Chapter 7

Schema Independent XML Compressor

Baydaa Al-Hamadani
University of Huddersfield, UK

Zhongyu (Joan) Lu
University of Huddersfield, UK

Raad F. Alwan
Philadelphia University, Jordan

ABSTRACT

XML has become the standard way for representing and transforming data over the World Wide Web. The problem with XML documents is that they have a very high ratio of redundancy, which makes these documents demanding a large storage capacity and large network band-width for transmission. This study designs a system for compressing and querying XML documents (XMLCQ) which compresses the XML document without the need to its schema or DTD to minimize the amount of technologies associated with these documents. XMLCQ first compressed the XML document by separating its data into containers according to the path of these data from the root to the leaf, then it compressed these containers using a back-end compression technique. The compressed file then could be retrieved with any kind of queries applied. Only the required information is decompressed and submitted to the user. Depending on several experiments, the query processor part of the system showed the ability to answer different kinds of queries ranging from simple exact match queries to complex ones. Furthermore, this paper introduced the idea of retrieving information from more than one compressed XML documents.

1. INTRODUCTION

The eXtensible Markup Language (XML) is a W3C standard which has been widely used by both the commercials and the researchers. In the recent years, we have witnessed an increasing volume of XML digital information that either created directly as an XML document or converted from another type of data representation. The importance of XML comes from different factors, its ability to represent different data types in one document, solving the problem of long-term accessibility, and to becoming the solutions to interoperability problem (Al-Hamadani et al., 2009).
Due to the replication of the XML schema in each record, XML document is considered to be one of the self-describing data files, which means that these kinds of files have a lot of data redundancy in both of its tags and attributes (Ray, 2001). For the above reason the need to compress XML documents becoming increasingly dramatic. Furthermore, an extensive need evolved to retrieve information directly from the compressed documents and then decompress only the retrieved information (Ferragina et al., 2006).

Because of the wide range of XML documents usage and with different kinds of users, there are many issues should be taken into consideration. One issue is to deal with all kinds of queries to cover the entire user’s needs from a querying system. It is also important to minimize the amount of technologies associated with XML documents, such as Schema and Document Type Definition (DTD), in order to ease the use of retrieving information from them.

Although the XML documents’ schema or their DTD are very important in the design of these documents, they add more complexity to the retrieving process from the user’s point of view. Moreover, sometimes the schema cannot be seen by the user or there is one schema that describes two or more different documents.

The remedy to the previous dilemmas is to design a compression technique that has the capability to compress the XML document without any reference to the schema or the DTD of this document. This compressor has the ability to retrieve information from the compressed version and decompress only the relevant information.

This study reveals a new technique called “XMLCQ” which consists of two stages. In the first stage, it separates the data part of the XML document into several containers according to the path of that data within the document. Then each of the containers is compressed separately using a back-end compressor. The compressed file is added into a database which can be used as a reference to get the compressed document in the retrieval stage. The second stage is to process different kinds of XPath queries. The compression ratio was compared with other queriable XML compressors and the XPath queries were tested against XPathMark benchmark.

Collecting the compressed files in a database gives an expansion to the XPath query, if the user does not know the exact XML document that contains the required information. Moreover, if the information is disseminated within several documents, then the user can take advantage of retrieving all the relevant information from all these documents.

2. RELATED WORKS

Recently, large numbers of XML compression techniques have been proposed. Each of which has different characteristics. This section discusses the differences between these compressors and their main features.

XML compressors can be classified into two classes either to be XML-blind or XML-conscious compressors. XML-blind or general purpose compressors deal with the XML document as a traditional text document ignoring its structure and apply the general purpose text compression techniques to compress them. These techniques can be classified into two main classes (Salomon, 2007), either to be statistical or dictionary based compressors (Augeri et al., 2007; Augeri, 2008). The statistical or the arithmetic compressors represent each string of characters using a fixed number of bits per character. PPM, CACM3, and PAQ are examples of this kind of compressors (Cleary & Witten, 1984; Moffat, 1990; Alistair et al., 1998). On the other hand, dictionary compression techniques substitute each string in the input by its reference in a dictionary maintained by the encoder. WinZip (http://www.winzip.com/), GZIP (http://www.gzip.org/), and BZIP2 (http://www.bzip.org/) are examples of this compression class.

On the other hand, XML-conscious compressors try to utilize the structural behaviour of XML documents in order to achieve better compression.