Indian Journal of Science and Technology, *Vol 11(45), DOI: 10.17485/ijst/2018/v11i45/117921, December 2018*
ISSN (Print): 0974-6846
ISSN (Online): 0974-5645

Compression Algorithms: Brotli, Gzip and Zopfli Perspective

Zubair Ali Syed[1*] and Tariq Rahim Soomro[2]

[1]Department of Computer Science, SZABIST, Dubai Campus, United Arab Emirates; s.zubair.ali@gmail.com
[2]College of Computer Science & Information Systems, IoBM, Karachi, Sindh, Pakistan; tariq.soomro@iobm.edu.pk

Abstract

Objectives: Websites response time is the most critical in today's arena. This study contributes to various compression algorithms, which enable Web Servers to accelerate Websites responses to meet the need of Clients. **Methods/Statistical Analysis**: Different techniques are used to improve the browsing experience of the Clients and acceleration of Websites responses. The prominent techniques in practice are Web contents compression to decrease the HTTP response. One of the popular and effective compression file formats used over the Internet is Gzip. Team Google come up with two algorithms, known as Zopfli and Brotli, which are probably potential replacement of Gzip, because of their performance. **Findings:** This study provides an overview about these algorithms and then compares them with different types of files by mimicking the http compression using fiddler tool. It also highlights the short coming of Brotli that should be overcome in near future as to become a major success. Brotli distinguished itself in the comparison with its counterparts and is the future technology and probably will replace Gzip. **Application/Improvements:** Brotli enabled browsers are the need of the day. Also Brotli version for mobile browser will ease the mobile users.

Keywords: Brotli, Compression, Decompression, Gzip, Zopfli

1. Introduction

The Web content compression is the most important technique used to accelerate the Websites response. Web servers and web clients uses the HTTP compression techniques to improve the transfer speed and utilization of bandwidth. It reduces the response times of web content by reducing the size of the HTTP response. However, nothing is free in this world, so as compression; it is the heaviest operation performed by Server. To get the better compression rate research oriented efforts are required.

The most popular and effective compression format on the Web is considered to be Gzip. Recently, team Google introduces Brotli – a new compression algorithm; which is currently talk of the world and believed to be a potential replacement of Gzip[1]. This paper is organized as follows; Section 1 will be compression of these algorithms, namely Gzip, Zopfli and Brotli; Section 2 will talk about methodology in the form of material and methods used; Section 3 will depict the results and finding by testing of different types of Web content (html, JavaScript, Apk, css files) using fiddler tool and compares the results of compression size and the time taken; finally, discussion about Brotli algorithm and future work will be discussed.

2. Compression Algorithms in Comparison

2.1 Gzip

Jean-loup Gailly wrote Gzip algorithm for the GNU project with Mark Adler wrote the decompression part of it. Deflate compression algorithm is used in Gzip, which

*Author for correspondence

is a combination of the LZ77 and Huffman coding[2]. Zlib library is the most common implementation of Gzip compression[3]. It has 9 preset quality settings labeled from 1 to 9, which can be divided into two groups:

1. (Levels 1–3) Fast compression but less compression ratio
2. (Levels 4–9) Slow compression but good compression ratio

With Zopfli library, Gzip gets better compression ratio, but with significantly slower compression speed as compared to Zlib.

2.2 Zopfli

Two Google employees wrote a reference implementation for Zopfli and it was released on 25/4/2013[4]. The algorithm encodes data into different formats, such as deflate, Gzip and Zlib. The name Zopfli comes from Swiss German dialect and means as special types of bread. It is considered as efficient deflate encoder in terms of size.

In typically situations, Zopfli produces smaller results than zlib's maximum compression, which is estimated at 3 to 8% smaller, but takes around 80% longer times than zlib. The decompressing speed for both algorithms is almost the same. Zopfli is best suited for one time static content compression due to its slower compression speed[5].

Various Zopfli implementations are available; as programming library in C language by Google, a PHP wrapper php, Zopfli[6] and a C# based library is also available by CompressSharper[7].

2.3 Brotli

Brotli is an open source data compression library. Brotli is developed keeping in view the, Huffman coding, modern variant of algorithm LZ77 and 2nd order context modeling. Here context modeling permits multiple Huffman trees for the similar alphabet using similar block.

As Zopfli algorithm is deflate-compatible; Brotli on the other hand is based on new data format and get compared to Zopfli get up to 26% higher compression ratio[1].

Researchers at Google conducted a study[1] and according to this study it was shown that Brotli is roughly as fast as Zlib's deflate implementation. It not only achieved higher data density by using a 2nd order context modeling, but it is reusing of entropy codes, along with larger memory window of past data and joint distribution codes.

A pre-defined static dictionary is used by Brotli; this dictionary used 13000 plus strings to its internal state for the purpose of "warming up". This dictionary is consists of most common words, commonly used phrases and frequent substrings derived from larger corpus of text and HTML files.

3. Methodology

The methodology used in this study is of qualitative nature and following algorithms were tested on a common JavaScript file, CSS file, News Website file and APK file:

1. GZIP
2. GZIP (Zopfli)
3. Deflate
4. Deflate (Zopfli)
5. Brotli

In this study fiddler version 2 is used to mimic HTTP compression. Here the Transformer Inspector permits to add or remove HTTP-based encodings from its response. Tests were conducted using different HTTP compression algorithms (GZIP, GZIP with Zopfli, Deflate and Brotli)[8] when applied to different content formats. The PC used to test the compression algorithms uses operating system as Windows 8.1 Pro 64 bit, Intel Core i7 (4 CPUs) with 8 GB RAM. For testing an executable for windows users have been used; and use release (v0.3) from GitHub for Win32 users, using VS 2015. The latest version can be downloaded from (https://bayden.com/dl/Brotli.exe) site and can be uncovered inside Fiddler[9], by placing brotli.exe inside Fiddler's\Tools\ subfolder and by restarting to see as it appear on the Transformer tab[10], as shown in Figure 1.

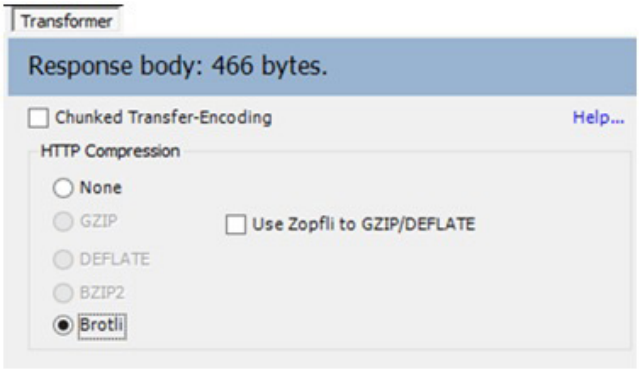

Figure 1. Transformer tab.

4. Results and Findings

Total five algorithms were tested on four common files and following are the results got after testing four different files using five different compression algorithms:

4.1 A Javascript file

A JavaScript file is examined with five algorithms, having the original file size of 195.44 KB and following are the results shown in Table 1.

Table 1. Comparison of JavaScript file

Compression algorithm	Size in KBs	% compression saved
GZIP	54.13	72.30
GZIP (Zopfli)	52.11	73.33
Deflate	54.29	72.22
Deflate (Zopfli)	52.1	73.34
Brotli	46.8	76.05

4.2 A css file

A CSS file on Gulf news site is examined with five algorithms, having the original file size of 78.11 KB and following are the results, shown in Table 2.

Table 2. Comparison of CSS file

Compression algorithm	Size in KBs	% compression saved
GZIP	53.99	30.87
GZIP (Zopfli)	53.27	31.80
Deflate	53.98	30.89
Deflate (Zopfli)	53.25	31.82
Brotli	51.92	33.53

4.3 A Response from a Site

An initial response from a news site is examined with five algorithms, having the original file size of 238 KB and following are the results, as shown in Table 3.

Table 3. Comparison of news website file

Compression algorithm	Size in KBs	% compression saved
GZIP	41.92	82.41
GZIP (Zopfli)	40.54	82.99
Deflate	42.21	82.29
Deflate (Zopfli)	40.52	82.99
Brotli	31.09	86.95

4.4 A Mobile app File

An apk file for Google play mobile app is examined with five algorithms, having the original file size of 18 MB and following are the results, shown in Table 4.

Table 4. Comparison of APK file

Compression algorithm	Size in MBs	% compression saved
GZIP	15.559	14.26
GZIP (Zopfli)	15.552	14.30
Deflate	15.563	14.24
Deflate (Zopfli)	15.552	14.30
Brotli	15.187	16.31

5. Discussion and Conclusion

In all the comparison cases Brotli has a lead with their counterparts. There are lots of areas related to Brotli that require future investigations. Major browsers should provide support for Brotli. In future these browsers' mobile version should be equipped with Brotli, this will provide mobile users definite ease. Using Brotli mobile users will enable to reduce battery use and lower data transfer fees.

Brotli's high compression ratio enables bad guys to make attacks easily; so protection against such attacks is the need of the day. Possible attacks are DoS attack and "Brotli bombing", these attacks should be addressed, so that new versions of these algorithms will be safer than the current. If a file contains all 0's, Brotli can perform 386516:1 compression ratio, this means 1389 bytes of data compressed can be uncompressed into 512 MB. Brotli is used in Web Open Font Format version 2 (WOFF2), WOFF2 is a font format that on an average provides 30% reduction in file size, which helps Web fonts load more quickly in compatible browsers. The open issues related to WOFF2 should be addressed for Brotli success in future. Is Brotli better and can fully replace Gzip in future? For static content compression definitely it is better and can replace but for the dynamic content it may or may not. On slow connections, Brotli has lead over Zlib for files larger than 64KB. Brotli is the development effort of a capable and talented team. Without any doubt the current implementation of Brotli will improve with time.

6. References

1. Introduction to Brotli: A new compression algorithm for the Internet [Internet]. [cited 2015 Npv 22]. Available from: http://google-opensource.blogspot.co.uk/2015/09/introducing-brotli-new-compression.html.
2. Gzip [Internet]. [cited 2018 Oct 19]. Available from: https://en.wikipedia.org/wiki/Gzip.
3. Zlib: A massively spiffy yet delicately unobtrusive compression library [Internet]. [cited 2015 Jan 15]. Available from: http://zlib.net/.
4. Data compression using Zopfli [Internet]. [cited 2015 Apr 22]. Available from: https://issuu.com/wnc4/docs/data_compression_using_zopfli.
5. Compress data more densely withZopfli [Internet]. [cited 2013 Feb 28]. Available from: http://googledevelopers.blogspot.ae/2013/02/compress-data-more-densely-with-zopfli.html.
6. PHP-Extension for compressing content with Zopfli a good Zlib (gzip, deflate) compatible compression which produces smallest possible results [Internet]. [cited 2015 Mar 03]. Available from: https://github.com/weltling/php_zopfli.
7. Compress Sharper [Internet]. [cited 2017 Aug 01]. Available from: https://archive.codeplex.com/?p=compresssharper.
8. Brotli [Internet]. [cited 2015 Sep 10]. Available from: http://textslashplain.com/2015/09/10/brotli/.
9. Results of experimenting with Brotli for dynamic Web content [Internet]. [cited 2015 Oct 23]. Available from: https://blog.cloudflare.com/results-experimenting-brotli/.
10. Comparison of Brotli, Deflate, Zopfli, LZMA, LZHAM, and Bzip2 Compression Algorithms [Internet]. [cited 2016 Jan 22]. Available from: https://plus.google.com/+MathieuRouvinez/posts/NMFiSGYKdft.