# Hierarchical Content Routing in Large-Scale Multimedia Content Delivery Network

Jian Ni, Danny H. K. Tsang, Ivan S. H. Yeung, Xiaojun Hei
Department of Electrical & Electronic Engineering
Hong Kong University of Science & Technology
{eenijian, eetsang, yshong, heixj}@ee.ust.hk

*Abstract*— **Content Delivery Network (CDN) is an intermediate layer of infrastructure that helps to efficiently deliver the ever increasing multimedia content from content providers to a large community of geographically distributed clients. Content routing is an essential component of CDN architecture. In this paper we propose a hierarchical content routing architecture for large-scale CDN, in which CDN servers perform intra-cluster and inter-cluster content routing based on a two-level hierarchical overlay network. We analyze the routing overhead and the corresponding CDN performance of different intra-cluster content routing schemes. In particular, we propose a semi-hashing based scheme for intra-cluster content routing and a content-query based scheme for inter-cluster content routing. Through qualitative analysis and simulations we show that the semi-hashing based scheme is scalable (small routing overhead), efficient (high content sharing efficiency), and flexible (adjustable parameters).**

## I. Introduction

Traditional WWW sites are changing into multimedia WWW sites and new multimedia networking applications seem to emerge everyday. With the increasing popularity of multimedia content (including traditional web content) over the Internet and the high quality-of-service (QoS) expected by the clients, Content Delivery Network (or Content Distribution Network, CDN), which derives from replicated server systems and cooperative caching/streaming systems, has recently been proposed and deployed to deliver the content from content providers to a large community of clients dispersed over different geographical areas.

CDN is an overlay network constructed from a group of geographically distributed CDN servers. It may be deployed by a CDN service provider such as Akamai [1] that partners with multiple ISPs. Alternatively, a big ISP like AT&T itself may provide CDN services and deploy CDN servers at the edge of its network, as shown in Figure 1.

CDN servers cooperate with each other and form an overlay network. When a client requests some content, the request is redirected to a nearby CDN server. This CDN server serves the client if it has the content. If not, it performs content routing to locate and deliver the requested content to the client. Therefore, the four main components of CDN architecture are: ***overlay network formation***, ***client request redirection***, ***content routing***, and ***last-mile content delivery***. Products based on this architecture particularly targeted for multimedia content streaming have been developed by companies such as SinoCDN [2]. In this paper we concentrate on content routing for on-demand multimedia content in CDN.

The rest of the paper is organized as follows: we summarize the four components of CDN architecture in section II. In section III we introduce the different content routing schemes of cooperative caching systems. In section IV we propose a hierarchical content routing architecture for large-scale CDN, in which CDN servers perform semi-hashing based intra-cluster content routing and content-query based inter-cluster content routing. We analyze and compare the performance of different intra-cluster content routing schemes using qualitative analysis and simulations in section V. We conclude the paper and present our future work in section VI.

## II. CDN Architecture

### A. Overlay Network Formation and Management

Most research ([3], [4]) constructs the overlay network with a ***flat*** topology, which does not scale well. Like ***hierarchical*** routing in the Internet, we believe that a hierarchical overlay network topology is required for a large-scale CDN to perform content routing scalably and efficiently. In order to form a hierarchical overlay network, the CDN servers first need to be grouped into clusters, either by manual configuration, or through some self-organizing scheme such as the binning scheme proposed in [5]. Then the CDN servers of the same cluster form a sub-overlay network. Each cluster selects one (or more) CDN server as the ***representative*** of this cluster. Representatives of different clusters form a higher-level overlay network. Thus a hierarchical topology is constructed.

### B. Client Request Redirection

There are several ways to redirect the request from a client to a nearby CDN server (we call this the client's ***local CDN server***), which is near the client so that content delivery takes place at the edge of the network where bandwidth is abundant. Request redirection can be ***nontransparent***, like explicit client configuration, or ***transparent***, that relies on some network elements such as L4/L7 switches, routers or DNS servers. DNS-based client request redirection schemes ([6], [7]) have recently become popular because of their simplicity and generality.

0-7803-7802-4/03/$17.00 © 2003 IEEE

### C. Content Routing

The local CDN server of a client is the client's entry point for accessing the CDN. It will initiate content routing if it does not have the requested content.

On-demand multimedia content (stored videos or movies) are normally requested by the clients ***asynchronously***. After locating the content, the local CDN server should preferably cache the content in order to serve future clients. It is not feasible for a single CDN server to cache all the content existing in the Internet because of its limited disk space; therefore, the cooperative caching scheme of the CDN servers and the content routing scheme are closely related.

Live multimedia content (Internet radio or television) cannot be cached in advance. However, since the content is ***synchronous***, a CDN server can aggregate all the requests for the same live content redirected to it, and becomes a multicast group member for that content. Multicast group members of a CDN form an overlay multicast tree that delivers the content from the original server to each member. This is known as Application Level Multicast ([8]). If a local CDN server that has not yet been a multicast group member for certain content receives a client's request for that content, it initiates multicast content routing to add itself to the multicast tree.

### D. Last-mile Content Delivery

After the local CDN server locates the requested content, it takes responsibility for delivering the content to its clients. Efficient techniques can be applied to achieve large-scale content delivery. For example, normally there are hundreds or thousands of clients requesting the same live multimedia content. The local CDN sever can use ***IP multicast*** or the ***splitting*** technique to deliver the content to the clients efficiently.

## III. Content Routing in Cooperative Caching System

There are two approaches used to distribute on-demand multimedia content to different CDN servers. One is the ***proactive*** approach as in replicated server systems. Content is replicated in the CDN servers in advance. Here content routing is easy because content locations are known beforehand. The other is the ***reactive*** approach as in cooperative caching systems. Content is cached in different CDN servers based on the 'feedback' of the clients. The reactive approach is more adaptive and efficient so it is more popular. Sometimes both approaches need to be combined to achieve a high performance. Here content routing is closely related to the cooperative caching scheme of the CDN servers.

There are several content routing schemes in the literature of cooperative caching. The most straightforward method is the query-based scheme ([9]), in which a proxy broadcasts a query for the requested content to other cooperating proxies if it does not have the content. Routing overhead includes significant ***query traffic*** and delay because a proxy must wait for the last 'miss' reply before concluding that none of the cooperating proxies has the content.

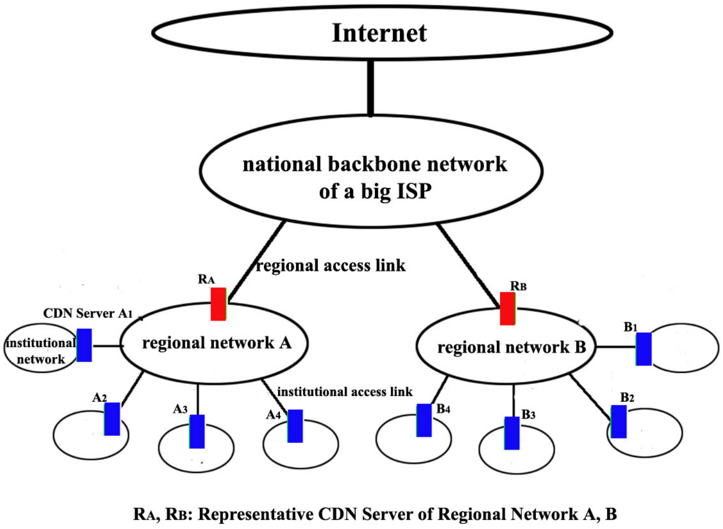
Internet
national backbone network of a big ISP
regional access link
RA
RB
CDN Server A1
institutional network
regional network A
regional network B
B1
A2
A3
A4
institutional access link
B4
B3
B2
RA, RB: Representative CDN Server of Regional Network A, B

Fig. 1. CDN Server Deployment By A Big ISP

In order to prevent flooding queries, the digest-based scheme was proposed ([10]). Each proxy maintains a content digest that includes the content information of other cooperating proxies. Once a proxy has cached/deleted some content, it notifies other proxies to update their content digests correspondingly. Therefore, a proxy knows where to route the request after checking its content digest. The major routing overhead is ***update traffic*** because update messages need to be exchanged frequently to make sure that all the cooperating proxies have the same and correct 'view' of others.

A centralized version of the digest-based scheme is the directory-based scheme ([11]), in which a directory server maintains the content information of all the cooperating proxies. Each proxy needs only to notify the directory server when an update occurs. A proxy queries the directory server when a local miss occurs. Compared with the digest-based scheme, the update traffic is greatly reduced but the directory server is a single point of failure because it handles the updates and queries from all the cooperating proxies.

A more efficient approach is the hashing-based scheme ([12], [13]). The cooperating proxies maintain the same hashing function. Based on the content's URL (or other unique identification), the addresses of the proxies and the hashing function, the request is redirected to a ***designated proxy*** for that content. Standard modulo hashing functions based on the number of the proxies are not consistent when proxies are added or deleted. Several approaches were proposed to solve this problem as in [12] and [13]. The hashing-based scheme has a small routing overhead (traffic, delay) and high content sharing efficiency but it requires that the cooperating proxies be located in close proximity because local requests are often redirected to and served by other designated CDN servers.

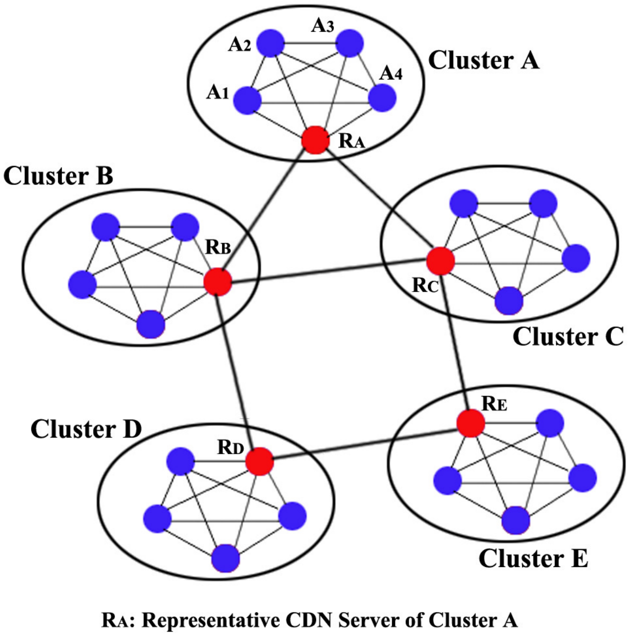

Fig. 2. Hierarchical Overlay Network Formed By CDN Servers

## IV. Hierarchical Content Routing for On-demand Multimedia Content in CDN

The overlay network formed by the CDN servers is the basis for performing content routing. Suppose a two-level hierarchical overlay network is formed as shown in Figure 2, either by manual configuration or self-organization. The CDN servers in the same cluster are close to each other. For simplicity, we assume that they are fully connected logically. The clusters are interconnected by the representative CDN servers of different clusters. The steps the CDN takes to serve a client's request are:

***1.*** Try to satisfy the client's request using the local CDN server.

***2.*** If step 1 fails, try to satisfy the client's request using a CDN server inside the cluster of the local CDN server.

***3.*** If step 2 fails, try to satisfy the client's request using a CDN server inside a nearby cluster.

***4.*** If step 3 fails, try to satisfy the client's request using the original content server.

If the local CDN server has the requested content and serves the client, we call it a ***local hit***. If not, the local CDN server initiates ***intra-cluster*** content routing. If the request is satisfied by a CDN server inside this cluster (including the local CDN server), we call it a ***cluster hit***. If not, ***inter-cluster*** content routing is performed. If the request is satisfied by any CDN server of this CDN, we call it a ***global hit***. In order to avoid retrieving the content from a remote CDN server and causing long delay, step 3 and step 4 can be performed simultaneously so that the original server is contacted earlier.

The local hit rate $H_{local}$ of a CDN is defined as the ratio between the total local hits and total requests arriving at the CDN. $H_{local}$ can be interpreted as the probability that a client's request is satisfied by the client's local CDN server. Similarly, the cluster hit rate $H_{cluster}$ and global hit rate $H_{global}$ of a CDN can be defined accordingly.

### *A. Intra-cluster Content Routing: Semi-hashing Based Scheme*

The content routing schemes described in section III are possible candidates for intra-cluster content routing in CDN. We prefer the hashing-based scheme because of its small routing overhead. In addition, for the pure-hashing based scheme, the content is distributed among the CDN servers without any duplication, so it has the highest content sharing efficiency. However, pure-hashing does not distinguish a local CDN server from other CDN servers. Local requests are often redirected to other designated CDN servers, so its local hit rate is quite low. As we will show in section V, the local hit rate is crucial to achieve a high content routing performance.

Instead of pure-hashing, we propose a ***semi-hashing*** based scheme, in which a local CDN server not only cooperates with other CDN servers using a (consistent) hashing function, but also allocates certain portion ($P_{local}$) of its disk space to cache the most popular contents for its local clients. We have designed an algorithm for a local CDN server to determine whether content $i$ is popular (for the local clients) based on the accumulative number of requests $r_i$ of content $i$ and the average accumulative number of requests $r_{ave}$ of all the popular contents that are currently cached in the local CDN server. Content $i$ is popular if $r_i > r_{ave} * \alpha$, where $\alpha$ is an adjustable parameter. In our simulations $\alpha$ was set to 1.

Upon receiving a request for some content from a client, the local CDN server performs the following distributed semi-hashing based intra-cluster content routing scheme:

```
IF I am the designated CDN server for the content
.........if I have the content, serve the client
.....else initiate inter-cluster content routing
ELSE IF the content is popular for my local clients
.........if I have the content, serve the client
.....else retrieve the content from the designated CDN
    server and then serve the client
ELSE Redirect the request to the designated CDN server
    and let it serve the client
```

### *B. Inter-cluster Content Routing: Content-query Based Scheme*

If intra-cluster content routing fails, inter-cluster content routing is performed to get the requested content either from a CDN server outside this cluster or from the original server. Here the hashing-based scheme is not appropriate because representative CDN servers of different clusters are normally geographically distributed. The digest-based scheme is also not suitable because it is extremely hard for a representative CDN server to maintain a huge and correct content digest including the content information of CDN servers of other clusters.

Suppose a hashing-based (or directory-based) scheme is implemented for intra-cluster content routing. We propose a

TABLE I

OVERHEAD OF DIFFERENT INTRA-CLUSTER CONTENT ROUTING SCHEMES

| Intra-cluster Content Routing Scheme | Intra-cluster Content Routing Traffic | | Average Intra-cluster Content Routing Delay |
|---|---|---|---|
| | Query Traffic Rate | Update Traffic Rate | |
| query-based scheme | $O(\beta(1 - H_{local})N)$ | 0 | $(1 - H_{local})D_1$ |
| digest-based scheme | 0 | $O(\beta(1 - H_{local})N)$ | 0 |
| directory-based scheme | $O(\beta(1 - H_{local}))$ | $O(\beta(1 - H_{local}))$ | $(1 - H_{local})D_{directory}$ |
| pure-hashing based scheme | 0 | 0 | 0 |
| semi-hashing based scheme | 0 | 0 | 0 |

***content-query*** based scheme for inter-cluster content routing. We use Figure 2 to illustrate the scheme. For example, the representative CDN server of cluster A, $R_A$ will query its neighbors $R_B$ and $R_C$ for the missed content. Neighbor $R_B$ replies with a ***hit*** message if it has the content; if not, it forwards the request to its own neighbors $R_C$ ($R_C$ ignores duplicate queries) and $R_D$. At the same time, based on the hashing function used in cluster B, $R_B$ queries the designated CDN server for that content in cluster B. Here $R_B$ only queries one CDN server in its cluster, but the probability of getting the content is high because, actually, all the CDN servers in cluster B are serving this request through the hashing-based scheme.

In order to avoid retrieving the object from a remote CDN server and flooding the request in the CDN, a Timeout value (if timeout, the original server is contacted) and a TTL number (when TTL decreases to 0, the query message will not be forwarded) are set for each query message.

## V. PERFORMANCE ANALYSIS AND SIMULATION RESULTS

Since there are not many choices for inter-cluster content routing, in this section we concentrate on analyzing the performance of different intra-cluster content routing schemes. We studied a cluster consisting of $N$ CDN servers, as shown in Figure 2. The average client request rate of this cluster was $\beta$ requests/minute. $H_{local}$ and $H_{cluster}$ were the local hit rate and cluster hit rate of this cluster.

### A. Intra-cluster Content Routing Overhead

A summary of the overhead of different intra-cluster content routing schemes is given in Table I. The routing traffic includes query and update traffic as explained in section III. The routing delay is the time needed for a local CDN server to locate where the requested content is (either in a CDN server inside this cluster or inter-cluster content routing is required). For any scheme, the probability of performing intra-cluster content routing is $1 - H_{local}$.

In the query-based scheme, once intra-cluster content routing is performed, the local CDN server queries other $N - 1$ CDN servers inside its cluster, so the (intra-cluster) query traffic rate is $O(\beta(1 - H_{local})N)$ and the average routing delay is $(1 - H_{local})D_1$, where $D_1$ is the average delay of locating the content after a query flooding. In digest or hashing based schemes, the local CDN server knows where to route the request after checking its content digest or hashing function; therefore, there is no query traffic and the routing delay is 0. But in the digest-based scheme an update occurs when a local miss happens (the probability is $1 - H_{local}$), so the update traffic rate is $O(\beta(1-H_{local})N)$. Other terms in Table I were similarly calculated. $D_{directory}$ is the average delay that a local CDN server takes to query the directory server. The results in Table I show that:

- Compared with query and digest-based schemes, hashing-based schemes have the smallest routing overhead and scale well with the number of CDN servers in a cluster.
- Compared with the directory-based scheme, hashing-based schemes are fully distributed without any central (directory) server.
- $H_{local}$ is an important metric for evaluating intra-cluster content routing efficiency. Similarly, $H_{cluster}$ is an important metric for evaluating inter-cluster content routing efficiency because the probability of performing inter-cluster routing is $1 - H_{cluster}$.

### B. CDN Performance

The major benefit of CDN is to save network bandwidth, reduce the original server's load, and provide better QoS to the clients. We evaluated the performance of CDN in respect of these three aspects.

If the overlay network is constructed from a CDN depicted in Figure 1: the CDN servers in the same regional network form a cluster, and we assume that the original server is outside this regional network; therefore, on average, the CDN saves $100H_{local}$ and $100H_{cluster}$ percent of the bandwidth usage of the institutional and regional access links respectively. If the physical network topology is unknown, it is impossible to calculate exactly how much bandwidth is saved; however, a qualitative analysis revealed that higher $H_{local}$ and $H_{cluster}$ indicate that more bandwidth is saved. Besides, with higher $H_{local}$ and $H_{cluster}$, the original server has less loading.

Because of the current best-effort Internet service model, it is difficult to guarantee the QoS to the clients quantitatively if content is delivered using the default Internet path. For multimedia content streaming that consumes a lot of bandwidth, we believe that a CDN server near the client (so that content delivery takes place at the edge of the network where bandwidth is abundant) provides better QoS to the client than a remote server. Higher $H_{local}$ and $H_{cluster}$ mean that more requests are satisfied by nearby CDN servers and thus relatively better QoS is received by the clients.

### C. Simulation Results

We constructed a simulation model to analyze the performance of the following intra-cluster content routing schemes:

1. ***Query, digest and directory-based schemes***. Without considering the content routing overhead, the upper bound performance of these schemes are the same, because either by querying peer CDN servers or checking the content digest (directory server), a local CDN server knows where to route the request.
2. ***Pure-hashing based scheme.***
3. ***Semi-hashing based scheme.***

The model consists of 6 CDN servers in a cluster. Each CDN server with 50Gbytes disk space can support a maximum of 300 clients simultaneously and uses the LRU-2 [14] algorithm to implement content replacement. There are three types of multimedia objects: 10,000 type-1 objects (each object is 20Mbytes and the streaming time lasts for 5 minutes), 1000 type-2 objects (100Mbytes each and 25 minutes long) and 1000 types-3 objects (500Mbytes each and 125 minutes long). We assume that the request arrival process is a Poisson process for each object of each type and the popularity of each object follows the Zipf's Law [15].

We examined the following performance metrics: local hit rate $H_{local}$, cluster hit rate $H_{cluster}$, blocking rate (the ratio between the blocked requests and total requests) and the load of the CDN servers. Here blocking means that the request cannot be served by the CDN because of the limited capacity of the local CDN server or the designated CDN server.

The results of $H_{local}$ and $H_{cluster}$ are shown in Figure 3 and Figure 4 respectively. As expected, the pure-hashing based scheme has the highest content sharing efficiency and thus the highest $H_{cluster}$. However, it does not distinguish the local CDN server from other CDN servers so its $H_{local}$ is quite low compared with other schemes. In query, digest and directory-based schemes duplicate content exists in different CDN servers, so their $H_{cluster}$ is low. $H_{cluster}$ of the semi-hashing based scheme ($P_{local} = 0.2$) is slightly less than that of pure-hashing but its $H_{local}$ improves a lot as a result of caching popular content for the local clients in the local CDN server. We repeated some simulations for 10 CDN servers in a cluster and the trend of the results was the same.

From the simulation results we also found that in hashing-based schemes the load was more evenly distributed among different CDN servers and their blocking rate was lower than that of other schemes as shown in Figure 5.

We conducted additional simulations to study the effect of the adjustable parameter $P_{local}$ (the portion of a CDN server's disk space allocated for local popular content) on the semi-hashing based scheme. The results are shown in Figure 6. Under different request patterns (Figure 6.(a), fixed average request rate (Poisson); Figure 6.(b), variable average request rate in different periods of a day (non-stationary Poisson)), we adjusted $P_{local}$ from 0 to 1. When $P_{local} = 0$, it is in fact the pure-hashing based scheme which has the highest $H_{cluster}$ but the lowest $H_{local}$. More disk space for hashing cooperation (smaller $P_{local}$), higher content sharing efficiency and thus higher $H_{cluster}$. More disk space for local popular content (larger $P_{local}$), more requests are satisfied locally and thus higher $H_{local}$. When $P_{local} = 1$, each CDN server reserves all its disk space for local clients and there is no cooperation among the CDN servers, so $H_{local}$ equals $H_{cluster}$. By manipulating $P_{local}$, we can make a tradeoff between $H_{local}$ and $H_{cluster}$.

A very important observation of Figure 6 is that $H_{cluster}$ almost linearly decreases with $P_{local}$, but $H_{local}$ does not

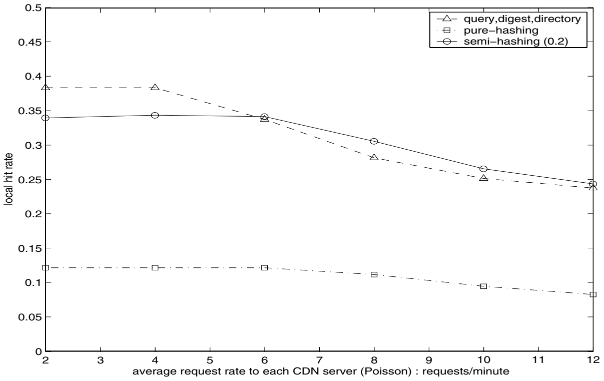

Fig. 3. $H_{local}$ of Different Intra-cluster Content Routing Schemes

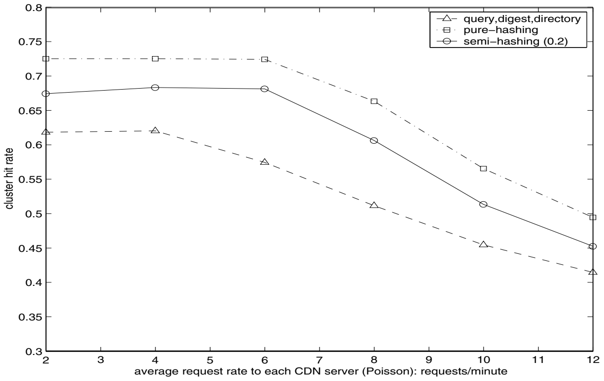

Fig. 4. $H_{cluster}$ of Different Intra-cluster Content Routing Schemes

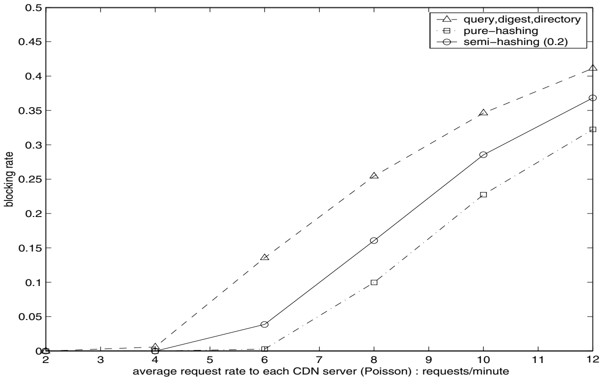

Fig. 5. Blocking Rate of Different Intra-cluster Content Routing Schemes

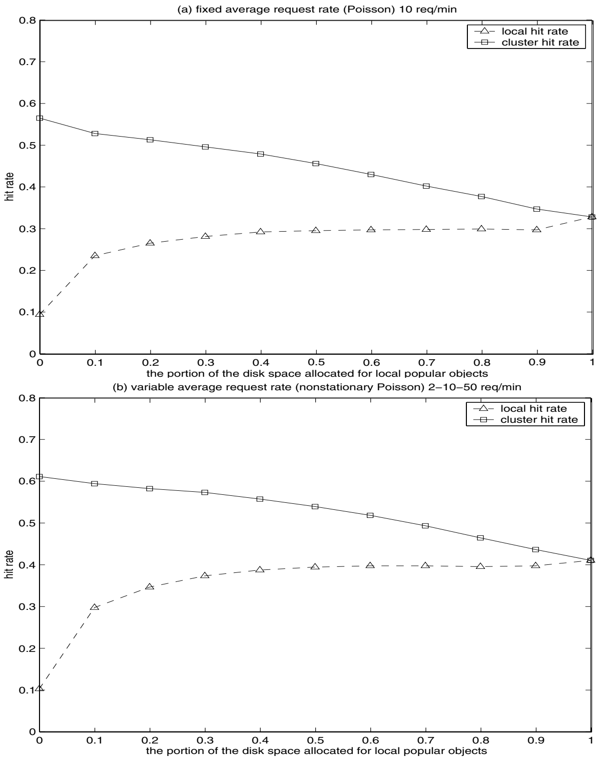

Fig. 6. Simulation Results of Semi-hashing Based Content Routing Scheme

linearly increases with $P_{local}$. Even a small portion (0.1-0.2) allocated for local popular content ensures a significant increase of $H_{local}$. This is because of the Zipfian popularity assumption of the objects: a small number of the most popular objects account for the majority of requests. While the Zifian popularity assumption may not be true for multimedia content, the authors in [16] found that the requests for videos on the WWW are even more biased towards popular titles than a Zipfian distribution: the top ten percent ranked titles account for 50% of all the requests. This implies that our semi-hashing based content routing scheme will perform even better in the real situation: a CDN server only needs to allocate a small portion of its disk space for local popular content to ensure both satisfactorily high $H_{local}$ and $H_{cluster}$.

## VI. Conclusions and Future Work

In this paper we proposed a hierarchical content routing architecture for on-demand multimedia content in large-scale CDN, in which CDN servers are grouped into clusters and perform intra-cluster and inter-cluster content routing. In particular, we proposed a semi-hashing based scheme for intra-cluster content routing and a content-query based scheme for inter-cluster content routing. By analyzing the performance of different intra-cluster content routing schemes in terms of their routing overhead and corresponding CDN performance through qualitative analysis and simulations, we concluded that the semi-hashing based scheme is scalable (small routing overhead that is independent of the number of cooperating CDN servers), efficient (high content sharing efficiency), and flexible (adjustable parameters) for intra-cluster content routing.

For live multimedia content, content routing is actually multicast content routing in which an application level multicast tree is built incrementally. In the future we will investigate multicast content routing schemes for live multimedia content based on the same hierarchical architecture.

CDN solves the scalability problem by providing large-scale and better-quality content delivery services in the current Internet. In fact, content routing consists of two parts: locating the requested content and selecting a network path to deliver the content. In this paper we concentrated on the first part and assumed that content is delivered using the default Internet path. However, such a system still does not guarantee the QoS and only follows the best-effort service model. We will investigate the integration of QoS guarantee into CDN in the future, e.g., designing QoS-based content routing schemes in the next-generation (DiffServ/MPLS based) Internet.

## Acknowledgment

This work was supported by the HKUST under Grant HIA01/02.EG06-Wireless Multimedia Networks. We also thank the anonymous reviewers for their helpful comments.

## References

[1] Akamai Technologies, URL: http://www.akamai.com.
[2] SinoCDN, URL: http://www.sinocdn.com.
[3] Y. H. Chu, S. Rao, and H. Zhang, ”A Case for End System Multicast,” Proc. ACM Sigmetrics, June 2000.
[4] D. G. Andersen, H. Balakrishnan, M. F. Kaashoek, and R. Morris, ”Resilient Overlay Networks,” Proc. 18th ACM SOSP, Oct. 2001.
[5] S. Ratnasamy, M. Handley, R. Karp, and S. Shenker, ”Topologically-Aware Overlay Construction and Server Selection,” Proc. IEEE INFOCOM 2002, June 2002.
[6] A. Shaikh, R. Tewari, and M. Agrawal, ”On the Effectiveness of DNS-based Server Selection,” Proc. IEEE INFOCOM 2001, April 2001.
[7] M. Andrews, B. Shepherd, A. Srinivasan, P. Winkler and F. Zane, ”Clustering and Server Selection using Passive Monitoring,” Proc. IEEE INFOCOM 2002, June 2002.
[8] D. Pendarakis, S. Shi, D. Verma and M. Waldvogel, ”ALMI: An Application Level Multicast Infrastructure”, Proc. 3rd Usenix Symposium on Internet Technologies and Systems, March 2001.
[9] D. Wessels and K. Claffy, ”Internet Cache Protocol (ICP), version 2,” Internet Engineering Task Force, RFC 2186, Sep. 1997.
[10] A. Rousskov and D. Wessels, ”Cache Digests,” Computer Networks and ISDN Systems, vol. 30, no. 22-23, pp. 2155-2168, Nov. 1998.
[11] S. Gadde, M. Rabinovich and J. Chase, ”Reduce, Reuse, Recycle: An Approach to Building Large Internet Caches,” Proc. Workshop on Hot Topics in Operating Systems, pp. 93-98, April 1997.
[12] V. Valloppillil and K. W. Ross, ”Cache Array Routing Protocol v1.0,” Internet Draft, Feb. 1998.
[13] D. Karger, T. Leighton, D. Lewin, and A. Sherman, ”Web Caching with Consistent Hashing,” Proc. 8th Int. WWW Conf., May 1999.
[14] E. O’Neil, P. O’Neil, and G. Weikum, ”The LRU-K Page Replacement Algorithm for Database Disk Buffering,” Proc. ACM SIGMOD’93, pp. 297-306, Washington, DC, May 1993.
[15] L. Breslau, P. Cao, L. Fan, G. Phillips and S. Shenker, ”On the Implication of Zipf’s Law for Web Caching,” Proc. IEEE INFOCOM 1999, March 1999.
[16] S. Acharya, B. Smith and P. Parnes, ”Characterizing User Access To Videos On the World Wide Web,” Proc. SPIE/ACM MMCN 2000, San Jose, CA, Jan. 2000.