IMPLICA[illegible] PEERS

ROZLINA MOHAMED

Doctor of Philosophy

ASTON UNIVERSITY

June 2014

LIST OF CONTENTS

LIST OF ACRONYMS

API	Application Programmable Instruction
CON	Coordinator Overlay Network
CPU	Central Processing Unit
CSD	Common Schema Description
DCT	Distributed Cache Table
DHT	Distributed Hash Table
HON	Hybrid Overlay Network
HT	Hash Table
HTML	Hypertext Markup Language
HTTP	Hypertext Transfer Protocol
GAV	Global Access View
GLAV	Global-Local Access View
HTTP	Hypertext Transfer (or Transport) Protocol
JXTA	Juxtapose (peer-to-peer protocol specification)
I/O	Input / output
IP address	Internet Protocol address
IR	Information Retrieval
LAV	Local As View
LFU	Least Frequently Used
LRU	Least Recently Used
NFU	Never Frequently Used
OS	Operating system
P2P	Peer-to-peer
PC	Personal computer
PDA	Personal digital assistant
RAM	Random Access Memory
RDF	Resource Description Framework
LAV	Local Access View
UML	Unified Modeling Language
SRDI	Shared Resource Distributed Index
TCP/IP	Transmission Control Protocol / Internet Protocol
TTL	Time-To-Live
W3C	World Wide Web Consortium
XQuery	Query language for XML data
XML	Extensible Markup Language

LIST OF FIGURES

LIST OF TABLES

Chapter 1: **Introduction**

1.1. **Overview**

The focus of this thesis is on peer-to-peer (P2P) networks and how to make query processing more efficient. It will explore problems with query routing and propose a new approach for reducing number of query re-routings by using query caching of historical data. The idea behind caching data in P2P networks is to keep information about where the data source(s) is located. Thus, the process of identifying data location for a repeated query does not have to start from scratch. There are many alternatives for query routing (Pourebrahimi, Bertels et al. 2005, Mohamed 2007, Mohamed and Satari 2009) but the goal of this PhD research is to provide an adaptation of current processes that stores some information about the data source location together with knowledge about the hierarchical schema structure of data at the source location and, more crucially, to do this at the local peer where the query is made, as this will ensure the local peer is able to determine routing directions for its query. The cached query is obtained from the query history that has been locally executed by the peer in a P2P network and used to access the data source directly rather than requesting a search for the location over the P2P network. Thus, this research introduces a new query routing approach based on caching and a new method of evaluating the impact of the caching on the super-peer network.

Providing users with cached queries has raised significant attention as a way of reducing the query processing cost. Query caching has been widely implemented in query processing over centralized and distributed, as well as P2P, data-sharing systems. The idea used in query caching in this research is to keep the hierarchical structure as a query schema. The query schema can then be used by subsequent queries that share some of this structure. These new queries will be able to exploit the location information of target data that is held within the shared part of the schema. In this research, pre-processing for query routing is proposed to facilitate the functionality. This pre-processing consists of locally identifying the location of the target data for a given query, then directly routing the query message towards the specified data location complete with the schema structure of the identified target data, instead of it being re-routed to several other locations. As a result, the number of messages being routed in the network is greatly reduced along with the associated query routing time. Consequently, the aggregate query routing cost is much lower when the query is matched at the peer level, because the data can be directly accessed. This thesis will explain how the query caching works at the local peer and will demonstrate the potential savings for different types of network architectures.

1.2. Research background

P2P technology has the potential to enhance large-scale database sharing (Androutsellis-Theotokis and Spinellis 2004, Bellahsène, Lazinitis et al. 2006, Modarresi, Mamat et al. 2008, Mohamed and Satari 2009). Moreover, P2P offers the possibility of exploiting the local content of any peer in a network by any other peer, thereby breaking information monopolies. The P2P system offers great flexibility and decentralization, in addition to being highly resistant to faults. This is due to the fact that P2P does not rely on any centralized resources. In conjunction with database query processing, the design of P2P applications should significantly improve the ability to find relevant or potential answers to any given query, optimize the search cost by reducing the network traffic and issues concerned with peers' availability and autonomy of shared data (Bellahsène and Roantree 2004, Brunkhorst and Dhraief 2005, Doulkeridis, Norvag et al. 2006, Doulkeridis, Nørvåg et al. 2008). In order to locate the query result, a query will be broadcast to several other peers that may or may not be able to obtain the required answer. However, they have to process the incoming query message and react to it. The reaction is either an acknowledgement message to the sender that they have obtained the answer or a resending of the query to its neighbor. In addition, a peer that received the query may decompose it into several sub-queries, then send these sub-queries to multiple neighboring peers. The resending process will occur until the answer is found or the query message reaches the maximum level of its TTL (*time-to-live*) value that has been initially setup for it. The TTL value is the number of hops (i.e. message passing from one peer to the next) allowed for one query message over the network. The process of broadcasting the query message to obtain the result is called *routing*.

Query routing in P2P networks is based on overlay networks that hide the physical network topology. Depending on how the peers in the overlay network are linked to each other, P2P networks can be classified as structured or unstructured. An unstructured peer-to-peer application is a P2P application that has no server function present whatsoever. All communication occurs between clients, who are designated peers, and this might be termed a "pure" P2P application. Illustration of unstructured P2P is depicted in Figure 1.1. Searching for 'X' begins when Peer 1 sends a query message asking for 'X' from neighboring peers, which are Peer 2 and Peer 4 as illustrated in Figure 1.1. Then, the message is re-routed to Peer 3, 5 and 6 until the requested answer is found. Once the query reaches the location of the target data, the requested data is directly sent to the requestor.

In contrast, structured peer-to-peer is a P2P application that introduces a server for control and coordination purposes. As with the unstructured peer-to-peer environment, information exchange is still being passed directly between the clients (designated as peers). However, the server assists in helping the peers find one another and may assist in coordinating connections between them, tracking their progress or status. Figure 1.2 illustrates the structured P2P network overlay, which shows that Peer 1 will only ask the data directory while searching for data 'X'.

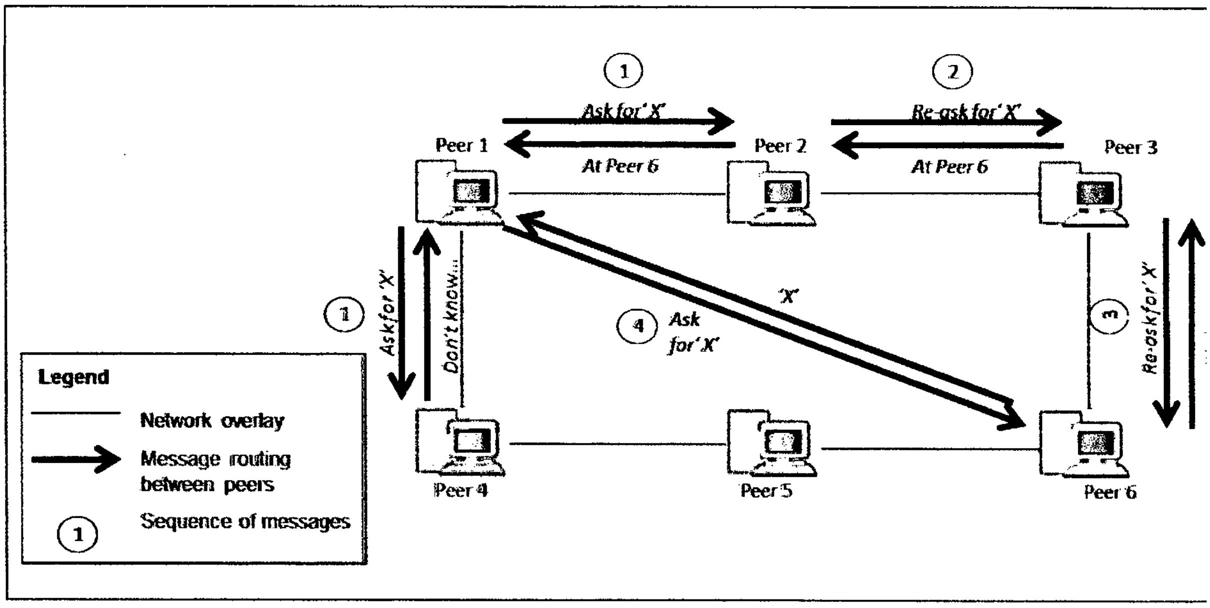

Figure 1-1Illustration of an unstructured network of querying for data 'X'

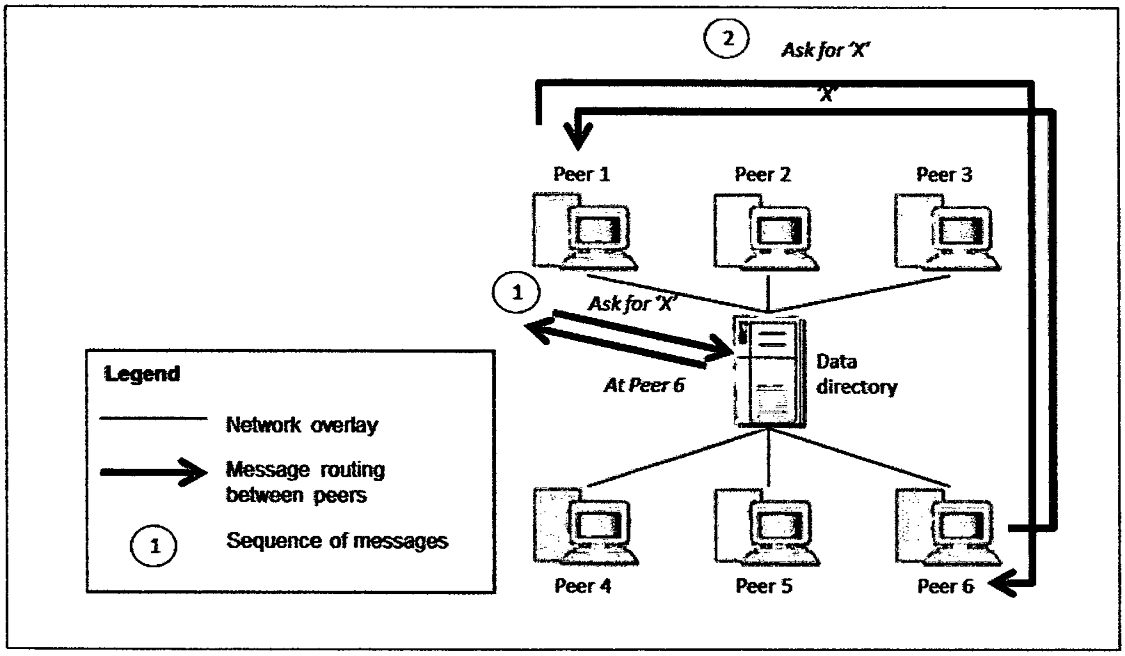

Figure 1-2Illustration of a structured network of querying for data 'X'

Query routing in unstructured networks has been implemented in Gnutella, Freenet and KaZaA whicharebased on the *query flooding* approach where each peer broadcasts the received query to directly connected peers(Good and Krekelberg 2003, Beijar 2010), whichare also known as *neighboring peers*.Generally, a P2P network consists of a large number of peer nodes where each peer is connected not to all other peers, but a small subset of the peers. In query flooding, if a peer wants to find a resource on the network, which may be on a peer it does not know about, it could simply broadcast its search query to its neighboring peers. If the neighbors do not have the resource, it then asks its neighbors to forward the query to their neighbors in turn. This is repeated until the resource is found or all the nodes have been contacted, or perhaps a *time-to-live* value (TTL), which is the network-imposed hop limit, is reached. Query flooding is normally practical for small networks with few requests. It contacts all reachable nodes in the network and so can precisely determine whether a resource can be found in the network. Instead, every query request may cause every peer within the same P2P network to be contacted. Each peer might generate a small number of queries; however, each such query floods the network. Thus, a larger network would generate far more traffic per peer than a smaller one, making it not scalable.Query flooding is efficient for locating popular data objects for which several duplicate copies exist in a large number of peers. On the other hand, query flooding would influence the search quality and performance cost for remote unpopular data objects, as an unpopular data object may not be found before the TTL limit is reached, or may incur a high search cost if it is eventually found(Ratnasamy, Francis et al. 2001).In query flooding, several duplicate copies may be available within the TTL limit. A large TTL number would increase query result expansion, and the possibility of finding the required result. However, the search cost exponentially increases with the expansion of search sites. Ideallythe query should onlybe broadcast to potential result locations, which would reduce the search cost without compromising the possibility of obtaining query results.

Besides unstructured, another logical P2P connection is structured P2P. The structured P2P can be divided into three sub-categories: central index, distributed index and publish/subscribe. Details of each category will be described in Chapter 2. In distributed index systems, query routing usesa Distributed Hash Table (DHT) to identify the location ofcontent and thusperform direct queries to the appropriate peers. DHT-based techniques have been widely used as the locatorin several structured P2P projects(Gong 2001, Ion Stoica 2001, Ratnasamy, Francis et al. 2001, Rowstron and Druschel 2001, Meshkova, Riihij et al. 2008). Peers and their data are given a unique identifier,and are grouped to create an overlay subnet. Each subnet will select at least one peer to become a super-peer. The super-peer is a hub which allows the local subnet to connect to other subnets in the network as a whole.At the same time, this super-peer is responsiblefor maintaining a routing table of its local subnet. The routing table consists of identifiers for peers in the local subnetand isused as a locator to identify the appropriate data location for answering queries. The query is routed to and processed by the peer that peer that contains the corresponding identifiers that match the required data. The idea of having the routing table as a locator for query result has been adopted in super-peer P2P network architectures. The routing table is used to improve the search efficiency as the routing table provides a deterministic routing and a high recall of the required data. However, selected peers for obtaining the routing table are required to give their high commitment to the network community. On the other hand, the peers that have the routing table would become a single point of failure for their cluster if they suddenly left the P2P network (Yang and Garcia-Molina 2003, Pourebrahimi, Bertels et al. 2005). Thus, the high flexibility and autonomous features offered by a dynamic P2P network could be abused by the selected peers that own the routing table on behalf of their cluster.

In this research, the super-peer P2P network overlay is chosen for embedding the alternative approaches as the use of central coordinating servers and directed search requests can be used to coordinate the peers' activities as well as reduce the number of message passing through the P2P network while searching for the answer. Accordingly, the directed search request contributes towards reducing the cost of query routing.

1.3. Research Motivations

Efficient query routing in P2P systems is a highly active research area with a plethora of publications (Brunkhorst, Dhraief et al. 2003, Yang and Garcia-Molina 2003, Leonidas Fegaras 2005, Ismail, Quafafou et al. 2009, Ismail, Quafafou et al. 2009, Fegaras 2010), which testifies to the importance of the topic. Efficient query routing aims to limit network bandwidth consumption by reducing the number of messages across the network and reducing the total query processing cost by minimizing the number of peers that contribute to the query results. Query routing in a super-peer network is a process of routing the query to a number of relevant peers without having to broadcast the query message to the whole network. The problem is concerned with the discovery of the relevant peer for a particular query (Ismail, Quafafou et al. 2009). As a result, data localization and routing in P2P networks are closely related to one another in producing the query result. A survey on the growth of P2P network traffic in Japan indicated that 63% of the residential traffic volume is contributed by P2P users, which is about 37% overall increment of P2P users per year (Cho, Fukuda et al. 2006).

Consequently, there have been attempts to optimize the P2P traffic by, '...placing super peers close to the subscribers and *caching popular content*.' (Tschofenig and Matuszewski 2008). However, caching the popular content in several places has potential pitfalls related to copyright issues (Lohmann 2006). In addition, the nature of the super-peer network overlay will always route the query via the super-peer node. Thus, it is important to keep a high availability of the super-peer nodes. The higher availability of super peers contributes towards several research directions, such as the intelligent selection of super-peer nodes (Gao and Min 2009, Min and Holliday 2009) and the use of multiple super-peer nodes for a single network cluster (Bellahsène and Roantree 2004, Pourebrahimi, Bertels et al. 2005, Bellahsène, Lazinitis et al. 2006). These research areas contribute towards maintaining the high availability of super-peer nodes. However, high availability of super peers comes with some trade-offs, such as an increase of the entire processing cost and the requirement for higher processing capabilities at client-peers. Furthermore, the traffic directed towards super-peers will remain unchanged. In contrast, this research contribution will suggest an approach of diverting the query direction towards the query result location rather than going-through the super-peer node (asking for query result locations). This means that the number of query requests to the super-peer node is decreased, hence reducing the network traffic towards the super-peer node.

This section discusses the motivation which supports the research in query routing over a super-peer network overlay. The discussion starts with an illustrated network, which represents a sample scenario as depicted in Figure 1.3. Let us assume that a user at Peer 1 in a super-peer network application posts a query in order to search for books on 'Database'. This query is labeled as *Q1*. In this scenario, let us assume that Super-peer A, which is the super-peer of Peer 1, does not have the required information. Thus, *Q1* is re-routed to its neighboring peers within the cluster (Peer 2, 3 and 4) and super-peers that have been logically connected (Super-peer B). Once again, let us assume that the required information is not obtained in the neighboring super-peer, and thus *Q1* is re-routed. The re-routing is repeated until the required information is obtained, or the maximum TTL value of the routing message is reached. Once the location for the required information is found, query answer retrieval and processing will be started.

As a second scenario, Peer 1 sends another query that is similar to *Q1*. The second query is labeled as *Q2*. *Q2* is searching for the author of a book entitled 'Database'. Again, Peer 1 contacts Super-peer A. Since the related information of *Q1* is captured by Super-peer A, *Q2* is not re-routed. Message passing is just between Peer 1 and Super-peer A. Then, the third query *Q3* is initiated by Peer 1, asking for details of the author of the 'Database'. Once again, communication between Peer 1 and Super-peer A is established for *Q3*. The scenarios illustrate that three queries *Q1*, *Q2* and *Q3* require service from Super-peer A.

Based on the above scenarios, there is no doubt that the super-peer has a high number of messages passing through it. Since the super-peer node is responsible for aggregating incoming client peer requests and forwarding them to the relevant peer or neighboring cluster, the super-peer workload is scaling with the number of query messages (Wu and Starobinski 2008). However, significant research concerning the message traffic that is always routed to the super-peer node has yet to receive particular attention from researchers. Therefore, the first aim of this research is to divert queries from being routed to the super-peer node by locally determining target location for similar or repeated queries. Thus, queries will only be sent to selected peer(s), which is the target location to obtain the query result (Wu and Starobinski 2008).

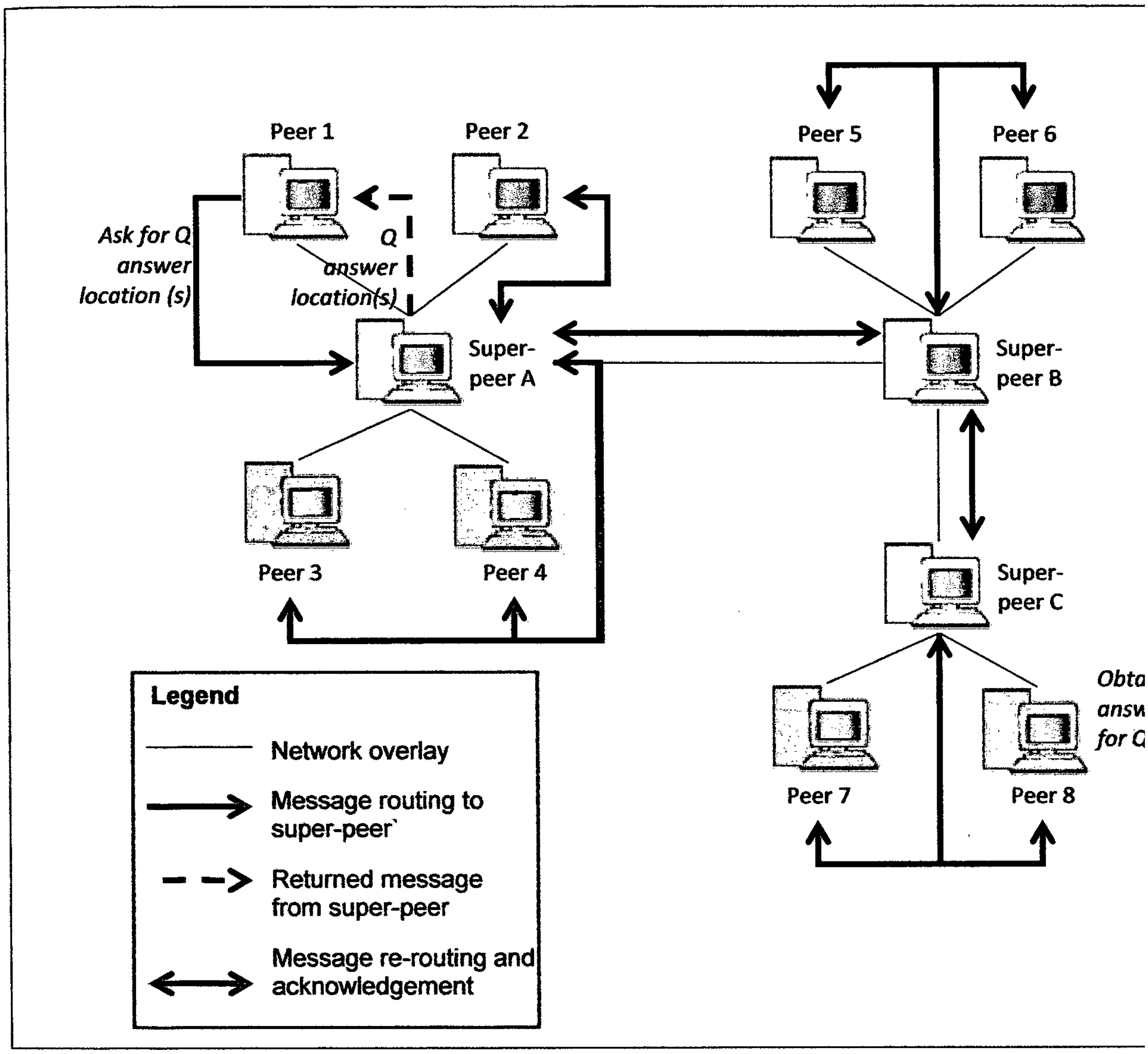

Figure 1-3An ordinary query routing in super-peer network

Besides reducing the number of re-routing messages, this researchwould also contributeto the reduction ofthe query re-writing effort. Before sending a query, it should be re-written in a form that can be understood by the receiver. In order that the recipient peer knows where to find the requested resource, the query has to be written using the same or similar schema of the receivers' data source. Therefore, caching the hierarchical schema structure of the specified data location is used in this research.

Research in database integration has widely explored the issues of caching a complete query message (known as 'materialized views'), caching an address of the data locations, or caching a query result(Watanabe and Kitagawa 2010, Sluijs, Iterbeke et al. 2011). However, caching actual results has led to copyright issues in P2P. Caching only the location of the result points the query routing towards a specific location (Sluijs, Iterbeke et al.), but does reduce the query processing and data access time. This research proposes a method for caching the pre-processed query, which is a skeleton of the query statement that comes from previous queries. By obtaining a pre-processed query, the new query does not require the entire query processing operation. As a result, the query processing cost is reduced and the query processing time is shortened.

The initial intention of this research was to reduce the number of re-routing messages and query processing costs. However, this led to the proposal of an additional query routing measurement that can be used specifically to compare the number of messages passing within the network according to the query routing approach used. The third research motivation contributes to the analysis of the initial system design. Previous researchers have introduced several performance models for measuring system performance, and the existing performance models are based on input that are available in the early stage of system development. For instance, software modeling in formal methods and Unified Modeling Language (UML) are amongst the best known approaches used during the requirements' analysis stage. To date, most system performance measurements carried out during the early stages of system development require additional system modeling as an input before the system is measured (Smith and Woodside 2000, Swan, Kutar et al. 2004, Abdullatif and Pooley 2009, Al Abdullatif and Pooley 2010, Albert, Cabot et al. 2011, Nieto, Costal et al. 2011). Thus, it requires some additional tasks for the software engineer, to transform their existing system model (as created during requirements analysis) into another system model required for the measurement. UML diagrams possess the capacity for being specifically interpreted (Parreiras and Staab 2010, Albert, Cabot et al. 2011, Nieto, Costal et al. 2011). This research proposes a new approach for predicting system performance based on a diagram available in UML.

In short, this research has three motivational factors. The first motivation is to divert query message routing away from the super-peer node for repeating queries. This first aim leads towards extending the capabilities of query routing in super-peer P2P network applications, which is tested on one of the super-peer applications named JXTA, a P2P platform specification developed by Sun Microsystems. JXTA can be used to develop a P2P system application by providing typical P2P operations such as registering a peer, creating a peer group, joining the peer group, resource query. Using JXTA, P2P application developers do not have to reinvent these basic operations.

The second motivation is to reduce the query processing cost by caching the existing manipulated query statement. Since the cached item consists of the skeleton of the query statement, the second aim will shorten the query re-writing task, which will result in a reduction of the query processing time.

The third motivation is to introduce a software performance measurement approach, which is used specifically for comparing the P2P applications conducting the query routing task. The third aim contributes to the area of software measurement, especially in the early stage of system development.

1.3.1 Research objectives

In conjunction with the motivational factors of this research, the following are the objectives of the research:

1. Design and develop an architectural framework as a pre-processing mechanism for assisting the query routing operation in super-peer networks on the JXTA platform (Mohamed and Buckingham 2010).
2. Determine performance bounds and identify the usability of caching at the local querying peer (as opposed to the super-peer) in attempting to reduce the query routing cost in super-peer P2P applications that share their XML data on the JXTA P2P platform.
3. Formulate a performance analysis model for comparing the query routing costs in a variety of super-peer routing approaches.

1.3.2 Research scope and limitations

In order to achieve the stated objectives, this research will focus on the query routing process for the JXTA platform while not looking at the query processing once the query result has been returned by the queried peer(s). This research is about caching at the peer level instead of at the super-peer, and it is an attempt to improve the process by breaking queries into sub-queries and linking the sub-queries to the most appropriate target peers that are sources for the sub-query's required data. Moreover, this research adopts a query caching concept in the proposed framework in order to prove the usability of query caching in a P2P environment.

1.4: Research overall achievement

This research has achieved several objectives that contribute to the research endeavor for P2P networks. The main contributions are as follows:

1. Creation of a computer taxonomy for P2P systems, which classifies various computer system architectures. The classification is aimed at showing the hierarchies of terms used in P2P architectures.
2. A comparative study of resource discovery mechanisms, which leads to a new resource discovery mechanism, as well as parameters for measuring the cost and benefits for each discovery mechanism.
3. A feasibility study on the use of materialized views in P2P queries processing. The use of materialized views for query processing in distributed database applications which will lead towards designing an architecture for implementing the query caching concept in a super-peer network. The research proposes a query cached list for keeping information about data source locations that have been used in previous queries executed locally at the peer.
4. Implementation of a query caching algorithm and its demonstration on JXTA platform. The query caching mechanism is used to keep the query history that has been executed by the local peer. This research is motivated by the belief that the amount of query routing is reduced when embedding the query cached list at the client-peer instead of the super-peer. A novel method of using UML sequence diagrams to monitor improvement in performance was developed to produce evidence that supports this belief.
5. An architectural design for pre-processing the query routing operations in JXTA P2P platform. The design has been implemented and evaluated, and has been shown to improve the performance, due to the reduction of the aggregate query routing time either at the client-peer or at the super-peer.
6. Algorithms for implementing the proposed query caching list together with the pre-processing mechanism for query routing have been developed. The algorithms have been implemented in

Java and piggy-backed on the JXTA platform for P2P super-peer networks to create the evaluation environment.

7. Comparative assessment of query routing performance. The main idea of comparative assessment is to compare the query routing strategies by considering each and every step (process) that contributes to the query routing process. Then, the processing time for each step is averaged and these averages are used to calculate the processing time for the process.

1.5. Thesis structure

This section presents an overview of the thesis organization.

CHAPTER 1: Introduction

The first chapter presents an introduction to the research undertaken, covering background, motivation, objectives, scope and limitations, as well as research achievements.

CHAPTER 2: Peer-to-peer Networks

This chapter gives necessary background information about P2P networks. It reviews some existing approaches in P2P system architectures in order to address the problem of single-point-of-failure, which is a motivational background for the research.

CHAPTER 3: Query Caching in Peer-to-peer Networks

This chapter creates a classification of the existing query routing approaches. It also presents some of the implementation issues associated with query caching and the importance of query caching in P2P environments.

CHAPTER 4: The Proposed Approach.

This chapter describes the logical foundations of the proposed query cached list and the pre-processing mechanism. It also describes the logical scenarios for testing the proposed concepts.

CHAPTER 5: Query Caching in JXTA.

This chapter provides the architecture and operations of the proposed concepts in the JXTA platform. It also presents the implications of implementing the proposed approaches on JXTA peers compared to the implementation of the ordinary query routing approach. The main purpose of JXTA is to provide a test environment for demonstrating the caching process and its effects.

CHAPTER 6: Performance Analysis.

This chapter focuses on the evaluation of the use of query cached lists and the proposed pre-processing mechanism for assisting query routing. The analysis compares routing performance with a query as the parameter. A UML sequence diagram is used to represent the various process flows of query routing. Each process flow is given a weight based on the average value of the actual processing time required for a specified task and a multiplier for the number of times it occurs in the overall query routing process. The product of weight and its multiplier represents the total processing time for a process flow. The summed process flows are then used to compare the performance of query routing approaches.

CHAPTER 7: Conclusions and Future Works

This chapter draws conclusions and identifies future work that could be carried out based on the achievements of this research or work that was identified but could not be addressed within the constraints of a PhD research project.