L-005

P2P-based Self-Coordination of CDN Surrogates

Merdan Atajanov† Chenyu Pan† Norihiko Yoshida†

1. Introduction

Computer networks have been increased in number and complexity very fast since their first appearance. But the technology can not keep the same pace, therefore bottlenecks and degradations became casual things for the Internet. In order to provide continuous access with acceptable response time, sophisticated load balancing and fault-tolerant solutions have to be applied. Among several proposals, there is a CDN system which is used to decrease the load concentration on the original servers, while providing decent response time for the clients.

The CDN system introduces surrogate (mirror) servers which provide reliable service where users are transparently directed to the replica that can best serve their requests. The purpose of the CDN is to distribute content across the vast Internet, so it's closer to the user than the location of the original content, thus achieving two important user expectations – performance and availability, while also improving the scalability and flexibility for content providers.

Akamai is a globe-wide commercial CDN service provider [1]. In order to provide the service where content is locally accessible to the users, it takes the concept of distributed caching, thus providing users with better web experience. Akamai is the pioneer in the commercial CDN, but now there are a lot of implementations including, commercial and free ones.

The surrogate coordination in the CDN service providers is very dependent on external factors: like a network topology and location of surrogates. The most important of all is that coordination work is done manually by service administrators. Our proposal is to apply the P2P techniques in order to improve the coordination among the surrogate servers in the CDN system.

The rest of the paper is organized as follows: Section 2 provides some general information; Section 3 describes the design of our system. Finally conclusion and future work are given in Section 4.

2. Surrogate coordination

The coordination among surrogates should be self-organized and independent; links connecting surrogates should have very high-speed connection among themselves. But this is a very expensive solution for the globally widely distributed servers.

The effectiveness of a CDN system is highly dependent on the fact if surrogate server can satisfy client's request. We propose two cases in construction of the CDN system.

(1) The CDN system where all surrogate servers are identical copies of the original server. However, constructing CDN system where all servers are identical copies of the original server is an expensive solution.

(2) The second case is the system where surrogate servers store partial contents of the original server. Nowadays in computer networks latter case is very frequent one, therefore we will concentrate our attention on this case.

The problem is to decide where content is to be replicated (placed) dynamically. When the client distribution and global network topology could be known beforehand, the static replica placement might provide better performance.

Decentralized peer-to-peer location services offer a distributed infrastructure for locating objects quickly, with guaranteed success and locality. In contrast to depending on single server to locate object, a query in peer-to-peer infrastructure is passed around the network until it reaches a node that can locate that object.

A P2P technology is a cluster of interconnected nodes, which both act as a client and server. It has three principles:

- sharing of resources
- decentralization
- self-organization

Getting benefit from principles of P2P will result in some improvement of administration of widely spread servers, thus much less work for the system administrators. The CDN system will be self-constructing and routing among servers will be performed in decentralized manner.

The best choice would be the P2P system which has correlation between the overlay topology and underlying network topology, where the logical neighbors are also the physical neighbors. Our proposal is to apply Tapestry [2] overlay location and routing infrastructure into the CDN. The key difference of Tapestry from other overlay systems like Chord [3] and CAN [4] is that it builds in an explicit correlation between overlay topology distance and physical network latency. Tapestry's insertion algorithms select the nearest nodes in the network distance for each overlay hop which satisfy the criteria. As a result, overlay distances correspond to physical network distances, and local searches do not incur long network traversals.

It is a well-known fact that main drawback of the CDN systems is a difficulty in managing and administrating widely

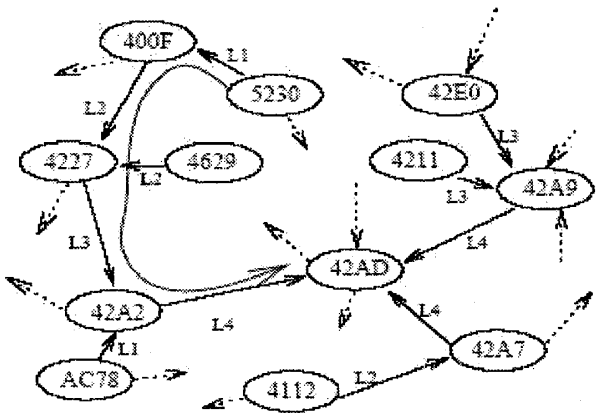

Figure 1: *The Tapestry Infrastructure.* [2]

† Saitama University

spread servers. With help of Tapestry peer-to-peer overlay constructed on the top of the CDN we expect to achieve better routing and location mechanism for the objects in the CDN. The figure 1 represents the Tapestry infrastructure. Tapestry design is inspired by the location and routing mechanisms introduced by Plaxton, Rajamaran and Richa [5]. The nodes route to the nodes one digit at a time (Ex: 4*** -> 42** -> 42A* -> 42AD). Objects are associated with particular "root". Replica's *ID* is close to the root's *node-ID* in the routing table, when the digit can not be matched an object with similar *ID* is chosen. In the original Plaxton scheme object's root node is chosen as the node which matches the object's *ID* in the greatest number of trailing bits. This method is called *surrogate routing*. The servers publish replicas by sending messages toward the root node, leaving back-pointers at each node they traverse to reach the root node. The clients are directly redirected to the closest replica, when they encounter the back-pointer in the intermediate node on the way toward the root node [6, 7].

The structure of our system will be almost the same as in P2P Tapestry except the fact that surrogates servers in the CDN system will be static; there will not be many insertions and deletions of the surrogates. According to our scheme the root node for all objects in network is *original server*. Moreover, changes (content updates) will be done at the original server and only then propagated to the surrogate servers. Since original server is the root node for all objects in the network, content updates can easily be propagated to object locations. Root node (original server) will have some constant *node-ID* (Ex. 0000), where nodes will be routed to 0000 when they are searching for the object locations.

3. Design of our system

The nodes in the CDN system will be constructed in bottom-up manner according to the P2P Tapestry topology. Same as in the Tapestry system all servers will have *node-ID* and web content data will have *globally unique identifiers* (GUIDs). The location and routing of the object will be made as in the Tapestry infrastructure. Original server will be the root node for all surrogate servers in the network. All update messages and statistics will be carried out according to this topology.

At first client is redirected to the closest X server by DNS. If a server has the content requested by the client, then it replies. If the content is not available at X server, then X server looks for the server that has the content using the Tapestry mechanism. The root node for every object in the network is the original server, so X will route to the original server (root node). On the way to the root node if it encounters the back-pointer to a replica location, it will redirect the query to the replica location. Otherwise the object will be retrieved from the root node, because original server stores every object in the network. X server stores this content and replies to the client.

The servers will be composed of two parts: P2P part and CDN part. The P2P part is responsible for decentralized routing and location mechanism of nodes within the CDN system. Our implementation follows Tapestry with some modifications and extensions. The CDN part is responsible for content related issues like content storage, update, etc. as servers in ordinary CDN systems.

3.1 DNS redirection

When a client tries to access the CDN for the first time, it will be redirected by the DNS. This DNS has supplementary function, some kind of query processor, which is responsible of redirecting clients to the closest (appropriate) servers.

We think that our research will be employed in the IPv6 environment. We expect to take advantage from the IPv6 hierarchical addressing and routing infrastructure. When the DNS for the first time gets an encounter with the URL of the CDN site, it has no IP address match for it. Therefore it has to request it from the upper level DNS. Local DNS of the original server will give response to such kind of the requests with

Hostname ----- Multiple IP addresses

where multiple addresses are IP addresses of the CDN system. The client's local DNS updates its entry for the CDN URL and processes these multiple IP addresses with the query processor evaluating the most appropriate server for the client.

Redirection procedure is handled by TENBIN DNS [8, 9]. TENBIN is the general DNS with query preprocessor. The query preprocessor selects an IP address for the hostname using some server selection policies. The best feature of TENBIN DNS is that it imposes no need for the modification of the working DNS network. It is widely used in Japan in several projects like [10, 11], and already proved itself. With the IPv6 addressing policy, redirection of the clients to the most appropriate server will be a very easy task for the TENBIN selection mechanism.

4. Conclusion and future work

In this paper, we explore the surrogate coordination in the Content Distribution Networks. Our proposal is to apply Tapestry P2P overlay location and routing infrastructure to improve the surrogate correlation in the CDN system. A slightly modified Tapestry infrastructure tends to improve the routing among surrogates in the CDN. Moreover, it has a big role in dynamic placement of replicas in the CDN. Our future studies include evaluating the effectiveness of the Tapestry infrastructure in Content Distribution Networks.

References

[1] Akamai Technologies, Inc. http://www.akamai.com/

[2] Ben Y. Zhao, Ling Huang, Jeremy Stribling, Sean C. Rhea, Anthony D. Joseph, and John D. Kubiatowicz. Tapestry: A Resilient Global-Scale Overlay for Service Deployment. *IEEE Journal on Selected Areas in Communications,* pages 41-53, Vol. 2, NO. 1, January 2004.

[3] Stoica I., Morris R., Karger D., Kaashoek M.F., and Balakrishnan H. Chord: A scalable peer-to-peer lookup service for Internet applications. In proceddings of SIGCOMM (August 2001), ACM. http://www.acm.org/sigs/sigcomm/sigcomm2001/p12.html

[4] Ratnasamy S., Francis P., Handley M., Karp R., and Schenker S. A scalable content-addressable network. In

proceedings of SIGCOMM (August 2001), ACM.
http://www.acm.org/sigs/sigcomm/sigcomm2001/p13-ratnasamy.pdf

[5] C. Greg Plaxton, Rajmohan Rajaraman, and Andrea W. Richa. Accessing nearby copies of replicated objects in a distributed environment. In Proceedings of ACM SPAA. ACM, pp 311-320 (June 1997).

[6] Ben Y. Zhao, John Kubatowicz, and Anthony D. Joseph. Tapestry: An Infrastructure for Fault-tolerant Wide-area Location and Routing. U. C. Berkeley Technical Report UCB/CSD-01-1141, April, 2001.
http://www.cs.berkeley.edu/~ravenben/publications/CSD-01-1141.pdf

[7] Yan Chen, Randy H. Katz and John D. Kubiatowicz. Dynamic Replica Placement for Scalable Content Delivery. In Proceedings of 1st International Workshop on Peer-to-Peer Systems (IPTPS 2002), March 2002.
http://www.cs.rice.edu/Conferences/IPTPS02/184.pdf

[8] Toshihiko Shimokawa, Norihiko Yoshida, Kazuo Ushijima. DNS-based Mechanism with Pluggable Selection Policies, Trans. IEICE, Vol. J84-D-I, no.9, pp.1396-1403 (2001) (in Japanese)

[9] Toshihiko Shimokawa, Yuichi Koba, Ikuo Nakagawa, Bunji Yamamoto, and Norihiko Yoshida. Server Selection Mechanism using DNS and Routing Information in Widely Distributed Environment. Trans. IEICE, Vol. J86-B, no.8, pp.1454-1462 (2003) (in Japanese)

[10] "Ring Server" Project
http://ring.aist.go.jp/index.html.en

[11] "Live Universe" Project http://www.live-universe.org/