High Performance XML/XSLT Transformation Server

Fall 2016 Progress Report

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

CS 462 | CS Senior Capstone | Winter 2017

February 13, 2017

Abstract

An update on the development, stumbling blocks, and more broadly the execution of the High Performance XML/XSLT Transformation Server *XZES40 Transformer*.

Figure 1: Source: Wikimedia Commons [2]

Figure 2: Source: Apache Software Foundation [1]

1 Project Purpose

XZES40 Transformer is an implementation of a standard XML document transformer. This type of application takes two documents as input: an XML document and an XSLT document, and performs and XML document transformation. This can be thought of like how in an Excel document one might take a series of columns, perform some operation on that data, and set the output to be a new column, except that this can be highly automated.

XZES40 Transformer is unique in that it increases performance by caching processed XSLT and performs document transformations in parallel. The use of an in-memory cache will cut down considerably on the time required to perform a document transformation as much of the time transforming documents is spent compiling XSLT documents. The application will also accept transformation jobs over the internet via a web API, allowing users to perform document transformations without installing the application locally.

2 Project Status

Because all of our teammate is busy in this term, we are slow than our schedule that we planed in the design document. So far, we just complete the basic function of the XZES40-Transformer. The XZES40-Transformer can get one XML file and XLS file and generate a new XML file. The cache and the parallel computation will be completed during the beta version.

3 Remaining Tasks

3.1 Zixun Lu

3.2 Shuai Peng

The remaining tasks that I am responsible for is complete the cache function, web interface, and Windows package.

3.3 Elijah C. Voigt

4 Problems Encountered

4.1 Zixun Lu

4.2 Shuai Peng

When we are working on the basic document transformation, we face a strange problem. The structure and the implementation of our code should be no any problem. However the compile also say that we have expected unquality-id bugs for the xalanc++ transformer function. Elijiah and me search on the Google, and most of answers said that you should check the syntax of code and some semicolon error. We double check the code, but there is no error such that syntax and semicolon problem. So, we come together and write a new prototype for test xalanc++ transformer, and the prototype is work correctly. Thus, we decided to re-write the transformer.cpp, and make sure this code is only doing one things – transformer document. We change the structure of transformer.cpp from OOP to normal function code, and also make sure the document.cpp pair this file. Finally, we get over this bugs, and our program can transformer document successful.

4.3 Elijah C. Voigt

5 Interesting Code

5.1 Zixun Lu

5.2 Shuai Peng

This is the structure of the XZES40-Transformer.

```
vagrant:~xzes40/xzes40/transformer$ ls
bulid doc examples include lib Makefile README.md src test
```

This is the usage of the XZES40-Transformer.

```
vagrant:~xzes40/xzes40/transformer/bulid$ ./main
Usage:
  a.out --xml=input.xml --xsl=style.xslt [--out=output file]
```

This is the result of the XZES40-Transformer. We print the new xml file on screen for easy debug, but it will generate a xml file in the future.

```
vagrant:~xzes40/xzes40/transformer/bulid$ ./main --xml=simple.xml --xsl=simple.xsl --out=result.xml
<?xml version="1.0" encode="UTF-8"?><out>Hello</out>
```

5.3 Elijah C. Voigt

6 Relevant Media

6.1 Zixun Lu

6.2 Shuai Peng

6.3 Elijah C. Voigt

References

[1] *ASF Press Kit: Apache Software Foundation Logo*. URL: https://www.apache.org/foundation/press/kit/.

[2] *Wikimedia Commons: Oregon State University Logo*. URL: https://commons.wikimedia.org/wiki/File:Oregon_State_University_log