Adaptive Merging in LSM Trees

Albert Wu and Sam Fishman
CS 265, Spring 2016
RUM Conjecture
Introduction to LSM Trees

Leveled LSM Tree, $k = 4$
Introduction to LSM Trees

Size-tiered LSM Tree, $k = 4$
## I/O Performance for LSM Trees

<table>
<thead>
<tr>
<th></th>
<th>Leveled</th>
<th>Size-tiered</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Reads</strong></td>
<td>$\log_k (n / B)$</td>
<td>$k \log_k (n / B)$</td>
</tr>
<tr>
<td><strong>Writes</strong></td>
<td>$k \log_k (n / B)$</td>
<td>$\log_k (n / B)$</td>
</tr>
</tbody>
</table>

Read and write amplification - $k$ represents merging factor, $n$ represents the total number of bytes of data, and $B$ represents the number of bytes in the topmost level.
Project Goals

- Implement **mechanisms** for switching between different merge factors, merge strategies
- Design **policies** for when to switch between different merge parameters