LL-005

Design of Adaptive Network against Flash Crowds

Chenyu Pan† Merdan Atajanov† Toshihiko Shimokawa‡ Norihiko Yoshida†

1. Introduction

Today's Internet periodically suffers from flash crowds, a phenomenon triggered by a sudden, unanticipated surge in the volume of request rate to a particular website, often causing the server temporarily unreachable.

To address the problem, a number of researches has dedicated in finding appropriate ways. Efforts include solutions using CDN [1, 2], P2P overlay [3, 4] or web caching [5, 6, 7]. However, some of these solutions are expensive and some may work fine only during flash crowds but have overheads which cannot be neglected under normal network condition.

In this paper, we present a design idea of a self-tuning adaptive network, called FCAN (Flash Crowd Adaptive Network), which dynamically transits the architecture between anti-flash-crowds status and non-flash-crowds status as a possible approach to minimize the overheads. FCAN employs an Internet-wide infrastructure of cache proxy servers to perform peer-to-peer functions in dealing with the flash crowds effects and get it out of the way when normal client/server architecture works well.

The rest of the paper is divided as follows: Section 2 takes a closer look at the nature of flash crowds. Section 3 presents the overall design in detail. We overview its implementation issues in Section 4, and discuss some considerations in Section 5. Related works are described in Section 6, and the last is the conclusion.

2. Characteristics of Flash Crowds

Through the traces conducted in previous researches [2, 4, 6], we conclude some significant characteristics of flash crowds as listed below.

1. Sudden events of great interests, whether planned or unplanned, such as links from popular web sites (i.e. Slashdot effect) or breaking news stories (ex. Sep. 11th terrorist attack), trigger flash crowds.

2. Requests volume for the popular objects increases dramatically to tens or hundreds times more than normal, which is far beyond the capacity of normal web servers and pushes servers' availability down close to 0%.

3. The increase of the request rate is dramatic but relatively in short duration. Thus, traditional over-provision to handle the peak load may result servers stay practically idle most of the time.

4. Requests volume increases, while rapid, is far from instantaneous. In play-along [2] cases, the rate increase occurs gradually over the 15-minute interval. This gives the time for a system to detect a flash crowd before it reaches its peak.

5. CPU resources as well as the network bandwidth are the primary constraints bottleneck during a flash crowd. We must observe not only the server load but also the whole network performance.

6. The distribution of requested objects is Zipf-like. The number of clients is commensurate with the request rate. This is a big difference to rule out the DoS attack from flash crowds [2].

7. A small number of objects, less than 10%, is responsible for a large percentage of requests, more than 90%. This is a promising result indicating that caching of these documents might be a possible solution.

8. Over 60% of objects are accessed only during flash crowd. It implies normal web caches may not have these objects at the beginning of the flash crowd.

3. Design of FCAN

Generally speaking, current possible solutions against flash crowds can be divided into three categories: server-layer, client-layer and intermediate-layer solutions, according to a typical architecture of network.

Server-layer solutions are straight-forward but costly approaches. They extend object's availability either by over provision of the server and network based on peak demand or by CDN to increase server locations in advance.

An alternative is to let the clients share the popular objects among themselves, forming client-side P2P overlay. However, this kind of solutions is not transparent to the end users and remains low efficiency when the demands of hot objects decrease [3].

Our design is an intermediate-layer solution. We focus on a P2P-based cache proxy server layer. Hot objects requested during a flash crowd are cached in this layer and delivered to end users after conducting P2P searches inside the layer. Besides, to minimize the overheads caused by P2P functions, FCAN tunes the network to be adaptive by invoking P2P mode only when the C/S mode fails to fulfill the increasing requests. Three advantages can be seen from this schema.

1. No need of participations from end users. All operations are transparent to the end users.

2. No extra hardware investment. Cache proxies are widely deployed on the Internet. We put special wrappers on these already existed resources to avoid unjustified hardware over provision.

3. Easy control and management. Cache proxies are managed by network administrators. It must be easier to deploy P2P functions compared with uncontrollable and heterogeneous clients of the end users.

3.1 Basic Operation

Figure 1 illustrates how FCAN alleviates the symptoms associated with flash crowds. The web server and cache proxy servers belonging to the system are called Member Server and Member CP, shown in deep color. There exist other clients who connect to the web server directly or through common cache proxy.

In peaceful time, the typical client/server architecture satisfies most of the client requests. Each Member

†Saitama University
‡Kyushu Sangyo University

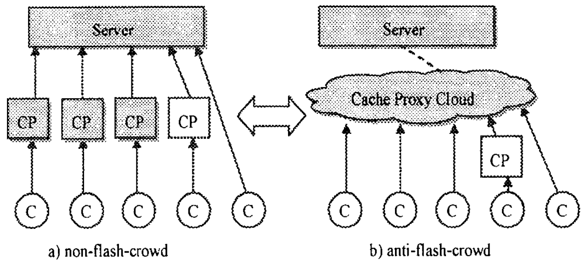

Figure 1: Changing Architectures of FCAN

Server and Member CP do little more than what normal ones do. Once a flash crowd comes, Member Server detects the increase in traffic load. It triggers all Member CPs to form P2P overlay, the cache proxy cloud shown in Figure 1 (b), through which all requests are conducted instead of bugging the origin server. Moreover, all subsequent client requests are redirected to this cloud by DNS redirection automatically. In the following subsections, we present the design detail of the P2P-based cache proxy cloud, DNS redirection and dynamic transition.

3.2 Cache Proxy Cloud

Cache proxy cloud is a quick-formed P2P overlay conducted by all Member CPs. A quick-formed P2P overlay has the features of being simple and lightweight. It can be quickly started and stopped with the change of the network traffics.

Currently P2P system has evolved from first generation to second generation. The first generation P2P systems use simple scoped search to locate objects, thus an object is equally likely to be available at any node within the P2P overlay. In contrast, the second generation systems, using a variety of dynamic hashing tables, assign each object to a particular set of nodes in the overlay. Put in other words, the second generation systems should be well-organized.

However, the contents a Member CP caches are determined randomly by client requests. They cannot be pre-determined or well-organized with hash tables in advance. Moreover, although the second generation systems save considerable traffic used for searching unpopular objects (the level of searching efficiency is $O(\log n)$ vs. $O(n)$, against the first generation systems), mathematical and simulation analysis in [8] shows that the searches of the first generation P2P systems can be designed to have low expected traffic requirements and low latency when searching for objects that are the interest of a flash crowd. Therefore, a lightweight and simple P2P system with the features of first generation is more suitable for FCAN in this context.

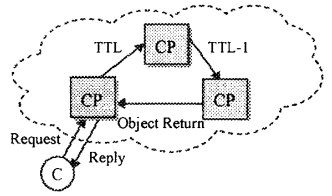

Figure 2: Scoped Search

In FCAN, each Member CP is primarily a regular cache proxy server during its normal mode of operation. It serves a fixed set of users in usual time but serves the requests arriving at the cloud from any user during the anti-flash-crowd status.

Through normal caching processes, Member CPs notify their existence to Member Server and Member Server is able to send/restore the alarm of a coming flash crowd to the corresponding Member CPs. Once a Member CP receives the alarm, it begins to find nearest neighbors using received neighbor list from Member server to form a self-organized P2P overlay and is willing to conduct the requests either from client users or neighbor peers. If the requested object is found, it returns the object to the requester, otherwise delivers the scoped search query to its neighbors with a decreased TTL time. (See figure 2)

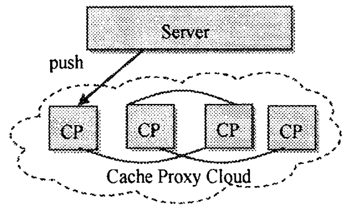

Figure 3: Push Service

We also notice that over 60% of objects are new to a flash crowd, they may not been cached at the beginning stage. Thus, we employ a "push" service (see figure 3) as a supplementary to improve the P2P search efficiency. Member Server has its file access references and trace histories of Member CPs. When there is a flash crowd for specified objects which have not been accessed by Member CPs, it pushes these objects to the P2P cloud by connecting to just one or two Member CPs. The delivered objects will soon be propagated in the cloud because of the P2P functions and the high demands.

3.3 Dynamic DNS redirection

To protect server and network from overload, flooded requests must be redirected to the cache proxy cloud.

A front-end load balancer is a basic mechanism that the Internet uses to realize the client routing. It intercepts all communications from the clients and relays them to the appropriate destination. However, increasing congestion on network paths suffering from flash crowd impedes the client requests reaching the load balancer. It also incurs "a single point failure" problem once the load balancer collapses. Hence, what we need is an approach that requires little participation of the server side.

Dynamic DNS redirection helps us to achieve the goal. It gives out the address of the cache proxy cloud instead of the origin server address when a client tries to resolve its name through a DNS server. The address of cache proxy cloud can be any address of Member CPs. Since once being redirected to one of Member CPs, client can utilize the whole P2P-based cache proxy cloud to complete the request.

As Fig 4 shows, we put special wrapper on server-side DNS that allows DNS lookup entries to be modified without shutdown of DNS service. Member Server determines when to modify the entries address and when

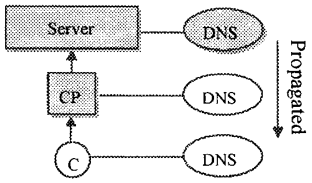

Figure 4: DNS Propagation

to restore it back. Once being modified, the new DNS entries wait for being propagated through the Internet.

3.4 Transition of Network

Here comes the question when and how to transit the network architecture?

Each Member Server and Member CP observes its load information. Member Server is responsible for detecting the forthcoming of a flash crowd. It observes the volume of server load and the slope of the tangent line for that load curve. Once a great Shock Level [6] increase exceeds to a predefined threshold in the past δ seconds, FCAN treats it as a coming flash crowd. It sends the alarm to all Member CPs to transit into anti-flash-crowd status and pushes un-cached objects with other necessary information, such as neighbor lists to the cloud. After that, it modifies the DNS lookup entries of the web site address to some of the Member CPs' and waits for new addresses being propagated through the Internet. As a result, DNS gradually redirects subsequent requests from the server to the cache proxy cloud and makes load distribution inside the cloud.

During the anti-flash-crowd status, Member server observes the traffic level on each Member CPs. Upon detecting the average load decreases under the predefined threshold in the past δ seconds, FCAN treats it as the leaving of the flash crowd. Member Server restores the DNS lookup entries and notifies all Member CPs to stop P2P search. Then, everything is back to normal.

4. Implementation Overview

We put special wrappers on normal cache proxies and DNS server to form FCAN network system.

The implementation of wrappers on cache proxies consists of two main parts. One is P2P overlay construction. As discussed in section 3.2, FCAN needs a first generation P2P system to carry out the cache proxy cloud. After a series of study, we have adopted PROOFS system presented in [3] as the best suitable schema. PROOFS is a simple, lightweight and naturally robust approach. It shuffles peer neighbors to achieve the randomness of P2P overlay and uses scoped search to deliver objects atop that overlay. Using random technology, it doesn't require heavy preconfiguration which allows quick start and stop. It locates objects efficiently since each search is randomized, even the first few hops of the new query can reach the objects. Besides, relying on randomness, PROOFS can achieve low latency delivery, even when peers dynamically join/leave the overlay with time and when there exists peers that limit their participation. However this simple searching algorithm incurs flooding problem when searching for unpopular objects. We offset this weakness by invoking PROOFS only for the duration when the searching object is heavily demanded.

Another part of Member CP wrapper is load observation. Load observation reports the real time load on Member CP to the Member Server with attempt to assist the Member Server calculating the whole traffic level of network. Load observation uses IP address instead of http domain name to communicate between Member CP and Member Server. That is to avoid the infinite loop once DNS server entries have been modified.

We adopt TENBIN system [9] to implement the wrapper on DNS server. TENBIN is our research product, and already used in practice, for example, "Ring Servers" and "Live Eclipse" projects [10]. It works as a proxy before DNS Server, intercepting the client name requests, and returning the optimal IP address under a given policy. With TENBIN, we can dynamically modify DNS lookup entries to realize client redirection and config special policy to achieve load balance among P2P cloud. The details of TENBIN and its collaboration with DNS have already been presented elsewhere [9, 10]

5. Considerations

5.1 DNS Propagation

DNS acts an important role in FCAN. We rely on dynamic DNS redirection to realize the client routing. Thus, the time for a new DNS entry being propagated through Internet is essential to FCAN. We have obtained several experiences on DNS propagation from experiments and practical use of TENBIN, and they confirm our design.

5.2 Network Deployment

FCAN needs an infrastructure support for widely distributed caching. We require the collaboration of network administrators when deploy the system across the Internet. However, we argue that gaining the cooperation from network administrators is much easier than from unknown clients.

6. Related Works

The idea of flash crowd alleviation via building self-organized P2P overlay was previously described in [3, 4, 7]. We use results from that area extensively. However, these solutions mainly rely on the client-side cooperation. They have to be deployed on user's desktop which cannot be accepted in some occasions. And some lacks of adaptive ability while facing the leaving of a flash crowd.

There have been similar works of cache proxy layer solutions against flash crowds. BackSlash [5] is a web mirroring system built on a distributed hash table overlay. It uses this overlay to cache hot objects as FCAN does. However, the mirror servers have to be invested and well-organized in advance which incurs the operation complexity and low extensibility of the system. The solution using multi-level caching [6] argues that with proper replacement algorithms a caching infrastructure designed to handle normal web loads can be enough to handle flash crowds. However, it's a pitty that currently the system lacks adaptive algorithms to handle flash crowd flexibly.

Other works related to adaptive network include: Adaptive CDN [2], focusing on the CDN network, the high cost solution, and NEWS [11], imposing congestion control on application level which scarifies some user requests to achieve a high network performance. We benefit from these researches and propose our own improved design.

7. Conclusion

This paper focuses on the basic design of an adaptive network against flash crowds, which owns the advantages of simplicity, low-cost, efficiency, flexibility and transparency to the end user. It dynamically constructs P2P overlay on cache proxy server layer to alleviate the traffic load from web server according to the network condition.

The next steps in this research involve closer study of possible solution to non-cacheable objects, detailed component designs for wrappers, better improvement of P2P search efficiency, and simulation-based evaluations.

References

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

[2] J. Jung, B. Krishnamurthy, and M. Rabinovich. Flash crowds and denial of service attacks: Characterization and implications for CDNs and web sites. In Proceedings of the 11th International World Wide Web Conference, pp. 252–262. IEEE, May 2002.

[3] A. Stavrou, D. Rubenstein and S.Sahu. A lightweight, robust P2P system to handle flash crowds. IEEE Journal on Selected Areas in Communications, Vol.22, No.1,January 2004. http://www1.cs.columbia.edu/angel/research/P2P Exper-camera.pdf

[4] V. N. Padmanabhan and K. Sripanidkulchai. The case for cooperative networking. In Proceedings of the First International Workshop on Peer-to-Peer Systems (IPTPS 2002), pp. 178–190, Cambridge, MA, USA, March 2002.

[5] T. Stading, P. Maniatis, and M. Baker. Peer-to-peer caching schemes to address flash crowds. In Proceedings of the First International Workshop on Peer-to-Peer Systems (IPTPS 2002), pp. 203–213, Cambridge,MA, USA, March 2002.

[6] I. Ari, B. Hong, E. L.Miller, S.A.Brandt and D.E.Long. Managing Flash Crowds on the Internet. Proc. MASCOTS 2003. http://www.cse.ucsc.edu/elm/Papers/mascots03 flash.pdf

[7] S. Iyer, A. Rowstron, and P. Druschel. Squirrel: A decentralized peer-to-peer web cache. Proc. PODC2002. http://research.microsoft.com/antr/PAST/squirrel .pdf

[8] D. Rubenstein and S.Sahu. An analysis of a Simple P2P Protocol for Flash Crowd Document Retrieval. Technical report, Columbia University, November 2001. http://citeseer.ist.psu.edu/rubenstein01analysis.htm

[9] T. Shimokawa, N. Yoshida and K. Ushijima. DNS-based Mechanism for Policy-added Server Selection. in Proc. Int'l Conf. on Advances in Infrastructure for Electronic Business, Science, and Education on the Internet, 6 pages, August 2000

[10] T. Shimokawa, Y. Koba, I. Nakagawa, B. Yamamoto and N. Yoshida. Server Selection Mechanism using DNS and Routing Information in Widely Distributed Environment (in Japanese).Trans. IEICE, Vol.J86-B, no.8, pp.1454–1462 (2003)

[11] X. Chen and J. Heidemann. Flash crowd mitigation via an adaptive admission control based on application-level measurement. Technical Report ISI-TR-557, USC/ISI, May 2002.