

# COMPARISON OF XML PARSING THROUGH PHP API AT DIFFERENT NODE VALUE BY USING VARIOUS OPERATING SYSTEMS

<sup>1</sup>AMITESH SAXENA, <sup>2</sup>SNEHLATA KOTHARI

<sup>1</sup>Research Scholar, faculty of science, Pacific academy of higher education and research university, Udaipur  
<sup>2</sup>H.O.D. IT Department, Pacific University, Udaipur

**Abstract**— A parser is one of the components in an interpreter or compiler, which checks for correct syntax and builds a data structure (often some kind of parse tree, abstract syntax tree or other hierarchical structure) implicit in the input tokens. Parsers may be programmed by hand or may be (semi-)automatically generated by a parser generating tool. Various techniques are available for parsing formal languages. The objective of this paper is to compare these techniques. The paper is organized in two sections. The first section does discuss about the parsing technique and process. In the second section we study and compare three parsing techniques theoretically. Finally the paper concludes with the suggestion of best parsing technique.

## I. INTRODUCTION

Parsing is the process of breaking a data block into smaller chunks by following a set of rules, so that it can be more easily interpreted, managed, or transmitted by a computer. Spreadsheet programs, for example, parse a data to fit it into a cell of certain size.

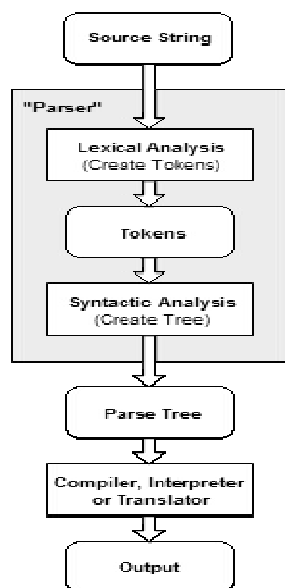


Fig 1:- overview of parsing process

XML parsing is the process of reading an XML document and providing an interface to the user application for accessing the document. An XML parser is a software apparatus that accomplishes such tasks. In addition, most XML parsers check the well-formedness of the XML document and many can also validate the document with respect to a DTD (Document Type Definition) or XML schema. Through the parsing interface, the user application can focus on the application logic itself, without dwelling on the tedious details of XML.

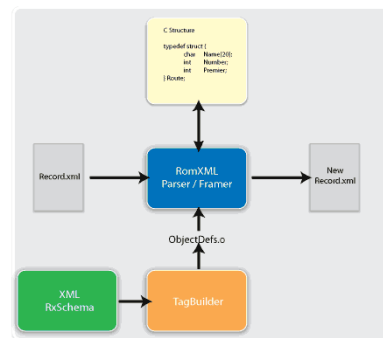


Fig 2:- overview of XML parser

```

<message>
  <to>you@yourAddress.com</to>
  <from>me@myAddress.com</from>
  <subject>XML Is Really Cool</subject>
  <text>
    How many ways is XML cool? Let me count the ways...
  </text>
</message>
  
```

Fig 3:- XML file Format

XML parsers can be classified into two broad categories, based on the types of API that they provide to the user applications for processing XML documents

**Document Object Model (DOM):** DOM is a tree-based interface that models an XML document as a tree of various nodes such as elements, attributes, texts, comments, entities, and so on. A DOM parser maps an XML document into such a tree rooted at a Document node, upon which the application can search for nodes, read their information, and update the contents of the nodes.

**Simple API for XML (SAX):** SAX is an event-driven interface. The application receives document information from the parser through a Content Handler object. It implements various event handlers in the interface methods in Content Handler, and

registers the Content Handler object with the SAX parser. The parser reads an XML document from the beginning to the end. When it encounters a node in the document, it generates an event that triggers the corresponding event handler for that node. The handler thus applies the application logic to process the node specifically

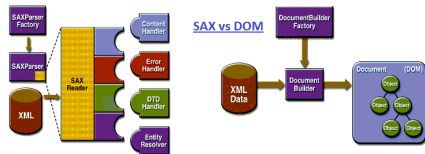


Fig 4:-DOM v/s SAX parser

**Parsing Techniques**

**Classical parsing**

Classical parsers for formal languages have been known for many years. They conventionally accept a context-free language defined by a context free

**Fuzzy Parsing**

Fuzzy Parsing (Koppler, 1997) was designed to efficiently develop parsers by performing the analysis on selected parts of the source instead of the whole input. It is specified by a set of fuzzy context free sub grammars each with their own axioms. Unlike conventional parsing, it does not require strict adherence to a language grammar. It scans for instances of the axioms and then parses according to the grammar. It makes parsing more robust since it ignores source fragments including missing parts, errors and deviations therein – that subsequent analyses abstract from anyway.

**Research work**

In my research work I have parsed the XML file through PHP API on various operating systems like Windows Xp, Windows 7, Windows 8, Fedora and UBUNTU. By parsing the XML file through different operating system I have taken the reading in form of time (in milliseconds) required to parse a file. By all observation I have plotted the behavior of all operating system for minimum, maximum and average value. I also tested this for three different node values that are 531 nodes, 1123 Nodes and 1666 nodes.

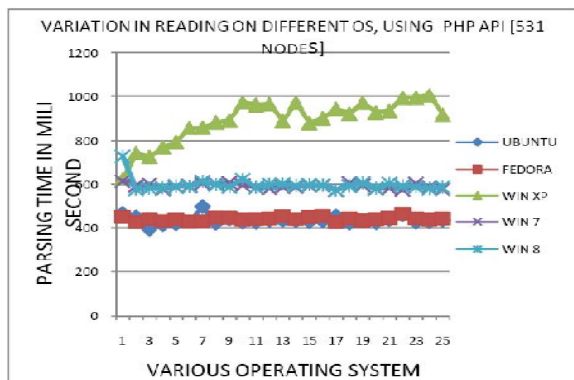


Fig 5:- comparison of variation in parsing time at node value 531 in various operating systems

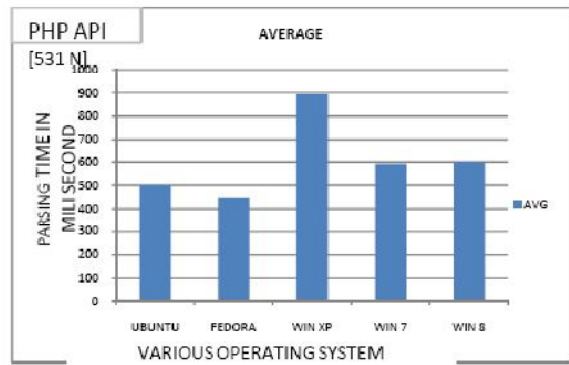


Fig 6:- Test result for average parsing time at node value 531 in various operating systems

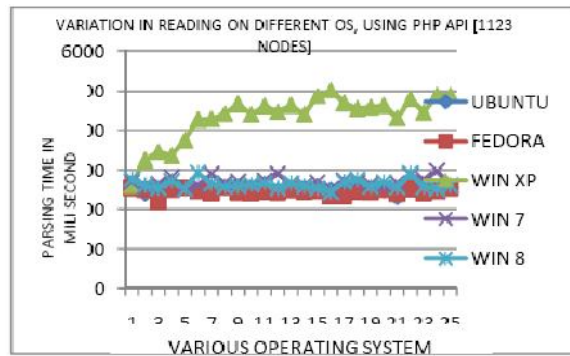


Fig 7:- comparison of variation in parsing time at node value 1123 in various operating systems

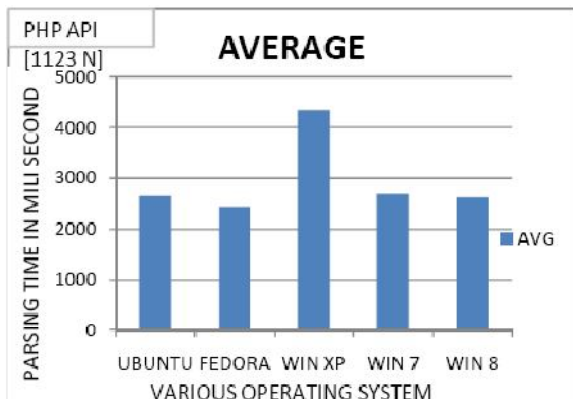


Fig 8:- Test result for average parsing time at node value 1123 in various operating systems

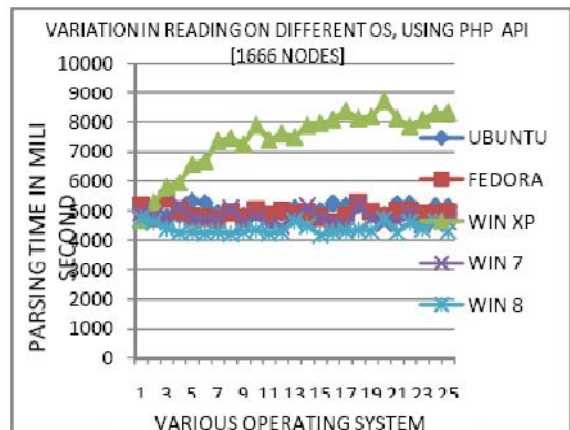


Fig 9:- comparison of variation in parsing time at node value 1666 in various operating systems

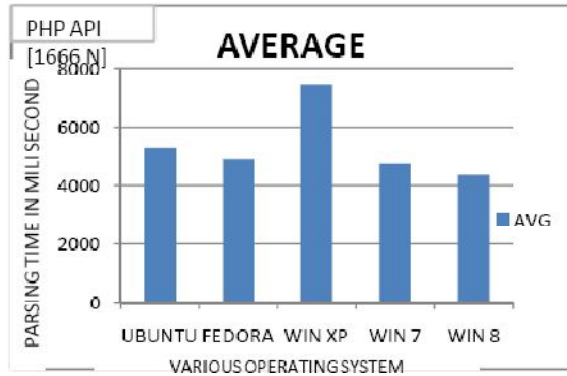


Fig 10:-Test result for average parsing time at node value 1666 in various operating systems

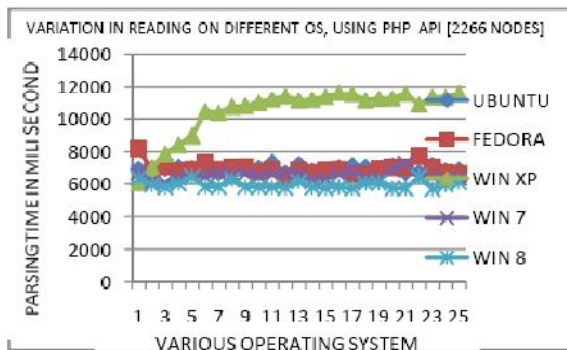


Fig 11:-comparison of variation in parsing time at node value 1666 in various operating systems

★★★

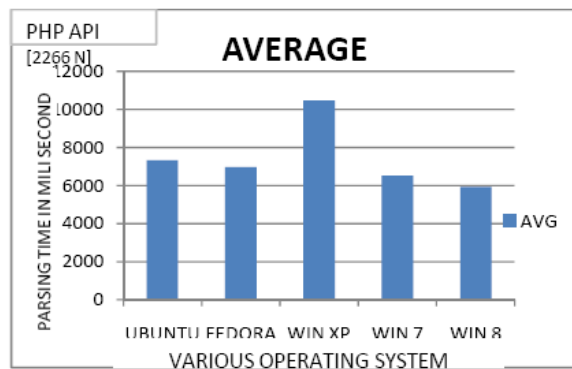


Fig 12:-Test result for average parsing time at node value 2266 in various operating systems

## CONCLUSION

Form the all above testing I have concluded that the different operating system has different time to parse the XML file and minimum time is for windows 7 but no one is up to date because XML parser is a technology, which is changing rapidly for the moment, there is no single study or research that would valid forever.

## REFERENCES

- [1] <http://en.wikipedia.org/wiki/Parsing>.
- [2] Jens Nilsson, Proceedings of the 11th International Conference on Parsing Technologies (IWPT), Paris, October 2009 c Association for Computational Linguistics.
- [3] Zisman, A., "An Overview of XML", Computing & Control Engineering Journal, 2000.

- [4] Karre, S. and Elbaum, S., "An Empirical Assessment of XML Parsers", 6th Workshop on Web Engineering, 2002, pp. 39-46.
- [5] Michael, C., "XML Parser Comparison", 2000, <http://www.webreference.com/xml/column22/2.html>
- [6] Tong, T. et al, "Rules about XML in XML", Expert Systems with Applications, Vol. 30, No.2, 2006, pp. 397-411.
- [7] Kiselyov, O., "A better XML parser through functional programming", LNCS 2257, 2002, pp. 209-224.
- [8] Elliotte, R.H., "SAX Comformance Testing", XML Europe, 2004
- [9] J. Anez, "Java XML Parsers- A Comparative Evaluation of 7 Free Tools", Java Report Online, 1999.
- [10] Mohseni, P., "Choose Your Java XML Parser", 2001.