DNS-on-CRAQ: DNS with Strong Consistency for Update-frequent Environment

Yuhan Zhou
zhouyuhan@pku.edu.cn
Peking University
Beijing, China

Wenrui Liu
liuwenrui@pku.edu.cn
Peking University
Beijing, China

Xiaolong Huang
huangxiaolong@pku.edu.cn
Peking University
Beijing, China

Chenren Xu*
chenren@pku.edu.cn
Peking University
Beijing, China

ABSTRACT

ACM Reference Format:
Yuhan Zhou, Wenrui Liu, Xiaolong Huang, and Chenren Xu. 2021. DNS-on-CRAQ: DNS with Strong Consistency for Update-frequent Environment. In *Proceedings of ACM Conference (Conference'17).* ACM, New York, NY, USA, 2 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

The Domain Name System (DNS) is a distributed database that provides a directory service to translate domain names to IP addresses. In this kind of hierarchical system, caching is an important mechanism for reducing workloads and latency, but caching incurs inconsistency between cached copies and reference records. Such cache consistency problem could be a serious concern in an environment where IP address update is frequent. To reduce inconsistency, current DNS servers employ Time-To-Live(TTL) value for each record. The server deletes the corresponding record when the TTL value decreases to zero.

Unfortunately, the traditional TTL mechanism only supports weak consistency. The client may get an outdated copy from the cache. Strong consistency, which is defined as providing the guarantee that a read to a record always sees the latest written value, cannot be achieved.

There has been some approaches to address this problem and basically they can be divided into two classes: (1) Adapt a dynamic TTL based on the update or query frequency of a record. Such approaches propose a metric for DNS update frequency(e.g. *EAI* in *ECO-DNS* [1]) and a TTL allocation scheme (e.g. *DNScup* [2]). These approaches make little modifications to exising DNS servers but their strong consistency are not fully proved. (2) Develop a novel Name Service System such as *DMAP*[5] and *Auspice* [3]. These approaches decouple the identifier and location information for a network host and reach a global consistent name service system. However they require a new global infrastructure and now-existing DNS service has to be replaced.

In this paper we propose *DNS-on-CRAQ*, a third approach which reaches strong consistency and is compatible with existing DNS service. We use *CRAQ* [4], which is a method for replicating data across multiple servers to reach consistency and a DNS server behaves like a node in a *CRAQ* chain. The following sections present the design, implementation, and evaluation of out system.

Conference'17, July 2017, Washington, DC, USA

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM…$15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

2 TECHNICAL APPROACH

The main *DNS-on-CRAQ* design are as follows.

- A node in *DNS-on-CRAQ* represents a DNS server. It can store DNS items. A DNS item consists of multiple versions of a domain name, each including a monotonically-increasing version number and an additional flag whether the version is clean or dirty. Initally, all versions are marked as clean.
- When a node receivers an update of a record, the nodes add this version to its list of the corresponding item. If the node is not the tail, it marks this version as dirty, and propagates the write to its successor. Otherwise, if the node is the tail, it marks the version as clean and notify all other nodes by sending an acknowledgement backwards.
- When an acknowledgment for an item version arrives at a node, the node marks the version as clean and delete all prior versions.
- When a node receives a query request for an item, if the latest known version of the requested item is clean, it returns this value. Otherwise, if the latest version is dirty, the node sends an ask message to the tail for the last clean version. The node then returns that version of the item.

The whole name-IP address mapping update procedure is illustrated in Figure 1. When a domain name owner request to refresh a new IP address from DHCP server, DHCP server sends a write message to the head node. After updating all DNS nodes, the head reply a message to DHCP server. DHCP server then reply an update finished to the domain name owner.

3 EXPERIMENT DESIGN

We build our system based on an open-source implementation of *CRAQ* (**https://github.com/despreston/go-craq**) and simulate DNS query process with our system.

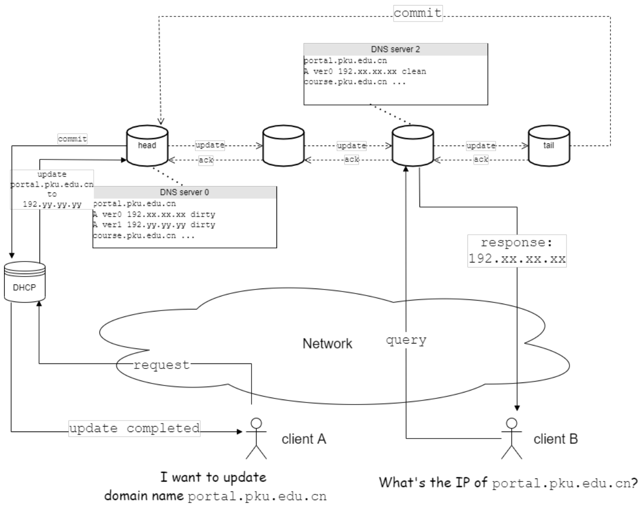

Figure 1: Name-IP mapping update procedure

We try to compare the performance of *DNS-on-CRAQ* and traditional TTL mechanism. At a high level, we are interested in read latency based on accurate query, read throughput, hit rate and overhead.

ACKNOWLEDGMENTS

We gratefully thank our mentor Chenren Xu for the opportunity he provides us to do this work. And Des Preston, who implements *CRAQ* and lets anyone us it.

REFERENCES

[1] Chen Chen, Stephanos Matsumoto, and Adrian Perrig. 2015. ECO-DNS: Expected Consistency Optimization for DNS. In *2015 IEEE 35th International Conference on Distributed Computing Systems*. 256–267. https://doi.org/10.1109/ICDCS.2015.34

[2] Xin Chen, Haining Wang, and Shansi Ren. 2006. DNScup: Strong Cache Consistency Protocol for DNS. In *26th IEEE International Conference on Distributed Computing Systems (ICDCS'06)*. 40–40. https://doi.org/10.1109/ICDCS.2006.31

[3] Abhigyan Sharma, Xiaozheng Tie, Hardeep Uppal, Arun Venkataramani, David Westbrook, and Aditya Yadav. 2014. A Global Name Service for a Highly Mobile Internetwork. *SIGCOMM Comput. Commun. Rev.* 44, 4 (aug 2014), 247–258. https://doi.org/10.1145/2740070.2626331

[4] Jeff Terrace and Michael J. Freedman. 2009. Object Storage on CRAQ: High-Throughput Chain Replication for Read-Mostly Workloads. In *2009 USENIX Annual Technical Conference (USENIX ATC 09)*. USENIX Association, San Diego, CA. https://www.usenix.org/conference/usenix-09/object-storage-craq-high-throughput-chain-replication-read-mostly-workloads

[5] Tam Vu, Akash Baid, Yanyong Zhang, Thu D. Nguyen, Junichiro Fukuyama, Richard P. Martin, and Dipankar Raychaudhuri. 2012. DMap: A Shared Hosting Scheme for Dynamic Identifier to Locator Mappings in the Global Internet. In *2012 IEEE 32nd International Conference on Distributed Computing Systems*. 698–707. https://doi.org/10.1109/ICDCS.2012.50