How to Crawl the Web

Hector Garcia-Molina
Stanford University

Joint work with Junghoo Cho
Stanford InterLib Technologies

- Economic Concerns
  - IP Infrastructure
- Information Loss
  - Archival Repository
- Information Overload
  - Value Filtering
- Service Heterogeneity
  - Interoperability
- Physical Barriers
  - Mobile Access

InterLib Technologies

Stanford
WebBase Architecture

WWW

Repository
Multicast
Engine

WWW

Webbase API

Retrieval Indexes

Feature Repository

Repository

Multicast Engine

Web Crawler

Client
Client
Client
Client
Crawling at Stanford

- WebBase Project
- BackRub Search Engine, Page Rank
- Google
- New WebBase Crawler
Crawling Issues (1)

- Load at visited web sites
  - robots.txt files
  - space out requests to a site
  - limit number of requests to a site per day
  - limit depth of crawl
  - multicast
Crawling Issues (2)

- Load at crawler
  - parallelize

Diagram:
- init
- get next url
- get page
- extract urls
- visited urls
- to visit urls
- initial urls
- web pages
Crawling Issues (3)

- **Scope of crawl**
  - not enough space for “all” pages
  - not enough time to visit “all” pages

- **Solution: Visit “important” pages**
Crawling Issues (4)

- Incremental crawling
  - how do we keep pages “fresh”? 
  - how do we avoid crawling from scratch?

Focus of this talk...
Web Evolution Experiment

- How often does a web page change?
- What is the lifespan of a page?
- How long does it take for 50% of the web to change?
- How do we model web changes?
Experimental Setup

- **February 17 to June 24, 1999**
- **270 sites visited (with permission)**
  - identified 400 sites with highest “page rank”
  - contacted administrators
- **720,000 pages collected**
  - 3,000 pages from each site daily
  - start at root, visit breadth first (get new & old pages)
  - ran only 9pm - 6am, 10 seconds between site requests
How Often Does a Page Change?

- Example: 50 visits to page, 5 changes
  \[ \Rightarrow \text{average change interval} = \frac{50}{5} = 10 \text{ days} \]

- Is this correct?

![Diagram showing changes over time with a 1-day interval and page visits marked with dashed lines.](image-url)
Average Change Interval

<table>
<thead>
<tr>
<th>Interval</th>
<th>Fraction of Pages</th>
</tr>
</thead>
<tbody>
<tr>
<td>≤1 day</td>
<td>0.23</td>
</tr>
<tr>
<td>&gt;1 day, ≤1 week</td>
<td>0.15</td>
</tr>
<tr>
<td>&gt;1 week, ≤1 month</td>
<td>0.15</td>
</tr>
<tr>
<td>&gt;1 month, ≤4 months</td>
<td>0.15</td>
</tr>
<tr>
<td>&gt;4 months</td>
<td>0.30</td>
</tr>
</tbody>
</table>
Average Change Interval — By Domain

Fraction of pages

Days

≤ 1 day > 1 day

≤ 1 week > 1 week

≤ 1 month > 1 month

≤ 4 months > 4 months

com
netorg
edu
gov
How Long Does a Page Live?
Page Lifespans
Page Lifespans

Method 1 used

fraction of pages
Time for a 50% Change
Modeling Web Evolution

- Poisson process with rate $\lambda$
- $T$ is time to next event
- $f_T(t) = \lambda e^{-\lambda t}$ \hspace{1em} ($t > 0$)
Change Interval of Pages

for pages that change every 10 days on average

Poisson model
Change Metrics

- **Freshness**
  - Freshness of element $e_i$ at time $t$ is
    \[
    F(e_i; t) = \begin{cases} 
      1 & \text{if } e_i \text{ is up-to-date at time } t \\
      0 & \text{otherwise}
    \end{cases}
    \]
  - Freshness of the database $S$ at time $t$ is
    \[
    F(S; t) = \frac{1}{N} \sum_{i=1}^{N} F(e_i; t)
    \]
Change Metrics

- **Age**
  - Age of element $e_i$ at time $t$ is

  $$A(e_i ; t) = \begin{cases} 
  0 & \text{if } e_i \text{ is up-to-date at time } t \\
  t - (\text{modification } e_i \text{ time}) & \text{otherwise}
  \end{cases}$$

  - Age of the database $S$ at time $t$ is

  $$A(S ; t) = \frac{1}{N} \sum_{i=1}^{N} A(e_i ; t)$$
Change Metrics

Time averages:

\[ \overline{F}(e_i) = \lim_{t \to \infty} \frac{1}{t} \int_{0}^{t} F(e_i; t) \, dt \]

\[ \overline{F}(S) = \lim_{t \to \infty} \frac{1}{t} \int_{0}^{t} F(S; t) \, dt \]

similar for age...
Crawler Types

- **In-place vs. shadow**
  - In-place:
    - Web: $e_i$
    - Database: $e_i$
  - Shadow:
    - Database: $e_i$

- **Steady vs. batch**
  - Crawler on
  - Crawler off
  - Time
Comparison: Batch vs. Steady

Batch mode:
- crawler running
- freshness peaks

Steady mode:
- crawler running
- freshness remains constant
Shadowing Steady Crawler

![Diagram showing freshness over time for crawler's collection and current collection with and without shadowing.]

- The top graph illustrates the freshness of the crawler's collection over time, showing an increase in freshness as time progresses from 0.5 to 3.
- The bottom graph shows the freshness of the current collection, indicating a decrease in freshness over the same time period.
- The graph labeled "without shadowing" highlights the difference in freshness compared to the graph labeled "shadowing."
Shadowing Batch Crawler

Diagram showing the comparison between a crawler’s collection and the current collection. The graphs display freshness over time, with shaded areas indicating periods without shadowing.
## Experimental Data → Freshness

<table>
<thead>
<tr>
<th></th>
<th>Steady</th>
<th>Batch</th>
</tr>
</thead>
<tbody>
<tr>
<td>In-Place</td>
<td>0.88</td>
<td>0.88</td>
</tr>
<tr>
<td>Shadowing</td>
<td>0.77</td>
<td>0.86</td>
</tr>
</tbody>
</table>

- Pages change on average every 4 months
- Batch crawler works one week out of 4
Refresh Order

- **Fixed order**
  - Example: Explicit list of URLs to visit

- **Random Order**
  - Example: Start from seed URLs & follow links

- **Purely Random**
  - Example: Refresh pages on demand, as requested by user
Freshness vs. Order

*steady, in-place crawler, pages change at same average rate*

\[ r = \frac{\lambda}{f} = \text{average change frequency} / \text{average visit frequency} \]
Age vs. Order

Age/I = Age / time to refresh all N elements

- fixed-order
- random-order
- purely-random

\[ r = \frac{\lambda}{f} = \text{average change frequency} / \text{average visit frequency} \]
Non-Uniform Change Rates

\[ \text{fraction of elements} \]

\[ g(\lambda) \]

\[ \lambda \]
Trick Question

- Two page database
- $e_1$ changes daily
- $e_2$ changes once a week
- Can visit pages once a week
- How should we visit pages?
  - $e_1 e_1 e_1 e_1 e_1 e_1 \ldots$
  - $e_2 e_2 e_2 e_2 e_2 \ldots$
  - $e_1 e_2 e_1 e_2 e_1 e_2 \ldots$ [uniform]
  - $e_1 e_1 e_1 e_1 e_1 e_2 e_1 e_1 \ldots$ [proportional]
  - ?
Proportional Often Not Good!

- Visit fast changing $e_1 \implies$ get 1/2 day of freshness
- Visit slow changing $e_2 \implies$ get 1/2 week of freshness
- Visiting $e_2$ is a better deal!
Selecting Optimal Refresh Frequency

- Analysis is complex
- Shape of curve is the same in all cases
- Holds for any distribution \( g(\lambda) \)
Optimal Refresh Frequency for Age

- Analysis is also complex
- Shape of curve is the same in all cases
- Holds for any distribution $g(\lambda)$
Comparing Policies

<table>
<thead>
<tr>
<th>Policy</th>
<th>Freshness</th>
<th>Age</th>
</tr>
</thead>
<tbody>
<tr>
<td>Proportional</td>
<td>0.12</td>
<td>400 days</td>
</tr>
<tr>
<td>Uniform</td>
<td>0.57</td>
<td>5.6 days</td>
</tr>
<tr>
<td>Optimal</td>
<td>0.62</td>
<td>4.3 days</td>
</tr>
</tbody>
</table>

In-place, steady crawler;
Based on our experimental data
[Pages change at different frequencies, as measured in experiment.]
Building an Incremental Crawler

- Steady
- In-place
- Variable visit frequencies
- Fixed order

Also need:
- Page change frequency estimator
- Page replacement policy
- Page ranking policy (what are “important” pages?)

See papers....
The End

- Thank you for your attention...
- For more information, visit http://www-db.stanford.edu
  Follow “Searchable Publications” link, and search for author = “Cho”