strategy Recommendations

(ONLINE) Resource & Workload Management
- Strategy Generator

Resources to rent
- 2 VMs of Type A
- 3 VMs of Type B

Query scheduling
- VM₁ (Type A) queue
  - Q₁, Q₁, Q₂, Q₂, Q₂,…
- VM₂ (Type B) queue
  - Q₂, Q₂, Q₂, Q₁, Q₁,…

Q1 x 10
Q2 x 200

IaaS Provider
Supervised Learning

identify classes

create training data

generate classifier

classes == actions

- dispatch a query to a VM
- provision new VM

context of actions

- identify best decisions
- extract cost-related features

decision tree

- describe (context, action)
- verifiable & interpretable

Model Generator
"To be the best, learn from the best" (D. LaCroix)

**Offline Learning**
- identify best decisions
  1. Generate small workload
  2. Build decision graph
     - query assignment
     - VM provisioning
  3. Find optimal (minimum cost) solution (path)
  4. Extract context of optimal step-by-step decisions

**Model Generator**

**Runtime Scheduling**
- apply model
  1. Repeat for many sample workloads
  2. Build a training set of (feature, action)
  3. Train a classifier

- Use classifier for
  - batch scheduling
  - online scheduling
  - performance vs cost exploration
Search for Optimal

Cost Model
- Resource usage
  - VM start up time + query execution time
- Violation fees
- Penalty function
Search for Optimal

A* search for optimal

Model Generator
Search for Optimal

A* search for optimal

Graph-based approach Pros
- Step-by-step decisions
- Graph reduction techniques
- Fast search for optimal

Model Generator
Feature Extraction

Agnostic to
- Query semantics
- Performance goal (SLO)
- Workload size

Decision: Assign to VM

Features:
- unassigned : true
- unassigned : false
- cost of assigning : $0.2
- wait time on VM: 20sec
- % of in VM: 0%
- % of in VM: 0%
Decision Model

wait time?  

>=2 

new VM 

<2 

is unassigned?  

true 

false 

cost of assign?  

<100 

assign 

>=100 

assign 

is unassigned?  

true 

false 

assign 

new VM
Decision Model

wait time?  
>=2  
<2  
new VM  
is unassigned?  
true  
false  
cost of assign?  
<100  
>=100  
assign  
is unassigned?  
true  
false  
assign  
new VM  
Strategy Generator
Decision Model

- Wait time? 
  - \( \geq 2 \)
  - \( < 2 \)

  - New VM
  - Is unassigned?
    - True
    - False
      - Cost of assign? 
        - \( < 100 \)
        - \( \geq 100 \)
          - Assign
          - Is unassigned?
            - True
            - False
              - Assign
              - New VM
Experimental Setup

**Training Data**
- 3000 samples
- 10 TPC-H templates
- 18 queries/sample

The chart shows the cost (cents) for different query execution times. The performance goal is that query execution time should be less than or equal to x seconds (same deadline per template).
Experimental Setup

**Training Data**
- 3000 samples
- 10 TPC-H templates
- 18 queries/sample

![Graph showing Cost (cents) for PerQuery, Average, Max, and Percent with WiSeDB, Optimal, and more cost](image)

average latency of the workload $\leq x$ secs
Experimental Setup

**Training Data**

- 3000 samples
- 10 TPC-H templates
- 18 queries/sample

The figure shows a bar chart with the following categories:

- **PerQuery**
- **Average**
- **Max**
- **Percent**

The y-axis represents the **Cost (cents)**, ranging from 0 to 60. The x-axis represents the different performance metrics. The chart includes two categories for comparison:

- **WiSeDB**
- **Optimal**

The optimal performance goal is defined as the **max latency <= x secs** (longest query in the workload).
Experimental Setup

**Training Data**
3000 samples
10 TPC-H templates
18 queries/sample

![Graph showing cost in cents across different execution times for WiSeDB and Optimal]

- PerQuery
- Average
- Max
- Percent

*execution time of 90% of queries in the workload <= x secs*
Experimental Setup

**Training Data**
- 3000 samples
- 10 TPC-H templates
- 18 queries/sample

**Testing Data**
- 10 TPC-H templates
- varied queries/workload
Experimental Setup

**Training Data**
3000 samples
10 TPC-H templates
18 queries/sample

**Testing Data**
10 TPC-H templates
varied queries/workload

cost: resource utilization + penalties

**AWS Cloud**
fees penalty $0.01/sec of violation
Effectiveness (small workloads)

**Training Data**
- 3000 samples
- 10 TPC-H templates
- 18 queries/sample

**Testing Data**
- 10 TPC-H templates
- 30 queries/workload
  *Optimal: Brute force*

WiSeDB models are within 8% of the minimum cost solution
Effectiveness (large workloads)

Training Data
3000 samples
10 TPC-H templates
18 queries/sample

Testing Data
10 TPC-H templates
5000 queries/workload

One heuristic cannot fit all

WiSeDB learns the right heuristic

Best: shortest query first
Best: longest query first
Best: top-90% shortest then 10% longest queries
Offline Learning

Original SLO
- Q1: 3min, $0.12/Q1
- Q2: 1min, $0.2/Q2

Relaxed SLO
- Q1: 4min, $0.05/Q1
- Q2: 2min, $0.1/Q2

Stricter SLO
- Q1: 2.5min, $0.15/Q1
- Q2: 0.7min, $0.13/Q2

Data Management Application
- (Offline) Training
  - Model Generator
  - Strategy Recommendations
- (Online) Performance Management

IaaS Provider
- VM
- SQL Server
- MySQL
- PostgreSQL
- Oracle

Q1
Q2
Strategy
Recommendations

- new SLO
- new scheduling graph
- new optimal decisions (path)
- new model

expensive (brute force/sample)

echange only the SLO & **reuse** the original graph

deployment of performance vs cost trade offs
Adaptive Modeling

Fast search with A* best-first search

explore-first heuristic:
\[ \min \{g(n) + h(n)\} \]

cost so far

lower bound for cost to a goal node

Strategy Recommendations

- tighter SLAs cost more
  - old cost < new cost
  - \( h(n) = \) old optimal cost

- tighter SLAs give faster search
  - better heuristic
  - no graph generation
Adaptive Training

**Training Data**
- 3000 samples
- 10 TPC-H templates
- 18 queries/sample
- 15% stricter SLA

*Scratch*: training a new model  
*Adaptive*: adapting the original model

Adaptive training time is 96-94% less than original training time
Performance vs Cost Exploration

WiSeDB generates models for 10s of alternative SLOs within secs
- Keeps k-top significant ones
  - Earth Mover’s Distance
- No query execution is required

Model estimates cost/template & expected performance
- Assumes a given cost model

User picks desired model

<table>
<thead>
<tr>
<th>SLO Type</th>
<th>Q1</th>
<th>Q2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Original SLO</td>
<td>3min, $0.12/Q1</td>
<td>1min, $0.2/Q2</td>
</tr>
<tr>
<td>Relaxed SLO</td>
<td>4min, $0.05/Q1</td>
<td>2min, $0.1/Q2</td>
</tr>
<tr>
<td>Stricter SLO</td>
<td>2.5min, $0.15/Q1</td>
<td>0.7min, $0.13/Q2</td>
</tr>
</tbody>
</table>
Online Scheduling

• Scheduling & provisioning for one query at a time

• Batch-based models not effective for online tasks
  • Do not account for query arrival rate/wait times

• WiSeDB approach
  • Generate a new model upon arrival of new query
  • Adapt previous model to reduce training overhead
  • Reuse past models, when feasible
Online Scheduling

### Workload Spec
- **Q1**: 2min

### SLO Spec
- **Q1**: 3min

### Workload Spec
- **Q2**: (2+0.5)min
- **Q3**: 2min

### SLO Spec
- **Q2**: 3min
- **Q3**: 3min

- training batch: new query + queued queries
- add wait time in expected latency
- slow for high arrival rates

---

Strategy Generator

- Online Scheduling
  - training batch: new query + queued queries
  - add wait time in expected latency
  - slow for high arrival rates
Online Scheduling

- **Model Reuse**: reuse model with similar expected latencies/template
- **Linear Shifting**: treat as a tightened SLA

### Workload Spec
- **Q1**: 2min
- **Q2**: (2+0.5)min
- **Q3**: 2min

### SLO Spec
- **Q1**: 3min
- **Q2**: 3min
- **Q3**: 3min

---

![Timeline Diagram with VMs and QoS SLA](image-url)
Effectiveness (online scheduling)

Testing Data

30 queries/workload
10% from optimal

Query wait time < 1 sec

WiSeDB can leverage existing models to offer effective scheduling in a online manner
Offline Learning

Advantages

- Abstracts away complex decisions
- Generates custom heuristics per application
- Explores Performance vs Cost trade-offs

Data Management Application

(OFFLINE) Training
- Model Generator
- Strategy Recommendations

(ONLINE) Resource & Workload Management
- Strategy Generator

IaaS Provider

VM

Microsoft SQL Server, MySQL, PostgreSQL, Oracle
Offline Learning

Limitations
- Static models
- Batch scheduling
- Known cost model

Data Management Application

(OFFLINE) Training
- Model Generator
- Strategy Recommendations

(ONLINE) Resource & Workload Management
- Strategy Generator

IaaS Provider

Database Providers:
- Microsoft SQL Server
- MySQL
- PostgreSQL
- Oracle
Outline

- Motivation
- Offline Learning
  - Explicit vs Implicit Modeling
  - Reinforcement Learning
- Online Learning
- Conclusions
(Explicit) Performance Prediction

- DBMS-related challenges
  - isolated vs. concurrent query execution
  - low accuracy for new query types ("templates")
  - extensive off-line training
  - state-of-the-art: 15-20% prediction error

- Cloud-related challenges
  - "noisy neighbors"
  - numerous resource configurations
  - predictions errors accumulation
WiSeDB: Implicit Performance Modeling

- Explicit performance models are NOT necessary for:
  - monetary cost management
  - resource & workload management
  - offer performance SLA and keep penalties low

Wish List #2

- Implicitly model query latency
  - predict monetary cost ( & violation penalties)
- Online training for dynamic environments
  - Automatic scaling & workload distribution
Reinforcement Learning

- Continuous learning
- Explicit reward modeling
- Action selection
  - maximize reward

Environment

agent

action

reward

internal state
(past experiences)

observation

Environment
CMABs
(Contextual Multi-Armed Bandits)

Contextual Multi-Armed Bandit Problem

Armed Bandit = Slot Machine

Which slot machine to play (action) so that you walk out with the most $$$ (reward)?
CMABs in WiSeDB
(Contextual Multi-Armed Bandits)

**Contextual Multi-Armed Bandit Problem**

Slot Machine = Virtual Machine

*Which machine to use (new/old) (action) so that you execute the incoming query with minimum cost $$ (cost)?*
CMABs in WiSeDB
(Contextual Multi-Armed Bandits)

Action (per VM)
- Accept
- Pass to next /new VM
- Down one VM type

Reward
- $$ cost: processing & SLA violation penalties

Observation
- environment context (query, VM)
- action
- $$ cost

Data Management Application

SLA
internal state (past experiences)

action
cost $$
observation

IaaS Provider

VM Tier 1

VM

VM

Tier 2

VM

VM

VM
CMABs in WiSeDB
(Contextual Multi-Armed Bandits)

Action (per VM)
- Accept
- Pass to next/new VM
- Down one VM type

Reward
- $$ cost: processing & SLA violation penalties

Observation
- environment context (query, VM)
- action
- $$ cost

Data Management Application
- action
- cost $$
- observation

IaaS Provider
- VM Tier 1: pass, down
- VM Tier 2: accept

SLA
internal state (past experiences)
CMABs in WiSeDB
(Contextual Multi-Armed Bandits)

**Action (per VM)**
- Accept
- Pass to next/new VM
- Down one VM type

**Reward**
- $$ cost: processing & SLA violation penalties

**Observation**
- environment context (query, VM)
- action
- $$ cost

---

**Data Management Application**

- **SLA**
- **internal state** (past experiences)
- **action**
- **cost $$**
- **observation**

---

**IaaS Provider**

- Tier 1
- Tier 2
CMABs in WiSeDB
(Contextual Multi-Armed Bandits)

- **Action (per VM)**
  - Accept
  - Pass to next / new VM
  - Down one VM type

- **Reward**
  - $$ cost: processing & SLA violation penalties

- **Observation**
  - environment context (query, VM)
  - action
  - $$ cost

---

**Data Management Application**

*SLA*

(pass, context, $$)

(down, context, $$)

(accept, context, $)

*action*

*cost $$*

*observation*

---

**IaaS Provider**

VM Tier 1

VM Tier 2
CMABs in WiSeDB
(Contextual Multi-Armed Bandits)

Action (per VM)
- Accept
- Pass to next /new VM
- Down one VM type

Reward
- $$ cost: processing & SLA violation penalties

Observation
- environment context (query, VM)
- action
- $$ cost

Data Management Application

- SLA
- action
- cost $$
- observation

IaaS Provider

Tier 1
- VM
- pass

Tier 2
- VM
- pass

- accept
Online Learning

Context Features
- VM context
  - memory, I/O rate
  - # queries in queue
- Query context
  - tables used
  - # table scans
  - # joins
  - # spill joins
  - cache reads

Data Management Application
- Model Generator
- Context Collector
- Experience Collector

IaaS Provider

Databases:
- Microsoft SQL Server
- MySQL
- PostgreSQL
- Oracle
Online Learning

**Action Selection**
- **Explore** opportunities
  - gather information
- **Exploit** “safe” actions
  - make best decision given current information
- Thompson sampling

**Data Management Application**

- **Model Generator**
  - action

- **Context Collector**

- **Experience Collector**

IaaS Provider

- VM
- VM
- VM
- VM

Database Systems:
- Microsoft SQL Server
- MySQL®
- PostgreSQL
- Oracle®
Probabilistic Action Selection

- Select action according to probability of being the best
- Past observations $D = \{(x_i, a_i, c_i)\}$
  - modeled by likelihood function over cost $c : P(c | \alpha, x, \theta)$
  - $\theta$: parameters of likelihood function: splits of a regression tree
    - if (# joins in the query = 1) and (queries in the queue = 3) => cost = $$

- Posterior distribution of $\theta$ (Bayes rule)
  - $P(\theta | D) \propto \prod P(c_i | a_i, x_i, \theta)P(\theta)$
  - $P(\theta)$: prior distribution of parameters $\theta$

- Choose action $\alpha'$ to minimize cost for perfect model $\theta^*$

$$\min_{a'} \mathbb{E}(c | a', x, \theta^*)$$
Probabilistic Action Selection

- Exploitation: pick action based on mean of posterior $P(\theta|D)$
  $$\min_{a'} E(c \mid a', x) = \int E(c \mid a', x, \theta) P(\theta \mid D) d\theta$$

- Exploration: pick a random action

- Thompson Sampling: balance exploration/exploitation

  Select random action according to probability that it is the best
WiSeDB Action Selection

Sample random parameter $\theta_i$ according to $P(\theta_i | D)$

context $x_i$

Select best action $\alpha_i$ according to $\theta_i$

$D = D \cup (x_i, a_i, c_i)$

argmin $E(c | x_i, a_i, \theta_i) [\alpha_i]$

Observe cost $c_i$

Update the experience set

Create new model

Select a random decision tree and pick best action according to it

Update the experience set

Create new model
**Effectiveness**

**Training Data**
- 30 query sequence
- 22 TPC-H templates
- Repeat until convergence

*Optimal*: brute force (NP-hard)

*Clairvoyant*: perfect cost model

**Amazon AWS**
- t2.large, t2.medium, t2.small

WiSeDB models can perform at the same cost as a perfect cost model
**Effectiveness (concurrency)**

**Training Data**

- 22 TPC-H templates
- 900 queries/hour
- Poison distribution

*Clairvoyant*: perfect cost model

*One query/vCPU*: 1-2 queries

*Two queries/vCPU*: 2-4 queries

**WiSeDB models handles concurrency levels with no pre-training or tuning**
Adaptivity

Training Data

13 TPC-H templates
900 queries/hour
Poison distribution
Max SLO

_all new at once_: 7 new templates
every 2000 queries (after convergence)

_new over time_: 1 new template
every 500 queries

WiSeDB models quickly adapt to new unseen before templates
More details...


[CloudDB2016] Workload Management for Cloud Databases via Machine Learning, Ryan Marcus, Olga Papaemmanouil,


Next Steps: Batch Scheduling

- Train once, use “forever”?  
  - obsolescence detection and correction via SVMs

Data Management Application

- Cost Management
- SLA Management
- Resource Provisioning
- Workload Scheduling

- IaaS Provider
Next Steps

**Batch Processing (Offline Learning)**

- Concurrent query execution
- Hybrid (offline/online) model
- Exploratory Query Execution

**Data Management Application**

- Cost Management
- SLA Management
- Resource Provisioning
- Workload Scheduling

**IaaS Provider**
Next Steps: Online Learning

- Query Scheduling
  - query ordering actions

- Shut-down strategy
  - hill-climbing learning

- Training overhead
  - search space reduction
Next Steps: Tenant Placement

Database-as-a-Service
- Managed DBMS
- Relational & NoSQL DBs
- Cost effective tenants
  assignment to resources
- SLO-awareness
Conclusions

- Cost and SLA management for IaaS-deployed DBs are not becoming simpler

- WiSeDB demonstrates how ML techniques can help
  - discover customized solutions for app-specific SLAs
  - automate complex application management decisions
  - adapt to workload and resource configurations
  - build systems that perform beyond unaided human heuristics
Our Database Group

Ryan Marcus
- Cloud Databases
- Machine Learning

Kyriaki Dimitriadou
- Interactive Data Exploration
- Machine Learning

Zhan Li
- Benchmarking Optimizers
- Statistical Analysis
THANK YOU

Questions?