A Topology Adaptation Protocol for Structured Superpeer Overlay Construction

Changyong Niu, Jian Wang, and Ruimin Shen

Department of Computer Science and Technology,
Shanghai Jiaotong University,
Shanghai, China
{cyniu, jwang, rmshen}@sjtu.edu.cn

Abstract. Peer-to-peer networks can be divided into structured and unstructured based on their overlay topologies. In reality, unstructured p2p networks with superpeers have proved their capacities to support millions of users simultaneously. However, applications deployed on this kind of overlay networks, such as file-sharing, require flooding or gossip-based message routing, which puts more overhead on underlying networks and provides no guarantee on resource discovery. In this paper we propose an overlay adaptation protocol which allows structured superpeer overlay construction from unstructured p2p overlay networks with the potential to leverage the advantages of structured p2p overlay networks such as efficiency, scalability and guaranteed look-up services. The simulation result shows that our protocol can build the structured superpeer overlay with efficiency and scalability.

1 Introduction

P2P networks can be classified as being either structured or unstructured. Recent developments of structured [1,2] and unstructured [3,4] overlay networks point to a new direction for overlay research to address these major challenges such as scalability, efficiency and flexibility.

Structured P2P networks implement efficient distributed P2P lookup services by organizing the peers in structured overlay networks and determining routing messages to the peer that is responsible for a given key. These implementations of distributed lookup service are often referred to as Distributed Hash Tables (DHTs). In contract, nodes in unstructured overlay network have the flexibility to choose the number and destinations of their connections, and adapt them to network heterogeneity for improved network performance. Especially by introducing the concept of superpeer [5, 6] into unstructured overlay network, the heterogeneity of P2P networks is further explored without compromising their decentralized nature. However, unstructured overlay networks often require flooding or gossip to route messages, which limits their efficiency and puts more overhead on the underlying physical networks.

In this paper we explore the integration of structure and unstructured overlay network, in a way that superpeers are selected from operational unstructured overlay networks and further organized into structured network to tackle the challenges of scalability, efficiency and flexibility. The potential advantages of this method include:

H. Zhuge and G.C. Fox (Eds.): GCC 2005, LNCS 3795, pp. 953 – 958, 2005.

- Load Balance: load balance among superpeers is achieved not only on the number of object references maintained but also on the number of clients handled by superpeers.
- DHT Enhanced: By organizing superpeers into structured overlay, applications over resulting overlay networks can easily take advantages of DHTs.
- Efficiency: simulation shows that the presented protocol is efficient for superpeer overlay construction in terms of underlying network overhead.

Rest of this paper is organized as follows. Related superpeer overlay construction protocols are presented in section 2. In section 3, the proposed protocol for building structured superpeer overlay will be described in detail. In section 4, the protocol performance will be analyzed and experimental results will be illustrated.

2 Related Work

The concept of superpeer has been introduced to improve the performance of P2P overlay networks and above applications. A superpeer is expected to be more stable and powerful to operate as a server for a group of common peers, and as an equal to other superpeers. Some superpeer overlay designs have been introduced in literature. The system in [7] is based on a two-tier hierarchical overlay network. All nodes are part of the bottom tier: a Chord ring. In a Chord ring, each node keeps track of other nodes that are 1/2-ring away, 1/4-ring away, 1/8-ring away, etc. To construct the upper superpeer tier, the Chord ring is divided into a number of arcs with equal length. In each arc, some node is selected as superpeer of this arc, and all superpeers form the upper tier. Another two-tier superpeer overlay is the system in [8]. Each node belongs to exactly one group of the bottom tier and different groups form separate overlays which may or may not be the same in structure. One or more superpeer is selected from each group and all superpeers form the upper tier overlay which is a modified Chord ring. And the relationship between superpeer and group peer is many-to-many.

Other superpeer overlay networks include HONet [9]. In HONet, peers are first clustered into groups based on localities in a structured way. Then superpeers selected from each group are connected using random walk. Superpeer selection and hierarchical overlay construction protocol presented in [6] is gossip based. The resulting overlay resembles the characteristics of one-to-many relationship between superpeer and group peer; random network structure of superpeer overlay and minimized number of superpeers (also called target superpeer overlay). Same as HONet, nodes in groups are not totally separated but are connected through gossip based links. So the robustness of protocol is achieved in price of network overhead. However the total overhead of overlay construction scales sub-linearly with the size of the network, which is very efficient.

P2P overlay networks have been used in many applications, such as application-level multicast, file-sharing, media streaming, data storage and most recently semantic layer of Knowledge Grid [10].

3 Structured Superpeer Overlay Construction

We start the superpeer overlay construction from a network consisting of a large collection of nodes, in which each node having a unique identifier can potentially communicate with any other node through message exchanges. Also the network is highly dynamic with each node joining and leaving, even crashing at any time. Nodes in the network are heterogeneous with different capacities. To differentiate the nodes according to these heterogeneities, each node is associated with a value representing its capacity according to its own characteristics. Based on this information, we can select the nodes with larger capacities to act as superpeers and determine the number of clients that a superpeer may handle.

The protocol presented in this paper will gradually adapt an unstructured overlay into a structured superpeer overlay, which has a number of important advantages compared to the flat unstructured overlay networks.

- Handling heterogeneity: the upper-layer overlay is consisting of more powerful and stable superpeers.
- Transparency: client joining, leaving and crashing under some superpeer are totally transparent to other groups. And the client floating, i.e. from one superpeer to another, is also transparent from object lookup services.
- Efficiency: the structured organization of superpeers may further provide more efficient lookup services; also this efficiency may further lead to fewer network overheads.

3.1 Superpeer Selection and Structurization

Instead of building a superpeer overlay from scratch, we build the superpeer overlay as an additional overlay, superimposed over an existing connected topology. The rationale behind this is that even if all superpeers crashed simultaneously, the network would not be disconnected and the superpeer overlay may still be rebuilt from scratch.

We use tapestry as the superpeer overlay structure and implement the adaptation algorithm in Peersim. Although specific structured overlay protocol is used in this work, the overlay adaptation paradigm is general enough to adopting other DHT protocols. The construction may be divided into two stages.

The idea of the first stage superpeer selection is fairly simple. We assume each node in the existing overlay knows about its own capacity. Besides the neighbor table, each node also maintains a table of superpeers and superpeer/client table. The superpeer table of a node *A* maintains a sample of superpeers that *A* has already seen in current overlay; and superpeer/client table maintains the current clients of *A* if *A* is a superpeer or the current superpeer of *A* if it is a client. The role of superpeer and client is determined by competition. The competition strategy is illustrated in Table 1**,** which is similar to the superpeer selection algorithm in [6]. However, instead of generating the minimum number of superpeers, we have adapted the algorithm for load balance purposes. Each superpeer has three satisfaction levels: unsatisfied if its number of clients less than its pre-defined min-client threshold; satisfying if the number of clients is between min-client and max-client threshold; and over-satisfied if the number of clients exceeds max-client. Over-satisfied is only allowed when a

superpeer can take over another superpeer and all its clients. And load balance is handled in top-layer overlay compete strategy.

Table 1. Let p being a local peer and *p-superpeer* being its superpeer table; L_p is p's current load; C_p is the capacity of p; $C_{p\text{-}max}$ is the max-client threshold of p; $C_{p\text{-}min}$ is the min-client threshold of p;let r, q being temporary remote peers

```
(1)  S={r| r ∈ p-superpeer && C_r > C_p && L_r < C_r-max }
(2)  Probing the load (used capacity) L_q of q that q ∈ {q | q belongs
     to S}
(3)  Finding q that (C_q-L_q) > L_p
(4)  If found q then q accommodate p and all its clients
(5)  Else finding q that max(L_q-max-L_q)
(6)     Transfer min (L_q-max – L_q, L_p) clients for p to q
(7)     if exists client r ∈q and C_r > C_p then
(8)        exchange the role of r and p, letting p to be a client of q and r
(9)        to be a superpeer taking over the clients of p
```

After the selection of the first stage superpeers, every superpeer in current snapshot will apply to join the upcoming superpeer overlay. Here we draw the same assumption as tapestry does, that there are well-known federation and static tapestry nodes in the overlay for bootstrap purpose.

Since the goal of the top-tier overlay is load balance, it is reasonable that the resulting overlay contains large capacity superpeers and each superpeer have some client in its domination. To achieve this goal, we design a superpeer compete strategy which is described as follows.

Table 2. p is the node querying for a superpeer; S_0 is it's accessing point to top-layer superpeer overlay; L_X is node X's current load; C_X is the capacity of X; $CMAX_X$ is the max-client threshold of X; $CMIN_X$ is the min-client threshold of X; TEMP, t, S are temporary variables

```
(1)  TEMP =p; S= S0;
(2)  while (S != null){
(3)     if (LS > CMAXS) // S is not over-satisfied
(4)      { S accepts TEMP; break;}
(5)      find k from S's client table which has the max capacity
(6)      if (Ck > CTEMP)
(7)      {   t.superpeer=k.superpeer
(8)          t=k;       // TEMP replace k in the client table of S
(9)          TEMP = t; }
(10)     S pick out next hop H for S in his routing table
(11)     if (H == null)
(12)     { TEMP join as a supernode}
(13)     else
(14)     { S = H}
(17) }
```

p is the node querying for a superpeer; S_0 is it's accessing point to top-layer superpeer overlay, and L_X is node X's current load with C_X being the capacity of X; $CMAX_X$ is the max-client threshold of X; $CMIN_X$ is the min-client threshold of X; The superpeer finding algorithm is described as follows and illustrated in Table 2**.**

1. If S_0 is not over-satisfied and has the spare capacity to accept p as a client then the finding is success.
2. If S_0 could not accommodate p, take a look in S_0's client table to find C_k, where k is the node with largest capacity in the client table, and compare it with C_p. If C_p is larger then forward the finding to the routing successor of S_0.
3. If C_k is larger, then node p becomes the client of S_0, and the client node k gracefully leaving S_0, and using S_0 as the accessing point to find himself a superpeer. (k replaces p to be the node querying for a superpeer.)
4. The adaptation process continues until p successfully finding himself a superpeer or the the routing successor of S_0 being null. In the latter case, p joins the superpeer overlay as a superpeer, and the finding process is ended.

3.2 Load Balance

We design a milder way to explore the capacity heterogeneity and with a focus on load balance and fairness. To achieve this, each selected superpeer initializes a (min-client, max-client) pair. Each superpeer aggressively maintains its number of clients between min-client and max-client. When a new superpeer joins the top-tapestry overlay, it grabs clients from its neighbors to achieving this goal. And when the number of clients of a superpeer exceeds max-client, it splits some clients to its neighbors. As long as the resulting number of clients is between the threshold, a superpeer is always willing to accept (or split) clients from (or to) other superpeers. The rationale for a client could be reassigned to another superpeer is explained as follows. In a structured superpeer overlay, information of objects is published as (key, value) pairs. A key says K_1 is derived from the keywords associated with an object says O_1, and the value coupled with K_1 is corresponding to the machine hosting O_1. All (key, value) pairs are distributed among top-tapestry nodes as object references. As long as a lookup for a key gets the reference, the "value" within the reference could direct the query to the hosting machine, no matter which superpeer the hosting machine belongs to. So the client floating is totally transparent from object lookup.

4 Analysis and Future Work

Because we focus our current work on efficient protocol design of structured superpeer overlay construction, in the following we compare the efficiency of our protocol with that of [6] in terms of network overhead and operations performed by each node to form such superpeer overlays. To compute the overall network overhead, we aggregate messages sent by all nodes for neighbor information probing and neighbor transferring which are needed for overlay construction. The aggregations are performed under different network sizes with a power law node capacity distribution and max node capacity of 500. The comparison of total network overhead is illustrated in figure 1. For simplicity, we average the total messages

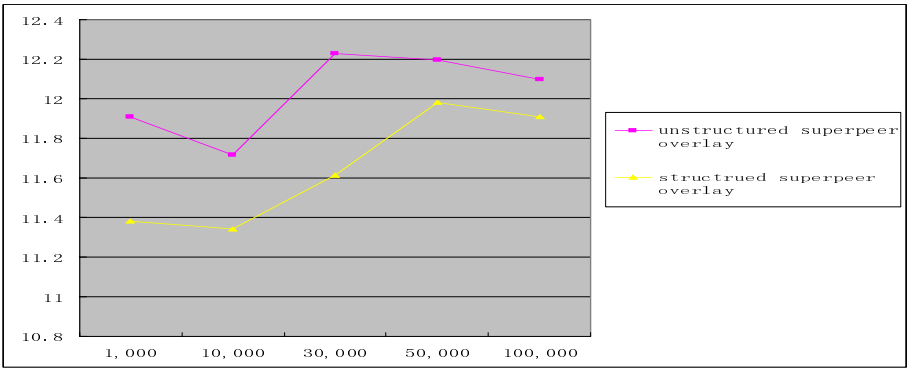

Fig. 1. Network overhead comparison between the constructions of unstructured and structured superpeer overlays: the main reason that our protocol performs better is because we focus on load balance among superpeers

among all nodes in the network. The average could be read as number of operations taken by each node for probing or transferring. As for our protocol, the operations also include messages for tapestry node join and routing table maintenance during overlay construction. And the experiment result is illustrated in Figure 1.

In this paper, we propose a topology adaptation protocol for building structured superpeer overlay. The simulation result shows that the protocol can build a structured superpeer overlay from an unstructured p2p network with efficiency and scalability. Future work includes to incorporate the model with the ability of handling locality heterogeneity and to explore the behavior of such hierarchical topology under churn.

References

1. Stoica, I., Morris, R., Kaashoek, M. and Balakrishnan, H., Chord: a scalable peer-to-peer lookup service for internet applications, Proceedings of ACM SIGCOMM 2001
2. Zhao, B.Y., Kubiatowicz, J. and Joseph, A.D., Tapestry: an infrastructure for fault-tolerant wide-area location and routing, Technical Report, University of California, Berkeley, 2001
3. "Gnutell." http://www.gnutell.com
4. Kermarrec, A.M., Massoulié, L. and Ganesh, A.J., Probabilistic reliable dissemination in large-scale systems. IEEE Transactions on Parallel and Distributed Systems, 2003
5. Yang, B. and Garcia-Molina, H. Designing a super-peer network, Proceedings of the 19th International conference on Data Engineering, 2003
6. Montresor, A., A robust protocol for building superpeer overlay topologies, Technical Report UBLCS-2004-8
7. Manku, A.T., Cheng, Y., Kumar, V., and Savage, S., Structured superpeers: leveraging heterogeneity to provide constant-time lookup proceeding of IEEE Workshop on Internet Applications, 2003
8. Garces-Eric, L., Biersack, E.W., Felber, P.A., Ross, K.W., and Urvoy-Keller, G., Hierarchical peer-to-peer systems, Proceeding of INFOCOMM'03, 2003
9. Ruixiong Tian, Yongqiang Xiong, Qian Zhang, Bo Li, Ben Y. Zhao, Xing Li, Hybrid overlay structure based on random walks, Proceeding of iptps05, 2005
10. Zhuge, H., Liu, J., Feng, L., Sun, X., He, C., Query routing in a peer-to-peer semantic link network, Computational Intelligence, Volume 21, 2005