Indian Journal of Science and Technology, Vol 9(40), DOI: 10.17485/ijst/2016/v9i40/103278, October 2016
ISSN (Print) : 0974-6846
ISSN (Online) : 0974-5645

# XML Indexing Techniques for Handling Large Amounts of Data

**Chang Kyun Pyo[1], Seok Jun Yun[2] and Gab Sang Ryu[3*]**

[1]Korea Information and Communication Industry Institute, Seoul, Republic of Korea; capyo@naver.com
[2]Republic of Korea Army, Republic of Korea; talky@hanmail.net
[3*]Department of Computer, Dong Shin University, Republic of Korea; gsryu@dsu.ac.kr

## Abstract

**Objectives:** This paper presents an effective XML indexing technology for input/output process and retrieval of large amounts of data to exchange structured documents in the Internet space. **Methods/Statistical Analysis:** In these days, many data are produce in various forms by Web media, the Internet of Things, and social media. However, the data are too complicated in type and large to effectively analyze, search, and rapid input/output processing with the traditional Relational DB system. In this paper, we have looked to measure the document searching speed, document-writing speed in order to assess the performance of the XML Information Retrieval and management system using an index model. Many cases contain hundreds of millions of different elements of an XML document. In Keyword-based approach, it is necessary for all the elements in the document tagging to a quick search because it does not contain the structural information for the XML document to a query. **Findings:** The XML is a widely used mark-up language in Internet, and a large amount of data is produce in XML format. In this paper, we propose two search methods for efficient search in a large amount of data with the XML indexing technology. First, existing studies show the true performance to lean to one side in the keyword search of large XML documents have a problem. Both informal and formal data processing, the Content Model was proposes to solve. Non-structured data to a right processing and utilization as NoSQL View in a variety of devices for rapid processing of the user interface was couple to Data. N-Screen or the like can be used without Viewer from various mobile devices, as well as the task of processing that is used in a large variety of services can be advantageous. **Improvements:** This work can also extended with other searching algorithms for handling large amount of data

**Keywords:** Big-Data, Indexing, I/O processing, XML

## 1. Introduction

XML (extensible markup language) is a standard format for exchanging structured documents and data in the Web environment in 1998 (W3C World Wide Web Consortium). XML has been adopted as a standard of structured data representation and document exchange on the Internet based on the advantages of excellent scalability and reusability therefore a number of areas such as academic materials, e-commerce, e-books are utilized. XML can provide user to exchange data with other systems by directly maintaining the semantic structure and defining tags. XML uses the Unicode to express a variety of characters, while not depending on the language. With features such as different platform availability, advantageous exchange of information between systems, XML has interoperability to be reuse in other applications. XML is use in the Internet standard documents, electronic publishing, medicine, management, law, electronic library, e-commerce, and so on. With the amount of the document in XML format increasing, the large-capacity and high-speed search has become an essential element. In an XML document, an element is the basic unit and hierarchy that structured in the form of a tree. Therefore, for the search of an XML document, which includes a specific path in the tree, the processing of these paths plays an important role in increasing the efficiency of the search. Data type is complicated and large, it can cause

*Author for correspondence

problems for efficient searching and data processing. In this paper, in order to input and output processing and retrieval of large amounts of data effectively, we present a proposal of indexing technology query utilizing XML.

## 2. Search Method

XML that has been adopts, as a standard of structured data representation and document exchange on the Internet is easy to use, based on the advantages of excellent scalability and reusability, e-commerce, e-books, a number of areas such as academic materials in are utilize. First, the structural based approach is a method for the search by using a representation of quality, such as XPath[1] and XQuery[2] in a manner, which is based hierarchical structure of an XML document. This method, the user is capable of use needs to know how the hierarchy of the XML document (XML DTD or XML Schema). On the other hand, Tag-based approach, users and structural information of the document, to be able to easily search for the required information without prior learning of complex query language. Thus, a search for large XML documents, since it is difficult to search know all the structures of complex XML documents, Tag-based approaches have been used more than the structure-based approach. Generally, XML Tag search method will return the minimum of a common ancestor, including all of the quality keyword (LCA, Lowest Common Ancestor[3]) as a search result. The basic unit of the index is the element in the XML document in Figure 1 shown an example look for the lowest common ancestor in the keyword search method. For example, subtree a user with root a "proceeding" and the keyword "AA" in the document is to be in the search results.

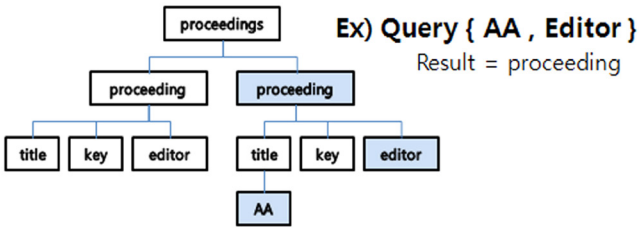



**Figure 1.** Examples of searching LCA

## 3. Feature of XML Information Retrieval

In the XML information retrieval, in general, it based on the hierarchical structure of the XML document. Since the well-defined hierarchical structure presented in the document, as well as the contents of the document, it is possible to perform a search on the structure. To use the XML query expression such as XPath[4] and XQuery[5] in order to clarify the search in this way has been subdivided, such queries, the user hierarchy on the XML document (XML DTD or XML it is available only when you know the schema) to some extent. Part that could contribute to the improvement of the search performance of the query XML document of such expressive structure has been well known how immature and relevant information about the user to represent the structure search. If you want to see actually its effect by a non-professional of data that do not know whether the find in a structure is rare. More recently, tips structure that contains a query is also published findings that do not improve search accuracy[6]. Nevertheless, most users, the search results rather than the entire XML document when viewing, wants to view only a few key regions constituting the document[7,8]. Users think that the element is more useful for their action, because it is easy to find useful information in a specific element than the full document[9,10]. Therefore, XML information retrieval systems, it is necessary for us to browse and search for elements suitable to the query from the XML document. To develop an XML information retrieval system for such complex user requirements, it will face many difficulties. Hierarchically well structured elements including some information meaningful, important functions of the document collection and not a function of the user's query. Therefore, the administrator can be constructed a user-friendly system to search easily what meaningful XML document is a suitable substitute for complex query language. So by using the keyword has been made various studies to find the XML document structurally[11-14]. Although structural information in most of the research on XML information retrieval is represented by element, using the value of the attribute of the element, since there is a case to describe the structure of the document, XML information retrieval system, taking into account also these attributes us to be[15]. XML document is increased there has been a lot of discussion about how to handle the XML document. In particular, how to save the XML document it may be classify to vary depending on the point of view, but in this paper, is to look to see depending on how you want to store XML documents in the database. When viewed from this perspective, a large split save (decomposition) method and the virtual division there is a (virtual fragmentation) storage system[16].

## 4. XML Query

Studies on the query processing of XML repository is actively being made. Lorel, XMLQL, XQuery, XPath in a variety of ways, such as XML query language is proposed. These query terms are and have all the characteristics of the path query (Query path), can be represented as a regular expression (regular expression). Path query in the document shown in Figure 2, the meaning of "/ db // food / name" is meant to retrieve the name of all the food that the 'db' in the document root to the top. This technique is to build the index graph for all paths that can occur in a separate XML data structure. For processing the search query paths after the original data graph as navigation route index by reducing the size of the search space techniques to reduce the query processing cost. Figure 3 is a representation of the Tag-based indexing structured XML to facilitate the search and input, modify structure. Figure 4 presents the case that you add a schema structured XML to modify the schema Tag-based search and can see the ease than the existing R-DB.

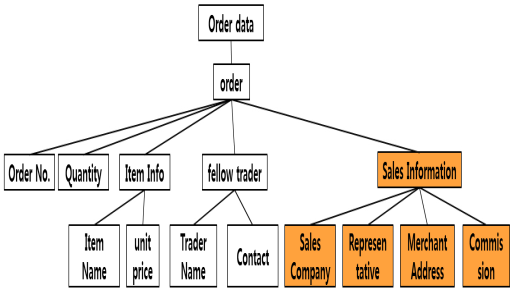

**Figure 2.** Path query


**Figure 3.** Examples of tagging

## 5. Results

### 5.1 Test Result

In this paper, we have looked to measure the document searching speed, document writing speed in order to assess the performance of the XML Information Retrieval and management system using an index model. There are many cases that contain hundreds of millions of different elements of an XML document. In Keyword-based approach, it is necessary for all the elements in the document Tagging to a quick search because it does not contain the structural information for the XML document to a query. If the keyword-based approach for the Tagging Tag in order to deal with large XML documents using Figure 5, and compared the results for a search rate Figure 6.




```
<Data Buy>
 <Order>
  <Order Number> AAAA-999-X </ order number>
  <Order Number> 57 </ Quantity>
  <Product>
   <Name> Electronic Modules </ Name>
   <Price> 35000 </ price>
  </ Products>
 <purchase information>
  <Contact name> Yamada production </ contact name>
  <Dealer info> 000-0000-0000 </ Dealer info>
 </ purchase information>
 <sales information>
    <vendor name> Gimpo Logistics</ Vendor name>
    <salesperson name> Joe's </ salesperson name>
    <Distributor address> miyashita@miyabussan.xx </ distributor Address>
    <Commission> 5 </ vendor>
 </ Sale Information>
 </ Order> ... Omitted ...
</ Buy Data>
```

**Figure 4.** Examples of XML schema changes

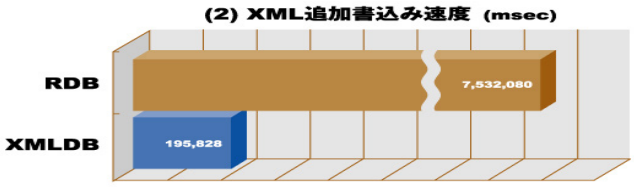



**Figure 5.** Writing speed

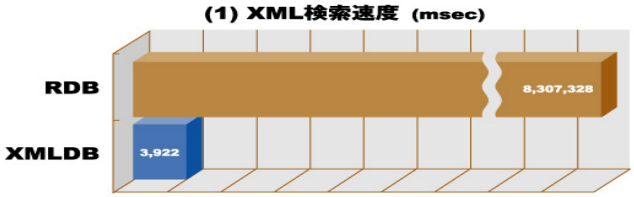



**Figure 6.** Searching speed

## 6. Conclusions

This paper presents a Method for efficient retrieval of large index XML documents as shown in Figure 7. First, existing studies show the true performance to lean to one side in the keyword search of large XML documents have a problem. Both informal and formal data processing, the Content Model was proposed to solve. Non-structured data to a right processing and utilization as NoSQL View in a variety of devices for rapid processing of the user interface was coupled to Data. N-Screen or the like can be used without Viewer from various mobile devices, as well as the task of processing that is used in a large variety of services can be advantageous.

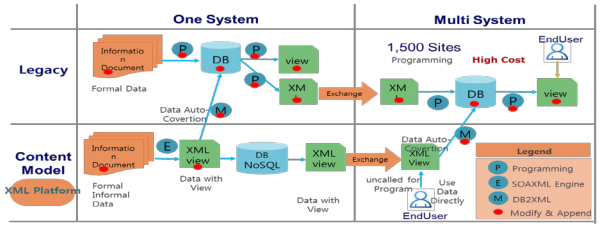



**Figure 7.** Compare the legacy system and the XML platform system

## 7. References


1. XML Path Language (XPath) Version 1.0. https://www.w3.org/TR/xpath/. Date Accessed: 07/09/2015.
2. Chamberlin D. XQuery: An XML query language. *IBM System Journal*. 2002; 41(4): 597-615.
3. Xu Y, Papakonstantinou Y. Efficient Keyword Search for Smallest LCAs in XML Databases. Proceeding of the 2005 ACM SIGMOD international conference on Management of data, USA. 2005, p.527-38.
4. Hoang Do Thanh Tung, Dinh Duc Luong. An Improved Indexing Method for Xpath Queries, *Indian Journal of Science and Technology,* 2016 Aug; 9(31):1-7.
5. XQuery 1.0: An XML Query Language http://www.immagic.com/eLibrary/ARCHIVES/SUPRSDED/W3C/W010607D.pdf. Date Accessed: 07/06/2016.
6. Trotman A, Lalmas M. Why Structural Hints in Queries do not Help XML-Retrieval, *SIGIR '06 Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*. 2006, p.711-12.
7. Larsen B, Tombros A, Malik S. Is XML Retrieval Meaningful to Users? Searcher Preferences for Full Documents vs,Elements, *SIGIR '06 Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval,* 2006, p. 663-64.
8. Kamps J, Koolen M, Lalmas M. Where to Start Reading a Textual XML Document?. *SIGIR '07 Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development In Information Retrieva*. 2007, p. 723-24.
9. Kim HS, Son HJ. Users Interaction with the Hierarchically Structured Presentation in XML Document Retrieval, *Springer Berlin Heidelberg,* 2005 Nov; p. 422-31.
10. Betsi S, Lalmas M, Tombros A, Tsikrika T. User Expectations from XML Element Retrieval, *SIGIR '06 Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2006, p.611-12.
11. Florescu D, Kossmann D, Manolescu I. Integrating Keyword Search into XML Query Processing, *Proceedings of the 9th international World Wide Web conference on Computer networks,* 2000 Jun,pp.119-35.
12. Shimizu T, Terada N, Yoshikawa M. Kikori-KS: An Effective and Efficient Keyword Search System for Digital Libraries in XML. *Springer Berlin Heidelberg*, 2006 Nov, p. 390-99.
13. Guo L, Shao F, Botev C, Shanmugasundaram J. XRANK: Ranked Keyword Search over XML Documents, *SIGMOD '03 Proceedings of the 2003 ACM SIGMOD International Conference on Management of Data,* 2003, p. 16-27.
14. Structural Feedback for Keyword-Based XML Retrieval. http://dbis.eprints.uni-ulm.de/1268/. Date Accessed: 09/09/2015.
15. Ko SK, Choy YC. A Structured Documents Retrieval Method supporting Attribute-based Structure Information. *SAC 2002, Proceedings of the 2002 ACM Symposium on Applied Computing*. 2002, p.668-74.
16. Francois P. Generalized SGML repositories:Requirements and modelling. *Computer Standards and Interfaces*. 1996 Jan;18(1):11-24.