A Path Based Internet Cache Design for GRID Application

Hyuk Soo Jang[1], Kyong Hoon Min[3], Wou Seok Jou[2], Yeonseung Ryu[1], Chung Ki Lee[1], and Seok Won Hong[1]

[1] Department of Computer Software, MyongJi University
San 38-2, Yong In, KyungGi, Korea
[2] Department of Computer Engineering, MyongJi University
[3] Samsung Electronics Co. Ltd, Core Lab.(WCDMA), Suwon, Korea

Abstract. Internet users are most likely opening multiple windows and surfing several sites concurrently with frequent site changes within a relatively short period of time. This work proposes a web cache organization algorithm, which can satisfy the frequent site changes effectively with low cost. The algorithm is based on the collected statistics of the visited sites and the pattern analysis of the site change. Our study suggests that the proposed path based cache scheme outperforms the existing algorithms in the hit ratio and response time dramatically.

1 Introduction

Most of the current web cache mechanisms are based on the hierarchical structure like the CERN and Harvest/Squid [1]. Each host needs to specify a cache server in advance, while the cache servers are in general hardwired each other hierarchically [2,3,4]. The cache needs to be organized to adapt the dynamic usage patterns of the current internet users, who are likely to open multiple windows and visit several sites concurrently with frequent site changes within a short period of time. Therefore, we need to build a new cache organization algorithm to adapt the dynamic site changes of the users.

We collect and analyze requested URLs, visited site sequences and routed paths in the client-server GRID environments. The analysis result shows that most of the requests traverse the same routes to a certain branching point. Only a handful number of different paths exist thereafter. We categorize them into a few groups based on the traversing path. Then, each cache or partition of a cache is redesigned to take care of each group.

2 A New Web Cache Organization Algorithm

This paper finds out that many URL requests follow a common route until they reach a certain point. Neighboring organizations running similar application like GRID are likely to use the same URLs, ISPs and routes. While the number of ISP are relatively small and the requested URL routes are in most cases directed to a

M. Li et al. (Eds.): GCC 2003, LNCS 3033, pp. 455–458, 2004.
© Springer-Verlag Berlin Heidelberg 2004

few ISPs, the routing paths are not so plenty that it is rather easy to categorize them into a few groups based on the common path. Our study shows that many internet users request similar URLs regardless of their locations, when the age, interest and education level of the users are homogeneous in similar application.

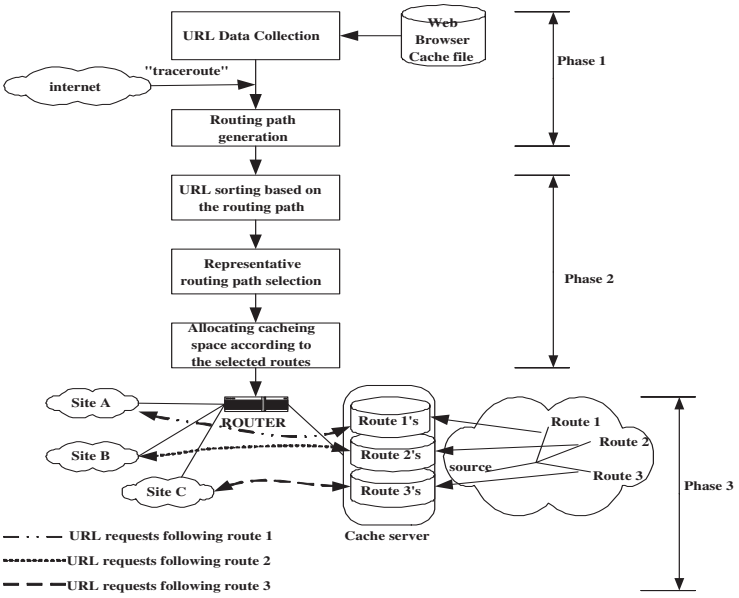
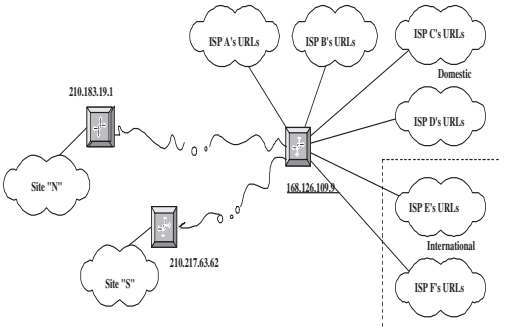


Fig. 1. A proposed web cache algorithm

The suggested algorithm is depicted in the figure 1. In the first phase, data are collected to find out the requested URLs in an organization. In the second phase, the requested URLs are sorted based on the routing paths. The URLs sharing common paths are categorized into a group and those remaining URLs not even belonging to a certain group can be put into another group. Then all the URLs are classified into several groups. We need to count the number of requests per group to decide which group is heavily referred. A cache (or a portion of a cache) is favorably allocated to the heavily referred group. In the third phase, the actual test operation is done according to the users' URL requests. The request is directed to the corresponding cache according to the URLs.

3 Performance Analysis of the Algorithm

We test the algorithm on two closely located sites, called "N" and "S", running similar business application. Three computers of the "N" and the "S" are selected randomly and the history data of the web browsers are analyzed so that each "N" or "S" uses almost identical path until 168.126.109.9 router as shown in

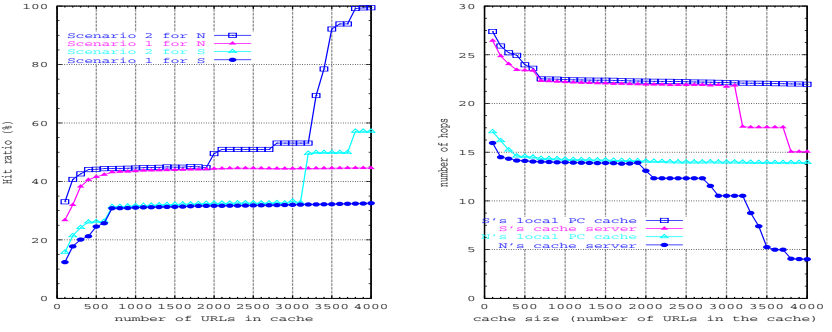


(a) Routing paths originating from "N" and "S"

Preferred Ranks	Request Rates
1	30%
1+2	47%
1+2+3	58%
1+2+3+4	68%

(b) Request rates against the preferred routing paths

Fig. 2. Routing paths and statistics



(a) Hit ratio (b) Response time

Fig. 3. Experiment results based on scenario 1 and 2

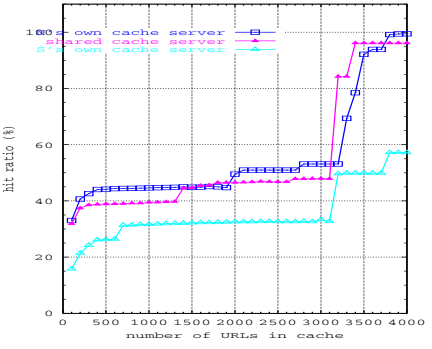


Fig. 4. Hit ratio based on the scenario 2 and 3

the figure 2(a) and then several routes exist thereafter. Based on the 16,000 and 14,000 collected data from the "N" and the "S" respectively, we found that there are a few routing paths which most of the requests follow as shown in the figure 2(b). Four most visited routing paths can cover 68% of the whole requests and the most preferred path takes care of 30% of the entire requests. The performance analysis is done based on the following three scenarios: (I) in case of no cache, (II) when each "N" or "S" has its own cache, (III) when a single cache is shared by both "N" and "S".

In the scenario I, the "N" or "S" does not have a cache server in an institutional level, but uses its local cache in the PC as a web browser's cache. A cache server is used in the scenario II and III. The size of the cache is measured based on the number of URLs to be stored. The hit ratio is increased in the scenario II compared with the scenario I as shown in the figure 3(a). The performance of the "N" looks better than the "S", but the results are not conclusive. The response time is proportionally reduced as the hit ratio is increased as depicted in the figure 3(b). The performance comparison is done for the scenario 2 and 3 in the figure 4 and the hit ratio is not decreased even in the case of sharing a single cache server. It suggests that we can save the total cost if the sites share a cache server.

4 Results and Future Work

This paper shows an algorithm to design a cache system based on the routing paths of the requested URLs. The performance analysis shows that the new algorithm performs well in response time and hit ratio even in the fast site changes of the users. Also, the hit ratio is not degraded when a cache server is shared by multiple sites.

References

1. K. Claffy D. Wessels. Icp and the squid web cache. *IEEE Journal*, Apr. 1998.
2. J. Almeida L. Fan, P. Cao. Summary cache: A scalable wide-area web cache sharing protocol. *ACM*, 1998.
3. K.W. Ross. Hash routing for collection of shared web cache. *IEEE Network*, Nov. 1997.
4. E. W. Zegura S. Bhattacharjee, K. L. Calvert. Self-organizing wide-area network caches. *IEEE INFOCOM*, 1998.