Electronic Communications of the EASST
Volume 37 (2011)

Workshops der wissenschaftlichen Konferenz Kommunikation in verteilten Systemen 2011 (WowKiVS 2011)

Preliminary Study on World-Wide Implementation of Adaptive Content Distribution Network

Yuta Miyauchi, Noriko Matsumoto, Norihiko Yoshida, Toshihiko Shimokawa

11 pages

Guest Editors: Horst Hellbrück, Norbert Luttenberger, Volker Turau
Managing Editors: Tiziana Margaria, Julia Padberg, Gabriele Taentzer
ECEASST Home Page: http://www.easst.org/eceasst/ ISSN 1863-2122

Preliminary Study on World-Wide Implementation of Adaptive Content Distribution Network

Yuta Miyauchi[1], Noriko Matsumoto[1], Norihiko Yoshida[1], Toshihiko Shimokawa[2]

[1] {yuta,noriko,yoshida}@ss.ics.saitama-u.ac-.jp
Graduate School of Science and Engineering
Saitama University
Saitama 338-8570, Japan

[2] toshi@is.kyusan-u.ac.jp
Graduate School of Information Science
Kyushu Sangyo University
Fukuoka 813-8503, Japan

Abstract: A conventional Content Distribution Network (CDN) has a static structure, therefore, it is not an effective solution to a *flash crowd*, that is a rapid increase in server load caused by a sudden access concentration. We have proposed an adaptive CDN, FCAN (Flash Crowds Alleviation Network), which changes its structure dynamically against flash crowds. In this paper, we verify FCAN on a real world-wide network. Through some experiments, we confirmed that FCAN achieves load distribution effectively.

Keywords: Content Distribution Network, DNS Redirection, Flash Crowds.

1 Introduction

A Content Distribution Network (CDN) has attracted attentions as an alternative model to Client/Server (C/S) Model. CDN is a network consisting of geographically distributed servers. However, a conventional CDN has a static structure, so that it is not an effective solution to a *flash crowd*, i.e. a rapid increase in server load caused by a sudden access concentration.

We have proposed an adaptive CDN, FCAN (Flash Crowds Alleviation Network), which changes its network structure dynamically against flash crowds [1, 2]. In this paper, we verify FCAN on a real world-wide network.

This paper is organized as follows: Section 2 provides a brief overview of flash crowds. Section 3 presents an overview and the characteristic functions of FCAN. Section 4 describes experimental evaluations of a simple prototype. Section 5 summarizes some related studies to alleviate flash crowds. Section 6 contains some concluding remarks.

2 Flash Crowds

The term “flash crowd” was coined in 1973 by a science fiction writer Larry Niven in his short novel “Flash Crowd” [3]. In the novel, cheap and easy teleportation enabled tens of thousands of

people worldwide to flock to the scene of anything interesting almost instantly, incurring disorder and confusion.

The term was then applied to similar phenomena on the Internet in the late 1990's. When a Web site catches the attention of a large number of people, it gets an unexpected and overwhelming surge in traffic, usually causing network saturation and server malfunction, and consequently making the site temporarily unreachable. This is the "flash crowd" phenomenon on the Internet.

We call the content which attracts attention of clients *hot content*. Below are characteristics and incidents of flash crowds.

Through analyses of real traces [4, 5], some significant characteristics can be concluded as follows:

- The increase of the request rate is dramatic but relatively in short duration. A flash crowd lasts as long as the attention span of the concerned audience, from hours to days, which is relatively short compared to the life span of a Web application.

- The volume of request increases, while rapidly, is far from instantaneous. In the case of the Play-along TV show, the rate increase continued for 15 min. before it reached its peak. Another case, the September 11, 2001 event, resulted in a massive load on the CNN Web site which doubled every 7 min., finally reaching a peak of 20 times higher than the normal load [6, 7].

- Network bandwidth is the primary constraints bottleneck. CPU may be a bottleneck if the server is serving dynamically generated contents. For instance, on the morning of September 11, dynamic pages on the MSNBC news Web site consumed 49.4% of "500" (server busy) error codes [8]. However, MSNBC quickly switched to serving static HTML pages, and the percentage of the error status codes dropped to 6.7%. Observations also revealed that network bandwidth became the primary constraint bottleneck, and the closer paths are to the server, the worse they are affected [8]. It is reported that modern PCs could sustain more network throughput than 1 Gbps when serving static files [9], while the network bandwidth of a Web site is typically much lower [10].

- A small number of content, less than 10%, is responsible for a large percentage of requests, more than 90%. For instance, the MSNBC traces from September 11 showed that 141 files (0.37%) accounted for 90% of the access, and 1086 files(2.87%) for 99% of the access [8]. Moreover, the set of hot contents during a flash crowd tends to be small to fit in a cache. This is a promising result implying that the caching of these 10% contents can be a solution to flash crowds. We also observe that this "10/90" rule of reference follows the Zipf-like distribution, in which the relative probability of a request for the i'th most popular content is proportional to $1/i^{\alpha}$ [11]. This property distinguishes flash crowds from attack traffic which is generated automatically by "bots".

- Over 60% of content is accessed only during flash crowds. In addition, among the 10% hot contents, more than 60% are new to being cached. For instance, 61% of contents were uncached in the Play-along case, and 82% in the Chile case [10]. This implies usual Web caches may not provide the desired level of protection. have the requested contents at the

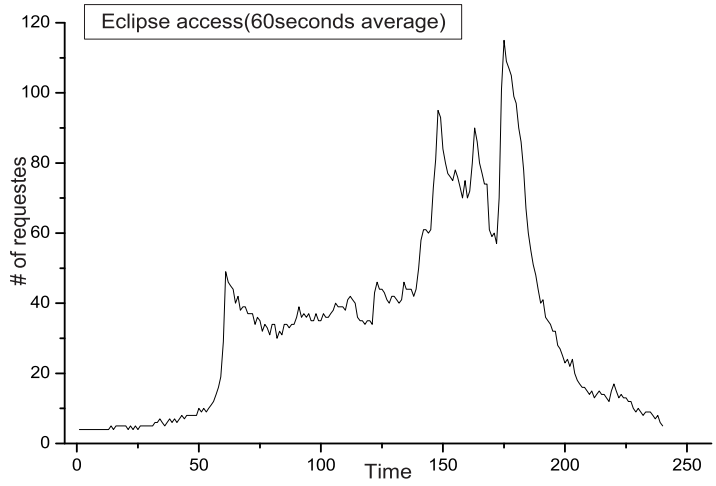


Figure 1: Accesses to the web server of “LIVE! ECLIPSE 2006”

beginning of a flash crowd. forwarded to the origin server. would be served from the caches, a large number of initial cache misses will be period of time.

- The number of clients in flash crowds is commensurate with the request rate. This feature can be used to rule out malicious requests. During a flash crowd, spikes in requested volumes correspond closely with spikes in the number of clients accessing the site. The increase in traffic volume occurs largely because of the increase in the number of clients, and most requests come from a large number of client clusters. However, because a server usually slows down during a flash crowd, per-client request rates are lower than usual. This indicates that legitimate clients are responsible for the performance of a server.

Sudden events of great interest trigger flash crowds, whether planned or unplanned. Planned ones include such as Internet broadcast, distribution of virus pattern files, and update of software. Although coming of a flash crowd can be predicted, it is difficult to estimate its scale. Unplanned ones include such as a news web site upon a major incident, and a web site referred from a popular site. We cannot predict coming of flash crowds. For example, the terrorist attack of September 11 (2001) caused flash crowds on CNN web site, as mentioned earlier. Figure 1 shows the traffic volume of web site during the solar eclipse based on a real access log provided from “LIVE! ECLIPSE 2006” [12].

3 FCAN

In this section, we present an overview and characteristic functions of our FCAN.

3.1 Overview

CDN and Peer-to-Peer (P2P) are widely used for content delivery, however, any model has not been an effective solution to flash crowds.

- A conventional CDN has a static structure, therefore it is effective if the server load stays high. However, the system gets waste of resources if the server load is low, and it cannot work if the load exceeds the prediction.
- It is difficult to predict when a flash crowd comes. Therefore, it is waste of resources that content provider enlarges its CDN in advance. Individual and non-commercial web sites can seldom get help from CDN when they suffer flash crowds.
- A P2P model has an advantage in scalability, however, it lacks reliability and security when it is composed of clients. Furthermore, it is not transparent to clients.

FCAN is an adaptive CDN which takes the form of C/S or CDN depending on the amount of accesses from clients. Specifically, in the C/S mode, a server provides contents to clients as in a traditional C/S. In the CDN mode, when the server detects the coming of a flash crowd, volunteer cache proxies in the Internet construct a temporary P2P network and provide the content on behalf of the server. These volunteer proxies are recruited in advance out of providers and organizations. In case servers in such providers and organizations suffer from flash crowds, they will be helped by other volunteer proxies. FCAN is built upon this mutually-aiding policy. Figure 2 shows an overview of FCAN.

3.2 Content Sharing

The proxy network is a pure P2P network. Therefore, it is highly fault-tolerant and scalable. Unlike traditional P2P systems, it does not include clients into the network itself in order to assure reliability and security.

The hot content is first pushed to a proxy network from the server when the network is formed. FCAN employs PROOFS [13] searching algorithm. If a cache proxy gets a hot content request from a client and does not have the content, the proxy begins the P2P search. It selects several neighbor nodes at random, and sends search queries to them. Then it replies the requested content to the client if the content is copied from any other node. During a flash crowd, a small number of content is responsible for a large percentage of requests, so that it is able to alleviate traffic congestion in proxy network and to prevent an access concentration on the server by sharing hot contents among cache proxies.

A cache proxy which received a search query decreases the time-to-live (TTL) value by one if it does not have requested content. Then it discards the query if TTL is zero, otherwise it forwards the query to random neighbor nodes. A big value of TTL causes unnecessary packet flooding in the proxy network because queries are duplicated in a exponential fashion, while a small value of TTL limits the searching scope. To assure the content hit rate, FCAN sets a small TTL at first, and increments it each time a query fails, following the technique of Expanding Ring [14]. Figure 3 shows an outline the P2P search where the TTL value and the fanout are set to 2.

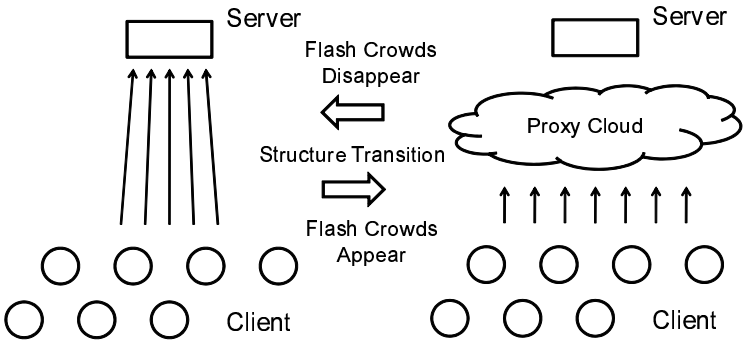


Figure 2: FCAN overview

3.3 DNS Redirection

FCAN uses DNS-based redirection, i.e. the authoritative DNS server redirects an access to an appropriate node depending on the network structure. In the C/S mode, all accesses are directed to the server, while in the CDN mode, accesses are redirected to a node in the proxy network. This change of redirection is done by dynamic update of the DNS record, and alleviates and distributes the server load in the entire network. Figure 4 shows an outline of client access redirection in FCAN.

We use TENBIN [15] for the authoritative DNS server, because it is a high-performance DNS, it allows server selection policies to be changed dynamically, and it allows DNS lookup entries to be changed dynamically.

3.4 Structure Transition

The server and the cache proxies in the proxy network always monitor the amount of accesses they receive from clients and evaluate the load of the network. The system switches to the CDN mode if all nodes' loads are higher than a certain threshold, and switches back to the C/S mode if

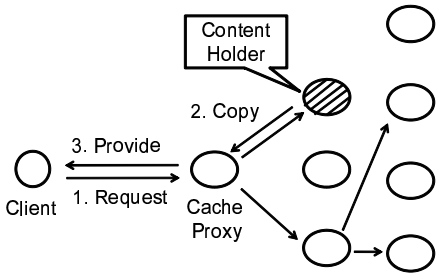
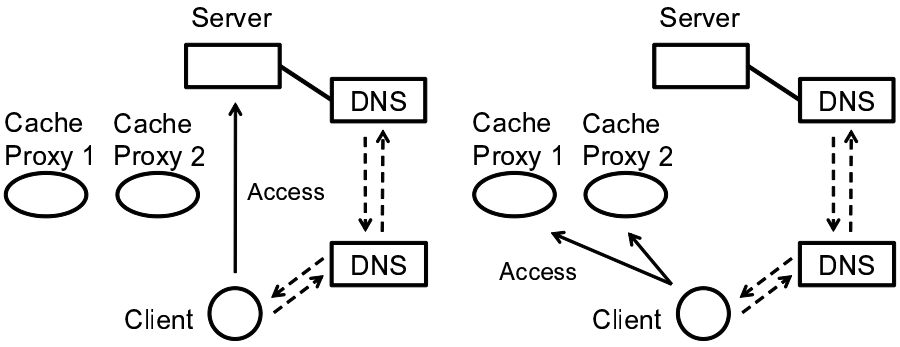


Figure 3: P2P search in proxy network



(a) In C/S mode (b) In CDN mode

Figure 4: DNS redirection

lower. Each cache proxy sends its own load information to the server periodically, and the server determines whether to perform structure transition.

We use two thresholds to prevent “thrashing” between the two mode. The threshold for transition from the C/S mode to the CDN mode is set to higher than the one for transition from the CDN to C/S.

During the flash crowd, the volume of requests increase, while rapidly, is far from instantaneous, so that there is time to perform structure transition.

In peaceful times, the conventional C/S architecture satisfies most of the client requests. A server and cache proxies, both of which comprise FCAN, do little more than what normal ones do. When a flash crowd comes, the server detects the increase in traffic load. It triggers a subset of the proxies to form an overlay, through which all requests are conducted. All subsequent client requests are routed to this overlay by DNS-based redirection. If the subset of proxies is not large enough to handle the amount of requests, new proxies are invited, and the overlay is enlarged. When the flash crowd declines, some proxies leave, so that the overlay shrinks and is eventually released.

The server-side procedure is outlined as follows: (1) Selects a subset of proxies to form a CDN-like overlay of surrogates; (2) Triggers an update of DNS records to change the look-up entries of the Web site from the server’s address to those of the proxies, so that subsequent requests are gradually redirected to the proxies along with DNS propagation; (3) Disseminates (“pushes”) the flash-crowd object to the selected proxies, because more than 60% of the flash-crowd objects are uncached prior to the arrival of the flash crowd, as mentioned above; (4) Prepares to collect and evaluate statistics for the object from the involved proxies, so as to determine dynamic reorganization and release of the overlay.

The proxy-side procedure is outlined as follows: (1) Changes its mode from a proxy to a surrogate (or, in the strict sense, a mixed mode of a forward proxy and a surrogate); (2) Stores flash-crowd objects permanently, which should not expire until the flash crowd is over; (3) Begins monitoring the statistics of request rate and load, and reporting them to the server periodically.

When the member server detects the leaving of the flash crowd, the involved proxies are dismissed one by one with the following procedure: (1) The server updates the DNS records; (2) The server notifies the proxy to be dismissed; (3) The proxy changes its mode from a surrogate to a proxy.

The CDN-like overlay transits back to the normal C/S mode when all the proxies are dismissed. They are not all dismissed at once, since the low load may be just temporary, and the system should therefore remain in the anti-flash-crowd mode for a while.

4 Experiments and Evaluation

We conducted some preliminary experiments on a real network with a prototype of our system. In this section, we present the experiment environment, and describe results and the evaluations.

In our experiments, we borrowed some hosts from universities and companies around the world, and uses them as a server node, proxy nodes, and client nodes. Table 1 summarizes the hosts' profiles and their roles in the experiments. In the rest of this section, a name of the country implies the host in the country.

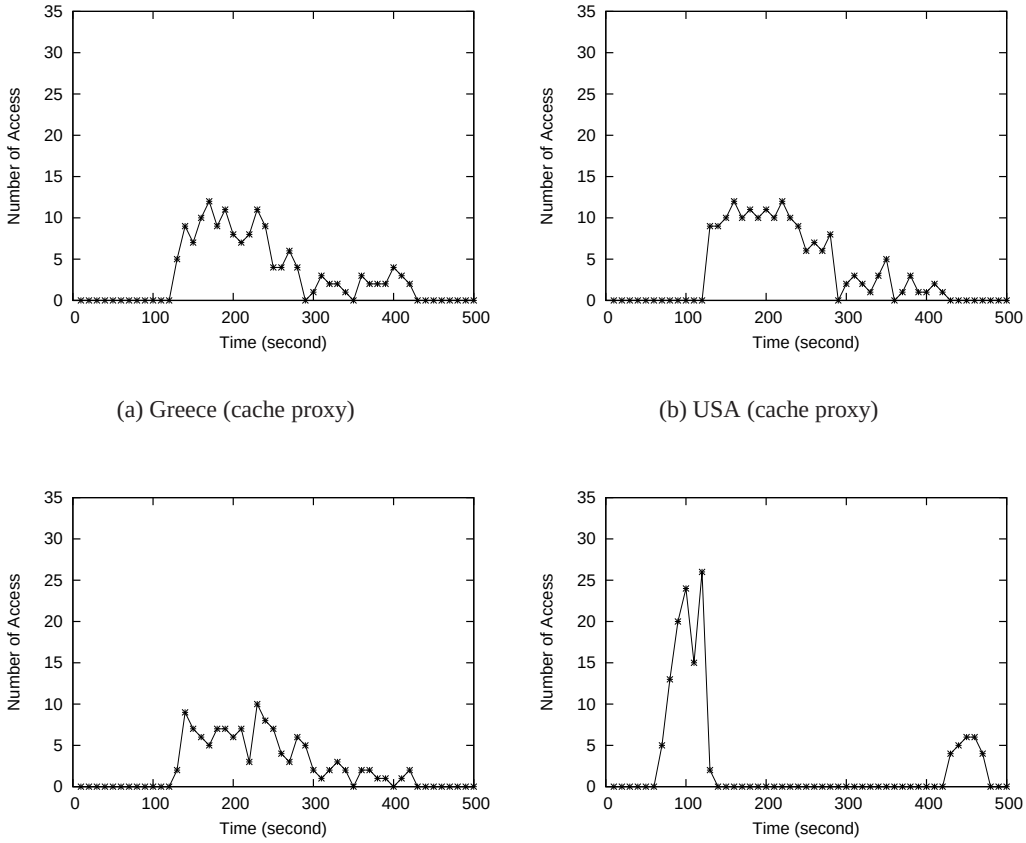
In the experiments, thresholds for load detection are defined beforehand based on some experiences. Workloads on th real Internet varies, and automatic and dynamic configuration of the thresholds is difficult. We suppose they may be configured based on the server capacity and the network bandwidth around the server.

Figure 5 shows access logs of the nodes. We compared traffic volumes on the server with FCAN and without FCAN in Figure 6. Structure transition to the CDN mode occurred at the 120th second and to the C/S mode occurred at the 420th second. The traffic volume dropped at the transition in the system with FCAN, while it continued increasing in the system without FCAN. We, therefore, confirmed that structure transition was performed as a result of the increase of client requests, and FCAN achieved dynamic load balancing.

Figure 5 also shows that the authoritative DNS server redirected accesses to appropriate nodes depending on the network structure. All the requests from clients were not directed to the server

Table 1: Hosts used in the experiments

Role	Country (Affiliation)
Server	Japan (Saitama University)
Cache Proxy	Greece (Aristotle University of Thessaloniki) USA (University of California-Irvine) Australia (The University of Melbourne)
Client	Austria (University of Innsbruck) China (Beihang University) Germany (Hochschule Furtwangen University) Japan (Saitama University)
DNS	Japan (Saitama University)



(a) Greece (cache proxy)

(b) USA (cache proxy)

(c) Australia (cache proxy)

(d) Japan (server)

Figure 5: Access transitions on each node

but to the proxies in the CDN mode, and vice versa in the C/S mode after the 420th second.

Table 2 shows the amounts and ratios of redirected accesses to each cache proxy during the CDN mode. It must be ideal that ratios of redirected accesses to the proxies would be equal, because the DNS redirected requests in the round-robin fashion. However, Table 2 shows accesses from Germany were mostly redirected to USA.

A similar phenomenon was observed regarding Austria. All accesses were redirected to the server correctly in the C/S mode. However, in the CDN mode, the first access was redirected to Australia but all the subsequent accesses were redirected to USA.

In these experiments, the time-to-live (TTL) value of the DNS records was set to zero in order to cancel the cache effect for address resolution in all the DNS servers in the world. The phenomenons above implies that there must be some DNS servers somewhere in the world which neglect the TTL value and make caches of the address resolution at their own discretion. This phenomenon was analyzed to some extent elsewhere [16].

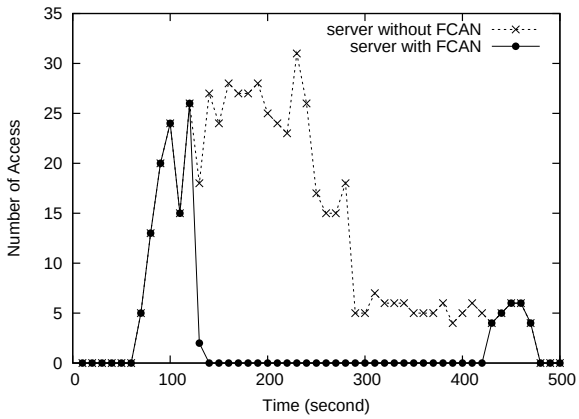


Figure 6: Comparison of access transitions

5 Related Works

In this section, we summarizes some researches to alleviate flash crowds. These researches are divided into three categories: server-layer, intermediate-layer and client-layer solutions, according to typical architectures of networks.

Server-layer Solution: Systems in this category form delivery networks on server side similar to conventional CDNs. This is a costly approach. The systems are inefficient on and difficult to deal with short-term Internet congestion. CDN with Dynamic Delegation [10] and DotSlash [17] are in this category for example.

Intermediate-layer Solution: Systems in this category let proxy servers work together for load balancing. Proxies in the system are mostly volunteers, and often less powerful than servers in the server-layer solution, however caching techniques help to alleviate server load during flash crowds by filtering out repeated requests from groups of clients which share a proxy cache. Multi-Level Caching [18], BackSlash [19] and CoralCDN [20] are in this category. FCAN is basically included in this category as well, however FCAN has extensions with some dynamic and adaptive features.

Table 2: Amounts and ratios of redirected accesses

	Cache Proxy		
Client	Greece	USA	Australia
China	55 (33.95 %)	47 (29.01 %)	60 (37.04 %)
Germany	42 (25.93 %)	86 (53.09 %)	34 (20.99 %)
Japan	55 (43.65 %)	44 (34.92 %)	27 (21.43 %)
Total	152 (33.78 %)	177 (39.33 %)	121 (26.89 %)

Client-layer Solution: Systems in this category make clients help each other in sharing contents so as to distribute the load burden from a centralized server. Clients form P2P overlay networks and use search mechanisms to locate resources. This is a costless approach. However, it is difficult to manage and control the clients, and to make the system reliable, secure and transparent to users. CoopNet [21] and PROOFS are in this category.

6 Conclusion

To handle flash crowds effectively, FCAN adopts *structure transition* which changes its network structure adaptively depending on the amount of accesses from clients. In our preceding studies, we examined FCAN only in simulation. In this paper, we aimed at examining FCAN on a real world-wide network. Through some experiments, we confirmed that FCAN achieved load balancing and handled flash crowds effectively.

We are still at a starting point toward practical implementation and promotion of FCAN. Future research directions include: (1) adaptive resizing of the proxy network on the world-wide environment, (2) more effective access redirection, possibly based on the locations of clients, and (3)adaptive distribution of not only static contents but also stream contents.

Bibliography

[1] C. Pan, M. Atajanov, M. B. Hossain, T. Shimokawa, N. Yoshida. FCAN: Flash Crowds Alleviation Network Using Adaptive P2P Overlay of Cache Proxies. *IEICE Tr. Comm.*, E89-B:4, 1119–1126, 2006.

[2] N. Yoshida. Dynamic CDN against Flash Crowds. *Content Delivery Networks* (Rajkumar Buyya, Al-Mukaddim Khan Pathan, and Athena Vakali, eds.), Springer, 277–298, 2008.

[3] L. Niven. Flash Crowd. *The Flight of the Horse*, Ballantine Books, 99–164, 1973.

[4] A. K. Iyengar, M. S. Squillante, L. Zhang. Analysis and Characterization of Large-Scale Web Server Access Patterns and Performance. *World Wide Web*, 2:1–2, 85–100, 1999.

[5] S. Lorenz. Is your Web site ready for the flash crowd? *Sun Server Magazine* 2000/11, http://www.westwindcos.com/pdf/sunserver_11900.pdf, 2000.

[6] W. LeFebvre. CNN.com: Facing a World Crisis. *Proc. USENIX Annual Tech. Conf.*, http://tcsa.org/lisa2001/cnn.txt, 2002.

[7] A. Chandra, P. Shenoy. Effectiveness of Dynamic Resource Allocation for Handling Internet Flash Crowds. *Tech. Report*, TR03-37, Dept. of Computer Science, Univ. of Massachusetts Amherst, 2003.

[8] V. N. Padmanabhan, K. Sripanidkulchai. The Case for Cooperative Networking. *Proc. 1st Int. Workshop on Peer-to-Peer Systems*, 178–190, 2002.

[9] S. Saroiu. Bottleneck Bandwidths. http://www.cs.washington.edu/homes/tzoompy/sprobe/webb.htm, 2001.

[10] J. Jung, B. Krishnamurthy, M. Rabinovich. Flash Crowds and Denial of Service Attacks: Characterization and Implications for CDNs and Web Sites. *Proc. 11th Int. World Wide Web Conf.*, 252–262, 2002.

[11] L. Breslau, P. Cue, L. Fan, G. Phillips, S. Shenker. Web Caching and Zipf-like Distributions: Evidence and Implications, *Proc. INFOCOM 1999*, 126–134, 1999.

[12] LIVE! ECLIPSE. http://www.live-eclipse.org/, 2010.

[13] A. Stavrou, D. Rubenstein, S. Sahu. A Lightweight, Robust P2P System to Handle Flash Crowds. *IEEE Journal on Selected Areas in Communications*, 22, 6–17, 2002.

[14] Q. Lv, P. Cao, E. Cohen, K. Li, S. Shenker. Search and Replication in Unstrctured Peer-to-Peer Networks. *Proc. 16th ACM Int. Conf. on Supercomputing*, 2002.

[15] T. Shimokawa, N. Yoshida, K. Ushijima. Flexible Server Selection Using DNS. *Proc. Int. Workshop on Internet 2000 (in IEEE-CS 20th Int. Conf. on Distributed Computing Systems)*, A76–A81, 2000.

[16] Y. Kamiya, F. Tanizaki, T. Shimokawa, Y. Miyauchi, N. Yoshida. Some Observations on DNS Cache Influences on Request Redirection in Dynamic CDN. *Proc. Int. Conf. on Telecomm., Networks and Systems*, 256–258, 2010.

[17] W. Zhao, H. Schulzrinne. DotSlash: A Self-configuring and Scalable Rescue System for Handling Web Hotspots Effectively. *Proc. Int. Workshop on Web Caching and Content Distribution*, 1–18, 2004.

[18] I. Ari, B. Hong, E. L. Miller, S. A. Brandt, D. E. Long. Managing Flash Crowds on the Internet. *Proc. 11th IEEE/ACM Int. Symp. on Modeling, Analysis, and Simulation of Comp. and Telecomm. Sys.*, 246–249, 2003.

[19] T. Stading, P. Maniatis, M. Baker. Peer-to-Peer Caching Schemes to Address Flash Crowds. *Proc. 1st Int. Workshop on Peer-to-Peer Systems*, 203–213, 2002.

[20] M. J. Freedman, E. Freudenthal, D. Mazieres. Democratizing Content Publication with Coral. *Proc. 1st USENIX/ACM Symp. on Networked Systems Design and Implementation*, 2004.

[21] V. N. Padmanabhan, K. Sripanidkulchai. The Case for Cooperative Networking. *Proc. 1st Int. Workshop on Peer-to-Peer Systems*, 178–190, 2002.