Scaling challenges

Two facets of scaling:

- Performance scaling: query rate & traffic growth
- Operation scaling: maintenance increase
Pseudo Quality of Service (QoS): a set of MySQL servers is accessible via 4 distinct ports:

<table>
<thead>
<tr>
<th>HAProxy Frontend Port</th>
<th>MySQL Port</th>
<th>Main Pool members</th>
<th>Backup Pool members</th>
<th>Gracefulness¹</th>
<th>Use case</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;$port&gt; 3306</td>
<td>&lt;$port&gt; 3306</td>
<td>All non-lagging slaves</td>
<td>Only master</td>
<td>let live</td>
<td>Apps business critical and not tolerant to replication lag</td>
</tr>
<tr>
<td>1&lt;$port&gt; 13306</td>
<td>&lt;$port&gt; 3306</td>
<td>All non-lagging slaves</td>
<td>Only master</td>
<td>kill connection</td>
<td>Not tolerant to disconnection</td>
</tr>
<tr>
<td>2&lt;$port&gt; 23306</td>
<td>&lt;$port&gt; 3306</td>
<td>All slaves (lagging)</td>
<td>let live</td>
<td>Apps tolerant to replication lag or not business critical</td>
<td></td>
</tr>
<tr>
<td>3&lt;$port&gt; 33306</td>
<td>&lt;$port&gt; 3306</td>
<td>All slaves (lagging)</td>
<td>kill connection</td>
<td>Tolerant to disconnection</td>
<td></td>
</tr>
</tbody>
</table>

¹ Achieved via HaProxy backend option: on-marked-down shutdown-sessions
How do applications connect to HAProxy?

- Multi-host jdbc url, for example: `jdbc:mysql:loadbalance://[host1],[host2]...`
- Consul DNS: DNS records are returned based on HAProxy health-check
Pros & Cons

- [+] Load balance reads across healthy MySQL replicas
- [+] Pseudo quality of service (QoS)
- [+] Ensure replicas serving traffic are not lagging
- [+] Fast take in/out of rotation mysql backends for manual maintenance
- [-] Harder to track connections
- [-] Requires some client side tweaking or some tolerance to disconnection.
Q & A