XML DOM Parsing Using Tree-Branch Symbiosis Algorithm

Ms. Ruchita A Kale, Ms. Yogita S. Alone, Mr. Gaurav J. Sawale, Ms. Archana B. Pahurkar

Abstract—The XML (eXtensible Markup Language) has become the de facto standard for information representation and interchange on the Internet. As extensible Markup Language (XML) becomes widespread, it is critical for developers to understand the performance characteristics and operational of XML processing. The processing of extensible Markup Language (XML) documents has been regarded as the performance bottleneck in most systems and applications. XML parsing is performed a core operation on an XML document for it to be accessed and manipulated. Using the tree branch symbiosis algorithms XML documents are parsed the document and its elements are stored in a single table of database. It is not necessary to read the according to their hierarchical structure. In this paper, we proposed the hash function when applied on the database would speed up the access time hence improve the XML processing performance means accessing time is less.

Index Terms—eXtensible markup language (XML), Tree-Branch symbiosis, DOM, SAX.

1 INTRODUCTION

eXtensible Markup Language means XML. XML is a meta-language derived from Standard Generalized Markup Language and is used to store and exchange structured information. eXtensible Markup Language has become a de facto standard for data representation and exchange. XML data processing become more and more important for server workloads like Web Servers and Database Servers and also in messaging database and document processing.

XML is designed to provide flexible information identification in web documents. The important role of XML is the representation and exchanging the any kind of structured document because it is platform-independent, human readable and extensible and also its own defined well data format. XML data processing has its two technologies- DOM and SAX. DOM (Document Object Model) is a platform and language-neutral interface to represent XML document as an object oriented model. The representation of DOM is tree like and all data is in memory that’s why it is less efficient in the term of storage and time. SAX (Simple API for XML) is an event-driven, serial access mechanism for accessing XML document. A Simple API for XML (SAX) parser reads an XML document as a flow and invokes call back functions provided by the application.

A eXtensible Markup Language (XML) document essentially can be represented as tree structure data model, the XML document thus can be regarded as the serialization of tree model in a depth-first search, left to right travelling order. For representation of tree structure DOM object is much more popular format. A DOM parser is much more compound and much slower. Because DOM parser store entire data in memory, and at the time of data accessing, every time data accessing or reading from the memory and degrading the speed of processing. Some advantages of DOM parser are-
- DOM is easier to use than SAX because it provides a familiar tree structure of objects.
- Structural manipulations of the XML tree, such as reorder elements, adding to and deleting elements and attributes from tree, and renaming elements, can be performed.
- Interactive applications can store the object model in memory, enabling users to access and manipulate it. Accessing the data from memory easier and user freely.
- A pluggable, scalable DOM can be created that considerably improves scalability and efficiency.

The stages XML processing: parsing are access, modification, and serialization. Although parsing is the most expensive operation, there are no detailed studies that compare the processing steps and associated overhead costs of different parsing models, tradeoffs in accessing and modifying parsed data, and XML-based applications’ access and modification requirements.

Storing XML document in a relational database that maintains its nodes and their relationships is a popular way in a business application, which needs high precision of data and their relationship. The Web technology M&S, a series of high performance model for XML processing are needed be created to meet the characteristics if magnanimous data exchange and
processing.

The storage of a XML document in a relational database with its original structure in general business application, the several important benefits are-

- A node and its path can be queried easily and fast due to its perspicuity structure.
- The data can be managed by DBMS easily.
- It is useful for data transfer.

2 RELATED WORK

A. Detailed XML Parser Overview

In 2002, Srikanth Karre et.al explains that the parsing of XML documents can be done using two approaches, Event Based Parsing and Tree Based Parsing. In Event Based Parsing, the XML data is parsed sequentially, one component at a time, and the parsing of events such as the start of a document, or the end of a document are reported directly to the application. SAX (Simple API for XML) is the standard API for event-driven parsing. In Tree Based Parsing, the XML document is compiled into an internal tree structure and stored in main memory [1]. Applications can then use this tree structure for navigation and data extraction. For example, the Document Object Model (DOM) uses tree based parsing, providing a standard set of objects for representing HTML and XML documents, a standard model of how these objects can be combined, and a standard interface for accessing manipulating them.

In 2006, Tong, T. et al. studied the XML parser. They conclude the work of parser can read the XML document components via Application Programming Interfaces (APIs) in two approaches. For stream-based approach (also known as event-based parser), it reads through the document and signal the application every time a new component appears. As for tree-based approach, it reads the entire document into a memory resident collection of object as a representation of original document in tree structure [5]. As a result, tree-based approach is not suitable for large-scale XML data because it can easily run out of memory.

Chengkai Li- XML Parsing, SAX/DOM, explains that in every application that takes XML document to process, parsing is the first important step to be done. Also explains that DOM previously was used for modeling HTML after that the different levels of DOM were proposed and that has been used to parse the XML document which models the XML document as a tree of node where the application can read, write and update the contents of the nodes whereas SAX (Simple API for XML) is a event driven interface which reads the XML document, generates the events and triggers the corresponding XML handlers [24].

B. Parallel XML Parsing

In 2007, Wei Lu et.al. states that there are a number of ways to improve XML parsing performance. One of the approaches would be to use pipelining. In this approach, XML parsing could be divided into a number of stages. Each stage would be executed by a different thread. This approach may provide speedup, but software pipelining is often hard to implement well, due to synchronization, load-balance and memory access costs. More promising is a data-parallel approach. Here, the XML document would be divided into some number of chunks, and each thread would work on the chunks independently. As the chunks are parsed, the results are merged [7]. To divide the XML document into chunks, and could simply treat it as a sequence of characters, and then divide the document into equal-sized chunks, assigning one chunk to each thread. This requires that each thread begin parsing from an arbitrary point in the XML document.

Since an XML document is the serialization of a tree-structured data model (called XML Infoset) traversed in left-to-right, depth-first order, such a division will create chunks corresponding to arbitrary parts of the tree, and thus the parsing results will be difficult to merge back into a single tree. Correctly reconstructing namespace scopes [7].

The results of parsing XML can vary from a DOM-style, data structure representing the XML document, to a sequence of events manifest as callbacks, as in SAX-style parsing. The parallel approach focuses on DOM-style parsing, where the tree data structure is created in memory that represents the document. Their implementation is based on the production quality libXML2 parser, which shows that their work applies to real-world parsers, not just research implementation.

C. Evaluating performance Using Different Techniques Or Algorithm

In 2013, V.M. Deshmukh et. al. presented the performance study of XML data parsing by evaluating the parsers using time as a parameter. They mainly focused on different data structures which are linear in nature like stack, array, queue and linked list. Data structure based parser works in main memory and uses various data structure for parsing. In the implementation, the proposed parser removes the elements from document and serially checks if the document is well formed or not using Linked list, Queue, Stack and Array simultaneously, which increases its performance over SAX and DOM parser. Proposed the conclusion by observed analysis and graphical results that the data structure based parser is efficient than others.
They compare the performance of different parsing like DOM, SAX and also the different data structure. Using the different data structure for accessing or reading the data from the database required less time as compared to others method. Also they explain the design and development of an efficient XML parsing algorithm, Parsing is a core operation performed before an XML document can be navigated, queried, or manipulated. Recently, high performance XML parsing has become a topic of considerable interest.

In 2010, Gong Li et. al.- Present a XML processing model on data exchange between XML document and relational database, model parsing a XML document to a DOM object, kernel of the model was the tree-branch symbiosis algorithm, by which, the efficiency of DOM building will be promoted significantly. The processing of XML document’s has been regarded as the performance bottleneck in most systems and applications. A number of techniques have been developed and improve the performance of XML processing, ranging from the schema-specific model to the streaming-based model to the hardware acceleration. These methods only address parsing and scheduling the XML document in memory. Although there are a few of works have discussed the efficiency of the data read-write between XML and Relational Database, they constructed the DOM and reading relational database synchronously and neglected the differences of pace between DOM (a general format of XML document in memory) building and relational database reading, which will reduce the performance of the entire system.

A new processing model for storing and building XML document in data transfer between XML and relational database. In this Model, a XML document is parsed and its elements are stored in a single table of database instead, it is not necessary to read the nodes according to their hierarchical structure, thus leveraging the workload of DOM building to memory by the algorithm called Tree-Branch intergrowth. All data was stored in relational database. Using tree branch symbiosis algorithms for XML processing, they shows the comparison between old algorithms and tree branch symbiosis algorithm [16].

3 SYSTEM DESIGN

The system used DOM parser for the parse the particular file. The input to the system is a XML file and output is in the form bar diagram. The following Component diagram explains the external view of the system.

In the above component diagram, the various DOM parsing document the form and saved in evaluate on the time, of token, results combine table and per-operation. Finally, parison in chart.

In the proposed system, DOM parser is used for the parse the files the XML using parser. After parser parse and arrange in of tree structure the whole tree is memory and its performance and the database form all basic on the database. shows the component form of bar proposed system. DOM parser is the parse the file. The input the system is a XML file that we need to check file is not. The parser various XML files and evaluate its performance on the basis of parsing time, searching time of token and represent in the form of bar chart. The detail internal working of the system has been explained using the following flow diagram of the proposed system.
In Tree-Branch symbiosis XML processing model the nodes are stored in single database instead of multiple data in the form of AVL tree using hashing technique and all data tree branch symbiosis algorithm for XML processing, arrange the improvement was only in aspect of total processing time because traditional methods in the precondition of high performance this work is done in memory that the cost is almost the same with

REFERENCES

[22] Michael R. Head† Madhusudhan Govindaraju, “Parallel Processing of Large-Scale XML-Based Application Documents on Multi-core Architectures with PiXiMaL”, Fourth IEEE International Conference on eScience.
[24] Chengkai Li,”XML Parsing, SAX/DOM”.
[25] Li Zhao , Laxmi Bhuyan,” Performance Evaluation and Acceleration for XML Data Parsing”.