An Effect of Combining CDN with Reactive In-network Guidance Method

Yutaro Inaba Yosuke Tanigawa Hideki Tode
Dept. of Computer Science and Intelligent Systems
Osaka Prefecture University, Osaka, Japan
Email:{inaba@com.,tanigawa@,tode@}cs.osakafu-u.ac.jp

***Abstract*—These days, server load has been increasing because of various factors such as larger content size and flash crowd. Content Delivery Network (CDN) is widely used to deal with this problem. In CDN, replica servers are placed dispersedly in all over the Internet to achieve high QoS content delivery. On the other hand, Breadcrumbs (BC) method is proposed to implement content-oriented network framework on the conventional IP network. BC is a guidance information which is created in routers to utilize user's cache. In this paper, we operate CDN and BC framework together on IP network to improve the performance on content retrieval and acquisition. Finally, we make a CDN model and evaluate the coexisting approach through simulation. Simulation results show that our approach can greatly reduce server load.**

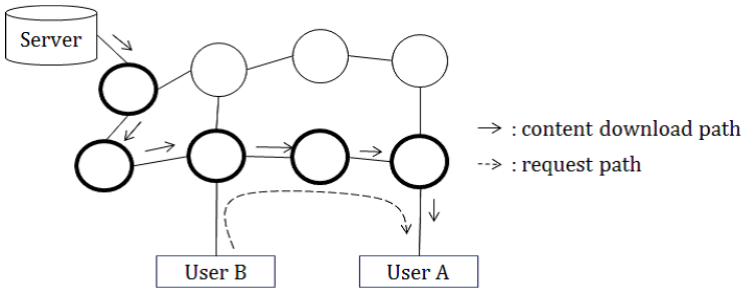

Fig. 1. BC method behavior

I. Introduction

These days, server load has been increasing because of various factors such as larger content size and flash crowd of requesting specific popular contents. CDN [1] [2] is widely used to deal with this problem. In CDN, replica servers (surrogate servers) are placed dispersedly in network. Content providers replicate their contents to surrogate servers and contents are delivered to users from there. Using CDN, users' requests can be dispersed, and thus fast and stable content delivery becomes possible. However, CDN has some disadvantages. CDN administrators need to manage surrogate servers, which are distributed in a wide area of network. This results in high management cost. As a result, using CDN also costs to some degree.

On the other hand, BC method [3] is proposed to implement content-oriented network framework on the conventional IP network. BC entry is created at each router on the download path when a content is downloaded to a user, and the user makes a cache of the content. BC entries are utilized to route a request to the user's cache. Then, the content is delivered from the user's cache. We can implement the BC method on the conventional IP network with only small change of current system [4] [5] because of it's passive and simple approach.

In this paper, we operate CDN and BC framework together on the IP network. Our main goal is to decrease server load and improve the performance on content retrieval and acquisition. In this approach, user's request may be routed by BC to not server side but another user side who has the requested cache, and then the content is delivered to the user. As a result, access to surrogate servers would decrease. Benefits of this approach are as follows:

- Reduction of cost for content providers to use CDN.
- Reduction of cost for CDN administrators to operate their surrogate servers because of less access.
- Bringing indirect benefits to users such as improvement of content providing service and less charge for the service.

II. Breadcrumbs

Breadcrumbs (BC) is a guidance information to route requests to a user's cache. BC entry is shown in TABLE I. BC entry consists of 5 items: ContentID, UpHop, DownHop, DownloadTime and RequestTime. When a content is downloaded, BC entry is created at each router on the download path. This series of BC entries is called BC trail. When a request encounters a node with BC entry for requested content, the request is routed by BC trail to DownHop node until the request reaches a user's cache. From the cache, the content is sent to the requesting user by a shortest path by IP routing. A behavior of BC method is illustrated in Fig. 1. In the figure, A BC entry is created in each router on the download path when a content is downloaded to user A. After that, user B requests the same content. The request is routed to user A by BC trail. This is why we can reduce server load by utilizing user's cache.

III. CDN model

To make a CDN model, we need to consider the following topics:

TABLE I. BC ENTRY

Attribute	Description
ContentID	Global file ID
UpHop	ID of node from which the file was forwarded
DownHop	ID of node to which the file was forwarded
DownloadTime	Time when the file passed through the node lastly
RequestTime	Time when the file was requested at the node lastly

- Where to place surrogate servers
- Which surrogate server to redirect a request
- How to make a cache

A. Where to place surrogate servers

We determine where to place surrogate servers according to Greedy algorithm [6]. The first surrogate server is placed on a router where sum of hop number from all users is minimum among all routers. The second and the subsequent surrogate servers are placed on other routers so that sum of hop number from all users to their nearest surrogate servers is minimum.

B. Which surrogate server to redirect a request

We redirect requests to a surrogate server by DNS redirection. In this method, as a response to user's name resolution request, DNS selects best surrogate server for the user and sends the server's IP address to it. In our simulation, DNS selects nearest surrogate server from the user according to pre-computed topology information.

C. How to make a cache

When a request reaches a surrogate server, if the requested content is not on the surrogate server, it requests the content to an original server. When the surrogate server receives the content from the original server, it makes a cache of the content and sends it to the user. In Fig. 2, we demonstrate basic behavior of the CDN model. The detail procedure in Fig. 2 is as follows:

1) A user sends name resolution request
2) A DNS sends best surrogate server's IP address
3) A user requests a content to the selected surrogate
    a) If surrogate server does not have requested content, it requests the content to the original server
    b) The original server sends the content to the surrogate server, and the surrogate stores the content cache.
4) The surrogate server sends the content to the requesting user

IV. EVALUATION

A. Simulation scenario

We set each parameter as shown in TABLE II. Unlike original BC method, we assume that contents are only cached in users, and routers only have BC entries to save their capacity. That makes it more realistic to implement our approach on the current Internet though routers' extra works can be remained at least.

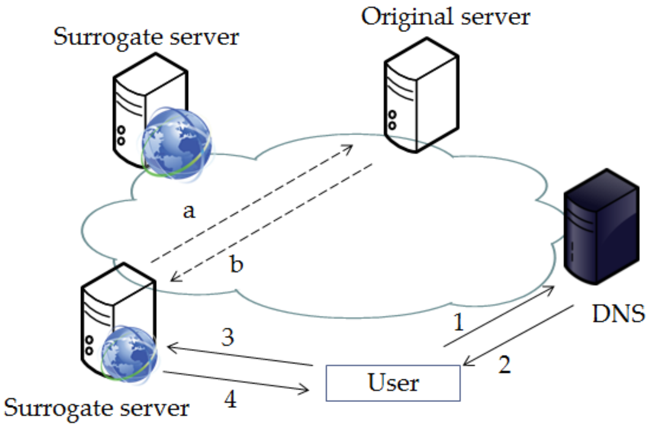

Fig. 2. CDN system

TABLE II. PARAMERTERS

Parameter	Value
# Routers	1000
# Users	5000
# Original servers	50
# Surrogate servers	5
# Contents	10000
Interval of request generation per user	2000

Note that we use BC+ method [7] instead of original BC method because BC method has a routing loop problem. The problem is that requests are transfered within specific routers forever and cannot reach where it is supposed to be.

1) Network topology: As to network topology, we assume random network based on Waxman model($\alpha = 0.3, \beta = 0.05$) [8]. Original servers and surrogate ones are connected with a router at random. All routers are connected with five different users.

2) Requests for contents: Each user requests a content at the independent, identical and exponetially distributed random interval. Each user requests a content according to Zipf-Mandelbrot with exponential cutoff distribution ($\alpha = 1.0, q = 20, \beta = -1.5^{-3}$) [9].

3) Packets and delay: We assume each packet size is constantly 1,500 Byte. A request and a control packet consist of one packet, and a content consists of 768,000 packets. This content size approximately corresponds with a content which bit rate is 5 Mbps and length is 30 minutes. In our simulation, we suppose each router has an enough processing capability, and delays are constant. It consistently takes 2.3 ms that one packet travels from a router to it's next router.

4) Compared methods: We evaluate the coexisting approach by comparing following systems:

- CDN+IP: CDN with conventional IP routing.
- CDN+BC(n): CDN with BC method (coexisting approach). We vary user's cache space n from one to five contents. Cache replacement policy follows Least Recently Used (LRU).

We use the following evaluation metrics:

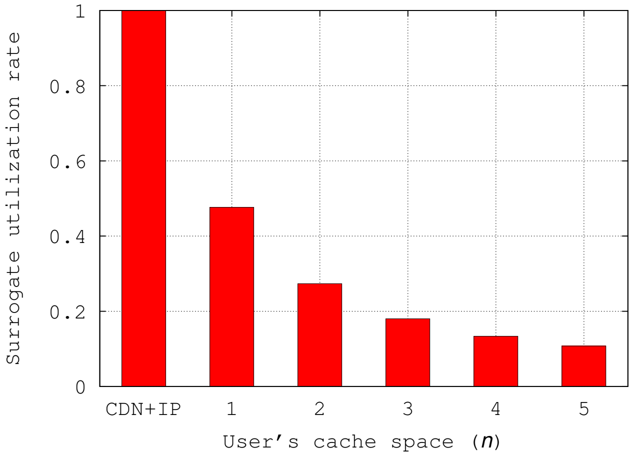

Fig. 3. Surrogate utilization rate

TABLE III. Average hop counts

Method	Request hop	Content hop
CDN+IP	5.87	5.87
CDN+BC(1)	7.03	6.28
CDN+BC(2)	7.74	6.41
CDN+BC(3)	8.11	6.32
CDN+BC(4)	8.23	6.16
CDN+BC(5)	8.25	6.01

- Surrogate utilization rate: A proportion of requests handled by a surrogate server to all the generated requests.
- Average hop counts: Average number of hops at which requests and contents are sent.

B. Results

Figure 3 shows a proportion of requests handled by a surrogate server to all generated requests. A horizontal axis represents user's cache space, and a vertical axis represents surrogate utilization rate. In CDN+IP, all requests are transfered to a surrogate server. Surrogate utilization rate, therefore, equals one. In CDN+BC, some requests are guided by BC; then a content is delivered from a user's cache, and thus, surrogate utilization rate decreases. We can see that surrogate utilization rate decreases as user's cache space increases and each user can have more contents to offer.

TABLE III represents average hop counts of request and content. In CDN+BC, both request and content hop counts become large compared with CDN+IP. Using CDN+BC, requests could be guided by BC and redirected to far user. As a result, average hop counts become large value. In particular, request hop count shows larger increasing rate because requests do not always be transfered based on minimum hop route because of BC-based routing.

V. Conclusion

In this paper, our goal is to decrease server load and improve the performance on content retrieval and acquisition. To achieve this, we presented operating CDN and BC method together. Using CDN, users' requests can be dispersed to surrogate servers. Thus, each surrogate server experiences less load, and each user can retrieve the content from the best surrogate server. This results in higher QoS. As to BC, we can utilize users' caches holding target contents by routing requests to them. BC method is very simple and passive approach. Therefore, we can implement BC method on conventional IP network without major change of current system.

Simulation results show that combining CDN with BC method results in great reduction of CDN utilization although average hops increase to some degree. We are planing to consider better request guidance method as a future work.

ACKNOWLEDGMENTS

This research was supported in part by National Institute of Information and Communication Technology (NICT), Japan.

References

[1] A. Passarella, "Review: A Survey on content-centric technologies for the current Internet: CDN and P2P solutions," Computer Communications, vol. 35, no. 1, Jan. 2012, pp. 1-32.
[2] J. Dilley et al, "Globally Distributed Content Delivery," IEEE Internet Computing, vol. 6, no. 5, 2002, pp. 50-58.
[3] E. J. Rosensweig and J. Kurose, "Breadcrumbs: efficient, best-effort content location in cache networks," in Proc. IEEE INFOCOM 2009, Apr. 2009, pp. 2631-2635.
[4] T. Yagyu, M. Kakida, Y. Tanigawa, and H. Tode, "Prototype Development of Active Distribution of In-network Guide Information for Contents Delivery," in Proc. IEICE Technical Report, vol. 111, no. 468, NS2011-211, Mar. 2012 (in Japanese), pp. 179-184.
[5] T. Yagyu, "Synergic Effects among Plural Extensions of Breadcrumbs for Contents Oriented Networks," in Proc. AFIN 2013, Aug. 2013, pp. 35-42.
[6] L. Qiu, V. Padmanabhan, and G. Voelker, "On the placement of web server replicas," in Proc. IEEE INFOCOM 2001, 2001, pp. 1587-1596.
[7] M. Kakida, Y. Tanigawa, and H. Tode, "Breadcrumbs+: Some Extensions of Breadcrumbs for In-network Guidance in Content Delivery Networks," in Proc. SAINT 2011, July 2011, pp. 376-381.
[8] B. M. Waxman, "Routing of multipoint connections," IEEE J. Selected Areas in Communications, vol. 6, no. 9, Dec. 1998, pp. 1617-1622.
[9] M. Kakida, Y. Tanigawa, and H. Tode, "Constraint-Based Distribution Method of In-Network Guidance Information in Content-Oriented Network," in Proc. NS2013-126 , Nov. 2013, pp. 61-66.