DEWS: A Decentralized Engine for Web Search

Presented by
Prof. Raouf Boutaba
Web Search: Today

• Contemporary Web Search:
  – Logically centralized
  – Company controlled

• Problems
  – Censorship
  – Biased ranking
  – Privacy
Web Search : Decentralization

• Using P2P networks – YacY, Faroo
  – Search overhead
  – Churn

• DEWS:
  – P2P network between Webservers not end-hosts
  – Both decentralized and stable
Challenges

- Indexing the voluminous Web
- Resolving Web queries
- Ranking search results
- Incremental retrieval

DEWS addresses the first 3 Challenges
**Conceptual Overview**

**Web Server (WS) DHT:**
- **Pros:**
  - Very stable
  - 1 or 2 hop lookup via link cache
- **Cons:**
  - Additional overhead on WS
Plexus DHT

• Why Plexus\[^{[1]}\]?
  – Efficient routing with dynamic load-balancing
  – Supports approximate matching

• How Plexus works:
  – Generates a bit-pattern from advertisement/query keywords
  – Decodes this pattern to codewords using a Linear Binary Code
  – Routes using the generator matrix of the LBC

• Modification to Plexus routing
  – DEWS aggregates routing messages and packs multiple queries in one message

Indexing Mechanism

- **Base URL**
  - hash
  - codeword
    - Plexus Routing
      - Website index node
      - $u_i$

- **Keywords**
  - $\{k_{i,j}^{rep}\}$
    - DMP, n-gram
    - Bloom-filter
    - Pattern
      - List decoding
      - codewords
        - Plexus Routing
          - Inverted index nodes
          - $\gamma(k_{i,j}^{rep})$

- Used for Decentralized PageRank
- Used for Keyword Relevance
Decentralized PageRank

\[ w_i = (1 - \eta) + \eta \sum_{t=1}^{g} \frac{w_{it}}{L(u_{it})} \]
Distributed Inverted Index

$$u_i, \{v_{i1}, \beta(v_{i1})\}, \{<k_{i1}, r_{i1}>, \ldots <k_{i2}, r_{i2}>\}$$

$$\beta(v_{i1}) \quad \beta(v_{i2}) \quad \ldots \quad \beta(v_{iv})$$

$$\gamma (k_{i1}) \quad \gamma (k_{i2}) \quad \ldots \quad \gamma (k_{ig})$$

Overlay

---

Hash-map

Soft-link

PageRank

Keyword Relavance

\[\beta(u_i)\]
Resolving Web Query

Query keywords

Keyword-1
- DMP, n-gram
- Bloom-filter

Pattern
- List decoding

codewords
- Plexus Routing

Inverted index nodes

Keyword-2

Pattern
- List decoding

codewords

Inverted index nodes

Pagerank weight of $u_i$

Relevance of $u_i$ to $q_l$

1 if $q_l$ is in $u_i$;
0 otherwise

$$\text{rank}(u_i) = \sum\sum \vartheta_{il} (\mu \cdot w_i + (1 - \mu) \cdot r_{il})$$

Query keyword

$\{<u_i, w_i, r_{il}>\}$
Evaluation

• Simulation Setup
  – Web Track dataset from LETOR 3.0
    • ~ 1 million webpages and ~11 million hyperlinks
  – WS network size – up to 100,000 nodes.

• Measurements
  – Routing performance: scalability & overheads
  – Ranking performance: accuracy & convergence rate
  – Search performance: flexibility & accuracy

• Here we present two important results
Routing Performance

Observations:

- Advertisement hops do not increase significantly with network size
- Route aggregation in DEWS significantly reduces advertisement overhead
- URL advertisement requires more hops than keyword advertisement

Modified Plexus in DEWS

Original Plexus

Advertisement Scalability
Observations:

- Spearman’s footrule distance decays rapidly with simulation time, which indicates fast convergence of our distributed ranking algorithm.

- Variation in Top-20 and Top-100 elements is not high => DEWS is close to centralized ranking.
Summary

• DEWS is a self-indexing architecture for the Web
  – provides censorship resistance
  – delivers unbiased ranking of search results
  – makes it hard to track users’ search history

• Future Research:
  – Support for incremental retrieval in DEWS
    • Can be achieved by gradually increasing decoding radius in Plexus routing.
  – Develop a working prototype of DEWS and deploy in the Web
Questions?