Query Processing in a Self-Organized Storage System

Hannes Mühleisen, supervised by Robert Tolksdorf
Distributed DBs - Goals

- Scalability
  - Data, Queries, Nodes
- Robustness
  - Node/Network failure
- Adaptiveness
  - “Fair” distribution of load
Clustered / Federated

[Bernstein81, Epstein78]
Global Laws

[Harren02, Karnstedt04, Rösch05]
Probabilistic Request Routing

[Lindgren03]
[Wilensky97, NetLogo Ants model]
## Distribution Paradigms

<table>
<thead>
<tr>
<th></th>
<th>Scalability</th>
<th>Adaptability</th>
<th>Robustness</th>
<th>Completeness</th>
<th>Complex Queries</th>
</tr>
</thead>
<tbody>
<tr>
<td>Stand-Alone</td>
<td>low</td>
<td>high</td>
<td>low</td>
<td>high</td>
<td>✓</td>
</tr>
<tr>
<td>Federated</td>
<td>high</td>
<td>high</td>
<td>fair</td>
<td>high</td>
<td>✓</td>
</tr>
<tr>
<td>Global-Law</td>
<td>high</td>
<td>fair</td>
<td>high</td>
<td>high</td>
<td>✓</td>
</tr>
<tr>
<td>Probabilistic e.g. Swarms</td>
<td>high</td>
<td>high</td>
<td>high</td>
<td>fair</td>
<td>?</td>
</tr>
</tbody>
</table>
Research Question

Can complex queries be evaluated efficiently in a swarm-based distributed storage system?
Mutable Moving Query Plans

- parse ✓
- rewrite ✓
- optimize ✓
- execute

Based on?

Move & repeat

Where?

Catalog ❌

[Papadimos03, Battré08]
1. r(#)  
2. r(*)

p(#) = 53%  
p(*) = 3%

p(#) = 2%  
p(*) = 78%

p(#) = 2%  
p(*) = 10%
Handling Routing #Failures

\[ r(#) \]

\[ p(#) = 0\% \]

what now?

\[ p(#) = 0\% \]

Trackback!
Evaluation Methodology

# Participating Nodes / Query

Not actual data!
Evaluation Methodology

# Results / Query

Optimal Plan
Moving Mutable Plans
Static Plan Routing

Better
Not actual data!
Thank You!

Questions?

Web Page: http://hannes.muehleisen.org