Web cache memory compression for optimizing the performance in web browsers

G. ShanmugaSundaram
Associate Professor: Department of Information Technology
Sri Manakula Vinayagar Engineering College
(Affiliated to Pondicherry University)
Puducherry, India.

S. Aswini
PG Scholar: Dept. of Computer Science & Engineering
Sri Manakula Vinayagar Engineering College
(Affiliated to Pondicherry University)
Puducherry, India.

Abstract— **Web browsers are used on daily basis, where web caching plays an important role in saving the web pages and thereby, it helps the users in re-launching the web pages faster when they use it again. The study is done under the concept of compression in the web, which is highly focused where different kinds of compression techniques with detailed background are studied. While studying the existing works from previous researchers, the problem issues are analyzed. Other problem issues related to this work are discussed and it will be taken further in the future works.**

Keywords— web performance, caching, data compression, cache memory, web browsers

I. INTRODUCTION

Web caching is termed as the storage of web objects, which helps the users to access same links faster if accessed again [11]. The web objects are the web pages consisting of texts, images and videos. The web cache is mainly used for reducing latency and traffic [10]. There are four types of web caching namely browser cache, proxy cache, transparent proxy cache and reverse proxy cache. The browser cache [12] is the storing of objects based on session, where the web pages visited gets instantaneously saved in cache which helps the user to load faster if the user tries to load the same web page again. [3] The proxy cache is found between the origin web server and client, which evaluates the HTTP headers along with inter cache communication protocols such as ICP (Internet Cache Protocol), WCCP (Web Cache Control Protocol), HTCP (HyperText Caching Protocol) and CARP (Cache Array Routing Protocol). The transparent proxy cache is termed as the cache used in enabling the content gateway to respond to internet requests, where it redirects the traffic flow into the proxy cache [22]. The reverse proxy cache is termed as the load balancer between the multiple web servers. The reverse proxy caching can also be used in the modification of HTTP requests that can be helpful in redirects, rewrites and improving caching ability. [9]

Compression is termed as the encoding of information. Compression can be done in Text, images, audio and video. Data compression is termed as the reduction of the size of data. The content specific optimization is one of the technique used in reducing the size of delivered resources. The content can be cached in three formats such as compressed, uncompressed, and incompressible.

Lossless Compression is termed as reconstruction of the original message from the compressed message.[2] Lossy Compression is termed as reconstruction of the approximate message from compression, which can't reproduce the exact original message like in lossless compression. The lossless compression is highly preferred because it can exactly produce the original contents, whereas in lossy compression, it could possibly result in losing some contents or pixels. [17]

There are different types of lossless compression algorithms namely LZ77/78 (Lempel Ziv 77/78), LZSS (Lempel Ziv Storer Szymanski), LZMA (Lempel Ziv Markov chain Algorithm) [2]. Both statistical coders and dictionary coders [1] are required, because the statistical coders provide better compression, whereas the dictionary coders provides in enhancing performance (i.e.) runs faster. The Prediction by Partial Matching (PPM) is an example of statistical coders [24], whereas Lempel Ziv (L-Z) methods are the examples of dictionary coders. [23]

Maintaining higher compression performed in higher speed is one of the challenging task. If the task is accomplished in pushing and embedding it to web browsers, then there will be 3x the network performance, which will enhance or internet speed very well, where less memory will cached temporary in physical memory, where it will also enhance the overall performance of our systems too. [8]

HTTP is expanded as HyperText Transfer Protocol, which is termed as the stateless application level protocol that provides data transmission through web such as text, images, audio and video. The transfer of data (text, images, audio and video) is done in the form of compressing them through content encoding and content decoding, where the actual transmission is done in the form of digital signals through signal processing. The digital signals transmission are performed through delta coding (encoding/decoding). The HTTP compression is done through HTTP request generation and HTTP response generation, where the transmission of data occurs. [7, 16, 28]

The rest of the paper is organized as it follows:- Section II explains the background of this survey, where it focusses on the algorithmic techniques used previously along with drawbacks. Section III discusses on the related works done on techniques which was discussed in section II, where the works will be briefly explained along with the limitations. Section IV discusses further on the issues in detail. Section V concludes the paper.

II. BACKGROUND

(a) Aim of the study:

- To enhance the performance of web browsers by compressing the web cache memory, which optimizes thereby reducing the traffic and latency.
- To study about the lossless compression algorithms relevant to web caching and memory compression.

(b) Algorithm techniques:

(i) LZ77:-

It is a sliding window technique where the dictionary consists of a set of fixed length phrases from a window into the previously processed text. This algorithm uses "lookahead buffer" and "search buffer". The lookahead buffer is filled with the first symbols of the data that has to be encoded, whereas the search buffer is filled with pre-defined symbols of the input alphabet. We dig the code starting from the search buffer and then to the lookahead buffer. [14]

Since LZ77 faced the drawbacks in strings replacement, So LZSS had to grab the place of LZ77.

(ii) Lempel Ziv Storer Szymanski (LZSS):-

LZSS (Lempel Ziv Storer Szymanski) is termed as "a dictionary encoding technique that attempts to replace a string of symbols with reference to a dictionary location of the same string". [17]

Comparison of LZSS over LZ77:-

- In LZ77, the phrases in the text window were stored as a single contiguous block of text. LZSS stores text in contiguous manner, but it creates an additional data structure that improves on the organization of the phrases.
- At each phrases passes out to lookahead buffer and into encoded portion of the text windows in LZ77. The same followed in LZSS, but it adds the phrase to a tree structure which is a binary search tree. The savings are created by using the tree makes the compression efficient as well as encourages experiments with longer window sizes.
- LZ77 output tokens consist of a phrase offset, a match length and a character that followed the phrase. LZSS instead allows pointers and characters to be freely inter-mixed.
- Under LZ77, the encoder output a dummy match position with a length of zero for every symbol. LZSS instead uses a single bit as a prefix to every output to indicate whether it is an offset/length pair or a single symbol for the output [17].

There were also some drawbacks noticed in LZSS algorithm:-

- LZSS compresses poorly when it starts since it does not have any data in the text window for matches.
- LZSS faces trouble with preloading, where it has no idea what type of data will come in the input stream.
- The compression speed of LZSS may suffer as the window grows due to the extra work required to navigate and maintain a larger binary tree.
- The compression of smaller files in LZSS will suffer due to increased time required to build a full dictionary.
- Finally, the LZSS faces inefficiency in handling the duplicate strings. It’s possible to free up the space used by these duplicate strings in the text window allowing for expansion of the dictionary. However, the side effect is the decompression program has to keep track of the duplicate strings, which will result in significant cutback in expansion speed [17].

(iii) LZ78:-

Instead of fixed length phrases from a window into the text, it just builds up phrases into one symbol at a time by adding a new symbol to an existing phrase when a match occurs [6, 13]. There were two drawbacks [24] analyzed in LZ78 algorithm:

- While parsing the input string, the related information crossing the phrase boundaries is lost. In many situations, there would be significant number of patterns crossing phrase boundaries and these patterns would probably affect the next symbol in that sequence.
- The convergence rate of the LZ78 to the optimal predictability is slow.

(iv) Lempel Ziv Markov chain Algorithm (LZMA):-

The Lempel Ziv Markov chain Algorithm (LZMA) is used in the performance of lossless data compression, where lossless data compression is termed as the reduction of size restoring the original one without the loss of data. [20] The algorithm is similar and more efficient than the existing LZ77 algorithm [14]. The idea of LZMA comes from the dictionary based schemes (LZ77/LZ78) and markov models where it uses range encoder [26]. LZMA was derived from LZ77 and LZ78, which is combination of dictionary based scheme and Markov chains. LZMA uses range encoding. [5] In other algorithms, Huffman encoding is used. So, what is difference between the range encoding and Huffman encoding? In range encoding, all symbols in a message gets encoded into one number. But in Huffman encoding, each symbol is assigned with a bit pattern and then, all the bit patterns are linked together. Hence, higher compression is seen in Range Encoding.

Markov chain is termed as the mathematical system which undergoes transitions from one state to another on a finite state space. The Markov chain can be represented in probabilistic state diagrams where the transition between the finite states are being labelled with the probabilities of occurrences [15]. Where Markov chain is used in LZMA? It is used in parsing of source data, Encoding and in Finite State Compressor (Finite state machine). In what ways the Markov chains related with LZMA?? 1. Variable-to-Variable length codes are used in which the number of source symbols are encoded and number of bits encoded as per the codeword [18, 20]. If the future behavior of a process depends only on the present state, but not on the past, then the process is termed as the markov process [15]. Let X (t) = number of bytes transmitted in time t, so that the sequence [X (t), t € [0, ∞]] forms a pure web transmission process, which helps in determining the web performance.

The Markov chain is not that efficient when compared to Markov Chain Monte Carlo (MCMC) method and Hidden Markov method. Now, this needs to be improved.

The range encoding is termed as one of the entropy coding method where it produces a space efficient stream of bits for representing a stream of symbols and their probabilities [3]. The range decoding reverses the encoding process. The range coding is also better than the arithmetic coding [21] since it performs fast arithmetic coding. Range coding performs faster arithmetic coding. It renormalizes one byte at a time, rather than one bit at a time. Thus, it runs 2x faster. It uses bytes as encoding digits instead of bits.

LZMA has issues in both statistical coding as well as dictionary coding where the performance is minimal [5].

III. RELATED WORKS

In this section the recent works which uses the lossless compression algorithm are presented.

According to Christopher Hoobin et el.,[1]: The LZ77 is adopted with reference to [13], where it relates to Gzip mechanism. The Gzip works under Deflate algorithm and works under Huffman encoding and decoding. The range encoding/decoding is better than Huffman encoding/decoding since it works under fast arithmetic coding. Hence, thereby it requires improvement.

According to Parekar et el., [2]: The lossless data compression algorithms are addressed such as RLE (Run Length Encoding) algorithm, LZ77 [14], LZW and LZMA. The concept of LZMA does not satisfy speed in terms of compression.

According to Seema et el.,[3]:The web prefetching mechanism was addressed for improving cache size, which will gradually improve the performance. Yet, [3] failed to address the data compression, which could have been one of the best measure for enhancing web performance.

According to M.Sudha et el.,[4]: The LZMA mechanism is used in successfully accomplishing the compression with reference to [17]. But, [4] failed to show the performance values and also failed to prove it is better than others through comparison. The LZMA has drawbacks in sliding window dictionary which is possibility of overhead when heavy process is performed. So, it requires improvements upon all basis.

According to Karthik GopalRatnam et el.,[25]: An optimized LZ77 tool was done with reference to [14]. But, experiencing some drawbacks in LZ77, then [25] proceeded to improvise it using LZMA mechanism tool [26] including block sorting data compression [27], where Static HTML Transform (SHT) and Semi-Dynamic HTML Transform (SDHT) was proposed. Yet, it failed to satisfy the compression in CSS page bloats which includes various fonts and styles, which led those CSS pages getting uncompressed instead of compressing them.

According to Fairuz S.Mahad et el.,[9]: The attempt to improve web server performance inorder to reduce the overloading of network traffic and congestion. But, it failed in terms of memory compression.

IV. FURTHER DISCUSSIONS

From the background and related works, a lot of problem issues are being analyzed in detail. Coming from section III, the related works were discussed regarding the recent works as well as the limitations faced in [1], [2], [3], [4], [17], [25], [14], [26],[27] and [9]. The problem issues regarding the compression were discussed towards the performance improvement by means of lossless algorithms [2]. The lossless algorithms such as LZ77/78 [13, 14] were discussed, where it led to the birth DEFLATE mechanism [28, 29] where GZip and ZLib [30] were developed. Also, LZMA [2, 4, 5, 14, 20, 26] was being discussed which is a part of lossless algorithm. The difference between Deflate and LZMA is that LZMA uses faster arithmetic coding known as range coding [3, 21], which is better than the Huffman coding used in Deflate. Both LZMA and Deflate (GZip) along with block sorting [27] was used in [25]. Yet, it failed to defeat the CSS page bloats, which will downgrade the web optimization level especially while loading the websites and during transferring files, the uncompressed data will also be added due to the bloats. In [9], the two-tier cache where it used SQUID cache, which is good to improve performance in web servers. But, [9] failed to focus more on cache storage, where heavy logs might lead to overhead since it hasn't followed memory compression by compressing the cache.

The product-based web browsers like Google Chrome, Mozilla Firefox, Opera, Internet Explorer and Safari also undergoes some flaws. While going through depth study on

Google Chrome browser, it is found that the Google Chrome swaps different processes which is advantageous for users to avoid crash of whole browser and gets only that specific tab crashed since every tab undergoes one process. But, it has huge drawback in excess virtual memory utilization which might be possibly lower system performance if more tabs are opened and battery level is easily decreased due to processes usage, this is little bit similar (not exactly) to thrashing concept seen in operating systems.

In Mozilla firefox, cache memory management issue is there, where the browser leads to more cache memory usage while compared to other web browsers.

V. CONCLUSION

From this research survey, the web caching, compression and algorithms such as LZ77, LZ78 and LZMA are addressed along with the issues and analysis being discussed in the aspect of performance, where the solutions will be used as a part in upcoming future works.

REFERENCES

[1] Christopher Hoobin, Simon J. Puglisi, Justin Zobel, "Relative Lempel Ziv Factorization for efficient storage and retrieval of web collections", 38th International Conference on Very Large Databases, Proceedings of the VLDB Endowment, Vol.5, No.11, 2011.

[2] Parekar P.M., Thakare S.S, "Lossless Data Compression – A Review", International Journal of Computer Science and Information Technologies, vol.5, no.1, 2014.

[3] Seema, Priyanka Makkar, "An approach to improve the web performance by prefetching the frequently access pages," International Journal of Advanced Research in Computer Engineering and Technology, vol.1, no.4, June 2012.

[4] M.Sudha, Dr.S.Palani, "A novel implementation of wavelet transform and Lzma for compression and decompression of document images", International Journal of Innovations in Engineering and Technology (IJIET), Vol.2, Issue 3, June 2013.

[5] Paolo Ferragina, Giovanni Manzini, "On compressing the textual web", ACM, February 2010.

[6] E.Jebamalar Leavline, D.Asir Anthony Gnana Singh, "Hardware implementation of LZMA data compression algorithm", International Journal of Applied Information Systems, Vol.5, No.4, March 2013.

[7] Khushali Tirghoda, "The affection of HTTP compression in the performance of core web components", International Journal of Scientific & Engineering Research, Vol.3, Issue 5, May 2012.

[8] Eric Gavaletz, Dominic Hamon, Jasleen Kaur, "In-Browser Network Performance Measurement," IEEE transactions on Parallel and Distributed Computing and Systems, November 2009.

[9] Fairuz S.Mahad,Wan M.N.Wan-Kadir,"Improving web server performance using two-tier web caching," Journal of Theoretical and Applied Information Technology, vol.52, no.3, June 2013.

[10] Waleed Ali,Siti Mariyam Shamsuddin,"Integration of Least Recently Used Algorithm and Neuro Fuzzy System into client side web caching," International Journal of Computer Science and Security, vol.3, no.1, 2009.

[11] Jim Challenger, Arun Iyyengar, Paul Dantiz,"A scalable system for consistently caching dynamic web data," IEEE Computer and Communication Societies, vol.1, March 21-25 1999.

[12] K.Muralidhar,Dr.N.Geethanjali,"Improving the performance of the browsers using fuzzy logic," International Journal of Engineering Research and Technology, vol.3, no.1, August 2012.

[13] Jacob Ziv, Abraham Lempel, "Compression of Individual Sequences via variable rate coding", IEEE transactions on Information Theory, Vol.No.5, September 1978.

[14] Jacob Ziv, Abraham Lempel, "A universal algorithm for sequential data compression", IEEE transactions on Information Theory, 1977.

[15] Probability, Statistics and random processes (3rd Ed – T. Veerarajan).

[16] Jon Butler, Wei-Hsin Lee, Bryan McQuade, Kenneth Mixter, "A proposal for shared dictionary compression over HTTP", Google, Inc. September 8, 2008.

[17] The data compression book (Second Edition – Mark Nelson and Jean Loup Gail.

[18] Francisco Claude, Antonio Farina, Miguel A. Martinez-Prieto, "Indexes for highly repetitive document collections", ACM transactions, October 2011.

[19] Fundamentals of computer algorithms (Second Edition – Ellis Horowitz, Sartaj Sahni, Sanguthevar Rajasekaran)

[20] Farina.A, Navarro.G, Parama, "Word-based statistical compressors as natural language compression boosters", IEEE transactions on data compression, March 2008.

[21] Glen G.Langdon, "An introduction to arithmetic coding", International Business Machines Corporation (IBM) transactions, Vol.28, No.2, 1984.

[22] J.B.Patil,B.V.Pawar, "Integrating Intelligent Predictive Caching and static prefetching in web proxy servers," International Journal of Computer Science and Engineering, vol. 3, no.2, February 2011.

[23] Alistair Moffat, "Implementing the PPM data compression scheme", IEEE transactions on Communication, Vol.38, No.11, November 1990.

[24] Przemysław Skibiński, "Improving HTML compression", IEEE data compression conference, 2008.

[25] Karthik GopalRatnam, Daniel J.Cook, "Active Lezi: An incremental parsin algorithm for sequential prediction", International Journal on Artificial Intelligence Tools, 2004.

[26] Igor Pavlov, "7-zip compression utility",http://www.7-zip.org, 2009.

[27] Burrows, M., Wheeler, D. J, "A block-sorting data compression algorithm", SRC Research Report 124. Digital Equipment Corporation, 1994.

[28] Deflate compression algorithm, http://www.gzip.org/algorithm.txt.

[29] P.Deutsch, "RFC 1951: Deflate compressed data format specification version 1.3", May 1996.

[30] P. Deutsch, J.L.Gailly, "RFC 1950: ZLib compressed data format specification version 3.3", May 1996.

[31] R. Fielding, H. Frystyk, Tim Berners-Lee, J. Gettys, J. C. Mogul, P. Leach, L. Masinter, "RFC2616 – HyperText Transfer Protocol (HTTP/1.1)", June 1999.