Effective Compression for the Web: Exploiting Document Linkages

Raymond Wan Alistair Moffat

rwan@cs.mu.oz.au alistair@cs.mu.oz.au

Department of Computer Science and
Software Engineering
The University of Melbourne

1 February 2001[†]

[†] Twelfth Australasian Database Conference 2001

Outline

Motivation

Base compression

Method 1: Use of common priming text

Method 2: Exploiting document linkages

Test website

Simulation results

Conclusion

Motivation

Some text compression algorithms are:

- Ziv-Lempel (i.e., GZIP, COMPRESS)
- Prediction by Partial Matching (PPM)
- Burrows-Wheeler Transform with Move-to-Front (BWT) (i.e., BZIP2)

Good compression ratios for large text files when allowed a long "learn" period

But, HTML files are small, typically 10 kB in size

Motivation (continued)

Purpose: to improve the compression of small documents on the web

Simulations on a website conducted using the following methods:

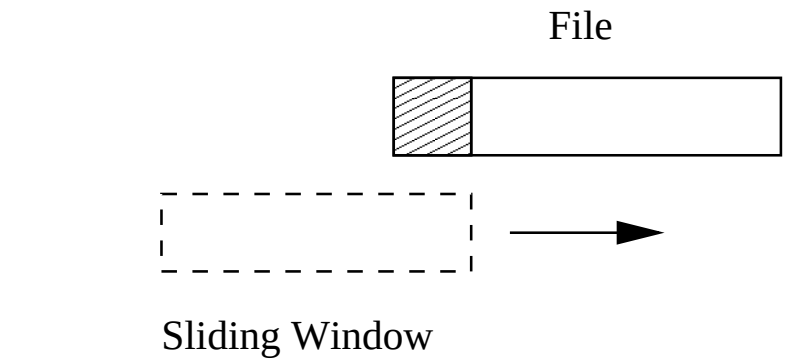
- Use of common priming text
- Exploiting document linkages

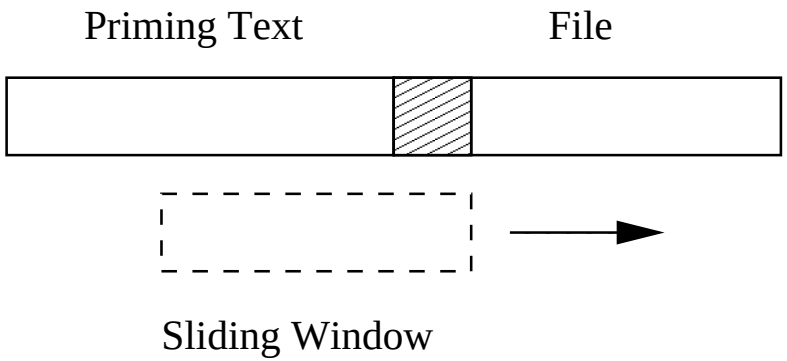
Base compression

GZIP was chosen as an underlying compression algorithm because:

- Even though it does not perform as well as PPM, it is faster and uses less memory
- Unlike BZIP2, it is an online compression algorithm

Base compression (continued)





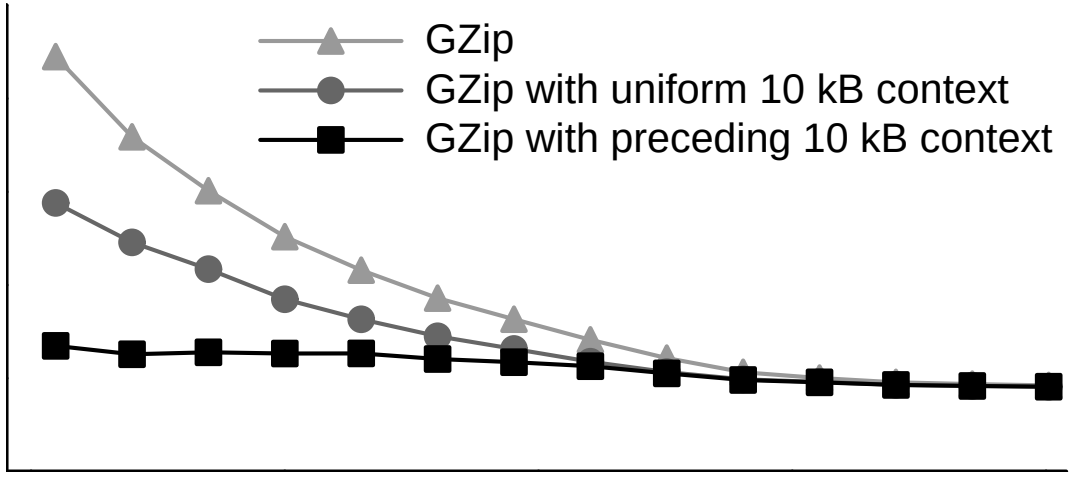
Base compression (continued)

Example: 20 MB of text marked up in SGML

Partition into 20 sections

Fragments made from the initial bytes of each section

Each of the last 19 fragments compressed and averaged for each point



Method 1: Use of common priming text

Priming texts made as follows:

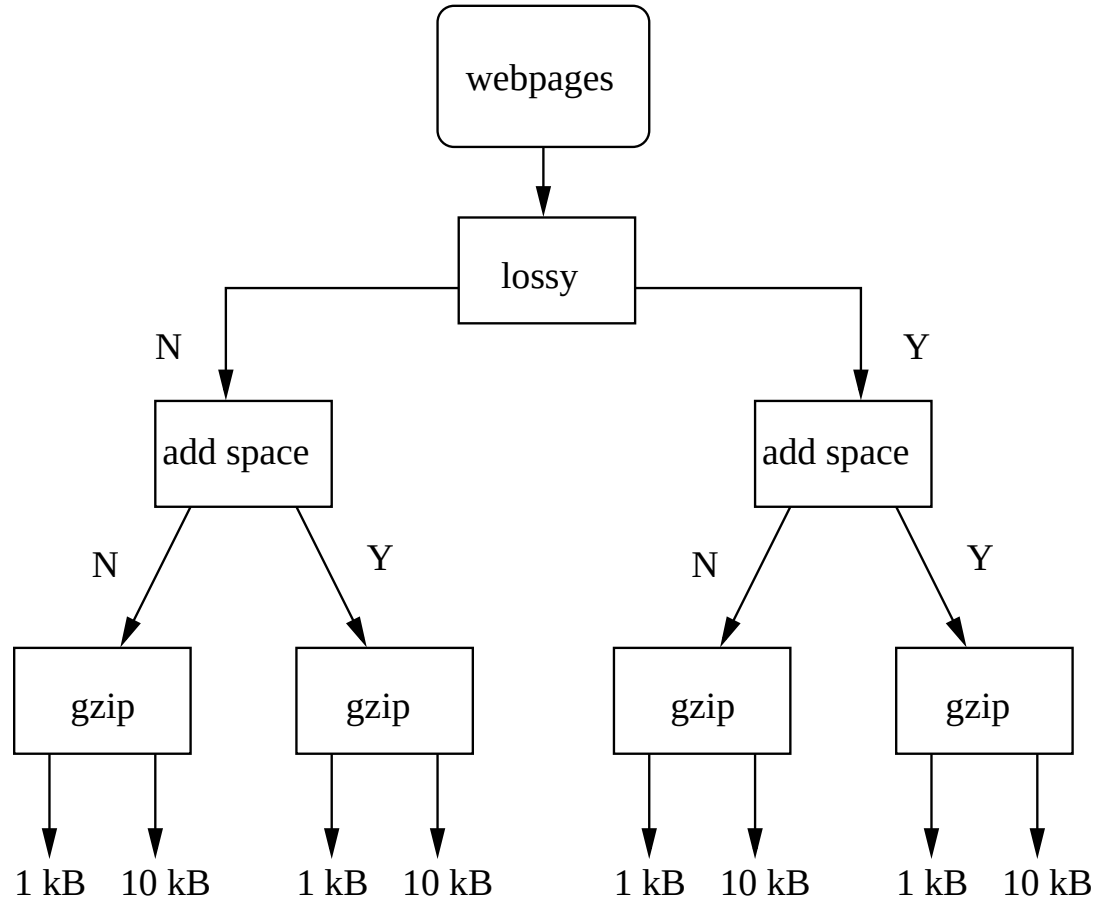
1. Optional lossy filtering performed
2. List of words occurring in the most HTML files sorted by decreasing frequency
3. Concatenation of words into a priming text, possibly with a space character after each word

Lossy filtering:

- Removal of comments
- Removal of extraneous whitespace between tags
- Case-folding of tags to lower case

Size: 1 kB or 10 kB after GZIP

Method 1: Use of common priming text (continued)




```
HREF</HTML></A><TITLE></TITLE><P><HTML>http</BODY>
wwwThis<BR><HR></H1></UL><UL><LI></EM><EM>generate
d<HEAD></HEAD>that2000footerEST<STRONG></STRONG>NA
MESubject</SMALL><SMALL>sortedDateemailhypermailDT
```

Method 2: Exploiting document linkages

Use previously visited HTML file

Essentially, this context file is transmitted for “free”

But, may need to retain state information

Test website

Snapshot of our department's website (http://www.cs.mu.oz.au/) on 24 July 2000

- 68,361 HTML files totalling 721.0 MB → average is about 10.8 kB per HTML file
- 14,168 HTML files of them (totalling 147.2 MB) were accessed that week → 20.7% of the total files were retrieved

Test website (continued)

One week's worth of web server logs with internal requests removed

Filtered so that requests grouped by IP address

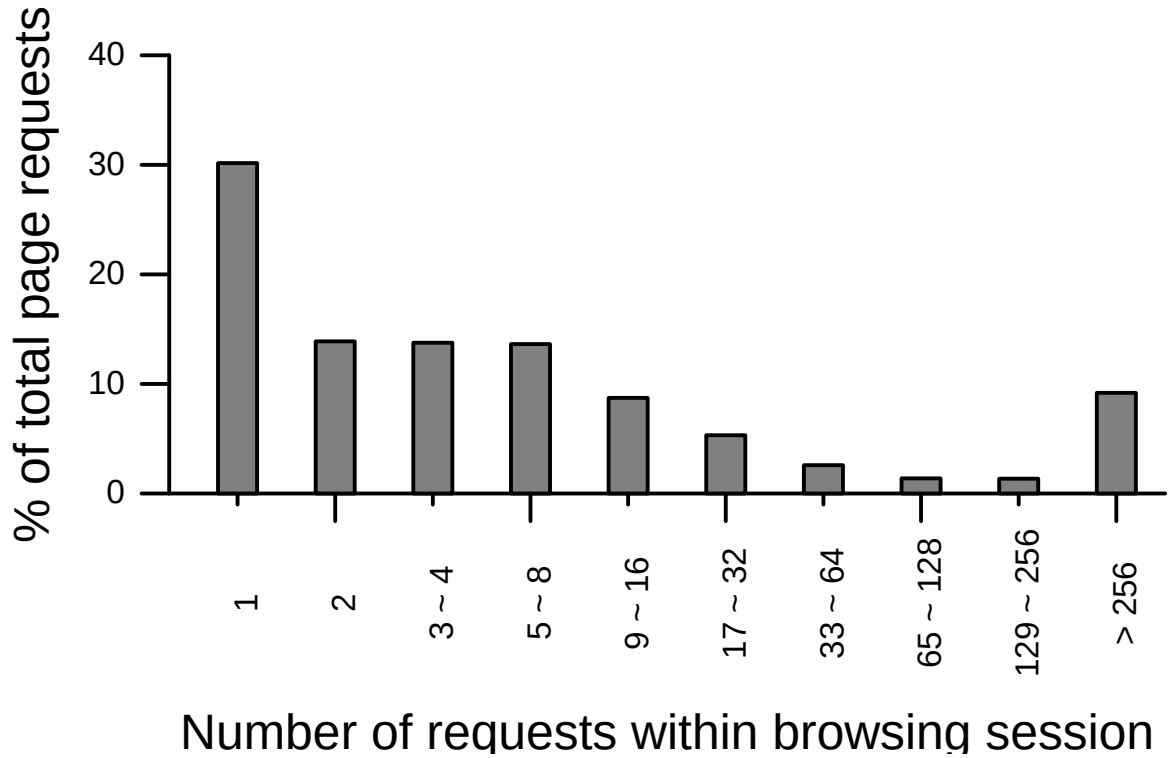
Define a **browsing session** as:

> *A series of file accesses by a particular IP address such that no more than 5 minutes have lapsed between any two consecutive page accesses*

21,067 browsing sessions and 48,343 total HTML page requests → average just over 2 HTML pages per session

443.1 MB of HTML files transferred that week

Test website (continued)



Simulation Results

Two types of measurements performed, expressed as percentages:

- Diskspace - Total diskspace required to keep the original HTML file and the compressed version of it, if it was accessed at least once
- Network bandwidth - Percentage of bytes transmitted when compared to using no compression, summed over all accesses in the log period

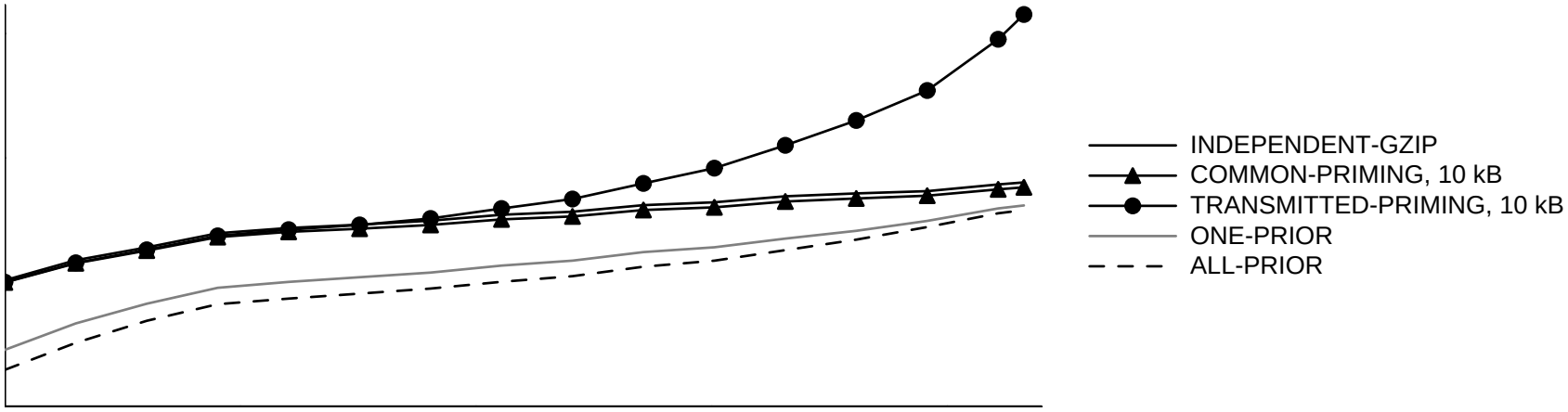
Diskspace is increased when a file is accessed the first time, while the network bandwidth is affected each time a file is transferred

Simulation Results (continued)

Method	Context Size	Common / Transmitted	Spaces	Prior Pages	Disk space (%)	Network bandwidth (%)
None	–	–	–	–	100.0 (119.4)	100.0 (95.0)
GZIP	–	–	–	–	104.7 (104.4)	27.9 (26.5)
GZIP	1 kB	Common	–	–	104.5 (104.2)	27.1 (25.7)
GZIP	1 kB	Transmitted	–	–	104.5 (104.2)	31.5 (30.1)
GZIP	–	–	–	one	109.0 (108.6)	24.5 (23.5)
GZIP	–	–	–	all	–	23.9 (22.9)
GZIP	10 kB	Common	X	all	–	23.1 (22.1)

Note: Percentages in parentheses indicates the effect of lossy filtering

Simulation Results (continued)

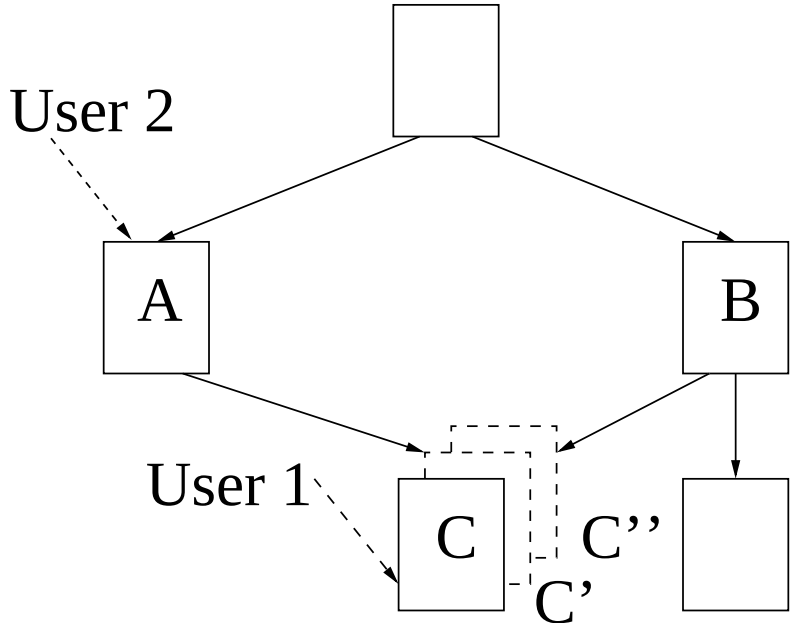


decreasing browsing
session length

Possible implementation

Use HTTP's Referer field

Rename links in a file so that the files it links to are ones compressed using it as a context



Performed in real-time with each request or pre-calculated when the web server is idle

Bookmarks would require special treatment

Related work

Measured bandwidth savings when files compressed individually and if tags case-folded

Use of links to improve document ranking by search engines

Delta encoding

Pre-fetching

Conclusion

Compression of short HTML files can be improved by combining:

- Existing algorithms like GZIP
- Priming texts or previously visited files

Tradeoffs between storage, bandwidth, and processor time