How Caching Queries at Client-peers Affects the Loads of Super-peer P2P Systems

Rozlina Mohamed[1, 2], Christopher D. Buckingham[1]

[1] *School of Engineering & Applied Science, Aston University, Birmingham B4 7ET, United Kingdom*

{mohamedr, c.d.buckingham} @aston.ac.uk

[2] *Faculty of Computer System & Software Engineering, University Malaysia Pahang, Kuantan 25000, Malaysia*

Abstract

Super-peer P2P systems strike a balance between searching efficiency in centralized P2P systems and the autonomy, load balancing and robustness provided by pure P2P systems. Super-peer is a node in the super-peer P2P system that maintains the central index for the information shared by a set of peers within the same cluster. The central index handles the searching request on behalf of the connecting set of peers and also passes on the request to neighboring super-peers in order to access additional indices and peers. In this paper, we study the behavior of query answering in super-peer P2P systems with the aim of understanding the issues and tradeoffs in designing a scalable super-peer system. We focus on where to post queries in order to retrieve the result and investigate the implications for four different architectures: caching queries and caching query results at the super-peer; caching the data location of previous queries; and an ordinary P2P system without any caching facilities. The paper discusses the tradeoffs parameters between architectures with respect to caching, highlights the effect of key parameter on system performance.

Keywords: Peer-to-peer, super-peer network, query routing, query answering.

1. Introduction

Peer-to-peer (P2P) systems are distributed system without centralized control, where each peer is an autonomous node. P2P system is popular as a file-sharing application. Kazaa [1] and YouTube [2] are among the most famous P2P systems for sharing huge amount of files. Another application of P2P system, called schema-based has recently drawn considerable attention. Instead of sharing the files between peers, each peer in schema-based P2P is a database management system itself. Hyper[3], Edutella[4], Piazza[5], Auto-Med[6] and Bibster [7] are the example of peer database management research projects. In contrast to the federated database systems, each peer in schema-based P2P is free to declare its own schema.

In a broad sense, there are three major types of P2P system topology: pure, centralized and super-peer. This classification is reflects the varying degrees of decentralization in processing tasks and sharing resources among peers in the network.[8]. Amongst these topologies, super-peer network strikes a balance between inherent result searching efficiency in centralized network; and the autonomy, load balancing and robustness provided by pure P2P network. Super-peer is a node in the P2P system that maintains the central index for the information shared by a set of peers within the same cluster (we denoted as *client-peer*) and other interconnected super-peers. The index of shared resource is a collection of metadata. Metadata is the structured data that describes the characteristics of resources. This index is used as a routing index directory for assisting query routing. Query routing is technology which uses compressed index from multiple peers to 'route' queries to where they may be fulfilled.

In addition to maintaining the routing index, super-peer is responsible for providing routing information for queries that has been posted by its client-peers [4]. Besides that, super-peer is also responsible to retain the registry of respective client-peers, such as peer join or withdrawal from the network and peer updates their shared information. Due to the abovementioned responsibilities, super-peer is the most important node within its cluster. Meanwhile, super-peer is also considered as an autonomous peer node. As an autonomous peer in a P2P system, super-peer node is free to join and leave the network. Thus, super-peer would become a single-point-of-failure when it suddenly withdrawal from the system. Therefore, a fault tolerance mechanism is highly required to overcome the problem that related to the single-point-of failure [8-10].

Our research goal is to study the scalability of query answering process in the schema-based P2P systems

978-1-4244-2020-9/08/$25.00 ©2008 IEEE

based on super-peer network. Later, we referred it as super-peer P2P systems. Directing to our goal, in this paper we are comparing the tradeoffs parameters for the cost of query answering while considering query caching as an alternative approach to the query routing in P2P system based on standard super-peer network. Thus, we are comparing the query routing in four different super-peer system architectures in the next section. Section 3 is the comparative discussions for parameters that associate with the cost of query answering in each P2P architecture. Conclusion of this paper is in Section 4.

2. Super-peer Network Architectures

In this section, we are comparing the query routing process in four different architectures in the super-peer P2P systems. These architectures are classified as: i) standard super-peer systems, ii) query caching (also known as materialized views) at super-peers, iii) query result caching at super-peer, and iv) caching the actual data source location at client-peers. We begin by describing the basic concepts of query routing in a standard super-peer system.

2.1. Standard super-peer systems

In this paper, a standard super-peer systems is referred to P2P system based on super-peer network topology such as Hyper project as described in [3, 11]. Our terms super-peer and client-peer are referred to 'hyperpeer' and 'peer' in Hyper. In our research context, we are considering that the super-peer is responsible to maintain the routing index. Thus the query routing messages in the network is based on the super-peers' indexes. The routing index is an index of shared resources located at peers in the network. This index keeps either the actual resource location or direction resource location [12]. Most of the data sources in schema-based P2P systems are provided in XML (Extensible Markup Language) [5, 6, 13, 14] and RDF (Resource Description Framework) [15, 16] format. In our research we are considering XML as a shared data format while XQuery as the query language. XQuery is considered because of the recommendation by W3C for querying the XML data [17].

Searching for query result location in schema-based P2P system is seen as identifying the keyword from the query message, and matched this keyword with the existing metadata of schema provided in the routing indexes. Thus, the query message is forward by the querying peer towards resource location that has been identified from the routing indexes. There are several approaches for constructing routing index. For instance, XQP and XPeer [14, 18] have constructed the routing index as a single XML tree, where peer shared schema is used to construct the tree. Meanwhile, PGrid, CAN and Chord [19] routing based on overlays which is a distributed hash-tables (DHT), while Edutella overlays is using HyperCup concept [4, 20]. We are aware of the network overlays would affect the searching strategies. However, we just assume that the routing direction is determined by the routing index information no matter what kind approaches network overlays. P2P network overlay is the connection between nodes in the network. Please refer to [19, 21] for further reference on network overlays.

As a basic illustration on how the query routing works in this environment, we illustrate a query routing scenario in Figure 1. The same scenario will be used throughout this paper. Figure 1 show that client-peer *p1* as the *querying peer* sends a query to request for data 'a' and 'b'. We assume that data 'a' is obtained by client-peers *p2* and *p3* while *p4* is the location of data 'b'. Figure 2 shows the sequence of query routing in a standard super-peer system. The super-peers index obtaining the metadata of shared resources. Thus, the *querying peer p1* sends the request message for the source location of 'a' and 'b' to super-peer *sp1*. This message is a request for looking up for promising remote peers. Then, the same request message is routed to the neighboring super-peers which are *sp2* and *sp3*. Therefore, request for data location of 'ab' messages are consulted by super-peers' index *sp1, sp2* and *sp3*. Since the super-peer index is the collection of metadata, returned message from these super-peers are the metadata concerning for further query terms. From the collected metadata, *p1* would determine where to route his query. In this scenario, the query is routed to *p2, p3* and *p4* request for the results.

2.2. Materialized views at super-peer

Materialized view can be defined as saved or cached query. The materialized view has been implemented in database integration since the era of data warehousing system to the federated database system [22, 23]. In P2P environment, this similar concept has been adapted by Edutella [24]. In this subsection, we are describing a schema-based P2P system with view-based query processing. We are referring to [24] as the P2P systems that adopting the materialized views concept. One of the significant reasons for implementing materialized views is saving the overhead of query manipulation costs. This is because; data for the view is assembled when the view is created. Thus, the query is manipulated based on the pre-manipulated data.

Figure 4 is the sequence diagram to illustrate the query routing when materialized view is obtained by the super-peer. In this figure, we are assuming that all required data source information is cached by the super-peer *sp1* view. Thus, searching for required data location of 'ab' is based on the pre-stored queries in super-peer *sp1*. Instead of retrieving metadata only (as in the standard super-peer systems), the cached query mechanism would be able to return the complete queries for further routing. In this scenario, *sp1* would return the existing matched cached queries to *p1* that will be used for further routing to *p2, p3* and *p4.* We are aware of the number of return results is closely related to the query matching techniques that has been implemented. There are some query matching techniques that allow for approximately matched. The approximately matching would be able to return better results in terms of quality and quantity. However, the query matching technique is out of our scope of this paper. Please refer to for further readings on the query processing of similar queries. [19, 25].

2.3. Query result caching

Instead of caching the queries (materialized views), the third approach is about caching the query result. In P2P systems, the query result caching is implemented when the system are suffer of response time [26, 27]. The query response time can be varied depending on variety of factors such as network load, interval peer failures, low performance peers, network bandwidth and etc. By storing the query result, the *querying peer* would be able to obtain the required data just by sending a complete query message to the respective peer. In this scenario we assume that the super-peer caching the query results. Then, the queried peer would be able to manipulate the query and also return the cached result. As illustrate in Figure 3, the *querying peer p1* sends a complete query to *sp1*. Query result that has been obtained in *sp1* is returned to *p1*. Other peers doesn't involved in this routing.

2.4. Source Cached List (SCL) at client-peer

SCL is the caching strategy that only cached the information of actual data location (source) from the previous query. In contrast to previous architectures, our proposal is to cached the resource location on client-peer as discussed in [28]. Enabling the client-peer to obtain the routing direction, this similar concept has been proposed in P2P-DIET as discussed in [9, 29]. P2P-DIET can be seen as a fault-tolerance module for super-peer failure. P"P-DIET are applying an ad-hoc query processing where fast encoding profile of peers has been indexed at client-peers. Thus, each participated peer has to obtain the profile index of each local client-peers which is very similar to super-peers routing index. We had seen this approach as replicating the super-peer indexes at every peer in the network. Instead of maintaining the fast encoding index, our client peers are maintaining the SCL for assisting the query routing which is cheaper than maintaining the index. We are referring to [28] as the third architecture for further discussion. This architecture is piggy-back on the standard super-peer system while the cached information that consists of resource locations is obtained by client-peer.

The query routing scenario in this architecture is illustrates in Figure 5. This figure shows the retrieval for data ‘a’ and ‘b’ is directly route to the resource locations *p2, p3* and *p4.* Once again, we are assuming the required location information is already being cached by *p1* SCL. In this architecture, querying peer would be able to send the complete query to the selected peer which is the most promising data location that has been identified by the SCL. Thus, the queried peer will return the required results.

For the architectures in 2.2, 2.3 and 2.4, in the case of the required data is missing (not cached), normal query routing as in standard super-peer architecture is applied, where the query or sub-query is routed to *sp1* for further re-routing.

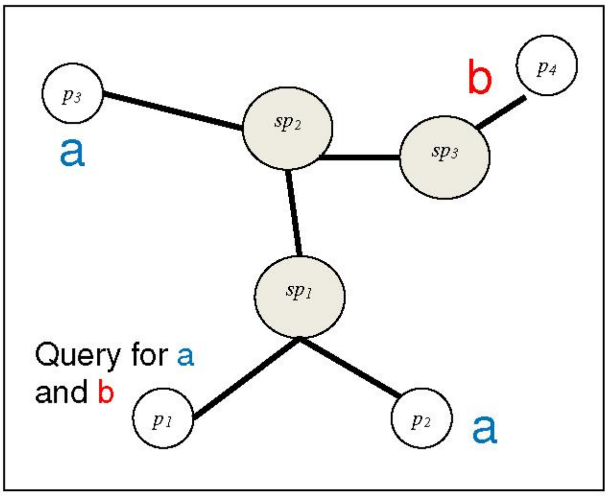

Figure 1. Example of query scenario

3. Comparative discussion

The cost of query processing is the sum of operational components cost. There are many possible cost metrics. However, most metrics reflect the amounts of system resources consumed by the query processor. System resources may include: i) CPU time, ii) disk access time for local data access, and also iii) the network bandwidth for communication when the resource is distributed. In this section, we are not intending to compare the actual cost of query processing between the abovementioned architectures. But, we are identifying the contributing factors towards the query processing cost that we consider as cost parameters. We are considering the following parameters to compare the tradeoffs factors for the architectures that have been discussed in the previous section.

- Number of message passing
- Cost of routing directory mechanism

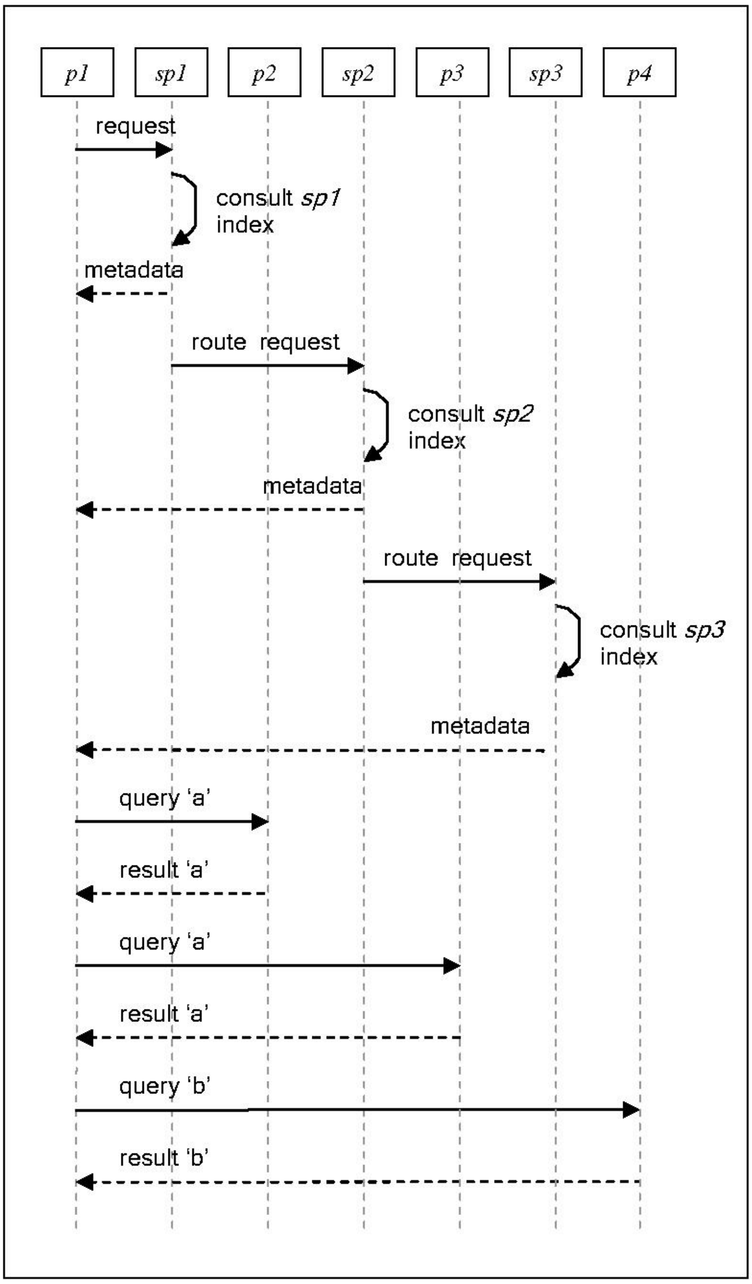

Figure 2. Sequence diagram for query routing scenario in a standard super-peer system

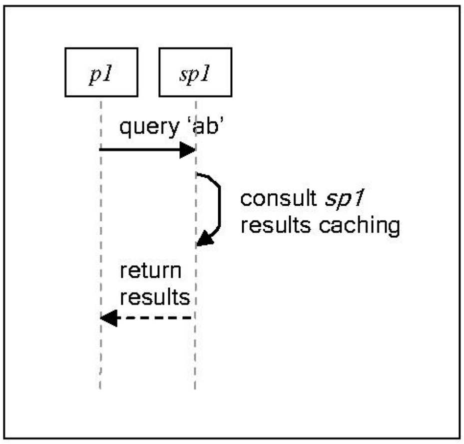

Figure 3. Sequence diagram for query routing scenario in a super-peer system with query results caching

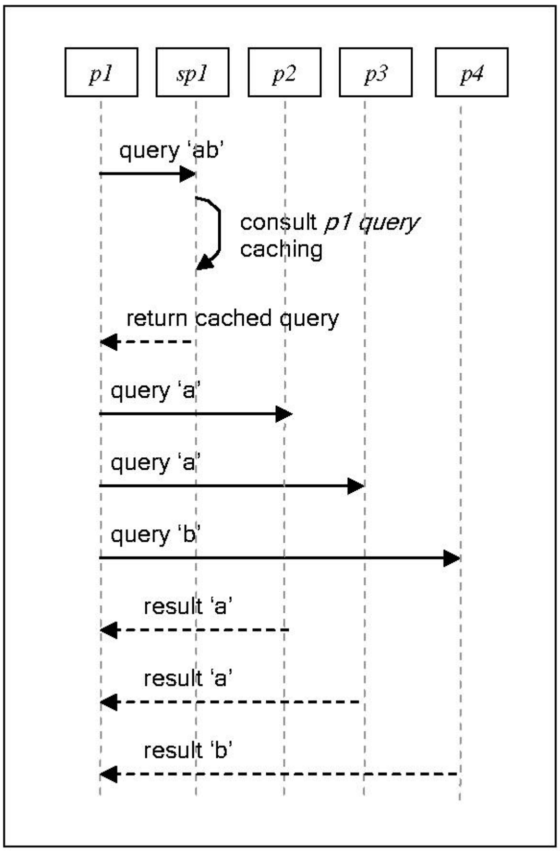

Figure 4. Sequence diagram for query routing scenario in a super-peer system with query caching

We are aware of the other operational costs such as cost of query processing at querying peer and cost of query manipulation at the queried peer. These kinds of costs are considered as constant since it is subsist within compared architectures. In this paper, we are intends to identify the trade-offs parameter in query processing cost. Therefore, any constant costs are excluded in our comparison. The following sub-sections will derive the cost parameters that have been considered within four different architectures.

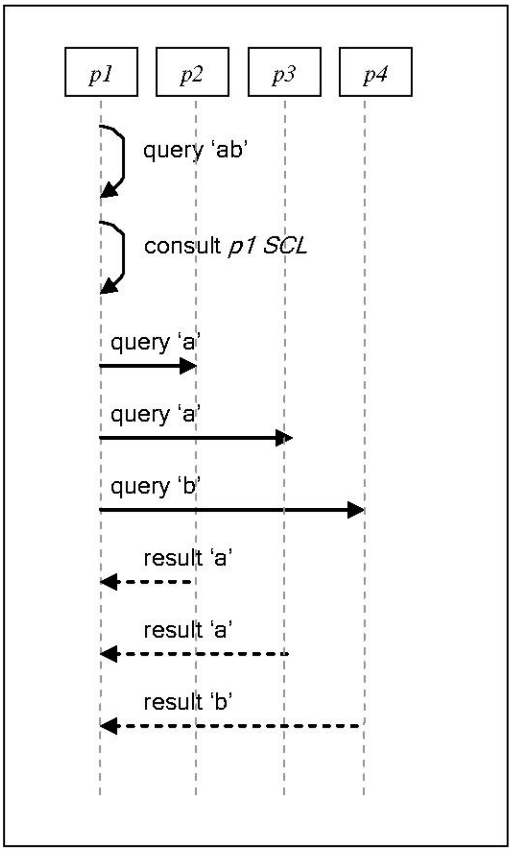

Figure 5. Sequence diagram for query routing scenario in a super-peer system with SCL

3.1. Message passing

In order to be effective, a P2P search mechanism has to address the following two conflicting requirements while keeping the low computational costs: i) providing a quick query response time with highly peer shared data scalability; and ii) providing a high quality query results while reducing the number of querying peers and number message passing in the network. With the intention to support the abovementioned conflicting requirements, research in query processing within P2P environment has utilize the use of caching facilities. Somehow, the caching is used to obtain the queries or query results, or even the data source location. As shown in Figure 3 to 5, the use of query results caching require the least number of message passing. This is because; the message passing is just between querying peer and the peer who obtained the cached information and actual data. However, the use of cached query result is correlated to the outdated result which is against the second requirement.

To prove the use of cached would reduce the number of messages passing in the network thus reduction in the number of queried peer, we have extended an existing simulator that has been used in [30]. This simulator has been written in Java and it is initially based on pure P2P network with the routing direction towards the schema location that has been cached. We have modified this simulator for simulating a super-peer network routing with and without caching mechanism. There are three routing behaviors have been compared, a standard super-peer routing which is without cached, a super-peer with a cached mechanism located at super-peers and a super-peer system with cached mechanism located at client-peers. In this simulation, each client-peer is directly connected to a single super-peer. Based on the pre-existing simulation setup that has been constructed, there are 900 peer nodes where each super-peer is allocated to four client-peers. For the architecture with the caching facilities, each corresponding node obtaining for 50 cached data. This cached data is used for query routing purpose. Each query has 8 TTL (time-to-live) values. Based on the simulation result, caching mechanism has tremendously reduced the number of message passing in the network. Caching at client-peer has gives slightly more reduction.

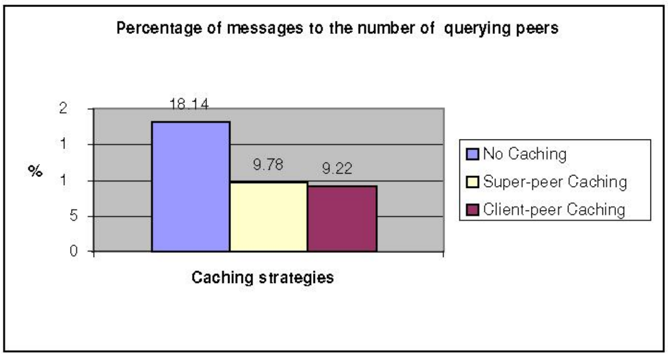

Figure 6. Percentage of messages to the number of querying peers

As a conclusion to this sub-section, we are definitely agreed that the use of caching would be able to reduce the number of message passing in the network.

3.2. Routing directory mechanism

As the shared data and the participated peers continue to grow exponentially, locating for desired information becomes more difficult. Routing index that is used to locate the query required data is grow proportionally to the number of connected peers and the shared data [27]. On the other hand, the query response time is also proportional to the number of queried peers[27]. So, it could be wise to investigate alternatives. In this section, we are comparing the mechanisms that have been used as routing directory which are super-peer routing index (CRI), query caching (CQC), query result caching (CRC) and SCL (CSCL). The followings are 2 cost variables that have been taken into account.

- Cost of searching within the directory (per message)
- Maintenance cost

Table 1: Cost comparison

Cost	CRI	CQC	CRC	CSCL
Maintenance	$\frac{Nt}{dist(Nt)} * N$	$\sum_{q=1}^{Q} Nt_q$	$\sum_{q=1}^{Q} R_q$	$\sum_{k=1}^{K} Loc_q$
Searching	$O(dist(Nt))$	$O(Q)$	$O(Q)$	$O(1)$

Table 1 derived the comparison between architectures. *N* is the number of nodes in the network, *Nt* is the number of tuples while *dist(Nt)* is the number of distinct tuples. *Q* is the number of queries and *K* is the number of key. *R* is the cost associate to the store a query result while *Loc* is the cost associate to the cost to store a data location. As shown in Table 1, searching and maintenance cost for the super-peers' routing index is proportional to the number of connected nodes, *N*. Meanwhile, the query and query result caching cost is proportional to the number of queried being processed for cached, *Q*. However, the SCL maintenance is free from the growth of connected of peers *N* and cached queries *Q* as well. This is because; the cached data location in SCL is indexed by key *K* which is not proportional to the abovementioned factors. Therefore, as a conclusion of this sub-section, we are recommending the SCL as the best routing directory, as an alternative option for the routing index of scalable super-peers P2P systems

4. Conclusion and future works

In this paper, we are studying the behavior and the trade-offs parameters of query answering process within four different architectures that are based on super-peer P2P systems. Modeling the cost of search mechanism is a complex task. We can model the cost based on the resource used in the P2P (eg. network, storage space, processing power) or based on user experience (eg. query response time). In current P2P system, the critical resource is the network [12]. Therefore, this paper has focus on the network and we use the number of message generated by each architecture as the comparative parameter. It doesn't means that user-based factor is unimportant, but by focusing on the network we are also improving the user-based factor.

Since the main significant of our compared architectures are the routing directory mechanisms, this mechanism is also selected as the second comparative parameter. Within each routing directory mechanism, we are dividing 2 main cost variables: maintenance and searching cost. These selected variables would be able to give major impact for the comparison.

Our future works is to derive the actual cost of each architecture and scale-up each parameter and variable. This work has to be done in order to identify for the most considerable cost of actions.

References

[1] Sharman, N., Ltd. *Kazaa*. 2002-2006 [cited; Available from: http://www.kazaa.com.

[2] YouTube, L. *YouTube*. 2008 [cited; Available from: http://www.youtube.com/.

[3] Calvanese, D., et al., *Hyper: A Framework for Peer-to-Peer Data Integration on Grids*. Lecture Notes in Computer Science, 2004. 3226/2004(Semantics of a Networked World): p. 144-157.

[4] Nejdl W., W.B., Qu C., Decker S., Sintek M., Naeve A., Nilsson M., Palmer M., Risch T. *EDUTELLA: A P2P Networking Infrastructure Based on RDF*. in *1th international conference on World Wide Web*. 2002. Honolulu, Hawaii, USA: ACM.

[5] Igor, T., et al., *The Piazza peer data management project*. SIGMOD Rec., 2003. 32(3): p. 47-52.

[6] Boyd, M., et al., *AutoMed: A BAV Data Integration System for Heterogeneous Data Sources* Lecture Notes in Computer Science, 2004. 3084/2004(Advanced Information Systems Engineering): p. 82-97.

[7] Haase P., B.J., Ehrig M., Menken M., Mika P., Plechawski M., Pyszlak P., Schnizler B., Siebes R., Staab S., Tempich C. *Bibster - A Semantics-Based Bibliographic Peer-to-Peer System*. in *International Semantic Web Conference (ISWC2004)*. 2004. Hiroshima, Japan: Springer Berlin Heidelberg.

[8] B. Pourebrahimi, K.B., S. Vassiliadis. *A Survey of Peer-to-Peer Networks*. in *16th Annual Workshop on Circuits, Systems and Signal Processing*. 2005.

[9] Stratos, I., K. Manolis, and T. Christos, *P2P-DIET: an extensible P2P service that unifies ad-hoc and continuous querying in super-peer networks*, in *Proceedings of the 2004 ACM SIGMOD international conference on Management of data*. 2004, ACM: Paris, France.

[10] Mohamed, R. *A Review of Peer-to-peer Systems*. in *International Conference on Electrical Engineering*

and Informatics (ICEEI 2007). 2007. Bandung, Indonesia.

[11] Diego, C., et al., *Logical foundations of peer-to-peer data integration*, in *Proceedings of the twenty-third ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems*. 2004, ACM: Paris, France.

[12] Crespo, A. and H. Garcia-Molina. *Routing Indices for Peer-to-Peer Systems*. in *22 nd International Conference on Distributed Computing Systems (ICDCS'02)*. 2002. Washington, DC, USA: IEEE Computer Society.

[13] Fegaras, L., He, W., Das, G., Levine D. *XML Query Routing in Structured P2P Systems*. in *DBISP2P2P Workshop in 32nd International Conference on Very Large Data Bases (VLDB)*. 2006. Seoul, Korea.

[14] He, W., Fegaras, L., Levine, D. *Indexing and Searching XML Documents Based on Content and Structure Synopses*. in *24th British National Conference on Databased (BNCOD)*. 2007. Glasgow UK: Springer.

[15] Nejdl, W., Wolpers, M., Siberski, W. , Schmitz, C., Schlosser M., Brunkhorst, I., Loser, A. *Super-Peer-Based Routing and Clustering Strategies for RDF-Based Peer-To-Peer Networks*. in *12th international conference on World Wide Web*. 2003. Budapest, Hungary: ACM.

[16] Kokkinidis, G. and V. Christophides, *Semantic Query Routing and Processing in P2P Database Systems: The ICS-FORTH SQPeer Middleware* Lecture Notes in Computer Science, 2005. 3268/2005(Current Trends in Database Technology - EDBT 2004 Workshops): p. 486-495.

[17] W3C (2007) *XQuery 1.0: An XML Query Language*.

[18] Sartiani, C., Manghi, P., Ghelli, G., Conforti, G. *XPeer : A Self-Organizing XML P2P Database System* in *9th International Conference on Extending Database Technology (EDBT), PhD Workshop*. 2004. Heraklion, Greece: Springer Berlin / Heidelberg.

[19] Marcel, K., et al., *Cost-Aware Processing of Similarity Queries in Structured Overlays*, in *Proceedings of the Sixth IEEE International Conference on Peer-to-Peer Computing*. 2006, IEEE Computer Society.

[20] Mario, S., et al., *HyperCuP — Hypercubes, Ontologies, and Efficient Search on Peer-to-Peer Networks*. Lecture Notes in Computer Science, 2003. Volume 2530/2008(Agents and Peer-to-Peer Computing): p. 133-134.

[21] Beverly, Y. and G.-M. Hector, *Improving Search in Peer-to-Peer Networks*, in *Proceedings of the 22 nd International Conference on Distributed Computing Systems (ICDCS'02)*. 2002, IEEE Computer Society.

[22] Batini B., L.M., Navathe S. B., *A Comparative Analysis of Methodologies for Database Schema Integration*. ACM Computing Surveys, 1986. 18(4): p. 323-364.

[23] Al-Mourad, M.B., Gray, W. A., Fiddian, N. J. *Semantically Rich Materialisation Rules for Integrating Heterogeneous Databases* in *22nd British National Conference on Databases (BNCOD 22)*. 2005. Sunderland, England, UK: Springer Berlin / Heidelberg.

[24] Brunkhorst, I., Dhraief, H., *Semantic Caching in Schema-Based P2P Network*. Lecture Notes in Computer Science, 2007. 4125/2007: p. 179-186.

[25] Benjamin, A., et al. *Approximating Aggregation Queries in Peer-to-Peer Networks*. 2006.

[26] Bobby, B., et al., *Efficient Peer-To-Peer Searches Using Result-Caching*. Lecture Notes in Computer Science, 2003. 2735(IPTPS): p. 225-236.

[27] Christian, Z., B. Srikanta, and W. Gerhard. *Standing on the Shoulders of Peers: Caching in Peer-to-Peer Information Retrieval*. in *33rd international Conference on Very large Data Bases (VLDB)*. 2007. Vienna, Austria.

[28] Rozlina, M. and D. Christopher, Buckingham. *Pre-processing for Improved Query Routing in Super-peer P2P Systems*. in *2008 IEEE Region 5 Technical, Professional, and Student Conference* 2008. Kansas City, USA.

[29] Manolis, K., et al., *Selective information dissemination in P2P networks: problems and solutions*. SIGMOD Rec., 2003. **32**(3): p. 71-76.

[30] Doulkeridis, C., Norvag, K., Vazirgiannis, M. *Schema Caching for Improved XML Query Processing in P2P Systems*. in *6th IEEE International Conference on Peer-to-Peer Computing*. 2006. AUEB, Greece: IEEE Xplore.