An Empirical Evaluation of XML Compression Tools

Sherif Sakr

School of Computer Science and Engineering
University of New South Wales

1st **International Workshop on Benchmarking of XML and Semantic Web Applications (BenchmarX'09)**

20 April 2009

XML Compression: Why?

- XML has become a popular standard with many useful applications.
- XML is often referred as **self-describing data**.
  - On one hand, this self-describing feature grants the XML great flexibility.
  - On the other hand, it introduces the main problem of **verbosity**.
- XML compression has many advantages such as:
  - Reducing the network bandwidth required for data exchange.
  - Reducing the disk space required for storage.
  - Minimizing the main memory requirements of processing and querying XML documents.

XML Compressors: Classifications I

With respect to the awareness of the structure of the XML documents:

- **General Text Compressors**: They are *XML-Blind*, treats XML documents as usual plain text documents and applies the traditional text compression techniques.
- **XML Conscious Compressors**: They are designed to take the advantage of the awareness of the XML document structure to achieve better compression ratios over the general text compressors.
  - **Schema dependent compressors**: Both of the encoder and decoder must have access to the document schema information achieve the compression process. *They are not commonly used in practice*.
  - **Schema independent compressors**: The availability of the schema information is not required to achieve the encoding and decoding processes.

XML Compressors: Classifications II

With respect to the ability of supporting queries:

- **Non-Queriable (Archival) XML Compressors**: They do not allow any queries to be processed over the compressed format. They are mainly focusing to achieve the highest compression ratio.
  - *By default, general purpose text compressors belong to the non-queriable group of compressors.*
- **Queriable XML Compressors**: They allow queries to be processed over their compressed formats. The compression ratio is usually worse than that of the archival XML compressors. The main focus is to avoid full document decompression during query execution.
  - *The ability to perform direct queries on compressed XML formats is important for many applications which are hosted on resource-limited computing devices such as: mobile devices and GPS systems .*
  - *By default, all queriable compressors are XML conscious compressors as well.*

Examination Criteria

In our study we considered, to the best of our knowledge, **all** XML compressors which are fulfilling the following conditions:

- Is publicly and freely available either in the form of open source codes or binary versions.
- Is schema-independent.
- Be able to run under our Linux version of operating system.
  - Ubuntu 7.10 (Linux 2.6.20 Kernel)
  - Ubuntu 7.10 (Linux 2.6.22 Kernel)

Examined Compressors

XML Compressors List

Compressor	Features	Code Available	Compressor	Features	Code Available
GZIP (1.3.12)	GAI	Y	XGrind	SQI	Y
BZIP2 (1.0.4)	GAI	Y	XBzip	SQI	N
PPM (j.1)	GAI	Y	XQueC	SQI	N
XMill (0.7)	SAI	Y	XCQ	SQI	N
XMLPPM (0.98.3)	SAI	Y	XPress	SQI	N
SCMPPM (0.93.3)	SAI	Y	XQzip	SQI	N
XWRT (3.2)	SAI	Y	XSeq	SQI	N
Exalt (0.1.0)	SAI	Y	QXT	SQI	N
AXECHOP	SAI	Y	ISX	SQI	N
DTDPPM	SAD	Y	XAUST	SAD	Y
rngzip	SQD	Y	Millau	SAD	N

Symbols list of XML compressors features

Symbol	Description	Symbol	Description
G	General Text Compressor	**S**	Specific XML Compressor
D	Schema dependent Compressor	**I**	Schema Independent Compressor
A	Archival XML Compressor	**Q**	Queriable XML Compressor

Testing Corpus (Data Sets)

- Determining the XML files that should be used for evaluating the set of XML compression tools is not a simple task.
- The documents of our corpus are classified into four categories:
  - **Regular Documents**: Regular document structure and short data contents. They reflect the XML view of relational data. The data ratio of these documents is in the range of between 40% and 60%.
  - **Irregular documents**: Very deep, complex and irregular structure. More challenging in terms of compression efficiency.
  - **Textual documents**: Simple structure and high ratio of the contents is preserved to the data values. The ratio of the data contents of these documents represent more than 70% of the document size.
  - **Structural documents**: No data contents at all. 100% of each document size is preserved to its structure information. They are used to assess the claim of XML conscious compressors on using the well known structure of XML documents for achieving higher compression ratios on the structural parts of XML documents.

Testing Corpus (Data Sets)

Data Set Name	Document Name	Size (MB)	Tags	Number of Nodes	Depth	Data Ratio
EXI	Telecomp.xml	0.65	39	651398	7	0.48
	Weblog.xml	2.60	12	178419	3	0.31
	Invoice.xml	0.93	52	78377	7	0.57
	Array.xml	22.18	47	1168115	10	0.68
	Factbook.xml	4.12	199	104117	5	0.53
	Geographic Coordinates.xml	16.20	17	55	3	1
XMark	XMark1.xml	11.40	74	520546	12	0.74
	XMark2.xml	113.80	74	5167121	12	0.74
	XMark3.xml	571.75	74	25900899	12	0.74
XBench	DCSD-Small.xml	10.60	50	6190628	8	0.45
	DCSD-Normal.xml	105.60	50	6190628	8	0.45
	TCSD-Small.xml	10.95	24	831393	8	0.78
	TCSD-Normal.xml	106.25	24	8085816	8	0.78
Wikipedia	EnWikiNews.xml	71.09	20	2013778	5	0.91
	EnWikiQuote.xml	127.25	20	2672870	5	0.97
	EnWikiSource.xml	1036.66	20	13423014	5	0.98
	EnWikiVersity.xml	83.35	20	3333622	5	0.91
	EnWikTionary.xml	570.00	20	28656178	5	0.77
DBLP	DBLP.xml	130.72	32	4718588	5	0.58
U.S House	USHouse.xml	0.52	43	16963	16	0.77
SwissProt	SwissProt.xml	112.13	85	13917441	5	0.60
NASA	NASA.xml	24.45	61	2278447	8	0.66
Shakespeare	Shakespeare.xml	7.47	22	574156	7	0.64
Lineitem	Lineitem.xml	31.48	18	2045953	3	0.19
Mondial	Mondial.xml	1.75	23	147207	5	0.77
BaseBall	BaseBall.xml	0.65	46	57812	6	0.11
Treebank	Treebank.xml	84.06	250	10795711	36	0.70
Random	Random-R1.xml	14.20	100	1249997	28	0
	Random-R2.xml	53.90	200	3750002	34	0
	Random-R3.xml	97.85	300	7500017	30	0

Testing Environments

- To ensure the consistency of the performance behaviors of the evaluated XML compressors, we ran our experiments on two different environments.
- One environment with **high** computing resources and the other with considerably **limited** computing resources.

	High Resources Setup	**Limited Resources Setup**
OS	Ubuntu 7.10 (Linux 2.6.22 Kernel)	Ubuntu 7.10 (Linux 2.6.20 Kernel)
CPU	Intel Core 2 Duo E6850 3.00 GHz, FSB 1333MHz 4MB L2 Cache	Intel Pentium 4 2.66GHz, FSB 533MHz 512KB L2 Cache
HD	Seagate ST3250820AS - 250 GB	Western Digital WD400BB - 40 GB
RAM	4 GB	512 MB

Performance Criteria

We measure and compare the performance of the XML compression tools using the following metrics:

- **Compression Ratio**: represents the ratio between the sizes of compressed and uncompressed XML documents.

  Compression Ratio = (Compressed Size) / (Uncompressed Size)

- **Compression Time**: represents the elapsed time during the compression process.

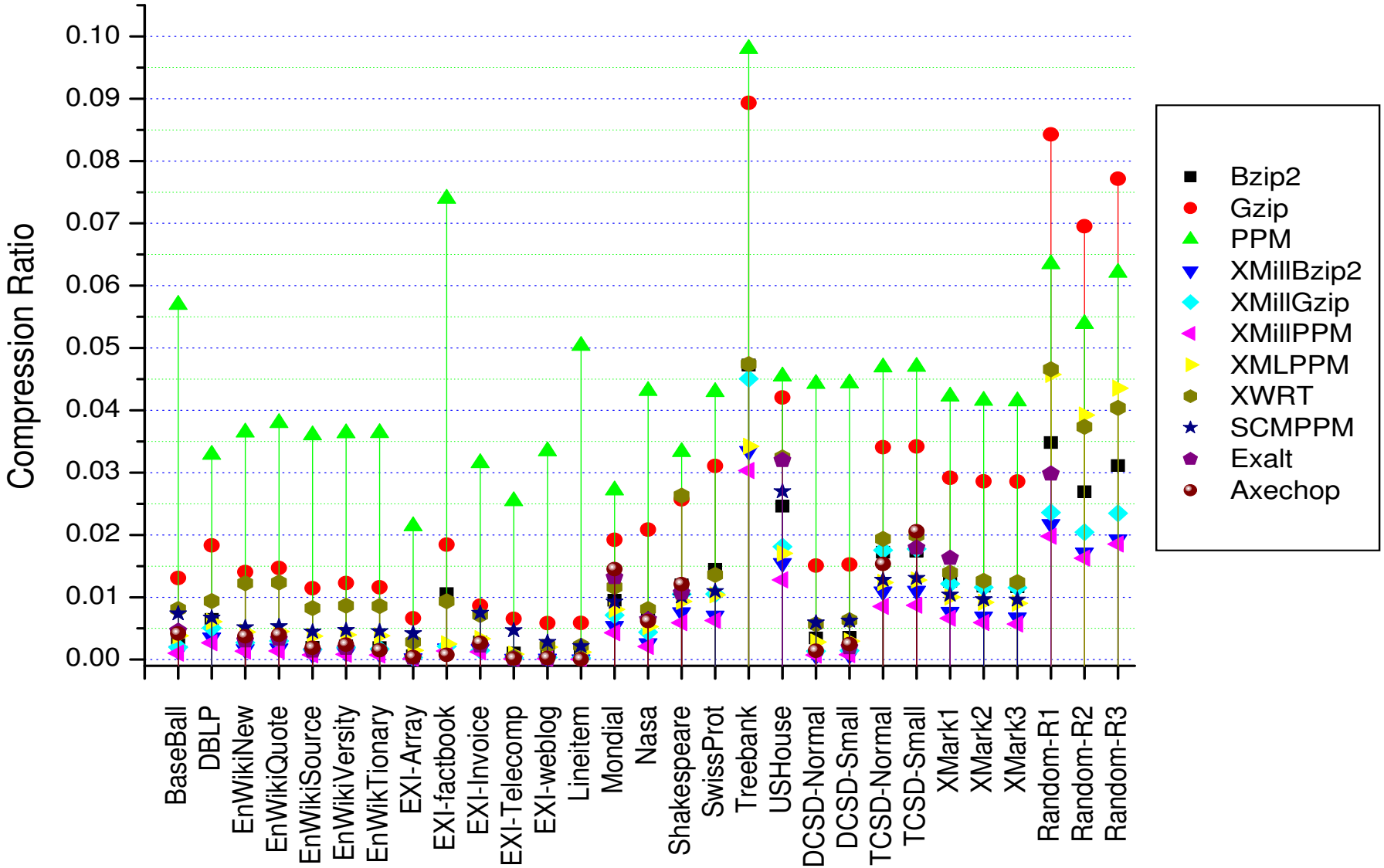
- **Decompression Time**: represents the elapsed time during the decompression process.

For all metrics: **the lower the metric value, the better the compressor**.

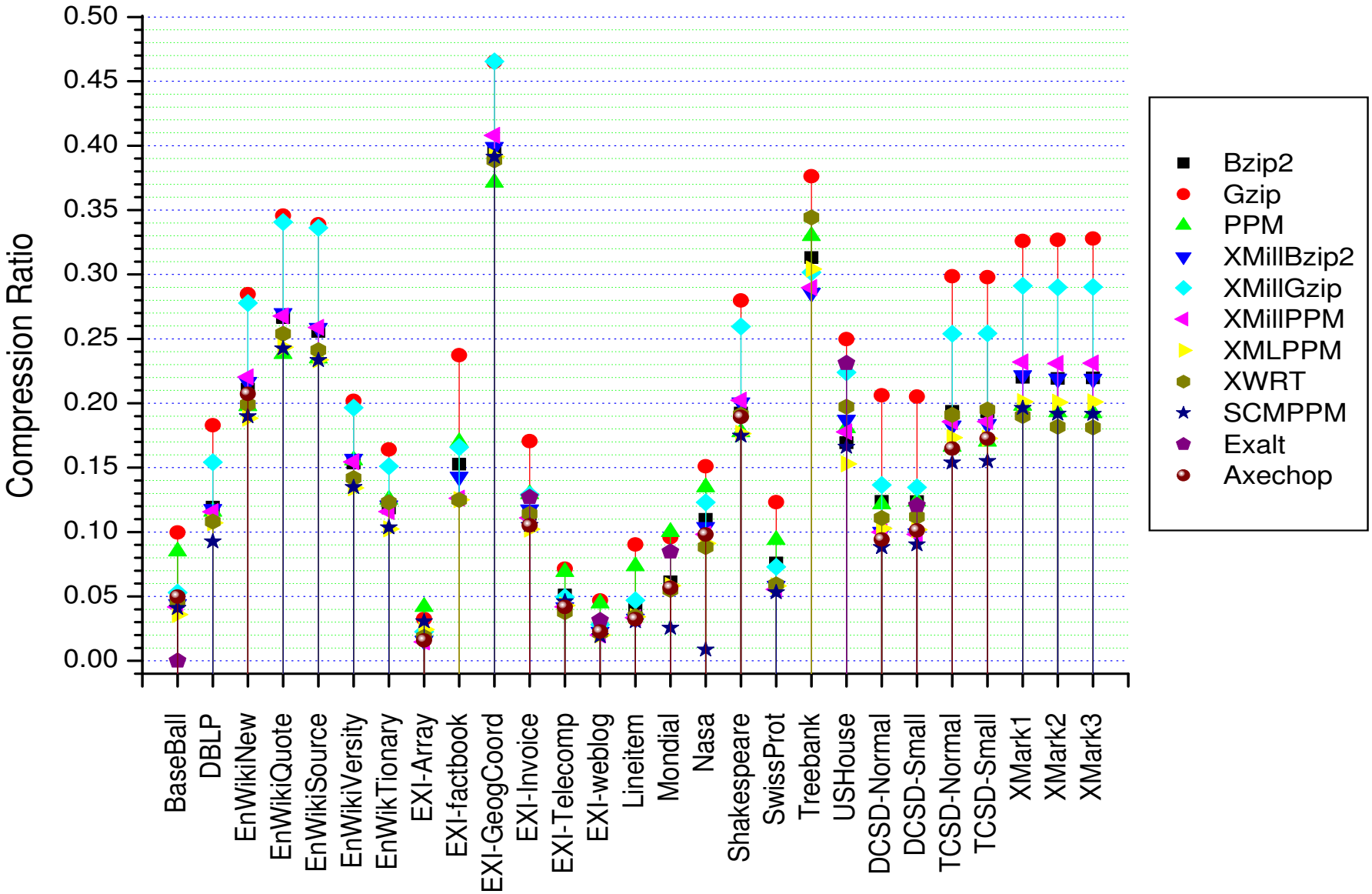
Experimental Framework

- We evaluated 11 XML compressors: 3 general purpose text compressors and 8 XML conscious compressors.
- Our corpus consists of 57 documents: 27 original documents, 27 structural copies and 3 randomly generated structural documents.
- We run the experiments on two different platforms.
- For each combination of an XML test document and an XML compressor, we run two different operations (compression - decompression).
- To ensure accuracy, all reported numbers for our time metrics are the average of five executions with the highest and the lowest values removed.
- We created our own mix of Unix shell and Perl scripts to run and collect the results of these huge number of runs.
- The web page of this study provides access to the test files, examined XML compressors and the detailed results of this study. http://xmlcompbench.sourceforge.net/.

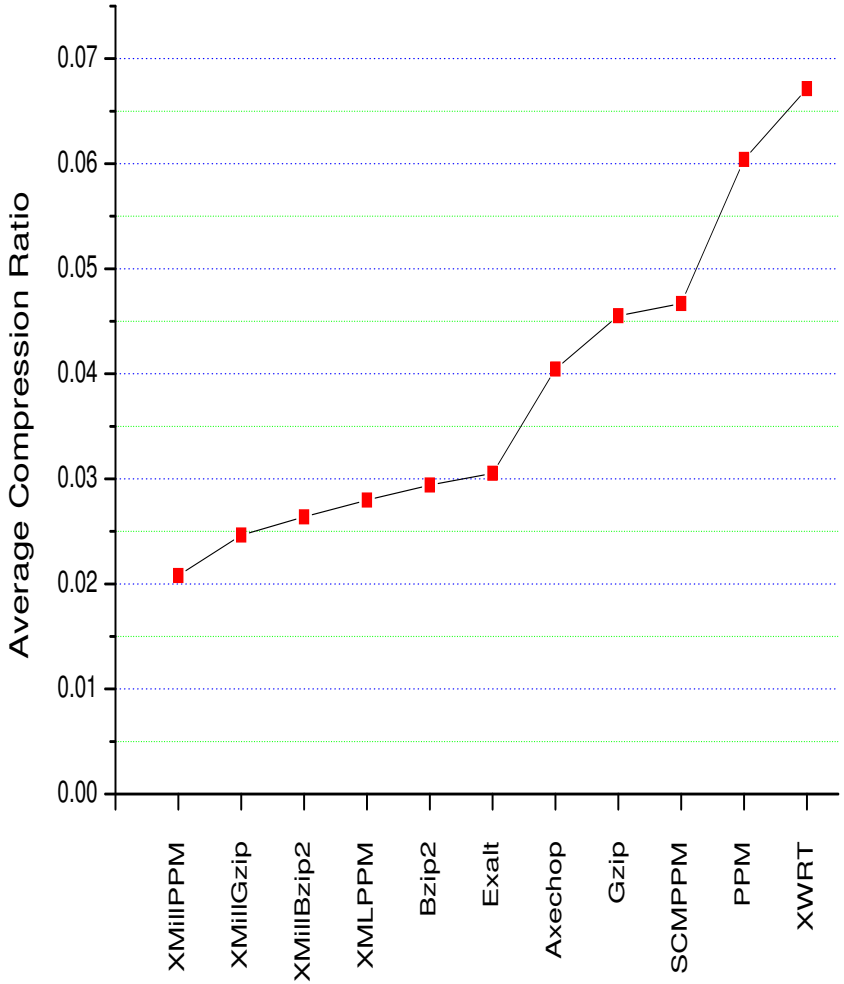
Experimental Results : Detailed Compression Ratios of *Structural* Documents



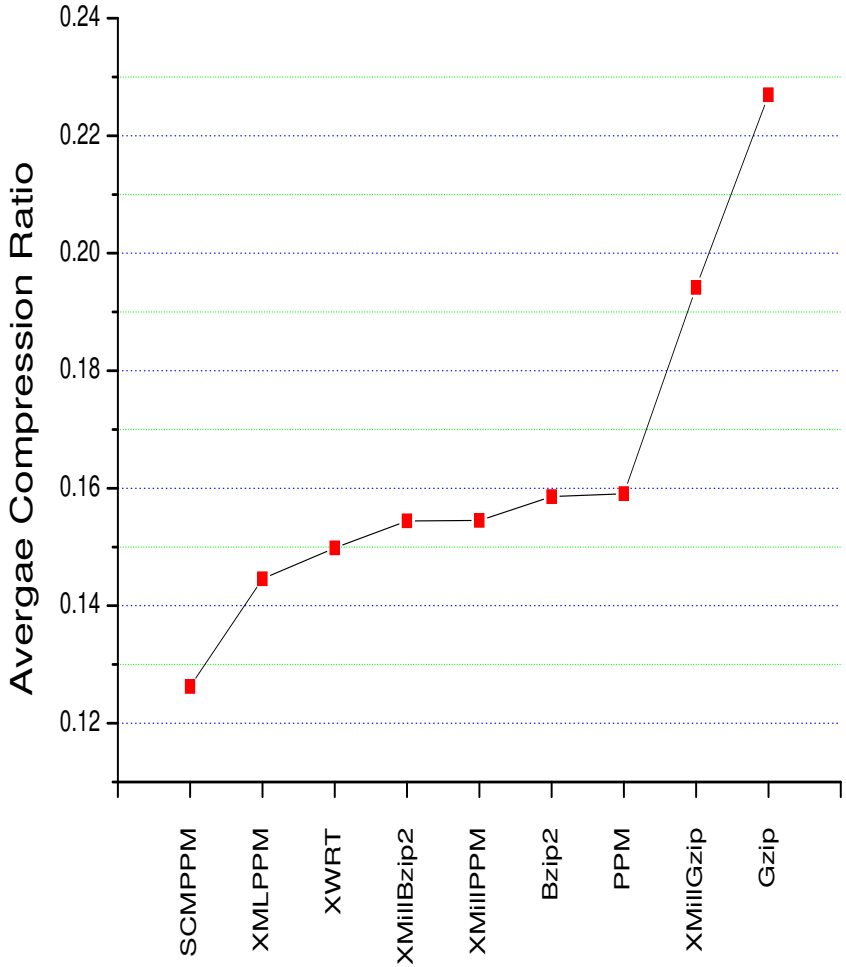
Experimental Results : Detailed Compression Ratios of *Original* Documents



Experimental Results : Average Compression Ratios

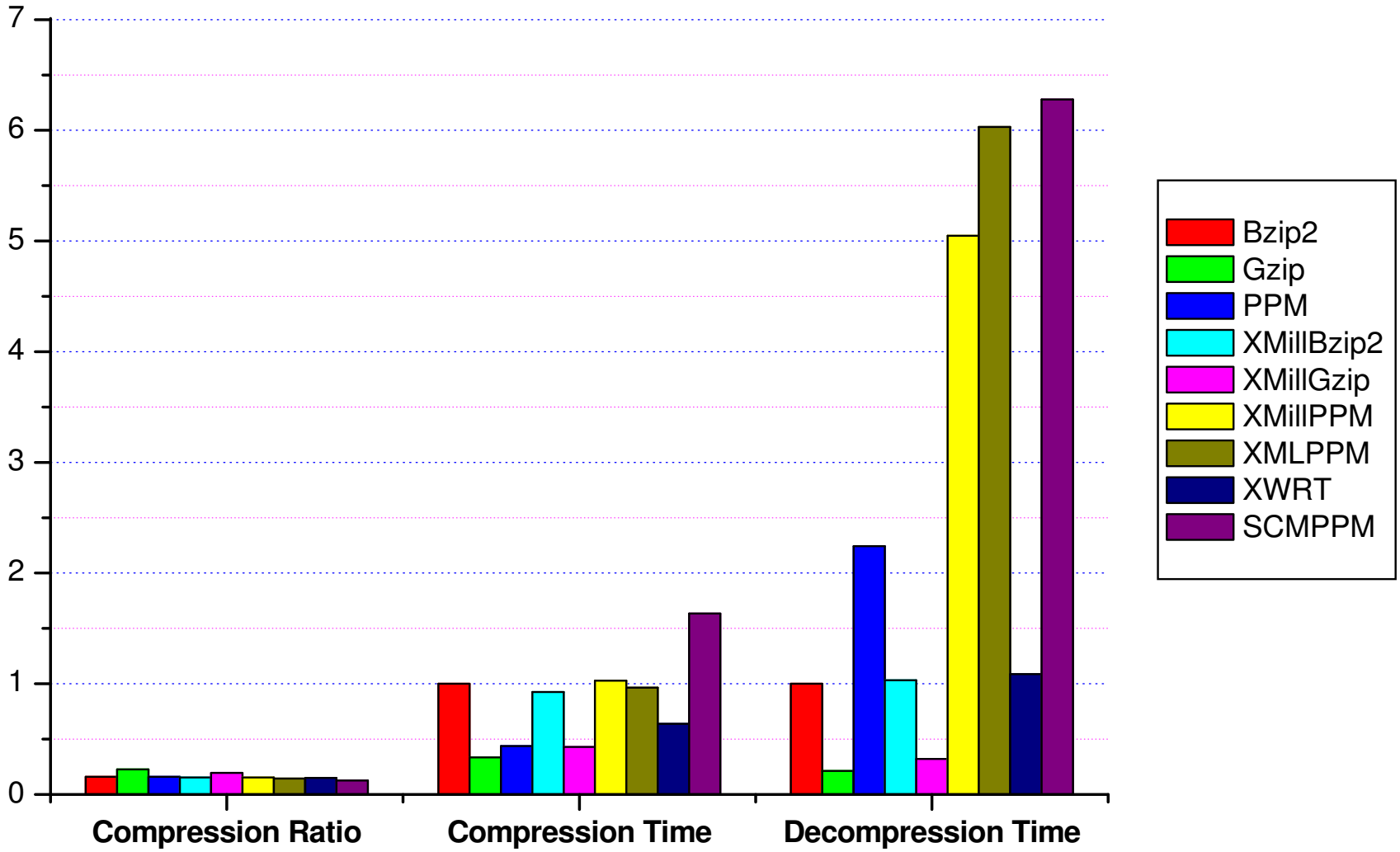


(a) Structural documents.



(b) Original documents.

Overall Performance of XML Compressors

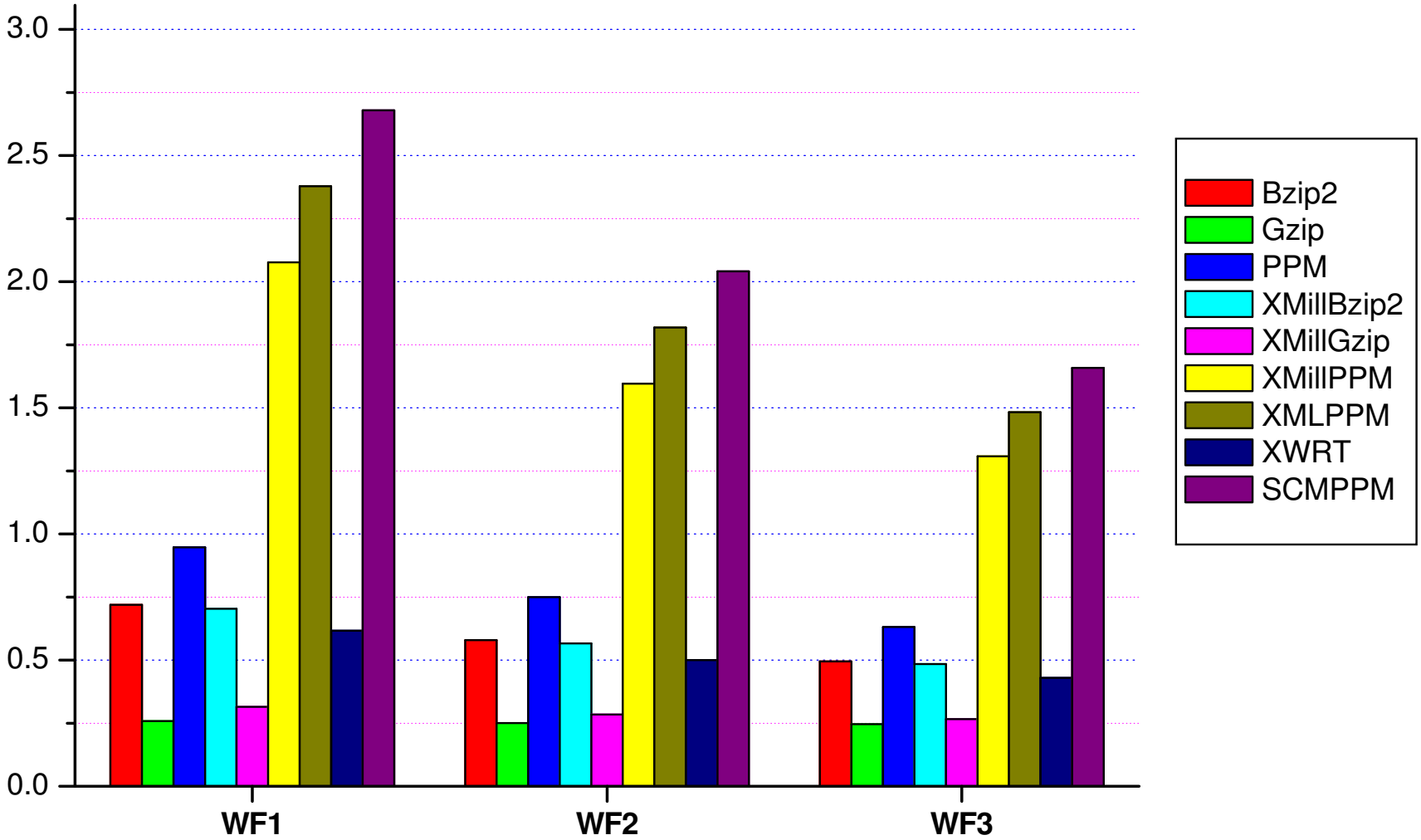


Proposed Ranking of XML Compressors

- The results of our experiments have not shown a *clear winner*.
- Different ranking methods and different weights for the factors could be used for this task. Deciding the weight of each metric is mainly dependant on the scenarios and requirements of the applications where these compression tools could be used.
- We used three ranking functions which give different weights for our performance metrics:
  - WF1 = (1/3 * CR) + (1/3 * CT) + (1/3 * DCT).
  - WF2 = (1/2 * CR) + (1/4 * CT) + (1/4 * DCT).
  - WF3 = (3/5 * CR) + (1/5 * CT) + (1/5 * DCT).

  where **CR** represents the compression ratio metric, **CT** represents the compression time metric and **DCT** represents the decompression time metric.

Proposed Ranking of XML Compressors



Conclusions

- The primary innovation in the XML compression mechanisms was introduced in XMill by separating the structural part of the XML document from the data part and then group the related data items into homogenous containers that can be compressed separably. Most of the following XML compressors have simulated this idea in different ways.
- The dominant practice in most of the XML compressors is to utilize the well-known structure of XML documents for applying a pre-processing encoding step and then forwarding the results of this step to general purpose compressors.
- There are no publicly available solid implementations for grammar-based XML compression techniques and queriable XML compressors.
- The authors of the XML compressors should provide more attention to provide the source code of their implementations available.

Conclusions

We believe that this paper could be valuable for both the **developers** of new XML compression tools and interested **users** as well.

- **For developers**, they can use the results of this paper to effectively decide on the points which can be improved in order to make an effective contribution.
  - We recommend tackling the area of developing stable efficient queriable XML compressors. Although there has been a lot of literature presented in this domain, we are still missing efficient, scalable and stable implementations in this domain.
- **For users**, this study could be helpful for making an effective decision to select the suitable compressor for their requirements.
  - For users with highest compression ratio requirement, the results of our experiments recommend the usage of either the PPM compressor with the highest level of compression parameter or the XWRT compressor with the highest level of compression parameter.
  - For users with fastest compression time and moderate compression ratio requirements, gzip and XMillGzip are considered to be the best choice.

The End

Thank You