When free is not really free
Towards Benchmark as a Service?

Avrilia Floratou (University of Wisconsin – Madison)
Jignesh M. Patel (University of Wisconsin – Madison)
Willis Lang (University of Wisconsin – Madison)
Alan Halverson (Microsoft Jim Gray Systems Lab)
Motivation

On-site Relational Database Systems (RDBMSs)

Complexity of:
• Choosing
• Configuring
• Maintaining the RDBMS and server

Users must also consider the long-term:
• License costs
• Maintenance costs
• Administration costs
Cloud-based Database-as-a-Service (DaaS)

DaaS provider responsible for:
- Operating
- Administering the RDBMs and server.

User pays “on-demand” based only on the computing resources that are actually consumed.

DaaS offerings with multiple pricing options (e.g. based on different types of compute resources) can address a variety of users’ needs.
Motivation

Cloud-based Database-as-a-Service (DaaS)

Do the current DaaS offerings actually simplify the process of running a database workload?

Users must make an upfront decision of choosing a DaaS offering while the long-term performance and cost consequences are harder to figure out.
An example pricing model

<table>
<thead>
<tr>
<th>Type of compute resources (DBInstance class)</th>
<th>Hourly Machine Usage fee</th>
<th>Hourly License Cost Fee</th>
<th>Commercial DBMS</th>
<th>Hourly License Cost Fee</th>
<th>Open-Source DBMS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Small</td>
<td>$0.11</td>
<td>$0.04</td>
<td>$0.16</td>
<td>$0</td>
<td>$0.11</td>
</tr>
<tr>
<td>Large</td>
<td>$0.44</td>
<td>$0.20</td>
<td>$0.64</td>
<td>$0</td>
<td>$0.44</td>
</tr>
<tr>
<td>HM XL</td>
<td>$0.65</td>
<td>$0.20</td>
<td>$0.85</td>
<td>$0</td>
<td>$0.65</td>
</tr>
<tr>
<td>HM Double XL</td>
<td>$1.30</td>
<td>$0.40</td>
<td>$1.70</td>
<td>$0</td>
<td>$1.30</td>
</tr>
<tr>
<td>HM Quadruple XL</td>
<td>$2.60</td>
<td>$0.80</td>
<td>$3.40</td>
<td>$0</td>
<td>$2.60</td>
</tr>
</tbody>
</table>
Example

Wisconsin Benchmark Query 21

Insert into TMP
Select min (unique3) from TABLE1
Group by onePercent

<table>
<thead>
<tr>
<th></th>
<th>SQL Server</th>
<th>MySQL</th>
</tr>
</thead>
<tbody>
<tr>
<td>Monthly Storage fee</td>
<td>$25</td>
<td>$25</td>
</tr>
<tr>
<td>Hourly Machine Usage fee</td>
<td>$1.30</td>
<td>$1.30</td>
</tr>
<tr>
<td>Hourly License Cost fee</td>
<td>$0.65</td>
<td>$0</td>
</tr>
<tr>
<td>Execution Time</td>
<td>185 sec</td>
<td>621 sec</td>
</tr>
</tbody>
</table>
Example

The total cost is dominated by the storage fee.

Cumulative Monthly User Cost

<table>
<thead>
<tr>
<th></th>
<th>Cumulative Monthly User Cost ($)</th>
</tr>
</thead>
<tbody>
<tr>
<td>MySQL</td>
<td>25</td>
</tr>
<tr>
<td>SQL Server</td>
<td>15</td>
</tr>
</tbody>
</table>

#Repetitions/Month = 1
Example

Using a more expensive (per hour) DBMS, can save 54%!

Cumulative Monthly User Cost ($)

MySQL
SQL Server

#Repetitions/Month = 4000
Cost Model

Monthly User Cost (MUC)

- Hourly Machine Usage fee \((mc)\)
- Hourly License fee \((lc)\)
- Monthly Storage fee per GB \((stc)\)

Database Size (GB): \(DS\)
Hours of utilization: \(H\)

\[ MUC = H \times (mc + lc) + DS \times stc \]
Experimental Setting

- **Server Configuration:**
  - 2 quad-core processors @2.13GHz
  - 32 GB of RAM
  - 12 146 GB SAS drives

- **DBMSs:**
  - MySQL (Community Server 5.5.9)
    - 64-bit Ubuntu Server 9.10
  - SQL Server 2008 R2 (Data Center Edition)
    - 64-bit Windows Server 2008 R2 Enterprise Edition

- **DBMS Configuration:**
  - 24 GB buffer pool
  - 10 data disks, 1 log disk
  - InnoDB storage engine for MySQL
Workload

- Wisconsin Benchmark
  - Two tables of 80 GB each
  - One table of 8GB

- 6 different types of workloads
  - OLTP (HeapWorkload, IndexedWorkload)
  - DSS (HeapWorkload, IndexedWorkload)
  - Mixed (HeapWorkload, IndexedWorkload).
Cost Model Parameters

\[
MUC = H \times (1.30 + l_c) + 25
\]

\[
mc = $1.30 \\
stc = $0.10 \text{ per GB} \\
DS = 250 \text{ GB} \\
H: \text{Hours of utilization}
\]

\[
l_c = \begin{cases} 
$0, \text{MySQL} \\
$0.65-$3.90, \text{SQL Server}
\end{cases}
\]
Consists of 14 queries (DSS and OLTP)
Queries in the workload are run sequentially
Total Time: \( \sum \text{Query time} \)

<table>
<thead>
<tr>
<th>SQL Server Time (sec)</th>
<th>MySQL Time (sec)</th>
</tr>
</thead>
<tbody>
<tr>
<td>3441</td>
<td>8079</td>
</tr>
</tbody>
</table>

How does this 2.3X performance gap affect the monthly user cost for different query loads?
Mixed Indexed Workload

Cumulative Monthly User Cost ($)

<table>
<thead>
<tr>
<th>Service</th>
<th>Repetitions/Month</th>
<th>Cost</th>
</tr>
</thead>
<tbody>
<tr>
<td>MySQL</td>
<td></td>
<td>25</td>
</tr>
<tr>
<td>SQL Server (Lc = $0.65)</td>
<td></td>
<td>20</td>
</tr>
<tr>
<td>SQL Server (Lc = $1.30)</td>
<td></td>
<td>25</td>
</tr>
<tr>
<td>SQL Server (Lc = $2.60)</td>
<td></td>
<td>30</td>
</tr>
<tr>
<td>SQL Server (Lc = $3.90)</td>
<td></td>
<td>35</td>
</tr>
</tbody>
</table>

#Repetitions/Month = 1
Mixed Indexed Workload

Using the commercial DBMS can save up to 28% (lc = $0.65)

Cumulative Monthly User Cost ($)

MySQL
SQL Server (lc = $0.65)
SQL Server (lc = $1.30)
SQL Server (lc = $2.60)
SQL Server (lc = $3.90)

#Repetitions / Month = 30
Using the commercial DBMS can save up to 35% and at the same time being 2.3X faster!

Cumulative Monthly User Cost ($)

MySQL
SQL Server (Ic = $0.65)
SQL Server (Ic = $1.30)
SQL Server (Ic = $2.60)
SQL Server (Ic = $3.90)

#Repetitions/Month = 300
The User’s Perspective

Cost
Performance

DBInstance
Different
hardware
configurations
Different DBMSs

Workload
Storage space
Workload
Characteristics
(e.g. query
frequency)
Towards Benchmark-as-a-Service

User

BaaS

Database/ Workload Characteristics

DaaS provider

Price Quote

SLA
Closer to the “utility” model.
Reduces complexity.

Motivation to find the most optimal way to run the backend DBMS engine.
Reduces the operational cost.
The system operates more efficiently.
Directions for Future Work

- Setting up BaaS is challenging
  - How should the user describe the workload?
  - How should the provider run a mix of workloads that started with a BaaS?
  - How should the provider monitor changes in workloads that started with a price quote from the BaaS?
Conclusions

- Argue that existing cloud pricing models are too complex for the user.
- Influence of different types of DBMSs on performance/cost when running database workloads on the cloud.
- Propose a new type of service (BaaS) that will introduce transparency and clarity when pricing DaaS.
“We really don’t need any new benchmarks. Every DaaS customer has a benchmark – his/her workload! What we really need is Benchmark-as-a-Service, and not new benchmarks!”

-- Jignesh Patel