Robust LSM-Trees Tuning for Workload and Resources Uncertainty

Authors: Andy Huynh, Harshal Chaudhari, Evimaria Terzi, Manos Athanassoulis

IBM Mentor: Josh Berkus
Tuning Data Systems

Workloads → Resources → Database → Performance
Tuning Data Systems

How often do our assumptions hold up?
How often do we have to retune our systems?
Outline

LSM Trees: A Review

Tuning Problem

Modeling and Deploying LSM Trees
Outline

LSM Trees: A Review

Tuning Problem

Modeling and Deploying LSM Trees
# LSM Trees: A Review

<table>
<thead>
<tr>
<th>Levels</th>
<th>Mem</th>
<th>Disk</th>
</tr>
</thead>
<tbody>
<tr>
<td>Buffer</td>
<td>$m_{\text{buff}}$</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>$m_{\text{filter}}$</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>...</td>
<td></td>
<td></td>
</tr>
<tr>
<td>L</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

$T$ is the size ratio of the buffer to the disk.
LSM Trees: Compactions

<table>
<thead>
<tr>
<th>Levels</th>
<th>Mem</th>
<th>Disk</th>
</tr>
</thead>
<tbody>
<tr>
<td>Buffer</td>
<td>$m_{\text{buff}}$</td>
<td>Tiering splits levels into multiple sorted runs</td>
</tr>
<tr>
<td>1</td>
<td>$m_{\text{filter}}$</td>
<td>Leveling keeps levels as one large sorted run</td>
</tr>
<tr>
<td>2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>...</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$L$</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Levels

Mem

Disk

Tiering splits levels into multiple sorted runs

Leveling keeps levels as one large sorted run
Outline

LSM Trees: A Review

Tuning Problem

Modeling and Deploying LSM Trees
Tuning Problem

\( w \): Workload
\( D \): LSM Tree Design
\( \phi \): Fixed System Parameters
\( C \): Cost (Latency)

\[ D^* = \operatorname{argmin}_D C(w, D, \phi) \]
Uncertainty Regions

Cloud resource variance

Workload drift

Multi-tenant nodes
Robust Tuning Problem

\( \mathbf{w} : \) Workload

\( D : \) LSM Tree Design

\( \phi : \) Fixed System Parameters

\( C : \) Cost (Latency)

\[ D^* = \arg \min_D C(\mathbf{w}, D, \phi) \]

\( U_w : \) Uncertainty Region of Workloads

\[ D^* = \arg \min_D C(\hat{\mathbf{w}}, D, \phi) \]

s.t., \quad \hat{\mathbf{w}} \in U_w
Robust Designs

![Graph showing three curves labeled D1, D2, and D3, with points w_{far-}, w_0, and w_{far+} on the x-axis and cost (latency) on the y-axis.]

![3D plot showing Range Reads %, Valid Read %, and Point Write % axes with points w_{far-}, w_0, and w_{far+}.]
Outline

LSM Trees: A Review

Tuning Problem

Modeling and Deploying LSM Trees
Modeling LSM Trees

Accurate modeling is extremely important
Modeling Reads

Bloom filters introduce a *false positive rate*

Reads can be split between *empty* and *non-empty* reads

Probabilistic model on expected number of I/Os
Modeling Writes

Write speeds depend heavily on **buffer size** relative to entries

**Compaction policy** plays a heavy role to frequency of writes

Amortized cost of write I/Os over a sustained period
Closing Thoughts

LSM Trees operate in uncertain environments requiring robust tunings.

Any tuning pipelines requires accurate models.

Can we extend this design paradigm to other data structures?

Working towards promising results!

Questions? Contact me: ndhuynh@bu.edu, ndhuynh.com