7. Übung zur Vorlesung “Internet-Suchmaschinen” im Sommersemester 2009

Prof. Dr. Gerd Stumme, Wi.-Inf. Beate Krause

08. Juli 2009

1 Metasuchmaschinen

Eine Metasuchmaschine ist eine Suchmaschine, die die Benutzeranfragen an mehrere andere Suchmaschinen weiterleitet und dem Benutzer deren Ergebnisse präsentiert.

1. Stellen Sie genauer dar, welche Schritte in einer solchen Metasuche durchgeführt werden müssen!

   Lösungsvorschlag:

   a) Anfrage entgegennehmen
   b) Anfrage an alle Suchmaschinen weiterleiten
   c) Ergebnisse abwarten
   d) Ergebnisse aggregieren
      i. Gleiche Ergebnisse erkennen und zusammenfassen
      ii. Ranking berechnen
   e) Ergebnisse präsentieren
   f) evtl. Feedback entgegennehmen

2. Nennen Sie drei Probleme, die bei der Metasuche gelöst werden müssen, und schlagen Sie Lösungen vor!

   Lösungsvorschlag:

   a) Robustheit. Die Metasuchmaschine muss gegenüber Ausfall, Latenz usw. einzelner Suchmaschinen unempfindlich sein. Der einfachste Weg ist, nach

einen gewissen Timeout die Anfrage abzubrechen und die betreffende Suchmaschine zu ignorieren.

b) Screen Scraping. Suchmaschinen bieten oft kein spezielles Interface, um einen programmgesteuerten Zugriff zu ermöglichen. Oft wird daher das normale HTML-basierte Interface genutzt und die Ausgaben geparst, was fehleranfällig ist. Dies kann vermieden werden, wenn es z. B. eine Web-Service-Schnittstelle gibt. Andernfalls muss der HTML-Parser bei jeder Änderung der zugrundeliegenden Suchmaschine angepasst werden.

c) Aggregieren der Ergebnisse. Ergebnisse verschiedener Suchmaschinen müssen so zusammengefasst werden, dass jede Seite im Endergebnis nur einmal auftaucht. Dazu müssen z. B. auch die Vorschauschnipsel verwertet und die URLs normalisiert werden.

d) Ranking. Die verschieden Suchmaschinen liefern jeweils ein Ranking der Ergebnisse, aus denen ein zusammenfassendes Ranking berechnet werden muss. Dies könnte z. B. durch eine Durchschnittsbildung der Einzelplatzierungen erfolgen. Allerdings werden dann Problematiken wie unterschiedliche, zugrundeliegende Indexierung und Rankingalgorithmen nicht berücksichtigt.

3. Nennen Sie zwei Möglichkeiten, die Ergebnisse in einer Metasuchmaschine zu aggregieren.

   Lösungsvorschlag:

   a) Alle URLs, die von den verschiedenen Suchmaschinen auf eine Suchanfrage hin angegeben werden, werden gecrawlt. Mit den heruntergeladenen Seiten kann ein globales Ranking der Seiten basierend auf Standard-Rankingalgorithmen errechnet werden.

   b) Es werden nur die URLs und die Rankings der einzelnen Suchmaschinen an die Metasuchmaschine übermittelt. Jede Suchmaschine erhält dann bei der Ermittlung des Gesamtrankings ein unterschiedliches Gewicht, je nach Qualität der Suchmaschine. Die Qualität kann zuvor (manuell) evaluiert werden, und / oder durch Benutzerklicks auf das abschließende Ranking verändert werden.

2 Struktur des WWW

1. Betrachten Sie die Struktur des Web nach der Grafik in Kapitel 10, Seite 15.

Welche der gezeigten Bestandteile können in der Regel von Suchmaschinen erfasst werden?

Lösungsvorschlag:

Ohne Weiteres wird eine Suchmaschine nur die Seiten der zentralen, großen Zusammenhangskomponente (*Central Core*) sowie des Teils *Out* erreichen können. Alle anderen Teile, also die *Tendrils*, *In* und die *Tubes*, werden nicht gefunden.

2. Wie können die anderen Teile des Web von Suchmaschinen entdeckt werden?
   Lösungsvorschlag:

   In der Regel werden Seiten außerhalb von Central Core und Out nur von einer Suchmaschine gefunden werden, wenn sie ausdrücklich dort angemeldet werden.

3. Es gibt einen Aufsatz, der behauptet: “Breitensuche im Web liefert gute Webseiten” (Marc Najork and Janet L. Wiener. Breadth-first crawling yields high-quality pages. Proc. 10th World Wide Web Conference, Hong Kong, 2001).

   Begründen Sie, warum Breitensuche für Suchmaschinen eine gute Strategie ist, um relevante Suchresultate anbieten zu können!

   Lösungsvorschlag:

   Interessante Webseiten ziehen durch guten Inhalt viele Hyperlinks auf sich. Dadurch ist die Wahrscheinlichkeit hoch, dass sie früh in einer Breitensuche gefunden werden. Zudem hat das Web einen geringen Durchmesser.

   Andere Strategien, wie z. B. Tiefensuche, bergen die Gefahr, dass tief in weitverzweigte Unterstrukturen des Web vorgedrungen wird, ohne dabei näher liegende, relevante Seiten aufzufinden.

4. Betrachten Sie die Linkstruktur der folgenden Webseiten:

   Seite A verweist auf die Seiten C und D.
   Seite B verweist auf die Seiten F, E und G.
   Seite C verweist auf die Seiten F und E.
   Seite D verweist auf die Seite B.
   Seite E verweist auf die Seite F.

   In welcher Reihenfolge werden die Seiten in den Index bei Anwendung der Breitensuche eingefügt? Beginnen Sie mit der Seite A. Wie ist der Status der jeweiligen Seite bei Durchlauf der obigen Beschreibung?

   Lösungsvorschlag:

Web page visit order	Status
A (Start)	indiziert
C (von A)	indiziert
D (von A)	indiziert
E (von C)	indiziert
F (von C)	indiziert
B (von D)	indiziert
F (von E)	schon indiziert
E (von B)	schon indiziert
F (von B)	schon indiziert
G (von B)	indiziert

Die Reihenfolge der Indexierung ist: A, C, D, E, F, B, G.

3 Spidering im Hidden Web

1. Das sog. *Deep Web* oder *Hidden Web* umfasst diejenigen Seiten des WWW, die von "normalen" Spidern nicht gesehen werden. Dazu gehören Seiten, die erst nach Anmeldung sichtbar sind, oder solche, die erst durch Benutzerinteraktion erzeugt werden, z. B. Suchresultate in Online-Shops.

   Könnte man solche Seiten in den Index der Suchmaschine aufnehmen? Wenn ja, wie? Lösungsvorschlag:

   Eine Möglichkeit wäre, dass der Spider mit zusätzlichen Fähigkeiten ausgestattet wird, um Hidden-Web-Seiten zu erreichen. So könnte etwa ein Katalog mit Suchfunktion gecrawlt werden, in dem systematisch Suchbegriffe durchprobiert werden.

   Eleganter und ressourcenschonender wäre die Variante, dass ähnliche Techniken wie bei Metasuchmaschinen angewendet werden, um erst zum Suchzeitpunkt die Resultate der "normalen" Websuche mit der Suche im Hidden Web zu kombinieren.

   Ein Beispiel dafür ist z. B. `http://www.a9.com`. Dieser Dienst sucht im Web und parallel in anderen Quellen, beispielsweise im Katalog von Amazon.

2. Die bisher betrachteten Crawler haben Links extrahiert, indem der HTML-Quelltext

von Webseiten geparst und Elemente wie `<A>`, `<link>` usw. berücksichtigt werden.

a) Welche Bestandteile moderner Webseiten bleiben noch unberücksichtigt? Lösungsvorschlag:

   Dynamische Inhalte, wie z. B. durch Javascript nachgeladene und nachträglich in die Webseite eingebaute HTML-Fragmente, können mit dieser Technik nicht erfasst werden, da nur der statische HTML-Teil berücksichtigt wird.

b) Wie könnte man diese erfassen? Lösungsvorschlag:

   Ein Spider müßte alle Inhalte berücksichtigen, die auch ein Browser in einer interaktiven Benutzersitzung ausführt, also z. B. Flash-Objekte abspielen, Javascript ausführen, usw.

c) Warum wird dies in der Regel nicht gemacht? Lösungsvorschlag:

   Einerseits wäre eine solche Verarbeitung sehr rechenintensiv und damit zeitaufwendig, andererseits dienen viele solche dynamische Inhalte als optische Verzierung (z. B. animierte Intros usw.) und tragen nicht wesentlich zum Seiteninhalt – soweit es die Suche angeht – bei.

4 Spidering

1. Fokussiertes Spidering bedeutet, dass Webseiten, auf denen man Inhalte zu einem bestimmten Thema vermutet, bevorzugt eingesammelt werden.

   Wie müssen Sie den Algorithmus und die Datenstrukturen auf Seite 7 des Kapitels modifizieren, um fokussiertes Spidering zu realisieren?

   Lösungsvorschlag:

   - Die Datenstruktur Q muss statt LIFO oder FIFO nun eine Prioritäts-Warteschlange (*priority queue*) sein, die in der Lage ist, jeweils die URL mit der höchsten Relevanz zurückzugeben.
   - Am Ende des Algorithmus steht statt *Füge N an das Ende von Q an*:
     - Bewerte N mit einem Gewicht w anhand einer Bewertungsstrategie
     - Füge N mit dem Gewicht w in Q ein.

2. Wie können verschiedene Anforderungen an die Such-Reihenfolge, etwa: der Spider soll bevorzugt Seiten zum Thema “Hommingberger Gepardenforelle” sam-

meln und gleichzeitig nicht mehr als eine Seite pro Host pro Minute abrufen, in den Spidering-Algorithmus eingebaut werden?

Lösungsvorschlag:

- Möglichkeit 1: Die Vergleichsoperation, nach der die Prioritäts-Warteschlange aus Aufgabe **??** sortiert ist, muß entsprechend angepaßt werden. So könnte dann etwa gleichzeitig die Bandbreite des Hosts und die erwartete Relevanz der Zielseite in die Sortierung eingehen.
- Möglichkeit 2: Die Anforderung "begrenzte Zugriffshäufigkeit pro Host" kann dadurch erfüllt werden, dass parallel mehrere Warteschlangen verwendet werden, um hohen Durchsatz zu erreichen, von denen jede jedoch mit beschränkter Zugriffshäufigkeit läuft.

3. Skizzieren Sie eine Grobarchitektur für einen Spider, der die folgenden Möglichkeiten bietet:

   - Multithreading bei der Bearbeitung von HTTP-Requests (Netzwerklatenz!) und beim Extrahieren von Links.
   - Berücksichtigen von Robots.txt-Dateien
   - Filtern von unerwünschten URLs (z.B. CGI-Skripte)
   - Insgesamt möglichst hoher Durchsatz
   - Garantierte Wartezeit zwischen zwei Requests auf jedem Host

   Lösungsvorschlag:

   Skizze des Webcrawlers "Mercator" (aus: A. Heydon and M. Najork, "Mercator: A scalable, extensible web crawler," World Wide Web, vol. 2, no. 4, pp. 219–229, 1999).

   Die URL Frontier enthält eine Anzahl von Queues, entsprechend der Anzahl von HTTP-Workern. Jeder HTTP-Worker greift auf genau eine Queue zu. Die URLs werden so mittels Hashing auf die Queues verteilt, daß Seiten eines Hosts stets in der selben Queue verarbeitet werden. So wird eine Mindestwartezeit pro Host garantiert.

4. Welche Bestandteile eines Spiders könnten zum Flaschenhals werden? Nennen Sie drei Beispiele und machen Sie Vorschläge, wie z. B. durch Verteilung Abhilfe schaffen kann.

   Lösungsvorschlag:

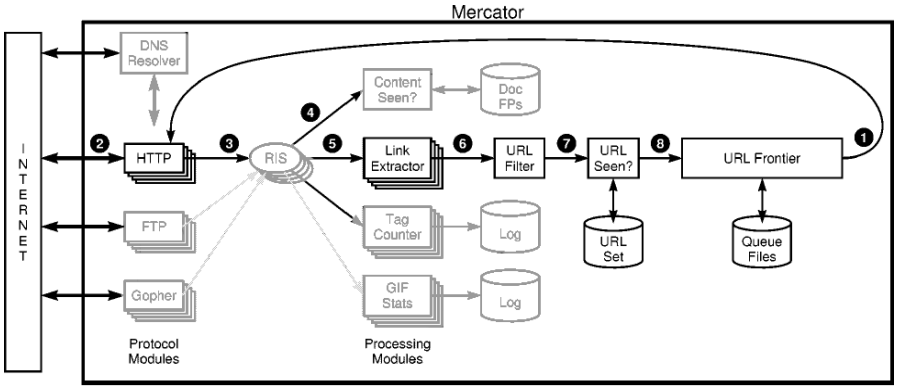

Figure 1. Mercator's main components.

DNS-Lookups Ein Spider muß sehr viele Hostnamen in IP-Nummern auflösen. In Mercator z. B. wurden DNS-Lookups komplett neuimplementiert, da die Standardlösung zu langsam war. Caching von DNS-Informationen kann weiterhin zur Leistungssteigerung beitragen.

Filesystem Da kontinuierlich große Datenmengen gespeichert werden müssen, kann das Filesystem zum Flaschenhals werden. Google etwa hat das "Google File System" entwickelt, um große Datenmengen verteilt auf tausende Rechner schnell und fehlertolerant ablegen zu können.

Verwaltung der Warteschlange Eine große Anzahl noch zu crawlender Seiten muß verwaltet werden. Wie oben in Mercator skizziert, kann hier durch Verteilung der Warteschlange ein Flaschenhals vermieden werden.

5. Betrachten Sie folgende Situation: es gibt im Web 10^{10} öffentlich abrufbare Seiten. Sie haben einen verteilten Crawler, der pro Rechner eine Webseite pro Sekunde abrufen kann. Sie möchten jede Seite im Web einmal pro Monat crawlen, um aktuell zu bleiben.

   Wieviele Rechner benötigen Sie dazu? Welche Gesamtbandbreite ist notwendig, wenn jede Webseite im Schnitt 20 kB groß ist?

   Lösungsvorschlag:

   Unter den gegebenen Vorausssetzungen kann ein Rechner $3600 \cdot 24 \cdot 30 = 2.592.000$ Seiten im Monat abrufen. Man benötigte also in diesem Fall $10^{10}/2.592.000 = 3.858$ Rechner, um die Anforderungen zu erfüllen.

   Jeder Rechner erzeugt dabei eine Netzwerklast von 20 kB/s, insgesamt benötigt man also etwa 77 MB/s.