High Performance XML/XSLT Transformation Server

Fall 2016 Progress Report

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

CS 461 | CS Senior Capstone | Fall 2016

June 4, 2017

Abstract

A progress report for development during the Fall 2016 term the high-throughput XML/XSLT transformation server project. Outlining the development progress, stumbling blocks, and solutions during the planning stage of development.

Figure 1: Source: Wikimedia Commons [2]

Figure 2: Source: Apache Software Foundation [1]

1 Program Purpose and Goals

The XZES40-Transformer project is a collaboration between the Oregon State University Computer Science Capstone course and Steven Hathaway in affiliation with the Apache Software Foundation.

1.1 Purpose

The purpose of the XZES40-Transformer is to perform high throughput XML/XSLT document transformations. Given an XML 1.0 and XSLT 1.0 input document our application will return a transformed XML 1.0 formatted document. In addition to transforming the document we are adding key optimization to the transformation pipeline to increase throughput, allowing for users of the system to accomplish more in a day.

1.2 Goals

The goal of the project is to create an Open Source application which performs faster XML/XSLT transformations than older XML/XSLT document transformers. In theory the application would be competitive with paid proprietary applications, but will at the very least it will be more performer than Open Source alternatives.

We will achieve increased performance by adding a caching layer and parallel transformations to our application. Caching will be added in key areas like the compilation stage and the transformation stage. Parallel computation will be implemented in the parsing stage as that is computationally heavy but should not create race-conditions.

The application will be exposed over a Web API for remote use of the application. This will make the application convenient to use as anybody with a web-browser and connection to the host server, and will ensure users do not need to install the package locally. Exposing the application over a remote client helps users maximize the use of the application as the Caching layer will be collective for all users, making the massive cache a problem for "the cloud" instead of a burden on local systems.

The application will, as a stretch goal, compile on and be packaged for multiple platforms including Linux, Berkeley Software Distribution (BSD), and Windows.

The following technologies will be used in the process of development:

- The **Apache Xalan-C++ and Xerces-C++** libraries will be used in transforming XML documents.
- The **International Components of Unicode (ICU) C++** library will be used to convert files to and from Unicode.
- The **Apache webserver** and a **Python Common Gateway Interface (CGI) script** will be used to provide the service over the internet to web-browser and command-line interface (CLI) clients.
- **Python** will be used to create and distribute the **CLI**.
- **Bootstrap** will be used to create the web interface so it is aesthetically appealing and usable.
- **FPM** and **WIX** will be used to package the application on Unix-like and Windows systems respectively.

2 Project State

Although we have not yet begun developing code for our project, we have begun working with our client to create a Development Virtual Machine which will be used for C/C++ code development. The VM will be used as soon as development begins and should reflect the production environment as it will look when the project is being used *for real*.

The VM is a 25+GiB Debian Linux VM with the following packages, libraries, and tools:

- The Xerces, Xalan, and ICU libraries we are required to use.
- Common C/C++ Build Dependencies.
- Git, Text Editors, Gnome and common VM tools.
- The TeXLive series of packages.

Although big it should will no doubt be feature rich enough for us to carry out development.

3 Problems Encountered

Although we have did the technology reviews, there is still problem about the XZS40-Transformer.

- Early on in the term our team had a hard time staying on the same page understanding the project as a whole. This has mostly been resolved as we worked together regularly on document writing.
- Our team confused about the format of some documents, including the Design Document. After we speaking with our TA we re-structured our documents to fit the requirements and gain a better understanding of the assignment goals.
- Toward the end of the term our client (Steven Hathaway) became less highly available and was too busy to have in-person meetings with us. This was not necessarily a show stopper, and was expected with the holiday season, but it did prove problematic in our turnaround.

4 Retrospective

Weeks	Positives	Deltas	Actions
Week3	We met with our client to discuss his vision for the application, initial requirements, and any resources we would need.	We needed to start working on the problem statement for the project.	Next we needed to meet with our sponsor to obtain a development virtual machine and finish the Problem Statement
Week4	We met with our sponsor to obtain a Debian Linux development virtual machine (which was 25+GiB!) and ask further clarifying questions about how best to move forward with our Problem Statement and Client Requirements document.	The Client Requirements document needed to be written and signed. We also needed to stat thinking about how best to use the VM our sponsor provide to us.	We needed to complete the Client requirement documents and contact our sponsor to sign our revised Problem Statement document.
Week5	We spent this week clearing up a **lot** of fundamental confusion about what our project is, what problem we were fixing, and how we were going to create a solution for that problem.	We didn't truly understand what we should be doing for our program, so we need to spend a lot of time getting caught up.	We needed to re-factor the Client Requirements documents next week after getting caught up.

Week6	We got into the habit of working together on our documents, scheduling almost daily meetings to work on and complete projects. This was effective in helping us complete a good Client Requirements document.	Our client was too busy to contact, so we were not able to get our document signed in person. We were able to get signed digitally, but we needed to keep our client's schedule in mind going forward.	Next we needed to start work on our the technology review due soon.
Week7	We partitioned our project into roughly 12 equally sized parts for the Technology Review and started work on that document.	We got trouble with our technology review as some elements we did not fully understand. To fix this we needed to do further research.	We later met with our TA to talk about the technology review to correct formatting and learning outcome misunderstandings.
Week8	We completed the Technology review at the deadline and began working on the Design Document.	Unfortunately not as much work was done on the Design Document as we had hoped because we as a group did not understand the intention of the IEEE format prescribed for the assignment. We had to rewrite a lot of the design document to organize it correctly.	We would try finish the rough draft of the design document the following week.
Week9	This week was Thanksgiving, so that got in the way of much getting done.	We finished the rough draft for our design documents, but had to talk with our TA to see if we were structuring it correctly.	We would finish the design documents next week after meeting with our TA and get ready for the final presentation/report.
Week10	We completely restructured our Design Document and turned in an unsigned draft because our client did not get back to us before the deadline.	We need contact our client, and request the sign document as soon as possible.	We needed to completely write, finish, and record the progress report/presentation, hopefully during the weekend of Fall week 10 before finals.

5 Conclusion

Looking to the future we are excited to work on the application. There are a few holes in our design document that may influence the implementation time-line, but for the most part our application is coming together nicely. In designing the application we were not met with any surprises, so what initially seemed like a simple application continues to look straight-forward.

All in all our project has not encountered many problems and if we are able to hit our deadlines the final product should work well. In practice this will be harder to achieve than just writing the code the design document lays out, just as so many horror stories regale. We will start early, work often, and stay focused.

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