To Share or not to Share?
Kevin Schmid, Manos Athanassoulis, Stratos Idreos
CS265 2014

**Individual Execution**

<table>
<thead>
<tr>
<th>SELECT CUSTOMERS</th>
<th>JOIN ORDERS</th>
</tr>
</thead>
<tbody>
<tr>
<td>NAT = FRENCH</td>
<td>YEAR = 2010</td>
</tr>
</tbody>
</table>

**Shared Execution**

<table>
<thead>
<tr>
<th>SELECT CUSTOMERS</th>
<th>JOIN ORDERS</th>
</tr>
</thead>
<tbody>
<tr>
<td>NAT = FRENCH</td>
<td>YEAR = 2010</td>
</tr>
</tbody>
</table>

**Pros**
- avoid repeated work
- better runtime guarantees
- less need for synchronization
- fits in existing systems

**Cons**
- can add unneeded work
- larger working sets

**Goal**
- improve existing shared hash join cost model
- improve shared planners

**Impact of Data Distribution**
- \(|\text{table1}| = X \) tuples, \(|\text{table2}| = 10X \) tuples
- binary-valued payloads
- vary data distribution

**Giannikis cost model:**
join cost = builds + probes

**individual execution**
- \(2 \cdot (\text{Build}(0.5\cdot|\text{x}|) + \text{Probe}(5\cdot|\text{x}|))\)
- \(\text{Build}(|\text{x}|) + \text{Probe}(10\cdot|\text{x}|)\)

**shared execution**
- \(\text{Build}(|\text{x}|) + \text{Probe}(10\cdot|\text{x}|)\)

**Execution Times**

\(X = 1.024M\) tuples

\(0 \leq d \leq \frac{1}{2}\)

<table>
<thead>
<tr>
<th>table1.A</th>
<th>0</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>d</td>
<td>(\frac{1}{2} - d)</td>
</tr>
<tr>
<td>1</td>
<td>(\frac{1}{2} - d)</td>
<td>d</td>
</tr>
</tbody>
</table>

X = 10.24M tuples

\(c\)rossover \(\text{more expensive?}\)