

The Use of Super Node to Process Query in Peer-to-Peer Networks

H. Saberi Najafi¹, J. Pourqasem², S.A. Edalatpanah^{1,*}

¹Department of Applied Mathematics, Faculty of Mathematical Sciences, University of Guilan, P.O. Box 41335-1914 Rasht, Iran

²Department of Computer Science, Faculty of Engineering, University of Guilan, P.O. Box 3756, Rasht, Iran

*Corresponding author: saedalatpanah@gmail.com

Received June 22, 2015; Revised July 15, 2015; Accepted July 20, 2015

Abstract Equal peers in peer-to-peer (P2P) networks may be drawbacks in term of bandwidth, scalability and latency of system. In this paper we use the super-peer model in order to query process in P2P networks. Each super-peer in this presentation utilizes the Hop-Count Routing Indices (HRI) technology to select the best neighbor super-peer. The latency of query process and the scalability of system is improved by using the HRI based search model. We evaluate the performance of this work by means of simulation based on GridSim tools which shows the efficiency of our approach compare to flooding method.

Keywords: peer-to-peer, super-peer, query process, structured and unstructured, routing indices

Cite This Article: H. Saberi Najafi, J. Pourqasem, and S.A. Edalatpanah, "The Use of Super Node to Process Query in Peer-to-Peer Networks." *Digital Technologies*, vol. 1, no. 1 (2015): 28-32. doi: 10.12691/dt-1-1-6.

1. Introduction

The P2P systems are based on the principle that every component of the system has the same responsibilities [20]. In this systems, each peer acts as a client and a server. The typical participants in P2P systems are common desktop which download music or video files over Internet TCP connections. Peers are highly dynamic as users can enter, depart, and rejoin the system unpredictably [21].

There are two main categories based on the structure in P2P systems: Structured and Unstructured [10,22,23]. The structured P2P systems use a rigid structure to interconnect peers. In structured systems, files and peers are indexes. The unstructured systems randomly connect peers to a static number of other peers and there is no information about the location of files.

Exploiting the heterogeneity of peers in network has proven to have great potential [11]. These systems assign additional responsibilities to high-capacity nodes, called super-peers [3]. A super-peer in super-peer network acts as a server to client peers. There is a two-level structure to organize peers [24,25]. In low level clients are connected to a single super-peer (in some case there is redundant super-peers [2]). In high level, super-peers are connected to each other to form an overlay network. Typically a super-peer and its clients are called cluster. The size of cluster is the number of nodes in the cluster, including the super-peer itself. Client peers only join to cluster's super-peer and rely on it. Each super-peer indexes the resource information of belonging nodes in cluster. When a client peer looks up a file, submits query to its super-peer and receives results from it, thus super-peers answer the queries on behalf of the client peers.

In this paper we present a query process based on super-peer networks. Each super-peer in this work maintains resource information of other neighbor super-peers by means of Routing Indices (RI) [6,17] technology. In this work the super-peers use a version of Routing Indices (RI), called Hop-Count Routing Index (HRI) and forward query to best neighbor super-peers based on HRI information. This way the performance of query process improved by avoid flooding the network.

The remainder of this paper is organized as follows. Section 2 discusses related work. In Section 3, we present a super-peer network, and Section 4 describes the Routing Indices (RI) technology. In Section 5 the query process in super-peer network is detailed. Section 6 provides the results of evaluation, and finally, Section 7 contains conclusions and future work.

2. Related Work

Peer to peer systems was introduced in 1999 by Napster first. It comprised a central server which stored the index of all files shared by the participating peers. To locate a file, a user queried the central server using the name of the file and received as a result the IP address of a peer owning the file. Then a direct connection was established between the requesting node and the node containing the file in order to download. The central index server of Napster is not scalable and there is a single point of failure and bottleneck in large-scale and dynamic environments. While Napster is historically considered as the first unstructured P2P system, the existence of a central server differentiates it significantly from today's unstructured P2P systems.

Gnutella is one of the most popular unstructured P2P systems that emerged after Napster. The search for file is fulfilled via the basic flooding strategy, where a query is propagated to all neighbors within a certain number of hops. This searching method was never stable, due to peers frequently disconnect. When the network grows large enough, it gets saturate and often cause huge delays. As a result, queries often drop and searches make unsatisfactory result as only minor portion of peers are searched [7]. Structured P2P [12] uses distributed indexing service such as Distributed Hash Table (DHT). Peers and files are typically mapped through the same hash function to a key space. The indices of peers and files are organized in a rigid structure according to their keys, which facilitates the search of files. Chord [8] is the first structured P2P system in which both peers and files are mapped through the same hash function to an m-bit key space. The peers in Chord are organized in a one-dimensional circle according to their keys. Each peer keeps the index of all files whose keys fall in the range between the key of its predecessor and its own key. The structured approaches generally speed up discovery operations and are scalable, however, Chord cannot support range queries and multi-attribute-based lookups and suffers from search latency.

The super-peer models [13,14,16] were proposed to achieve a balance between the inherent efficiency of centralized approaches, and the autonomy, load-balancing and fault-tolerant features offered by decentralized search. KaZaA is a P2P system which was introduced in March 2001 and adopted the super-peer model in its design. KaZaA was used for file sharing programs without any

server. There are two types of nodes in KaZaA, super nodes and leaf nodes. Each super node carries a set of leaf nodes and connects to other super nodes to form an overlay network. Super nodes maintain the resource information of leaf nodes. When a node looks up a resource, it sends a query to its super node, which initiates a search process in the overlay network [4]. Gnutella2 is another super-peer network which has two types of nodes: leaves (normal peer) and hubs (cluster-heads or super-peers). Each leaf node, has one or two connections to hubs. Cluster-heads index resources information of peers by means of a Query Routing Table and exchange the indices among the connected hubs. During the search a leaf node (peer) sends a request to a hub. If it answers, the peer downloads the requesting item directly from the peer that hosts the item. If not, the query is forwarded by the current hub to another hub which its address is taken from the routing tables of the super peers. The search ends when either the item is found or all known hubs are searched or a predefined search limit is reached [9].

3. The Super-peer Network

In a super-peer network, nodes are divided two category, super-peers and client peers [15]. Each super-peer is connected to a set of clients. Clients are joined to a single super-peer only (or in some case redundant super-peers [2]). A super-peer acts as a centralized server for a number of regular peers in each cluster, while at the same time the super-peers connect to each other to form an overlay network at a higher level.

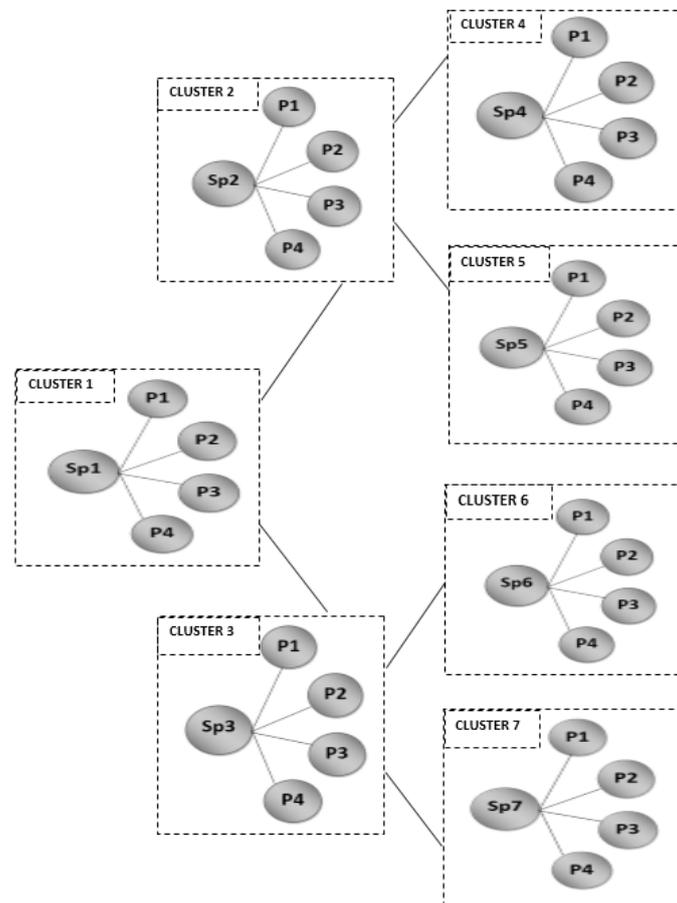


Figure 1. A super-peer network

Figure 1 illustrates a super-peer network in which each cluster contains the super-peer and its client peers. The cluster size is the number of nodes in the cluster (client peers and super-peer). Clusters can communicate with other in overlay network through super-peers based on P2P connections. Once a client joins the system it will send its metadata of resources to cluster's super-peer and the related super-peer will add this metadata to its indexes. When a node leaves the network, the related super-peer will remove its metadata from the index. In information update (e.g., insertion, deletion or modification of an item), clients will send the update to the super-peer as well. Furthermore, the super nodes act to search and query routing operations which are presented in the Section 5.

4. Routing Indices

The Routing Index (RI) [6] is used to improve the query routing in P2P networks [17,18,19], and to avoid flooding. The RI organizes the resource information of neighbor nodes and query is forwarded to neighbor peers based on these information. In this paper each super-peer utilizes a version of RI is called Hop-Count Routing Index (HRI) [6], which considers the number of hops needed to reach a datum. As is showed in Table 1, the HRI is represented as an $M \times N$ table, where M is the number of neighbor super-peers and N is the horizon (maximum number of hops):

Table 1. Example of HRI

	1 HOP			2 HOP		
File	Music	Document	Picture	Music	Document	Picture
Type	Rock	Word	Landscape	Rock	Word	Landscape
SP1	15	25	45	27	85	50
SP2	65	8	28	36	54	13

5.1. The Search Algorithm

Considering the Figure 1 and Algorithm 1, when node P1 in cluster 1 needs a resource (file), sends request to its cluster's super-peer (cluster 1). Then, super-peer (SP₁) searches the cluster for that request. If matching resources found, the address of the node which is owner of requested resources is send back to requested node. If not, the super-peer of cluster 1 selects the best neighbor super-peers based on HRI information and forwards query to it. We assume the cluster 2 is better than cluster 3. Consequently, the super-peer of cluster 2 searches locally and if there is no result, then forwards query to first best neighbor (cluster 4). If there is result in cluster 4, the cluster's super-peer returns a queryHit containing the address of the node that is owner of requested resource. Otherwise, the query is bounced back and the super-peer of cluster 2 forwards query to the second best neighbor (cluster 6). Generally, if the super-peer of cluster 2 does not have suitable resource and all neighbors (except cluster 1) is checked and the request is bounced back, the super-peer of cluster 2 will bounce back the query to cluster 1. In turn the super-peer of cluster 1 will forward the query to the second best neighbor (cluster 3).

the n th position in the m th row is the number of data elements that can be reached going through neighbor super-peer m , within n hops. When a new super-peer joins the network, it sends to all neighbor super-peers the information about the data which it controls. Also each super-peer forwards the update of data that is changed in belonging nodes to neighbor super-peers.

5. The Process of Client Query

When a client needs a resource (file), generates query message and sends it to the local super-peer. The super-peer processes the query locally in the cluster and looks up results. If it find the query result in belonging nodes, sends to the requesting node a queryHit containing the IDs or address of each client whose resource collection has produced a result. If not, the local super-peer selects best neighbor super-peer based on Hop-Count Routing Index (HRI) with respect to requested requirements. Then super-peer forwards a copy of the query to a selected neighbor super-peers. Whenever a resource that matches the specified criteria in the query is found, a queryHit is generated and is returned along the same path back to the querying node. Otherwise the selection process is repeated and query is forwarded by neighbor super-peers. This method is continued until to reach result or all neighbor super-peers is searched.

Algorithm 1

```

1:  $q$ : new incoming query
2: LocalResource: a resource in the cluster
3: NextBestNeighbor: a neighbor super-peer is selected
4: ToTry: the next neighbor super-peer
5: for incoming  $q$  do
6: LocalResource: = MatchQueryLocalResource ( $q$ )
7: if (LocalResource == null) then
8: NextBestNeighbor: = HRI ( $q$ , ToTry)
9: if (NextBestNeighbor == null) then
10: Recipient: = Sender ( $q$ )
11: else
12: Recipient: = NextBestNeighbor
13: end if
14: ForwardQueryToRecipient ( $q$ , Recipient)
15: else
16: SendResponseToRequester ( $q$ )
17: end if
18: end for

```

6. Experimental Results

This Section evaluates the performance of our new discovery technique with regard to the query response

time to obtain the result. Our simulation environment has conducted by GridSim tools [26], and consists of 100 of users, 100Mbs of network bandwidth, 10-second propagation delay, and 1500 packet per second of maximum transmission unit. The routers of network use the Routing Information Protocol (RIP) protocol, and their scheduling method is First Input First Output (FIFO). We

have performed the simulation at two Scenario. First, we consider neighbor super-peers of cluster1's super-peer at 1- hop count (SP_1), and compare the query process response time by using the flooding algorithm and HRI based search. In second Scenario, we have evaluated the results of query process by SP_1 at 2-hop count by using the flooding and HRI approaches.

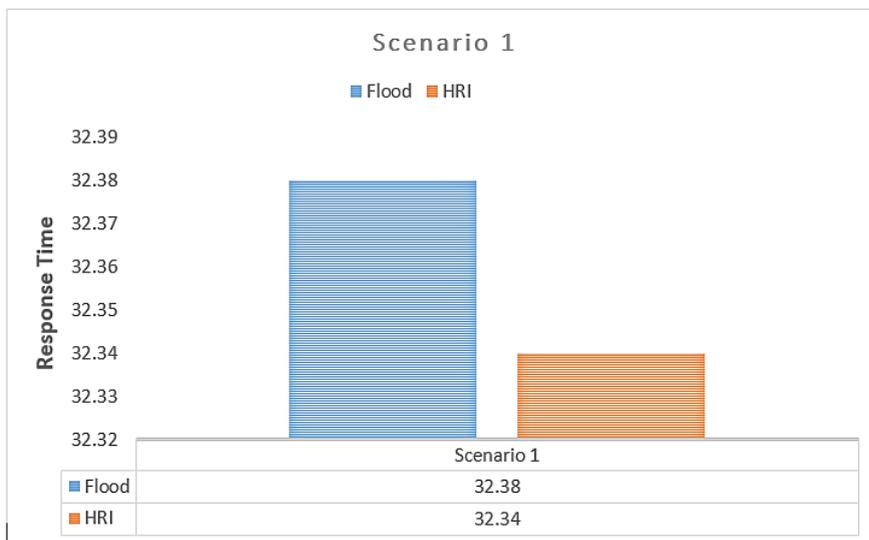


Figure 2. Scenario 1

Figure 2 illustrates the query response time is decreased by using the HRI table. The improvement of query response time in this Scenario in about 4%. Consequently, the super-peer networks that use the HRI tables can reduce the latency of query process. Results of second Scenario are shown in Figure 3. In this Scenario, the query result is obtained at 2- hop count. It can be seen, the query

response time is increased when the number of hop-count is added. Using the HRI table, we can improve the latency of query process by 12% in second Scenario. Therefore, the performance of query process is enhanced and the scalability of system is increased by avoiding of the high traffic load caused by flooding the network.

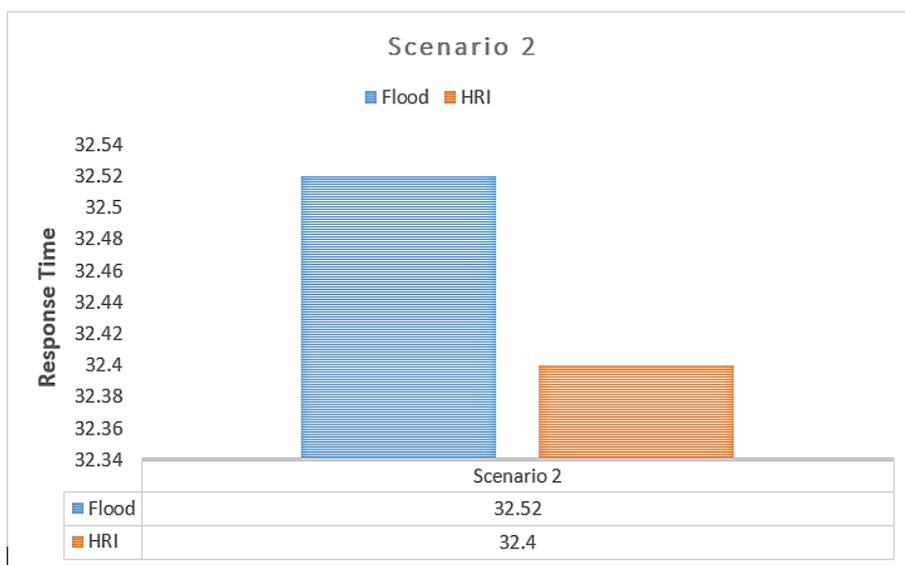


Figure 3. Scenario 2

7. Conclusions and Future Work

Often, in p2p networks, nodes are equal and have same responsibility that will be the drawback of system performance. In this paper we presented a super-peer

network and used super peers in order to process the request from client peers. Each super-peer employed a Hop-Count Routing Index (HRI) and forwarded query to best neighbor super-peer. We provided the search algorithm in this work to describe the request process functionality. Furthermore, the evaluation of our presented approach was conducted by means of GridSim tools. The

results of evaluation showed the response time of query process and the system latency was improved in presented search model. Consequently, the HRI based search method made the P2P networks more scalable than the flooding strategy.

As a guideline for future work, we will use the summarization technique in order to reduce the size of information that is exchanged between neighbor super-peers. The summarization techniques will decrease the overhead of data updating and improve the network bandwidth in dynamic environments.

References

- [1] Crespo, A., & Garcia-Molina, H. (2002). Routing indices for peer-to-peer systems. In *Distributed Computing Systems, 2002. Proceedings. 22nd International Conference on* (pp. 23-32). IEEE.
- [2] Yang, B., & Garcia-Molina, H. (2003, March). Designing a super-peer network. In *Data Engineering, 2003. Proceedings. 19th International Conference on* (pp. 49-60). IEEE.
- [3] Cao, Z., Li, K., & Liu, Y. (2008, January). A multi-level Super peer based P2P architecture. In *Information Networking, 2008. ICOIN 2008. International Conference on* (pp. 1-5). IEEE.
- [4] Chen, S., Zhang, Z., Chen, S., & Shi, B. (2008). Efficient file search in non-DHT P2P networks. *Computer Communications*, 31(2), 304-317.
- [5] Oppenheimer, D., Albrecht, J., Patterson, D., & Vahdat, A. (2005, July). Design and implementation tradeoffs for wide-area resource discovery. In *High Performance Distributed Computing, 2005. HPDC-14. Proceedings. 14th IEEE International Symposium on* (pp. 113-124). IEEE.
- [6] Crespo, A., & Garcia-Molina, H. (2002). Routing indices for peer-to-peer systems. In *Distributed Computing Systems, 2002. Proceedings. 22nd International Conference on* (pp. 23-32). IEEE.
- [7] Meshkova, E., Riihijärvi, J., Petrova, M., & Mähönen, P. (2008). A survey on resource discovery mechanisms, peer-to-peer and service discovery frameworks. *Computer networks*, 52(11), 2097-2128.
- [8] Stoica, I., Morris, R., Karger, D., Kaashoek, M. F., & Balakrishnan, H. (2001). Chord: A scalable peer-to-peer lookup service for internet applications. *ACM SIGCOMM Computer Communication Review*, 31(4), 149-160.
- [9] Sharifkhani, F., & Pakravan, M. R. (2013, August). A review of new advances in resource discovery approaches in unstructured P2P networks. In *Advances in Computing, Communications and Informatics (ICACCI), 2013 International Conference on* (pp. 828-833). IEEE.
- [10] Trunfio, P., Talia, D., Papadakis, H., Fragopoulou, P., Mordacchini, M., Penmanen, M., & Haridi, S. (2007). Peer-to-Peer resource discovery in Grids: Models and systems. *Future Generation Computer Systems*, 23(7), 864-878.
- [11] Cholvi, V., Felber, P., & Biersack, E. (2004). Efficient search in unstructured peer-to-peer networks. *European transactions on telecommunications*, 15(6), 535-548.
- [12] Gautam, S., & Cheng, X. (2014). Comparative Experiments on Resource Discovery in P2P Networks. *Journal of Next Generation Information Technology*, 5(1), 89.
- [13] Liu, M., Harjula, E., & Ylianttila, M. (2013). An efficient selection algorithm for building a super-peer overlay. *Journal of Internet Services and Applications*, 4(1), 1-12.
- [14] Mastroianni, C., Talia, D., & Verta, O. (2005). A super-peer model for building resource discovery services in grids: Design and simulation analysis. In *Advances in Grid Computing-EGC 2005* (pp. 132-143). Springer Berlin Heidelberg.
- [15] Awan, A., Ferreira, R. A., Jagannathan, S., & Grama, A. (2006). Unstructured peer-to-peer networks for sharing processor cycles. *Parallel Computing*, 32(2), 115-135.
- [16] Tan, Y. H., Lü, K., & Lin, Y. P. (2012). Organisation and management of shared documents in super-peer networks based semantic hierarchical cluster trees. *Peer-to-Peer Networking and Applications*, 5(3), 292-308.
- [17] Caminero, A. C., Robles-Gómez, A., Ros, S., Hernández, R., & Tobarra, L. (2013). P2P-based resource discovery in dynamic grids allowing multi-attribute and range queries. *Parallel Computing*, 39(10), 615-637.
- [18] Marzolla, M., Mordacchini, M., & Orlando, S. (2007). Peer-to-peer systems for discovering resources in a dynamic grid. *Parallel Computing*, 33(4), 339-358.
- [19] Puppini, D., Moncelli, S., Baraglia, R., Tonello, N., & Silvestri, F. (2005). A grid information service based on peer-to-peer. In *Euro-Par 2005 Parallel Processing* (pp. 454-464). Springer Berlin Heidelberg.
- [20] Mousavi Khaneghah, E., Mirtaheri, S. L., Sharifi, M., & Minaei Bidgoli, B. (2014). Modeling and analysis of access transparency and scalability in P2P distributed systems. *International Journal of Communication Systems*, 27(10), 2190-2214.
- [21] Navimipour, N. J., & Milani, F. S. (2014). A comprehensive study of the resource discovery techniques in Peer-to-Peer networks. *Peer-to-Peer Networking and Applications*, 8(3), 474-492.
- [22] Gaur, H., Gaur, P., & Choudhary, V. (2015). Different search Algorithms in Unstructured P2P Systems. *International Journal of Engineering Technology and Management (IJETM)*, 2(3), 8-13.
- [23] Navimipour, N. J., Rahmani, A. M., Navin, A. H., & Hosseinzadeh, M. (2014). Resource discovery mechanisms in grid systems: A survey. *Journal of Network and Computer Applications*, 41, 389-410.
- [24] Teng, H. Y., Lin, C. N., & Hwang, R. H. (2014). A self-similar super-peer overlay construction scheme for super large-scale P2P applications. *Information Systems Frontiers*, 16(1), 45-58.
- [25] Kurve, A., Griffin, C., Miller, D. J., & Kesidis, G. (2015). Optimizing cluster formation in super-peer networks via local incentive design. *Peer-to-Peer Networking and Applications*, 8(1), 1-21.
- [26] Sulistio, A., Cibej, U., Venugopal, S., Robic, B., & Buyya, R. (2008). A toolkit for modelling and simulating data Grids: an extension to GridSim. *Concurrency and Computation: Practice and Experience*, 20(13), 1591-1609.