OPEN ACCESS
Indo-JC
Indonesian Journal of Computing
ISSN 2460-9056
socj.telkomuniversity.ac.id/indojc

Ind. Journal on Computing
Vol. 1, Issue. 1, March 2016. pp. 37-48
doi:10.21108/indojc.2016.1.1.8

Performance Analysis in Web Based Data Uploading using LZ77 Compression and Chunking Method

Hadary Mallafi [#1]
# PT Telekomunikasi Indonesia Tbk.
Bandung, Indonesia
[1] mallafi.hadary@telkom.co.id

Abstract
One of the limitations in data uploading process is the maximum request length, besides that the data size that us transferred is also an issue because it influences the data sending cost. One of the way to cope with the problem of maximum request length is by downsizing the file size (chunking). Another way to do it is by enlarging the maximum reques length. Downsizing the file size can be done by chunking the files into a smaller size or by compressing it. In this paper, the author conducted a research about the file compression process that is done in client server using the technology of AJAX and Webservice. In addition to that, the file compression is combined with file chunking. In this research, the compression method that is used is dictionary based i.e. Lempel Ziv 77(LZ77). This compression is used since it can be performed in AJAX. The analysis that is made by the researcher about the compression ratio, data sending process speed, compression time, decompression time, the compression method capability in handling the maximum request length and the combination method of compression and chunking in uploading process. The result of this research shows that compression method can handle the maximum requet length. Based on the experiment conducted, the relations between the compression ratio and window length is positively corelated. It means that the greater the window length is the more the compression ratio is. Meanwhile, the relation between window length and uploading time is negatively linearly corelated. It means that the greater the window length is the faster the uploading time is. In addition, it can also be observed that the relation between the decompression and the file size is positively linearly correlated. It means that the greater the file size is the more time needed for decompression is.

Keywords: component; Compression, AJAX, Webservices, Chunking, Lempel Ziv 77(LZ 77).

Abstrak
Salah satu batasan dalam proses uploading data adalah dengan adanya maximum request length, selain itu pula ukuran data yang dikirim menjadi sebuah perhatian karena menyangkut dari biaya pengiriman data. Salah satu cara mengatasi batasan maximum request length yaitu dengan memperkecil ukuran file, cara yang lain adalah dengan memperbesar ukuran maximum request length. Untuk memperkecil ukuran file dapat dilakukan dengan memotong-motong file menjadi ukuran yang lebih kecil ataupun mengkompresi file tersebut. Pada paper ini penulis melakukan penelitian mengenai proses kompresi file yang dilakukan secara client-server dengan menggunakan teknologi AJAX dan Webservis. Selain itu pula proses kompresi file tersebut dikombinasikan dengan metode pemotongan file(chunking). Pada penelitian kali ini metode kompresi yang digunakan berbasis kamus yaitu Lempel Ziv 77(LZ77). Metode kompresi ini digunakan karena dapat dilakukan pada AJAX. Analisis yang dilakukan oleh penulis mengenai rasio kompresi, kecepatan proses pengiriman data, waktu kompresi, waktu dekompresi, kemampuan metode kompresi dalam mengatasi maximum request length, serta kombinasi metode kompresi dan chunking dalam proses uploading. Hasil dari penelitian ini membuktikan bahwa metode kompresi dapat menangani maximum request length. Berdasarkan hasil percobaan yang dilakukan hubungan antara rasio kompresi dan window length saling berkolerasi positif yang artinya seiring bertambahnya window length maka rasio kompresi ikut bertambah. Sementara itu hubungan antara window length dan waktu uploading adalah berkolerasi linier negative artinya semakin bertambah window length maka

Received on February 2016. Accepted on March 2016

waktu uploading semakin cepat. Hal lain yang dapat diketahui adalah hubungan antara dekompresi dengan ukuran file berkolerasi linier positif artinya bertambahnya ukuran file maka waktu dekompresi akan semakin lama.

Kata Kunci: komponen, kompresi, AJAX, Webservice, *Chunking*, Lempel Ziv 77 (LZ77)

I. INTRODUCTION

THE recent information development requires a kind of technology that can accomodate the rapid data exhanges. In delivering the data from one point to another, the cost that is expected is the minimum one and the time that is expected is surely the fastest one. One of technologies that can handle that kind of problem is web. Besides its reachability internet/intranet website has information access with the greatest spread [3]. Exchanging files on internet is actively done by people. It is proven by the increasing popularity of some websites such as rapidshare, 4shared, gudangupload and etc[2].

In the process of uploading a file, there is common problem faced by user i.e. the file size. If someone wants to upload a large size file, he or she has a problem of request length limit in the server. Enlarging the maximum request length limit will surely requre larger memory on the server. The large file size requires the larger bandwidth on the transmission media. Besides that, the time length of sending data in information transmitting media is also the emerging problem.

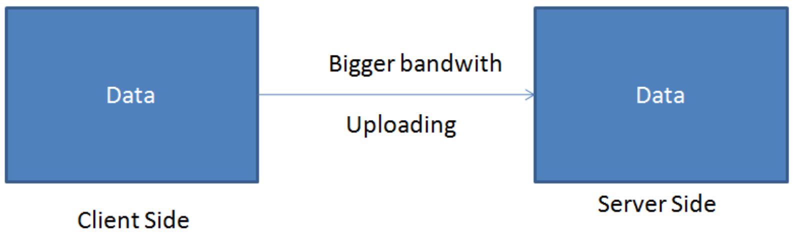

Figure 1. Uploading File Ilustration

To cope with that problem, one of the way that can be done is by downsizing the uploaded file size. There are some ways to downsizing the file, one of which is by chunking it (dividing it into smaller parts). Generally, the chunking method is done by splitting the file size on the client and sending it one by one and then rejoined on the server[2]. Another way to do it is by compressing the file and uploading it. In [3], a high data reduction rate can be achieved by the traditional variable-size block-level deduplication techniques. Hence, they require a high processing overhead due to data chunking. [3]

There are various compression algorithm that can be used, for example A 2-D EGG compression that is based on wavelet [1] and vector quantization, LZ77 compression that uses dictionary in the process[4], Huffman Coding that uses binary trees, run length coding that is done in text compression if there are some letter displayed in order. The implementations for the LZ77 encoders and Huffman encoders that form the basis for a full hardware implementation [4].

In this research, a data compression method is used through LZ77 algorithm. In addition, that method is combined with chunking method, and the performance is then analyzed.

II. THE SCOPE AND METHOD

A. The Research Questions and The Research Aims

The research questions are follow. How to do a data compression on the client and a data decompression on the server as well as how to combine compression method and chunking method in uploading process. How is the performance of the combination of compression and chunking method in data uploading. The scope or context of the researh are follow. The data that is meant is text and pictures with maximum capacity of 2 MB. The picture file is the picture file with JPG extension. The tools that are used to measure is built using AJAX, webservice and XML. The AJAX application development on the client depends on the used web browser. The web browser criteria are the ones that enable the file reading using xmlstreamer and partial reading.

Generally, the main aim of the research is to combine compression method and chunking method to to upload data. The followings are the details :

1) Analyzing the compression-decompression method on the client and on the server in data uploading process.

2) Analyzing the performance of the combination of compression and chunking method in data uploading process.

B. A Brief Review of LZ77 and Chunking

1) Lempel Ziv 77(LZ77) :LZ77 is a compression algorithm to compress the sequential data tat is reversible in nature. It means the original order of the data elements can be recovered without loss in terms of its originality.
In LZ77, there is a term “window” that consists of dictionary (search buffer) and look ahead buffer.

2) Chunking :In adopting the chunking method, files will be separated and divided into several criteria of grouping that can be seen at the picture.

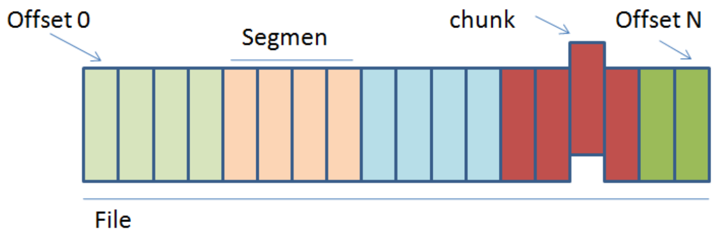

Figure 2. Chunking model

- File : the overall files that will be uploaded started from 0 up to offset n.
- Segment: The separated files in which each file consists of one or more segments and they will possess one thread connection to upload their own segment.
- Chunk : a small part of segment that will be uploaded onE by one to the server.

C. Research Method

The system designed to be used as the measuring tools is file uploading adoption using chunking and compression method based on AJAX related to the client and web services with platform .net on server’s side.

1) Data Compression Model :Data compression that uses LZ77 algorithm with its model is as follows:

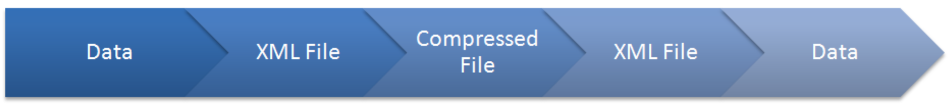

Figure 3. Data compression model

2) Uploading data model using chunking method:

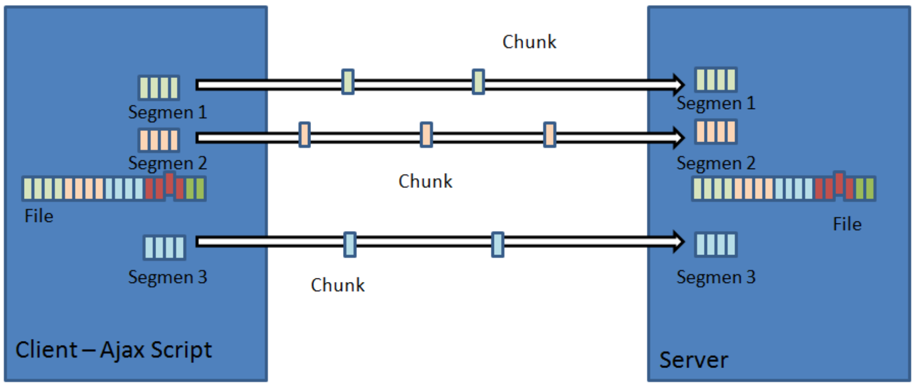

Figure 4. Uploading data using chunking method

3) *Uploading data model using compression method:*

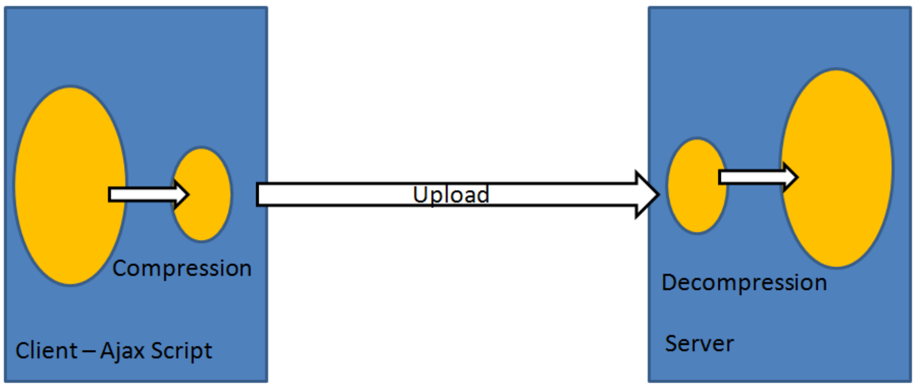

Figure 5. Compression upload model

4) *Uploading data model using the combination of chunking and compression method:*

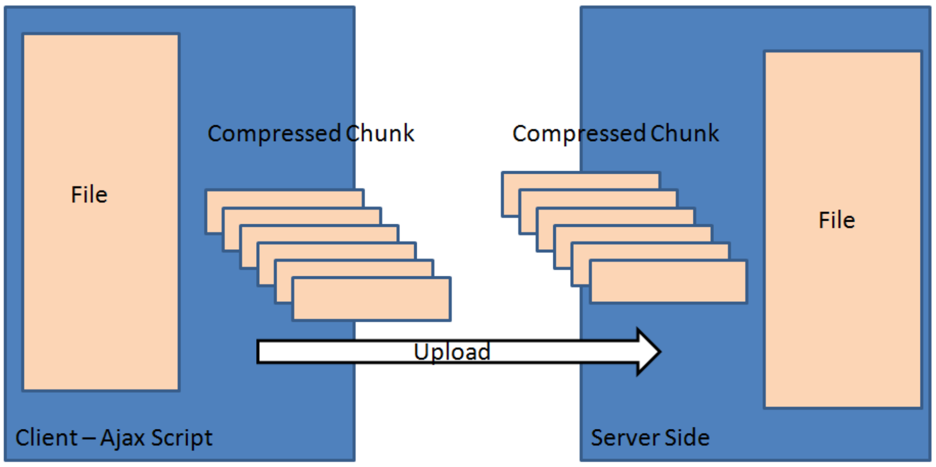

Figure 6. Uploading model using compression and chunking

III. VARIABLE AND TESTING

The software designed in this paper as the measuring tools uses AJAX and Web Service where the design overviews are as follows:

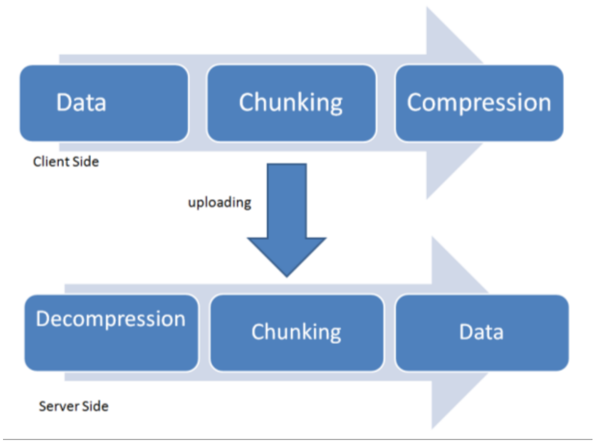

Figure 7. Aplication Design.

A. Testing Scenario

1) The system will be input by file 10Kb, 20Kb, 40Kb, 80Kb, and will be then observed up to certain file size in which the compression time is around 30 minutes.

2) The system will possess the file input and its window length size is repeatedly changed by the sequence 100, 200, 400, 800, 1600, and 3200 up to the memory leak on client. The ratio compression notification will be then processed on each window length change.

3) Uploading some of file size on two system including POST on conventional web and uploading them using compression method by changing the maximum request length become bigger or smaller than their file size. After this process, the success level measurement may be conducted further.

4) Uploading some of the file with the different amount of segment and observing its uploading time then.

5) Uploading ten pictures with different size and observing their psnr value .

B. Results

1) Scenario 1 :The result of scenario 1 is presented at the following chart.

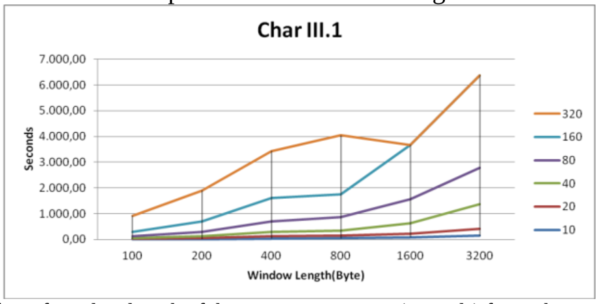

Figure 8. Effect of window length of the compression time (seconds) for each size of a text file.

The time of compression process on compression method is influenced by window length and its file size, in which the maximum file size, based on the chart, as testing file is 320 kb with window length 800. This results shows that more window length size indicates longer compression time. The compression time by the size 320 Kb and window length 800 is 2297.203 seconds which is equal to 38.29 minutes between compression time and file size.

2) Scenario 2: The result of scenario 2 can be seen at Figure 8 and Figure 9. Based on Figure 8 and Figure 9 the window length size influences the compression ratio so that if the window length is getting bigger, it leads to bigger compression ratio. This events is caused by the big window length size makes the search space and comparison space of bytes increase, therefore the compressed bytes and compression ratio are greater.

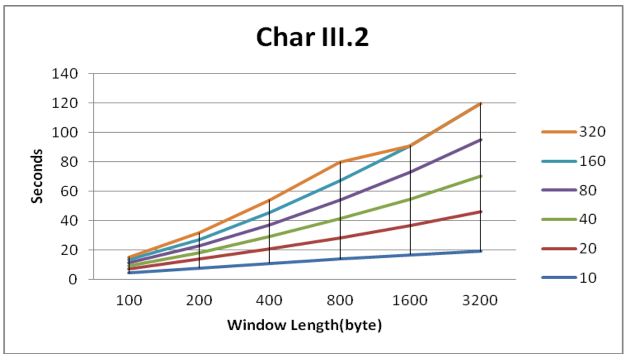

Figure 9. Window Length influence on the compression ratio (%) for each size of the text file

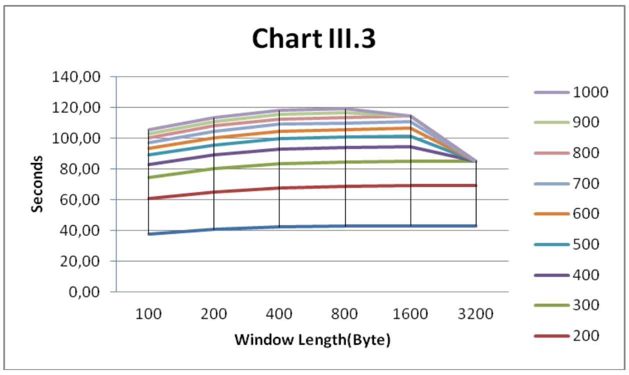

Figure 10. Window Length influence on the compression ratio (%) for each image file size.

The relation of window length and compression ratio is positive linear. This can be shown towards coefficient correlation R which has a value of more than 0.85 between compression and window length. The text file of window length change gives a big influence represented by the coefficient correlation R which has value of more than 0.85. While the picture file does not make an influence since its coefficient correlation is less than 0.85. The change of both data is caused by whether or not the data content is unique. According to the window length influence, the content of binary file data is more unique than that of text file data.

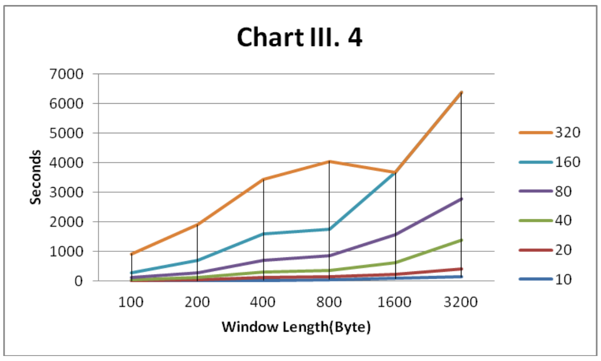

Figure 11. Effect of Window Length of the compression time (seconds) for each size of the text file.

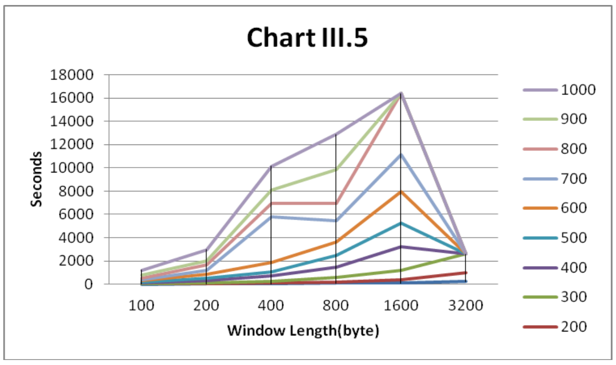

Figure 12. Effect of window length of the compression time (seconds) for each image file size.

Chart III.4 and III.5 show the file compression time, while the main factors largely influencing the time are file size and window length. It can be seen that the bigger the window length size, the longer time of compressing is. It is because when the window length is bigger, the buffer memory will be bigger, so that the browser will be slower. The compression time also increases linearly in line with the increase in file size, which is shown with the coefficient relation R bigger than 0.85. The file size also affects the compression time linearly which is reflected in the coefficient correlation R bigger than 0.85. In this study, the file size is two times bigger than the initial size. Besides that, there is also an external factor affecting the time for compression, i.e. the application used in the client's computer.

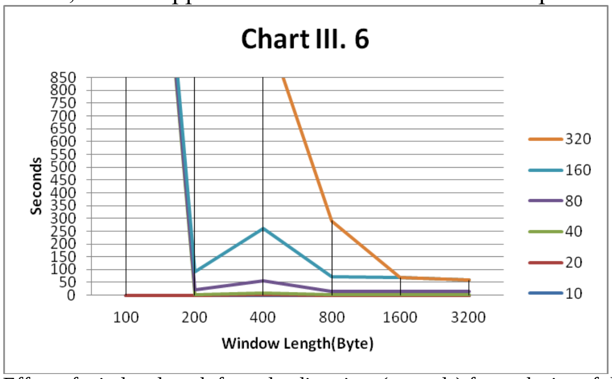

Figure 13. Effect of window length for uploading time (seconds) for each size of the text file.

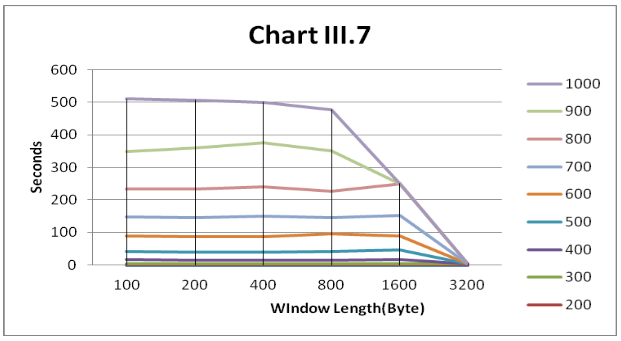

Figure 14. Effect of window length for uploading time (seconds) for each size of each image file.

Chart III.6 and III.7 show the relation among the uploading time, file size and window length The uploading time is the time set when the data are being sent which is compressed in the server. It also comprises the response time taken from server to client pertaining to the information on the file creation. It can be seen that time becomes faster in line with the increase in window length. It is because if the window length is getting bigger, the file size will be smaller since the compression ratio will be bigger. The time decreases linearly in line with the changes in window length as reflected from negative R coefficient value.

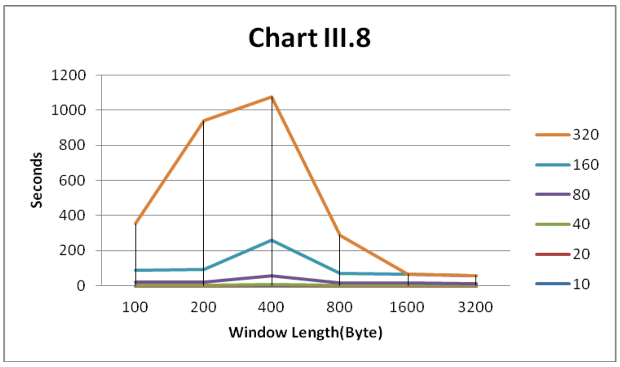

Figure 15. Effect of Window Length of the decompression time (seconds) for each size of the text file

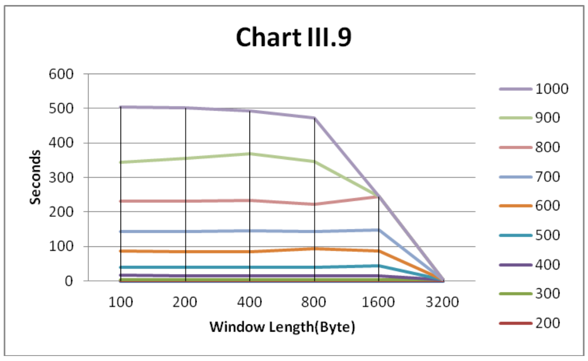

Figure 16. Effect of Window Length of the decompression time (seconds) for each image file size.

Chart III.8 and III.9 show decompression time. Decompression time is not influenced by window length size as can be seen from the negative correlation coefficient which closes to 0. However, the decompression time is influenced by the file size, where the decompression time correlates with file size linearly that can be seen from the R coefficient which is bigger than 0.85. It means that the bigger the file size the longer the decompression time is.

3) *Scenario 3:* The test result based on the scenario 3 can be seen in the following table 1.

TABLE 1.

COMPARISON SUCCESS RATE UPLOADING POSTS AND UPLOADING COMPRESSION FOR MAXIMUM REQUEST SIZE THAT IS DIFFERENT LENGTH.

	Succes Rate Uploading Post		**Succes Rate Uploading compression**	
File Size(Kbyte)	**MRL =ukuran file-1Kb**	**MRL=ukuran file+5Kb**	**MRL= ukuran file-2Kb**	**MRL=ukuran file+5Kb**
10	0/10	10/10	10/10	10/10
20	0/10	10/10	10/10	10/10
40	0/10	10/10	10/10	10/10
80	0/10	10/10	10/10	10/10
160	0/10	10/10	10/10	10/10

320	0/10	10/10	10/10	10/10

Tabel II.

Comparison Uploading Success Rate Compression for Maximum Request Size That is Different Length.

File Size (Kb)	**Success rate uploading compression**		**Success rate uploading Compression**	
	MRL=File size-4Kb	**MRL= File size +5Kb**	**MRL= File size - 8Kb**	**MRL= File size +5Kb**
10	0/10	10/10	0/10	10/10
20	0/10	10/10	0/10	10/10
40	0/10	10/10	0/10	10/10
80	10/10	10/10	0/10	10/10
160	10/10	10/10	10/10	10/10
320	10/10	10/10	10/10	10/10

Remarks: n/m success rate refers to the n times of successful experiment multiplied by m times of experiments.

The experiment in this scenario uses 200 byte window length. It can be seen that the uploading on the compression can handle the maximum request length (MRL), where in the compression upload, it can be seen that data uploaded can be minimized. The file size after the compression can be seen from the compression ratio, i.e. by the decrease in the file size, the file, though it exceeds the MRL, can be successfully uploaded file though it is bigger than the MRL size can be uploaded. Therefore, it can be said that the upload by using the compression method can pass the MRL constraint. Based on table 3.11, success rate for uploading compression with the MRL size bigger or smaller than the file size reaches the 100% of success. However, the success rate for the MRL less than file size reaches the 0% of success. While for MRL which is bigger than the file size, the success rate is 100%. The ability of compression method in handling the issue of maximum request length largely depends on the system ability in compressing data.

4) *Scenario 4* :The test result obtained from scenario IV can be seen in the following chart:

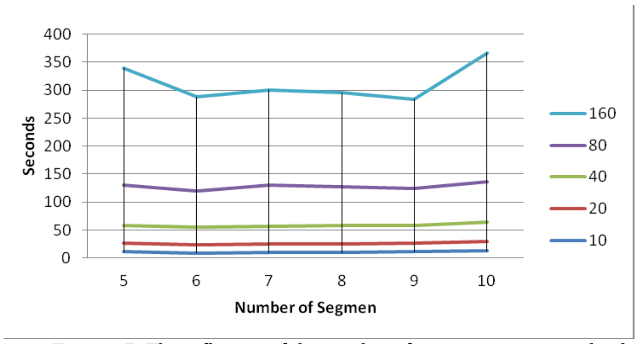

Figure 17. The influence of the number of segments in time uploading.

In this scenario test, the test used the maximum file size of 160kb. It is because if the experiment uses the file size bigger than that, there will be connection reset and request time out within the system. Connection reset and request time out happen since the web service chunking waits the compression time of client too long. Based on the table III.12, it can be seen that relation of the number of segments and uploading time is a positive linier, which can be seen from the coefficient value of R correlation. Positive linear here means that if the number of segment increases, the uploading time will increase too.

5) Scenario 5 :The test of scenario V by uploading 10 pictures in different size results in :

TABLE III.

PNSR VALUE ON THE IMAGE FILE BEFORE UNDERGOING A PROCESS OF COMPRESSION THAN THE IMAGE FILE AFTER EXPERIENCING DECOMPRESSION PROCESS.

Ukuran file	**Nilai PSNR**
100	~
200	~
300	~
400	~
500	~
600	~
700	~
800	~
900	~
1000	~

Based on the calculation on the PSNR value, it can be known that the picture quality after the decompression is unlimited which means that the picture before the compression is not different from the picture after the decompression, which shows that the LZ77 compression method is a lossless compression method.

IV. CONCLUSION

Uploading data by using the compression method can handle the constraint maximum request length since the compression method can minimize the file size. The compression ratio by using LZ77 compression method is influenced by window length where the influence is seen clearly if the compression process is conducted on the text file, instead on the picture file. The uploading time is influenced by the window length size where the linier correlation between the two factors is negative linier which means that by the increase of window length, the uploading time will also. Compression time is influenced by the window length where the linier correlation between the two factors are positive linier which means that the bigger the window length size, the bigger compression time is.

Decompression time is influenced by the file size where the linier correlation between the two factors is positive linier which means that the bigger the decompression time, the bigger the file size is. The changes in the number of segments in the combination of chunking method and compression method influence the uploading time that the bigger the segment number, the longer uploading time is. The maximum file size of 2 MB cannot be reached. The picture quality after the compression-decompression process equals to the one before the compression-decompression process. The uploading capability in the final test refers to the file size of 320kb. It is due to the browser ability in processing data.

REFERENCES

[1] Adiwijaya, M Maharani, BK Dewi, FA Yulianto, B Purnama, Digital Image Compression using Graph Coloring Quantization Based on Wavelet-SVD, Journal of Physics: Conference Series 2013 volume 423, 12-19.

[2] Asleson, Ryan and Nathaniel Schuta. 2006. Foundations of AJAX. New

York: Apress.

[3] Daehee Kim, Sejun Song, Baek-Young Choi, SAFE: Structure-aware file and email deduplication for cloud-based storage systems, International Conference on Cloud Networking (CloudNet), 2013, pp. 130 - 137 2008

[4] Rigler, S., Bishop, W., Kennings, A., FPGA-Based Lossless Data Compression using Huffman and LZ77 Algorithms, Canadian Conference on Electrical and Computer Engineering, 2007, pp. 1235 – 1238