AltaVista

Richard L. Sites
Digital Equipment Corp.
Palo Alto, CA

DECUS, September 1996
Acknowledgment

AltaVista was created by Mike Burrows and Louis Monier, with the help of many other people.
AltaVista Outline

- Fetch web pages (Scooter)
- Build web index
- The index itself
- Lookup in web index
- Queries from the Internet
- News groups fetch/build/index/lookup
- Statistics
- Conclusions
Fetch Web Pages (Scooter)

- Starting with some URL, fetch that page
- Respect robot exclusion standard
- Deliver page to index builder
- Find all contained URLs
- Add to list of URLs to be fetched
- Do not allow duplicates on the list
- Do not visit the same site very often
- Take first unfetched on list URL and loop
Fetch: original source HTML

Those... <a href="news:alt.folklore.urban">urban legends</a>. ...
original <a href="whalestory.html">e-mail</a>, ...
<br/>
<center><img src="line.gif"></center>
<br/>
... there is the <a href="http://alpha.mic.dundee.ac.uk/ft/july/whale2.avi">full news report</a>, ...
<br/>
... Quicktime ... <a href="ftp://ftp.xmission.com/pub/users/g/grue/whale.qt">on this link</a>. ...
There have been <img src="http://www-hons-cs.dcs.st-and.ac.uk/cgi-bin/nph-count?width=6&link=www.st-and.ac.uk/~www_sa/personal/fs1/whale.html"> visitors here ...
Fetch: what Scooter sees

http://www.st-and.ac.uk/~www_sa/personal/fs1/whale.html

Those... <a href="news:alt.folklore.urban">urban legends</a>. ... original <a href="whalestory.html">e-mail</a>, ...

<center><img src="line.gif"></center>
<br>... there is the <a href="http://alpha.mic.dundee.ac.uk/ft/july/whale2.avi">full news report</a>, ...
<br>... Quicktime ... <a href="ftp://ftp.xmission.com/pub/users/g/grue/whale.qt">on this link</a>. ... There have been <img src="http://www-hons-cs.dcs.st-and.ac.uk/cgi-bin/nph-count?width=6&link=www.st-and.ac.uk/~www_sa/personal/fs1/whale.html"> visitors here ...
Fetch: what Scooter *does*

http://www.st-and.ac.uk/~www_sa/personal/fs1/whale.html

news: alt.folklore.urban ignore
whalestory.html ADD
line.gif ignore
http://alpha.mic.dundee.ac.uk/ft/july/whale2.avi ignore
ftp://ftp.xmission.com/pub/users/g/grue/whale.qt ignore
Fetch: details

- Scooter only indexes files ending in
  - .html, .htm, .text, .txt
- Only indexes files with
  - no more than 8 levels of directory
- If it takes time $t$ to fetch a page, wait $100\times t$
  to fetch again from same site
  - guarantees less than 1% load on any site
- Details will be refined over time
Fetch: details

- URL duplicate lookup:
  - 50M URLs at 50 characters each = 2.5GB
  - Too big to keep in memory on small machine
  - 64-bit URL signature used instead
  - Partitioned so average 6 bytes each = 300 MB
  - If new URL signature matches existing one, don’t add to URL list
Fetch: details

- Scooter runs with about 800 threads
- Each is fetching a page from somewhere in the world
- It takes about 5 days to build a full index from scratch
Build Web Index

- Fetch passes each page to index builder
Build: original source file

<html>
<head><title>How to deal with a beached whale</title></head>
<META Name="description" Content="The fabulous story of The Exploding Whale in full colour detail. There are pictures and video to explore.">

<H3>The Story of the</H3><BR>
<h1><i>Exploding Whale</i></h1>
...
<img src="whale.gif" width=200 height=152 hspace=40 align=right><p>
<br>It was a big whale.
<br><br>It was a smelly whale.
<br><br>Most importantly, it was a dead whale.
</p></html>
Build: AltaVista sees *just words*

**ORIGINAL:**

```html
<html>
<head><title>How to deal with a beached whale</title></head>
<META Name="description" Content="The fabulous story of The Exploding Whale in full colour detail. There are pictures and video to explore.">
</html>
```

**ALTAVISTA SEES:**

```html
<html>
<head><title>How to deal with a beached whale</title></head>
<META Name="description" Content="The fabulous story of The Exploding Whale in full colour detail. There are pictures and video to explore.">
</html>
```
Build: *just words*

**ORIGINAL:**

```html
<H3>The Story of the</H3><BR>
<h1><i>Exploding Whale</i></h1>
<img src="whale.gif" width=200 height=152 hspace=40 align=right><p>
<br>It was a big whale.
<br><br>It was a smelly whale.
<br><br>Most importantly, it was a dead whale.
```

**ALTAVISTA SEES:**

```html
<H3>The Story of the</H3><BR>
<h1><i>Exploding Whale</i></h1>
<img src="whale.gif" width=200 height=152 hspace=40 align=right><p>
<br>It was a big whale.
<br><br>It was a smelly whale.
<br><br>Most importantly, it was a dead whale
```
Build: case & accents

Voilà  le  Printemps
Voila  printemps
voilà
voila

Index original word, original without accents, without uppercase, and without either
Build: sees all these words

The Story of the story

Exploding Whale exploding whale

It was a big whale it

It was a smelly whale. it

Most importantly, it was a dead whale ♣ most
Build: numbers the words

The Story of the
the story
1 2 3 4

Exploding Whale
exploding whale
5 6

It was a big whale
it
7 8 9 10 11

It was a smelly whale.
it
12 13 14 15 16

Most importantly, it was a dead whale. ♣
most
17 18 19 20 21 22 23 24 25 26
Build: list where each word is

The Story of the
the story
1 2 3 4
Exploding Whale
exploding whale
5 6

It was a big whale
it
7 8 9 10 11

It was a smelly whale.
it
12 13 14 15 16

Most importantly, it was a dead whale
most
17 18 19 20 21 22 23 24

September, 1996  p19
Build: the word index

The Story of the

Exploding Whale

It was a big whale

It was a smelly whale.

Most importantly, it was a dead whale.
## Build: one web page, word index

<table>
<thead>
<tr>
<th>Word</th>
<th>Occurrences</th>
</tr>
</thead>
<tbody>
<tr>
<td>a</td>
<td>9 14 21</td>
</tr>
<tr>
<td>big</td>
<td>10</td>
</tr>
<tr>
<td>dead</td>
<td>22</td>
</tr>
<tr>
<td>exploding</td>
<td>5</td>
</tr>
<tr>
<td>Exploding</td>
<td>5</td>
</tr>
<tr>
<td>gif</td>
<td>25</td>
</tr>
<tr>
<td>importantly</td>
<td>18</td>
</tr>
<tr>
<td>it</td>
<td>7 12 19</td>
</tr>
<tr>
<td>Its</td>
<td>7 12</td>
</tr>
<tr>
<td>most</td>
<td>17</td>
</tr>
<tr>
<td>Most</td>
<td>17</td>
</tr>
<tr>
<td>of</td>
<td>3</td>
</tr>
<tr>
<td>smell</td>
<td>15</td>
</tr>
<tr>
<td>story</td>
<td>2</td>
</tr>
<tr>
<td>Story</td>
<td>2</td>
</tr>
<tr>
<td>the</td>
<td>1 4</td>
</tr>
<tr>
<td>the</td>
<td>1</td>
</tr>
<tr>
<td>was</td>
<td>8 13 20</td>
</tr>
<tr>
<td>whale</td>
<td>6 11 16 23 24</td>
</tr>
<tr>
<td>Whale</td>
<td>6</td>
</tr>
<tr>
<td>♠</td>
<td>26</td>
</tr>
</tbody>
</table>
Imagine placing 30M web pages end-to-end as 13 billion words, then building a full word index:

Build: \textit{the full index}

\begin{table}[h]
\centering
\begin{tabular}{|c|c|}
\hline
Word & Word Number \\
\hline
\texttt{a} & 9 14 21 345 7012 7122 400123 ... \\
\texttt{the} & 1 4 35 ... 12999888777 \\
\texttt{zzz} & 2444888 ... \\
\texttt{999} & ... \\
\texttt{♣} & 26 258 860 1792 ... \\
\hline
\end{tabular}
\end{table}
The Index Itself (96/8/29)

- 30M pages, 13B words, 45GB on disk
  - Duplicate (identical) pages indexed only once
- Just the word lists, in alphabetical order
  - “the” has 380,383,961 entries
  - Also a second-level index of just the words
- 45GB cached in 6GB of main memory
  - UNIX mmap of 45GB, paged in/out
- **No** 32-bit machine can do this!
Index: on 7 machines

- 45 GB
  - 6 GB
    - 10 CPUs
- 45 GB
  - 6 GB
    - 10 CPUs
- 45 GB
  - 6 GB
    - 8 CPUs
Index: eventually, on 7 machines
Lookups use word lists in index
Lookup in Web Index

- Run through lists for each word in query
- If page has right combination, save URL
- Sort URLs by weighting function
  - Words near front
  - Words repeated
  - Words close together
- Deliver first 200 back to user
  - add summary text
Lookup in Web Index

◆ Typical lookup takes 1/2 second
◆ Typical lookup takes 50 disk accesses
◆ Each TurboLaser has 8 or 10 CPUs, does about 40 lookups & 2000 page faults per second
◆ Result page points to *originals* (not stored at Digital)
◆ Weighting details will vary
Queries from the Internet

- Incoming queries come to one of three front-end machines
- Front-ends: initial text, help, etc. (50%)
- TurboLasers: web lookups (45%)
- Others: Usenet news lookups (5%)
- FDDI ring connects them all
- Load balancing/failover
Queries: what goes wrong

- Someone: 1000 queries/sec
- Someone: 1..5000 character query
- Many: entire web page as query
- Schools: dozens of simultaneous identical queries
- Major airline: pages duplicated (unauthorized and out of date)
Lookup in News Index

- Lookups use word lists in index
News groups: fetch/build/index/lookup

- Fetch: from 14,000 news groups
- Build: continuously add/delete
- Index: 100x smaller than web
- Lookup: same code
- Result page points to *copies* stored at Digital, and also to *original*
  - via your news server ("L")
- Load balancing/failover
Statistics: AltaVista load

Accesses/Day (millions)

12/14/95
12/28/95
1/11/96
1/25/96
2/8/96
2/22/96
3/7/96
3/21/96
4/4/96
4/18/96
5/2/96
5/16/96

0 2 4 6 8 10 12 14

2 4 8 10 CPUs

20 CPUs

28

36
AltaVista has changed the world!

- Find information
- Answer questions
- Find pictures
- Find long-lost friends
- Correlate newsgroup postings
- Sell products
- Increase commerce
AltaVista

- Fastest, most comprehensive web search
- Free, no advertisements
- Products spinning off:
  - Index corporate intranets
  - Index your PC disks
  - Index mail
  - Mirror sites ...

- Alpha computers, Digital UNIX
- NO 32-bit machine can do this!