Computer Standards & Interfaces 49 (2017) 44–58

Contents lists available at ScienceDirect

Computer Standards & Interfaces

journal homepage: www.elsevier.com/locate/csi

A hierarchical P2P clustering framework for video streaming systems

Sercan Demirci [a], Asil Yardimci [c], Muge Sayit [a], E. Turhan Tunali [d], Hasan Bulut [b,*]

[a] International Computer Institute, Ege University, Bornova, Izmir 35100, Turkey
[b] Department of Computer Engineering, Ege University, Bornova, Izmir 35100, Turkey
[c] Tubitak National Research Institute of Electronics and Cryptology, Gebze, Kocaeli 41470, Turkey
[d] Department of Computer Engineering, Izmir University of Economics, Balcova, Izmir 35330, Turkey

ARTICLE INFO

Article history:
Received 25 January 2016
Received in revised form
4 May 2016
Accepted 1 August 2016
Available online 6 August 2016

Keywords:
Peer-to-Peer networks
Overlay networks
Hierarchical clustering
PlanetLab

ABSTRACT

In this study, a novel overlay architecture for constructing hierarchical and scalable clustering of Peer-to-Peer (P2P) networks is proposed. The proposed architecture attempts to enhance the clustering of peers by incorporating join, split, merge and cluster leader election mechanisms in a fully distributed manner. It takes delay proximity of peers into account as distance measure. By constructing hierarchical clustering of peers, the control message overhead and maintenance such as host departure/host join overhead are decreased. Theoretical comparisons on overheads of the proposed system with that of other systems from literature are studied. The control mechanism for dynamic peer behavior of the architecture is tested over PlanetLab. The performance metrics used are end-to-end delay, diameter, cluster head distance, occupancy rate, peer join latency, accuracy and correctness. The test results are compared with Hierarchical Ring Tree (HRT) and mOverlay architecture. In addition, a P2P video streaming application is run over the proposed network overlay. Streaming tests show that video streaming applications perform well in terms of received video quality if hierarchical clusters considering delay proximity are used as underlying network architecture.

1. Introduction

Overlay networks support many applications such as file sharing, video streaming and resource discovery. In addition, overlay networks provide flexibility, scalability and adaptability. They can also be used for multicast routing that is independent of the network layer support [1].

Structured overlay networks utilize the network topology and query content to make operation more efficient. Many of those systems use Distributed Hash Table (DHT) [2]. Among the structured overlay networks, Chord [3], CAN [4], Pastry [5] and Tapestry [6], have emerged as flexible infrastructures for building large P2P systems based on various DHTs and have excellent load balancing properties. There has also been research on range queries [7]. But these approaches decompose the query range in several sub-ranges which increases the overhead to process a range query. Also, [8] exploits the network proximity for Pastry, which is a DHT based structured P2P system, and requires the knowledge of some of Pastry nodes at any location before joining the system. [9] studied how geometries like hypercubes, rings, tree-like structures, and butterfly networks used in DHT based P2P systems affect the resilience and proximity of the system. They underlined the importance of hop-proximity and provided some insight that may be useful in DHT routing designs.

Unstructured P2P networks do not have any particular structure. They are composed of peers joining the network without any prior knowledge of the topology. However, peers may become overloaded, the system may not scale well and there may be sudden increases in system size [2]. Also message overhead of a query search may also be high in unstructured networks due to the flooding technique used to send messages across the overlay [10]. Gnutella [11], Freenet [12], KaZaA [13], BitTorrent [14,15], HybridFlood [16] are examples of such unstructured overlay networks.

Clustering is an important issue in many large scale distributed systems. Constructing an efficient clustering can significantly enhance scalability. Hierarchy can be added to clustering for the purpose of improving message complexity and management. However, clustering is a challenging problem because of the network dynamics in P2P networks. Moreover, one usually faces a choice of clustering metric problem. While content metric helps organization according to interest [17–22], delay proximity improves response time. Partitioning-based, density-based, grid-based and model-based methods are other alternatives to hierarchical-based clustering. An excellent survey of clustering techniques can be found in [23].

In this work, we propose an architecture based on hierarchical clustering of peers according to delay proximity. The proposed

* Corresponding author.
E-mail addresses: sercan.demirci@ege.edu.tr (S. Demirci), asil.yardimci@tubitak.gov.tr (A. Yardimci), muge.fesci@ege.edu.tr (M. Sayit), turhan.tunali@ieu.edu.tr (E.T. Tunali), hasan.bulut@ege.edu.tr (H. Bulut).

http://dx.doi.org/10.1016/j.csi.2016.08.002
0920-5489/

architecture attempts to enhance the clustering of peers with join, split, merge and leader election algorithms. Unlike other systems, a threshold parameter is incorporated into the join and the merge algorithms. A peer joins a cluster if and only if its distance with the cluster leader is below a predefined threshold - different round trip time (RTT) values are used for different levels of the hierarchy. In this way, delay proximity within the cluster is limited with the threshold. Merge operation is used to eliminate clusters that have a few members. Two clusters are merged into a single cluster as long as the delay proximity between leaders is below the predefined threshold for that level. The studies in the literature do not use any threshold value for join and merge processes.

In join process, a peer measures its distance with the cluster leader. Since this value is used whether or not a peer joins the cluster, the cluster leader is desired to be at the center of the cluster. Thus, leader has minimum sum of delays with other peers. Choosing a peer at the center of the cluster affects the accuracy of the clustering. The cluster split operation is similar to k-center problem which is known to be NP-Hard. We have developed a novel approximation algorithm for split operation to keep the message complexity low.

The performance metrics used are clustering accuracy [24], correctness [24], end-to-end delay, diameter of the cluster, cluster head distance, occupancy rate and peer join latency. Due to the network dynamics in P2P networks, the leader of the cluster may change and this affects the clustering accuracy and correctness that shows the quality of clustering. We note that some of the well known architectures such as NICE, ZIGZAG and HRT did not report accuracy and correctness measures. To evaluate the performance of the proposed architecture, we tested it over PlanetLab [25]. It is a deployment environment, which consists of nodes located across the world and connected to each other with real Internet links. Researchers have the opportunity of deploying new protocols, running experiments and measuring network performance in realistic Internet scenarios. We compared our proposed architecture, HRT and mOverlay. We have studied the theoretical bounds for overhead and compared the values with that of mOverlay and HRT. Furthermore, we tested the performance of a P2P video streaming application running over the proposed architecture and gave the performance results for various sizes of networks.

The rest of the paper is organized as follows. In Section 2, proposed architecture is explained. Section 3 reports performance on PlanetLab environment together with a real video streaming application. Section 4 is devoted to related work and finally, in Section 5, concluding remarks are made.

2. Proposed hierarchical overlay network architecture

Construction of a hierarchical overlay over P2P network is a crucial task for many applications such as video streaming and search [26–29]. P2P overlay network clustering directly affects the performance of the applications. Nodes in a P2P system can be clustered based on geographic locations, delay proximity or interest proximity of nodes [30].

In this study, delay metric is used as a clustering parameter. Hence, nodes closer to each other based on this metric are clustered into the same cluster. The proposed architecture utilizes hierarchical levels, where clusters at the leaves of this hierarchy are composed of real nodes while clusters at the intermediate levels are composed of virtual nodes. Each cluster has a maximum delay diameter and/or a maximum number of nodes. Virtual nodes are leaders of the lower level clusters.

Fig. 1 shows a two-level hierarchical structure. Each cluster at level 0 has a leader and those leaders are members of clusters at level 1. One of the nodes in cluster at level 1 is chosen as the cluster leader.

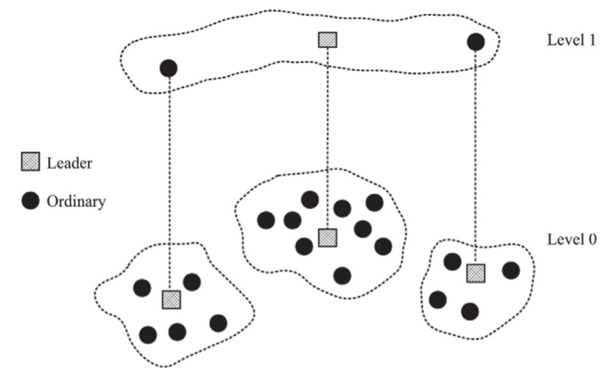

Fig. 1. Two-level overlay network.

Algorithms for cluster construction, cluster splitting, cluster merging, cluster leader election and hierarchy formation are developed.

2.1. Joining of a new peer

In our proposed architecture, there exist rendezvous points (RPs) whose addresses are known by every peer for bootstrapping mechanism. We use more than one RPs to prevent the bottleneck at the highest level, which can be a single point of failure. Each peer and cluster in the system has a unique identier. When a peer joins the system, it needs to find its proximity to some set of nodes. However, in some cases, it may not be possible to measure the RTT with these nodes, since they may be behind firewalls or NATs, where ICMP packets are blocked. In this case, an estimate of the RTT can be obtained using Vivaldi algorithm [31]. The Algorithm 1 (see Appendix A) finds the closest peer with respect to RTT. However, if RTT measurement fails with some nodes, it uses Vivaldi algorithm to estimate the RTT.

A new peer X, first communicates with one of the RPs and requests to join the system. The first peer that joins the system forms a cluster at level 0 and becomes the leader of that cluster. In addition, another cluster is formed at level 1 to represent the leader of the cluster at level 0. Other nodes that join the system, receive the list of nodes of the cluster at the highest level from RP. The joining peer, then, measures its delay to other nodes in the cluster. Then, it chooses the closest node to itself. If the delay with that node is below a threshold, then it joins that cluster. Otherwise, it forms a new cluster and assigns itself as a cluster leader. The threshold differs from level to level and it is expected to decrease as level number decreases in the hierarchy. Virtual clusters, which reside in upper layers of the hierarchy, are used for administrative purposes such as join, split and merge functionalities. The number of levels affects the number of times these operations are executed. Hence, the number of levels also affects the number of clusters in the system. The cluster size c_s is limited between $[k, 3k-1]$ and the total number of clusters including virtual clusters in the worst case is $(N-1)/(k-1)$ as shown in Theorem 1, where k is an integer number to define cluster limits and N is the total number of nodes in the system. Fig. 2 shows a case of the join process and the join algorithm is depicted in Algorithm 2 (see Appendix B). Theorem 2 shows that the worst-case join overhead of a node is $O(k\log_{c_s}N)$.

Theorem 1. *There are at most* $\frac{N-1}{k-1}$ *clusters in the system for* $k > 1$.

Proof. Each cluster size is between $[k, 3k)$ and total number of nodes is N. The clusters at the lowest level of the tree (leaf) are composed of physical nodes. The intermediate nodes are virtual nodes (including the root node). The height of the tree is h. Then there are $h+1$ levels (root node resides at level h) in the tree. Hence, $N = k^{h+1} \Rightarrow h = \log_k N - 1$. Since, each cluster can have at least k nodes (level 0), there are N/k clusters at the leaf, N/k^2