Comparison between Range-based and Prefix Dewey Encoding

Ebtesam Taktek, Dhavalkumar Thakker and Daniel Neagu
School of Electrical Engineering and Computer Science, Faculty of Engineering & Informatics, University of Bradford, Bradford, U.K.

Keywords: XML Labelling, Range based Encoding, Prefix Encoding, XML Compression.

Abstract: XML is an increasingly important area in the field of data representation and communication over the web. XML data labelling plays an important role in management of XML data since it allows locating the XML content uniquely in order to improve the query performance. This paper focuses on two schemes for labelling native XML databases where the data is represented as ordered XML trees and contain relationships between nodes. We present a comparison between range-based and prefix encoding with focus on achieving labelling time and memory size. In our proposed approach, we employ UTF-8, UTF-16, UTF-23 encoding and decoding for both the labelling schemes.

1 INTRODUCTION

XML data has become one of the most important issues in the field of databases. Existing research have been conducted to improve the storing, retrieving and querying of XML data (Tatarinov et al. 2002). The main approaches for facilitating query processing based on native XML databases are structural indexing and labelling scheme. Labelling schemes focuses on assigning a unique code to each node in XML trees as an encoding for the documents to reduces the query processing time (Subramaniam et al. 2014; Zadjali and North 2016; Alsubai and North 2017). However, one of the criticism of most of the encoding techniques is that they contain large label size (Yu et al. 2005; Subramaniam and Haw 2014; Liu and Zhang 2016).

There is no existing evaluation framework that can be used by XML labelling schemes to provide a comprehensive analysis of these two schemes

Our proposed technique compares Range-based encoding and Prefix Dewey encoding in order to achieve the fastest labelling time and to ensure the generation of short labels in term of memory. Therefore, we control the bits subsequent of the label value using UTF-8, UTF-16 and UTF-23 in terms of encoding/ decoding time.

The remaining sections of this paper is organized as follows: Section 2 presents the existing related work in this area while Section 3 describes the Dewey encoding and the Range-based encoding methods. In section 4 the experimental results and evaluation are discussed while Section 5 concludes the paper and recommends a future research direction.

2 RELATED WORK

There are different labelling schemes that have been proposed for efficiently processing native XML databases. This section reviews and address issues related to different labelling schemes.

(Tatarinov et al. 2002) have proposed a method called Local Order Encoding scheme, each node is assigned an integer number, which represents its relation position among its siblings. It is appropriate to reconstruct document order. The advantage is that it does not result in large label sizes and therefore each label has a fixed length, which is one byte for each node and uses UTF-8-character encoding scheme. However, fixed length in labelling is leading to overflow problems. Although, the local encoding does not support all kinds of structural relationship queries, such as to determine the relationship between the following and preceding nodes. Another advantage compared to other techniques is that, it has a low relabelling, which allows the following siblings of the new node to be relabelled.

In addition, (Tatarinov et al. 2002) have proposed Dewey encoding scheme for labelling XML trees based on Dewey decimal classification system, it is one of the prefix labelling scheme. In this method,

Taktek, E., Thakker, D. and Neagu, D.
Comparison between Range-based and Prefix Dewey Encoding.
DOI: 10.5220/0007229003640368
In *Proceedings of the 14th International Conference on Web Information Systems and Technologies (WEBIST 2018)*, pages 364-368
ISBN: 978-989-758-324-7
Copyright © 2018 by SCITEPRESS – Science and Technology Publications, Lda. All rights reserved

each label is presented as an integer number and delimiter ".", the delimiter is encoded and stored separately from the label value (Li et al. 2008). Each node (u) is labelled as a combination of its parent label and postfix integer number (x_i). If u is the x^{th}child of s in XML tree then label of u, label (u) is concatenation of label of s and x which is presented as label$(s).x$, where s is the parent of u. For example, if element label for u is 2.5.3 then its 4^{th} child label will be 2.5.3.4. If an element label is 5.1.3.1 then its parent label is 5.1.3, its first ancestor label is 5.1. The advantage of this method is that for any element label, we can easily extract node labels of its ancestors and determine the relationship between nodes. However, the drawback of Dewey scheme is not appropriate for dynamic XML data; inserting a new sibling node into XML tree using Dewey labelling scheme requires relabelling all its right sibling nodes along with their descendants and this has produced a large label size at the cost of extra storage.

(Xu et al. 2009) have proposed Dynamic Dewey encoding scheme (DDE), which is an update of Dewey encoding scheme to transform the original Dewey into a fully dynamic labelling scheme. The advantage of the DDE is that, the label has different length; starting with a byte for the first level and increases in depth in relation to the level value. So that can be appropriate for avoiding overflow problems. In addition, it has the ability to avoid re-labelling completely and support high query performance. The main drawback is that, a large label size. Especially when the depth increases, and frequent insertions occur between two siblings by applying the midpoint technique.

(Kobayashi et al. 2005) have proposed VLEI encoding scheme. VLEI scheme is applied to XML labeling. The data type is binary string. The VLEI encoding has used number 9 for the identifier. For example, when a child node is inserted, the label for the node becomes the label of its parent node + 9 + VLEI code. However, VLEI encoding used eight bytes for the VLEI code. The VLEI main drawback is that lead to overflow problem especially with skewed insertion. t is the new VLEI sequence code.

$$t = 1\,.\,\{0|1\}^*$$

$$\text{If } t.0\,.\{0|1\}^* < t < t.1\,.\{0|1\}^*$$

For example,

$$10 < 1 < 11\ and\ 100 < 10 < 101 < 1 < 110 < 11 < 111.$$

The authors in (Zhang et al. 2001) used Range based labelling scheme which aims to determine the structural relationships between nodes by using the related containment information. Each label is represented as a 3-tuple and has fixed-length. Interval scheme does not result in large label sizes but lead to overflow problems. *start, end* and *depth* are to identify exactly the position of an element. *start* is generated by a pre-order traversal of the document trees exactly finds the occurrence position. While *end* is the maximal start of elements in the sub-tree of current element and *depth* gives additional information to determine the parent-child relationship.

In summary, following from the related work, the main drawback we have identified in the existing labelling schemes is the growth of the label sizes in response to that we present a comparison between two schemes with focus on achieving labelling time and memory size. Our work compares Range-based encoding and Dewey encoding to ensure the generation of short labels size and to control the bits subsequent of the label value using UTF-8, UTF-16, UTF-23 in terms of encoding or decoding time and to achieve the fastest labelling time, ultimately impacting the query performance.

3 COMPARISONS BETWEEN DEWEY ENCODING AND THE RANGE-BASED ENCODING SCHEMES

In the Dynamic Dewey encoding scheme, each label has a different length; starting with a byte for the first level and it increases in relation to the level value. The length of labels can vary widely depending on the nodes position within the XML tree. However, prefix labels naturally extend when XML data is updated during frequent insertions, causing overflow problems. However, in the Local Order Encoding scheme**,** each node is assigned an integer number and each label has a fixed length label; which is one byte for each node and used UTF-8-character encoding (Yergeau 2003). Furthermore, in Dewey encoding, each label presented as combination of its parent label and postfix integer number by delimiter ".", the delimiter is encoded and stored separately from the label value (Li et al. 2008).

In contrast, in Range based labelling scheme, each label presented as combination of the *start, end, depth* values using "," as a delimiter. Furthermore, in Quaternary encoding QED (Li and Ling 2005) and SCOOTER encoding (O'Connor and Roantree 2012)

proposed different delimiter storage scheme; they used number "0" as delimiters and consequently, these schemes increase the decoding time because of the extra comparison operation to identify the 0 whether a bit or a delimiter.

In our proposed approach, we employed UTF-8, UTF-16, UTF-23 encoding and decoding (Yergeau 2003) for both Range based scheme and Dewey labelling scheme. We have done this to encode the bits subsequent of the label value and then to aid the generation of short label size to achieve the fastest labelling time.

4 EXPERIMENTAL WORK AND RESULTS

All experiments were run on an intel Core i7 processor with 8GB of main memory and 64-bit Operating System, running Windows 10 system. We run Range-based algorithm and Dewey labelling algorithm in Java IDE 8.2. We used UTF-8, UTF-16, UTF-23 to control the bits subsequent of the label value and to specify the position of a label value within a specific-level interval, the experiments evaluated the scheme's performance in terms of the label size.

We carried out the experimental process on different datasets (Miklau 2015). The XML datasets represent various features of XML trees such as number of nodes, file sizes, maximum depth, the degree of fan-out. The real datasets we have used are DBLP, TreeBank and Nasa. Table1 below gives the properties of these datasets and summarises their characteristics.

Table 1: Benchmarks datasets.

XML dataset	File size	Max depth	Max breadth	Number of nodes
TreeBank	82 MB	36	144493	2437666
DBLP	127 MB	6	328858	3332130
Nass	23 MB	8	80396	476646

4.1 Experimental Evaluation

The label initialisation experiment for both Dewey labelling and Range-based were implemented successfully. As the focus here is the comparative of fastest labelling initial time and generation of short labels in term of memory. The outcome of this experiment was also aimed to compare the labelling size based on UTF-8, UTF-16 and UTF-23. This experiment was intended to evaluate the Dewey labelling against Range-based schemes, the results showed that the experiment met its objective.

This experiment examined two parameters: initial labelling time and the total label size. The statistical analysis of the results in figure 1,2 and 5 presented that there was a significance difference between the two schemes. Dewey generated initial labels faster than Range-based scheme see. In addition, when the schemes were applied to a large XML dataset of size 127MB, the performance between the schemes were significant, by up to 50%.

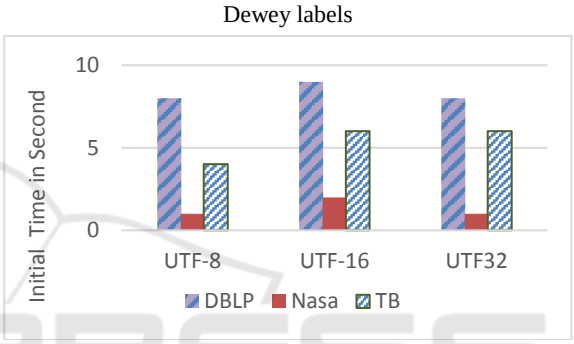

Figure 1: Initial labelling time for Dewey labelling scheme.

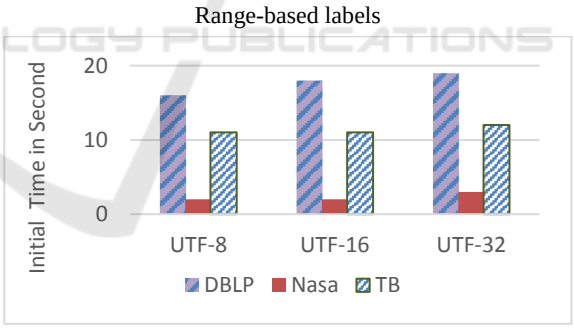

Figure 2: Initial labelling time for Range-based scheme.

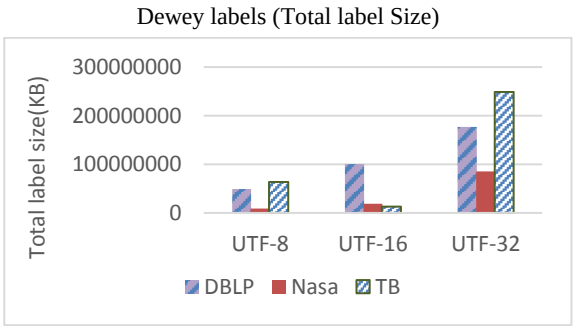

Figure 3: Total label Size (KB) for Dewey labels.

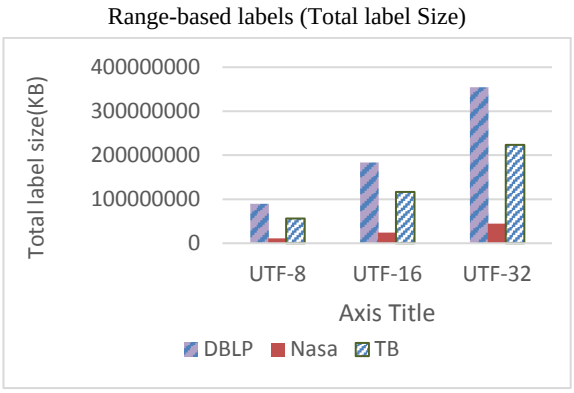

Figure 4: Total label Size (KB) for Range-based scheme.

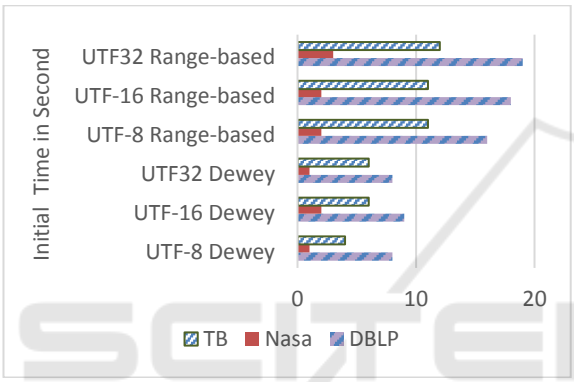

Figure 5: Initial labelling time for Dewey labelling and Range-based schemes.

In terms of controlling the bits subsequent of the label value using UTF-8, UTF-16 and UTF-23, The significance of the two schemes were justified in figure 3, 4 and 6. The overall label size for Dewey has a smaller label value in comparison to the Range based coding methods, except the in UTF-32 test, it has shown that Range-based scheme was performed better than Dewey scheme in Nasa and TreeBank datasets.

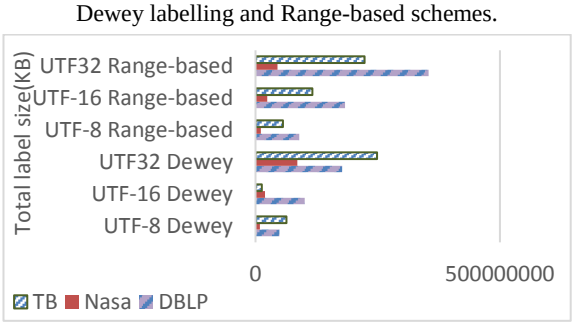

Figure 6: Total label Size (KB) for Dewey labelling and Range-based schemes.

5 CONCLUSION

This paper compares two XML labelling schemes, namely Range-based encoding and prefix encoding. The work was aimed to achieve the fastest labelling time and to ensure the generation of short labels in term of memory size and also to control the bits subsequent of the label value using UTF-8, UTF-16 and UTF-23 in terms of encoding/ decoding time. The overall label size for Dewey has a smaller label value in comparison to the Range based coding methods, except the in UTF-32 test, it has shown that Range-based scheme was performed better than Dewey scheme in Nasa and TreeBank datasets. In addition, when the schemes were applied to a large XML dataset of size 127MB, the performance between the schemes were significant, by up to 50%. In the future, we will generate an improved labelling scheme using an advance technique in comparison to the existing approaches. We will also apply other advanced encoding/decoding methods such as Elias-Fabonacci code to compare the performance of Range-based encoding and prefix encoding schemes in order to achieve the fastest labelling initial time and to ensure the generation of short labels in term of memory size.

ACKNOWLEDGEMENT

The authors would like to thank the anonymous reviewers for their helpful reviews in improving the quality of the paper.

REFERENCES

Alsubai, S. and North, S. D. (2017) A prime number approach to matching an XML twig pattern including parent-child edges. *2017. SCITEPRESS.*

Kobayashi, K., Liang, W., Kobayashi, D., Watanabe, A. and Yokota, H. (2005) VLEI code: An efficient labeling method for handling XML documents in an RDB. *2005. IEEE.*

Li, C. and Ling, T. W. (2005) QED: a novel quaternary encoding to completely avoid re-labeling in XML updates. *2005. ACM.*

Li, C., Ling, T. W. and Hu, M. (2008) Efficient updates in dynamic XML data: from binary string to quaternary string. *The VLDB Journal—The International Journal on Very Large Data Bases* 17 (3), 573-601.

Liu, J. and Zhang, X. X. (2016) Dynamic labeling scheme for XML updates. *Knowledge-Based Systems* 106, 135-149.

SCITEPRESS SCIENCE AND TECHNOLOGY PUBLICATIONS

Miklau, G. (2015) *Xml Data Repository. http://www.cs.washington.edu/research/xmldatasets/ [Online]. [Accessed August 2018]*.

O'Connor, M. F. and Roantree, M. (2012) SCOOTER: a compact and scalable dynamic labeling scheme for XML updates. *2012. Springer*.

Subramaniam, S. and Haw, S.-C. (2014) ME labeling: A robust hybrid scheme for dynamic update in XML databases. *2014. IEEE*.

Subramaniam, S., Haw, S.-C. and Soon, L.-K. (2014) Relab: a subtree based labeling scheme for efficient XML query processing. *2014. IEEE*.

Tatarinov, I., Viglas, S. D., Beyer, K., Shanmugasundaram, J., Shekita, E. and Zhang, C. (2002) Storing and querying ordered XML using a relational database system. *2002. ACM*.

Xu, L., Ling, T. W., Wu, H. and Bao, Z. (2009) DDE: from dewey to a fully dynamic XML labeling scheme. *2009. ACM*.

Yergeau, F. (2003) *UTF-8, a transformation format of ISO 10646*. 2070-1721.

Yu, J. X., Luo, D., Meng, X. and Lu, H. (2005) Dynamically updating XML data: numbering scheme revisited. *World Wide Web* 8 (1), 5-26.

Zadjali, H. and North, S. D. (2016) XML Labels Compression using Prefix-encodings. *2016. SCITEPRESS, Science and Technology Publications*.

Zhang, C., Naughton, J., DeWitt, D., Luo, Q. and Lohman, G. (2001) On supporting containment queries in relational database management systems. *2001. Vol. 30. ACM*.

SCITEPRESS
SCIENCE AND TECHNOLOGY PUBLICATIONS