A UWS Case for 200-Style Memento Negotiations

Zhiwu Xie, Prashant Chandrasekar, Edward A. Fox
Virginia Tech
Blacksburg, VA 24061,USA
{zhiwuxie, peecee, fox}@vt.edu

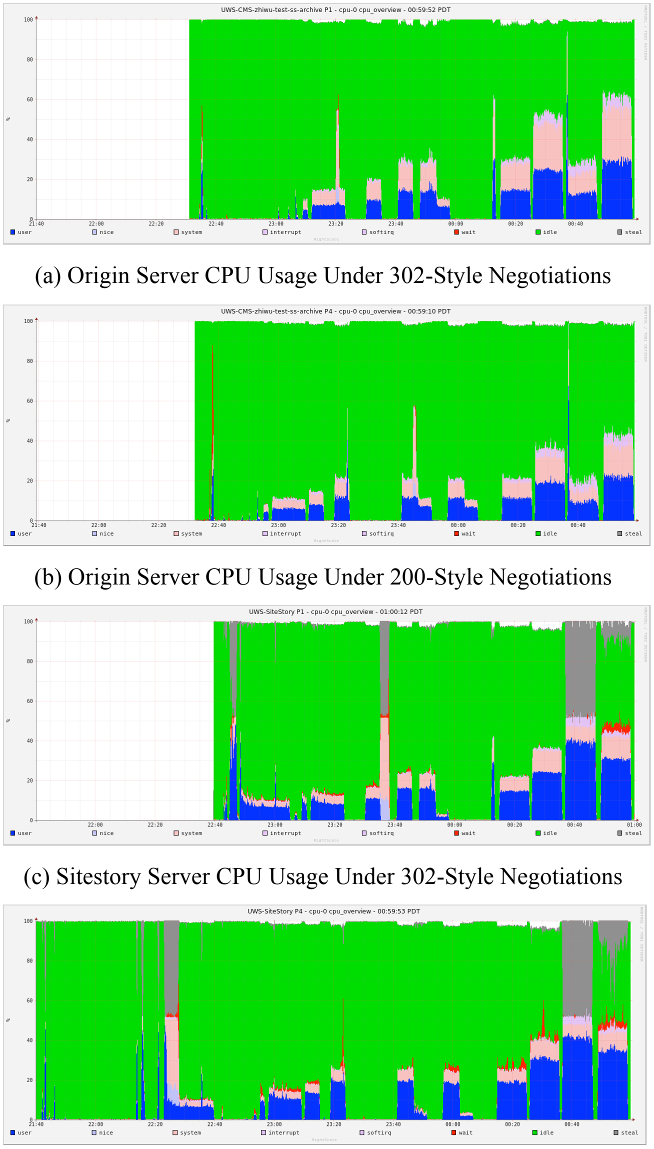
Uninterruptible web service (UWS) [1] is a web archiving application that handles server errors using the most recently archived representation of the requested web resource. The application is developed as an Apache module. It leverages the transactional web archiving tool SiteStory, which archives all previously accessed representations of web resources originating from a website. This application helps to improve the website's quality of service by temporarily masking server errors from the end user and gaining precious time for the system administrator to debug and recover from server failures. By providing value-added support to website operations, we aim to reduce the resistance to transactional web archiving, which in turn may lead to a better coverage of web history.

UWS retrieves the most recently archived copy from the SiteStory archive through the Memento protocol. Typical Memento implementations, SiteStory included, employ the 302-style Datetime negotiation as specified in RFC 7089 [2]. The Memento protocol also allows 200-style negotiation, which eliminates the second round-trip HTTP request/response. Instead of presenting to the client a list of options and letting the latter choose the applicable response, in a 200-style negotiation the server chooses the response directly without consulting the client. Such a negotiation style is useful when consulting the client is either unnecessary or expensive. In typical UWS use cases, the origin server and the SiteStory archive are located in separate servers connected in a LAN or WAN. The latency between them is therefore not negligible. The 302-style content negotiation requires two round-trips, not only doubling the latency but also causing the Apache UWS process to wait. This may potentially make the origin server the bottleneck of the request/response chain. On the other hand, although the SiteStory archive holds multiple Mementos per Original Resource and new Mementos are constantly added to the archive, we only retrieve the most recently archived Memento. We thus have a case where 302-style Datetime negotiation is both expensive and unnecessary.

In light of the UWS use case, we extended SiteStory archive implementations to handle Memento requests with this new pattern, and conducted experiments to compare it with the 302-style negotiation. As shown in Figure 1, the results show that with the 200-style negotiation we can reduce the origin server load by one-third to one-fourth without significantly increasing the load on the SiteStory archive.

Acknowledgments

This work was supported in part by the Web Archiving Incentive Awards, funded as part of Columbia University Libraries' 2013 Mellon Grant for Collaborations in Web Content Archiving.



(a) Origin Server CPU Usage Under 302-Style Negotiations

(b) Origin Server CPU Usage Under 200-Style Negotiations

(c) Sitestory Server CPU Usage Under 302-Style Negotiations

(d) Sitestory Server CPU Usage Under 200-Style Negotiations

Figure 1. Comparing 302- and 200-Style Memento Negotiations

REFERENCES

[1] 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*. 241-242.

[2] Van de Sompel, H., Nelson, M. and Sanderson, R. 2013. HTTP Framework for Time-Based Access to Resource States-Memento. IETF RFC 7089.