ISSN (Online) 2394-2320

International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)
Vol 4, Issue 11, November 2017

Knowledge-based Secure Dynamic Cache Update For Domain Name System

[1] Sivaprakash C, [2] Arvind Kumar C S, [3] Prabha S, [4] Dr. Pauline. A
[1] Assistant Professor & Head, Dept. of ECE, Sri Sairam College of Engineering, Bengaluru,
[2] UG Student, Dept. of ECE, Sri Sairam College of Engineering, Bengaluru
[3] UG Student, Dept. of ECE, Sri Sairam College of Engineering, Bengaluru
[4] Professor & Head, Dept. of CSE, SEA College of Engineering, Bengaluru

Abstract: - **The core of DNScup (DNS cache update protocol) uses a dynamic lease technique to keep track of the local DNS name servers whose clients are tightly coupled with an Internet server. DN2IP mapping change of the corresponding Internet server, its authoritative DNS name server proactively notifies these local DNS name servers still holding valid leases. Although the notification messages are carried by the User Datagram Protocol (UDP), the dynamic lease also minimizes storage overhead and communication overhead, making DNScup a lightweight and scalable solution. Based on client query rates (or service importance to their clients), it is the local DNS name servers themselves that decide on whether or not to apply for leases (or renewal) for an Internet service. On the other side, the authoritative DNS name server grants and maintains the leases for the DNS resource records of the Internet service the major components of the DNScup prototype include the detection module, the listening module, the notification module, and the lease-track file. DNScup achieves the strong cache consistency in DNS and significantly improves its availability, performance, and scalability.**

I. INTRODUCTION

Strong cache consistency is defined as the model in which no stale copy of a modified original will be returned to clients, whereas weak cache consistency is the model in which a stale copy might be returned to clients. Therefore, without strong cache consistency among DNS name servers, it is cumbersome to invalidate the out-of-date cache entries. The inefficient and pathological DNS cache update due to weak consistency quite often causes service disruption. More importantly, three recently emerged reasons, in practice, cast serious doubt on the efficacy of weak DNS cache consistency provided by the TTL mechanism: There are many unpredictable mapping changes due to emergency situations such as terror attacks and natural disasters, in which the loss or failure of network resources (servers, links, and routers) is inevitable, and we have to immediately redirect the affected Internet services to alternative or backup sites. Maintaining DNS cache consistency is critical under such an exceptional circumstance, since people need service availability at the crucial moment. The dynamic DNS technique, which provides prompt IP mapping for a server at home or a mobile host using a temporary IP assigned by the Dynamic Host Configuration Protocol (DHCP), makes the association between a domain name and its corresponding IP address much less stable. The TTL-based DNS redirection service provided by Content Distributed Networks (CDNs) only supports a coarse-grained load balance and is unable to support quick reaction to network failures or flash crowds without sacrificing the scalability and performance of DNS. Thus, cache inconsistency poses a serious threat to the availability of Internet services. This is simply because during the cache inconsistency period, the clients served with out-of-date DN2IP mappings cannot reach the appropriate Internet servers or end hosts.

II. LITERATURE SURVEY

DNS performers of either root name servers or local DNS name servers and many scientists have studied their caching effectiveness in the past decade.

- ❖ Danzig et al. measured the DNS performance at one root name server and three domain name servers. They identified a number of bugs in DNS implementation, and these bugs and mis-configurations produced the majority of DNS traffic.

- Cranor et al. identified local and authoritative DNS name servers from large DNS traces, which is useful for locating the related DNS caches.

- Park et al. identified internal failures as a major source of delays in the Planet Lab test bed and proposed a locality and proximity-aware design to resolve the problem. They utilized a cooperative lookup service, in which remote queries are sent out when the local DNS name server experiences problems, to mask the failure-induced local delay. In their design, they considered the importance of cache at the local DNS name server for providing hared information to all local clients and avoided a design that makes the cache useless.

However, none of the previous work focuses on DNS cache consistency. DNS cache inconsistency may induce a loss of service availability, which is much more serious than performance degradation. By contrast, maintaining strong cache consistency in the Web has been well studied.

- Liu and Cao showed that achieving strong cache consistency with server invalidation is a feasible approach, and its cost is comparable to that of a heuristic approach like adaptive TTL for maintaining weak consistency.

- Yin et al proposed volume lease and its extension for maintaining Web cache consistency to reduce the cost of server invalidation and its scalability.

- Mikhail and Wills proposed Management of Objects in a Network using Assembly, Relationships, and Change Characteristics (MONARCH) to provide strong cache consistency for Web objects, in which invalidation is driven by client requests. They evaluated MONARCH by using snapshots of collected contents. The weakness of MONARCH is that it does not consider the dynamics of Web page structures. The adaptive lease algorithm has been proposed in to maintain strong cache consistency for Web contents. A Web server computes the lease duration on the fly based mainly on either the state space overhead or the control message overhead. However, in their analytical models, the space and message overhead are considered separately without gauging the possible trade-offs. Thus, the performance improvement of the adaptive lease algorithm is limited.

- Cohen and Kaplan proposed proactive caching to refresh stale cached DNS resource records in order to reduce the name resolution latency. However, the client driven pre-fetching techniques only reduce the client perceived latency and cannot maintain strong cache consistency.
- Cox et al. considered using the Peer-to-Peer system to replace the hierarchical structure of DNS name servers. For example, for a given Web server, we can search a distributed hash table (DHT) to find its IP address instead of resolving it by DNS. However, compared with conventional DNS, the main drawback of this alternative approach is the significantly increased resolving latency due to P2P routing, although the approach has a stronger support for fault tolerance and load balance.

In contrast, DNScup is an effective enhancement to the current DNS implementation, which can fix the problem in a timely and cost- effective manner. Although DNS caching does not support strong consistency, the DNS Dynamic Update mechanism maintains a strong consistency between the primary master DNS name server of a zone and its slave DNS name servers within the same zone. The DNS Dynamic Update mechanism and its enhanced secure version have been proposed and implemented to support dynamic addition and deletion of DNS resource records within a zone because of the widespread use of DHCP. According to the DNS Dynamic Update protocol, once the primary master has processed dynamic updates, its slaves will be automatically notified about these changes via zone transfers. Researchers have utilized the DNS Dynamic Update protocol to achieve end-to- end host mobility. In terms of DNS semantics, proposed DNS cache update mechanism can be viewed as an external maintaining a strong cache consistency for the Domain name system 1059 extension to the DNS Dynamic Update protocol, which makes the implementation and deployment of DNScup much easier. The required modifications and additions to the current DNS implementation are minimized.

III. EXISTING SYSTEM

In the Existing system, the local DNS in the server converts the request given by the client in to its Internet Protocol address to the same server. The clients request will search the server thoroughly to satisfy the client's request. When the request is not found within the server, it does not send any acknowledge to the client. Since the client has to wait to receive the reply from the server for a long time. Otherwise the client will search another server until the request will satisfy. This will make the waste of time. It supports only week cache consistency, in which a modified original stale copy might be returned to clients. Since the old copy of the request may resend to the client's request. Although most of the domain-name-to-IP-address mappings are infrequently changed, the current approach to coping with an expected mapping change is cumbersome.

Limitation of Existing System:

- Existing DNS only supports weak cache consistency by using the Time-to-Live (TTL) mechanism. The TTL field of each DNS resource record indicates how long it may be cached.
- Although most of the domain-name-to-IP-address (DN2IP) mappings are infrequently changed, the current approach to coping with an expected mapping change is cumbersome.
- Weak cache Consistency is the model in which a modified original stale copy might be returned to clients.
- Existing DNS cache implementations employ different approaches in query load balancing at the upper levels.
- They suggested longer TTLs for popular sites to reduce global DNS query load.

Objective of Proposed System

- To communicate the server from client through Authoritative server.
- Authoritative server checks main servers respective to client request and response from main server to client.
- To investigate the dynamic of mapping changes, conclude that maintaining strong cache consistency is essential to prevent potential losses of service availability.

The proposed system consists of client, server and authoritative name server called middleware. The client's request will be analysis and converted domain name into Internet protocol (IP) address by authoritative name server and identifies the appropriative server for the status of active, and gets the required information for the requested client. If the server status is de-active, the middleware will send the alert signal to the client, that the server is de-active, and for try after some time.

IV. CONCLUSION

Strong cache consistency can provide fine-grain load balance, quick responsiveness to network failure or flash crowd, and end-to-end mobility, without degrading the scalability and performance of DNS. DNScup uses dynamic lease to reduce the storage overhead and communication overhead.

The major components of the DNScup prototype include the detection module, the listening module, the notification module, and the lease-track file. When the request limit is exceeded for a particular server, middleware automatically forward that request to another server. So client need not wait for long time. When a server failed another server will handle all requests. This process makes project more effective performance, more availability, much scalability and user friendly.

REFERENCES

1. IEEE Transactions on Knowledge and Data Engineering, Vol 19,No.8, August 2007.

2. D.Eastlake, Domain Name System Security Extensions, RFC 2535, Mar'99.

3.P. Mockapetris, Domain Names - Concepts and Facilities, RFC 1034,Nov 1987.

4.P.Mockapetris, Domain Names - Implementation and Specification, RFC 1035, Nov. 1987.

5. Wellington, Secure Domain Name System Dynamic Update, RFC 3007, Nov. 2000.

6.Internet System Consortium, http://www.isc.org,2006

7.Dynamic DNS Provider List, http://www.technopagan.org/dynamic/2006.

8. IRcache home page, http://www.ircache.net/2006.

9. V.Ramasubramanian and E.Sirer, "The Design and Implementation of a Next Generation Name Service for the Internet", Proc.ACM Ann. SIGCOMM-'04, pp. 331-342, Aug 2004

10. Content delivery and Distribution Networks, http://www.webcaching.com/cdns.html,2006.