Partial Query Processor For Compressed Xml

Vijay Gulhane[1], Dr. M. S. Ali [2]
Sipna COET, Amravati, Rpof. Ram Meghe COEM,Badnera

Abstract

Query processor plays a very important role in xml database systems. However, efficient work has been done to study query processing in XML database systems. Query to xml database refers to a information or data that search where data is located in xml source file. The basic components of any query processor is an indexing scheme .query processing strategies attempt to investigate a more effective auxiliary structure, such as an indexing scheme, to aid querying compressed XML databases. The queriable compressors are themselves strengthened to support efficient querying over compressed XML data. This means the goal here is an analytical model for querying compressed databases, which optimize the query engine of a compressor.

1. Introduction

Query processor palys an important role in the database management system. This paper first introduces the concept of query processing and Parsing, and takes the review of the parsering technologies for the XML documents. A new query processor is scheme based on LZ free algorithm and SAX parser approach . Next the Query processor, and to start an XML parse

From the XML compresser compressed XML document is loded in the loder of query processor . Depending on the type of query fired query engine processes the query. Timer keeps the the record of query processing required timing.

After evalution it was found that our approach gives improved performance , Query response time of XVSGC is better than the XGRIND and the more than Xqzip and Xqzip+ as it is a non quarable compressor. comparisions with the existing qurable Xml compressor, XVSGC achieves significantly improved query performance compared to Xpress also.

The compression in xml file and the indexing for query processor affects response time in two ways. First, before a compressor starts its execution, memory space has to be allocated to the process. These memory are used to store the execution code, copies of files, and any temporary objects produced which is intermediate step for the compressor and query processor. Second, some applications, such as DBLP complete dataset, have high demands on memory. Their executions will be significantly slowed down. The Approach use uses SAX interfaces and classes also Lample –Ziv to achieve a partial decompression.

2. Related Work

As shown in figure 2.1 there are few compressors which are qurable sence from 2000 to 2011.

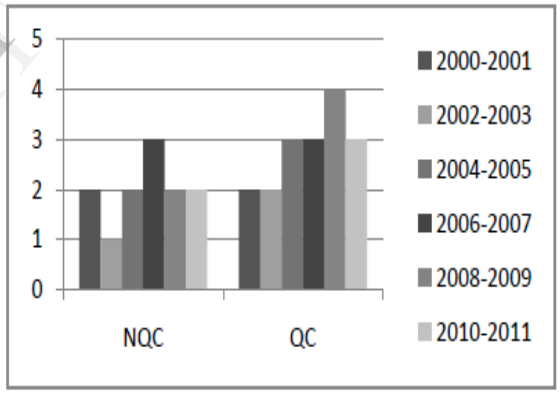

Figure:2.1: Distribution of Compressor over the year

A program or module that checks a well-formed syntax and provides a capability to manipulate XML data element. Navigate through the XML document .extract or query data elements Add/delete/modify data elements.

These SAX classes and interfaces fall into five groups:

1.interfaces implemented by the parser:Parser and AttributeList (required), and Locator (optional)

2.interfaces implemented by the application:DocumentHandler, ErrorHandler, DTDHandler, and EntityResolver (all optional: DocumentHandler will the most important one for typical XML applications)

3.standard SAX classes:InputSource, SAXException, SAXParseException, HandlerBase (these are all fully implemented by SAX)

4.optional Java-specific helper classes in the org.xml.sax.helpers package:ParserFactory, AttributeListImpl, and LocatorImpl (these are all fully implemented by the SAX Java distribution)

5.Java demonstration classes in the nul package:SystemIdDemo, ByteStreamDemo, CharacterStreamDemo, and EntityDemo, all of which can be run as Java applications; there is also a DemoHandler class that all four share

3. EFFECTIVE QUERY PROCESSING

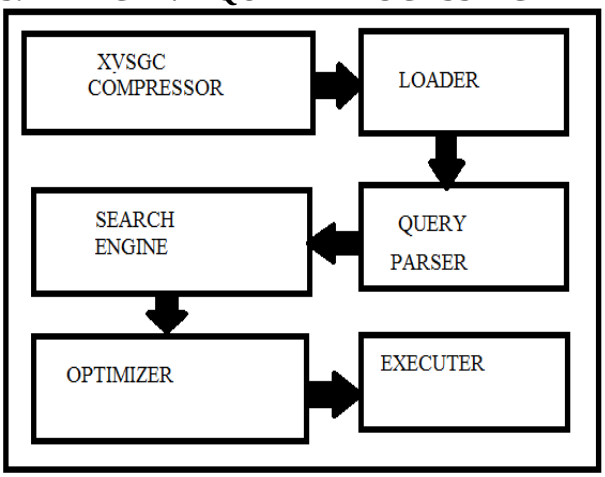

Figure: 2-2 Partal query processor

We base our work on the principle of LZF that XML compression partial query processing techniques (like operators,indexes, values for query optimization etc.can be used together when properly combined. This principle has been stated and forcefully validated in the domain of relational query processing [1],[3].Thus, it is important in the XML dataset for the partial efficient qury processing. Above figure shows the query processor . In the query processor we has the input from the xml compresser and qury processor loads the compressed file by the loder. This compressed file processes for the parsing and query by the user is handel by the Query engine . once qurey is obtain then data or values are serched by the search engine that is by the index values. In the optimization phase partial qurey is optimize and exicuts the results with find data or not found. In additional to this we are designed a timers for the both in the time format(HH:MM:SS:MS) viz at compressor side and one at query processig side. It contains the following modules:

1.The loader and compressor converts XML documents in acompressed , yet queryable format, using compression algorithms and the query work loader acess that file.2.The compressed repository stores the compressed documents and provides: (i) compressed **data**

3.The queryprocessor processes the compressed documents and provides: :(i) compressed data Elements

(ii) Values,evaluates queries over compressed documents, Allows For efficient evaluation over the compressed repository .

Besides the components mentioned in the figure, there are number of less import a helper components in our system. During the compression, a stream of data is produced. As can be seen in Figure2.3 the structure of the stream is very simple. It starts with a short header block which contains the identication of the XVSGC format and information about the compressed data. After the header block, the compressed data follows.

Figure :2.3 Structure of compressed File

3.2 OUR CORPUS BREAF

SwissPort

SWISS-PORT is a curated protein sequence database which strives to provide a high level of annotations (such as the description of the function of a protein, its domains structure, post-translational modifications, variants, etc.), a minimal level of redundancy and high level of integration with other databases.

Treebank (partially encrypted)

English sentences, tagged with parts of speech. The text nodes have been encrypted because they are copy written text from the Wall Street Journal. Never the less, the deep recursive structure of this data makes it an interesting case for experiments.

Mondial

World geographic database integrated from the CIA World Fact book, the International Atlas, and the TERRA database among other sources.

DBLP Computer Science Bibliography

The DBLP server provides bibliographic information on major computer science journals and proceedings. DBLP stands for Digital Bibliography Library Projects

Shakespeare :

A collection of the plays of William Shakespeare in XML [5].The first four datasets given above are regarded as data-centric as the XML documents have a very regular structure, whereas the last one is regarded as document-centric as the XML documents have a less regular structure: Yahoo and UWN. Table 3.1 Shows Corpus Set Details.

Table:3.1 Corpus Set Details

File Name	Description size	Elements	Attributes	Max Depth	Avg-Depth
Mondial	1 MB	22423	47423	5	3.59274
Swissport	109 MB	2977031	2189859	5	3.55671
Treebank	82 MB	2437666	1	36	7.87279
Dblp	127 MB	3332130	404276	6	2.90228
Yahoo	24KB	342	0	5	3.76608
UWN	2 MB	66729	6	5	3.95243
Ebay	34.7 KB	156	0	5	3.7564
Ubid	19.8 KB	342	0	5	3.7661
321gnoe	24.9 KB	311	0	5	3.7653

Testing Environment

The experiments were performed on a windows Intel® Core™i3 CPU @ 3.10 GHz 3.10 GHz with installed memory (RAM) 2.00 GB), 32 bit operating system. In the tests, the compressors were run under their default settings.

4. Experimental Evaluation:

The scheme of search engine, PARSER, proposed LZ base compressed query processing and studied for different load. The performance metrics use to mesure the performance of the query processor are minimum time taken to execute the result.
The time taken to compress documents is obtained by running the corresponding processes repeatedly three times and taking the average of the three runs. The main reason for doing this is to reduce the disk I/O influences on the results by loading the whole document into the physical memory. Calculations of CR1 and CR2 are done using the following formules-
The compression ratio is defined as follows:
CR1 =

$$\frac{\text{Size of compressed file X 8}}{\text{Size of Original file}}$$

bits/byte

CR2 =

$$\left(1 - \frac{\text{Size of compressed file}}{\text{Size of original file}}\right) \text{X } 100$$

Compression Ratio Factor (CRF):- Normalize the Compression Ratio of XVSGC with Respect to XMill and XGRIND

CRF 1=CRXVSGC/XMILL
CRF 2=CRXVSGC/XGRIND

Compression Time Factor (CRT):- Normalize the Compression Time of XVSGC with Respect to XMill and XGRIND.

CRT1=CRTXVSGC/XMILLCRT
2=CRTXVSGC/XGRIND

Query Response Time(QRT): Time Required T to execute the query
Following Table3.4 shows the results of compression CR1 and CR2 with the time period and average time over the different types of document as stated above.

3.4: Auction Dataset

Auction Data KB		CS	CR1	CR2	T1	T2	T3	TA
Yhoo	24.8	21.7	7	12.5	0.708	0.733	0.725	0.722
Ebay	*34.7*	*41.6*	*9.59078*	*-19.885*	*0.843*	*0.84*	*0.84*	*0.841*
Ubid	19.8	13.5	5.45455	31.8182	0.729	0.717	0.715	0.7203
321gnoe	24.9	22.5	7.22892	9.63855	0.729	0.722	0.72	0.7237

From the above table we can observe that Auction Dataset are tested for the time period . Range of variation is in between 0.720 to 0.841 The minimum time required for the UID and maximum time required for the EBay . Here time period is in milliseconds

The results of the evaluation for the different document are shown in following table. And

comparative graphs are shown.

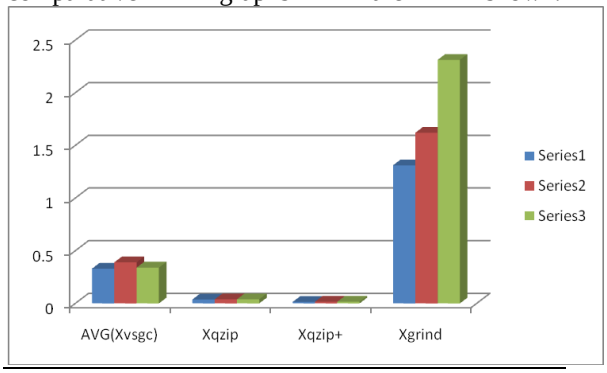

Figure: 4.1 Query Performance OF XVSGC With Others of Textual Documents

Above figure 4.1 shows the Query response time for the different compressor. From the above figure we can conclude that Query response time of XVSGC is better than the XGRIND and the more than Xqzip and Xqzip+ as it is a non quarable compressor.

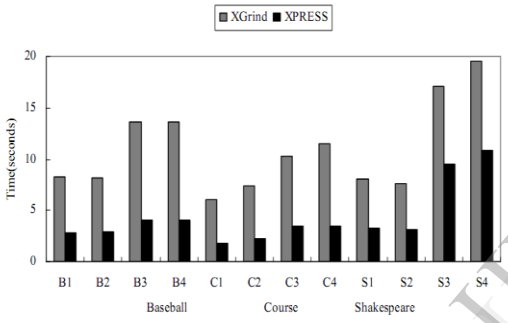

Figure 4.2 : Query Evaluation Time (From: XPRESS:AQueriableCompressionforXMLData by Jun-KiMinMyung-JaeParkChin-WanChung)

Query processor refers to a collective set of strategies that processes the queries in some sort of time. The basic component of any query processor is the parser and the Query engine. Query engine attempt to resolve the query. Search engine and the Parser is heart of the query processor. Search engine searches the exact match for the incoming queries and then it optimize it for the result.

After evalution it was found that our approach gives improved performance , Query response time of XVSGC is better than the XGRIND and the more than Xqzip and Xqzip+ as it is a non quarable compressor. comparisions with the existing qurable Xml compressor, XVSGC achieves significantly improved query performance compared to Xpress also.

5 References

[1]Abramson, N. 1963. Information Theory and Coding. McGraw-Hill, New York.

[2]AlHamadani, Baydaa "Retrieving Information from Compressed XML Documents According to Vague Queries" July, 2011 University of Huddersfield Repository http://eprints.hud.ac.uk/[3]

[3]Andrei Arion, Angela Bonifati, Ioana Manolescu, Andrea Pugliese "XQueC: A Query-Conscious Compressed XML Database" ACM Journal Name, Vol. , No. , 20, Pages 1–31.

[4]Andrei Arion1, Angela Bonifati2, Gianni Costa2, Sandra D'Aguanno1,etel "Efficient Query Evaluation over Compressed XML Data" E. Bertino et al. (Eds.): EDBT 2004, LNCS 2992, pp. 200–218, 2004. _c Springer-Verlag Berlin Heidelberg 2004

[5]Apostolico, A. and Fraenkel, A. S. 1985. Robust Transmission of Unbounded Strings Using Fibonacci Representations. Tech. Rep. CS85-14, Dept. of Appl. Math., The Weizmann Institute of Science, Rehovot, Sept.

[6]Augeri, C. J., Bulutoglu, D. A., Mullins, B. E., Baldwin, R. O. & Leemon C. Baird, I. (2007). An analysis of XML compression efficiency. Proceedings of the 2007 workshop on Experimental computer science, ACM, San Diego, California.

[7]Debra A. Lelewer and Daniel S. Hirschberg " Data Compression"

[8] David Salomon, Data Compression: The Complete Reference, pub-SV, 2004.

[9]Elias, P. 1987. Interval and Recency Rank Source Coding: Two On-Line Adaptive Variable-Length Schemes. IEEE Trans. Inform. Theory 33, 1 (Jan.), 3-10.

[10]Faller, N. 1973. An Adaptive System for Data Compression. Record of the 7th Asilomar Conf. on Circuits, Systems and Computers (Pacific Grove, Ca., Nov.), 593-597.

[11]G. Antoshenkov. Dictionary-Based Order-Preserving String Compression. VLDB Journal 6, page 26-39, (1997).

[12]Gllager, R. G. 1978. Variations on a Theme by Huffman. IEEE Trans. Inform. Theory 24, 6 (Nov.), 668-674.

[13]Gregory Leighton and Denilson Barbosa "Optimizing XML Compression (Extended Version)" arXiv:0905.4761v1 [cs.DB] 28 May 2009

[14]G. Cleary, I.H. Witten, Data compression using adaptive coding and partial string matching, IEEE Trans. Commun. OM-32 (4) (1984) 396–402.

[15]GZip Compressor, http://www.gzip.org/.

[16]H. Liefke and D. Suciu. XMill: An Efficient Compressor for XML Data. Proceedings of the ACM

SIGMOD International Conference on Management of Data, pp. 153-164 (2000).
[17]Horspool, R. N. and Cormack, G. V. 1987. A Locally Adaptive Data Compression Scheme. Commun. ACM 16, 2 (Sept.), 792-794.