***I.J. Education and Management Engineering*, 2019, 4, 1-10**
Published Online July 2019 in MECS (http://www.mecs-press.net)
DOI: 10.5815/ijeme.2019.04.01

Available online at ***http://www.mecs-press.net/ijem***

Effective XML Compressor: XMill with LZMA Data Compression

Suchit A. Sapate[a]

[a,] *Persistent Systems Ltd., Nagpur, 440022, India*

Received: 06 February 2019; Accepted: 25 April 2019; Published: 08 July 2019

Abstract

The XMill is an efficient XML compression tool which takes the advantage of awareness of XML. XMill compresses the data on the basis of three principles- separate the XML structure from the data, group related data and apply the semantic compressors. The XMill uses the gZip library to compress the XML string data for increasing the compression ratio. Here we have proposed a new method to increase the compression ratio of XMill tool. In this method we have added the 7Zip library to the XMill tool; 7Zip library uses the LZMA algorithm to compress the data. LZMA is an enhanced & improved version of LZ77 algorithm which is used in the gZip library. LZMA algorithm has following features over the LZ77 algorithm

- Uses up to 4GB dictionary length instead of 32KB for removing the duplicate data.
- Uses the look-a-head approach instead of greedy approach.
- Uses the optimal parsing, shorter code for recently repeated matches.
- Uses the context handling.

Due to the above features our proposed approach achieves the best compression ratio with a comparable compression speed.

Index Terms: XML, XMill, LZ77, LZMA, 7Zip, gZip.

1. Introduction

Now a day XML has become an important standard for representing or exchanging the data on World

* Corresponding author.
E-mail address: suchit.sapate2005@gmail.com

Wide Web [1]. However, self-describing nature, flexibility, simplicity and portability makes the XML more popular in the data communication, but it suffers a verbosity problem [2]. Many effective XML compressors come into scope due to the verbosity problem of XML. There are two types of compressors on the basis of XML awareness: General text-based compressor and XML concise compressor [3].

The general text-based compressor is mostly used for compressing the general text files and it includes data compressors such as gZip, WinRAR, 7Zip [3]. 7Zip compressor by default uses the 7Z format and that format by default uses the LZMA method to compress the data [6]. LZMA is an improved version of LZ77 algorithm which improves the data compression ratio [13,16].

XML concise compressors are aware about the XML structure so that they can take an advantage of XML structure to compress the data which increases the compression ratio [3]. The XMill is one of the type of XML compressor which eliminates the redundant data by identifying the similarities between the semantically related data [12]. The XMill also uses the gZip library to compress the XML string type of data [7]. To improve the compression ratio of XMill compressor we have added 7Zip library in addition with gZip to compress the XML string data.

We have specified the XMill basic introduction in the Section II and 7Zip introduction in the Section III. We have proposed the new approach for increasing the compression ratio those details are provided in the Section IV and Section V. Section VI contains the experimental results which show how our approach is better than the traditional XMill compressor.

2. Related Work

The The XMill was developed in summer 1999 in AT&T Labs by Hartmut Liefke and Dan Suciu [5]. This is an XML Aware compressor which uses the structure of XML to compress the XML [12]. The source code of this tool is now moved to the SouceForge Project so now new version of XMill tool is provided on this site [15]. The XMill compress the data on the basis of following three principles [7]:

a. Separate the XML structure from the data: XML tags and attributes create the XML structure so in this principle they separate the XML tags and attributes of the data [12]. Data can be a data item which contains the sequence of characters to represent the element contents and the attribute values [7,15].
b. Group related data item: It groups the related data items into the containers [15] for example it groups all <title> data items to form one container, same as all <id> data items formed another container [8]. These containers are then compressed separately just like the column-wise compression in the relational database.
c. Apply semantic compressors: XMill uses the semantic or specialized compressors to the different containers for compressing the XML efficiently [7,15]. For compressing textual data it uses the GZip library.

3. 7Zip Introduction

7Zip is a general text compressor which is used for archiving purpose. 7Zip is an open source compressor and most of the code of this compressor comes under the GNULGPL license [4,9]. 7Zip compressor uses the 7Z format by default [6]. 7Z format provides following main features to the 7Zip compressor:

- Achieves high compression ratios [6,10]
- Supports strong AES256 encryption [10]
- Compresses large file up to approximately 16 exbibytes i.e. 16000000000 GB [10].

- Supports the solid compression that means while compressing multiple files are treated as a single solid block due to this compression ratio increases [10]..

The 7Z is an open architecture which allows addition of any new compression methods into it [6]. At the time of paper writing following compression methods defined in it.

LZMA

It is enhanced and improved version of LZ77 algorithm which uses the sliding dictionary up to 4GB length instead of 32KB for eliminating the redundant data [6,10].

LZMA2

It is enhanced and improved version of LZMA algorithm which supports better multithreading than LZMA [10].

PPMd

It contains the PPMdH (PPMII/cPPMII) code with small changes [6]. PPMII is an improved version of PPM [10].

BZIP2

It uses the BWT algorithm. BZIP algorithm internally uses the arithmetic coding for compressing data but BZIP2 uses the Huffman coding instead of arithmetic coding [10].

Deflate

This method uses the combination of Huffman coding and LZ77 algorithm. This method compresses the file with high speed, but the compression ratio is not much higher. This method uses the dictionary-based compression approach and the dictionary length could be up to 32 KB for eliminating the redundant data [10].

7Z format uses the LZMA method by default to compress the data [6]. 7Zip also supports numerous other compression formats such as Zip, gZip, bZip, XZ, tar & WIM [4].

4. Proposed Methodology for XML Compression

XMill compressor uses the gZip library to compress the XML string data. As per our analysis we have found that we can improve the compression ratio of XMill compressor by adding the 7Zip library in it. So, we have provided one more option in XMill compressor to compress the XML string data by 7Zip library. By default, this library uses the LZMA algorithm to compress the data. LZMA algorithm uses up to 4GB length dictionary for eliminating the duplicate string to this, compression ratio improved as compared to LZ77 algorithm, but long size of dictionary makes this algorithm slower. LZ77 algorithm depends on the greedy approach for parsing whereas LZMA depends on the Look-a-head approach which makes the compression process more effective than the gZip.

4.1 Proposed System Architecture

The XML file is parsed by a SAX8 parser that sends tokens to the path processor. Every XML token (tag, attribute, or data value) is assigned to a container. Tags and attributes, forming the XML structure, are sent to the structure container. Data values are sent to various data containers, according to the container expressions, and containers are compressed independently.

The core of XMill is the path processor that determines how to map data values to containers. The user can control this mapping by providing a series of container expressions on the command line. For each XML data value the path processor checks its path against each container expression, and determines either that the value has to be stored in an existing container, or creates a new container for that value.

Containers are kept in a main memory window of fixed size (the default is 8MB). When the window is filled, all containers are compressed by 7Zip LZMA algorithm, stored on disk and the compression resumes. In effect this splits the input file into independently compressed blocks.

The decompressor is simpler, and its architecture is also like compressor architecture but data flow from bottom to top instead top to bottom and it is the contrast of compression process. After loading and decompressing the containers, the decompressor parses the structure container, invokes the corresponding semantic decompressor for the data items and generates the output.

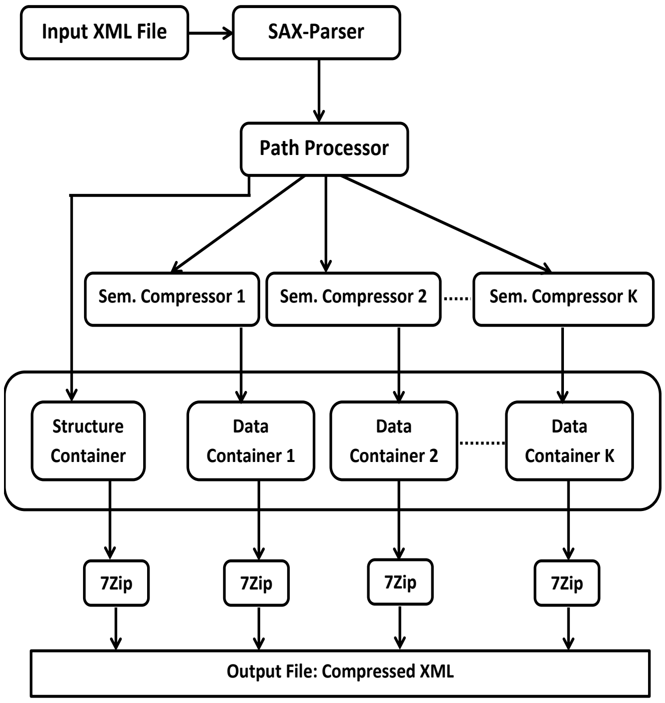

Fig. 1. Proposed System Architecture for Compression

5. Experimental Results and Discussion

In this section we have performed the wide range of experiments to compare the compression performance of the existing XML compressors such as XMill, gZip and LZMA with our approach that is XMill-LZMA. Here we have compared the compressors based on following four parameters.

1. Compression Ratio: This can be calculated by finding the ratio between the compression size and the original size of XML file.

$$\boxed{\textit{Percentage Compression Ratio} = ((1) - (\textit{Compressed Size}) \div (\textit{Original Size})) \times (100)}$$

2. Compression Time: This represents the duration required to compress the XML file.

3. Decompression Time: This represents the duration required to decompress the compressed XML file.
4. Compressed File Size: This parameter represents the size of the file after the compression.

Experimental Environment

We have used following environment set up to run the performance analysis of GZip, LZMA, XMill and XMill-LZMA compressors.

Table 1. Experimental Environment Specification

Operating System:	Window 7 Enterprise
Processor:	Intel Core 2 Duo T6670 CPU 2.20 GHz
RAM:	4.00 GB (3.46 GB Usable)
System Type:	32-bit Operating System
Hard Disk:	320 GB (TOSHIBA MK3276GSX ATA)

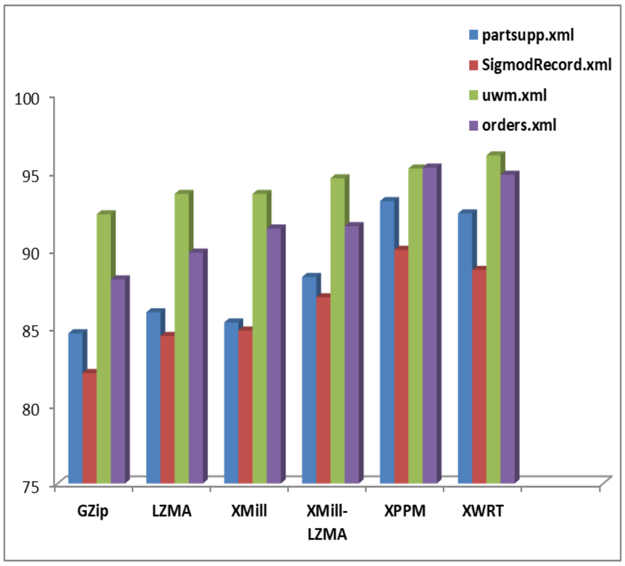

Fig. 2. Compression Ratio Graph

Experimental Datasets

We have used following XML files to do the performance analysis of GZip, LZMA, XMill and XMill-LZMA compressors [18].

Table 2. Experimental Database

XML File Name	File Size	Elements (Tags)	Attributes	Max. Depth	Avg. Depth
partsupp.xml	2 MB	48001	1	3	2.8333
SigmodRecord.xml	467 KB	11526	3737	6	5.14107
uwm.xml	2 MB	66729	6	5	3.95243
orders.xml	5 MB	150001	1	3	2.89999

Experimental Comparison and Discussion

The compression ratio graph i.e. figure 4 shows the XPPM compressor better than all other compressors but the process of compression of XPPM is too slow. The proposed compressor i.e. XMill-LZMA achieves the better compression ratio than the XMill compressor with comparable compression time. The compression ratio improvements of all algorithms are calculated with respect to GZip algorithm. Table 3 shows that compression ratio of proposed approach is 4.09% better than the GZip algorithm.

Table 3. Compression Ratio Evaluation

Algorithm Name	Avg. Compression Ratio	Improvement Evaluated with respect to GZip
GZip	86.7775	--
LZMA	88.4675	1.95%
XMill	88.78	2.31%
XMill-LZMA	90.33	4.09%
XPPM	93.4075	7.64%
XWRT	92.9825	7.15%

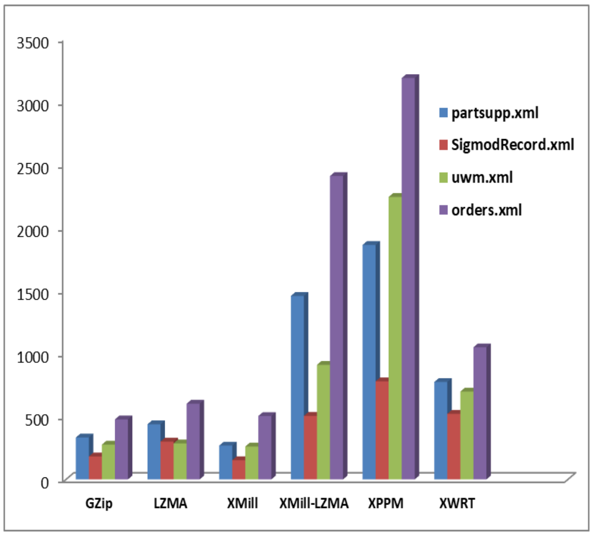

Fig. 3. Compression time Graph

GZip-based compressors have faster compression time, but they have the worst compression ratio, so the XMill compressor has the worst compression ratio. The compression time graph i.e. figure 3 shows the XMill compressor better than all other compressors. As compare to XPPM, the proposed compressor achieves the overall best average compression ratio with less cost term of compression time. Table 4 shows the compression time evaluation with respect to XPPM algorithm. The XMill algorithm needs 85.3% less time with respect to XPPM that means the XMill compression process is faster than XPPM. The proposed approach needs 34.57% less time than XPPM algorithm.

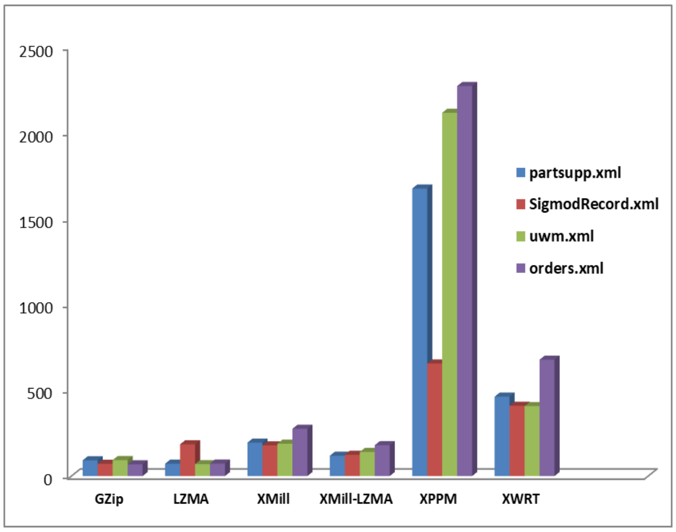

Fig. 4. Decompression time

Table 4. Compression Time Evaluation

Algorithm Name	Avg. Compression Time	Improvement Evaluated with respect to XPPM
GZip	316.895	-84.28
LZMA	406.2125	-79.85
XMill	296.355	-85.30
XMill-LZMA	1319.2425	-34.57
XPPM	2016.255	--
XWRT	759.4775	-62.33

The decompression time graph figure 4 shows the XMill-LZMA compressor takes very less time than other XML aware compressors. Table 5 shows the decompression time evaluation with respect to XPPM algorithm. The XMill-LZMA algorithm needs 91.62% less time than the XPPM. So, this approach is much optimized approach in terms of decompression time parameter.

Table 5. Decompression Time Evaluation

Algorithm Name	Avg. Decompression Time	Improvement Evaluated with respect to XPPM
GZip	81.2875	-95.16%
LZMA	99.7275	-94.06%
XMill	209.4125	-87.53%
XMill-LZMA	140.715	-91.62%
XPPM	1679.1375	--
XWRT	489.2575	-70.86%

Table 6 shows the memory space required to store the compressed file with respect to GZip algorithm. The XMill-LZMA needs 27.38% less space to store the compressed file. Experimental results of compressed file size graph i.e. figure 5 shows the XMill-LZMA needs very less memory size to store the compressed XML file as compare to the XMill tool.

Table 5:- Compressed File Size Evaluation

Algorithm Name	Avg. File Size	Improvement Evaluated with respect to GZip
GZip	305.36	---
LZMA	265.0775	-13.19%
XMill	247.865	-18.83%
XMill-LZMA	221.765	-27.38%
XPPM	138.475	-54.65%
XWRT	145.5625	-52.33%

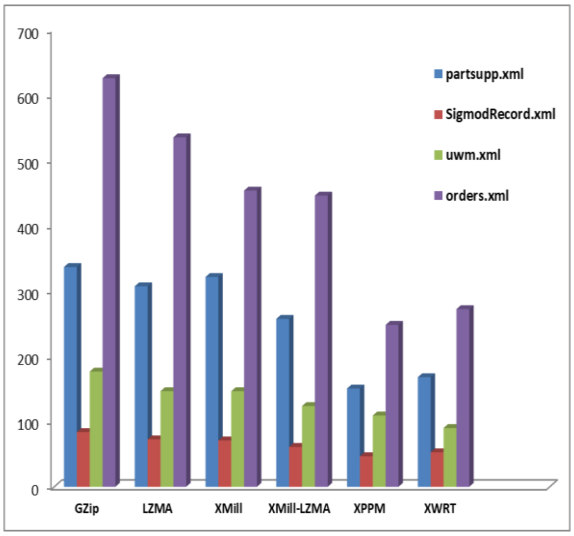

Fig. 5. Compressed file size

The following summary report is used to evaluate the performance of the algorithm in the respective parameter. This report provides the guidelines for selecting the most appropriate XML compression tool on the basis of need and applicability. For e.g. when user has a requirement, compression size of file will be very low with compromising the compression speed in that case XPPM will be better choice, but user want low compressed file size with medium speed then XMill-LZMA will be good choice of that requirement. The rank specifies the performance of the respective algorithm in performance evaluation parameter. Rank 1 signifies the best performance and Rank 6 signifies the worst performance in that category.

Table 6. - Summary Report

Algo.	**Compression Ratio**	**Compressed File Size**	**Compression Time**	**Decompression Time**
XPPM	Rank 1	Rank 1	Rank 6	Rank 6
XWRT	Rank 2	Rank 2	Rank 4	Rank 5
XMill-LZMA	Rank 3	Rank 3	Rank 5	Rank 3
XMill	Rank 4	Rank 4	Rank 2	Rank 4
LZMA	Rank 5	Rank 5	Rank 3	Rank 2
GZip	Rank 6	Rank 6	Rank 1	Rank 1

6. Conclusions

In this paper we have presented our approach to enhance the XMill tool. With the experimental results we have shown that how this approach is more effective than the traditional XMill approach. This approach increases the compression ratio and reduces decompression time of XMill. This approach is used when user wants good compression ratio by compromising the speed of compression such as when a user wants to send some XML file via mail in that case this approach is used because it saves the network bandwidth. For archiving the files this approach is beneficial as compare to traditional XMill approach.

References

[1] Vojtech Toman, “Compression of XML Data,” Charles University, Master’s Thesis at Department of Software Engineering, March 2003

[2] Mark Nottingham and David Orchard,"On XML Optimization," BEA Systems Position Paper, Binary Interchange of XML Workshop, 2003.

[3] Sherif Sakr, "XML compression techniques: A survey and comparison," Elsevier, Information and Software Technology, 75 (2009) 303–322, 2009.

[4] Wikimedia Foundation, Inc, “7-Zip,” 29 May 2014, http://en.wikipedia.org/wiki/7-Zip .

[5] Smitha S. Nair, "XML Compression Techniques: A Survey," Department of Computer Science, University of Iowa, USA, https://people.ok.ubc.ca/rlawrenc/research/Students/SN_04_XMLCompress.pdf.

[6] Igor Pavlov, “7-Zip,” 2013, http://www.7-zip.org/.

[7] H. Liefke and D. Suciu, “XMill: An Efficient Compressor for XML Data,” Proc. of ACM SIGMOD Intl. Conf. on Management of Data, May 2000.

[8] Pankaj M. Tolani, Jayant R. Haritsa, “XGRIND: A query-friendly XML compressor,” in: ICDE ’02: Proceedings of the 18th International Conference on Data Engineering, IEEE Computer Society, Washington, DC, USA, 2002, p. 225.

[9] Markhor, "CodePlexProject Hosting for Open Source Software," Feb 6, 2012, version 70.

https://sevenzipsharp.codeplex.com/
[10] Wikimedia Foundation, Inc, “7Z,” 19 May 2014, http://en.wikipedia.org/wiki/7z.
[11] Wikimedia Foundation, Inc, “LZ77 and LZ78,” 21 April 2014, http://en.wikipedia.org/wiki/LZ77_and_LZ78.
[12] Wilfred Ng, Lam Wai Yeung and James Cheng, "Comparative Analysis of XML Compression Technologies," World Wide Web: Internet and Web Information Systems, 9, 5–33,Springer Science + Business Media, Inc. Manufactured in The Netherlands,DOI: 10.1007/s11280-005-1435-2, 2005.
[13] Wikimedia Foundation, Inc, “Lempel-Zip-Markov chain algorithm,” 5 June2014, http://en.wikipedia.org/wiki/Lempel%E2%80%93Ziv%E2%80%93Markov_chain_algorithm.
[14] E.Jebamalar Leavline and D.Asir Antony Gnana Singh, "Hardware Implementation of LZMA Data Compression Algorithm," International Journal of Applied Information Systems, Foundation of Computer Science FCS, Volume 5, Issue-4, March 2013.
[15] H. Liefke and D. Suciu, “An Extensible Compressor for XML Data,” Proc. of ACM SIGMOD Intl. Conf. on Management of Data, 2000.
[16] Nandan Phadke, Omkar Bahirat, Tejaswi KONDURI, and CHANDRAMA THORAT, "PARALLEL DATA COMPRESSION USING LZMA," International Journal of Advanced Computational Engineering and Networking, Volume-1, Issue-2, April-2013.
[17] XMill Compressor, http://sourceforge.net/projects/xmill.
[18] XMLDataset. http://www.cs.washington.edu/research/xmldatasets/www/repository.html

Author’s Profile

Suchit A. Sapate pursed Bachelor of Engineering in Computer engineering department from University of Nagpur, India in 2008 and Master of Technology in Computer Science & engineering department from University of Nagpur, India in 2015. He is currently working as Project Lead in Persistent Systems Ltd. since 2008. His main research work focuses on Data Analytics and Data Mining. He has published 3 papers in reputed international journal.

How to cite this paper: Suchit A. Sapate," Effective XML Compressor: XMill with LZMA Data Compression", International Journal of Education and Management Engineering(IJEME), Vol.9, No.4, pp.1-10, 2019.DOI: 10.5815/ijeme.2019.04.01