Problem Statement

CS 461 - CS Senior Capstone

XZES40 | Group #40

Zixun Lu (luzi), Shuai Peng (pengs), and Elijah Voigt (voigte)

November 9, 2016

Abstract

1 Introduction

2 Research and Benchmarking

2.1

2.2

2.3

3 XML/XSLT Document Transformation

3.1

3.2

3.3

4 XML/XSLT Document Compilation

XML document compilation is the process by which an XML formatted document is taken and parsed into an in-memory object. The library we will be using, as requested by or client, will be the Xerces-C XML parser library; this review will evaluate that library. The other point of wiggle-room we have is in what way we parse and store the document in memory.

4.1 Xerces-C

The Apache Xerces-C++ library is a feature rich XML parsing library. It implements the XML 1.0 standard, it is well documented, and implements DOM as wel as SAX methods of document parsing. Most importantly it integrates well with Xalan-C++ which is able to exactly perform all of our application needs.

This library was also explicitly requested by our client, a member of the Apache Software Foundation, so we will have organization support if we choose to use it.

4.2 DOM Parsing

The first method of XML document parsing is DOM parsing. This method stores the entire parsed XML object in memory to be traversed later on. This operation is memory intensive but is useful for carrying out many operations on a document. [**?**]

4.3 SAX parsing

In contrast, SAX document parsing is carrying out in a stream, using callbacks to trigger transformations on an XML document. This method is less memory intensive and can result in a performance boost, but requires a document stream for transformations to happen. [**?**]

4.4 Conclusion

We will be using the Xerces-C++ library to accomplish this task as it is feature rich enough to give us leeway in our development if we need it, while also being lean enough that it should not affect performance when compared with other options.

As for choosing between SAX parsing vs DOM parsing, we will most likely choose DOM parsing since it fits our application best. DOM parsing produces an easily cached tree which we can store and retrieve. That said, it may be worth-while to investigate using SAX parsing early on in development because we find a notable performance boost. Some amount of SAX proof of concept work will be done, but DOM will be the targeted document transformation method.

15 References

[?] [?] [?]

References

[1] *Apache Xalan*. "http://xalan.apache.org.

[2] *Apache Xerces*. http://xerces.apache.org.

[3] *Events vs. Trees*. http://sax.sourceforge.net/event.html.

[4] *International Components of Unicode*. http://icu-project.org.

Elijah C. Voigt

_____

Signature

_____

Date

Zixun Lu

_____

Signature

_____

Date

Shuai Peng

_____

Signature

_____

Date

Steven Hathaway

_____

Signature

_____

Date