Web Archiving Inconsistency: A Research Agenda

Zhiwu Xie[1], Herbert Van de Sompel[2], Jinyang Liu[3], Johann van Reenen[4], Ramiro Jordan[4]

[1]Virginia Tech
Blacksburg, VA 24061
zhiwuxie@vt.edu

[2]Los Alamos National Laboratory
Los Alamos, NM 87545
herbertv@lanl.gov

[3]Howard Hughes Medical Institute
Ashburn, VA 20147
liuj@janelia.hhmi.org

[4]University of New Mexico
Albuquerque, NM 87131
{jreenen,rjordan}@unm.edu

Scaling web applications usually boils down to a tradeoff between consistency and latency. Very large web operations typically favor low latency, hence purposefully sacrifice strict consistency in the sense of serializability. By definition, the breakdown of serializability may cause the web applications to disseminate, albeit ephemerally, inaccurate and even contradictory information. If captured and preserved in the web archives as historical records, such information will degrade the overall archival quality. Despite its near omnipresent in the popular web, such relaxation in data consistency is not widely reported nor thoroughly studied by the web archiving community.

A prior study [1] provided tools to estimate the theoretic bounds of data inconsistency. However, such estimations require inside knowledge of the specific system used to build the web service, which is rarely available to the public. In addition, this estimate only bounds the inconsistency stemming from key-value pair replication, and does not take into consideration its compounding introduced by the application logic and the other sources of inconsistency from the web stack, such as caching. The exemplary inconsistency bounds reported in prior studies [1-4] are typically too low to be discernible by human users. However, even cursory browsing experiments can easily expose the contradictory information disseminated by popular web applications such as Twitter, Facebook, and Sina Weibo, themselves considered highly valuable for long-term preservation. There exists an obvious need to bridge the gap between the seemingly harmless inconsistency indicated by the key-value store benchmarking and the obvious information contradictions as displayed in the actual web applications built on top of them.

To assess the extent to which data inconsistency impacts the archival quality, we adopted a black box approach and built a simplified feed-following application using the cloud-base key-value store DynamoDB. We then simulated its operation with synthetic workloads similar to Twitter timeline requests [5]. The results indicated that a non-trivial portion of the timeline archive may contain observable inconsistency, and the inconsistency window may extend significantly longer than that observed at the data store. In our experiment, as much as 6.27% of the responses contain observable conflicts, and on average they are observed 823 seconds after the missing messages are created. This inconsistency window is in agreement with the Twitter and Weibo browsing experience, indicating the compounding effect creates much higher inconsistency than the reported theoretic bounds.

While the simulated study described above gets us closer to better understand web archiving inconsistency, it does not directly address the specific inconsistency resulting from archiving a specific web site. By nature such inconsistency may differ from one web application to another and drifts even within the same web application when the systems, software, or the load changes. For example, the Twitter Streaming API and the Timeline API may show different inconsistency levels, and the Timeline API's inconsistency level may also drift depending on the Twitter load and other factors. Work is currently underway to start detecting and documenting these levels of inconsistency in Twitter and Sina Weibo within the NISO Altmetrics Data Quality Working Group.

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.

REFERENCES

[1] Bailis, P., Venkataraman, S., Franklin, M.J., Hellerstein, J.M. and Stoica, I. 2012. Probabilistically bounded staleness for practical partial quorums. *Proc. VLDB Endow.* 5, 8 (Apr. 2012), 776–787.

[2] Bermbach, D. and Tai, S. 2011. Eventual consistency: How soon is eventual? An evaluation of Amazon S3's consistency behavior. *Proceedings of the 6th Workshop on Middleware for Service Oriented Computing* (New York, NY, USA, 2011), 1:1–1:6.

[3] Rahman, M.R., Golab, W., AuYoung, A., Keeton, K. and Wylie, J.J. 2012. Toward a principled framework for benchmarking consistency. *Proceedings of the Eighth USENIX conference on Hot Topics in System Dependability* (Berkeley, CA, USA, 2012), 8–8.

[4] Wada, H., Fekete, A., Zhao, L., Lee, K. and Liu, A. 2011. Data consistency properties and the trade-offs in commercial cloud storage: the consumers' perspective. *Proceedings of the 5th biennial Conference on Innovative Data Systems Research (CIDR)* (Asilomar, CA, USA, 2011), 134–143.

[5] Xie, Z., Van de Sompel, H., Liu, J., van Reenen, J., & Jordan, R. 2013. Archiving the relaxed consistency web. *Proceedings of the 22nd ACM international conference on information & knowledge management*, 2119-2128.