High Performance XML/XSLT Transformation Server

Spring 2017 Midterm Progress Report

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

CS 462 | CS Senior Capstone | Winter 2017

May 10, 2017

Abstract

An update on the development of the High Performance XML/XSLT Transformation Server named *XZES40 Transformer*.

Figure 1: Source: Wikimedia Commons [2]

Figure 2: Source: Apache Software Foundation [1]

1 Project Purpose

The purpose of the XZES40 document transformer is to accept input XML and XML stylesheet documents, perform an XML transformation with those files, and rerun the resulting transformation.

The application also adds the following useful features:

- Provides a user with a web interface to request XML transformations via file upload.
- Caches processed documents, reducing overhead on subsiquent transformations.
- Performs document transformations in parallel to service multiple users and requests simultaneously.

2 Project Status

2.1 Zixun Lu

2.2 Shuai Peng

For now, our team have finished our major requirement of our program. Our program has basic XML and XSLT style sheet transformation, caching old file, parallel computation, web interface, and user custom parameter passing. The command line interface and install package is option for our program, we may add it in the future. The multiply platform is hard, this may not support for our program. We have did a test to make sure that our program runs correctly, and the result of test is good. We did nice job for our program. Although we still missing some error catch, those error catch dose not impact our program, and the program still runs correct, so we may fix it in the future. To improve our program, we are revising the document. The documents will be easy to read and understand by user.

2.3 Elijah C. Voigt

3 Remaining Tasks

3.1 Zixun Lu

3.2 Shuai Peng

Here is what I should do in the future.

- we need polish our document to make it easy to read and understand.
- I need go throught what we have, and perpare for the EXPO day presentation.
- I may need do more test to make sure that it is not broken during we EXPO demo.

3.3 Elijah C. Voigt

4 Problems Encountered

4.1 Zixun Lu

4.2 Shuai Peng

The major problem since last term is that we get new requirement from the client. Cline give us a new requirement that we need pass the parameter from the website to our transformer. Thus we need add a new feature for that. However, during we were developing this function, we faced a problem is that we need additional file supporting for the XML and XSLT file transformation, and those files can not import to our transformer, because those file looks like plugin for the XML and XSLT file. We talked to our client, but he did not response for us, so we finally decided to ignore that

additional file, and we just simply install into our system. Therefore, our program is support only simple transformation, and simple parameter passing.

After we fix the additional file, we get some wired bug for the parameter passing. Elijah and me dose not a good front end developer. When we passing the parameter from command line, it works perfect, but if we change it to the website, we get bugs for the parameter passing. We did lots of test on that, and finally we fix it by check the Ajax problem. It looks like the Ajax parse the string into HTML, so our transformer can not get correct data from the website. We fix it via forcing string pass in the Ajax.

4.3 Elijah C. Voigt

5 Interesting Code

5.1 Zixun Lu

5.2 Shuai Peng

Here is our parameter passing code for the transformation. This part of code is parsing the request from the website.

```
xzes::job_t* xzes::parse_request( char* input )
{
    xzes::job_t *out = new xzes::job_t;

    std::string tmp (input);

    std::vector<std::string> tmpv = xzes::split(input,',');

    out->jid     = tmpv[0];
    out->xml.uri = tmpv[1];
    out->xsl.uri = tmpv[2];
    out->out.uri = tmpv[3];

    // Stream ends in an empty "," for buffer cruft.
    for (int i = 4 ; i+2 < tmpv.size(); i += 2)
    {
        param_t x = {tmpv[i], tmpv[i+1]};

        printf("%s:%s\n", x.key.c_str(), x.val.c_str());

        out->param.push_back(x);
    }


    return out;
}
```

5.3 Elijah C. Voigt

6 Relevant Media

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