The Web of the Future

Gerhard Weikum
weikum@cs.uni-sb.de
http://www-dbs.cs.uni-sb.de

Challenges:
• Performance and QoS Guarantees
• World-wide Failure Masking and Continuous Availability
• Intelligent Information Search

Importance of quality guarantees not limited to Web → e.g., DFG graduate program at U Saarland

From Best Effort To Performance & QoS Guarantees

"Our ability to analyze and predict the performance of the enormously complex software systems ... are painfully inadequate"
(Report of the US President’s Technology Advisory Committee)

• Very slow servers are like unavailable servers
• Tuning for peak load requires predictability of workload × config → performance function
• Self-tuning requires mathematical models
• Stochastic guarantees for huge #clients
  \[ P \{\text{response time} \leq 5 \text{ s}\} > 0.95 \]

Example: Video (& Audio) Server
Partitioning of continuous data objects with variable bit rate into fragments of constant time length \(T\)
Periodic scheduling in rounds of duration \(T\)

Stochastic model:
\[ T_{\text{total}} = T_{\text{sock}} + \sum_{i=1}^{n} T_{\text{sock}} + T_{\text{trans}} + N \cdot f \cdot \sigma \cdot \text{response} \]
\[ \sum_{i=1}^{n} P[T_{\text{total}} > t] \leq \inf \left[ e^{-\theta / f} / f \cdot \sigma \cdot \text{response} \right] \]
Chernoff Bound

Auto-configure server: admission control, #disks, etc.

The Need for World-Wide Failure Masking

Please review and place your order
Place your order

Your server command (process id #20) has been terminated. Re-run your command (severity 13) in
/export/home/WWW/your-reliable-eshop.biz/mdb_1300_db.mbl

The Need for Performance and QoS Guarantees

Check Availability
(Look-Up Will Take 8-25 Seconds)

Internal Server Error.
Our system administrator has been notified. Please try later again.

Outline
✓ Performance and QoS Guarantees
✓ Continuous Service Availability
• Intelligent Information Search
• State of the Art & Research Challenges
• Focused Crawling
Crawling, Analysis, and Indexing of Web Documents

- Crawling
- Surfing Internet Cafes
- In Internet cafes with or without Web Suit
- Extraction of relevant words
- Linguistic methods: stemming
- Construction of weighted features (terms)
- Thesaurus (Ontology)
- Index (B-tree)
- URLs

Vector Space Model for Content Relevance

- Query \( q \in \{0,1\}^F \)
- Documents are feature vectors \( d_i \in \{0,1\}^F \)
- e.g., using \( d_i = w_i / \sqrt{\sum w_j^2} \) \( w_i = \frac{freq(f_i, d_i)}{\max freq(f_i, d)} \) \( \log \frac{\# docs}{\# docs with f_i} \)
- Similarity metric: \( \text{sim}(d_i, q) = \sum_{j=1}^{F} d_{ij} \cdot q_j \)

Dimension of a Large-Scale Search Engine

- > 10 Terabytes raw data
- > 10 Mio. terms
- > 150 Mio. queries per day
- < 1 sec. average response time
- < 30 days index freshness
- > 1000 Web pages per second crawled

High-end server farm:
10 000 Intel servers each with
> 1 GB memory, 2 disks, and partitioned, mirrored data, distributed across all servers, plus load balancing of queries, remote administration, etc.

(In-) Effectivity of Web Search Engines

- AltaVista: Fermat’s last theorem
- Google: Moment-generating Functions; Chernoff’s Theorem: http://www.siam.org/catalog/mcc10/bahadur.htm
- Yahoo: Moment-generating Functions; Chernoff’s Theorem: http://www.siam.org/catalog/mcc10/bahadur.htm
- Mathsearch: No matches found.

Dimensions of a Large-Scale Search Engine

- > 10 Terabytes raw data
- > 10 Mio. terms
- > 150 Mio. queries per day
- < 1 sec. average response time
- < 30 days index freshness
- > 1000 Web pages per second crawled

High-end server farm:
10 000 Intel servers each with
> 1 GB memory, 2 disks, and partitioned, mirrored data, distributed across all servers, plus load balancing of queries, remote administration, etc.
From Observations to Research Avenues

Key observation:
yes, there are ways to find what you are searching, but intellectual time is expensive!
→ requires „intelligent“ automation

Research Avenues:
- Structure and annotate information: XML
- Organize documents „semantically“: ontologies
- Leverage machine learning: automatic classification
- More computer time for better result: focused crawling

Goal:
should be able to find results for advanced info request in one day with < 5 min intellectual effort that the best human experts can find with infinite time

Challenge: Expert Web Queries
- Where can I download an open source implementation of the ARIES recovery algorithm?
- Find the text and notes of the western song Raw Hide.
- What are Chernoff-Hoefding bounds?
- Find Fermat’s last/Wile’s theorem in MathML format.
- Are there any theorems isomorphic to my new conjecture? Find related theorems.
- Which professors from D are teaching DBS and have research projects on XML?
- Which Shakespeare drama has a scene where a woman talks a Scottish nobleman into murder?
- Who was the Italian woman that I met at the PC meeting where Moshe Vardi was PC Chair?

Challenge: (Meta-) Portal Building
- Who are the top researchers in the database system community? Who is working on using machine learning techniques for searching XML data?
- What are the most important results in large deviation theory?
- Find information about public subsidies for plumbers. Find new EU regulations that affect an electrician’s business.
- Which gene expression data from Barrett tissue in the esophagus exhibit high levels of gene A01g?
- Are there metabolic models for acid reflux that could be related to the gene expression data?

Outline
✓ Performance and QoS Guarantees
✓ Continuous Service Availability
  • Intelligent Information Search
✓ State of the Art & Research Challenges
  • Focused Crawling

Our Research Agenda

Focused Crawling

critical issues:
  - classifier accuracy
  - feature selection
  - quality of training data

Performance and QoS Guarantees
Continuous Service Availability
Intelligent Information Search
State of the Art & Research Challenges
Focused Crawling

null
Naives Bayes Classifier with Bag-of-Words Model

\[
\begin{align*}
\text{estimate } P(d|c_k) \text{ for } f \text{ as } f \sim P(f|d|c_k) P(d|c_k) \\
\text{with term frequency vector } \vec{f} \\
= \Pi_{c_k} P(f|d|c_k) P(d|c_k) \\
\text{with feature independence} \\
= \Pi_{c_k} \left( \frac{\text{length}(d)}{f_i} \right) P_k^{m_i} \prod_{i=1}^{m} \frac{f_i}{P_k} \\
\text{with multinomial distribution of each feature} \\
= \left( \frac{\text{length}(d)}{f_1 f_2 \cdots f_m} \right) P_k^{m_1} P_k^{m_2} \cdots P_k^{m_n} \\
\text{with binomial distribution of each feature vector and} \\
\begin{cases}
\frac{n}{k_1 k_2 \cdots k_n} & \text{if } \sum_{i=1}^{n} f_i = \text{length}(d) \\
= & \text{otherwise}
\end{cases}
\end{align*}
\]

Example of Naive Bayes (2)

classification of \(d_7\): (0 0 1 2 0 0 3 0)

\[
P(d|c_k) = \frac{m_k!}{f_1! f_2! \cdots f_m!} P_k^{m_1} P_k^{m_2} \cdots P_k^{m_n}
\]

for \(k=1\) (Algebra): \[
6 \left( \frac{f_1^3 f_2^3 f_3^3 f_4}{12^3} \right) = \frac{216}{6} = 36
\]

for \(k=2\) (Calculus): \[
6 \left( \frac{f_1^3 f_2^3 f_3^3 f_4}{12^3} \right) = \frac{216}{6} = 36
\]

for \(k=3\) (Stochastics): \[
6 \left( \frac{f_1^3 f_2^3 f_3^3 f_4}{12^3} \right) = \frac{216}{6} = 36
\]

Result: assign \(d_7\) to class \(C_3\) (Stochastics)

Classification using Support Vector Machines (SVM)

Training:
Compute \textit{separating hyperplane } \(w^T \bar{x} + b = 0\) that maximizes the min.
Distance of all positive and negative samples to the hyperplane

Decision:
Test new vector \(\bar{x}\) for membership in \(C\): \(w^T \bar{x} + b > 0\)

Feature Selection for Hierarchical Classification

Recursively assign new document to best positively tested topic

For topic \(C_i\) based on most discriminative features:
select features \(X\) with highest mutual information
(relative entropy, Kullback-Leibler divergence)

\[
MI(X_i, C_i) = \sum_{X \in \{X_i\}} \frac{P(X \cap C_i) \log \frac{P(X \cap C_i)}{P(X \cap C)}}{P(C)}
\]

Best features for Data Mining (vs. Web IR vs. XML):
mine, knowledge, OLAP, pattern, discov, cluster, dataset, ...

Feature Space Construction & Meta Strategies

\begin{itemize}
\item possible strategies:
  \begin{itemize}
  \item single term frequencies or \textit{i}tfidf with \(n\) n-grams
  \item term pairs within proximity window (e.g., support vector, match anchor......)
  \item text terms from hyperlink neighbors
  \item anchor text terms from neighbors
\end{itemize}
\item meta strategies (over \(m\) feature spaces for class \(k\)):
  \begin{itemize}
  \item unanimous decision: \(C_i = 1\) if \(\sum_{f \in \mathcal{F}} \rho(f, C_i) = m\)
  \item weighted average: \(C_i = 1\) if \(\sum_{f \in \mathcal{F}} \rho(f, C_i) \geq \tau\)
\end{itemize}
\item strategy \(\nu\) with best ratio of estimated precision to runtime cost
\end{itemize}

\[\text{with } \hat{\rho} \text{ estimator } \frac{\hat{\rho}(\nu)}{\hat{\rho}(\nu)} \text{ (Jaccard's) for precision of model } \nu \text{ for class } k \text{ based on leave-one-out training}\]
Link Analysis using Kleinberg’s HITS Algorithm

For web graph $G=(V,E)$ and topic-specific base set $B \subseteq V$ find

good authorities with authority score $x_p = \sum_{(p,q) \in E} y_q$
and good hubs with hub score $y_p = \sum_{(p,q) \in E} x_q$

Iterative approximation of principal Eigenvectors

$\dot{x} = A^T y$
$\dot{y} = A x$

High authority scores indicate good topic representatives

Implementation of the HITS Algorithm

1) Determine sufficient number (e.g., 50-200) of „root pages“ via relevance ranking (e.g., using tf*idf ranking)
2) Add all successors of root pages
3) For each root page add up to $d$ successors
4) Compute iteratively the authority and hub scores of this „base set“ (of typically 1000-5000 pages)
   with initialization $x_i := y_i := 1/|\text{base set}|$
   and normalization after each iteration
   → converges to principal Eigenvector (Eigenvector with largest Eigenvalue in the case of multiplicity 1)
5) Return pages in descending order of authority scores (e.g., the 10 largest elements of vector $x$)

Drawbacks of HITS algorithm:
• relevance ranking within root set is not considered
• susceptible to „topic drift“ → extended variants of HITS

Experiment on Information Portal Generation (1)

for single-topic portal on „Database Research“

start with only 2 initial seeds: homepages of DeWitt and Gray

goal: automatic gathering of DBLP author homepages
   (with DBLP excluded from crawl)

learning phase for imroved feature selection and classification:
   depth-first crawl limited to domains of seeds
   followed by archetype selection and retraining (for high precision)
harvesting phase for building a rich portal:
   prioritized breadth-first crawl (for high recall)

Experiment on Information Portal Generation (2)

result after 12 hours (on commodity PC):
• 3 mio. URLs crawled on 30 000 hosts, 1 mio. pages analyzed,
• 0.5 mio. pages positively classified
• found 7000 homepages out of 30 000 DBLP authors,
   712 authors of the top 1000 DBLP authors
   with 267 among the 1000 best crawl results

+ postprocessing for querying and analysis:
  • ranking by SVM confidence, authority score, etc.
  • clustering, relevance feedback, etc.

Ongoing and Future Work

• Deep Web exploration with auto-generated queries
• Exploiting ontological knowledge
  e.g.: search for a „woman talking someone into murder“
• Construct richer feature spaces
• Exploiting linguistic analysis methods
  e.g.: „cut his throat“ → act: killing
  subject:... object:...
• Generalized links & semantic joins, e.g., named entities
• Identifying semantically coherent units
• Combining focused crawling with XML search
  → auto-annotation of HTML, Latex, PDF, etc. docs
  → cross-document querying à la XXL
• User guidance & portal admin methodology
• Exploitation of surf trails from user community

Example Ontology (based on WordNet)

woman, adult female – (an adult female person)
⇒ amazon, virago – (a large strong and aggressive woman)
⇒ donna – (an Italian woman of rank)
⇒ geisha, geisha girl – (...)
⇒ lady (a polite name for any woman)
⇒ wife – (a married woman, a man’s partner in marriage)
⇒ witch – (a being, usually female, imagined to have special powers derived from the devil)
Ongoing and Future Work

- Deep Web exploration with auto-generated queries
- Exploiting ontological knowledge
e.g.: search for a “woman talking someone into murder”
- Construct richer feature spaces
- Exploiting linguistic analysis methods
e.g.: “... cut his throat ...”

- Subject: ...  
- Object: ...

- Generalized links & semantic joins, e.g. named entities
- Identifying semantically coherent units
- Combining focused crawling with XML search
  → auto-annotation of HTML, Latex, PDF, etc. docs
  → cross-document querying à la XXL
- User guidance & portal admin methodology
- Exploitation of surf trails from user community

Towards “Semantically Coherent” Units

- Teaching: DBS, IR, ...
- Research: XML, Auto-tuning, ...

Summary: Strategic Research Avenues

Challenges for next-decade Web information systems:
- Self-organizing systems built out of self-tuning components
  for performance and differentiated QoS guarantees
- Trouble-free, continuously available Web services
  with perfect failure masking to application programs
- Intelligent organization and searching of information
  based on synergy of DB, IR, CL, ML, and AI technologies
  → large-scale experiments
  → more and better theoretical underpinnings

Conceivable killer arguments:
- Infinite RAM & network bandwidth and zero latency for free
- Smarter people don’t need a better Web