M-Ring: A Distributed, Self-Organized, Load-Balanced Communication Method on Super Peer Network

Tsung-Han Lin, Tsung-Hsuan Ho, Yu-Wei Chan and Yeh-Ching Chung
System Software Laboratory, Department of Computer Science, National Tsing-Hua University
jjoha58@gmail.com, anson@thho.net, ywchan@sslab.cs.nthu.edu.tw, ychung@cs.nthu.edu.tw

Abstract - *Many peer-to-peer file sharing systems have been proposed to take the locality and heterogeneity into account. The two-layered architecture is one of the most widespread systems with abilities by classifying peers into groups and serving some powerful peers as the super peers. In order to communicate with other sets of super peers, each super peer has to connect with all the other super peers within its neighboring group or other groups by some gateway-like super peers. However, it may be the problem of the single-point-of-failure if using peers as the gateway-like peers. In this paper, we propose a distributed, self-organized and load-balanced communication method – M-Ring. In this method, each super peer connects with other sets of super peers within its neighboring group by constructing its link table. Also, each super peer in the different groups has a unique identity within the same identity space. Besides, we notice a "overlap handle scope" feature while all super peers who are in the different groups are in the same identity space. We use this feature to enhance the efficiency of query of super peers. Our method can reduce the total requirements of storage space and bandwidth by storing different portion of metadata. The simulation results show that our approach efficiently reduces the overhead of connections among super peers within their neighboring groups.*

1. Introduction

The peer-to-peer (P2P) technology has been one of the most widely used applications today. It provides an environment to make peers have a decentralized control, load-balanced communication and cooperation model. However, due to the nature autonomy and dynamics of peer-to-peer networks, load balance and decentralized control are still the main challenges of P2P research.

The structured system has been proposed as one of the efficient, fast and robust architectures in the current P2P systems, such as Chord [12], CAN [10], Pastry [11] and Tapestry [15]. All these algorithms are distributed hash table (DHT)-based routing algorithms. In the DHT-based system, every peer is given a unique identifier through a well-known consistent hashing function within the same namespace. In each routing hop, the query message was routed to a peer whose identifier was numerically closer to the given identifier. By using these routing algorithms, the routing efficiency will reach to $O(\log n)$. However, in large-scale P2P systems, all peers may spread all over the world. Two peers physically far away to each other may close with each other logically. It will suffer from the latency problem of unbalanced routing in the system.

Recently, some researches have been proposed on the issues of locality in the large-scale P2P systems. One of the well-known methods was the clustering method by clustering peers into groups in terms of physical location and making the groups communicate with each other.

The two-tier architecture was to be in favor. The main idea of the two-tier architecture is to choose one or some powerful peers from each group as super-peer [14]. However, although the mechanism successfully achieves efficient routing performance, it still might cause hop-spot or single-point-of-failure situation easily.

In [14], authors described a method called "super-peer redundancy" to deal with single-point-failure situation. That is to use multiple super-peers in each cluster to reduce the overhead and solve the problem. Although this mechanism overcomes the above two problems, it also introduces storage and synchronization overhead. To avoid the problem, [4] proposed the method which uses a special peer as a gateway for incoming messages and lets each super-peer request only one part of metadata from its normal peers to reduce the storage overhead. However, it may still cause the single-point-of-failure problem when the special gateway-like peer leaves the systems without any notification to the other peers.

In this paper, we proposed an efficient communication method on a super peer network – M-Ring. It efficiently connected with every super peer in the different group without using any extra gateway-like super peers to solve the single-point-of-failure problems.

The remainder of this paper is organized as follow. Section 2 discusses related work. Section 3 introduces the

978-0-7695-3125-0/08 $25.00 © 2008 IEEE
DOI 10.1109/I-SPAN.2008.30

architecture of M-Ring and its operations. The simulation results are presented in Section 4, and we conclude the work in Section 5.

2. Related Work

In [14], the authors chose some powerful peers in the system as the centralized servers used in Napster [8]. Representative applications based on super peer architectures are Gnutella [2] and KaZaA [3]. This mechanism takes advantage of the heterogeneity of capability of peers to achieve large performance improvement of distributed search.

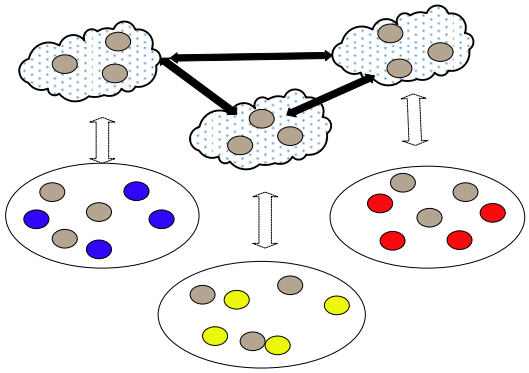
After the research of [14], some researchers focus on the advantage of heterogeneity of peers. In [1], the authors organize peers as disjoint groups and choose some peers as the super peers in each group. Groups use normal DHT-based algorithms to communicate with each other. In the HP2P system [9], these groups were treated as virtual nodes that formed with the Chord ring. Super-peers in each virtual node were served as communicators between two virtual nodes. However, since each virtual node requests different scope of identity space, it will spend a lot of jumps in the global Chord ring to the correct position if a node has to publish a metadata of a file in the network. In [16], when a file published to a super-peer, the super peer broadcasts the new file information to all other super peers in the different groups. Although the method can provide a high performance of routing procedure, it will need a lot of bandwidth when a new file was published.

PASS [4] used a special gateway-like super-peer chosen from the sets of super peers to handle the all incoming messages originated from other groups. Therefore, it has not produced any overhead of the stabilization even if the number in the sets of super peer was large. On the other hand, it could reduce the requirements of storage space since each super peer in the same group had different requesting identity scope. However, the single-point-of-failure may occur at the special gateway-like super peer. In [4], the authors did not give any robust method to handle the leave of super peer. They just used the backup peers to backup the contents of the leaving super peers, but they did not have a mechanism to notice other peers what they have had left.

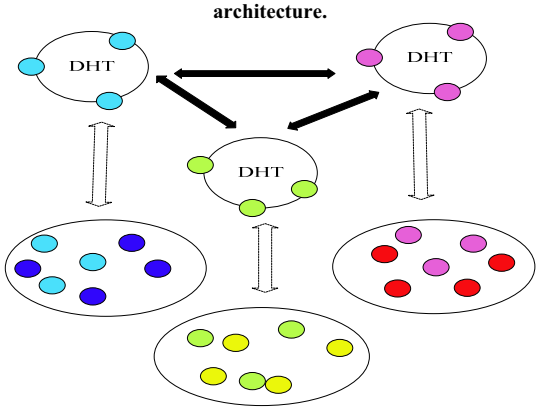
The goal of our research is to let each super peer maintain its link table which links to the neighboring sets of super peers efficiently without any fully connected or using some special gateway-like peers to connect with super peers of other clusters. M-Ring not only reduces the overhead of join/leave procedures of super peers, but avoids the single-point-of-failure problems.

3. M-Ring

In this section, we describe the basic idea and its architecture.



(a) An overview of the traditional two-layered super-peer architecture.



(b) An overview of the M-Ring architecture

Figure 1. The comparison between M-Ring and the traditional super peer architecture.

3.1. The M-Ring Architecture

Figure 1(a) shows a traditional two-tiwe super-peer architecture mechanism. In [14], neighboring clusters have fully connection among the sets. Furthermore, each super peer in the same set has the same information of normal peers. Although this redundancy method has mighty reliability and availability, it also introduces large synchronization and storage overhead. Consequently, in [14], recommended degree of redundancy in each group is 2~3. In [4], the authors separate the indices of the super-peer. It means each super peer has different requesting index region to reduce the storage overhead of super-peers.

In M-Ring, each super-peer from the same cluster is responsible for different index range. Instead of going through the special node to communicate with other clusters, each super peer maintains its link table for storing out-going peers. Besides, all super-peers from the same cluster will not store duplicate link information. Therefore, each super-peer has different forwarding target.

In Figure 1(b), super-peers from the same cluster form a structured ring topology. Every set of super-peers is treated as a Virtual Region (VR). For peers distributing in different clusters in M-Ring, they can communicate with each other through VR's.

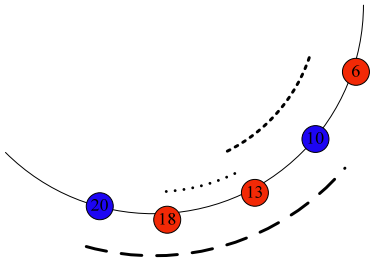


Figure 2. The overlap handle scope situation.

3.2. Overlap Handle Scope

For super-peers self-organization and distributing the communication overhead among all the super-peers, we assume that every super-peer has a unique identifier and all of them share the same name space. Communication between VR's does not go through gateway-like method. We use the concept of the overlap handle scope, a super-peer choose an out-going target in another VR whose identifier is smallest one in the responsible range of origin peer.

In Figure 2, there are two VR's in the system. The peers in red VR handled by Peer20 are peer18 and peer13, peer20 choose a peer13 as out-going target for red VR. Using the above concepts, VR's in the M-Ring system can minimize number of forwarding hops.

3.3. The Link Table of the M-Ring

In order to communicate with other VR's, we construct a link table in each super-peer. Each link table has *k*-1 entries where *k* presents number of VR's. Each entry in stores the information of proper out-going peer in other VR using the mechanism mentioned in section 3.1. When a super-peer wants to communicate with other VR, it can check its link table and find the link id of the desired VR. Then, it can link to the target node by the link table efficiently.

The example in Figure 3(a) shows the link table of peer10. The first link of peer10 was linked to peer6 in the red VR. The reason why we chose peer6 as its red VR communicator was that peer6 had the feature of the overlap handle scope with peer10 during the key 5~6. It also has the smallest identifier in the handle space of peer10 (See Figure 3(b)). But if there does not have any red peer in its handle space, like peer4, it will choose the first node, peer6, next to peer4 as its communicator which also has a handle scope overlap feature.

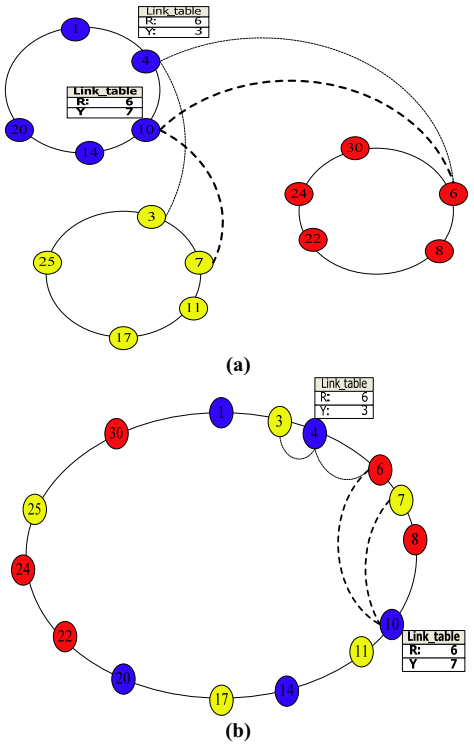


Figure 3. Link table of peer10 and peer4 in M-Ring.

In order to minimize the number of hops in each local VR, we also maintain one more column which stores the information about the next peer of its target node of link of the same ring.

3.4. Super Peer Selection

Recently, some researches have been proposed how to select a peer as a super peer [6] [7]. In the M-Ring system, we classify peers according to their abilities. A powerful peer which has larger bandwidth, more storage space or faster CPU speed will be chosen as a super peer. Oppositely, the weaker peer who will not be selected is a normal peer.

Another, in our M-Ring system, each normal peer sends its physical information to its parent which is super peer. All the super peers in the M-Ring system will select a powerful normal peer as a new super peer according to the physical location.

3.5. Node Operations

When a normal peer joins the M-Ring, it contacts with any existing peer to connect with the super peer SP in the VR. Then, the new peer sends a join message and its metadata information to one of the SP's.

When a peer *n* has been chosen as a super peer in its cluster, it must send a join message to a existing super peer *p* to initial join process. If the topology was formed

with the Chord topology, it will perform the join algorithm of the Chord.

Consequently, the peer *n* build up its own link table for each VR other with previously mentioned in section 3.1. Figure 4 describes the join procedure of the super peer in pseudo code.

It is also important to keep the link table work well. If a new super-peer joins or leaves VR without any notification, the performance of outer hops will be encumbered. Super-peers in the system keep running the update procedure periodically to make sure the efficiency. The update procedure of the link table is similar to the join procedure. Each super-peer sends the FindSuccessor (predecessorID) messages to other VR according to its link table and checks if needed to update the entry or not. Figure 5 shows the pseudo codes of the update procedure the link table.

In the M-Ring system, a peer may leave the system dynamically or fail voluntarily. In order to cope with this situation, every peer will ask its previous node to obtain its link entry to the yellow ring and run the same link table update procedure to renew the link table entry.

On the other hand, since each super-peer in the system owns parts of the metadata information, if a super-peer suddenly leaves the VR without any notification, VR will take a lot of extra efforts to recover the lost metadata information. In order to cope with the problem, we propose the replication method. When a super peer joins one of the VR, it will not only store the metadata information which it has charged, but also store its next peers' metadata. For example, when a super peer *l* leave the VR, the front peer *f* will send *l*'s charging metadata to *l*'s back peer *b* to complete *b*'s charging metadata set.

```
Procedure SuperPeer.join(){
    localVR.join(bootstrapPeer);
    LinkTable link_table = getLinkTable(previousPeer);
    for(int updateVR=0; updateVR<link_table.size(); updateVR++){
        tempEntryInfor =
          link_table[updateVR].FindSuccessor(previousPeer.GetId());
        if(tmpEntryInfo != link_table[updateVR])
            link_table[updateVR] = tmpEntryInfo;
    }
}
```

Figure 4. The pseudo codes of the join procedure of the super peer.

```
Procedure SuperPeer.linktableUpdate(updateEntry){
    tmpEntryInfo =
        link_table[updateEntry].FindSuccessor(previousPeer.GetId());
    if(tmpEntryInfo != link_table[updateEntry])
        link_table[updateEntry] = tmpEntryInfo;
}
```

Figure 5. The pseudo codes of the update procedure of the link table.

4. Simulation

First, we describe our simulation results about the number of average hops when messages are routed to the outside VR set and show the speedup results by adding the second column in the link table. Second, in order to perceive the overhead when super peers join/leave, we give a comparison with the number of message when a join/leave procedure executed on a super peer. Lastly, we compare the cost of the total bandwidth among the M-Ring and other methods.

Number of clusters	4
users per group	1000
total metadata number on VR	1000
query rate	0.01 per user per sec
Update rate	0.02 per user per sec
join/leave message size	80 bytes
Query message size	80 bytes
Update message size	120 bytes
metadata size	120 bytes

Table 1. Configuration parameters and default value

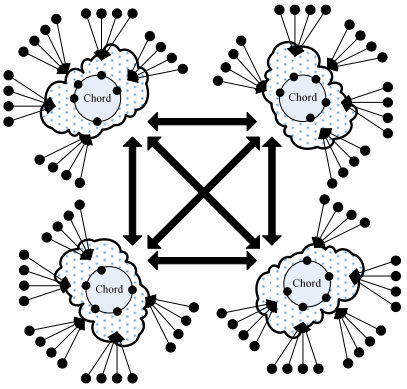


Figure 7. A 4-clustering super peer topology.

4.1. Environment

We divide the environment of the M-Ring architecture into four clusters, like Figure 7. That is there are four super peer sets (VR) in the environment. Each VR uses the Chord ring as its local topology. In order to simulate the performance of the M-Ring architecture, we use similar parameters described in [14]. Every super peer has three basic operations which are query, update, and join/leave, separately. The cost of query message and join/leave is 80 bytes, and the cost of update message is about 120 byte since it includes the file metadata information. The query and update rate is 0.01 and 0.02 per user per second, respectively. In [5], they found that the lifetime of super peer in KaZaA is 149 minutes. As a result, we will compute the total bandwidth during the time interval and observe that the number of message

routing during this period. The detail parameters are listed in Table 1.

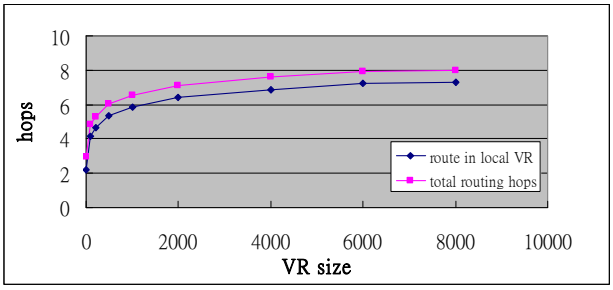


Figure 8. Average hops of the two VRs.

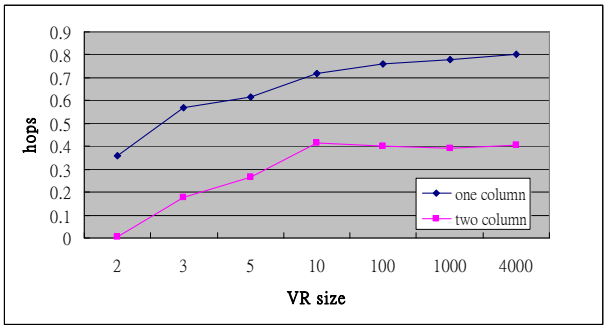


Figure 9. The speedup of the routing procedure.

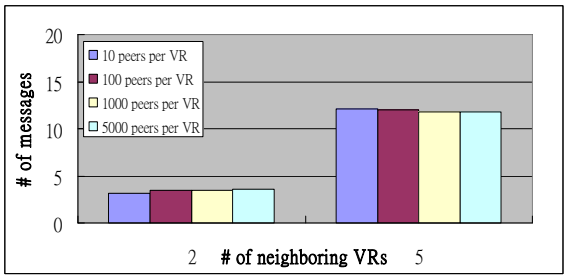


Figure 10. The VR's size affects the message number when the join procedure executes.

4.2. The Number of Average Hops in Each Outer VR

Figure 8 shows the number of routing hops between two neighboring clusters. It is obviously to notice that the routing hops in the first local VR is base on its topology. Since the topology of the VR is constructed by the Chord ring in this simulation, the growing line is the same as that of the Chord ring. We can also notice that when the query messages are delivered to outer VR, the average number of hops of each outer VR is less than 1. In Figure 9, if we use only the information of one column in the link table, the average outer hops is around 0.8 hops. If we use the information of two columns in the link table, it can reduce half of hops in each outer VR. This simulation results show that M-Ring has a good routing behavior and does not waste any query hops in each outer VR.

4.3. The Number of Messages When Join Procedure Executed

We demonstrate that when a super peer joins the M-Ring, how many messages will have to be delivered in order to stabilize our system. Figure 10 show the size of the VR will not affect the number of messages but the number of neighbors of the VR will. In Figure 11, we give a comparison between the M-Ring and normal overlay which has fully-connected super peers. We can find out the number of messages in the M-Ring is much smaller than that of the normal overlay. Since the normal overlay has to broadcast the new joining peer's information to all super peers of its local group and other super peers in neighboring groups to ensure that the system works well, it takes a lot of messages to make sure all the super peers have the information of the new node. In the M-Ring system, because each node only needs to maintain the information of one (or two) node in its neighboring VR, when a new node comes, the new attendant just needs to construct its link table entry. Moreover, the super peer in its neighboring VR also needs to inspect if the new peer has to replace there link table. In the join procedure of the M-Ring architecture, there are only some super peers who have to deliver the messages. But in the normal overlay, all super peers in the system have to be notified about the information of the new peer's join. For this reason, the message number of the M-Ring is much smaller than that of the normal overlay.

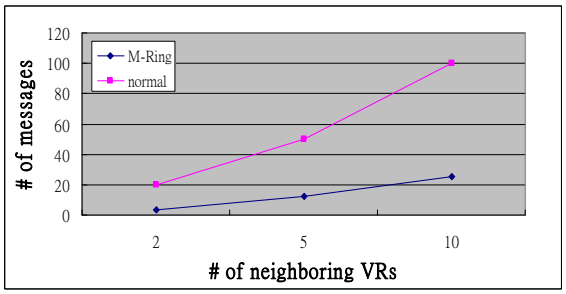


Figure 11. The comparison between two super peer overlays when join procedure executes.

4.4. The Total Bandwidth Cost of the M-Ring

We compute the aggregate bandwidth in 149 minutes. There are three basic operations of each overlay of super peer that are query, update, and join/leave. On the average, if the size of network is stable, the number of the leave of peers is the same as the one of the join. We assume that our system is stabilized. In Figure 12, we

observe that the M-Ring has better cost of bandwidth if each set of super peer has more than 5 super peers. This is because in the normal overlay of super peer, when a node executes an update procedure, its parent needs to send this update information to all the other super peers in the same group. In other words, the bigger size of the group of super peer, the more aggregated bandwidth will be required. On the contrary, in the M-Ring system, the update messages only need to be delivered to the super peer who takes charge of the update files. On the other hand, when a super peer joins/leaves the M-Ring system, it only requires a little cost of bandwidth because each node in the M-Ring stores only parts of the metadata information.

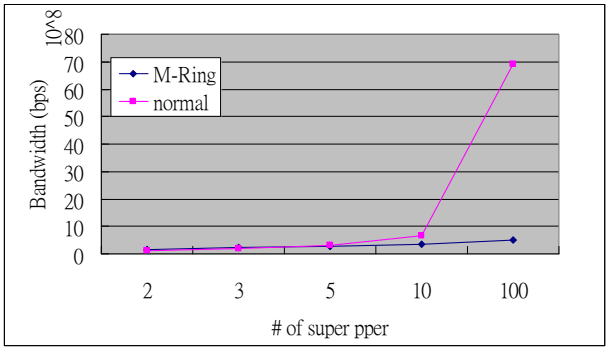


Figure 12. A comparison of the total bandwidth cost between two VR's.

5. Conclusion

This paper presents a communication method among sets of super peer – M-Ring. Also, we propose the feature when super peers are in the same identity space and use this feature to minimize the outer hops in each neighboring sets of super peer. The M-Ring architecture has no fully-connected feature or uses any special gateway-like peers but it makes each super peer construct its own link table. Our simulation results show that M-Ring is efficient to stabilize the outer connections. Our method achieves the improvement of decreasing the storage space and bandwidth cost at each super peer.

6. References

[1] Luis Garcés-Erice, Ernst W. Biersack, Pascal Felber, Keith W. Ross, Guillaume Urvoy-Keller, "Hierarchical Peer-to-Peer Systems", in *Parallel Processing Letters,* Vol 13, No 4, pp. 643-657, December 2003.

[2] Gnutella website. http://www.gnutella.com.

[3] KaZaA website. http://www.kazaa.com.

[4] Gisik Kwon, Kyung D. Ryu, "An Efficient Peer-to-Peer File Sharing Exploiting Hierarchy and Asymmetry", *Proceedings of the 2003 Symposium on Applications and the Internet(SAINT'03),* pp. 226–233, 2003.

[5] Jian Liang, Rakesh Kumar , Keith W. Ross, "The KaZaA Overlay: A Measurement Study", *Computer Networks Journal (Elsevier),* 2005.

[6] Virginia Lo, Dayi Zhou, Yuhong Liu, Chris GauthierDickey, and Jun Li, "Scalable Supernode Selection in Peer-to-Peer Overlay Networks", *Proceedings of the 2005 Second International Workshop on Hot Topics in Peer-to- Peer Systems (HOT-P2P'05),* OR, USA, July, 2005.

[7] Su-Hong Min, Joanne Holliday, Dong-Sub Cho, "Optimal Super-peer Selection for Large-scale P2P System", *2006 International Conference on Hybrid Information Technology (ICHIT'06).*

[8] Napster website. http://www.napster.com.

[9] Zhuo Peng, Zhenhua Duan, Jian-Jun Qi, Yang Cao and Ertao Lv, "HP2P: A Hybrid Hierarchical P2P Network", *Proceedings of the First International Conference on Digital Society(ICDS'07).*

[10] Sylvia Ratnasamy, Paul Francis, Mark Handley, Richard Karp, and Scott Shenker, "A scalable content-addressable network", *Proceedings of SIGCOMM 2001*, Aug. 2001.

[11] Antony Rowstron and Peter Druschel, "Pastry: Scalable, distributed object location and routing for large-scale peer-to-peer systems", in *IFIP/ACM International Conference on Distributed Systems Platforms (Middleware)*, (Heidelberg, Germany), pp. 329–350, November 2001.

[12] Ion Stoica, Robert Morris, David Liben-Nowell, David R. Karger, M. Frans Kaashoek, Frank Dabek, and Hari Balakrishnan, "Chord: A Scalable Peer-to-Peer Lookup Protocol for Internet Applications", *IEEE/ACM Transactions on Networking,* VOL. 11, NO. 1, FEBRUARY 2003.

[13] Zhiyong Xu, Rui Min and Yiming Hu, "HIERAS: A DHT Based Hierarchical P2P Routing Algorithm", *Proceedings of the 2003 International Conference on Parallel Processing (ICPP'03).*

[14] Beverly Yang, Hector Garcia-Molina, "Designing a Super-Peer Network", *Proceedings of the 19th International Conference on Data Engineering(ICDE'03).*

[15] Ben Y. Zhao, John Kubiatowicz, and Anthony D. Joseph, "Tapestry: An infrastructure for fault-tolerant wide-area location and routing", *Tech. Rep. UCB/CSD-01-1141,* Computer Science Division, University of California, Berkeley, Apr 2001.

[16] Yingwu Zhu, Honghao Wang Yiming Hu, "A Super-Peer Based Lookup in Structured Peer-to-Peer Systems", in *Proceedings of the 16th International Conference on Parallel and Distributed Computing Systems(PDCS'03),* Nevada, USA, August 2003.