High Performance XML/XSLT Transformation Server
Spring 2017 Final Project Report

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

OSU CS Senior Capstone 2016-2017

June 1, 2017

Abstract

Abstract text.

CONTENTS

would function, and what it would do differently than existing systems. The just of it being that a few layer of caching would be added to the transformation sever to speed up re-compilation of old documents (a common procedure).

Since last week we have finalized our Problem Statement document, confirmed it with our sponsor, and signed.

We have encountered no problems so far.

This coming week we will meet with our sponsor again to get setup with development environments. We decided on initially targeting Debian Linux as our platform, so we will be given VMs that are more or less the environment our application will be running on in production.

5.3.1.2 2016-10-21

This last week we continued editing our problem statement and met with our client to obtain a virtual machine for future development. The virtual machine was created by the client and included a slew of development tools for the package we will be developing and LaTeX development if we need.

There we no show-stopping problems encountered this week. One difficulty may be with the size of the virtual machine, which is almost 25GB, but once we have a stable development environment configured it shouldn't be an issue.

This coming week we will complete the Client Requirements document. We will also get a revised copy of our Client Requirements doc and Problem Statement doc signed and turned in on time.

5.3.1.3 2016-10-28

Since last week we have begun work on our Client Requirements documentWe turned in a rough draft, but will have to make a lot of edits before turning in a final document next week.

There was a lot of confusion about our what our application is, how it will be implemented, what it will do, and its core purposeTo fix this our group had a meeting. We decided to have daily meetings to improve our document writing workflow. Future documents will be written with all three of us so we're on the same page. We will have daily hour+ long writing meetings.

We will spend next week refactoring the Client Requirements documentOnce we complete a draft we like we will get it signed and submit this document.

5.3.1.4 2016-11-04

This week we completed the client requirements document and had it approved by our clientHe seemed impressed with the scope of the project.

Our client was too busy to have a meeting with us, but not too busy to sign the document, so it worked out.

This coming week we will begin work on the technology reviewNow that we have a solid workflow for writing documents as a team I think this will be straight-forward.

5.3.1.5 2016-11-11

Since last week we have started working on our Technology Assessment document, and to an extent (at least internally) our design documentThis document is not yet complete, but good work has definitely been put into it. Since bridging the knowledge gap I think that we've made good progress on unifying the design, and technologies being using in the project.

Our client has also supplied us with ample development scripts for our projectSince we have not started development these are not yet useful, but will be come winter term.

I don't recall any problems yetI was unfortunately unable to participate in daily meetings because of travel on the holiday weekend, but I don't think this will cause tremendous problems.

This next week (starting Sunday) we will complete the technology assessment document and begin work on the design documentI hope to have a first draft of the design document by this coming Friday, but that may be wishful thinking.

5.3.1.6 2016-11-18

Since last week we finished our tech review and started the design document.

We were sort of down to the wire when we submitted the tech review, but got it in before the midnight on Monday deadline.

We started working on the design document but are a little confused about the exact format of the document.

This coming week, before thanksgiving, we will try to get a rough draft of the design document done.

5.3.1.7 2016-11-25

Since last week we have begun working on our Design document and we created a repository to store code by our Clients request.

We are confused about the exact format of the design document, we will ask the TA about this to get it cleared up.

This coming week we will finish the design document and make an initial draft of the term summary due during finals week.

5.3.1.8 2016-12-02

Since last week we completely re-wrote our Design Document and turned it in, unfortunately unsigned due to time constraints with out clientA large amount of time was spent restructuring the document to more closely fit the IEEE standard we were adhering to.

We also began work on the Progress report document in hopes of getting it done before the start of Finals Week.

The largest problem we encountered this week was having to re-structure our documentWe misunderstood the IEEE structure for this assignment so we spent a lot of time re-writing the document after talking with the TA.

This coming week we will finish the Progress Report and Progress Report Presentation.

Over break we will also make headway on the actual assignment portion of the project, hopefully setting us ahead of schedule if all goes well.

5.3.2 Winter Term

5.3.2.1 2016-01-13

Over break I completed a large chunk of the project's structureThis included:

- Outlining the source code file-structure
- Added skeleton code for key classes, functions, and headers in the code.
- Added initial project documentation as well as code-documentation.
- Added a Makefile which currently compiles the project.
- Added a Vagrant virtual machine for lighter-weight developmentThis includes a setup script which can be used to provision a Continuous Integration system when we start using one.

Unfortunately I was the only member who was able to (or chose) to work on the project so there are a lot of decisions left to be made (changes to the design of the project including whether to use Redis or a library provided by Steven Hathaway for caching, weather to use Boost for cross-os compilation)

This week I will complete the structure of the project so work can begin on transformation and caching.

The team-members who did not engage with the project over the beak will catch up by reading what has been written and documented, getting familiar with important libraries, and setting up their development environments.

5.3.2.2 2016-01-20

Since last week we went started going over what we need to do for the project in finer detail, learning our parts of the project, and learning the required tools we need to complete the projectWe also started to plan in greater detail what and when we need to get tasks done? Tracking our progress better.

Shuai and Lu did not work on the project over the break so they are still catching upHopefully we won't be eventually behind and will eventually get back on schedule.

This coming week we will hopefully complete the basic transformation capabilities of our project, start bench marking competing products, and add continuous integration to pull requests.

5.3.2.3 2016-01-27

Since last week we have added Travis CI support and made progress in completing our basic functionality.

Shuai was unable to complete the basic transformation functionalityI spent the majority of Sunday January 28 (I know this is due Friday but I was late and might as well include this info) making major refactors to Shuai's contributions.

Lu has still not gotten any work done on this project.

This coming week I hope to get the basic transformation complete so that we can complete our alpha release on time.

Once we complete the basic transformation then there is a very short list of features to implement for the beta and final releaseThese are namely:

- Caching parsed documents.
- Daemonizing the application.
- Exposing the application to the web over an HTTP with a CGI script.
- Creating a Website for users to use the application.

Once those features are complete we will have finished the most important parts of our applicationTo finish any of those though we need to complete the core feature of document transformation.

5.3.2.4 2016-02-03

We have made very little progress since last weekAs mentioned previously, I worked for most of a sunday debugging the code that shuai worked on. Since then we have not figured out a solution to the bug.

Lu has not been communicating nearly enough with the group, and we have not solved this bugOnce we fix a show-stopping bug we should be able to start making progress individually on the application.

I have asked Lu to work on the bugIf he is not able to fix it I will do my best with Shuai to fix the bug before the end of next week so we have a working alpha release.

5.3.2.5 2016-02-10

This week our group (Shuai and myself) finally tackled and completed the core of our project.

The biggest problem was completed, but we still have a long way to goOur biggest problem now is finishing all of the work we have ahead in the time we have available.

Strictly speaking though we didn't encounter any new problems, just fixed existing problems.

That said Zixun Lu has still not completed the benchmarking for the projectThis is a problem as we will not be able to assess the success of our project without that information.

This coming week I will try to start and finish the daemon-ization of the program, but with the progress report assignment most actual development is practically suspended.

5.3.2.6 2016-02-17

Since last week we have made minimal progress on the applicationI have begun work on demonizing our application.

Process daemonizaion was not a problem we thought out well enough so a lot of planning has to happen pretty late in the game for that part of our application to work correctlyI think I know a way to execute the idea using best practices, but so far it's not been easy.

Tomorrow (Sunday) we will meet and develop for a full day to hopefully power through some problems we've been havingI am not optimistic that we will complete anything meaningful. I hope to be proven wrong.

5.3.2.7 2016-02-24

Since last week I made progress on and eventually merged the daemon codeI also helped Shuai Peng with the Caching code.

The daemon works fairly wellIt has not been battle tested, nor has it been designed to handle errors or problems well. These have not cause any headaches yet.

This sunday Shuai and I will complete and merge the Caching codeI will begin work on the Web API. We will also demo what we have this Thursday.

5.3.2.8 2016-03-03

Since last week we have not made substantial technical progressAfter merging the Caching code Shuai and I have focused on other coursework. I personally have been researching how to write the CGI script for our application, but not as much time has been spent as I would like.

We also met with our ClientThis was productive and expectations were tempered as we have been missing deadlines. I still believe we will be able to get a working prototype completed by our deadline, however it will not be nearly as polished as I hoped and expected given the complexity of the project.

Zixun Lu has not been able to complete his portion of the project and we are as of yet far past our deadlines for the Benchmarking sprintHe has been reassigned to other parts of the project which do not require C++ coding (benchmarking, website frontend, and the system package) as this seems to be intimidating for him.

This coming week I will finish a prototype, and hopefully merge, the CGI script portion of our project as well as any setup scripts to start an apache server.

This will include:

- The actual Python CGI script(s).
- Setup scripts for installing and enabling an Apache server CGI service.
- Possibly some systemd services.

Actually, may I go on a quick tangent about how useful a systemd service can be for this project? Using systemd we can limit the CPU and RAM usage of an application, thus giving us a 'first line of defense' against runaway cacheNot that I think that is a likelyhood, however it is a nice 'stopgap' just in case. We can set the application, in the init system layer, to restart and wipe the cache whenever it reaches some threshold. A more ideal solution would be to garbage collect the cache every so often so we don't need emergency safeguards, but if it works it works!

5.3.2.9 2016-03-10

Is it already week 9? That term flew by...

Since last week we haven't made a terribly large amount of progressI've gotten the CGI script working so we can (hopefully soon) start communicating with the outside world and running jobs over the 'net.

Originally I tried implementing the CGI script using 'fastcgi' which was touted as being the way to do this, but it didn't workOur client helped out and told us to just use the regular old boring 'cgi'.

I am very close to getting this working, so I'll get that done before we need to demo.

5.3.2.10 2016-03-17

Good news! We finished the demo!

As the clock struck 10:30 I finally got the demo workingYou could upload two files and get the transformed output in your browser. It took a long time, but it was totally worth it.

5.3.3 Spring Term

5.3.3.1 2016-04-07

We encountered many problems on the way, and have many more problems to come, but the big ones related to this were:

1) CGI scripts are non-trivial to setup.
2) Error handling and debugging cgi script was non-trivial.
3) Slightly tweaking (or in some cases entirely re-writing) parts of the code was time consuming.

This coming week I am going to finish finals and take a brief vacation, but starting next term I will spend a substantial amount of time fixing up our pull codebase for Expo and making sure it is ready for the client to take overThis includes documenting the codebase, 'sanding the edges', and any tools Steven needs to own he project when we leave the project.

5.3.3.2 2016-04-14

Since last week we have completed benchmarking of our application (or at least a first draft) and implemented a prototype of parameter passing (https://github.com/XZES40/XZES40-Transformer/pull/38).

The biggest problem we encountered was that we had to hack together an environment to actually do the benchmarkingA complete application would allow us to upload header files for our benchmarked XML documents and pass parameters, but we have yet to complete these features.

This coming week I would like to complete the parameter passing and integrate it into our website interfaceI will add the web interface and form processing / sending this week.

Next week we will try to add dependency file processingThis will be similar to the parameter passing feature, but adding files to the build job instead of key=value parameters.

5.3.3.3 2016-04-21

Since last week I have begun working on passing parameters to the application through the web UI.

The hardest part of this is that I have to format the data in such a way that the input form data is sent as JSON to the end CGI scriptThis can be done with JQuery, but I'm not a frontend person so it's taking longer than the rest of my components did.

This coming week I want to merge parameters and a big docs pushWe should be code-complete by friday the 28th (my birthday!) so I'll make sure we have something presentable by then.

5.3.3.4 2016-04-28

Since last week we have made leaps and bounds.

- We (Shuai and I) fixed a lot of last minute bugs and added some much needed revised documentation.
- I made our website dynamically submit transformation jobs and load the response content.
- I learned JQuery (for the website).

We ran into a few bugs, especially around form processing and displaying results to the user on the frontendThankfully we fixed most of those bugs and by Sunday we should have all of that ironed out for Code Completion.

This coming week we are going to merge the last pull requests we have for Code Completion and begin working on our demo and written documents for the course.

As I mentioned last week, today is my birthday, so I'm going to take a day offWe got a lot done and really brought it all together at the last minute. Now we just need to put a bow on it.

5.3.3.5 2016-05-05

Since last week we completed our code and poster and submitted both of thoseWe did not work on the code for our project, favoring instead to work on other homework to get ahead for the end of the term.

We had a few hiccups with our code, trying to merge a lot of changes together at the last minuteThankfully it passed the smoke tests so we were feature complete.

This coming week I would like to add more tests to the application, however as a team we will probably not do this.

5.3.3.6 2016-05-12

Since last week we finished a progress report (ahead of schedule!) and started prepping for Expo.

No problems have been encountered this week.

Up next: Expo!

5.3.3.7 2016-05-19

Since last week we just made sure everything was ready for expo.

We didn't encounter any problems at expoThank god.

This coming week I will get the three short writings out of the way if possible.

5.3.3.8 2016-05-26

My biggest regret in the course of this project was over-engineering solutions which did not need to be madeThe biggest example of this was when I spent all of winter break outlining the code we needed for our project, creating skeleton code for the majority of our C++ work, and in the end a large swath of that was removed to get the project done.

This was a blessing and a curseI of course learned a good lesson, and had a good understanding of our project going into winter term, but it was a blow to the ego when a bunch of my freetime was wasted.

The biggest skill I've learned over the course of this project was time management, and allocating work fairlyI tend to feel an urge to over-work myself when others are underperforming. This is unfair to myself and the people under-performing. So my new skill is probably something like "tend to your own garden".

The biggest skills I can use in the future on the technical side are some nifty javascript skills and some nice apache system admin skills.

On the non-technical side I've gotten very good at managing a small team and I've learned a lot of lessons the hard wayAs mentioned above, I have a new understanding of how to allocate jobs and stick to that allocation. Don't over work yourself just to get the job done, hold those who agreed to a job accountable. Otherwise you'll pull your hair out trying to get somebody to do a job you're already doing for them.

I am glad to have contributed to the Apache Software FoundationI am also glad that this project is over.

I learned from my teammates many lessons about management, and how as a manager you should meter your expectations a lotDon't expect anything from your teammates, make everything explicit, and make sure you stick to your job.

I believe the client is satisfiedAlthough we did not know this going in, this was more or less a prototype project and to that end we definitely finished the prototype.

I will volunteer myself as a contributor for this project going forwardI want to support the Apache Software Foundation and this is a pretty simple way to do that.

At Expo we were surrounded by really exciting project so we didn't get much attention at ExpoThat said I did meet someone from Nvidia who was very excited about our project and told me to send her an email when I was looking for a job, so that was a success!

6 Expo poster

A copy of the

7 Project documentation

8 Learning: Technical

9 Learning: Personal

9.1 Zixun Lu

9.2 Shuai Peng

9.3 Elijah C. Voigt

Why we made XZES40 Transformer.

Useful, Challenging, and Open Source

This project was chosen because it provided students the opportunity to create a real-world impact, solve an interesting problem, and contribute to the Free, Libre, and Open Source Community (FLOSS).

The software may be used by individual, organization, and enterprise users to XML perform document transformations. This is a necessary and time consuming task which can now be done faster and *in the cloud*. It proved technically challenging in that it required an understanding of C programming, in-memory caching, parallel computation, application networking, and extensive library usage. The final product was shared with the FLOSS community, giving both our client, the Apache Software Foundation, and any the rest of the world a tool which is useful, usable, and free (*in more than one way*).

Applications and uses

XML is a machine readable data markup language used to store any well-formed topic. XML Stylesheets are used to express transformations of a given XML document. These are used together to store large amounts of data, and "ask" the data a question like "How many employees in this XML file took less than three days of vacation, and make more than $50,000 per year?"

XML doument transformation is used for largely utilitarian purposes in a wide veriety of fields. Police departments, corporate offices, academic institutions, and countless other organizations need and use XML documents and transformations.

The XZES40 application provides an easy to use and access platform for transforming such documents over the web.

CS CAPSTONE PROJECT: XZES40 TRANSFORMER

High Performance XML/XSLT Transformation Server

Optimized for Speed and Convenience

The XZES40 Transformer takes two files as input, an XML document and XSLT stylesheet, and a variable number of custom build parameters or dependency files, and generates a new XML document. While this type of application is not new the concept has been improved with key optimizations and made accessable over the internet via a Web interface.

The strengths of this system are are that it does not need to be installed locally on a user's system, but is still able to perform it's tasks in a timely manner.

The application uses the Apache Xalan and Apache Xerces C++ libraries to perform XML document transformations and a Python CGI script running through an Apache HTTPD server to respond to incomming requests. The transformer daemon is optimized to perform fast transformations by caching previously encountered XML documents and performing document transformations in parallel when possible. The Python CGI script communicates with the Daemon over a local network socket; multiple requests are sent to the daemon and each one is serviced in a POSIX thread. Once the transformation is complete it is sent back to the user.

The web service runs on a remote server atop existing techologies so it can be run, managed, and modified by anybody with Linux and Apache HTTPD experience.

Conclusions/Outcomes

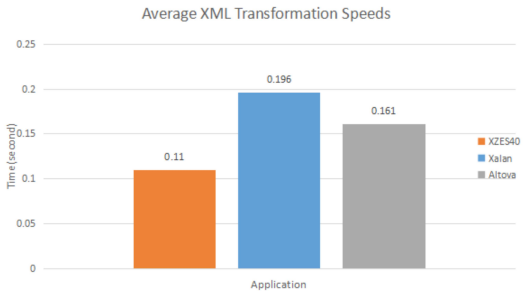
The developed application can generate a new XML file given an XML input, Stylesheet input, and custom build parameters via a Web interface, all without error.

The minimium requirements were completed, like transforming XML documents, as well as many of the optimizations and benefits we set out to complete. The most directl benefit of these being a web interface to access the transformer over the internet. In addition to this we successfully completed a simple in-memory caching system and parallel transformation of the documents. In the end we gave users an application fast enough to be competitive with enterprise software, while still being convenient to usable via the browser.

Results

The following is a table which represents the benchmark results of transforming 19 files 3 times each on the Altova enterprise XML transformer, Xalan Open Soruce transformer, and our XZES40 web Transformer. The files tested came from the Xalan project source code repository.

filename	XZES40	Xalan	Altova
builddocs.xml	0.109s	0.204s	0.187s
buildlibs.xml	0.109s	0.214s	0.193s
charter.xml	0.110s	0.221s	0.190s
ommandline.xml	0.110s	0.201s	0.178s
download.xml	0.113s	0.205s	0.169s
extensions.xml	0.108s	0.196s	0.177s
faq.xml	0.108s	0.188s	0.165s
getstarted.xml	0.108s	0.193s	0.162s
index.xml	0.107s	0.189s	0.151s
Install_save.xml	0.108s	0.212s	0.162s
install.xml	0.110s	0.208s	0.147s
overview.xml	0.110s	0.179s	0.155s
programming.xml	0.108s	0.169s	0.143s
samples.xml	0.106s	0.177s	0.135s
secureweb.xml	0.105s	0.199s	0.139s
test-faqs.xml	0.109s	0.185s	0.136s
usagepatterns.xml	0.121s	0.205s	0.147s
average	0.110s	0.196s	0.161s



The team behind XZES40 Transformer.

Sponsor

Steven Hathaway
The Apache Software Foundation
Email: shathaway@apache.org

Team Members

Elijah C. Voigt
Email: voigte@oregonstate.edu

Shuai Peng
Email: pengs@oregonstate.edu

Zixun Lu
Email: luzi@oregonstate.edu

About the Team

Elijah, Shuai, and Zixun are are fourth year Computer Science students at Oregon State University. Elijah is an Applied CS student focusing on mathematics and computer security, hoping for a career in systems engineering and games development. Shuai is a Systems CS student and is passionate about understanding and designing complex systems, looking for a career in software engineering. Zixun is a double major in CS and Business, intending to be an entrepreneur.

The team also thanks and appreciates Steven Hathaway who gave much of his time and patience to the project.