EmptyHeaded: A Relational Engine for Graph Processing

Chris Aberger, Susan Tu, Kunle Olukotun, and Christopher Ré
Stanford University
In theory, theory and practice are the same.
New join algorithms translate to big gains!

4-Clique: LiveJournal
(5M nodes, 70M edges)

<table>
<thead>
<tr>
<th>Database</th>
<th>Time (s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Vertica</td>
<td>&gt;30m</td>
</tr>
<tr>
<td>PostgreSQL</td>
<td>&gt;30m</td>
</tr>
<tr>
<td>Spark</td>
<td>&gt;30m</td>
</tr>
<tr>
<td>LogicBlox</td>
<td>338</td>
</tr>
<tr>
<td></td>
<td>2</td>
</tr>
</tbody>
</table>

EmptyHeaded =
(1) Theory -> Use GHGs
(2) Systems -> Exploit SIMD
Pairwise joins are suboptimal

Facebook(x,y) ⋈ Twitter(y,z) ⋈ Text(x,z)

Panic: Best known bound is $O(N^{3/2})$ and any pairwise join plan takes $\Omega(N^2)$. 
Instead of computing joins over relations in a pairwise manner, compute them over attributes in a multiway fashion.

**Algorithm:** Only Foreach and Set Intersection.
Demystifying the WC-Optimal Algo.

Facebook(x,y) ⋈ Twitter(y,z) ⋈ Text(x,z)

```
for x in Facebook[] ∩ Twitter[]
for y in Facebook[x] ∩ Twitter[]
for z in Twitter[y] ∩ Text[x]
  out ← out ∪ (x,y,z)
```
In EmptyHeaded, theory and practice are the same.

High-Level Engines
- Outperforms *LogicBlox* by 19x-3500x
- Outperforms *SocialLite* by 8x-3500x

Low-Level Graph Engines
- Outperforms *PowerGraph* by 3x-10x
- Outperforms *Snap-Ringo* by 2x-11x
- *Competes* within 0.98x-4x of *Galois*

Standard graph workloads (PageRank, Triangle, SSSP) and pattern queries
EmptyHeaded = Theory + Systems

- Theory (GHDs)
- Systems (SIMD)
Generalized hypertree decompositions (GHDs) yield even better runtimes.

- Gottlob et al. & Puttagunta et al. [PODS ‘16]

**Key Idea:** This is our analog of relational algebra to represent logical query plans.

**Enables:** Classic query optimizations like early aggregation and pushing down selections

**Key Insight:** Creates an execution DAG.
GHDs in 1 minute.

\[ WC: O(N^3) \]

\[ GHD+WC: O(N^{3/2}) + |OUT| \]

>3500x speedup
EmptyHeaded = Theory + Systems
**Data Layout: Trie Representation**

**friends (or foes?)**

<table>
<thead>
<tr>
<th>src</th>
<th>dst</th>
</tr>
</thead>
<tbody>
<tr>
<td>C. Ré</td>
<td>M. Stonebraker</td>
</tr>
<tr>
<td>C. Ré</td>
<td>D. DeWitt</td>
</tr>
<tr>
<td>C. Ré</td>
<td>A. Pavlo</td>
</tr>
<tr>
<td>C. Ré</td>
<td>J. Hellerstein</td>
</tr>
<tr>
<td>K. Olukotun</td>
<td>M. Stonebraker</td>
</tr>
<tr>
<td>K. Olukotun</td>
<td>D. Patterson</td>
</tr>
</tbody>
</table>

**Dictionary Encoded ID’s for each node**

**Panic:** Sets are skewed in several different ways!
Exploiting SIMD: The battle with skew

Goal: Design an execution engine that automatically exploits SIMD parallelism.

Challenge: cope with skew in data

- **Cardinality Skew**
  - Solution: Choose amongst SIMD algorithms!

- **Density Skew**
  - Can we do better than choosing amongst SIMD algorithms?
  - Solution: Use multiple representations!

>400x speedup
Conclusion

- **GHDs** to represent **logical query plans** in addition to WC Optimal join algorithm result in >3500x speedup

- **Multiple representations** and **set intersection algorithms** optimized for **SIMD parallelism** result in >400x speedup

- Theory + Systems translates to promising results!
  - **Outperforms** LogicBlox, SociaLite, PowerGraph and Snap-Ringo by 2-3500x
  - **Competes** within 0.98x-4x of Galois

Thanks! Christopher Aberger
www.stanford.edu/~cabberger

Try me:
https://github.com/HazyResearch/EmptyHeaded