IntraCache: An Interest group-based P2P Web Caching System

Huifang Cheng, Zhimin Gu, Junchang Ma
Department of Computer Science and Engineering, Beijing Institute of Technology, Beijing 100081
chenghuifang@126.com, zmgu@x263.net

Abstract

An interest group-based P2P browser cache collaborative system, named IntraCache, is proposed in the paper. IntraCache is scalable, resilient to node failures and easy to manage nodes. In the system, the peers with similar interest are organized into autonomous group by PB grouping method and documents are located by similarity based search method. Trace-driven experiments show that PB-Grouping method can utilize local browser cache more efficiently than previous grouping methods. Even if using small cache size, PB grouping method can get preferable hit ratio. Moreover, the interest group-based search method can more efficiently prune the P2P search space and reduce the latency than previous search methods.

1. Introduction

Peer-to-Peer network is recently attracting a great deal of attention [1, 2, 3, 4]. It has been increasingly deployed for many fields, such as file sharing, P2P computing, instant message, search engine and so on. In this paper, we focus on another application: browser cache sharing, which is thought as one of the promising applications that could be profited from P2P substrates [5, 6, 7]. Unlike existing web caching techniques that typically are controlled by the proxies, P2P web caching techniques look at how to utilize browser cache of each client node.

Proxy caching technique, such as Netcache [8] and Squid [19], is an effective solution to quickly access and reuse the cached data and to reduce internet traffic to web servers. This method is easy to manage and locate nodes, but not scalable and resilient to single node failure due to their centralized control. Moreover, if the requests miss in the proxy, proxy will forward it to upper level proxy or server. This technique neglects to consider whether the missed contents exist in other clients' browser cache. Since there is no cache sharing among different peers, even if the requested document is available in a node in local LAN, the request may still be forwarded to web server, which will incur both long response time and high cost.

Current P2P web caching schemes can solve the problems of proxy partly; however, they have their drawbacks respectively. As far as we know, the building block of substrate of existed P2P web caching systems is the notion of peer-group, in which a number of nodes are organized by DHT algorithms(Squirrel[5]) or physical locations(Browser-Aware proxy server[6, 7]). DHT organization is not appropriate for the P2P web caching application, since it assigns file storage based on a random hash function, not based on what files users already possess, however, in P2P web caching network, users share cache that they have, and it would be impractical to imagine users storing files other than their own. In the other organization based on physical locations, to find a specific object, peer must search the whole P2P space, which is time-consuming. What's more, the above methods can not make use of semantic information among the sharing contents of peers.

To address the above limitations, we propose an interest group-based P2P web caching system, named IntraCache, which looks at how to exploit browser caches of nodes in intranet and use interest group model to organize peers in the system. Peer interest group is a set of active peers, who have the same interests and are involved in sharing browser web cache. For example, if node A has visited the web site about sports for many times, and a node B also has accessed the same web site time after time, then both of them have the same interest on sport and can become an interest group.

In the rest of paper, we will first give an overview of the system topology of IntraCache in section 2. Section 3 gives interest-group based reconfiguration and interest group-based search method. Section 4 describes simulation experiments and performance evaluations. We conclude the paper in section 5.

2. Overlay Network of IntraCache

Our proposed overlay network of IntraCache is shown in Fig. 1, which consists of three kinds of peers: fat peer, thin peer and P2P registrar. P2P registrar, who has a fixed IP address and runs software named Registrar, provides three functions. First, it assigns a unique peerID for each peer in the system. Second, it maintains the peer's current status information including the IP address and whether the peer is online or offline. Third, it records the relationship between fat peer and thin peer and the interest group information including the scale and current interest vector. Fat peer may be a past proxy server or produced dynamically by thin peers in one group. The Peer Manager and Index Builder thread running in the fat peer manage the peer status and index information in one group respectively. In addition, some hot browser cache contents also store in fat peer. This storage approach is helpful to accelerate the search speed and reduce the network communication traffic. Thin peer is a common node, which can communicate and share resources directly with any node in the P2P network. There are three threads

1-4244-0910-1/07/$20.00 ©2007 IEEE

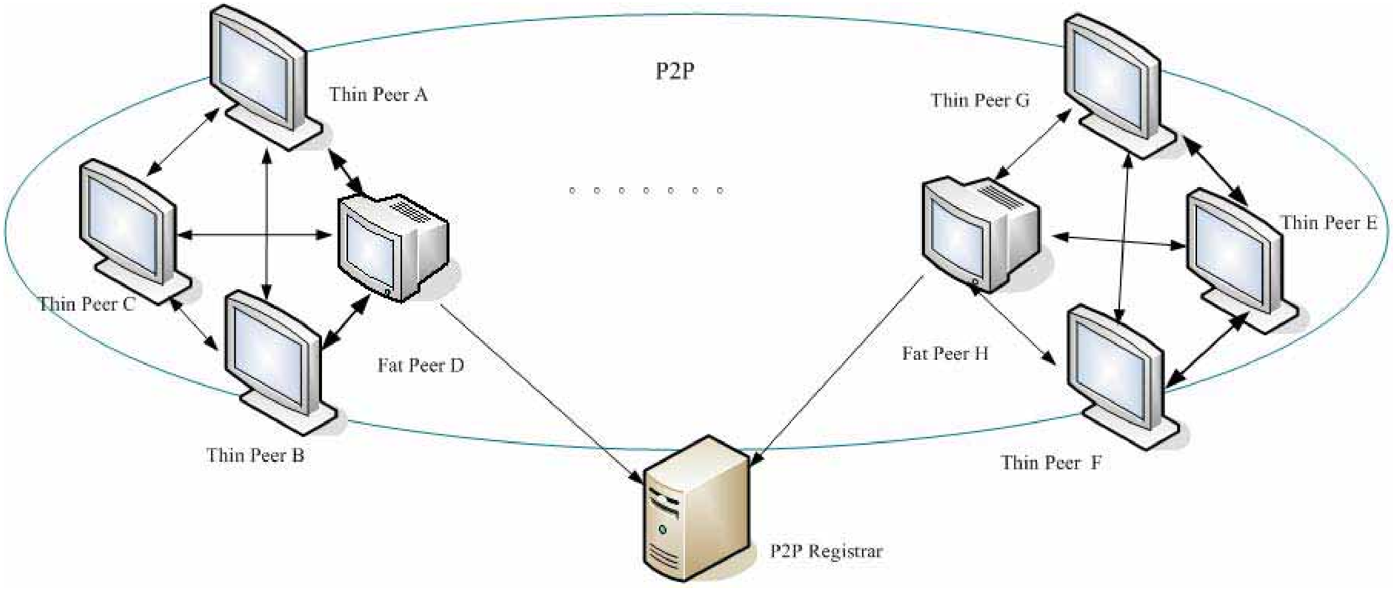

Figure 1. The overlay network of IntraCache

running in the thin peer. The first is Local Proxy, which is used to intercept the requests sent by users; the second is Index Collector, which is used to report index information of node; the last is Cache Monitor, which is used to monitor the update of browser cache of peer so as to keep consistency with the indices of fat node. User can use IE or FireFox explorers to send url request directly without revising the browser codes.

When web user A submits a url request, if the request is not present in the local cache, Local Proxy will forward it to the fat node. If the requesting document is present in hot web cache in fat node, fat node will send the content to peer A directly. Otherwise, fat peer will search the indices locally and pass the information about the document location B back to peer A. If B is not NULL, Peer A will access the request document from B. Otherwise, the request will be forwarded to other fat nodes to search the matching document. The index information in fat node is arranged by order, we can use binary-search method to decrease the search time (search time goes logarithmically with index number). Each index item consists of two parts: one is url, the other is peer information including peerID, IP address and so on.

This structure will not introduce single failure, if one thin node goes down, the requesting document can be accessed from other thin nodes; if one fat node fails, thin peer in the group can elect a new fat node; If the P2P registrar goes down, new thin peers can not be joined in the system, however, it will not influence the performance of the peers that have been in the system.

3. The Feature and Algorithms of IntraCache

This section introduces the features and algorithms of IntraCache in detail, which includes interest group-based reconfiguration and self-adaptive search method. Moreover, this section also presents how to extract the interest of peers and how to compute the interest similarity among peers.

3.1 Interest group-based Reconfiguration

Reference [9, 10] view P2P networks have many similarities with real social network; structuring P2P overlay network can use similar solutions in social network as reference. For example, in social system, people incline to form group and to share information efficiently among members in group. They usually elect one people as leader. These observations lead us to believe that a P2P web caching system made up of peers can also form similar group structures.

IntraCache uses interest group model to organize the peers with similar interest into autonomous group. The model divides interest into two classes--peer interest and group interest. Peer interest can be extracted from the urls of its accessed pages. Group interest is common interest of two or more nodes, which are dug by the urls of hot access web objects in interest group. The reason of this interest extraction method is shown in next section.

The interest group-based reconfiguration policy works as follows. Fat node and P2P login registrar are used to facilitate the dynamic reconfiguration. When a peer enters into the system, it first gets the urls of latest two days' accessed pages from the browser to construct the interest vector of node. At the same time, the peer will communicate with P2P registrar to send its current IP address and request all the interest vectors of current interest groups. The similarities with all interest groups will be computed by the peer locally and those with the higher similarity will be ranked higher. The similarity computing method is shown in next section in detail. Peer will register itself into the most beneficial interest group that has the highest similarity with current peer. At the same time, the K interest groups with highest value of similarity will be kept as the search candidates, where k is a system parameter that can be set by participating peers. Otherwise, if similarities between the new peer and current interest groups are all smaller than a threshold, new peer can create an interest group and become the manage node(fat node) of the group. Moreover, it needs to register the interest group information into P2P registrar.

The interest of user is constantly mutative, so P2P registrar needs to communicate with all fat nodes and get the interest vectors of interest group every definite period. These interest vectors are “light-weight” files, which will not consume much bandwidth. Fat nodes and P2P registrar will not become bottlenecks, since they just need to transfer the information of interest group. When the number of objects that have changed in thin peer reaches over 30 percents of the number of original objects, thin peer will re-compute similarities with current interest groups and change its registered interest group and the search candidate groups dynamically. The similarity computing process can run in background, it will not influence the other operations of peer. After a while, peers with common interests will be aggregated into clusters. Each interest group will be managed by a fat node.

Different from previous methods which aggregate users based on the trace usually, the organization method can reflect current user interest dynamically and can apply to P2P web caching system in which the interest of peer is changing. IntraCache distinguishes itself with the feature.

3.2 Interest Representation and Similarity Measurement

In this section, we define and explain in detail how to represent the interests and define the similarity measurement between peers. Peer’s interest can be obtained by its browsed pages. Recent researches [14, 15] show the navigation path of web users carries valuable information about user interests. If a web site is well designed, there will be strong correlation between the similarity of the url paths and similarity among the peer interests. We can get the clustering result of peer interest by clustering the url paths. Current aggravating algorithms are based on trace usually. From the user’s browsing logs, the following information could be gathered: the urls, the frequency of a url, the browsed time of a url and the order of pages accessed by individual web user. Based on the above information, there are several similarity measurements: UB, FB, VTB and VOB. All similarity measurement methods usually suppose the browsed pages are not related each other and nothing is considered but the weight of same pages. UB (Usage-based) method uses the number of common pages they accessed to measure the similarity. FB (Frequency-based) method uses the number of times they accessed the common pages at all sites to measure the similarity. In the VTB (Viewing Time-Based) method, the similarity between two users is measured more precisely by taking into account the actual time the users spent on viewing each web page. VOB (Viewing Order-Based) method considers two users have the same interest only when they access a sequence of web pages in the exact same order, the similarity between users is measured by checking the access orders of web pages in their url paths. The detailed formula can be seen in paper [14]. From the analysis, it can be seen that VOB and VTB are not applicable for the online clustering algorithms, since they need high cost to record the browsing time and access order of all urls in P2P web caching system. What’s more, both measurement methods just take into account the weight of same pages while neglecting the weight of similar pages. Actually the information hidden in similar pages is also very important to measure the similarity between peers.

To address the limitations, we propose a page-based (PB) approach. To precisely specify PB method, we first introduce some definitions:

Definition 1. Let P_1 and P_2 be two url paths, they are written by $s_1/u_1/u_2\ldots/u_n$ and $s_2/v_1/v_2/v_m$ respectively. Each of the s_1, s_2, u_i and v_i is called a path feature or an interest point. The length of a url is defined as the number of path features in the url, thus $|P_1|$ =n+1. We define path similarity between two urls P_1 and P_2 as the length of the longest common prefix of P_1 and P_2, which is denoted by PathSim (P_1, P_2).

For example, the urls http://www.nasa.gov/missions/research/fuller.html and http://www.nasa.gov/missions/index.html have a path similarity of 2.

Definition 2. Suppose there are m users U = {u_1, u_2 … u_m}, let P = {p_1, p_2 … p_N} be the distinct web page path sets of all peers. Each user’s interest can be denoted by a group of path features of all urls, which constitute a vector space. We use the most commonly used measure, Cosine function, to compute the similarity in the vector space model. Let $|p_i|$ be the length of the url of p_i, function a(u_1, p_i) represents that web page p_i is accessed by user u_1, the peer similarity between u_1 and u_2 is described as

$$\mathrm{PeerSim}(u_1,u_2)=\frac{\sum_{i=1}^{N}\sum_{j=1}^{N}a(u_1,p_i)*a(u_2,p_j)*\mathrm{PathSim}(p_i,p_j)}{\sqrt{\sum_{i=1}^{N}(a(u_1,p_i)*|\,p_i\,|)^2*\sum_{j=1}^{N}(a(u_2,p_j)*|\,p_j\,|)^2}}$$

This definition not only considers the weight of same pages between peers, but also takes into account the relevancy of similar pages, which defines the peer similarity based on more accurate level than other methods. The evaluation and comparison between PB and other methods are shown in next section.

3.3 Interest group-based Search

Searching for document is one of the key challenges in P2P web caching system. At present, search techniques in P2P system include flooding, directed flooding, iterative deepening, directed BFS and so on[16]. In the above methods, to locate content, the node sends query to any number of its neighbors; the neighbors will continue forwarding the query until the forwardness times exceed a threshold, which need great search time cost and result in high bandwidth burden.

We propose a self-adaptive interest group-based search method to locate documents in IntraCahe system. In IntraCache, each thin peer registers itself into the fat node, which has the highest value of similarity with current thin peer. Each thin peer also maintains K interest groups with higher value of similarity. In fact, the similarity value means the similarity of cache contents between thin peer and interest group. The requesting document of one thin peer have high probability in its own interest group, so the request of one peer will be first forwarded to its registered fat node. If the matching document is found, the information of targeted peer will be returned. Otherwise, the request will be forwarded to the highest interest group in the search candidate groups. If the accumulated search hops exceed a threshold, request will not be forwarded again. The value of threshold can reference the experiment value in next section. To more visually demonstrate the interest group-based search strategy, we show the algorithm by the following pseudo codes.

```
BOOL Search (MD5 url, int K, List peersets)
//url is the md5 value of the url of search document, K is the
number of search candidate groups, peersets is the targeted
peer list.
{
   int iTotalHops = 0;
   int count = 0;
   while(iTotalHops < MAX_HOPS &&count <K)
   //MAX_HOPS is the most forward hops that set in system
   {
   SendQueryToFatNode (InterestGroup[i], url);
   //send query to K interest group one by one
   GetQueryResults(PeerSets);
   //get the return query result
   if(peersets != NULL)
        return TURE;
   }
   // Once find the matching document, return.
   If(peersets==NULL)
       return FALSE;
}
```

Using this interest based strategy, every location request can self-adjust its search scope by the number of search candidate groups and maximal search hops. Once the matching document is found, the request will not be forwarded. This algorithm can assure that it can find the target node with least time and bandwidth.

4. Simulation Experiments

4.1 Traces

Table 1 lists the web traces we have used in the trace-driven experiments. The first four traces are from NLANR. They can be downloaded from [17] with authorized name and password. The four NLANR traces are denoted by uc, ny, bol1 and bol2 respectively. Client IP addresses are randomized from day to day in the trace, however, client IP addresses are very important in our study, so we used traces based on one day's log file. The last is collected on a medium education institution by ourselves, which is a proxy log and denoted by med-edu.

Table 1. Trace statistics

Traces	Requests	Bytes	Clients	Date
uc	453871	6.32GB	195	10/22/05
ny	398213	4.625GB	121	10/22/05
bol1	121867	2.48GB	137	10/23/05
bol2	34372	1.08GB	75	10/22/05
med-edu	4273330	61.2GB	417	01/01/05-01/07/05

4.2 Simulation Environment

In this section, we introduce the construction techniques to generate a P2P web caching network overlay network that can be used to simulate our algorithms. It is hard to build the overlay network using real peer wholly due to resource limits, so we need to produce a great deal of virtual peers. The virtual peer is actually a thread that generates user requests based on the real log trace. Fat peer, virtual thin peer and login server are connected by 100Mbps network. A least-recent used (LRU) replacement algorithm is used in our simulation.

We use three performance metrics: hit rate, search hop, latency and node cost. The experiments in section 4.3 first compare the clustering capability of PB, UB and FB methods. Based on the results of clustering results, we compare the hit rate improvement between PB grouping and random grouping method. In next section we measure the search hops of PB, UB and FB grouping methods and the retrieve latency of web document in IntraCache and proxy system. Section 4.5 gives the experiments results of additional cost on each node.

4.3 Hit Rate

We carried out two series of simulations. The first series of simulations are to evaluate and compare the clustering capability of PB, UB and FB grouping method. Based on the results of clustering tests, the second series of simulations demonstrate the hit rate improvement of similarity-based grouping (PB, UB and FB) over random grouping method. In the first series of experiment, each unique IP address in traces denotes a thin peer. At the start, all thin peers' cache is null, they fill in their cache with a part of urls in the real log according to their IP addresses. After this "warm-time" period, each node can use the access history information to compute the similarity with all interest groups in the system, the result will decide whether the node join the existing group or create a new group. The clustering algorithm uses half of each client's urls as the training set, the remaining urls are test set, which are used to test hit rate and other performances of clustering algorithm.

Figure 2 gives the test result of different clustering algorithms (PB, UB, FB) on ny, uc, bol1, bol2 and med-edu traces. If one interest group only has a node, the node is called isolated point. In the figure, x-axis is different traces, y-axis is the percent of the nodes that are non-isolated point and can be aggregated into interest groups. From this figure, it can be seen that compared with UB and FB, PB can cluster more nodes into interest groups to share browser cache. Especially for med-edu trace, UB method makes 5 percents of the nodes join in the interest group, FB method makes 50 percents of the nodes join in the interest group, our PB method makes 95 percents of the nodes join in the interest group. The reason is that UB and FB methods just consider the weight of same pages, but PB method not only considers same pages' weight, but also takes into account the similar pages' weight.

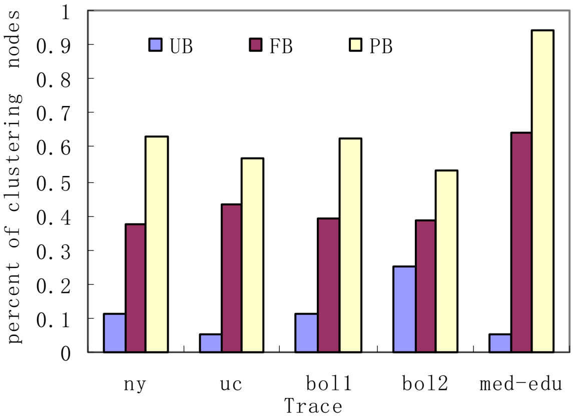

Figure2. The comparisons of PB, UB and FB method

Next, we will show an example to explain the computing results of PB, UB and FB methods. Table 2 lists the url space of two users.

table2. Url space of two user

IP Address	url	times
192.168.1.125	http://nasa.gov/missions/research/fuller.html	2
	http://nasa.gov/missions/index.html	3
192.168.2.133	http://nasa.gov/	4
	http://nasa.gov/missions/research/index.html	2

Since the two users have not access the same pages, their similarity is 0 according to UB and FB methods, however for PB method the similarity is larger than 0. Though the pages are not same, they come from the same directory. The directory structures of a web site indicate the classification for contents. The pages under the same catalog usually have strong similarity, if user has interest in some pages in one catalog, he maybe have interest in other pages in the same catalog.

Based on these observations, we think if the hidden information between pages can be taken into account, more users can be joined in the interest group, and then each node can advance the hit rate at the most. The idea is convinced by the following tests.

In the second series of tests, we operated hit rate experiments with different interest groups formed by PB, UB and FB methods respectively on uc, ny, bol1 and med-edu traces. Performance results of all the interest groups are quite consistent. Due to page limits, we show the typical change trends of hit rate of UB grouping, FB grouping, PB grouping and random grouping methods in Fig. 3, where the size of browse cache is scaled from 0.1M, 1M, 10M, 100M to 1000M. As shown in the figure, the hit rate in interest group formed by similarity-based methods (PB, UB and FB) is higher in that in random group(R-grouping). Moreover, PB grouping method can achieve higher hit rate than UB and FB grouping method, even with small cache size, the collaboration of large number of nodes can get preferable hit ratio. The results show that PB grouping method can more precisely describe the similarity between users than other methods.

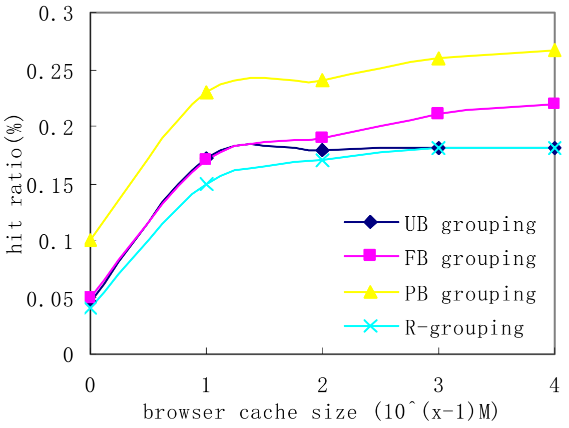

Figure 3(a). uc trace

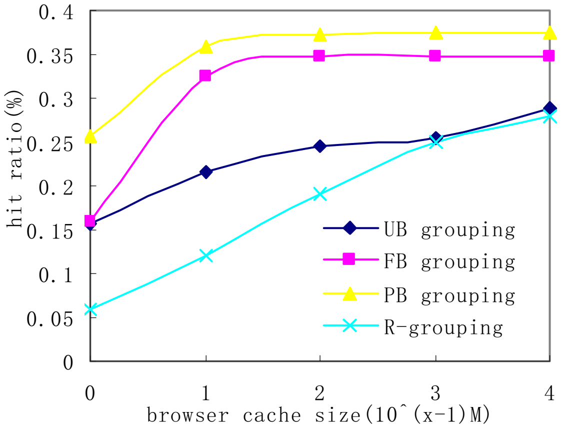

Figure 3(b). ny trace

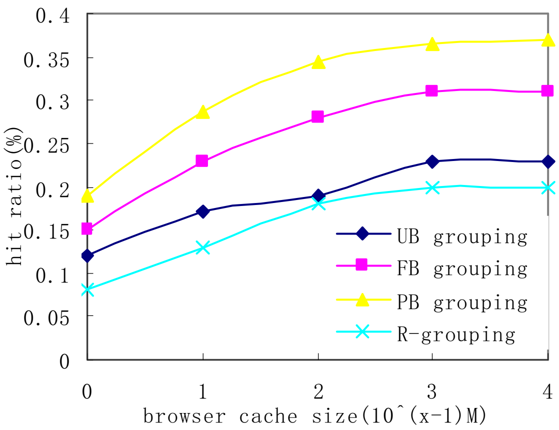

Figure 3(c). bol1 trace

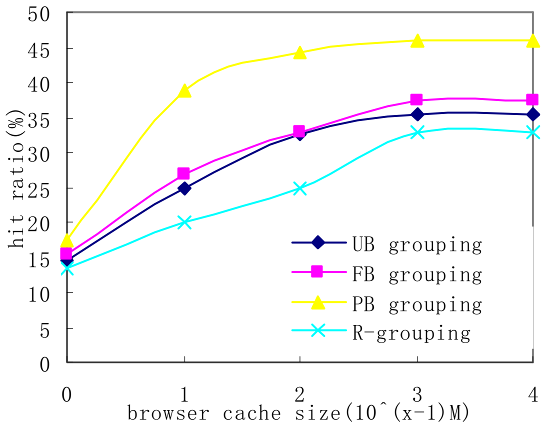

Figure 3(d). Med-edu trace

In P2P web caching system, peer's interest may change dynamically, so the clustering algorithm will be operated periodically. If clustering method needs great time costs to compute the similarity among users, this method will become senseless even if it can achieve high hit rate. Next we will show the time cost of PB, UB and FB method. The interest of group is represented by the urls of hot web cache in the interest group. The interest of peer is denoted by the urls of local browser cache. The time cost of clustering method presented in the paper depends on the number of urls in peer and interest group. Suppose the browser cache size is 100MB, the average size of document is 20KB. The relationship between clustering algorithms and the number of urls in interest group can be seen in Fig. 4, where the number of urls is scaled from 100,1000,10000,20000 to 30000.

From this figure, it can be seen that the cost of PB method is higher than UB and FB method. UB method needs the least cost. When the number of url is 100, 1000 and 10000, 20000 and 30000 the cost of PB method is 20ms, 221ms, 1063ms, 3560ms and 5350ms respectively. When peer logs in system, the time cost is acceptable. When peer's interest changes, the computing process will be run in background, it will not influence other operations of peers.

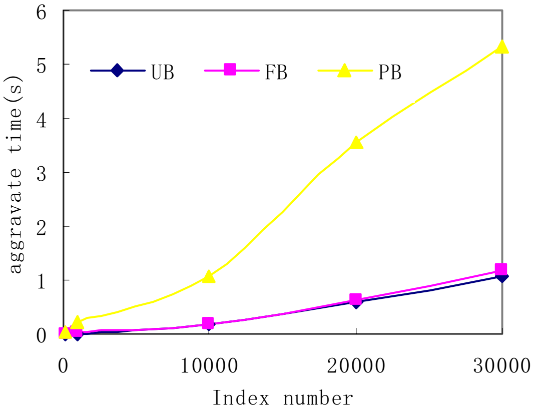

Figure 4. The cost of PB, UB and FB method

4.4 Latency

This section is mainly to measure two kinds of latency. One is search latency, which uses hop as metric; the other is the latency of accessing web document. From the above analysis, it can be seen that PB method has better clustering capability than other methods, so this section just lists the comparison between PB grouping and random grouping method. The search hop of PB grouping method on ny, uc, bol1, bol2 and med-edu traces can be seen in Fig. 5, where the search hops mean the number or interest group which must be searched to find a specific web document.

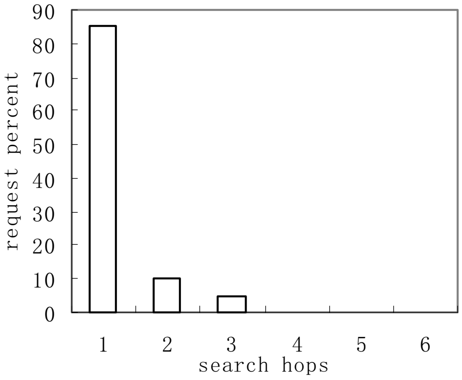

Figure 5(a). ny trace

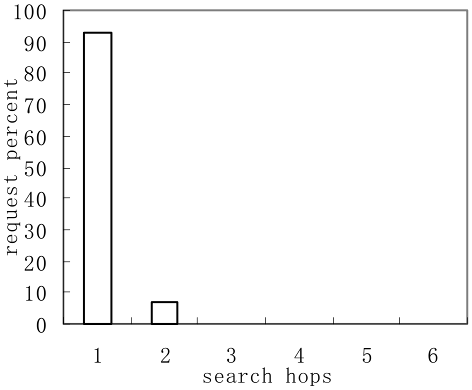

Figure 5(b). bol1 trace

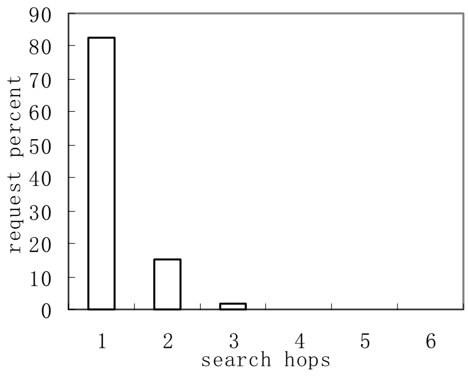

Figure 5(c). bol2 trace

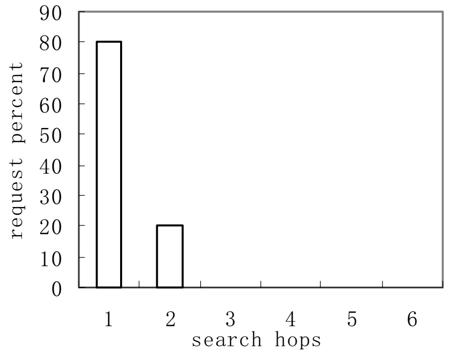

Figure 5(d). med-edu trace

From these results, it can be seen that most requests in the traces can be found in 1 or 2 hops, in other word, if one request can not be searched in two interest groups with highest similarity value, then the request can be forwarded to the origin server directly without searching in the IntraCache system. However, for random grouping method, peer must search all the interest groups to find a specific web document. For traditional flooding search method, TTL is set as 7 hops. For typical P2P web caching system Squirrel, search hops is related to the number of node in the system, it is about $O(\log_{16}N)$. When the node number exceeds 256, search hops will exceed 2. All these observations verify that PB grouping method can prune search space and correspondingly reduce search latency more efficiently than other search methods.

Figure 7 lists the retrieve latency in IntraCache, R-IntraCache (Random-Grouping), traditional proxy and server, where the data size is scaled from 10KB, 30KB, 70KB, 100KB, 200KB to 300KB. We construct two prototype systems to test the average latency of IntraCache and proxy respectively. If a request hit in the system, targeted peer transfers the same size file with the request document to the requesting peer; otherwise the request will be forwarded to the origin server. To eliminate the influence of the urls whose web servers are down or web documents are not present now, we adopt the following methods to compute latency of web server. We wrote a program that used Httperf[18] to retrieve the home pages of the top 50 popular web sites in the world from our institute laboratory, the program ran a test every hour and ran 48 hours in total. The average Net I/O is shown in Fig. 6. The two days show similar change trend. We use the fastest Net I/O (6 am, 188KB/s) to compute the latency of web server.

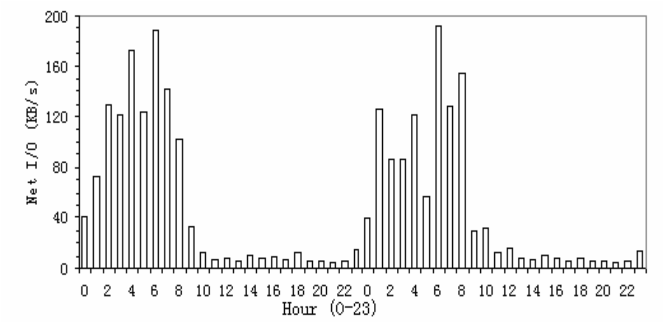

Figure 6. The latency of web server

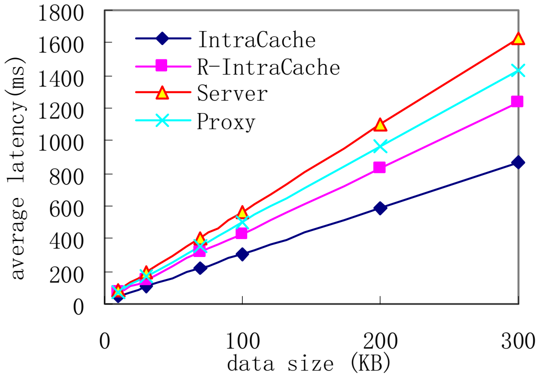

Figure 7(a). ny trace

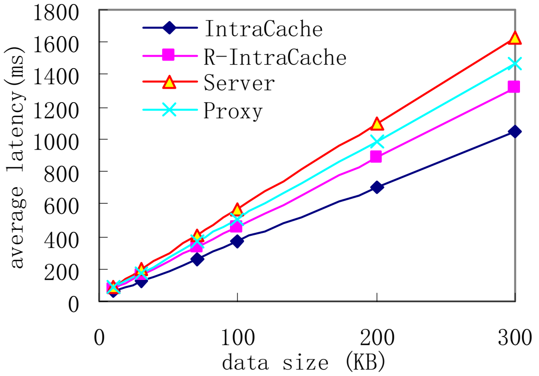

Figure 7(b). bol1 trace

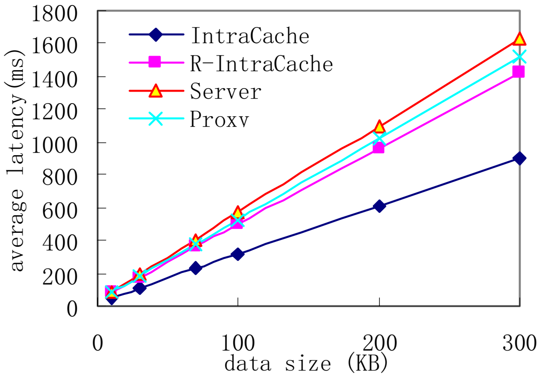

Figure 7(c). bol2 trace

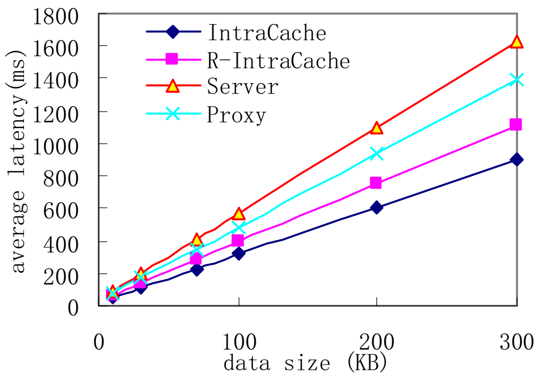

Figure 7(d). med-edu trace

4.5 Cost

The goals of the evaluation are try to answer the question about how much additional load do peers incur by participating in IntraCache application. At the start base memory cost is low: less than 2MB for thin peer and less than 3MB for fat peer. The base memory is used to run the software of fat node and thin node. Moreover, fat node needs extra space to store the indices. Each url is represented by a 16byte MD5 signature. Suppose each client has a browser cache with size of 100MB, the average cache size is 20KB; each browser has about 5K pages. The fat node just needs 80KB to store one thin node's indices.

5. Related work

Reference [5] presented a pure P2P web cache system named squirrel which was based on the Pastry [2] routing algorithm. Squirrel is scalable and resilient to single node failure, however, its assignment of file storage is not based on what files users already possess, but a random hash function. In P2P web caching network, users just share cache that they have, it would be impractical to imagine users storing files other than their own. It was described that an index server-relied P2P web cache model in two papers [6, 7], since it relies on a single proxy server, it is not only that single point of failure will be brought but the maximal client amount has to be set because of the limit of sever capacity, the system can not be scalable.

BuddyWeb [11] is the closest work with our analysis, there are three differences existing. First, the substrate of BuddyWeb is pure P2P structure, which can not avoid the disadvantages of pure P2P structure, while IntraCache adopts hybrid P2P structure, it can address the limitations. Second, BuddyWeb use a part of metadata in web pages to represent peer interest, it neglects to consider web pages whose metadata is missing or non-informal. Paper [13] shows vast majority of metadata provided on the public web is ad hoc in its creation, unstructured by any formal metadata scheme, which can not reflect peer interest fully, while we use the navigation path to extract peer interest, it is more accurate. Third, there is no latency improvement analysis of BuddyWeb in the paper while we perform trace-driven experiments to study the performance of IntraCache quantitatively.

6. Conclusion and future work

In this paper, we propose an interest based scheme to organize the P2P web caching system. Our scheme is unique in the following ways: first, it got peer interest from the url path information of its history access pages, so the similarity measurement is more precise to cluster peers with similar interests than previous methods'. Second, it can improve the hit rate heavily than random grouping. Third, it can efficiently prune the P2P search space and reduce the latency. In future, we plan to fully implement our scheme, and further optimize the similarity measurement algorithms.

7. References

[1] S. Ratnasamy, P. Francis, M. Handley, R. Karp, and S. Schenker. A scalable content addressable network. Proc. of ACM SIGCOMM'01, 2001.

[2] A. Rowstron and P. Druschel. Pastry: Scalable, distributed object location and routing for large scale peer-to-peer systems. IFIP/ACM Middleware 2001

[3] I. Stoica, R. Morris, D. Karger, M. F. Kaashoek, and H. Balakrishnan. Chord: A scalable peer-to-peer lookup service for internet applications. ACM SIGCOMM'2001, 2001.

[4] B. Zhao, J. Kubiatowicz, and A. Joseph. Tapestry:an infrastructure for fault-tolerant wide-area location and routing. Tech. Rep. UCB/CSD-01-1141, Computer Science Division, UC Berkeley, Apr. 2001.

[5] S. Iyer, A. Rowstron, and P. Druschel. SQUIRREL: A decentralized, peer-to-peer web cache. Proceedings of the 12th ACM Symposium on Principles of Distributed Computing (PODC 2002), July 2002.

[6] L. Xiao, X. Zhang, and Z. Xu. On reliable and scalable peer-to-peer web document sharing. Proceedings of 2002 International Parallel and Distributed Processing Symposium, Apr. 2002.

[7] Yonggen Mao, Zhaoming Zhu, and Weisong Shi. Peer-to-Peer Web Caching: Hype or Reality. Proceedings of the 10th IEEE International Conferences on Parallel and Distributed Systems, July 7-9, 2004. California.

[8] P.Danzig, NetCache architecture and depolyment, Proceedings of the 3rd International WWW Caching Workshop, Manchester, England , June 1998

[9] M. Khambatti, K. Ryu and P. Dasgupta. Peer-to-Peer Communities: Formation and Discovery. In: 14th IASTED Int'l. Conf. Parallel and Distributed Computing Systems (PDCS), Cambridge, MA, USA, 2002.

[10] M. Khambatti, K. Ryu and P. Dasgupta. Structuring Peer-to-Peer Networks using Interest-based Communities. In: Proceeding of International Workshop on Databases, Information Systems and Peer-to-Peer Computing (P2PDBIS), Humboldt University, Berlin, Germany, September 2003.

[11] XiaoYu Wang, WeeSiong Ng, BengChin Ooi, Kian-Lee Tan, AoYing Zhou. BuddyWeb: a p2p-based collaborative web caching system. Proceedings of the International Workshop on Peer-to-Peer Computing, 2002, 72~77.

[12] Bo Ling, Xiao-Yu Wang, Ao-Ying Zhou and Ng Wee-Siong. A Collaborative Web Caching System Based on Peer-to-Peer Architecture. Chinese Journal of Computers, Vol.28, No.2 Feb.2005.

[13] E.T. O'Neill, B.F. Lavoie, P.D. McClain. Web Characterization Project: An Analysis of Metadata Usage on the Web. http://digitalarchive.oclc.org/da/View Object.jsp?objid=0000003486.

[14] Jitian Xiao, Yanchun Zhang, Xiaohua Jia, Tianzhu Li. Measuring similarity of interests for clustering Web-users.
Proceedings of the 12th Australian Database Conference 2001 (ADC'2001). Washington, DC, 2001: 107-114.

[15] C. Shahabi, A. M. Zarkesh, J. Adibi, and V.Shah. Knowledge Discovery from Users Web Page Navigation. IEEE RIDE'97, 1997.

[16] Beverly Yang and Hector Garcia-Molina. Efficient Search in Peer-to-peer Networks. In Int.l Conf. On Distr. Computing Sys., Vienna, Austria, 2002.

[17] National Lab of Applied Network Research: http://www.ircache.net/. Sanitized access logs:ftp://ftp.ircache.net.

[18] D. Mosberger, T. Jin, Httperf—a tool for measuring web server performance, ACM SIGMETRICS Performance Evaluation, Dec 1998

[19] Squid Web Cache, www.squid.org/index.html