Received: Aug '22
Revised: Aug '22
Accepted: Sep '22

AN EMPIRICAL ANALYSIS OF XML PARSING USING MULTIPLE OPERATING SYSTEMS

Murti Mourya [1] and Dr. Ashish Chaurasia[2]
[1]Research Scholar Mansarovar Global University, Sehore (M.P.)
[1]Faulty of Govt. College Timarni
[2]Professor of Computer Science, Mansarovar Global University, Sehore (M.P.)

ABSTRACT

Extensible Markup Language (XML) is frequently used for online data exchange and transport. Thus, XML parser research—one of the most important XML technologies—has become crucial in this context. There are several XML parsers on the market right now, and many of them are developing, getting better, and getting more complex. Despite the fact that all parsers work toward the same objective, they vary in terms of specification, performance, dependability, and standard compliance. If the wrong choice is selected, it's probable that extra hardware would be needed, which would reduce productivity. To save time and improve processing speed while parsing XML documents and sending or sharing data using XML, it is critical to develop a system that enables us to parse an XML document in the least period of time possible. Recent studies have shown that the bulk of processing time is spent parsing, and many of these studies have concentrated on improving the processing performance of the XML Parser.

Keywords: XML Parser, Dom Parser, Operating System.

1. INTRODUCTION

The most recent comparison studies only considered parsing APIs while keeping in mind the idea of parsers. There hasn't been any study on how the environment, platform, or technique affect how long it takes to parse an XML document. No research has been done to identify how the operating system affects the parsing of XML documents. Therefore, it is vital to explain terms like XML, Parser, DOM API, and OS before getting into further detail about the research.

SGML, HTML, and XML Techniques Overview-

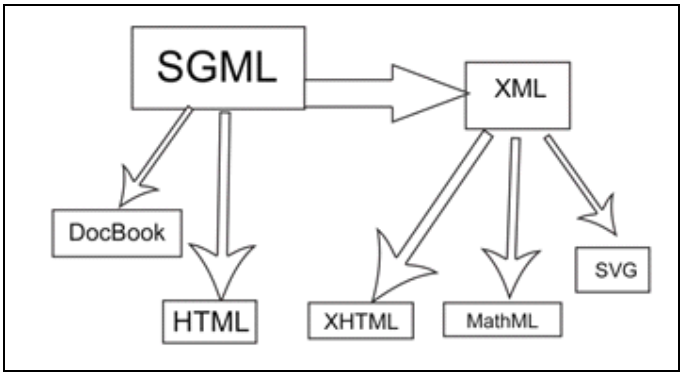


Figure: 1. SGML Subset and Extension

SGML

The ISO declared Standard Generalized Markup Language (SGML), a structural markup language for electronic documents, to be a global standard in October 1986.

HTML

HTML is a common computer language for creating web pages (Hypertext Markup Language). The term "hypertext" refers to the links that are present on a webpage and how online pages are linked to one another (HTML documents). HTML is used to "mark-up" text documents with tags that tell a web browser how to arrange them, as its name suggests. HTML was created to specify the organisation of documents, such as headings, paragraphs, lists, and other elements, in order to facilitate the sharing of scientific data among researchers. Today, HTML and a variety of tags are routinely used to create and format online pages.

XML

Extensible

It allows you to set your own tags and how they should be handled or displayed in the order they appear.

Markup

The most recognisable aspect of XML is its tags or elements. The elements in your HTML documents will be strikingly similar to those in XML. Actually, XML allows you to design your own set of tags.

Language

In addition to being called meta-language, XML also allows you to create your own tags. Compared to HTML, it is substantially more flexible. Languages like RSS, Math ML, and even tools like XSLT may be created using XML. A set of standards for text encoding that is both machine and human readable is also specified by XML.

2. LITERATURE REVIEW

An examination of XML parsers could highlight the advantages and disadvantages of various XML parsers in terms of their various features. In order to compare parsers, we look at how well they adhere to the XML standards set out by the World Wide Web Consortium [1]. The Organization for the Advancement of Structured Information Systems (OASIS), [2] a nonprofit organisation, produced a Conformance Test Suite for XML with approximately 2000 test cases by combining different test cases from a variety of sources (as of November 6, 2001). An investigation was carried out in 1999 by Anez [3] in order to ascertain whether or not XML and Java were capable of accurately representing and manipulating the Transport and Land Use (TLU) modelling data that is required for urban and regional planning. He evaluated the relative capabilities of seven distinct XML parsers by evaluating how well each one performed in terms of XML standards, speed, and memory use. James Clark, who is currently a contributor to the OASIS Test Suite, created an assessment tool to gauge the performance and memory requirements of several parsers while comparing their conformance with two sizable XML files (0.8 and 1.2 MB, respectively). He rates several parses (given in Table 1) based on the thousands of XML elements in the two XML files, each with one or more attributes and nesting in a four-level deep hierarchy.

Table: 1 Anez's Ranking of Several Parsers

Parser	Rank
IBM XML4J (XML For Java) V 1.1.4	Outstanding
James Clark's XP V0.4	Good
Microsoft XML (MSXML) V 1.9	Good
Micro Star Aelfred V1.1	Good
Sun XML (Under Construction)	Acceptable
Loria Sxp V 0.72	Acceptable
Data Channel XML Parser	Poor

For his examination in 1999, Claben [4] looked at IBM XML4J, Apache Xerces, Sun Project X, Microsoft MSXML, Oracle XML parser for Java, and James Clark XP. The several parsers were compared using a number of factors, including well-forkedness, validity, XML Schema, namespaces, XSL-T, SAX levels 1, 2, and DOM levels 1, 2.

In 1999, Cooper [5] used five XML documents with sizes ranging from 160 K to 5.0 MB and two Java parsers, two C parsers, one Perl, and one Python parser to study and analyse how parsing times varied with the parser development language. He discovered that C parsers are consistently quicker than other parsers. In a performance test he did in 2001, Mohseni [6] identified the MSXML rival parser that loaded documents the fastest.

In their study from 2002, Karre, S., and Elbaum, S. [7] used five java-programmed XML parsers for validation and examined three factors: correctness, speed, and storage space.

They developed a set of metrics and testing scenarios to test these parsers using an improved version of the OASIS XML Conformance Test Suite [10], and they found that XML parsers varied in many ways, giving developers insight into how various parsers behave.

The four stages of managing XML documents were identified by Lam T.C., Ding J.J., and Liu J.C. in 2008 [20]. These phases are access, modification, serialisation (figure 2), where parsing models have an impact on performance, and parsing, which is the most crucial stage in terms of performance.

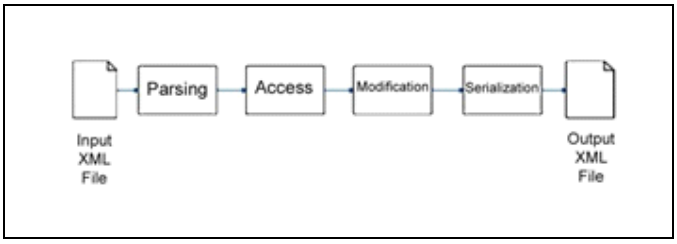


Figure: 2. Example of a XML memory tree representation

The term "parsing" refers to both the process of converting characters into a format that can be understood by a programming language as well as the lexical analysis procedure that is used to identify XML elements such as start nodes, end nodes, or characters by making use of regular expressions that have been established by the World Wide Web Consortium (W3C). After the application programming interface (API) executes access and change operations on the process's final data, the syntactic analysis of the document, which is the final stage of the parsing phase, determines whether or not the document fits with the criteria for generating XML documents. According to their research on the data formats, parsing methods, and the effects of these factors on XML processing, both DOM and VTD are effective when it comes to providing access to data in both directions. VTD is preferable to DOM because it requires less memory and parses data more quickly, making it better for complicated and frequent changes and appropriate for database applications. While SAX and StAX are great for memory-constrained and streaming applications, change or back-and-forth access is not their strongest suit. Although VTD seems to be a strong contender for hardware acceleration in the case of symmetric array structures, it is yet unclear whether or not it will be effective in real-world applications that make use of commercial hardware accelerators. The web services architecture was built on the semi-structured document language XML, which is widely used in document processing, databases, and messaging systems, claim V M Deshmukh and G R Bam in their note [25]. Validating XML documents requires parsing. The performance of XML parsing is known to be subpar when compared to transactional database operations.

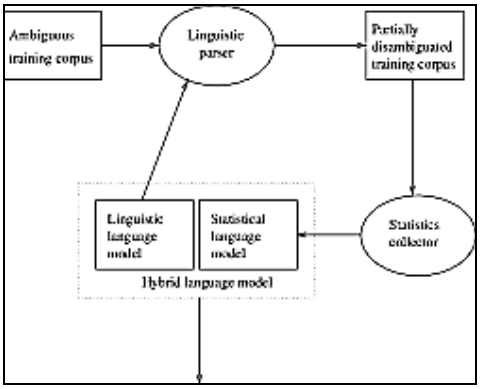


Figure: 3. Architecture View of Parsing

3. RESEARCH WORK

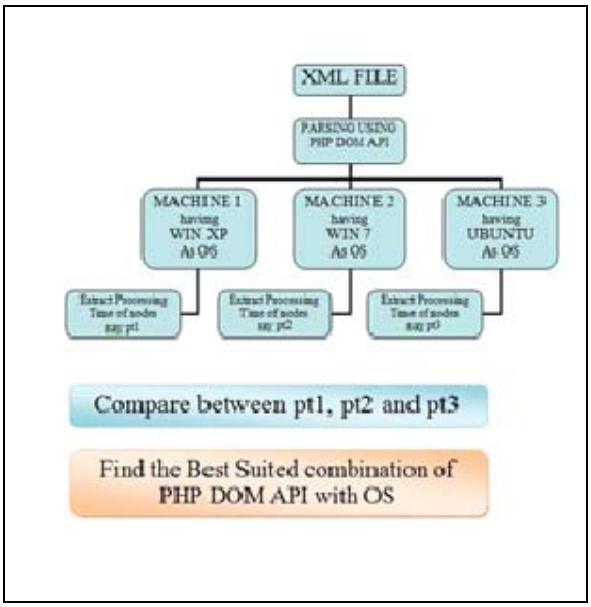
In The Proposed Field Of Investigation, Research Gaps Have Been Discovered: All relevant research have demonstrated that processing XML data, notably parsing XML data, takes up a sizeable portion of its execution time [26]. The input XML document is divided into digestible bits via data processing. Since parsing XML documents is required before any other actions can be performed, it accounts for a large amount of XML data processing. Studies reveal that roughly 30% of

web service applications are used for data processing. All comparative study has focused on the parser, hence the most current comparisons have only been done in terms of the parsing API.

The method, environment, or platform that might affect how long it takes to parse an XML document have not been studied. No research has yet looked into how the operating system affects the parsing of an XML document.

4. RESEARCH DESIGN

For the purpose of data analysis, we made use of descriptive statistics in conjunction with a 1x3 factorial ANOVA technique. For the purpose of comparison, we utilised mean, standard deviation, t-test, and z-test.



5. CONCLUSION

We are putting different parsing methods through their paces to determine which one is the most effective for each operating system. My research's working modal has been displayed in this exhibition. In order to conduct an analysis of the data, we relied on descriptive statistics in conjunction with a 1x3 factorial ANOVA method, as well as the mean, standard deviation, t-test, and z-test.

REFERENCES

1. Karre, S. and Elbaum, S., "An Empirical Assessment of XML Parsers", 6th Workshop on Web Engineering, 2002, pp. 39-46.
2. 2. Michael Claben, XML Parser Comparison, http://www.webreference.com/xml/column22/index.html. Feb 1999.
3. Elliotte, R.H., "SAX Conformance Testing", XML Europe, 2004.
4. E. Perkins, M. Kostoulas, A. Heifets, M. Matsa, and N. Mendelsohn, "Performance Analysis of XML APIs", in XML 2005 Conference proceeding, 2005.
5. Brian F. Cooper, Neal Sample, Michael J. Franklin, Gísli R. Hjaltason, Moshe Shadmon A Fast Index for Semi structured Data Proceedings of the 27th VLDB conference, Roma, Italy, 2001.
6. Chulho Ahn, Quing Li, Ramez Elmasri, Shalli Prabhakar, Niroj Manandhar, Do Youn Kim "A Survey of Three Types of XML Indexing Techniques" ACM Transactions on Computational Logic, Vol. 37, No. 4, 12 2005, Pages 1 -24.