Efficient Formation of Edge Cache Groups for Dynamic Content Delivery

Lakshmish Ramaswamy
Department of Computer Science
University of Georgia
Athens, GA-30602
laks@cs.uga.edu

Ling Liu
College of Computing
Georgia Tech
Atlanta, GA 30332
lingliu@cc.gatech.edu

Jianjun Zhang
College of Computing
Georgia Tech
Atlanta, GA 30332
zhangjj@cc.gatech.edu

Abstract

Cost-effective cooperation among a network of edge caches is widely accepted as an effective mechanism for enhancing the scalability, performance, and reliability of edge cache networks. However, the problem of how to form cache groups for achieving effective and efficient cooperation in edge cache networks has largely been unexplored. In this paper, we identify two important factors that need to be considered while forming cooperative groups, namely, network proximities of edge caches and network distances of the caches to the origin server. We propose two novel cache clustering schemes for accurately partitioning the caches of a given edge cache network into specified number of cache groups. The first scheme, called the Selective Landmarks scheme (SL scheme), accurately partitions the edge cache network into cooperative groups based on the network proximities of the caches. The second cache group formation scheme, called Server Distance sensitive Selective Landmarks scheme (SDSL scheme), provides a careful combination network proximities and server distances. Our experiments indicate that the proposed techniques can yield significant performance benefits.

1 Introduction

Cost-effective cache cooperation can be utilized to improve the performance of edge cache networks delivering dynamic web content in various ways such as *cooperative miss handling*, *collaborative document freshness maintenance* and *cooperative resource management* [1, 7, 11]. However, designing highly efficient edge cache networks poses several research challenges. Prominent among them are: Determining the appropriate number of edge caches required for the edge cache network, and their locations on the Internet, partitioning the edge caches into cooperative groups such that the cooperation is both effective and efficient, and designing architectures, mechanisms, and system-level facilities for supporting scalable, efficient, and reliable cooperation among the edge caches.

This paper considers the problem of partitioning the caches of an edge cache into cooperative groups. While most of the cooperative caching schemes make an underlying assumption that the cooperating caches are located in close vicinity of one another, there are no existing techniques for accurately partitioning the caches of the edge cache network into cooperative groups. Furthermore, none, to our best knowledge, has studied how the various system parameters impact cache group formation, and what effects the cache group formation can have on the performance of the edge cache network.

In this paper we systematically study the various aspects of this important problem. Our study shows that the network proximities of the edge caches and the network distances of the various caches to the origin server are the two important parameters that need to be considered while forming cooperative cache groups. Further, we design two concrete edge cache clustering techniques for accurately partitioning the caches of the edge network into cooperative groups. In summary, we make three original contributions in this paper:

- We present *selective landmarks-based group formation scheme (SL scheme)* – an Internet landmarks-based technique for clustering the caches into cooperative groups based on their mutual network proximities.
- We experimentally study the pros and cons of forming cache groups based on mutual cache proximities. Based on the findings of this study, we introduce the server distance as the second utility metric for forming cache groups.
- We present *server distance sensitive selective landmarks scheme (SDSL scheme)* – an efficient cache group formation scheme, which considers both the net-

work proximities of edge caches and their distances to the origin server. The SDSL scheme includes a clustering scheme that combines both cache proximities and server distances.

We report a series of experiments that we have performed to evaluate the proposed techniques. The initial results indicate that both SL and SDSL partitioning schemes can significantly improve the group formation accuracy thereby enhancing the performance of the cooperative edge cache network, and the SDSL scheme outperforms the SL scheme.

2 Problem Statement

Consider an edge cache network with an origin server (represented as Os) and N edge caches. Let the set of edge caches be represented as $EcSet = \{Ec_0, Ec_1, \ldots, Ec_{N-1}\}$. In this paper we assume that the scale of the edge cache network, and the locations of the edge caches and the server in the Internet are pre-decided. Also, we make a reasonable assumption that the request patterns of the edge caches exhibit considerable degree of similarity. Given such an edge cache network, the problem is to partition the caches into K disjoint groups, represented as $CGSet = \{CGroup_0, CGroup_2, \ldots, CGroup_{K-1}\}$, where K is a pre-specified parameter. Let $Size(CGroup_l)$ represent the number of edge caches in $CGroup_l$. The cache groups should be formed such that both the efficiency and effectiveness of cooperation are optimized.

An important factor affecting the performance of the cooperative edge cache network is the communication costs among the caches belonging to a cooperative group. The caches belonging to a group interact with one another very frequently for various purposes. This leads us to one of the important performance criterion that needs to be optimized while forming cache groups, namely, *average group interaction cost*. The average group interaction cost measures the average cost (latency) of transferring documents between any two caches belonging to the same group. Formally, let us define the interaction cost between two edge caches Ec_i and Ec_j (represented as $ICost(Ec_i, Ec_j)$) as the cost of transferring an average sized document between edge caches Ec_i and Ec_j. The group interaction cost of a cooperative group $CGroup_l$ (represented as $GICost(CGroup_l)$) is defined as the average of the interaction costs of all pairs of edge caches belonging to the cooperative group $CGroup_l$. The average group interaction cost of the edge cache network is defined as the mean of the group interaction costs of all groups within the edge cache network. The average group interaction cost is an indicator of the efficiency of cooperation the edge network.

Another important factor that affects the costs and benefits of cooperation in cache groups is the communication costs between the edge caches and the origin server. We defer the discussion on this factor to Section 4. For now let us suppose that the cache groups are to be formed with the objective of minimizing group interaction costs. The first challenge that we address is to devise accurate mechanisms to form cache groups such the above objective is achieved.

3 Selective Landmarks-based Clustering Scheme

The average group interaction cost of an edge cache network is determined to a large extent by the relative locations of the caches belonging to each group within the Internet. If the caches of a group are in close proximity, the group interaction cost of the cache group would be low, and vice-versa. Accordingly, the SL scheme creates cache groups such that the average network distance between any two caches (measured in terms of the roundtrip time (RTT) between them) belonging to the same group is minimized. We denote the network distance between nodes Ec_i and Ec_j as $Dist(Ec_i, Ec_j)$.

The SL scheme incorporates a unique Internet landmarks-based framework for precisely quantifying the relative positions of caches and server in the Internet. This position information is then utilized to cluster the caches into groups based on the network proximities of the caches. Conceptually, Internet landmarks (landmarks, for short) [3, 5, 10] are a set of few key Internet hosts that serve as a frame of reference for determining the relative position of any other node on the Internet. An arbitrary host H_i measures the round trip time to each of these landmarks, and uses these values to determine its relative location in the Internet. Concretely, the SL scheme works in three steps: (1) Choosing high quality landmark set; (2) Determining the relative node positions by probing the landmarks; and, (3) Creating groups through clustering edge caches

We refer to the node that coordinates the execution of the three steps as the *Group Formation-Coordinator* or the *GF-Coordinator* for short. We now explain these three steps of the SL scheme. We also illustrate each of the steps on an example edge cache network shown in Figure 1. The distance matrix in the figure shows the network distances (RTT values) between every pair of nodes. Note that the SL scheme itself does not require the construction of the complete distance matrix. We assume that the RTT values are symmetric, and hence show only the lower diagonal half of the matrix.

3.1 Choosing High Quality Landmark set

The first step in constructing the cache groups is to choose a set of landmarks, which collectively serve as the frame of reference. The caches and the origin server of the cooperative edge cache network repeatedly measure their network distance to these landmark nodes in order to determine their relative positions within the Internet. For ad-

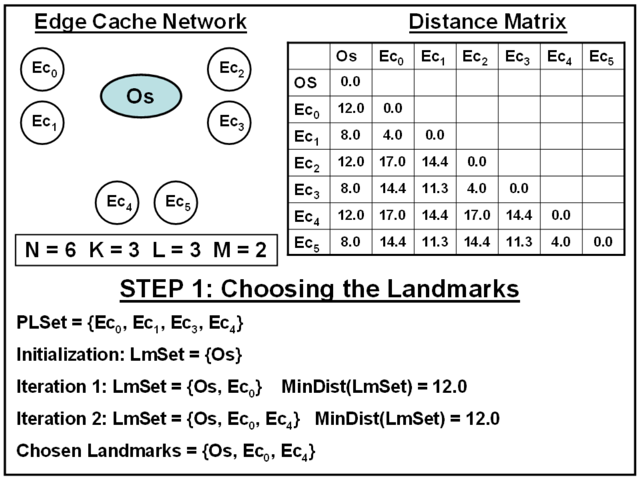

Figure 1: Choosing High-Quality Landmarks for Cloud Construction

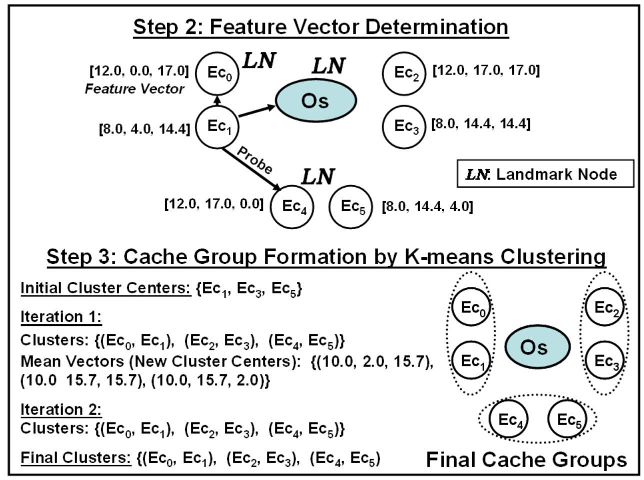

Figure 2: Determining Feature Vectors and K-means Clustering

ministrative purposes, we choose all the landmarks from the nodes belonging to edge cache network. Since these landmark nodes serve as the frame of reference for specifying the locations of the server and the caches, the quality of the landmark set is likely to have a significant impact on the accuracies of the groups formed by the scheme as well as their performances.

One of the most important properties of a good landmark set is that the landmark nodes have to be well dispersed among the set of edge caches. If the landmarks are well distributed, then the position information obtained by using them is more accurate, thereby yielding better quality cache groups. One way of ensuring that the landmarks are well distributed would be to choose them such that the minimum distance between any two nodes in the landmarks set (denoted as $MinDist(LmSet)$) is maximized. However, constructing a landmark set that satisfies this criterion requires that we know the RTT values between every pair of edge caches, which imposes significant measurement overheads on the network.

We have designed an approximation-based greedy strategy for building the landmark set. Suppose the number of landmarks is set to L. In our approach the origin server is always chosen as a landmark, since it is an important node in the edge cache network. Hence, we need to select $(L-1)$ edge caches to be included in the landmark set. In the first phase, the GF-coordinator randomly selects $M \times (L-1)$ edge caches as *potential landmark points*, where M is a configurable parameter such that $M \times (L-1) \leq N$. We call this set of edge caches as *potential landmark set ($PLSet$)*. The potential landmark points now determine their network distances to each of the other caches in the $PLSet$, and to the origin server by probing them multiple times.

In the second phase, we adopt a greedy strategy to choose $(L-1)$ edge caches from the $PLSet$, which along with the origin server forms the final landmark set. It is an iterative process, which at each iteration chooses the edge cache in the $PLSet$ that maximizes the current value of $MinDist(LmSet)$.

The bottom portion of the Figure 1 shows this step of the algorithm, wherein L is set to 3, and M is set to 2. The algorithm randomly chooses Ec_0, Ec_1, Ec_3, and Ec_4 to form the $PLSet$. The final landmark set comprises of the origin server Os, and the caches Ec_1, and Ec_4.

3.2 Determining Relative Positions of Nodes

The second step of the SL scheme determines the relative positions of the server and the caches in the edge cache network utilizing the landmarks from the previous step as the frame of reference. In the SL scheme the relative positions of the nodes are represented using simple *feature vectors*. Researchers in the past have proposed various techniques for representing the relative positions of the nodes in wide area network [5, 3, 13]. In contrast to the above, our approach uses a simple feature vector representation wherein the feature vector of a cache Ec_j, represented as FV_{Ec_j}, contains the network distance values between the cache Ec_j and various landmark points.

In the second step of the SL scheme, the edge caches construct their respective feature vectors by probing the landmark nodes multiple times and recording the average RTT values to each of them. The top portion of the Figure 2 shows this step of the SL scheme, and it also indicates the feature vectors of all the edge caches in our example network.

3.3 Creating Cache Groups through Clustering

The final step of the SL scheme clusters the edge caches into K groups on the basis of their feature vectors. This step utilizes the well-known K-means clustering algorithm to group the edge caches into cooperative groups [4]. We use the L_2 distance between two feature vectors to measure the positional dissimilarity of the corresponding edge caches.

The clustering step of the SL scheme is an iterative process, which works in three phases. In the *Initialization Phase* the algorithm *randomly* chooses K edge caches ensuring that all regions of the edge cache network are represented, and designates these edge caches as *cluster centers*. Further, each cache that is not chosen as a cluster center is assigned to its nearest cluster center, to obtain K initial clusters. The *Iterative Phase* has the following two steps which are executed iteratively until the *termination condition* is satisfied. (1) For each cache cluster, the algorithm computes the cluster's *mean vector*. These mean vectors are designated as the new cluster centers. (2) For each edge cache the algorithm calculates its L_2 distances from the new cluster centers. If the cluster center of the cache's current cluster is the nearest to the cache, it remains in the same cluster. Otherwise, the cache is re-assigned to the nearest cluster center. The iterative phase of the algorithm continues until the number of caches that were reassigned in the current iteration becomes minimal. At this stage the *Termination Phase* forms a cooperative cache group from each cluster and assigns a group ID. The lower portion of the Figure 2 demonstrates the clustering step of the algorithm.

4 Server Distance Sensitive Selective Landmarks Scheme

The SL scheme provides an accurate methodology for forming cooperative groups based on the mutual proximities of caches. However, through a simple experimental study we will show that the strategy of forming cooperative groups purely based on the network proximity of the edge caches has a serious limitation which adversely affects the performance of the cooperative edge cache network. The study also illustrates the importance of the communication costs between the various caches and origin server in cache group formation. This motivates us to design *server distance sensitive selective landmarks-based clustering scheme* - a novel cache group formation scheme that is sensitive to both mutual cache proximities and to the network distances between the server and the caches.

We will now discuss the simple experiment to illustrate the drawbacks of the SL scheme. The experimental setup is outlined in Section 5. We consider a cooperative edge cache network consisting of 500 caches, and partition the caches into specified number of cooperative groups using the SL scheme. In this experiment, the number of groups in the edge cache network is varied. We study the effects of average cache group size on the average cache latency. The average cache latency quantifies the performance of the cooperative edge cache network from clients' perspective. If a client request R_x arrives at an edge cache Ec_l of the edge cache network at time T_A, and Ec_l serves the request (possibly by contacting other caches in the group or the origin server) at time T_S, then the edge cache latency for the request R_x (represented by $EcLatency(R_x)$) is given by $EcLatency(R_x) = T_S - T_A$. The average cache latency of the edge cache network is defined as the mean of the edge cache latencies of all client requests arriving at any cache of the edge cache network within a fixed time period.

Figure 3 shows the average latencies of the 500-cache edge cache network as the average group size varies from 2 caches per group to 500 caches per group. In order to better explain the limitations of the SL scheme, the figure also indicates the average client latency of the 50 caches that are nearest (in the network proximity sense) to the origin server, and the average latency of the 50 caches that are located farthest from the origin server. Initially, all the three latencies start decreasing as the cache groups grow in size. After reaching minimum values, all the three latencies start to increase, as the average group size is further increased. This phenomenon is due to the fundamental trade-off between the group hit rates (which indicate effectiveness of cooperation) and the group communication costs (which indicate efficiency of cooperation), with respect to the size of the cache group. In general, higher hit rates lead to lower latencies, whereas higher group interaction costs results in higher latencies. As we increase the size of cache group increases, both of these parameters increase as well. However, when the group sizes are small, the hit rate improvements obtained by increasing the cache group sizes is the dominating factor influencing the average client latency. When the size of the group increases beyond a certain point the improvements in the group hit rates become small. Now the average group interaction cost becomes the dominating factor, thus leading to higher average client latencies.

Further, the average client latency of the entire edge cache network, and those of the 50 nearest and 50 farthest caches attain their minima at different average group size values. This observation illustrates a very important phenomenon: *The effects of the tradeoff between the average group hit-rates and the average group interaction costs are not uniform across all caches in the edge cache network. The interplay between these two important factors influences the performance of the cache groups in different ways and to different extents, depending upon the groups' relative positions with respect to the origin server.*

The main limitation of forming cooperative cache groups purely based on cache proximities is that it is not sensitive to the tradeoff between the group interaction costs and the group hit rates, and the varying effects this tradeoff can have on the performances of various cache groups. This limitation manifests itself in the following performance problem. When cache groups are formed purely based on cache prox-

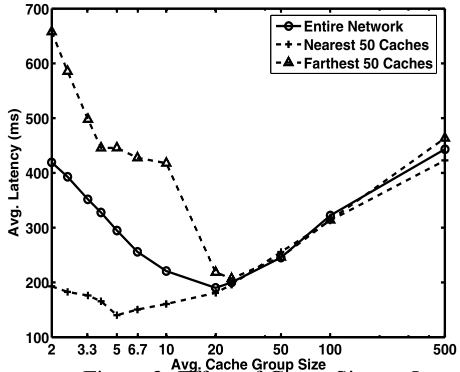

Figure 3: Effect of Group Size on Latency

imities, a considerable fraction of the edge caches would always be yielding sub-optimal performance, irrespective of the number of clouds being created.

4.1 Design of the SDSL Scheme

Our approach for countering the limitation of the SL scheme is based upon three important observations. First, in an edge cache network, for the caches that are situated farther away from the origin server the costs of processing group-wide misses would be very high. Therefore, achieving very high group hit rates would be crucial for the performance of these caches. Second, for the caches that are located close to the origin server the costs of reaching the origin server are relatively low. For these caches cooperation is beneficial only if the costs of interacting with other cooperating caches are minimal. Third, as we noted in the previous section, both the group hit rates and the group interaction costs increase as the groups become larger.

Based on the above observations, we now formulate our approach for cloud formation as follows: The cooperative cache groups are still formed on the mutual network proximities of the caches. However, we create compact cache groups (containing fewer caches) nearer to origin server and progressively increase the size of the groups as the distance between cache group and the origin server increases. This is done by allowing the far-away cache groups to be more spread out in the network than the cache groups located nearer to the origin server. Thus in our approach, the sizes of the cache groups and their spread within the network (in terms of the network distances among its caches) are proportional to their network distances to the origin server.

We now present the server distance sensitive selective landmarks scheme (SDSL scheme) - a concrete cloud formation scheme, which incorporates the above approach into the landmarks-based framework, which we have proposed. Notice that the objective of the first two steps of the SL scheme (choosing high quality landmarks and constructing the feature vectors of the nodes) is to accurately determine the relative positions of the caches and the server. Further, these two steps are orthogonal to the clustering criterion. Hence, the SDSL scheme too adopts the same steps for quantifying the relative positions of the caches and the server.

However, we need to design a clustering mechanism that forms cache groups such that each group contains caches in close network proximity, and the sizes of the cache groups increase with increasing server distances. In this paper we adapt the K-means clustering algorithm to partition the caches according to our new approach. However, any standard clustering algorithm may be similarly modified.

In the K-means clustering technique, if K groups were to be formed, the algorithm *randomly* selects K caches as initial cluster centers, while ensuring that all regions of the network are represented. Hence, any cache may be selected to an initial cluster center with equal probability. We adapt the K-means algorithm to implement our cloud formation approach by slightly altering the manner in which the initial cluster centers are chosen. The modified K-means algorithm chooses larger fraction of the initial cluster centers closer to the origin server, and selects fewer initial cluster centers as we move farther away from it. In other words, in the SDSL scheme, the probability that an edge cache is chosen as an initial cluster center is made inversely proportional to its distance from the origin server. Specifically if $Pr(Ec_j)$ represents the probability of selecting the edge cache Ec_j as an initial cluster center, and if $Dist(Ec_j, Os)$ represents the network distance between the origin server and Ec_j, then $Pr(Ec_j) \propto \frac{1}{(Dist(Ec_j,Os))^{\theta}}$, where $\propto$ represents proportional to symbol, and θ is a configurable system parameter that controls the sensitivity of the SDSL scheme towards the distances of the groups from the origin server. When θ is set to higher values, the scheme is more sensitive to server distance, and vice-versa. Once the initial cluster centers are selected according to the above equation, the SDSL scheme proceeds in a similar fashion as the SL scheme.

5 Experiments and Results

We have performed several experiments to evaluate the various aspects of the two cache group formation schemes proposed in this paper, and their effects on the overall performance of the edge cache network. Due to space constraints, in this paper we limit our discussions to three important sets of experiments.

We have implemented a discrete event simulator that models the functioning of the cooperative edge cache network. Details about the simulator and the various experimental setting are provided in the technical report version of the paper [8]. The simulations were executed on different network topologies that were generated through the

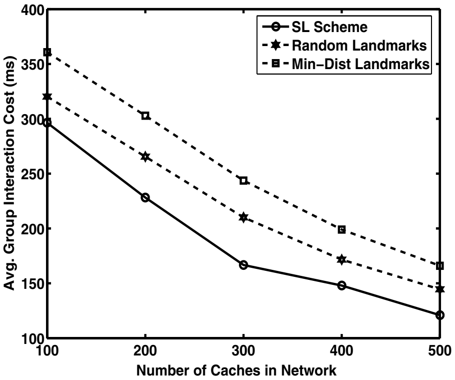

Figure 4: Effects of landmarks selection on accuracy (varying network sizes)

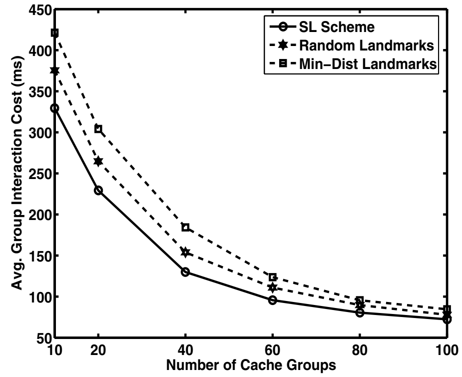

Figure 5: Effects of landmarks selection on accuracy (varying number of groups)

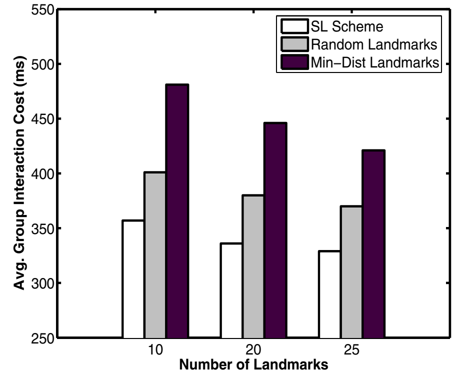

Figure 6: Effects of number of landmarks on clustering accuracy

GT-ITM network topology generator according to the hierarchical transit-stub model [12]. The configuration-settings that we have used for generating the topologies have been adopted by several previous research projects [2, 13]. The caches in the simulated edge cache network are driven by request-log files, while origin server reads continuously from an update log file. The caches implement utility-based document placement and replacement schemes [7]. Our datasets were derived from a real trace logged at a major IBM sporting and event web site [1].

We study the performance of the various mechanisms and techniques proposed in the paper using two important metrics, namely, *average group interaction cost*(defined in Section 2), and, *average latency* (defined in Section 4). The average group interaction cost can be used to measure the clustering accuracy. The average edge cache latency is used to measure the performance of the cooperative edge cache network from clients' perspective.

5.1 Evaluating Landmarks Selection Accuracy

In the first set of experiments we study the effects of network distance-based landmarks selection technique on the clustering accuracy. We consider edge cache networks containing between 100 caches and 500 caches. For each edge cache network, we generate cache groups using three landmarks-based schemes, which differ from one another only in the manner in which the landmarks are selected. The first is the SL scheme, wherein we use the greedy technique (discussed in Section 3.1) to choose the landmarks. In the second technique the landmarks are chosen randomly from the set of edge caches and the server. In the third scheme, the landmarks are chosen such that the distance between any two landmarks is minimized. We call this technique minimum distance landmarks selection technique (Min-Dist landmarks technique, for short). Except for the landmark selection technique, the second and third schemes proceed exactly in the same manner as that of the SL scheme. In this experiment the number of groups is set to be 10% of the total number of caches in the edge cache network.

Figure 4 indicates the average group interaction costs of the three schemes (in milliseconds). The average group interaction costs of the SL scheme are always lesser than the corresponding values for the other two schemes. The greedy landmarks selection technique of the SL scheme provides 8% to 26% improvement over the random landmarks selection technique, and 21% to 46% improvement over the minimum distance technique.

Our next experiment (Figure 5), evaluates the three landmarks selection techniques when the number of cache groups is varied. For this experiment we consider an edge cache network comprising of 500 caches. We then form different numbers of cache groups using each of the three techniques and measure the average group interaction costs and the average cache latencies. Figure 5 indicates the average group interaction costs of the three schemes. The results show that the high quality landmarks selection technique of the SL scheme results in better clustering accuracies than the other two techniques at all K values.

In the third experiment we evaluate the effect of the number of landmarks on accuracy of clustering achieved by different landmark selection schemes. The bar graph in Figure 6 indicates the average group interaction cost for the SL scheme, random landmarks scheme and Min-Dist landmarks scheme for an edge cache network of 500 caches when the number of landmarks is set at 10, 20 and 25. The number of cooperative groups formed in this experiment is 10. The graph shows that the clustering accuracies of all the three schemes improve as the number of landmarks increases. When the number of landmarks is increased beyond 25, the improvements are minor for all three schemes. Also, the selective landmarks-based clustering scheme performs better than both random landmarks scheme clustering

[1]The 2000 Sydney Olympic Games web site

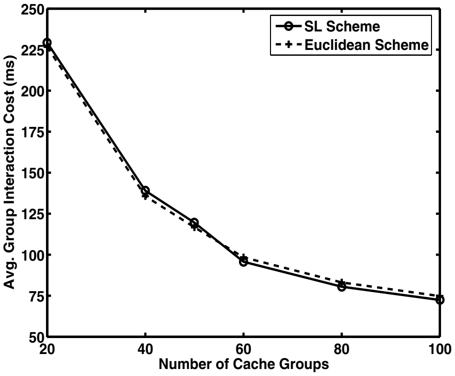

Figure 7: Effects of position representation on accuracy (varying number of groups)

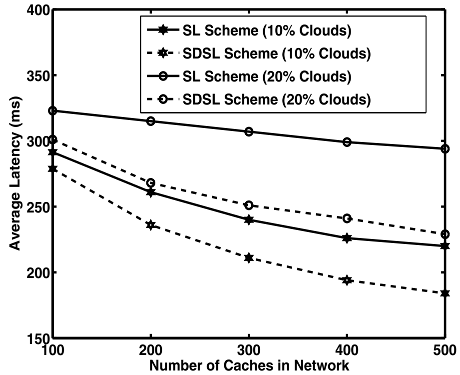

Figure 8: Comparison of SDSL and SL schemes wrt latency(varying network sizes)

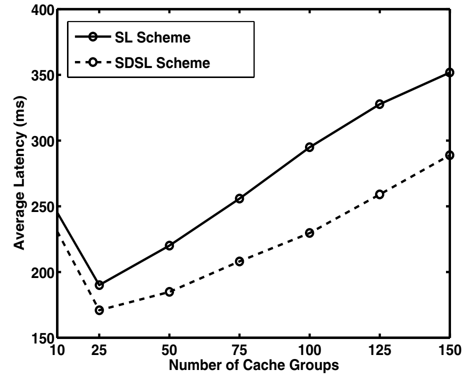

Figure 9: Comparison of SDSL and SL Schemes wrt latency (varying number of groups)

and the Min-Dist landmarks selection scheme irrespective of the number of landmarks used.

5.2 Evaluating Feature Vector Representation

In both SL and the SDSL schemes, the relative positions of the caches and the server within the network are represented through their feature vectors. Our next experiment studies the effects of the feature vector representation scheme on the clustering accuracy.

We compare our feature vector representation technique to the Euclidean-space representation method, wherein the nodes are mapped into a D-dimensional Euclidean space based on their relative distances to various landmarks [3, 5, 10]. The primary motivation for Euclidean-space mapping is to minimize the error between the actual network distances between nodes and the corresponding L_2 distances between their assigned coordinates. However, the process of mapping the nodes into a Euclidean space is computationally intensive. In our experiments, we use the Global Network Positioning (GNP for short) [5] technique to map the caches and server into Euclidean space. For both schemes we use the same sets of 25 landmarks, which are chosen through the greedy technique and form clouds through the K-means algorithm. In our discussion, we refer to the cache clustering scheme using Euclidean-space coordinates as *Euclidean-space clustering scheme*.

Figure 7 shows the average group interaction costs of the SL scheme and the Euclidean-space clustering on a cooperative edge cache network of 500 caches, when the numbers of groups varies from 10 to 100. The results show that the average group interaction costs of the SL schemes (which uses feature vector representation) are very similar to those of Euclidean-space clustering scheme. While for some K values, the Euclidean-space clustering performs marginally better than our scheme, for others the SL scheme yields slightly lower group interaction costs. This shows that for the cache group formation application, the simple feature vector representation scheme is sufficient to yield accurate clusters.

5.3 Evaluating the SDSL Scheme

In the final set of experiments we evaluate the SDSL strategy. The objective of these experiments is to study the benefits obtained by taking both the mutual proximities of caches and the distances between the server and various caches into account while forming the edge cache groups. We consider edge cache networks containing between 100 and 500 caches and form cache groups using both the SL scheme and the SDSL scheme. We quantify the effects of the cache group formation strategies on the performance of the cooperative edge cache network by their average cache latency values. In order to ensure fairness, we use the same set of 25 landmarks for both schemes.

Figure 8 indicates the average cache latency values of the two schemes, at two different experimental settings, as we vary the edge cache network size. In the first setting the number of cache groups in each edge cache network is set to 10% of the total number caches in the respective edge cache network, whereas for the second it is set to 20%. The SDSL strategy performs better than the SL scheme for edge cache networks of all sizes at both experimental settings. For example, the SDSL strategy improves the latency by more than 27% for an edge cache network with 500 caches when the number of groups is set to 20% of the total number of caches. Figure 9 indicates the average client latency of the SL and the SDSL strategies for the 500-cache edge cache network when the number of groups is varied. We again see that irrespective of the number of cache groups formed, the SDSL strategy always yields lower cache latency values than the cache proximity strategy. From the results of these experiments, we note that the SDSL improves the overall performance of the cooperative edge cache network by overcoming the drawbacks of the SL scheme.

6 Related Work

Cooperative caching was first proposed in the context of client-side proxy caches [11]. The research mainly focused on designing architectures and algorithms for caching predominantly static web content. Recently, edge cache network have adopted the concept of cache cooperation for serving client requests and for designing low-cost consistency maintenance techniques [1, 6, 7]. However, very few of these schemes have addressed the important problem of forming effective and efficient cooperative cache groups. In client-side cooperative caching systems, the caches are usually located within a single organization. Hence, the cooperation costs play a minor role in deciding which costs should cooperate with one another. In contrast, the caches of edge cache networks are globally distributed. For these systems, the manner in which the cooperative cache groups are formed has a significant impact on the effectiveness of cooperation.

The work by Shah et al. [9] on cooperative data dissemination system includes a scheme for constructing data dissemination trees. Our cooperative cache group formation technique differs from this tree construction scheme in a fundamental way. While in their scheme the dissemination trees are constructed based on the coherency requirements of the repositories, our mechanism for cache group formation is based on the network proximity of the caches and the network distances from the server to the caches.

Accurately quantifying the relative positions of nodes is important for the performance of many large-scale distributed systems. Recently, many researchers have proposed to utilize Internet landmarks-based techniques to address this problem. Schemes such as Global Network Positioning [5] and Vivaldi [3] map the Internet nodes into Euclidean space with the objective of minimizing the error between the actual distance of nodes and the L_2 distance of their Euclidean coordinates. In our schemes, the relative position of the caches and server are represented through simple feature vectors.

In short, the work described in this paper is unique in the sense that it proposes novel schemes for forming effective cooperative groups in edge cache networks.

7 Conclusions

While a number of architectures, protocols and techniques have been proposed to enhance cooperation within cache groups, the problem of partitioning the edge caches into cooperative groups has not received much attention. In this paper we identified two important utility factors that play crucial roles in cooperative group formation in edge cache networks. We proposed two concrete edge cache clustering schemes for partitioning the caches of a given edge network. The first scheme is called *selective landmarks-based scheme (SL scheme)*, and it forms groups on the basis of the network proximities of the edge caches. The second scheme is referred to as *server distance sensitive selective landmarks scheme*, which takes into account of both the network proximities of caches and their server distances when creating cache groups. We have reported our experimental evaluation of the proposed schemes. The initial results indicate that these techniques provide significant performance benefits.

Acknowledgements

This research is partially supported by NSF CSR, NSF ITR, DARPA, IBM Faculty Award, IBM SUR grant, and a grant by the UGA Research Foundation.

References

[1] Akamai Technologies Incorporated. http://www.akamai.com.

[2] M. Castro, P. Druschel, A.-M. Kermarec, and A. Rowstron. Scribe: A Large Scale and Decentralized Application-level Multicast Infrastructure. *IEEE Journal on Selected Areas in Communications*, 20(8), 2002.

[3] F. Dabek, R. Cox, F. Kaashoek, and R. Morris. Vivaldi: A Decentralized Network Coordinate System. In *SIGCOMM-2004*, August 2004.

[4] A. K. Jain, M. N. Murty, and P. J. Flynn. Data Clustering: A Review. *ACM Computing Surveys*, 31(3), 1999.

[5] E. Ng and H. Zhang. Predicting Internet Network Distance with Coordinates-Based Approaches. In *IEEE-INFOCOM*, June 2002.

[6] A. Ninan, P. Kulkarni, P. Shenoy, K. Ramamritham, and R. Tewari. Scalable Consistency Maintenance in Content Distribution Networks Using Cooperative Leases. *IEEE - TKDE*, July-August 2003.

[7] L. Ramaswamy, L. Liu, and A. Iyengar. Cache Clouds: Cooperative Caching of Dynamic Documents in Edge Networks. In *ICDCS-2005*, June 2005.

[8] L. Ramaswamy, L. Liu, and J. Zhang. Constructing Cooperative Edge Cache Groups Using Selective Landmarks and Clustering. Technical report, CS Department, University of Georgia, 2005.

[9] S. Shah, K. Ramamritham, and P. Shenoy. Resilient and Coherence Preserving Dissemination of Dynamic Data Using Cooperating Peers. *IEEE - TKDE*, July 2004.

[10] L. Tang and M. Crovella. Virtual Landmarks for the Internet. In *Internet Measurement Conference*, 2003.

[11] A. Wolman, G. M. Voelkar, N. Sharma, N. Cardwell, A. Karlin, and H. M. Levy. On the Scale and Performance of Cooperative Web Proxy Caching. In *SOSP-99*, December 1999.

[12] E. W. Zegura, K. Calvert, and S. Bhattacharjee. How to Model an Internetwork. In *IEEE-INFOCOM*, 1996.

[13] J. Zhang, L. Liu, C. Pu, and M. Ammar. Reliable Peer-to-peer End System Multicasting through Replication. In *IEEE P2P 2004*.