Overview

- Introduction
- URL Frontier
- Robust Crawling
- DNS
Introduction
Introduction

The basic crawl algorithm

- Initialize a queue of URLs ("seed" URLs)
- Repeat
  - Remove a URL from the queue
  - Fetch associated page
  - Parse and analyze page
  - Store representation of page
  - Extract URLs from page and add to queue
Introduction

Crawling the web

Seed Pages

Crawled Pages

URL Frontier

Web Spider

The Rest of the Web
Basic Algorithm is not reality...

- Real web crawling requires multiple machines
- All steps distributed on different computers
- Even Non-Adversarial pages pose problems
  - Latency and bandwidth to remote servers vary
  - Webmasters have opinions about crawling their turf
    - How “deep” in a URL should you go?
- Site mirrors and duplicate pages
- Politeness
  - Don’t hit a server too often
Basic Algorithm is not reality...

- Adversarial Web Pages
- Spam Pages
- Spider Traps
Minimum Characteristics for a Web Crawler

• Be Polite:
  • Respect implicit and explicit terms on website
  • Crawl pages you’re allowed to
  • Respect “robots.txt” (more on this coming up)

• Be Robust
  • Handle traps and spam gracefully
Desired Characteristics for a Web Crawler

- Be a distributed systems
  - Run on multiple machines
- Be scalable
  - Adding more machines allows you to crawl faster
- Be Efficient
  - Fully utilize available processing and bandwidth
- Focus on “Quality” Pages
  - Crawl good information first
Desired Characteristics for a Web Crawler

- Support Continuous Operation
  - Fetch fresh copies of previously crawled pages
- Be Extensible
  - Be able to adapt to new data formats, protocols, etc.
  - Today it’s AJAX, tomorrow it’s SilverLight, then....
• Frontier Queue might have multiple pages from the same host
• These need to be load balanced ("politeness")
• All crawl threads should be kept busy
Politeness?

- It is easy enough for a website to block a crawler
- Explicit Politeness
  - "Robots Exclusion Standard"
    - Defined by a "robots.txt" file maintained by a webmaster
    - What portions of the site can be crawled.
      - Irrelevant, private or other data excluded.
    - Voluntary compliance by crawlers.
    - Based on regular expression matching
Politeness?

- Explicit Politeness
- “Sitemaps”
  - Introduced by Google, but open standard
  - XML based
  - Allows webmasters to give hints to web crawlers:
    - Location of pages (URL islands)
    - Relative importance of pages
    - Update frequency of pages
  - Sitemap location listed in robots.txt
Politeness?

- Implicit Politeness
  - Even without specification avoid hitting any site too often
  - It costs bandwidth and computing resources for host.
### Politeness?

**Statistics for:** djp3.net

**Last Update:** 14 Jan 2008 - 02:59
**Reported period:** 2007

<table>
<thead>
<tr>
<th>Robots/Spiders visitors</th>
<th>Hits</th>
<th>Bandwidth</th>
<th>Last visit</th>
</tr>
</thead>
<tbody>
<tr>
<td>Googlebot</td>
<td>1393868+104</td>
<td>5.11 GB</td>
<td>31 Dec 2007 - 23:50</td>
</tr>
<tr>
<td>Inktomi Slurp</td>
<td>36668+221</td>
<td>554.25 MB</td>
<td>31 Dec 2007 - 23:55</td>
</tr>
<tr>
<td>MSNBot</td>
<td>19522+2</td>
<td>699.90 MB</td>
<td>28 Dec 2007 - 08:01</td>
</tr>
<tr>
<td>Unknown robot (identified by 'crawl')</td>
<td>15949+13</td>
<td>89.34 MB</td>
<td>31 Dec 2007 - 22:24</td>
</tr>
<tr>
<td>AskJeeves</td>
<td>7016+1</td>
<td>106.29 MB</td>
<td>31 Dec 2007 - 23:49</td>
</tr>
<tr>
<td>Google AdSense</td>
<td>2701</td>
<td>100.26 MB</td>
<td>31 Dec 2007 - 22:10</td>
</tr>
<tr>
<td>psbot</td>
<td>2268+1</td>
<td>80.48 MB</td>
<td>31 Dec 2007 - 09:59</td>
</tr>
<tr>
<td>Unknown robot (identified by 'robot')</td>
<td>930+1</td>
<td>19.10 MB</td>
<td>31 Dec 2007 - 09:34</td>
</tr>
<tr>
<td>Turn It In</td>
<td>350+1</td>
<td>6.32 MB</td>
<td>03 Sep 2007 - 15:44</td>
</tr>
<tr>
<td>BaiDuSpider</td>
<td>300</td>
<td>10.22 MB</td>
<td>26 Nov 2007 - 07:32</td>
</tr>
<tr>
<td>GigaBot</td>
<td>243</td>
<td>5.27 MB</td>
<td>30 Dec 2007 - 05:06</td>
</tr>
<tr>
<td>Scooter</td>
<td>90+3</td>
<td>288.75 KB</td>
<td>27 Nov 2007 - 14:30</td>
</tr>
<tr>
<td>PhpDig</td>
<td>91</td>
<td>2.28 MB</td>
<td>21 Oct 2007 - 09:51</td>
</tr>
<tr>
<td>WISENutbot</td>
<td>76</td>
<td>1.94 MB</td>
<td>13 Jan 2007 - 14:04</td>
</tr>
<tr>
<td>Magpie</td>
<td>25</td>
<td>43.48 KB</td>
<td>24 Dec 2007 - 00:51</td>
</tr>
<tr>
<td>Unknown robot (identified by hit on 'robots.txt')</td>
<td>0+16</td>
<td>4.38 KB</td>
<td>14 Nov 2007 - 03:43</td>
</tr>
<tr>
<td>EchoO!</td>
<td>14</td>
<td>287.09 KB</td>
<td>27 Dec 2007 - 13:56</td>
</tr>
<tr>
<td>Internet Shinchakubin</td>
<td>13</td>
<td>385.03 KB</td>
<td>27 Nov 2007 - 15:23</td>
</tr>
<tr>
<td>BBot</td>
<td>10</td>
<td>146.35 KB</td>
<td>13 Jun 2007 - 15:17</td>
</tr>
<tr>
<td>arks</td>
<td>8</td>
<td>142.24 KB</td>
<td>27 Nov 2007 - 12:25</td>
</tr>
<tr>
<td>MSIECrawler</td>
<td>8</td>
<td>263.02 KB</td>
<td>26 Dec 2007 - 11:16</td>
</tr>
<tr>
<td>The Python Robot</td>
<td>5</td>
<td>428.91 KB</td>
<td>23 Nov 2007 - 09:04</td>
</tr>
</tbody>
</table>

Back to main page
Politeness?

Monthly history

<table>
<thead>
<tr>
<th>Month</th>
<th>Unique visitors</th>
<th>Number of visits</th>
<th>Pages</th>
<th>Hits</th>
<th>Bandwidth</th>
</tr>
</thead>
<tbody>
<tr>
<td>Jan 2007</td>
<td>1221</td>
<td>2946</td>
<td>8938</td>
<td>30536</td>
<td>699.28 MB</td>
</tr>
<tr>
<td>Feb 2007</td>
<td>1179</td>
<td>3099</td>
<td>7852</td>
<td>20475</td>
<td>415.75 MB</td>
</tr>
<tr>
<td>Mar 2007</td>
<td>1120</td>
<td>3063</td>
<td>7099</td>
<td>18978</td>
<td>350.88 MB</td>
</tr>
<tr>
<td>Apr 2007</td>
<td>1362</td>
<td>3067</td>
<td>7175</td>
<td>30320</td>
<td>599.91 MB</td>
</tr>
<tr>
<td>May 2007</td>
<td>1612</td>
<td>3746</td>
<td>7584</td>
<td>25114</td>
<td>469.32 MB</td>
</tr>
<tr>
<td>Jun 2007</td>
<td>1474</td>
<td>3662</td>
<td>7138</td>
<td>22292</td>
<td>370.11 MB</td>
</tr>
<tr>
<td>Jul 2007</td>
<td>1592</td>
<td>4210</td>
<td>9165</td>
<td>24766</td>
<td>430.61 MB</td>
</tr>
<tr>
<td>Aug 2007</td>
<td>1658</td>
<td>4567</td>
<td>10600</td>
<td>24142</td>
<td>336.08 MB</td>
</tr>
<tr>
<td>Sep 2007</td>
<td>1458</td>
<td>4403</td>
<td>11149</td>
<td>24414</td>
<td>356.60 MB</td>
</tr>
<tr>
<td>Oct 2007</td>
<td>2148</td>
<td>5299</td>
<td>12877</td>
<td>36427</td>
<td>783.78 MB</td>
</tr>
<tr>
<td>Nov 2007</td>
<td>2890</td>
<td>6317</td>
<td>15300</td>
<td>40487</td>
<td>833.75 MB</td>
</tr>
<tr>
<td>Dec 2007</td>
<td>2748</td>
<td>6631</td>
<td>17553</td>
<td>42281</td>
<td>1.03 GB</td>
</tr>
<tr>
<td>Total</td>
<td>20462</td>
<td>51010</td>
<td>122430</td>
<td>340232</td>
<td>6.55 GB</td>
</tr>
</tbody>
</table>
Robots.txt - Exclusion

- Protocol for giving spiders ("robots") limited access to a website
  - Source: http://www.robotstxt.org/wc/norobots.html
- Website announces what is okay and not okay to crawl:
  - Located at http://www.myurl.com/robots.txt
  - This file holds the restrictions
Robots.txt Example

- http://www.ics.uci.edu/robots.txt

```text
User-agent: MOMspider
Disallow: /cgi-bin/
Disallow: /Admin/MOM/
Disallow: ~/fielding/MOM/
Disallow: /TR/
Disallow: /Server/
Disallow: /Document/
Disallow: /MetaServer/
Disallow: ~/eppstein/pubs/cites/
Disallow: fiorello/pvt/

User-agent: *
Disallow: /cgi-bin/
Disallow: /Test/
Disallow: /Admin/
Disallow: /TR/
Disallow: /Server/
Disallow: /Document/
Disallow: /MetaServer/
Disallow: ~/fielding/MOM/
Disallow: ~/kanderso/hidden/
Disallow: ~/eppstein/pubs/cites/
Disallow: fiorello/pvt/
Disallow: ~/dean/
Disallow: ~/wwwoffice/
Disallow: ~/ucounsel/
Disallow: ~/sao/
Disallow: ~/support/
Disallow: ~/icsdb/
Disallow: /bin/
```

# The Multi-Owner Maintenance Spider
# Script files
# Local MOMspider output
# Local MOMspider output
# Dienst Technical Report Server
# Dienst Technical Report Server
# Dienst Technical Report Server
# Dienst Technical Report Server
# Dienst Technical Report Server

# All other spiders should avoid
# Script files
# The test area for web experimentation
# Huge server statistic logs
# Dienst Technical Report Server
# Dienst Technical Report Server
# Dienst Technical Report Server
# Dienst Technical Report Server
# Local MOMspider output
# Ken Anderson's stuff
# Eppstein Database
# Eppstein Database

Private pages
Sitemaps - Inclusion


```xml
<?xml version="1.0" encoding="UTF-8"?>
<urlset xmlns="http://www.sitemaps.org/schemas/sitemap/0.9">
  <url>
    <loc>http://www.example.com/</loc>
    <lastmod>2005-01-01</lastmod>
    <changefreq>monthly</changefreq>
    <priority>0.8</priority>
  </url>
  <url>
    <loc>http://www.example.com/catalog?item=12&amp;desc=vacation_hawaii</loc>
    <changefreq>weekly</changefreq>
  </url>
  <url>
    <loc>http://www.example.com/catalog?item=73&amp;desc=vacation_new_zealand</loc>
    <lastmod>2004-12-23</lastmod>
    <changefreq>weekly</changefreq>
  </url>
  <url>
    <loc>http://www.example.com/catalog?item=74&amp;desc=vacation_newfoundland</loc>
    <lastmod>2004-12-23T18:00:15+00:00</lastmod>
    <priority>0.3</priority>
  </url>
  <url>
    <loc>http://www.example.com/catalog?item=83&amp;desc=vacation_usa</loc>
    <lastmod>2004-11-23</lastmod>
  </url>
</urlset>
```
Web Crawling Outline

Overview

• Introduction
• URL Frontier
• Robust Crawling
  • DNS
A Robust Crawl Architecture

1. WWW
2. DNS
3. Fetch
4. Parse
5. Seen?
6. Doc. Fingerprints
7. Robots.txt
8. URL Filter
9. URL Index
10. Duplicate Elimination
11. URL Frontier Queue
Processing Steps in Crawling

- Pick a URL from the frontier (how to prioritize?)
- Fetch the document (DNS lookup)
- Parse the URL
  - Extract Links
- Check for duplicate content
  - If not add to index
- For each extracted link
  - Make sure it passes filter (robots.txt)
  - Make sure it isn’t in the URL frontier
Domain Name Server

- A lookup service on the internet
- Given a URL, retrieve its IP address
  - www.djp3.net -> 69.17.116.124
- This service is provided by a distributed set of servers
- Latency can be high
  - Even seconds
- Common OS implementations of DNS lookup are blocking
  - One request at a time
- Solution:
  - Caching
  - Batch requests
Where is www.djp3.net?

1. Ask 192.5.6.30
   {A}.ROOT-SERVERS.NET = 198.41.0.4

2. Ask 72.1.140.145
   {A}.GTLD-SERVERS.net = 192.5.6.30

3. Use 69.17.116.124
   djp3.net Name Server
   {ns1}.speakeasy.net = 72.1.140.145

4. Give me a web page
   www.djp3.net = 69.17.116.124

www.djp3.net = 69.17.116.124
What really happens

The User

flickr:crankyT

Give me a www.djp3.net

The User

flickr:crankyT

Give me a www.djp3.net
DNS

Class Exercise

http://www.flickr.com/photos/lurie/298967218/
DNS

Class Exercise

http://www.flickr.com/photos/lurie/298967218/
Class Exercise

• Calculate how long it would take to completely fill a DNS cache.

http://www.flickr.com/photos/lurie/298967218/
Class Exercise

- Calculate how long it would take to completely fill a DNS cache.
- How many active hosts are there?
Class Exercise

• Calculate how long it would take to completely fill a DNS cache.
• How many active hosts are there?
• What is an average lookup time?
Class Exercise

- Calculate how long it would take to completely fill a DNS cache.
- How many active hosts are there?
- What is an average lookup time?
- Do the math.