Systematics Review and Comparative Analysis among Various XML Compression Techniques

Gaurav Jaiswal
Dept. of CSE
SRMS College of Engineering and Technology
Bareilly, India

Manvi Mishra
Dept. of IT
SRMS College of Engineering and Technology
Bareilly, India

***Abstract*— The Extensible Markup Language (XML) has been acknowledge as the defacto standard for data exchange over the web and data representation. But on the other hand its main drawback that of being huge in size. The huge document size means that the amount of information has to be stored, transmitted, and queried is often larger than that of other data formats. Several XML compression techniques have been introduced to deal with these problems. In this paper, we present an experimental study of available XML compression techniques and we provide guidelines for users for making an effective decision towards selecting the most suitable XML compression tool according their needs.**

***Keywords*- XML, XML compression, Binary XML, Fast Infoset, Efficient XML Interchange, Gzip**

I. Introduction

In recent years, XML [1] has become indispensable for web services, document markups and data formats. XML has been used in solving numerous problems such as storing large volumes of either structure or semi-structure data etc. It is also referred to as "self-describing data" because the schema is repeated in the document. This feature introduces the problems of "verbosity" of XML document, which increase the document size. Although disk capacity is less often a concern these days, transmitting XML-tagged data still requires significantly more bandwidth and longer parse time.

To tackle this problem, several researches proposed the use of XML compression techniques or tools with variety of perspectives. Some have aim to achieve minimal size [2] and other focus on efficient streaming [3].The aim of this paper is to provide a systematic review of the all XML compression techniques and find out the best compression technique among all.

The paper proceeds as follows. Section 2 introduces the existing binary XML compression techniques. Section 3 presents the experimented work. The detailed result of our experiments and related discussion are given in Section 4. Conclusions and future work have been discussed in Section 5.

II. REVIEW OF EXISTING COMPRESSION TECHNIQUE

Today, there are number of XML compression techniques available. This section describes a few of the most promising techniques such as – GZip, XMill, Fast Infoset (FI), BZip2, 7-Zip, Fujitsu XML Data Interchange (FXDI) and Efficient XML interchange (EXI).

A. GZip

This is a DEFLATE lossless general purpose dictionary compressors. Gzip [4] is a combination of the LZ77 algorithm and Huffman coding. It was developed by Jean-Loup Gailly and Mark Alder. The benefits of using such a tool are that it would be widely available in both open sources and commercial implementations. It provides better compression rates (40-50%) and freedom from patented algorithm. There is no need of knowledge of the document structure [5]. However the main problem of using GZip compressor to compress XML file is that the compression of attributes /Elements may be limited due to long – range dependencies between attributes and between elements.

B. 7-Zip

7-Zip [6] is another compression technique, uses the Lempel – Ziv – Markov Chain algorithm (LZMA), which is an improved version of LZ77. This gives superior results compared to GZip. The main features of 7Zip are that it supports 256 Bit AES cipher and command line interface. It has number of compression and non-compression archive formats.

C. BZip2

This compressor [7] uses lossless Burross Wheeler block sorting text compression algorithm with Huffman coding. This gives considerably better result than achieved by LZ77/LZ78 based compression. The BZip2 compress large files in blocks. The block size affects both the compression ratio achieved and the amount of memory needed for compression and decompression. The BZip2 Compression compresses file at a higher compression ratio than those compressed using GZip but it has slower performance. BZip2 will perform best on machines with very large cache.

D. XMill

XMill [8] was designed at AT and T labs and was developed by Hartmut Liefke and Den Suciu in 1999. It is a lossless schema independent user configurable XML compression. XMill compressor applies a pre-processing transform and then uses GZip compression [9]. XMill claims to reduce the network bandwidth. XMill is faster than GZip in XML publishing. The relative advantage of XMill depends on the

Proc. of the Intl. Conf. on Advances in Electronics, Electrical and Computer Science Engineering — EEC 2012
Edited by Dr. R. K. Singh.

ISBN: 978-981-07-2950-9

application it is used. The main drawback of using XMill is that if the input document size is 220KB. XMill is not efficient.

E. Fujitsu XML Data Interchange (FXDI)

The Fujitsu XML Data Interchange [10] is based on the W3C XML Schema Post Schema Validation Infoset (PSVI) using the Fujitsu Schema Compiler to compile World Wide Web Consortium (W3C) XML Schema into a "Schema corpus". FXDI goals are document compactness with fast encoder and decoder programs, which run with a small footprint without involving much complexity. FXDI works well with conventional XML document redundancy based compression such as GZip.Fujitsu XML Data Interchange (FXDI) [11] Format has been designed to serve as an alternative encoding of XML infoset that allows for more efficiency both in the exchange of data between applications and in the processing of data at each end-point. The goals set for FXDI to achieve in its design included document compactness and the ability to allow implementing decoder and encoder programs that run fast, are of small footprint without involving much complexity. Although FXDI performs much better when XML Schemas are prescribed before documents are processed, it is capable of handling schema-less documents and fragments by its support for infoset tokenization.

F. Fast Infoset (FI)

Fast Infoset is a standard based open binary format, based on XML information set ITU-TX.891/ISO/IEC 24824-1 [12]. The Fast Infoset technology provides an alternative to World Wide Web Consortium (W3C) XML syntax as a mean of representing instance of the W3C XML information set. Fast Infoset specifies the use of several techniques that minimize the size of the encoding and that maximise the speed of creating and processing Fast Infoset document. The use of tables and indexing is the primary mechanism for FI compression. Fast Infoset document 30-70% smaller than XML document. Fast Infoset approach includes the schema and application classes which are not schema optimized and allow restricted alphabets. Fast Infoset compression is much faster than using Zip-style compression algorithms on an XML stream, but they produce slightly larger files and not efficient for large XML files.

G. Efficient XML Interchange (EXI)

Efficient XML Interchange [13, 14, 15] is a specification of binary coding of the XML data. EXI is a very compact representation for the XML Information Set that is intended to simultaneously optimize performance and utilization of computational resource. For efficiently encoding XML streams, the EXI format is using a relatively simple algorithm, and a small set of data type representations. EXI is compatible with XML at the XML Information Set level, rather than the XML syntax level. This permits it to encapsulate an efficient alternative syntax and grammar for XML. EXI is "schema-informed", which allows utilizing the available schema information to improve compactness and performance. Additionally, the user may set any option to customize additional information such as schemaless document, data block size, compression, etc. These options make EXI more flexible and useful for the user. EXI has many option reflected in EXI header [10]. They are represented as an EXI Option document, which is an XML document encoded using the EXI format.

III. Experimental Analysis

The aforementioned compression mechanisms were evaluated in terms of their performance. Compression tests were executed on a machine having specification given in Table 1. We choose the powerful resource environment for better results.

Table 1 Specification of machine

Operating System	Windows 7
CPU	Intel Core 2 duo CPU T 5450 1.67 GHz, 2MB L2 cache, FSB 669MHz
Hard Disk	160 GB, Toshiba MK1637 GSX ATA
RAM	1 GB

In our study we consider all of the available compression tools which are satisfying the following conditions. It is freely available. The tools support either schema- dependent or schema independent file system. Compression tools should be able to run under the windows environment. On behalf of above condition, we examined five compressors: the three general purpose compressors (gzip, 7-Gip and bzip2), and two binary XML compressors (FI and EXI). Here performance is evaluated in terms of compression ratio.

To perform experimental work, different compression tools such as EXIProcessor [16], FI converter [17], GZ compressor, and 7-zip compression tool are used. Other tools are the EXIficient [18] and OpenEXI [19].

Here is an example of XML file used in this experiment.

```
<? XML version = "1.0" ?>
  <x:books xmlxns:x = "urn:books">
    <book id = "bk001">
        <author>Writer</author>
        <title>The First Book</title>
        <genre>Fiction</genre>
        <price>44.95</price>
        <pub_date>200-10-01</pub_date>
        <review>An amazing story of nothing</review>
    </book>
    <book id = "bk002">
         <author>Poet</author>
         <title>The Poet's First Poem</title>
         <genre>Poem</genre>
         <price>24.95</price>
        <review>Least poetic poems</review>
     </book>
</x:books>
```

Fig. 1 Example of Notebook.xml file

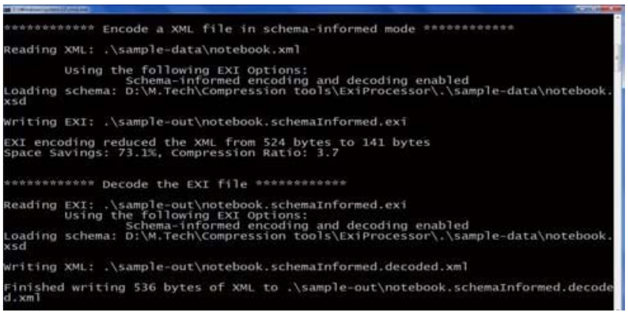
To obtain the performance three XML files (notebook.xml, oreder.xml, and store.xml) are used as an input and their compression size has been evaluated. After reviewing these XML data compression mechanism comparative analyses are done which are shown in table 2.

Table 2 Comparative analysis among various XML compression techniques

Parameters	EXI	FI	FXDI	XML+ Gzip
Organization	Agile delta	SUN Microsystem	Fujitsu	GNU
Standard Bodies	W3C	ISO/IEC-ITU-T	W3C	GNU-GPL
Human readable	✓	✓	✓	✓
Generality	✓	×	✓	×
Platform Independent	✓	✓	✓	✓
Availability	✓	✓	✓	✓
Compression Ratio	High	Low	Medium	Low
Schema Support	✓	×	✓	×
Compactness	High	Low	Medium	Very Low
Roundtrip Support	✓	✓	✓	✓

IV. Results and Discussions

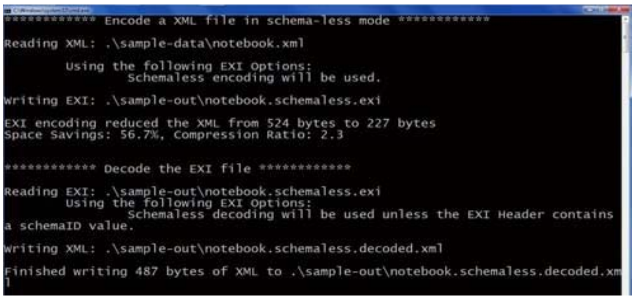
After identifying various XML compression mechanisms, comparative results are obtained. The results of experiments compare the compression ratio efficiency of available compression techniques, such as Gzip, 7-gip, bzip2, Fast Infoset, and Efficient XML Interchange.
The experiment results are shown in from fig.2 to fig.6. Collected result will be useful for determining what is the most efficient way to compress the XML files.



```
************ Encode a XML file in schema-informed mode ************

Reading XML: .\sample-data\notebook.xml

        Using the following EXI Options:
                Schema-informed encoding and decoding enabled
Loading schema: D:\M.Tech\Compression tools\ExiProcessor\.\sample-data\notebook.
xsd

Writing EXI: .\sample-out\notebook.schemaInformed.exi

EXI encoding reduced the XML from 524 bytes to 141 bytes
Space Savings: 73.1%, Compression Ratio: 3.7

************ Decode the EXI file ************

Reading EXI: .\sample-out\notebook.schemaInformed.exi
        Using the following EXI Options:
                Schema-informed encoding and decoding enabled
Loading schema: D:\M.Tech\Compression tools\ExiProcessor\.\sample-data\notebook.
xsd

Writing XML: .\sample-out\notebook.schemaInformed.decoded.xml

Finished writing 536 bytes of XML to .\sample-out\notebook.schemaInformed.decode
d.xml
```

Fig. 2 XML to EXI conversion in schema-informed mode using EXIProcessor

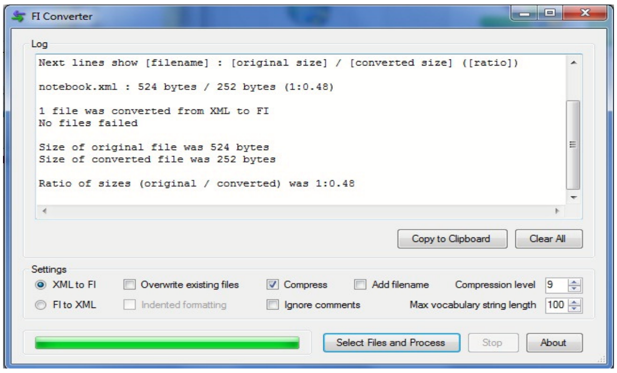
The fig.2 depicts the level of compression of notebook.xml files in schema-informed mode which saved 73.1% space using EXIProcessor.



```
************ Encode a XML file in schema-less mode ************

Reading XML: .\sample-data\notebook.xml

        Using the following EXI Options:
                Schemaless encoding will be used.

Writing EXI: .\sample-out\notebook.schemaless.exi

EXI encoding reduced the XML from 524 bytes to 227 bytes
Space Savings: 56.7%, Compression Ratio: 2.3

************ Decode the EXI file ************

Reading EXI: .\sample-out\notebook.schemaless.exi
        Using the following EXI Options:
                Schemaless decoding will be used unless the EXI Header contains
a schemaID value.

Writing XML: .\sample-out\notebook.schemaless.decoded.xml

Finished writing 487 bytes of XML to .\sample-out\notebook.schemaless.decoded.xm
l
```

Fig. 3 XML to EXI conversion in schemaless mode using EXIProcessor

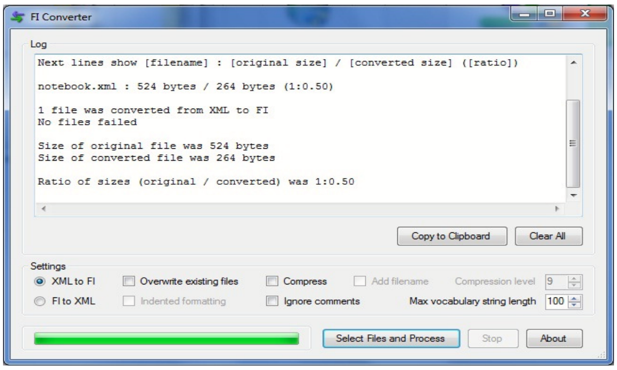
The fig. 3 shows the conversion of notbook.xml files into notebook.exi file in schemaless mode which saved 56.7% space using EXIProcessor. The Fig. 4 represents the XML to FI conversion in compress mode using FI converter. In this mode FI converter uses the gzip compression technique for compression.



```
Next lines show [filename] : [original size] / [converted size] ([ratio])

notebook.xml : 524 bytes / 252 bytes (1:0.48)

1 file was converted from XML to FI
No files failed

Size of original file was 524 bytes
Size of converted file was 252 bytes

Ratio of sizes (original / converted) was 1:0.48
```

● XML to FI ☐ Overwrite existing files ☑ Compress ☐ Add filename Compression level 9
○ FI to XML ☐ Indented formatting ☐ Ignore comments Max vocabulary string length 100

Fig. 4 XML to FI file conversion with compression using FI converter

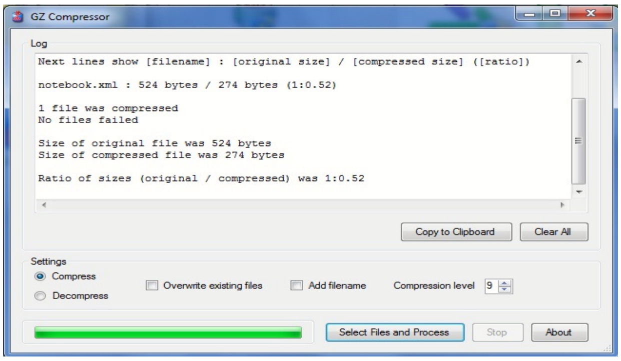


```
Next lines show [filename] : [original size] / [converted size] ([ratio])

notebook.xml : 524 bytes / 264 bytes (1:0.50)

1 file was converted from XML to FI
No files failed

Size of original file was 524 bytes
Size of converted file was 264 bytes

Ratio of sizes (original / converted) was 1:0.50
```

● XML to FI ☐ Overwrite existing files ☐ Compress ☐ Add filename Compression level 9
○ FI to XML ☐ Indented formatting ☐ Ignore comments Max vocabulary string length 100

Fig. 5 XML to FI files conversion without compression using FI converter

The fig. 5 represents the use of FI converter without using gzip compression technique. The fig.6 deals with the conversion of XML file to GZ file with their compressed file size and ratio of size using GZ compressor.



```
Next lines show [filename] : [original size] / [compressed size] ([ratio])

notebook.xml : 524 bytes / 274 bytes (1:0.52)

1 file was compressed
No files failed

Size of original file was 524 bytes
Size of compressed file was 274 bytes

Ratio of sizes (original / compressed) was 1:0.52
```

● Compress ○ Decompress □ Overwrite existing files □ Add filename Compression level 9

Fig. 6 XML to GZ files conversion

On the basis of this practical implementation we achieved the results shown in table 3.

Table 3 Comparative Compression results upon three different XML documents

File name	original file size (byte)	After Compression file size (in bytes)						
		Gzip	Bzip 2	7 zip	FI	FI + Gzip	EXI (w/o)	EXI (w)
Ored er.x ml	337	208	229	308	160	146	124	61
Note book .xml	524	274	300	381	264	252	227	141
Store .xml	1056	508	533	725	490	478	454	369

The performance results are also shown in graphical charts. These charts show the compression ratio of three different XML files and the compressed file size. Fig.7 represents the compressed XML file size using various compression techniques and fig.8 shows the compression ratio achieved by our three inputs using various compressions techniques or tools.

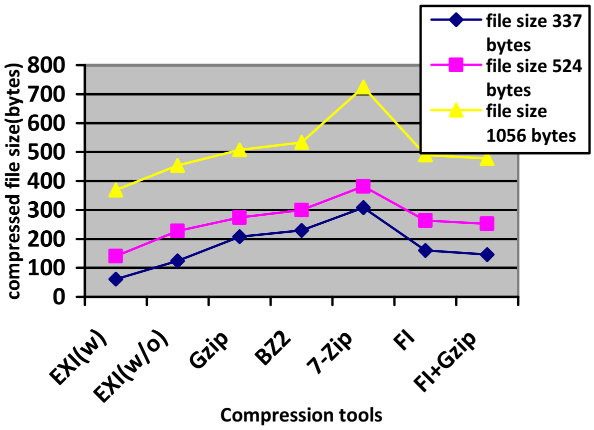


Fig.7 Compression results

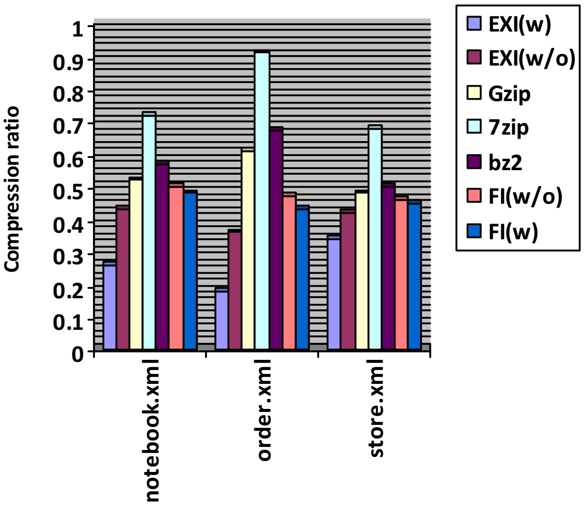


Fig. 8 Compression Ratio of various compression tools on different XML file size

V. CONCLUSION AND FUTURE WORKS

The experimental results shows that EXI in schema informed mode achieve best compression ratio than any other compression tools. The result shows the most compact candidates are in order EXI and FI. EXI is the best performer and FI is second one. In some instances, a general purpose compressor (Gzip) should be used. If maximum parsing speed is needed in XML intensive application compressors such as XMill may be useful [20].

Several avenues of research were not described during this study. In this study, decompression scenario has not been discussed. Some other important factors like response time and memory utilisation are also not evaluated. So our future work will be incorporate these essential parameters and concentrate on Efficient XML Interchange (EXI) techniques which is best one.

REFERENCES

[1] T. Bray, J.Paoli, C.M. Sperberg-Mcqueen, et al. *"XML 1.1", 2nd edition*, aug.2006.

[2] J. Adiego, G. Navarro, and P.Dela Fuente "*Using structured contexts to compress semistructured text collections*", Information processing and management, 2007.

[3] M. Girardot and N. Sundarsan, "*Millau: An encoding format for efficient representation and exchange of XML over the web*", Computer Network, vol. 33, pp.747-765, 2000.

[4] J. Gailly and M. Adler. Available: http://www.gzip.org/

[5] Eric Severson and Lee Fife, "*XML Compression: Optimizing performance of XML application in Flatiron solution corporation*", 2003.

[6] The 7-zip website. [Online]. Available: http://www.7-gip.org/ (2010).

[7] The bzip website. [Online]. Available: http://www.bzip.org/ (2012).

[8] The XMill website. [Online]. Available: http://www.liefke.com/hartmut/xmill/xmill.html/ (2004).

[9] H. Liefke and D. Suciu, "*Xmill: An efficient compressor for XML data*" in proceeding of the International Conference on Management of Data[SIGMOD], pages 153-164, 2000.

[10] W3C: "*Efficient XML Interchange Measurement Note*",. Available: http://www.w3.org/TR/2007/WD-exi measurement-20070725(October 7, 2009)

[11] W3C [Online]. Available : http://www.fujitsu.com/global/services/software/interstage/xbrltools/fxdiindex.html/(2006)

[12] The ITU website. [Online]. Available: http://www.itu.int/ITU-T/asn1/xml/finf.htm/.(2004)

[13] W3C: "*Efficient XML Interchange Best Practices*", from http://www.w3.org/TR/2007/WD-exi best-practices-20071219(2007, Dec 19).

[14] W3C: "*Efficient XML Interchange evaluation*", from http://www.w3.org/TR/2007/WD-exi-evalution-20090407 (2009).

[15] W3C: "*Efficient XML Interchange primer*", retrieved September, 2009, from http://www.w3.org/TR/2007/WD-exi-primer-20071219 (2007).

[16] The SourceForge website [Online]. Available: http://sourceforge.net/projects/exiprocessor/

[17] The noemax website [Online]. Available: http://www.noemax.com/downloads/fast_infoset_converter.asp/

[18] The sourceforge website [Online]. Available: http://exificient.sourceforge.net/?id=downloads

[19] The sourceforge website [Online]. Available: http://openexi.sourceforge.net/#download

[20] Christopher J. augeri, Dursun A. Bulutoglu, Barry E. Mullins, et al., "*An Analysis of XML Compression Efficiency*", June 2007. .