Final Report

Archiving Transactions Towards Uninterruptible Web Service

November 2015

Summary of Work

Software Development

We developed a UWS module and its associated configurations for Apache HTTP servers to handle both the single and multiple server configurations described in the original proposal. The module was originally developed as an Apache content handler module, and was later re-developed as an extension to the Apache proxy module. The reason for the re-development was to improve the UWS performance as well as reduce the code complexity and function duplications.

We also developed an extension to the SiteStory Archive that enabled the 200-style Memento response. By eliminating the extra HTTP roundtrip during the content negotiation, we were able to reduce the latency for UWS to retrieve archived copies. As a result, our experiments showed that the CPU load for the UWS service may be reduced by about one-fourth to one-third.

Test, Verification, and Performance Tuning

The bulk of the time was spent on testing, verification, and performance tuning, based on which we have conducted multiple code revisions and iterations.

Following the development of the initial, content handler based Apache module, we tested its performance against a number of test cases. We developed a semi-automated testing system on the Amazon Cloud that generated hard to compress content, configured server clusters, then tested the performance of the developed module using both httperf and JMeter.

The results showed that although the module functioned as expected and was able to mask application server errors from the end user, its performance was poorer than expected. Further investigation revealed that the latency to retrieve archived copy from the SiteStory Archive was high, causing longer waits at the UWS module, which in turn slowed down the Apache server. The high latency may be attributed to the default, 300-style Memento protocol implementation at the SiteStory Archive, which requires two HTTP round trips to retrieve an archived copy. After consulting Herbert Van de Sompel and the Los Alamos team, we extended the SiteStory Archive implementation and enabled the 200-style Memento response.

Despite the expected performance gain, our further testing showed that the Apache server could become unstable under moderately high server loads. We traced the problem back to the way our initial UWS module handled HTTP responses. Further investigation revealed a much simpler way to implement the intended UWS functions in Apache. Instead of programming a special content handler whose main purpose was to redirect HTTP traffic, we took advantage of the existing Apache ErrorDoc directive, and modified mod_proxy to redirect the HTTP traffic in the error mode. This fully eliminated the burden from our module to read, buffer, and write content.

The tests conducted following the re-development showed that the new UWS module and configurations can sustain much higher server load than the initial design.

Pilot Deployment

We then conducted pilot UWS deployments against mirrors of two departmental production servers: http://www.fox.cs.vt.edu and http://www.dlib.vt.edu, and performed exhaustive benchmarking against both. The results showed that:

- Under normal working conditions, the UWS module imposes no measurable performance penalty on the the web server in additional to that from the Apache SiteStory module, which archives successful HTTP responses. Indeed, the UWS functions are not activated under the normal working condition.
- Under the error mode when the application servers are out of service and the user requests must be handled by UWS and routed to the SiteStory Archive, when both the SiteStory Archive and the UWS are run on identical hardware, the server performance is bounded by the SiteStory Archive. When the later slows down or times out due to high load, UWS will also be slowed down and time out.
- We then performed UWS benchmarking by replaying the servers' actual visit logs at higher and higher rates, sustaining each rate for at least 10 minutes. The results indicated that http://www.fox.cs.vt.edu broke down at about 580 requests per second, and http://www.dlib.vt.edu broke down at about 340 requests per second, both far exceeding their normal work load as indicated by the visit logs.

Documentations and Result Dissemination

Documentation of the software and its deployment is openly available at the github repository listed in the next section. We have published 5 papers related to this project and web archiving at the 2015 ACM/IEEE-CS Joint Conference on Digital Libraries (JCDL) and its co-located Web Archiving and Digital Libraries (WADL) workshop, with the full bibliography also listed in the next section. We have organized the 2015 WADL workshop, published its proceedings as a special issue of Bulletin of IEEE Technical Committee on Digital Libraries (TCDL), and will edit an International Journal on Digital Libraries (IJDL) special issue on web archiving, to be assembled in 2016.

Outcomes and Deliverables

Source code

All source code developed by this project is openly available at https://github.com/peeceeprashant/UWS. The UWS module code is released under the Apache License version 2 to stay compatible with its upstream license. The modified SiteStory Archive code is released under the BSD license to stay compatible with its upstream license.

Publications

Xie, Z., Chandrasekar, P., & Fox, E. A. (2015). Using Transactional Web Archives To Handle Server Errors. In *Proceedings of the 15th ACM/IEEE-CS Joint Conference on Digital Libraries* (JCDL 2015, pp. 241–242). June 21-25, 2015. Knoxville, TN, New York, NY, USA: ACM. http://doi.org/10.1145/2756406.2756955

Fox, E. A., & Xie, Z. (2015). Web Archiving and Digital Libraries (WADL). In *Proceedings of the 15th ACM/IEEE-CS Joint Conference on Digital Libraries* (JCDL 2015, pp. 303–303). June 21-25, 2015. Knoxville, TN, New York, NY, USA: ACM. http://doi.org/10.1145/2756406.2756934

Xie, Z., Chandrasekar, P., & Fox, E. A. (2015). A UWS Case for 200-Style Memento Negotiations. *Bulletin of IEEE Technical Committee on Digital Libraries*, 11(2), October 2015, 1 page, http://www.ieee-tcdl.org/Bulletin/v11n2/papers/xie2.pdf

Fox, E. A., Xie, Z., & Klein, M. (2015). Introduction to the Web Archiving and Digital Libraries 2015 Workshop Issue: Web Archiving and Digital Libraries 2015 (WADL 2015) Overview. *Bulletin of IEEE Technical Committee on Digital Libraries*, 11(2), October 2015, 2 pages, http://www.ieee-tcdl.org/Bulletin/v11n2/papers/intro.pdf

Xie, Z., Van de Sompel, H., Liu, J., van Reenen, J., & Jordan, R. (2015). Web Archiving Inconsistency: A Research Agenda. *Bulletin of IEEE Technical Committee on Digital Libraries*, 11(2), October 2015, 1 page, http://www.ieee-tcdl.org/mediawiki/TCDL/Bulletin/v11n2/papers/xie.pdf

Challenges

As described above, the main challenge facing this project is to develop software that meets the performance expectations of web service operators. Although there is no major obstacle to implementing the intended functionalities, its performance is bounded by the existing software, tools, protocols, and implementations upon which this project is dependent -- many of which are outside of our control. We meet these challenges by extending one of our upstream dependencies and modifying the implementation plan on the other. The resulting performance far exceeds the normal working loads of our pilot project, with further improvements left for future work.

The second challenge we encountered was from the the pilot deployments. Ryan Chase, one of the original project consultants, left the job before the project started. Due to security concerns and the complexity involved, his successor was unable to provide the support to fully replicate the VT Computer Science Departmental website. We were, however, able to negotiate the replication of http://www.fox.cs.vt.edu, its heaviest loaded sub-site, and to run the pilot deployment using its visit log. We also ran into technical difficulties to replicate http://www.coe.vt.edu, because the website content was provided as a virtual machine that our test environment could not accommodate. We circumvented the issue by replicating an alternative website, http://www.dlib.vt.edu.

Changes

We made the following changes to the originally proposed work without significantly altering the goals, methods, implementations, and deliverables.

- The original proposal intended to extend the Varnish cache using the Varnish Configuration Language（VCL) for the multi-server deployment. After further investigation, it turned out that VCL was not equipped with capabilities to intercept an HTTP response and redirect the original request to servers of our choice. Implementing such a capability would require us to modify the Varnish source code. Besides distracting the development efforts, the resulting heavily modified Varnish would hardly win acceptance from regular website operators. We therefore decided to replace the Varnish-based multi-server deployment with an Apache reverse-proxy based deployment. Although we are unaware of any survey or statistics on the reverse-proxy server market, the popularity of Apache as a front-end server provides us with high confidence that this change will not diminish the practicality and significance of this project.
- As explained in the previous section, we changed the pilot deployments from http://www.cs.vt.edu and http://www.coe.vt.edu to http://www.fox.cs.vt.edu and http://www.dlib.vt.edu.
- We expanded the scope of the software development to include the extension of SiteStory Archive, one of our upstream dependencies. The purpose of this effort is to improve the performance of our software.

Future Work

We will further investigate methods to improve the UWS performance, and publish results from this project. We have issued the initial call for papers for the IJDL Special Issue on Web Archiving, with the bulk of reviewing and editing work starting in 2016.