LDBC SNB on Differential Dataflow

Frank McSherry
Materialize, Inc.
mcsherry@materialize.io
Differential Dataflow

A collection-oriented eDSL (in Rust) with:

Support for many operators: relational, map-reduce, (nested) mutual recursion.

Collections are described by streams of timestamped updates, and

Engine correctly updates all computations, in time proportional to the trace difference.
Knows
Comms
Posts
Forum

BI Q25

short paths
125 lines of code (17 ;s) (bidirectional Dijkstra)

edge scores
50 lines of code (5 ;s) (iterative roll-up)

forum filter
13 lines of code (1 ;) (joins, count, probe)
## Performance

### Experiment: Maintain 100 queries with updates (SF1, loaded by CreationDate).

<table>
<thead>
<tr>
<th>Batch size</th>
<th>Elapsed</th>
<th>Refresh</th>
<th>Throughput</th>
</tr>
</thead>
<tbody>
<tr>
<td>$10^3$</td>
<td>229.490s</td>
<td>14.37Hz</td>
<td>14,380 elt/s</td>
</tr>
<tr>
<td>$10^4$</td>
<td>109.740s</td>
<td>3.01Hz</td>
<td>30,071 elt/s</td>
</tr>
<tr>
<td>$10^5$</td>
<td>55.024s</td>
<td>0.60Hz</td>
<td>59,973 elt/s</td>
</tr>
<tr>
<td>$10^6$</td>
<td>42.780s</td>
<td>0.07Hz</td>
<td>77,138 elt/s</td>
</tr>
<tr>
<td>$10^7$</td>
<td>38.375s</td>
<td>0.03Hz</td>
<td>85,993 elt/s</td>
</tr>
</tbody>
</table>
Thoughts

LDBC has a variety of different queries, which is great (for me) but perhaps intimidating.

Should there be reference stats for individual queries, so that folks can attack easier ones?

Keep the queries hard, reach for stretch goals. (e.g., millisecond microbatches) (e.g., query maintenance, sharing)