To Nest or Not to Nest, When and How Much: Representing Intermediate Results of Graph Pattern Queries in MapReduce Based Processing

Padmashree Ravindra, HyeongSik Kim and Kemafor Anyanwu, North Carolina State University

RDF Graph Processing on MapReduce

I. Early Complete Unnesting: Reduce-side Full Replication
   - Pig’s COGROUP vs. JOIN (SJ1)
   - Unnest on join column
   - In reduceSJ1 FLATTEN
   - If MVPJoin, Redundancy ∝ multiplicity of MVP

II. Delayed Complete Unnesting: Map-side Full Replication
   - NTGA’s nesting-aware operators
   - Unnest on join column
   - In map, MVPJoin, map-side-unnest (replicates non-MVPs)
   - If MVPJoin, Redundancy ∝ multiplicity of MVP

III. Delayed Partial Unnesting: Map-side Partial Replication
   - Reduce redundancy factor by sharing data references across reduce() groups
   - Reduce shuffle costs
   - Partition based on Indirect Hashing of map output key k1 to Reducer space
   - partition(k1) = hashCode(func*(k1)) % r
   - Unnest on join column
   - In map, MVPJoin, map-side-partial-unnest
   - Redundancy ∝ range of func*
   - Special reduce() to handle multiple keys within a reduce function group

Note on func*: key % N
Value of N should be related to redundancy factor (average multiplicity of MVP, number of non-MVP columns), number of reducers etc.

Impact of MVP Redundancy in MapReduce

- Redundancy overhead → Red.Writes + Map.Reads + Map.LocalWrites + (Sorting + Data Transfer) ....
- Ripple effect across subsequent MR cycles

Unnesting Strategies for Efficient Management of MVPs

Impact of MVP Redundancy in MapReduce

- Redundancy overhead → Red.Writes + Map.Reads + Map.LocalWrites + (Sorting + Data Transfer) ....
- Ripple effect across subsequent MR cycles

Unnesting Strategies for Efficient Management of MVPs

<table>
<thead>
<tr>
<th>Strategy</th>
<th>Map Output Records</th>
<th>MVP Rep (%)</th>
<th>Map (s)</th>
<th>Reduce (s avg. shuffle)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pig-Def (4884)</td>
<td>4.903M</td>
<td>19.44</td>
<td>279</td>
<td>246 (192)</td>
</tr>
<tr>
<td>NTGA (6784)</td>
<td>4.873M</td>
<td>19.10</td>
<td>273</td>
<td>246 (159)</td>
</tr>
<tr>
<td>NTGAOpt (1000)</td>
<td>4.823M</td>
<td>18.14</td>
<td>249</td>
<td>257 (144)</td>
</tr>
<tr>
<td>NTGAOpt (60)</td>
<td>4.843M</td>
<td>17.74</td>
<td>237</td>
<td>255 (139)</td>
</tr>
<tr>
<td>NTGAOpt (40)</td>
<td>4.244M</td>
<td>16.80</td>
<td>223</td>
<td>245 (130)</td>
</tr>
<tr>
<td>NTGAOpt (20)</td>
<td>3.326M</td>
<td>15.13</td>
<td>185</td>
<td>212 (95)</td>
</tr>
</tbody>
</table>

References