Effective cum Efficient Bandwidth by LZW Compression Technique

[1]J.JELSTEEN
Assistant Professor,
Nehru Group of Institutions, Coimbatore.
Email : jelsteen@hotmail.om

[2]B. Rajendran
Lecturer in Computing, Institute of Business Studies (Enga Province),
National Capical District, Papua New Guinea.
Email : rajendran_b@hotmail.com

[3]J.ALICE PUSHPARANI
Assistant Professor, Nehru Institute of Technology, Coimbatore.
Email : ranialicejels@gmail.com

Abstract

The Internet and World Wide Web (WWW) have made electronic commerce a reality. According to an estimate the WWW presently contains more than 800 millions pages, encompassing about 6 terabytes of text data on about 3 million servers. As the number of web users is increasing rapidly, there is a need to build fast and efficient systems for handling users requests on the web. Jakob Nielsen's figures from his keynote talk at the QualityWeek 1999 Conference & Exhibition suggest that the webpage should be received within 7 seconds after the request from the user. The real answer to "too slow" probably is more subjective, but at the same time probably not too terribly different. The WebSite sever is configured properly and effectively when it has "enough capacity to meet the customer demands."

Estimating the actual capacity of a WebSite server requires a variety of different types of analyses. There are many deep, highly technical methods for calculating the capacity of a server to meet imposed load. However, a more practical way is to simply measure the perceived response time accurately, and scale up server capacity to make sure that maximum real-user experiences don't exceed unacceptable delays.

Keywords:LZW, LZW Compression, Bandwidth

Need for Compression

The need to improve effective bandwidth is motivated by the explosive growth of users and service providers using the web, which is currently severely straining the available bandwidth. Whilst in the past most net traffic has been text based, with file sizes measured in hundreds of kilobytes, the use of the html protocol to serve multimedia documents has increased file sizes to megabytes. This encroachment of bandwidth results in long delays in accessing documents, limits information availability to less users in a given time, and negatively affects net traffic using other protocols such as mail and news.

Performance

Compression shrinks computer data, making it cheaper and faster to send across the Internet. To the service provider, data compression provides three primary benefits, which in turn, are passed on to the user:

- Compressed packets consume less bandwidth.
- Compression reduces the latency of packets as they traverse the network, since packet length is shorter.
- Performance is significantly enhanced.

It is desirable to have faster transfer rates at lower costs in communication. Data compression addresses these demands by reducing the amount of data that must be transferred over a media with a fixed bandwidth, thus reducing the connection time. Data compression also reduces the media bandwidth required to transfer a fixed amount of data with a fixed quality of service, thus reducing the tariff on this service.

Website performance can be improved by using compression techniques such as LZW**.** The respective webpages are compressed and uploaded to the web server. Decompression takes place at the client place which is along with the compressed webpage. An analysis was also carried out to find the best compression techniques.

Introduction to Web Compression

The information highway doesn't exist. That may come as a surprise to people who have heard everything from a long-distance telephone network to the internet described as a data "superhighway".

The internet is the precursor of the ultimate global network. There is little doubt that when the global interactive network has finally evolved into the highway, it will still be called the Internet. But as quaintas the term “information highway” is beginning to sound, using it appropriately helps to draw a distinction between yesterday’s primarily narrowband interactive network(the current “Internet”) and today’s broadband interactive network (the “highway”). Whatever it ends up being called, constructing an interactive network of enormous capacity (“bandwidth”) is a big job. The need to improve effective bandwidth is motivated by the explosive growth of users and service providers using the web, which is currently severely straining the available bandwidth.

Whilst in the past most net traffic has been text based, with file sizes measured in hundreds of kilobytes, the use of the html protocol to serve multimedia documents has increased file sizes to megabytes. This encroachment of bandwidth results in long delays in accessing documents, limits information availability to less users in a given time, and negatively affects net traffic using other protocols such as mail and news.

There is no need to examine the role of motion picture/video compression standards such as MPEG, nor that of textual compressors, but acknowledge their importance in achieving the overall goal of improving network throughput.

In November 1999, the total amount of net traffic on NSFNET alone was 97,781 billion bytes. The traffic due to WWW was 3,126 billion bytes (second largest, and 14% of total traffic). FTP ranked first, and news and mail third and fourth respectively . This compares to 13,602,330 bytes (580th largest and 0.0004% of total traffic) for the WWW in June 2000 . This incredible increase is continuing at almost the same hyperexponential rate.

Web compression dramatically reduces bandwidth, in some cases cutting it by more than half. Speeds the pages to the end-user because packets are needed.

Problem Definition

The exponential rate of growth of the World Wide Web has led to an increase in Internet traffic, as well as a serious degradation in user-perceived latency while accessing “Web pages”. One way to reduce the latency is using compression. By compression it can be proposed that the maximum hit rate achievable by any compression method is just 20% to 50%.

The Bandwidth of a WebPages depends on the size of the web page. By compressing the WebPages the Bandwidth rate is either increased or decreased. The compression techniques are LZW. WebPages of different websites are downloaded and compressed. Those compressed WebPages are uploaded and bandwidth is calculated.

Calculation of Download rate:

Creating 5 threads and 50 total hits for all the five threads. CWinInet_URL class module, this is a wrapper class that encapsulates calls of several WinInet API finctions for easy handling of URLs. CWinInet_URL allows you to create, crack, encode and decode URLs. The connection time taken for the client to the server, the sending time of request from the client, the time taken for the first byte to receive and the time taken for the last byte received in the client’s browser are calculated. The difference between the time taken for the first byte and last byte against each thread is calculated using the formula

$$\text{Bytes received per milli second For each thread} = \frac{\text{Total bytes received}}{\text{(Time taken for the last byte – Time taken for the First byte)}}$$

$$\text{Bytes received per milli second} = \frac{\text{Total of bytes received per second different threads}}{\text{Total no of different threads created to hit the given page}}$$

Bytes received per millisecond for each compression techniques are calculated.

It is to be proved that download rate decreases through compression techniques (depends on the configuration of client machine) and compression also reduces the media bandwidth required to transfer a fixed amount of data with a fixed quality of service, thus reducing the tariff on this service.

Approach to the problem

The issues of measuring WebSite Performance are many and complex, it becomes clear is that users' perception of effective, useful response time is the dominant factor in assuring adequate capacity. This can be accomplished by realistic browser-based experiments that measure aggregate response time as a function of gradually increased parallelism.

WebSites are something entirely new in the world of software quality within minutes of going live, a WWW application can have many thousands more users than a conventional, non-WWW application. The immediacy of the WWW creates an immediate expectation of quality and rapid application delivery, but the technical complexities of a WebSite and variances in the browser make testing and quality control more difficult, and in some ways, more subtle.

Website architecture depends on Browser, HTML, Navigation, Server response and concurrent users. When data compression is used in conjunction with algorithms such as LZW - does modest compression which result in significant performance gains.

LZW (Lempel,Ziv,Welch) Technique

In LZW compression, the repetitive characters must be within about 4,000 characters (4096 bytes) of each other. The strength of LZW is that it produces an optimum combination of compression and performance. LZW achieves "lossless compression" which simply means no data is lost during compression and decompression. Lossless compression reduces data typically by about 1/2 - but nothing is ever lost. With LZW customers achieve acceleration without deterioration

Compression Algorithm

```
Step 1:   set w=1
Step 2:   Loop
              Read a character k
              If  wk exist  in the dictionary
                    w=wk
             else
                       output the code for w
                  add wk to the dictionary
                  w=k
            endif
          End loop
Step 3:   Stop
```

Let us consider a string

^WED^WE^WEE^WEB^WET

W	K	Output	Index	Symbol
Nil	^			
^	W	^	2	^W
W	E	W	3	WE
E	D	E	4	ED
D	^	D	5	D^
^	W			
^W	E	^W (2)	6	^WE
E	^	E	7	E^
^	W			
W	E			
^WE	E	^WE (6)	8	^WEE
E	^			
E^	W	E^ (7)	9	E^W
W	E			
WE	B	WE(3)	10	WEB
B	^	B	11	B^
^	W			
^W	E			
^WE	T	^WE(6)	12	^WET
T	EOF	T		

The compressed string is ^WED2E673B6T

Decompression Algorithm

(Decompression will take place in the client machine)
Step1 : Read a fixed length token k
(code or char)
step 2 : output k
step 3 : w = k
Step4: loop
read a fixed length code k
entry = dictionary entry k
output entry
add w + first char of entry to the dictionary
w = entry
endloop
Step 5: Stop
Uncompress the string ^WED2E673B6T

W	K	Output	Index	Symbol
Nil	^			
^	W	W	2	^W
W	E	E	3	WE
E	D	D	4	ED
D	2	^W	5	D^
2(^W)	E	E	6	^WE
E	6	^WE	7	E^
6 (^WE)	7	E^	8	^WEE
7 (E^)	3	WE	9	E^W
3(WE)	B	B	10	WEB
B	6	^WE	11	B^
6(^WE)	T	T	12	^WET
T	Eof			

The string ^WED^WE^WEE^WEB^WET
Result :

Comparison on the size of web pages for different compression Techniques

Five WebPages of different WebSites are taken. Size of Original WebPages is given. Such as covai.com, aesfirst.com, Coimbatore.com, madurai.com and yahoo.com
Webpage Size in Bytes

	Original	LZW
Covai	32790	19619
Madurai	14093	10195
Coimbatore	22811	15725
Aesfirst	13188	11061
Yahoo	53930	32699

Comparison on the Bandwidth of web page for different compression techniques

Detail report on Download Rate Analysis of WebPages for five WebSites
Bandwidth Analysis in Bytes

	Original	LZW
Covai	1639500	980950
Madurai	704650	509750
Coimbatore	1140550	786250
Aesfirst	659400	553050
Yahoo	2696500	1634950

Comparison on the download rate of webpages fordifferent compression techniques

Download time in seconds for each Webpage of five WebSites are given as follows

	Original	LZW
Covai	202	113
Madurai	202	78
Coimbatore	195	144
Aesfirst	86	79
Yahoo	623	393

Conclusion:

I have done this research on LZW technique, Simple Fast and Dictionary based Method. The paper is confined to LZW technique on Web Compression. I found that it is efficient in relation to storage in the web server and download time of a webpage decreased.

References

[1] R. Agarawal, A. Arning, T.Bollinger, M. Metha, J. Shafer, R. Srikant: “The Quest Data Mining System”, Proc. Of the 2nd Int’l Conference on Knowledge Discovery in Databases and Data Mining , Portland, Oregon, August, 1996.

[2] Eui-Hong (sam) Han, Anurag Srivastava and Bipin Kumar: “Parallel Formulations of Inductive Classfication Learning Algorithm” (1996).

[3] Jaiwei Han & Miceline Kamber (2001), “Data Mining: Concepts and techniques”.

[4] IBM QUEST Data Mining Project http:/ /www.almaden.ibm.com/cs/quest/ index.html

[5] Data Mining at Dun & Bradstreet http:/ /www.santafe.edu/~kurt/text/ wp9501/wp9501.shtml