Distributed Search with Rendezvous Search Systems

Christof Leng
Distributed Search: A Traditional View

- **Implement distributed key-value lookup**
  - e.g., a distributed hash table (DHT)
  - scalable & simple

- **Build everything else on top of it**
  - keyword search, XPath, range queries
  - map every operation to a number of lookups

- **Downside:**
  - implement every query language yourself
  - query processing not portable between DHTs
  - unforeseen side-effects & interactions

- **Experts needed for every application**
  - application domain
  - P2P networking
  - information retrieval

... need sophisticated queries ...
Rendezvous Search

- “Every query will meet each data item somewhere in the network”
  - don’t worry how it gets there
  - black box approach

- Implement the match functionality
  - as if it was local
  - re-use existing libraries

- Advantages:
  - separation of concerns
  - flexible
  - scalable (O(√n) search cost)
  - robust (O(√n) replicas)

... need sophisticated queries ...

Application Developer

Rendezvous Search System

Information Retrieval Library

glue code
Example Application
Data Center Solutions
Grid

Pro:
- simple
- efficient
- flexible

Con:
- not fault-tolerant
- not size-adaptive
- not load-adaptive
- two-dimensional structure

Reference:
ROAR – Rendezvous on a Ring

Pro:
- simple mapping to 1 dim.
- size-adaptive
- load-adaptive
- fault-tolerant

Con:
- centralized
- inefficient

Reference:
Structured Peer-to-Peer
Bit Zipper

**Pro:**
- z-order mapping
- DHT-based = decentralized
- fast tree-based replication

**Con:**
- relies on DHT’s reliability
- no replica maintenance

**Reference:**
W. W. Terpstra, S. Behnel, L. Fiege, J. Kangasharju, and A. Buchmann.
Bit Zipper Rendezvous—Optimal Data Placement for General P2P Queries.
Deetoo

**Pro:**
- 2 rings for projection
- includes replica maintenance

**Con:**
- maintenance overhead for 2 rings
- no responsibility ranges (probabilistic approach)
- additional overhead

**Reference:**
Unstructured Peer-to-Peer
Ferreira et al. – Random Walks

Pro:
- very resilient topology
- proven correctness

Con:
- probabilistic
- additional overhead
- slow & fragile random walks
- no topology protocol

Reference:
**Pro:**
- resilient & fast
- size-adaptive
- load-adaptive
- topology & replica maintenance included

**Con:**
- probabilistic
- additional overhead

**Reference:**
**Pro:**
- unstructured solution on top of DHT
- incremental deployment possible

**Con:**
- probabilistic
- additional overhead
- fragile

**Reference:**
Semi-Structured Peer-to-Peer
**Pro:**
- combines unstructured resilience with structured efficiency

**Con:**
- probabilistic
- lacks analysis and prototype evaluation

**Reference:**
## Comparison

<table>
<thead>
<tr>
<th></th>
<th>Data Center</th>
<th>Structured</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Name</strong></td>
<td><strong>Grid</strong></td>
<td><strong>ROAR</strong></td>
</tr>
<tr>
<td><strong>Pro</strong></td>
<td>simple</td>
<td>adaptive</td>
</tr>
<tr>
<td><strong>Con</strong></td>
<td>not adaptive</td>
<td>centralized</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>Unstructured</th>
<th>Semi-Structured</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Name</strong></td>
<td><strong>Ferreira</strong></td>
<td><strong>BubbleStorm</strong></td>
</tr>
<tr>
<td><strong>Pro</strong></td>
<td>simple</td>
<td>resilient &amp; adaptive</td>
</tr>
<tr>
<td><strong>Con</strong></td>
<td>fragile &amp; slow</td>
<td>probabilistic</td>
</tr>
</tbody>
</table>
Rendezvous Search & Cloud Computing

Key-Value Store Scalability

- similar problems
- similar requirements
- less research history

could benefit from
- size adaptivity
- traffic adaptivity
- fault tolerance
- algorithm diversity

Follow-up Work

Citation
Conclusion

- **Rendezvous Search is a powerful abstraction**
  - scalable
  - flexible
  - robust

- **A wide range of solutions is out there**
  - data center
  - unstructured P2P
  - structured P2P
  - semi-structured P2P

- **Ever needed decentralized but powerful search in a project?**
  - don’t try to reinvent the wheel!
Thank you!

Data Center | Structured P2P | Unstructured P2P | Semi-Structured P2P
---|---|---|---
2003: Google Grid
2004: Bit Zipper, ROAR DHT
2005: Ferreira, ROAR DHT
2006: BubbleStorm
2007: Deetoo, Hautakorpi, SplitQuest
2008: BubbleStorm
2009: Deetoo, Hautakorpi
2010: SplitQuest
2011: Follow-up Work, Citation

Questions

BubbleStorm

http://www.bubblestorm.net
http://www.dvs.tu-darmstadt.de