OPTIMAL ASSIGNMENT AS A LOCATION BASED SERVICE IN OUTSOURCED DATABASES

by

Ahmet Salih BÜYÜKKAYHAN

BSc., Computer Engineering, İstanbul Technical University, 2007

Submitted to the Institute for Graduate Studies in
Science and Engineering in partial fulfillment of
the requirements for the degree of
Master of Science

Graduate Program in Computer Engineering

Boğaziçi University

2011

OPTIMAL ASSIGNMENT AS A LOCATION BASED SERVICE IN OUTSOURCED DATABASES

APPROVED BY:

Prof. Taflan İ. GÜNDEM (Thesis Supervisor)	.....
Prof. Fikret GÜRGEN	.....
Assist. Prof. Mustafa AĞAOĞLU	.....

DATE OF APPROVAL: 19.04.2011

ACKNOWLEDGEMENTS

First of all, I would like to thank my thesis supervisor Prof. Taflan İ. GÜNDEM, for being such a good mentor, for his encouragements and for his enthusiasm in this work. I would also like to thank Prof. Fikret GÜRGEN and Assist. Prof. Mustafa AĞAOĞLU for their participation to my thesis committee and their valuable comments.

Special thanks to Özgür Oktay, my managers and mates for their understanding and support throughout my study. I would like to thank my friends who shared the happiness and pain with me.

Last but not least, I would like to thank my family for their endless love and support. This thesis is dedicated to them.

ABSTRACT

OPTIMAL ASSIGNMENT AS A LOCATION BASED SERVICE IN OUTSOURCED DATABASES

With the growth of mobile devices that have positioning capabilities, location based services promises great opportunities. Moreover to this, service providers would like to focus on their services instead of managing servers and they require flexibility to expand or shrink their infrastructure according to the market. These are the two strong drives for outsourced spatial databases. In the literature, several different queries such as nearest neighbor, K-nearest neighbor, proximity and privacy preserving techniques have been studied in outsourced spatial databases. In this thesis, the capacity and coverage constrained assignment query is adapted to the outsourced databases. Unlike the other assignment queries in fully connected graphs, we focused on sparse graphs which is more realistic for location based services. A novel spatial transformation strategy (square spiral encoding) is introduced to achieve privacy and performance requirements with approximate results. Approximate solution provides a trade off between result accuracy, location privacy and computation cost . For exact results, we also introduce a new method to calculate distance over encrypted spatial data. In the experiments, we compared the both methods and investigate their performance and costs.

ÖZET

DIŞ KAYNAKLI VERİ TABANLARINDA KONUM TABANLI SERVİS OLARAK OPTİMUM ATAMA

Konum belirleme yeteneklerine sahip mobil cihazların artmasıyla konum tabanlı servisler büyük fırsatlar vaat etmektedir. Bunun yanısıra, servis sağlayıcılar sunucuların bakım ve yönetimleri ile uğraşmak yerine sağladıkları servislere odaklanabilmeyi istemektedir. Servis sağlayıcıların bir diğer gereksinimi de bilişim altyapılarını pazar ihtiyaçlarına göre ayarlayabilmektir. Bu nedenlerden dolayı dış kaynaklı mekansal veri tabanlarına olan ilgi artmaktadır. Mekansal dış kaynaklı veri tabanlarında kişisel bilgileri korumaya yönelik çeşitli sorgular için geliştirilmiş yöntemler literatürde mevcuttur. Bu sorgulara örnek olarak, en yakın komşuyu bulma veya en yakın K tane komşuyu bulma verilebilir. Bu tez çalışmasında ise kapasite ve kapsama kısıtları olan bir atama sorgusu dış kaynaklı veri tabanları için uyarlanmıştır. Diğer atama sorgularından farklı olarak, konum tabanlı servisler için daha gerçekçi olması nedeniyle, seyrek diyagramlı atama sorgularına yoğunlaşılmıştır. Yaklaşık sonuçlar için hem gizlilik hem de performans gereksinimlerini karşılayan yeni bir mekansal transformasyon stratejisi (kare spiral kodlama) tanıtılmıştır. Sonucun isabetliliği ile konum gizliliği ve hesaplama maliyeti arasında bir denge mevcuttur. Örneğin, konum gizliliği artarsa, hesaplama maliyeti ve sorgu sonuçlarının doğruluk oranı azalmaktadır. Kesin sonuçlar için kullanılabilecek yeni bir yöntem daha önerilmiştir. Bu yöntem ile şifrelenmiş mekansal veriler arasında, şifre çözme işlemi yapmaya gerek kalmadan uzaklık hesaplanabilmektedir. Deneylerde her iki metot karşılaştırılmış, performans ve maliyetleri incelenmiştir.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF SYMBOLS/ABBREVIATIONS

C_q	Capacity of the query object q
C_d	Capacity of the query object d
$COST_{qd}$	Cost of assigning q to d cost of assigning q to d
E	Set of edges or arcs in the graph
G	Graph or weighted bipartite graph
Q	Set of query objects, bidders in auction
V	Set of vertexes or nodes in the graph

DB	Database
LBS	Location based service
MBI	Minimum bounding interval
MBR	Minimum bounding rectangle
MCF	Minimum cost flow
OPES	Order preserved encryption system
POI	Point of interest
SD	Spiral direction
SDK	Square spiral decryption key
SS-value	Transformed location data using square spiral space encoding

1. INTRODUCTION

With the growth of mobile devices that have positioning capabilities, location based services promises great opportunities. These ocation based services have already affected our life by providing us the location of the nearest hospital, bank or taxi station. In addition to the existing services, number of location based services increase almost ever day. In this fast growing business, location based service providers would like to focus on their services instead of managing servers and dealing with platform related problems. Those service providers also require flexibility to expand or shrink their infrastructure according to the market. These are the strong drivers to use outsourced spatial databases for location based services. However storing private data at the servers that are not fully trusted causes new privacy problems. Therefore, location based service providers should consider both data confidentiality and system performance in the outsourced databases.

Revealing location data could also cause other privacy violations. For instance; attackers could get the real identity of the individuals with great precision, based on the locations that they usually present such as homes, schools, work places etc., even if they hide their identity. Attackers could simply get their names from a web, phone or address book. identity protection and identification of seemingly anonymous users are studied for a long time in [1, 2].

In the literature, several different queries such as nearest neighbor, K-nearest neighbor, proximity and related privacy preserving techniques have been studied in outsourced spatial databases. They have proposed techniques based on anonymization, cryptography, Private information retrieval (PIR) and transformation. Each technique has some advantages and disadvantages. Majority of the anonymization approaches rely on a trusted intermediate party and they have a trade off between privacy degree and query accuracy. In less populated areas, the anonymization techniques does not work at all due to extremely large cloak regions. Cryptographic based approaches in general have very high computation and communication costs. PIR based approaches have lower

computation and communication costs when compared with cryptographic techniques but they are still expensive in terms of computation and communication costs. It is argued that sending the entire database to the client is more efficient when compared with the cost of privately retrieving items from the database using existing PIR techniques [3]. Furthermore, PIR based techniques requires prepossessing to provide the query results thus they are not applicable for dynamic queries. Existing transformation based techniques suffer from the privacy leaks and requires trusted intermediaries [4]. and the ones that provide strong privacy are just applicable to some specialize d queries such as NN and K-NN [5].

Motivated by the lack of dynamic queries in outsourced databases. We proposed efficient, privacy aware techniques that does not require any trusted third party for the assignment query. Even if our techniques could be used for other query types, we focused on the assignment query since existing location privacy aware methods are not sufficient to perform assignment query.

Assume that, there is location based service which retrieves the park spot by considering all park spot requests and available spaces in the park lots. This service does the capacity constrained assignment to reduce the overall costs (total burned gasoline, produced carbon dioxide etc.). Assume that, the users does not want to get a park lot which is far away from the determined threshold. So LBS provider should also consider the coverage distance for parking service.As a result, the service does the capacity and coverage contrained assignment on the spatial database.

In fact, the assignment query has been studied in spatial databases by several researchers. [6] proposed the assignment query for moving object databases and [7] proposed the capacity constrained assignment in spatial databases. However, [8] introduced the coverage distance to the capacity constrained assignment since it is more applicable to the real world scenarios. But there is no study for location privacy aware assignment query in the outsourced databases.

In this thesis, the capacity and coverage constrained assignment query is adapted to the outsourced databases. Unlike the other assignment queries in fully connected graphs, we focused on sparse graphs which is more realistic for location based services. We used Auction algorithm [9] due it is high performance on sparse graphs and its capability to be paralleled. The worst-case running time of the auction algorithm for integer data using scaling and appropriate data structures is $O(NA \log(NC))$ where N is the total number of bidders, A is the total number of all person-object pairs and C is the maximum absolute object benefit [10]. The average running time of the algorithm is $O(A \log N)$ [10]. In order to improve the performance we use some pruning strategies before applying the auction algorithm.

As we mentioned earlier, new location privacy aware techniques required to perform assignment query in outsourced databases. For this reason, a novel spatial transformation strategy (square spiral encoding) is introduced to achieve privacy and performance requirements with approximate results. Approximate solution provides a trade off between result accuracy, location privacy and computation cost. For exact results, we also introduce a new method to calculate distance over encrypted spatial data.

The rest of this report is organized as follows. Section 2, covers the existing work related to our proposed system for assignment query. Section 3, covers the background information to explain proposed algorithms and supporting index structures in the system. Section 4, covers the proposed assignment query for outsourced databases is presented with possible implementation details. Section 5, discusses about performance issues and evaluates the proposed system according to experiments. Finally, Section 6 summarizes and concludes the thesis work.

2. RELEATED WORK

Our work is related to the two main parts. First part is the assignment query for the database systems and second part is the outsourcing a location based service. In this section, we give information about previous related works and provide an overview about these parts. In optimal assignment section, we first mention about the optimal assignment problem, existing algorithms and then we briefly discuss the proposed assignment queries for the database systems.

2.1. Optimal Assignment

In literature, assignment problem is also known as bipartite matching problem [11, 12]. The bipartite matching problem is represented with bipartite matching graphs as seen in the Figure 2.1. The goal is to assign n items to another m items by maximizing or minimizing the objective function. If assigning an item to another item has a cost then the problem is called weighted assignment problem. In linear assignment, the number of items in left hand side should be equal to the number of items in right hand side and an item could not be assigned to more than one item and vice versa.

Weighted assignment problem is also a reduction of the minimum cost flow problem (MCF). In MCF, there is a weighted directed graph G(V,E) where V represents the vertexes and E represents the arcs between the vertexes. There are also two artificial vertexes called source and sink. Each arc has a weight and capacity. The MCF solution maximize the number of assigned pairs and minimize the summed weights respect to the capacity constraints. Capacity constrained assignment problem is mentioned in [7]. Unlike the MCF, capacity constraint assignment has only integer capacities and integer flows. Both problems could be solved using MCF algorithms with necessary capacities.

The Hungarian [13, 14] and the successive shortest path (SSP)[15] algorithms are the most popular algorithms for MCF problem. The lowest worst case performance of these algorithms are $O(V(E + VlogV))$ where V is the number of vertexes and E is the

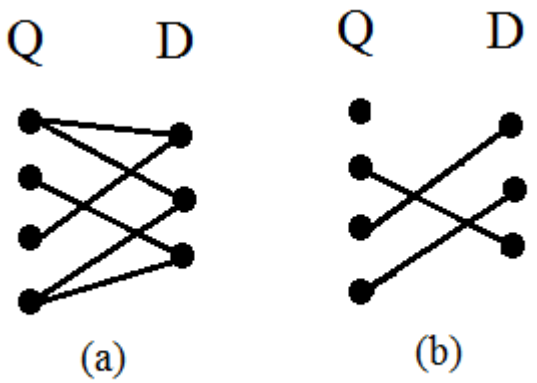

Figure 2.1. Bipartite graph (a) Before matching (b) After matching

number of edges. In fully connected bipartite graphs, $E = (\frac{V}{2})^2$ so the performance of these algorithms become $O(V^3)$.

The most efficient algorithm for sparse flow networks is Auction Algorithm [10]. Auction is used in business to sell the products with best prices to the buyers. Auction algorithm uses the same principle and could be applied to different problems including assignment, shortest path and transportation problems. It is shown that the worst-case running time of the auction algorithm for integer data using scaling and appropriate data structures is $O(NAlog(NC))$ where N is the total number of bidders, A is the total number of all person-object pairs and C is the maximum absolute object benefit [10]. Furthermore, Auction algorithm could be easily implemented to process in parallel as synchronous or asynchronous [16].

[6] proposed the assignment query to solve the assignment problem in moving object databases. The proposed query could be used only for one to one matching and the approximation algorithm produces 33% larger total cost when compared with the exact algorithms. [7] extended the assignment query by including capacity constraints. But this work focuses on the static data stored on the disks. Furthermore, they assumed the services have an infinite coverage and solved the problem in fully connected graphs. A recent study [8] presents a coverage and capacity constrained optimal assignment query. They use the more realistic scenario, where servers have a limited service region.

However they used the SSP algorithm with some enhancements and they mainly focus on the dynamic maintenance of the assignment in the memory, which is not the case in our problem. We focused on the snapshot assignment queries in our work.

None of the approaches mentioned above consider the outsourced databases where database is not fully trusted. To the best of our knowledge, there is no previous work that process the assignment query in parallel blindly over transformed or encrypted data.

2.2. Location Privacy

Location privacy is becomes more important with the growth of mobile devices that have positioning capabilities. Attackers could get the real identity of the individuals with great precision, based on the locations that they usually present such as homes, schools, work places etc., even if they hide their identity. Attackers could simply get their names from a web, phone or address book. Identity protection and identification of seemingly anonymous users are studied for a long time in [1, 2].

In the sections below, we have categorized the location privacy aware techniques as anonymity based approaches, cryptographic based approaches, private information retrieval based approaches and transformation based approaches. We also describe and discuss each of the approaches below.

2.2.1. Anonymity Based Approaches

Location [17, 18] and trajectory [19] K-anonymity is a major privacy paradigm for the protection of identity. K-anonymity protects a users private location information by disguising it among K - 1 other user locations or extending it from a point location to a cloaking area. In the first approach, user u and K-1 other users send their locations to form an anonymity set of locations. After that, this anonymity set of locations are sent to the server. Server executes the query for every location in the anonymity set and send back the query results. The query results are filtered and the user u gets the

query answer. Similarly second approach send a cloaking region that contains at least K-1 other users to the server. Server executes the query for entire cloaked region and send the result set back. The result set is filtered so the user u gets the query answer. These techniques try to ensure the users precise location can not be distinguished from the location of the other K-1 real/false positions (may include dummies) or the user's exact location within the cloaked region is not revealed to the untrusted server (e.g. outsourced database server) responding to location queries. In the literature, many approaches based on cloaking, K-anonymity, and dummies have been proposed to reduce the possibility of identifying a users location [1, 17-20].

First of all, most of the anonymization approaches rely on a trusted intermediate party. This means, all queries should be send to the anonymizer during the system's normal mode of operation. Anonymizer also increase the implementation and maintenance costs. Furthermore, anonymizers could became a potential point of failure or security breach. Instead of an anonymizer, queries could be anonymized among the users in a decentralized manner. In this method, each user should trust to all other users in the system. As a result, anonymization based approaches requires a trusted party either a trusted intermediate or trusted group of users. In general there is a trade off between service quality of the system with degree of the location privacy in anonymization based approaches. Anonymization based methods do not provide any protection to semantic locations , (hospital, etc.). Even if the server could not distinguish the K users, it ensures there are at least one user and at most K users in that hospital. Moreover to this, the concept of K-anonymity does not work in all cases. For instance, in a less populated area, the size of the cloaked region can be very large in order to include K-1 other users. Even worse than that, there may not be sufficient number of subscribed users to the service to provide K anonymity. On the other hand, anonymization based techniques are considerably efficient in terms of communication and computation overhead [17].

2.2.2. Cryptographic based approaches

This type of approaches utilize secure multi party computation schemes in order to blind the untrusted party (i.e., the server or another user) . The main advantage

of this approach is the Strong privacy guarantees. Encryption algorithms use one-way functions and it is almost impossible to decrypt the data without the key. Cryptographic techniques do not suffer from privacy leaks of anonymization and transformation. The challenge here is to execute the query efficiently over the encrypted data. [21] proposed bucketing technique to execute query over encrypted data. However, the communication cost and computational cost make such approach not suitable for location based services in a spatial database. For example, in [22] the distance between query point and each point of interest (POI) must both be transferred and then computed on the client. The reason is the loss of spatial information via encryption. On the other hand, order preserving encryption is an encryption system where the order of clear text values are also preserved between their encrypted correspondents. Details for the order preserving encryption system (OPES) is discussed in chapter 3.

2.2.3. PIR Based Approaches

These are the techniques for the well-known problem of Private Information Retrieval (PIR). The aim of the PIR is to get the ith record from an untrusted database size of n, without revealing the i to the database. Therefore, these approaches require a private spatial indexes on top of PIR operations for efficient spatial query processing, while the PIR scheme guarantees privacy. The PIR methods could be divided as hardware-based [23] and computational PIR [24] protocols. In general, they partition the location space in order to index the location data. They proposed various one dimensional or two dimensional portioning techniques to execute nearest neighbor queries. For instance; [24] proposed varonoi diagrams to support exact nearest neighbor queries. Like cryptographic approaches, PIR based approaches do not suffer from the privacy leaks of anonymization or transformation based approaches. Furthermore, PIR based techniques do not have extreme communication and computation costs such as cryptographic based techniques. However, PIR based techniques are just applicable to small set of static queries and could not support dynamic queries on dynamic objects. More over to this, existing PIR protocols are still expensive in terms of computation and communication costs. They require significant amount of resources. It is argued that sending the entire database to the client is more efficient when compared to the cost of

privately retrieving items from the database using existing PIR techniques [3].

2.2.4. Transformation Based Approaches

Transformation based approaches transforms the queries and data in order to prevent the untrusted database server from learning location information of the user. In [4], Hilbert space filling curves are utilized as one way transformations. Both the users and POIs are encoded using this transformation. Therefore, the queries are evaluated in the transformed space. Database server provides the transformed query results to the users and than users reverse the transformation efficiently using a trapdoor. This trapdoor is information is only known by the user and hidden from the server.

A recent work [25] proposed a framework called SpaceTwist. They blind the untrusted database server by incrementally retrieving POIs from a fake location which is called anchor point. [26, 27] are the other works that use transformation techniques to provide data location privacy but they do not protect the location of the users. Transformation based approaches do not require a trusted third party for the query processing such as anonymization and cloaking based approaches. Furthermore, they can utilize the existing indexes to use non-privacy aware servers which makes them to readily applied to existing location servers. The main challenge in transformation based techniques is maintaining the distance properties while preventing reverse transformations by the server or an attacker. For this reason, we have proposed a novel transformation approach to encode the space by utilizing square spirals as one way transformations.

3. PRELIMINARIES

In this section we will provide the basic information about the data structures and methods that are used in the proposed system. Furthermore, we evaluate the complexity of brute force attacks to find reverse transformation in square spiral encoding.

3.1. R Tree Index

R-tree based index structure, proposed in [28], is preferable for the good query performance for 2-dimensional static data. R-trees are similar to B-trees as a tree data structure, but R-trees are designed to index multi dimensional data. For this reason, R-trees are mostly used in spatial databases for indexing two-dimensional information, (X, Y) coordinates of geographical data. Each node of an R-tree could have a variable number of entries unless the number of entries are not exceeded the maximum number of entries. For each node, the maximum number of entries is same and pre-defined. Each child node splits the two dimensional space with possibly overlapping and hierarchically nested rectangles. Two pieces of data is stored within each non-leaf node. The first piece is a way of identifying a child node, and the second piece is a bounding rectangle of all entries within this child node [29]. But in leaf nodes, actual data element and the bounding rectangle of the data element are stored.

The insertion and deletion algorithms use the bounding rectangles to ensure that neighbor elements are in the same leaf node. Thus, a new element usually will go into the leaf node that requires the least enlargement in its bounding rectangle. Similarly, the search algorithms such as intersection or containment use the bounding rectangles to decide whether or not to search inside a child node. With this approach, most of the nodes in the tree are never traversed during a search. Even if the index is disk-resident and nodes correspond to disk pages, only visiting a small number of nodes is sufficient due to design of the index structure. This makes R-trees as suitable as B-trees for databases [30].

The R-tree index is completely dynamic; so inserts and deletes can be intermixed with searches and periodic re-organization is not required [30]. A spatial database contains collection of tuples representing spatial objects, and each tuple has a unique identifier to retrieve quickly when needed. Leaf nodes in an R-tree has index record entries of the form (I, tuple-identifi er) where I is an 2-dimensional rectangle which is the bounding box of the spatial object indexed and tuple-identifier refers to a tuple in the spatial database [28]. However, non leaf nodes in an R-tree has index record entries of the form (I, child-pointer) where child-pointer is a way of identifying address of a lower node in the R-tree and I represents a bounding box that covers all child bounding boxes in the lower nodes entries. In fact, the R-trees could be used for more than 2-dimensional data. In this case, each dimension I_i is a closed bounded interval [a,b] represents the extend of the object along dimension i. Since we use the R-tree in spatial databases, we focus on the 2-dimensional data stored in the R-tree.

Let M is the maximum number of entries that will fit in an R-tree node and let m (M / 2) is the minimum number of entries in an R-tree node. In this case, the Rtree satisfies the following characteristic [28]:

- Every leaf node except root node contains between m and M number of index records.
- For each index record (I, tuple-identifier) in a leaf node, I is the smallest rectangle that geometrically contains the spatial data object represented by the tuple identifier.
- Every non-leaf node except the root node contains between m and M children record.
- For each entry (I, child-pointer) in a non-leaf node, I is the smallest rectangle that geometrically contains the rectangles of the child nodes.
- The root node contains at least two children if it is not a leaf node.
- All leaves appears on the same level due to balanced property of the R-tree

The Figure 3.1 illustrates the R-tree indexing structure for spatial databases:

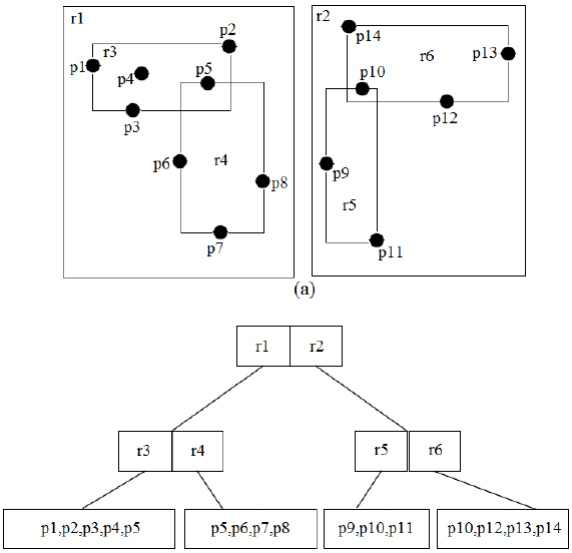

Figure 3.1. R-tree index structure (a) Geometric layout (b) R-tree index

In order to improve the query performance of an R-tree, the rectangle boundaries should be kept as small as possible. In literature, these small rectangles are called Minimum Bounding Rectangles or MBRs. Unlike B-Tree, R-tree allows overlaps between MBRs of nodes at the same level. Therefore, several paths may need to be traversed during searches. In our storage structure, distance squares R-tree index, which is a variant of R-tree, is used with making slight changes to adapt our assignment problem.

3.2. Order Preserved Encryption System

Order preserved encryption (OPES) is an encryption schema that preserves order of the values. The performance degradation of the cryptographic techniques could be reduced with order preserving attribute of the OPES. For example, the aggregate queries such as MIN, MAX could be executed over order preserved encrypted data without the need of decryption. Because, the order of the encrypted values and the order of the clear text values are the same. Therefore, the database just compares the encrypted values to answer the query. This class of encryption techniques are called order preserving encryption [31].

In order preserving encryption proposed by [31], there are three main phases which are Model Phase, Flatten Phase and Transform Phase. In Model phase, the data is

grouped into the buckets after the bucket boundarie s and the distributions are determined. Determining of the bucket boundaries is completed in two phases that are the growth phase and the prune phase. In the growth phase, large buckets are divided into smaller sub buckets until the number of objects in a bucket is below some predefined threshold. In the prune phase, algorithm tries to combine some buckets in order to decrease total cost of describing the system. The next main phase is the flatten phase. In Flatten phase, the plain text buckets is mapped into buckets in flatten space. Mapping function and scale factor are the parameters to flatten the data during flatten phase. The parameters for each bucket forms the key of the order preserved encryption. The last phase is the transform phase in which flattened buckets are transforme d into the target distribution. The target distribution is determined by the user according to encryption requirements. After the encryption, the order is preserved and data is sorted by OPES [31].

4. PROPOSED SYSTEM FOR OPTIMAL ASSIGNMENT

In literature, there are numbers of location privacy aware approaches in outsourced databases for well-known spatial query types. These approaches could be grouped as anonymity based, cryptography based, PIR based and transformation based. Cryptographic and PIR based techniques have high communication and computation costs and in general they are limited with small set of query types. On the other hand, anonymity based techniques require a trusted intermediary and the trade privacy with query accuracy. All these techniques are studied to perform well-known query types for spatial outsourced databases such as range queries and K-nearest neighbor (KNN) queries. However these well-known query types are local and static where POIs are mostly chosen as static and immutable objects and query objects are independent from each other in order to use benefits of pre-calculation.

In this thesis, we focus on a different query type to solve the capacity and coverage constrained optimal assignment problem in outsourced databases. In an assignment query, there is a point set Q of query objects (e.g. people looking for a park spot) and a point set D of data objects (e.g. park spots), where each $q \in Q$ has a capacity C_q and each $d \in D$ has a capacity C_d. The task is to minimize the assignment cost and maximize the assigned capacity subject to capacity and coverage constraints. This class of queries should consider whole system since a small local change could have large effects on the result set. In the optimal assignment, query objects generate the result set together. This means any change in one of the query object could return a different result set. For this reason, pre-calculation is not feasible for optimal assignment query in dynamic systems.

Due to properties of this query type, existing location privacy aware approaches could not be used. Thus, we proposed a novel transformation based technique to perform capacity and coverage constrained optimal assignment query in outsourced databases. In this technique, the location privacy of both query and data objects could be preserved and query is resolved on the transformed data. Furthermore, this technique does not

require any trusted third party so it reduces the system implementation, maintenance costs and prevents possibilit y of security vulnerabilities on the trusted intermediaries.

4.1. Assignment Query

In this section, we will describe the problem definition for the proposed assignment query. Different from [7], we did not use the model where service data objects have infinite coverage. Limited coverage is more realistic especially for the location based services (e.g. a service to find a free park spot in $1km^2$). In this work, we defined the cost function as Euclidean distance between the query and data objects but other cost functions could be used as well.

Problem Definition: set of query objects Q and service data objects D. The cost of assigning q to d is $COST_{qd}$ and this is proportional to Euclidean distance between q and d. Coverage radius of a service data object d is R . Capacity of a query object q is C_q and capacity of a service data object d is C_d .

Given:

- Q , D : set of query objects and service data objects
- C_q , C_d : Capacity of a query object q and service data object d
- R : Maximum coverage radius for the service data object

Calculated:

- $COST_{qd}$: cost of assigning query object q to data object d.
- C_{qd} : assigned capacity from q to d.
- A : set of assigned pairs

Objective:

Minimize the total cost, $\sum_{\forall q,d \in A} C_{qd} COST_{qd}$

Maximize the assigned capacity, $\sum_{\forall q,d \in A} C_{qd}$

Subject To:

$\sum_{q \in N} C_{qd} \leq C_q \qquad \forall d = 1, \ldots, n, \qquad (1)$

$\sum_{d \in N} C_{qd} \leq C_d \qquad \forall q = 1, \ldots, n, \qquad (2)$

$C_{qd} \geq 0 \qquad (3)$

$COST_R \geq COST_{qd} \geq 0 \qquad (4)$

$R > 0 \qquad (5)$

$C_{qd} \leq \min(C_q, C_d) \qquad (6)$

There are several algorithms proposed to solve capacity constrained assignment problem. This problem is also a variant of minimum cost flow problem which contains only integer flows. Due to coverage constraint, the bipartite graph becomes a sparse graph. The solution of this problem is the answer of the capacity constrained assignment query which we call assignment query in rest of the paper.

At the end, the proposed assignment query for outsource databases formulated as follows:

$$ASSIGN(Q : set\,of\,query\,objects,\ A : output)$$

where, Q is an array structure that keeps the encrypted location, identity values with coverage region of each client. For identity privacy the clients may use random id numbers or fake id numbers as long as they are unique in the system, A represents the assigned pairs which is the result of the query. The assignment query does not take inputs like service provider identity values, coordinates of the service point, or

costs of the paths between clients and service points. Because, we have utilized the outsourced spatial database to determine this kind of information. First of all, we find the candidate service points and their capacities from our dynamically updated database and then we calculate the assignment cost for each candidate assignment pair. In the following chapters, candidate service point search and cost computation processes are presented.

4.2. Approximate Method

In this section we highlight the proposed transformation technique. First of all, we will give a brief introduction and then we will describe the square spiral space encoding. Implementation details are explained in the following subsection.

We proposed a novel privacy aware transformation technique. This technique can be used for wide variety of queries including global and dynamic queries such as optimal assignment. In an assignment query, there are two set of objects. The first set of objects are the query objects that requests a location based service, e.g. people search for an empty park place, or call for an ambulance in an emergency case. On the other hand, there is another set of data objects which are dedicated for these location based services such as empty park places, vehicles like taxi cabs or ambulances.

Approximate distances could be calculated blindly in this transformation technique. Unlike other distance preserving transformation techniques [4] , it does not require any trusted component such as anonymizers or agents. Furthermore, the proposed transformation method uses one way function and provide more security than the traditional distance preserving methods which could be reverse transformed.

4.2.1. Square Spiral Encoding

In this section, we first review one way functions and important class of many-to-one dimensional mappings. After that, we highlight the square spirals which are used in our approach to achieve location privacy in outsourced databases.

A transformation is one way if the data can be easily transformed in single direction and it can not be reverse transformed in a reasonable time without any additional information [5] . Encryption is an example for the one way function. In order to make the decryption fast, we need a decryption key. For one way data transformation this additional information required for reverse transformation called trapdoor [5] . In general, many one-way transformations are reversible without the knowledge of the trapdoor using brute force attacks. But in this case, the reverse of the one way function should be complex enough in order to make such transformation computationally secure.

In [5] , they transformed the two dimensional data into one dimensional Hilbert values in order to perform secure KNN queries. Even if their method gets the approximate results, the query efficiency is significant. According to [32] , there are many one-way transformations that could be applied to a 2-D space of objects (e.g., random perturbation of points), but, the majority of such transformations do not respect the notion of distance and proximity. For this reason, identifying the right space encoders is very challenging. [32] use the Hilbert space filling curves due to their significant clustering capabilities however their proposed transformation function does not apply queries other than nearest neighbor and KNN queries.

In this work, we identify a 2-dimension to 1-dimension mapping function and we define the parameters of our mapping function as the trapdoor. So the query results are decrypted quickly by the query object or client. Such trapdoor will be only provided to the clients in order to reverse the encoded query results and get the response set back in its original format. The data objects use the transformation function to report their locations to the server and query objects use this transformation method to encode the queries to hide the location of the query object from the database server. Unlike the Hilbert transformation our method preserves the distance between encoded points.

In Mathematics, spirals are defined as curves which starts from a center and getting progressively farther away as it revolves around the center. Square spirals are a special form of the spirals where it starts from inside a square and traverse all the squares before getting farther away. The most interesting feature of square spirals is how they

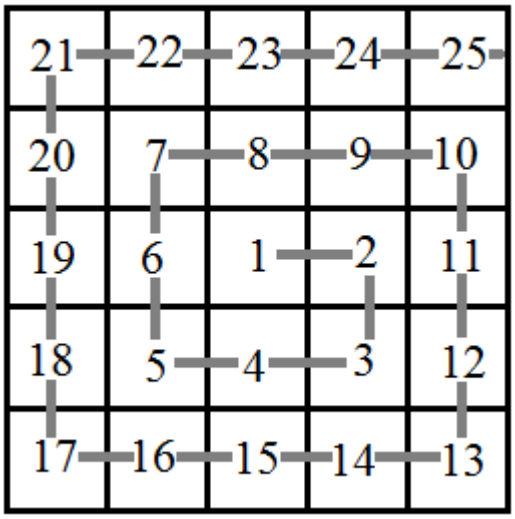

Figure 4.1. Square Spiral Space Encoding

can preserve the distance after transformations with satisfactory values when used in location-based services. This property suits our problem well as we are interested to address the assignment query in a transformed space. Furthermore, square spiral transformation is a one way function if the square spiral parameters are not known. These parameters, which collectively form a key for this one-way transformation , include the square spirals starting point (X_0 ; Y_0), square orientation θ , square spiral direction SD and square spiral scale factor Γ . We term this key, Spiral Decryption Key or SDK where SDK = { X_0 , Y_0 , θ , SD , Γ }.

Therefore a malicious entity, not knowing this key, has to exhaustively check for all combinations of square spiral parameters to find the right square spiral by comparing the square spiral values (SS-values) for all points of interests. As we show in below, we make it computationally impossible to reverse the transformation and get back the original points. Even a small error in approximating curve parameters will generate a completely different set of SS-values. We now prove this important property of our approach which makes more sense on the security of our proposed method.

Theorem 4.1. *The complexity of a brute-force attack to find the transformation key of the square spiral is O(2^{4p}) where p is the number of bits used to discretisize each parameter.*

Proof. The theorem above is proved for Hilbert curve transformation method by [5]. We use the similar method to prove this for square spiral transformation. In order to accurately find the square spiral's starting point, it should exactly lie on the intersection of two lines coming from each of the X and Y axes. Therefore exact values of both X_0 and Y_0 has to be determined in the continuous domain of X and Y axes. In theory, finding the right value for the starting point in a continuous space is impossible. But, in practice, the attacker could approximate X_0 and Y_0 by constructing the nest grid possible to guarantee that his best guess $(X_0^{'},\ Y_0^{'})$ located at an intersection of two edges, lies very close to (X_0, Y_0) so that $|X_0 - X_0^{'}| \leq \epsilon$ and $|Y_0 - Y_0^{'}| \leq \epsilon$,When ϵ is sufficiently small then replacing (X_0, Y_0) with $(X_0^{'}, Y_0^{'})$ generates a set of SS-values in differentiable from the original set. The attacker should thus search the entire space exhaustively for a very close approximation of this starting point. Using p bits the attacker can generate 2^p candidate values on each axis. Therefore, assuming a square region covering all POIs, the attacker's entire search space for the starting point will have 2^p * 2^p elements. Similarly, the entire continuous 360± space for θ should be discretized to the nest possible extent to ensure that $|\theta - \theta^{'}| \leq \epsilon$ for at least one value of $\theta^{'}$. With q bits, that attacker can generate 2^q different candidate values of θ each corresponding to a square spiral orientation. The square spiral scale factor Γ is a continuous number between 0 and 1 and thus similarly, r bits can divide the 0 to 1 range into 2^r values each can approximate Γ so that $|\Gamma - \Gamma^{'}| \leq \epsilon$ for at least one value of $\Gamma^{'}$. Assuming N different possibilities for the square spiral direction, the entire solution space will have 2^p * 2^p * 2^q * 2^r * N elements. Assuming $2^q = \mathrm{O}(2^p)$ and $2^r = \mathrm{O}(2^p)$ and since N << 2^p , the complexity of an exhaustive search is $\mathrm{O}(2^{4p})$ where p is the number of bits used by the attacker to represent each parameter. □

Theorem 4.2. *The difference between the largest SS-value of a level with the largest SS-value of the one level below is 8*level*

Proof. The largest SS-value of the square spiral can be formulated as

$$(2n+1)^2 = 4n^2 + 4n + 1 \quad \forall n = 0, 1, 2, ...$$

F(2)		G(2)		H(2)
	F(1)	G(1)	H(1)	
E(2)	E(1)	A(0)	A(1)	A(2)
	D(1)	C(1)	B(1)	
D(2)		C(2)		B(2)

Figure 4.2. Square Spiral Functions

Table 4.1. Table of Square Spiral Functions

Function List
$A(x) = 4x^2 - 3x + 1$
$B(x) = 4x^2 - 2x + 1$
$C(x) = 4x^2 - x + 1$
$D(x) = 4x^2 + 1$
$E(x) = 4x^2 + x + 1$
$F(x) = 4x^2 + 2x + 1$
$G(x) = 4x^2 + 3x + 1$
$H(x) = 4x^2 + 4x + 1$

The largest SS-value in the lower level is calculated for $n = n - 1$ and the difference gives the result.

$$(2n+1)^2 - (2(n-1)+1)^2 = (4n^2 + 4n + 1) - (4n^2 - 4n + 1)$$

As clearly seen the difference between the largest SS-values between levels are $8n$. □

This is used in the algorithm to calculate the distance to the center of square spiral and angle with the center point for any given SS-value.

4.2.2. Square Spiral Algorithm Phases

First of all, the square spiral parameters are determined by the LBS provider and distributed to all the clients via secure communication channel. The complexity of a brute-force attack to find the transformation key of the square spiral is $O(2^{4p})$ where p is the number of bits used to discretisize each parameter. The square spiral encryption is strong enough but these parameters could be updated regularly to increase complexity and to prevent brute force attacks. When clients get these parameters, they start to construct the square spiral from (X_0, Y_0) and traverse all region that service is provided. After constructed the square spiral, each visited point could be encrypted using the encryption function which maps points to SS-values (SS-value) = E(P_x, P_y). Similarly, any received data could be decrypted using the decryption function which returns the geometric center of the region that is represented by the SS-value (P_x, P_y) = D(SS-value). The construction of square spiral is the offline phase and clients do not have to store all the square spiral structure in the memory. After the construction of the square spiral they can just load the mapping table for nearby regions. Clients can use this square spiral structure unless square spiral encoding parameters are updated by the LBS provider.

The outsourced database does not have any information about the square spiral parameters. The encryption and decryption operations could be done only in the clients using square spiral parameters. The SS-values are stored in the database and the size of the DB is not dependent to the region where the clients are located. It is only dependent to the number of POIs or data objects. The SS-values of the data objects are stored in a B+ like tree due to efficient insertion, retrieval and removal capabilities. The difference is their keys. In this structure all keys are intervals and parent node intervals are minimum bounding intervals of the child nodes. We call these intervals MBI, which stands for minimum bounding intervals.

At first, database determines the POIs in the coverage region of the client by searching B+ tree index. After this, database gets the radius and angle to the center of the spiral for POIs and query point if they are not already calculated. Because the

calculated values are stored in the database and addressed with B+ tree index to prevent redundant computations. For any given SS-value the distance to the center of square spiral and angle with the center point is calculated using the algorithm below.

```
calculateRadiusAndAngle(x=SS-value)
{
   if ( x ≤ 1 ) then return (0,0);
   y:=1
   radius:=1
   while x > y
   {
      y += 8 * radius;
      radius++;
   }
   angle = 45 - (y-x)*(45/radius);
   return(radius,angle);
}
```

Figure 4.3. Algorithm to find radius and angle of SS-values

The assignment cost computation for each candidate pairs are explained in the next chapter. When we get the assignment costs, client and service points encrypted locations and identities then the Auction algorithm computes the optimal assignment in parallel. At the end, assigned pairs with SS-values are listed. Each client receives corresponding service point identity and SS-value. SS-values are decrypted in the client by using square spiral parameters as trapdoor. Finally clients get the location of the optimal service point without revealing their locations to the untrusted server.

4.2.3. Cost Computation

Different cost computations algorithms could be used in this assignment query but in our implementation we used the Euclidean distance between query object and data

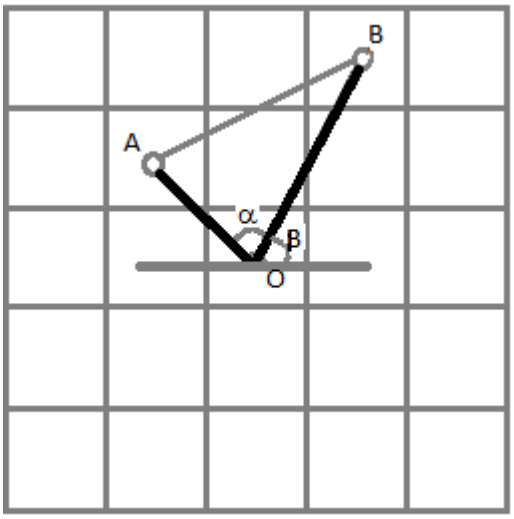

Figure 4.4. Distance Calculation

object as a cost function. For this reason, the distance between two SS-value should be calculated. In a triangle, if we know the length of two lines and the angle between them then we can calculate the length of the third line using Cosine theorem. We used to same formula to calculate the distances between two points. When we only know the radius and angle of a point we can calculate the distance. In square spirals the maximum distances are reached at corners when alpha is 45^o , 135^o , 225^o and 315^o .

If $|tan(A)| \leq 1$ then
  length $= |(\frac{R}{\cos(A)})|$
else
  length $= |(\frac{R}{\sin(A)})|$

Figure 4.5. Algorithm to find the distance to the center

Distance between points are calculated with Cosine Theorem

$$Distance(A,B) = \sqrt{(\frac{r1}{\cos(A)})^2 + (\frac{r2}{\cos(B)})^2 - 2(\frac{r1}{\cos(A)})(\frac{r2}{\cos(B)})(\cos(|A-B|))}$$

4.2.4. Approximation

The approximation is just done in the transformation phase. The scale of the smallest square could be changed according to the square spiral parameters. The assignment algorithm always use the center of the square in distance calculations. The upper bound could be calculated for this approximation. [8] has calculated the upper bound of the error in fully connected assignment graphs where assignment leaves unassigned query objects if every data objects are fully utilized. In our case, we consider the service coverage distance so it means the number of assigned pairs is at most

$$A \leq \min \left\{ \sum_{q \in Q} q.k, \sum_{d \in D} d.k \right\} = \gamma$$

Lets assume that M_{opt} is the optimum assignment when all points are in the exact positions. For all assigned pairs the assignment cost could increase at most $\gamma.\delta$ so we can write the following equation.

$$M^{'} \leq M_{opt} + \gamma.\delta$$

But $M^{'}$ may not be the optimal matching for $Q^{'}$ and $D^{'}$ so we can write the following equation

$$M^{'}_{opt} \leq M^{'} \leq M_{opt} + \gamma.\delta$$

When we replace all the points to the center of the small squares then for all assigned pairs the assignment cost could increase at most $\gamma.\delta$ so we can write the following equation.

$$M \leq M_{opt} + 2.\gamma.\delta$$

As a result, the upper bound of the assignment error is bounded by $2.\gamma.\delta$

4.3. Assignment Algorithm

The coverage region for the services could be in different shapes or could be circular. For ease of computation and presentation we used the square regions to define the coverage regions. We say that query object is covered by the data object, if the query object is in the square region centered in data object and radius is equal or smaller than the coverage distance. In real world, the coverage distances are limited so bipartite matching graph turns into a sparse graph.

There are several algorithms proposed for capacity constrained assignment problems but auction algorithm is the most efficient algorithm for sparse flow networks. Furthermore, it allows parallel or distributed execution. For these reasons, we modified the Auction algorithm to perform the optimal assignment query.

Assignment Algorithm:

This algorithms has two phases. First phase is the preparation phase for the parallel auction algorithm and second phase is the parallel capacity constrained auction algorithm. The auction algorithm does not use any approximation and returns the exact result with average performance = $O(NA \log N)$ [10] where A is the number of assigned pairs and N is the number of bidders.

Assignment Preparation Phase

- Eliminate all query and data objects with no capacity.
- Each process calculates the assignment costs for the given query object to each data object in the coverage
- Eliminate query objects that are not covered by any data object.
- Eliminate data objects that have not cover any query object.
- Initialize the price of each data object
- Put each query object to the queue

Bidding Phase

- While query object queue is not empty
- do
- If there is no capacity left in the query object then eliminate the query object
- If there is no data object to be assigned (due to high prices) then eliminate the query object
- Each query object makes bids for each capacity to max benefit data object in the coverage area
- If data object capacity is not full
- Add bid of the query object to the data object winners queue
- Decrement the capacity of the query object
- If data object capacity is full
- Add bid of the query object to the data object winners queue
- Set the data object price to the lowest bid in the queue
- Decrement the query object capacity and Increment the price with ϵ
- If data object capacity limit is exceeded
- Add bid of the query object to the data object winners queue
- Remove the lowest bid from the winners queue
- Set the data object price to the lowest bid in the queue
- Decrement the capacity of the query object and increment the price with ϵ
- Re-add the query object of the lowest bid to query object queue and increment the query object capacity
- loop

The described auction algorithm is an overview for the synchronous Jacobi paralleled version [16] of the auction algorithm with slight modifications and pruning strategies. The original algorithm maximize the assignment cost but by reversing the costs and prices, we changed the algorithm to minimize the assignment cost. Actually, the parallel auction algorithm is a black box for us. For this reason, we will not go into the details of the auction algorithm.

Table 4.2. Data objects table in the outsourced database

Object ID	OPES(X)	OPES(Y)
23	ff4234	34ee3
46	abcd5	920aa
45	fd544	cd234

4.4. Exact Method

Order preserved encryption (OPES) [31] is an encryption schema that preserves order of the values. It allows to directly process comparisons over encrypted data. So equality, range, MAX, MIN, and COUNT queries can be directly processed without decryption [31]. However difference between two encrypted values could not be calculated. In a spatial database, the difference between location of the two points gives us the distance. For a location based service, cost computation generally uses distance value with some coefficients. However, there is no proposal for calculating distance over order preserved encrypted data.

In our proposed system, the coordinates are encrypted separately via OPES. We could use the following comparison operators $<, =, >$ over the order preserved encrypted data. For this reason we store encrypted bounding corners in a look up table with their distances from the node. In our problem, assignment operation is constrained by coverage distance so we do not need to know distances more than the coverage distance. Choosing static objects to store encrypted distances will decrease the update costs significantly.

Optimum assignment of empty park spots to vehicles could be a good example for this technique. In this example, park spots are static but drivers looking for a park spots and available capacity of the park spots are dynamic. Beside the encrypted location data, static park spots need to send encrypted look up table for x and y coordinates according to distances. The idea behind this approach is calculating the distance by using order preserving property of the encryption and the look up table.

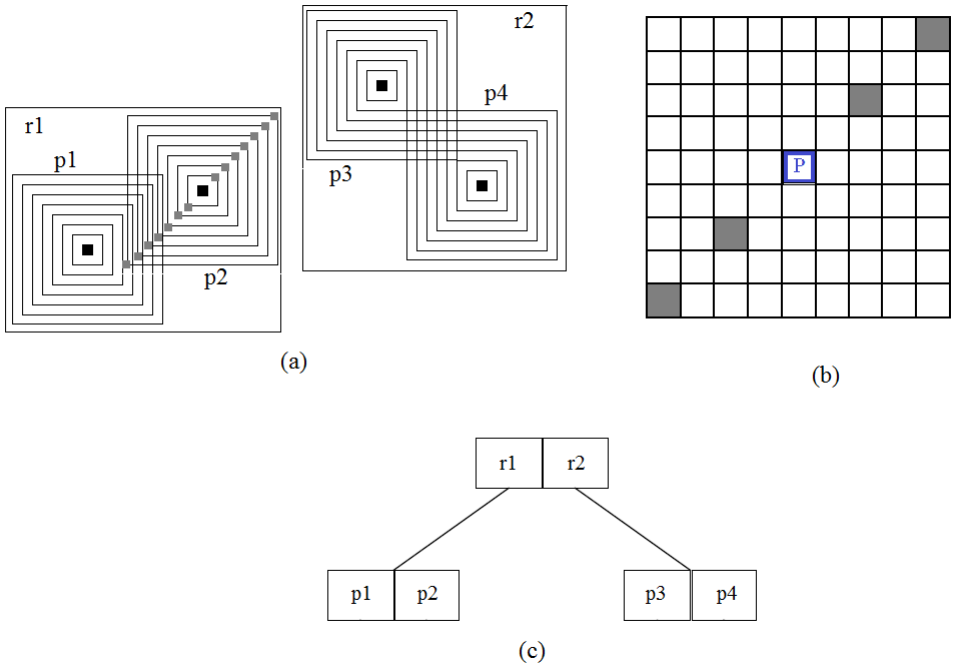

Figure 4.6. R-tree index for encrypted coordinates.(a) Geometric layout of R-tree (b) Close look to an MBR (c) R-tree index

First of all, system will check the location of the query object to determine if it is in the coverage of any data object. After getting list of the data objects that are in the coverage, it will compare the location of the query object to the values in the look up table. Since the query object is in the coverage, it could find the distance to the data object. The following table illustrates the database table where the location of the data objects stored as encrypted.

Data objects and their maximum coverage distances is bounded with MBRs and indexed by R-trees. The data objects which covers the query object are easily found via R-tree indexes. This means if a query object data is in the region of one or more MBRs in leaf nodes than the query object is in the coverage of those service data objects.

For each MBR there is a linked list used to map distances to encrypted values. After several comparisons, the distance between query and data object could be calculated. These encrypted values are also represented in Figure 4.4 (a) as gray squares. The coverage distance is limited because people do not like to get a park spot which is

far from their location.

```
For (i = coverage distance; i > 0; i − −)
{
    Get the max(X_i) and min(X_i);
    #Compare with query object coordinate X
    if (X_q == max(X_i)) then
        return Distance_i;
    else if (X_q > max(X_i)) then
        return Distance_i + 1;
    if (X_q == min(X_i)) then
        return Distance_i;
    else if (X_q < min(X_i)) then
        return Distance_i + 1;
   else
      continue;
}
```

Figure 4.7. Algorithm to find distance in X-axis over encrypted values

The algorithm to find the distance in Y-axis is similar to the algorithm described above. After we get the distances in both X-axis and Y-axis, simply euclidean distances are calculated using the following formula. $Distance = \sqrt{(X_d - X_q)^2 + (Y_d - Y_q)^2}$. The rest of the approach is the same as square spiral encoding. The costs are calculated and the bipartite graph is determined. The auction algorithm gets these inputs and send the result as ordered preserved encrypted values. These values are send to the clients that send the assignment query. Finally, clients decrypts the encrypted location data using decryption key.

5. COMPARATIVE PERFORMANCE RESULTS

In this section, we evaluate the proposed system. At first, we discuss the proposed data structures and their efficiency. Secondly we will consider the assignment algorithms. At the end, we investigate the comparative performance results and verify that proposed system meets the needs of mobile users.

Table 5.1. Experiment parameters and examined values

Parameter	Examined Values
Number of users (N)	25, 50, **100**, 200, 400 (K)
Number of servers (M)	100, 500, **1000**, 1500, 2000
Distributions	Uni/Uni, Uni/Gau, **Gau/Uni**, Gau/Gau
Standard Deviation (σ)	0.05, 0.1, **0.2**, 0.5, 1
Server Capacity (C_d)	32, 64, **128**, 256, 512
User Capacity (C_q)	**1**, 2, 4, 8
Coverage Radius (R)	2, 5, **10**, 15, 25
Unit Square Spiral Length (Γ)	**1**, 2, 5, 10, 15, 20

In the experiments, we generate sets of mobile users Q (query objects) and stationary servers D (data objects) with different cardinalities as N and M. The geographical coordinates are normalized to $[0, 1000]^2$ space. Table 5.1 presents the parameters and examined values during the experiments. Default settings are written in bold for your notice. We stabilized all the parameters in default settings and measured the effect of a single parameter in each experiment.

In default settings, the location of the users are generated by a Gaussian distribution which has standard deviation σ equal to 0.2 times the shortest path length to the furthest node from the center and mean at the center of the coordinate system. This model can simulate the crowded city center during working hours. M servers are generated by a Uniform distribution with equal server capacities C_d and radius R.

We repeated the simulations for 50 timestamps and just reported average observed values per time stamp. All experiments are executed on an Intel Core 2 Duo P7550 dual-core 2.26 GHz machine with 4GB of memory. For comparison purposes, we have executed multi core simulation also in Intel Zeon L5410 two quad-core 2.33 GHz machine with 8GB of memory.

In our evaluation, we compare the two methods square spiral transformation method and encrypted distance squares method. Both methods use the auction algorithm for the assignment query so the difference between methods are generally their data structures and their methods to calculate the assignment cost between users and servers. As a result, we reported the auction based assignment algorithm results independent from the methods mentioned above.

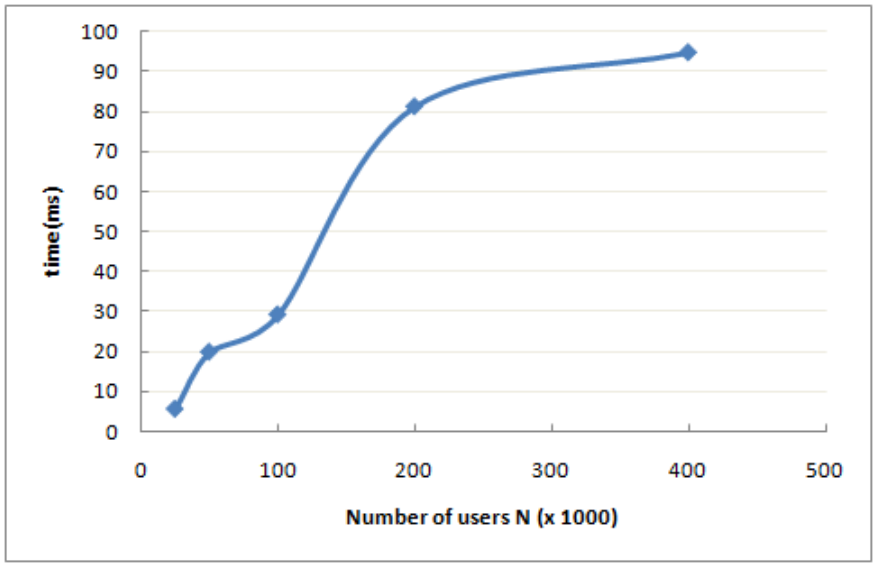

Figure 5.1. CPU time vs. user cardinality N

In Figure 5.1, we measured the CPU time for assignment for various user cardinalities N. The processing time increase with the increase in the number of users however the increase is not linear and slows down because lower M/N ratio implies that most of the assignments are done by locally.

In Figure 5.2, we vary the number of servers M and plot the CPU time graph. When the number of servers increases then the choices of each user increase. Searching for a maximum benefit server in the large candidate list is slower so the CPU time

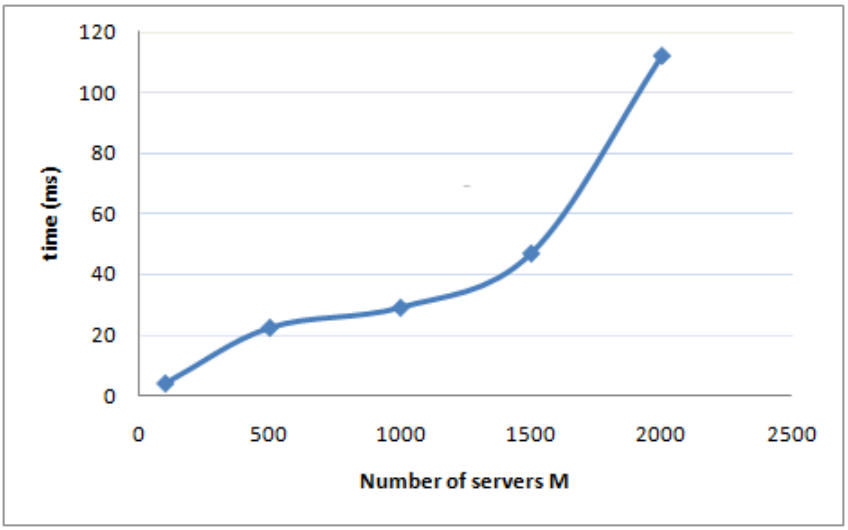

Figure 5.2. CPU time vs. server cardinality M

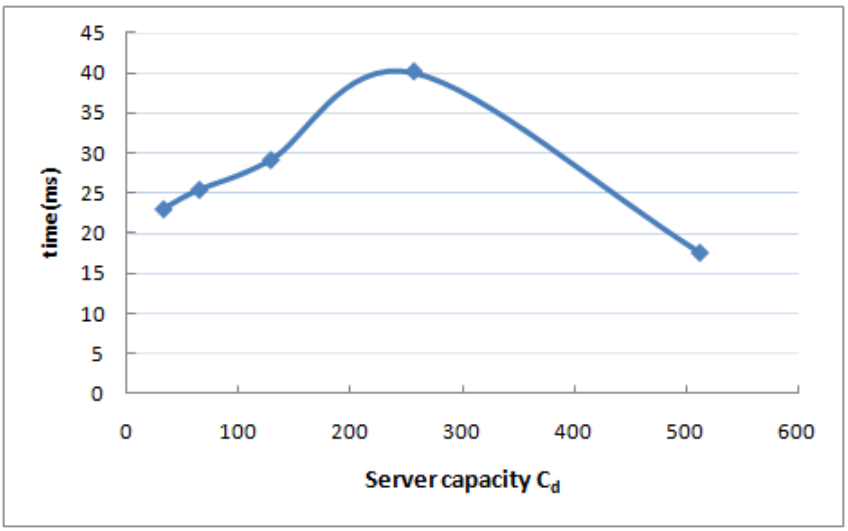

Figure 5.3. CPU time vs. server capacity C_d

increases with the increase in M.

Figure 5.3 measures the effect of the server capacity C_d on the performance. The running time of the Auction algorithm at first increases but then decreases. This is an expected result because if the server capacities are infinite then each user will get the maximum benefit server without any rejection. Therefore, when there are too much available capacity the problem becomes easier. Initially it increases because the server prices do not increase very fast when there are more capacity so each user could afford many bids.

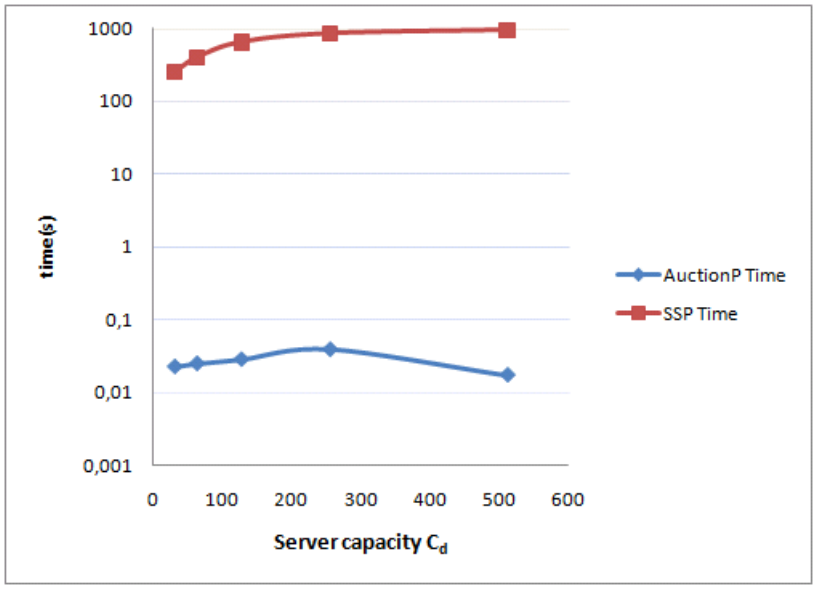

Figure 5.4. CPU time vs. server capacity C_d in logartihmic scale

In Figure 5.4, we compared the CPU time of the Auction algorithm and SSP algorithm in logarithmic scale. Since SSP algorithm has $O(N^3)$ complexity, Auction is much faster than the SSP. Unlike Auction algorithm, SSP CPU time does not decrease for very large capacity values because it requires more augmenting path calculations to complete the assignment.

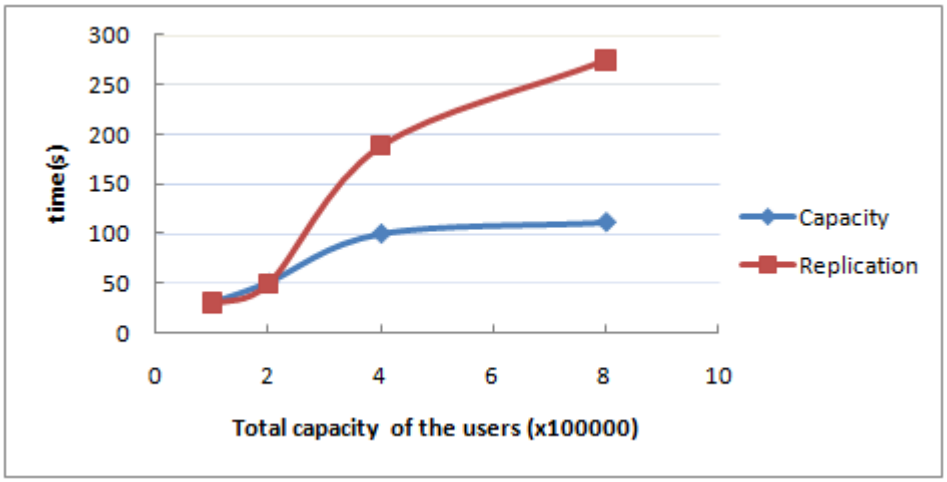

Figure 5.5. CPU time vs. user capacity C_q

In Figure 5.5 investigates the effect of user capacity C_q on the performance. The increase in the user capacity is very similar to the trend in Figure 5.1. Actually the increase in user capacity is very likely the increase in number of users. It is equal to

have multiple users at the same location. Instead of replicating many users, the users in the same location could make the bids together in order to reduce the computation time.

For this experiment, we replicate the 100K Gaussian distribution users to have 200K, 400K and 800K. The aim is here to compare replicated users with users that have more than one capacity. The user capacity increase does not effect the CPU time very much but the increase in number of users increase the time as well.

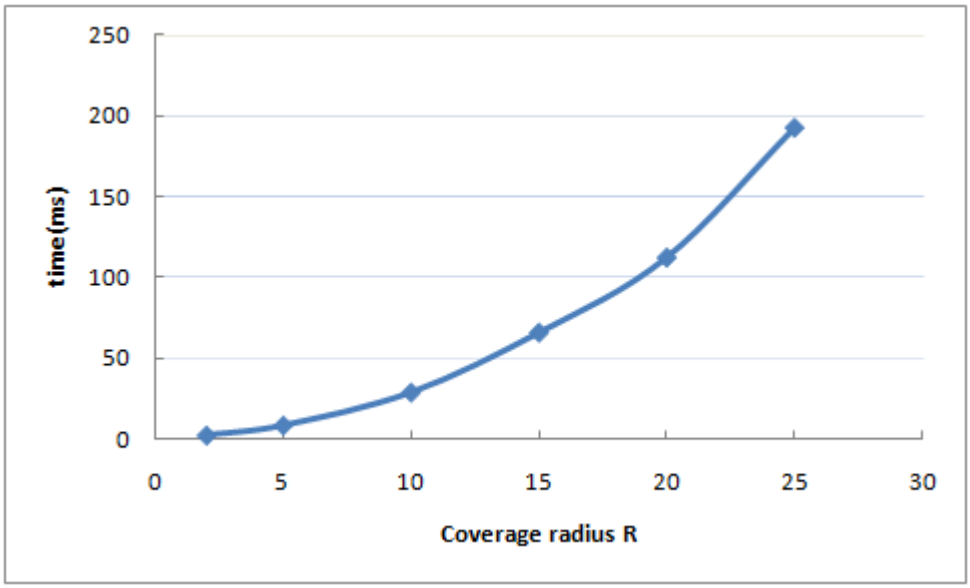

Figure 5.6. CPU time vs. coverage radius R

In Figure 5.6 examines the effect of coverage radius R on the performance. When the coverage radius increases the number of potential matching increases and the problem becomes harder. If the coverage radius approaches to infinity then the problem approaches to the assignment in the fully connected graphs not sparse graphs.

In Figure 5.7 investigates the CPU time versus standard deviation σ parameter of the Gaussian distribution which is used to generate N number of users. The running time is maximum when $\sigma = 0.2$. When σ is small, the users are very skewed and this leaves most servers under utilized so it is easy and faster. But when σ becomes larger than the distribution becomes similar to uniform and users are not overloaded at the center. This leads to more under utilized servers and lower running time. status open

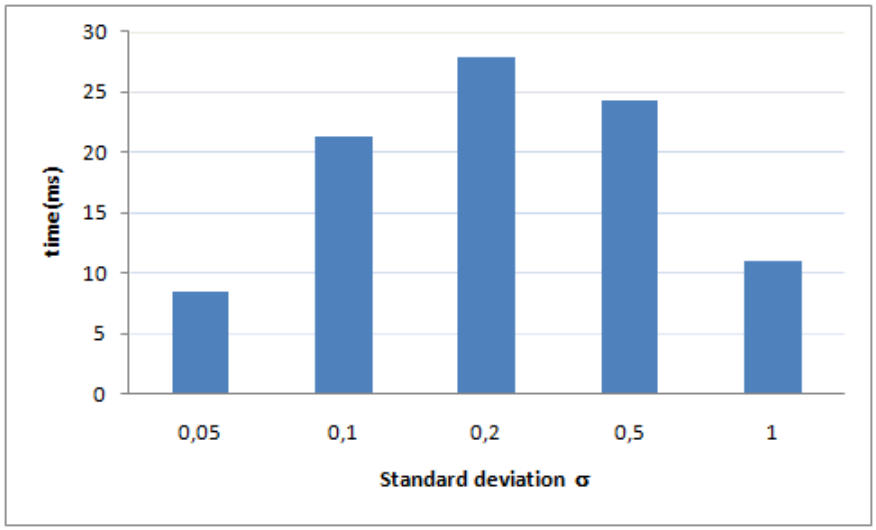

Figure 5.7. CPU time vs. standard deviation σ

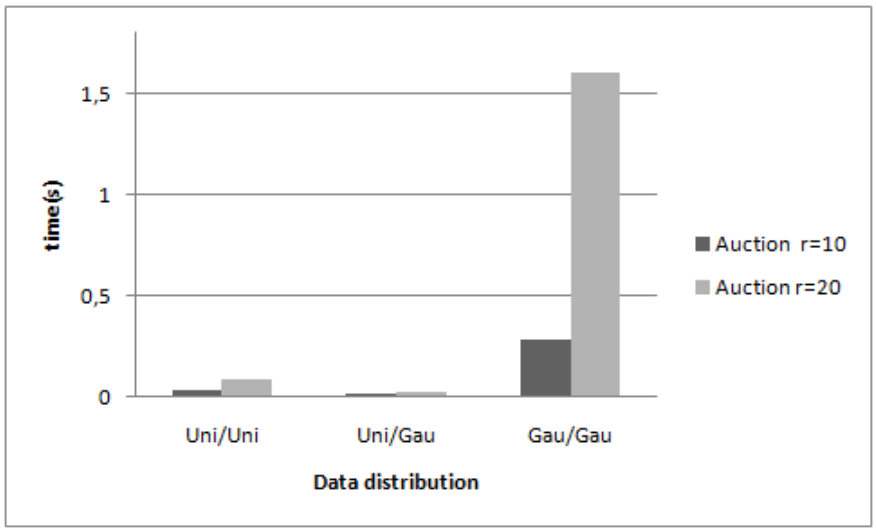

Figure 5.8. CPU time vs. data distributions

Figure 5.8 examines the different distribution combination for users and servers. Since some running times are very low, we added experiments for R = 20 intentionally. In the labels "Gau" stands for Gaussian distribution and "Uni" stands for Uniform distribution.

In Uni/Uni the servers are under utilized because the capacity of servers $C_d = 128$. There will not be price wars between the users. Uni/Gau is more under utilized because in this case servers are located at the center but the users are distributed equally to all the space. Most of the users will not be in the coverage of any server. The matching algorithm is executed just for the small area around the center where there are more

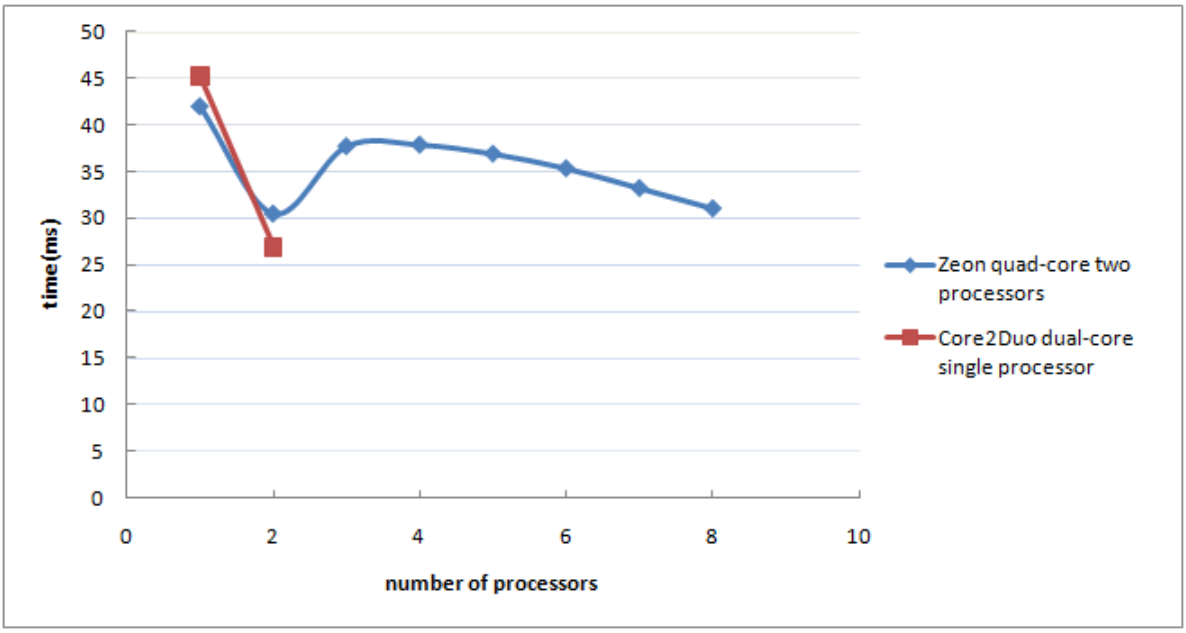

Figure 5.9. CPU time vs. number of processors

servers than users. Gau/Gau distribution maximize the users that are in the coverage which is similar to increasing coverage region to attract more users. Since the problem becomes harder, it takes more time than the others.

In Figure 5.9, we investigated the performance of the algorithm when there are multiple processors. We have done the testing in two machines one of them has a single dual core processor and the other one has two quad-core processors. We enable just a single processor at first and then we increase the maximum number of processors and measure the CPU time of the algorithm. We could set the maximum number of processors but we could not select the cores individually. Zeon has a high performance increase when there are two processors because two cores are selected from the same physical processor so they could use the shared cache. When there are 3 processors, the cache mismatches and context switching increase the running time but it will continue to decrease after that point since the parallelism bonus is more than the synchronization penalty.

In Figure 5.10 examines the memory usage of two different techniques. The first technique square spiral encoding stores the data in a tree structure which is similar to B-tree and the second method encrypted coverage distance squares uses R-tree based

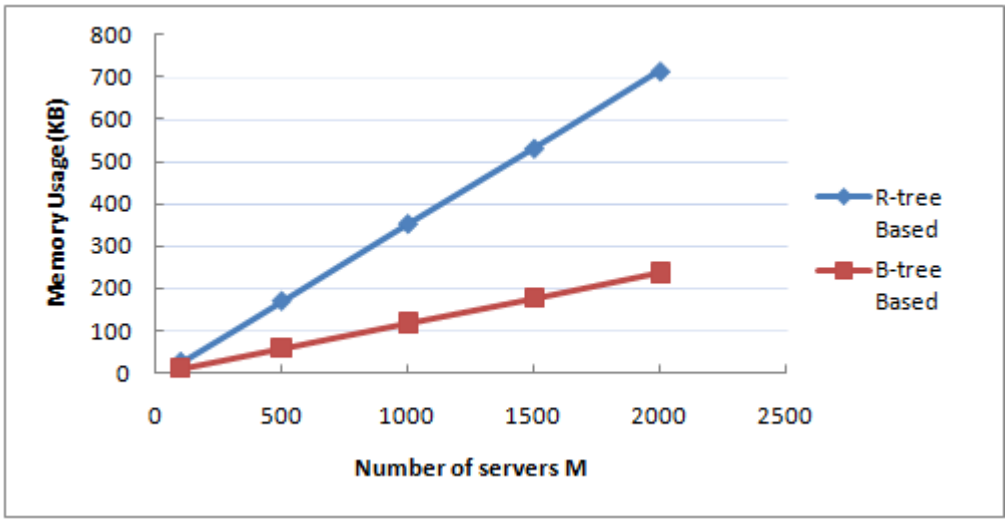

Figure 5.10. Memory usage vs. number of servers M

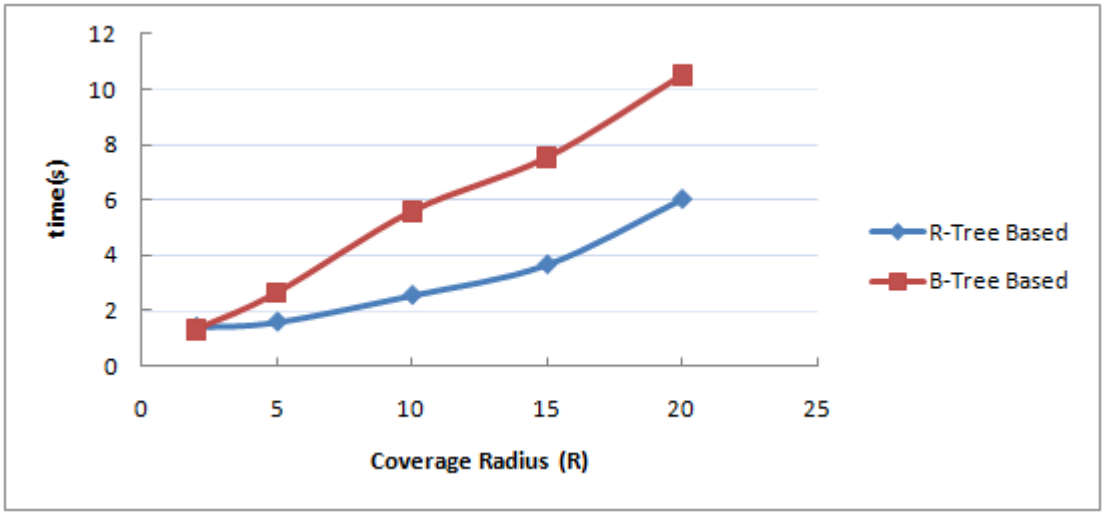

Figure 5.11. CPU time vs. Coverage Radius

structure. Since the second method stores two dimensional data with encrypted distance squares, it requires more memory space.

In Figure 5.11, we compared the performance of the two methods. The CPU time is the time to calculate the assignment costs of N users to M servers with respect to the coverage constraint R via proposed indexes. R-tree based index is more efficient to get servers or POIs in the coverage area. Furthermore, it does not require computations like square spiral encoding and could find the distance by just doing some comparisons.

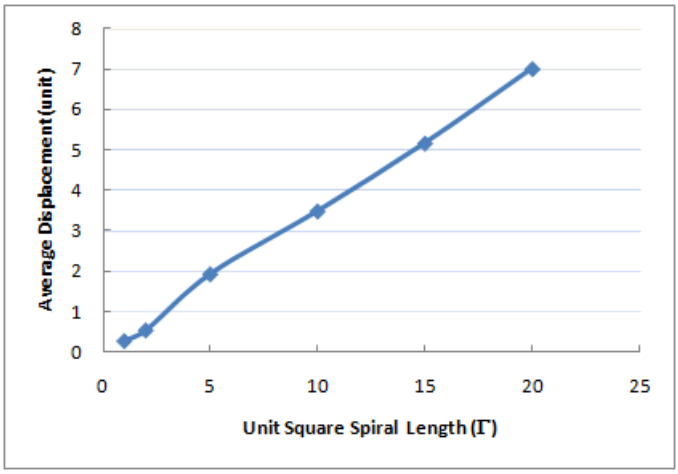

Figure 5.12. Displacement vs. unit square spiral length Γ

In Figure 5.12, we examined the displacement due to the unit square spiral length parameter Γ. The increase in this value will cause the approximation of the user location due to the users real location distance to the square center. In section 4, we have showed the trade-off between the approximation and the result accuracy but here we just calculate the average distance of the users to the center of their square. The displacement increase linearly with the unit square spiral length Γ.

6. CONCLUSIONS

The main challenge of the outsourced databases and cloud systems are the security and privacy. For this reason, many companies try to build their own private cloud instead of using public clouds. Location based services and location privacy becomes more important with the growth of mobile devices that have positioning capabilities. In this thesis, we proposed two novel methods to protect location privacy not only for static local queries but also for dynamic global queries such as assignment query. We also described the implementation details and discussed the comparative results. We focused on the queries where query objects (users) are mobile but data objects (servers) are stationary. Database finds the maximum matching and minimum cost with respect to the coverage and capacity constraints.

The classical assignment algorithms have $O(N^3)$ time complexity so they are not applicable to databases for large data sets. Some of the approximation algorithms promises a linear time complexity but their result accuracy is about 66%. We used the befit of coverage constraint and focused on the sparse graphs. We chose the best exact assignment algorithm for sparse graphs is proposed by Bertsekas [10] runs in average $O(A \log N)$ time complexity. The algorithm finds the maximum weighted matching therefore, we adapted it to find the minimum weighted matching.

Our main contribution is the introduction of the two location privacy aware methods. Both methods do not require any trusted third party or intermediary. The fist method uses the square spiral encoding to provide privacy and to preserve some of the spatial properties. We showed that, the complexity of a brute-force attack to find the transformati on key of the square spiral is O(2^{4p}) where p is the number of bits used to discretisize each parameter. We also describe the algorithms to calculate distance over encoded values and we explain the storage structures. We used the B+ tree due to its high performance on one dimensional data, but we implement the node keys as intervals for better search results. Due to the geometric properties of the square spiral encoding, the coverage regions could be defined as intervals and this increase the search

performance.

The second method uses the advantage of the order preserved encryption schema (OPES). The spatial coordinates are encrypted using OPES so we preserve the order of the encrypted values and we can use the $<, =, >$ operations. We implemented the data storage similar to R-trees but the leaf nodes are the maximum coverage distance rectangle of the object and it also linked to the lower distance rectangles. Therefore, the distance between query object and data object could be found by doing a few comparisons.

We compare the two proposed methods, even if both has different advantages the second method performs better than the square spiral encoding and is able to provide exact results. The disadvantage of these methods is the need of computation in the client side for encoding or encrypting to provide location privacy but it is negligibl e for the new generation mobile devices which provide high computation power.

As a future work, these methods could be adapted to other location based queries such as K-nearest neighbor or range queries. Encryption technique could be enhanced to enable indexing for dynamic objects and other space transformation and encoding methods could be studied for both privacy and efficiency.

APPENDIX A: OVERWIEW OF THE WHOLE SYSTEM

The Figure A.1 is the overview of the whole system for the proposed square spiral encoding method. Data objects transform their location using pre-determined square spiral decryption parameters and send the transformed location and available capacity to the outsourced database. Clients (query objects) transform their location and send the coverage region that they are interested. Database determines the assignment candidate pairs blindly according to coverage region.

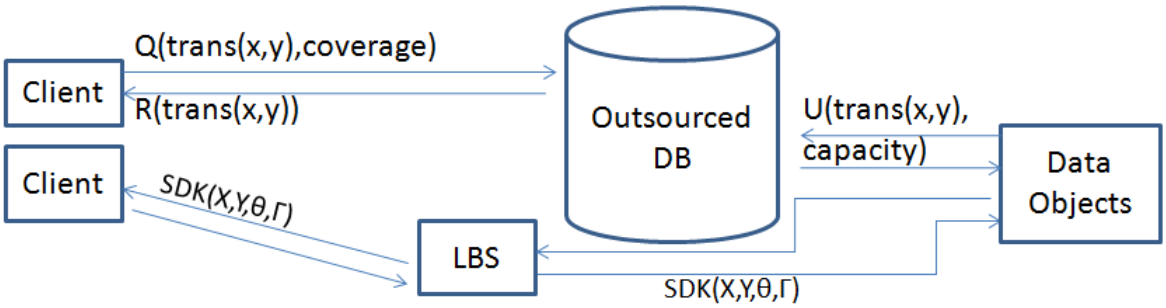

Figure A.1. Square spiral transformation system overview

The following Figure A.2 is the overview of the whole system for the proposed OPES encrypted distance squares method. Data objects encrypt their location using OPES. Furthermore, they calculate and send the bounding points according to the distances for the service coverage area and also they send the available capacity Clients encrypt their location and send the assignment query Database determines the assignment candidate pairs blindly using R-tree index.

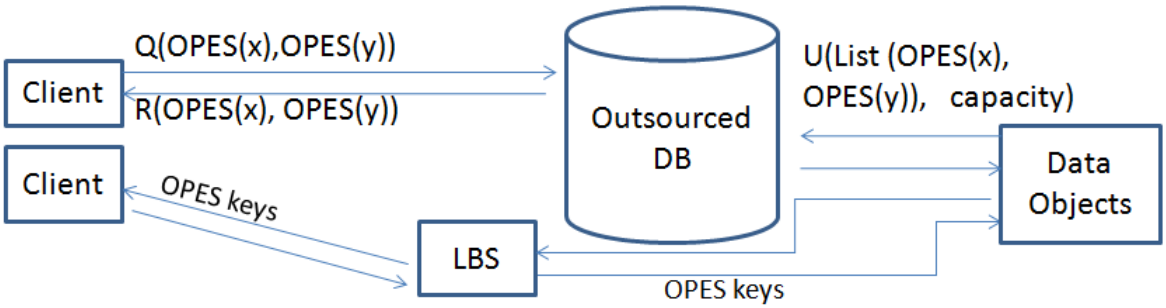

Figure A.2. Encrypted distance squares system overview

REFERENCES

1. Beresford, A. and F. Stajano, "Location privacy in pervasive computing", *IEEE Pervasive Computing Magazine*, Vol. 2, No. 1, pp. 4655, 2003.

2. Kalnis, P., G. Ghinita, K. Mouratidis and D. Papadias, "Preventing location-based identity inference in anonymous spatial queries" . IEEE TKDE, 2007.

3. Sion, R., "On the computational practicality of private information retrieval" , *Proceedings of the Network and Distributed Systems Security Symposium* , 2007. Stony Brook Network Security and Applied Cryptography Lab. Tech. Report 2007.

4. Lin, D., E. Bertino, R. Cheng, and S. Prabhakar, "Location Privacy in Moving Object Environments" , *Transactions on Data Privacy* , Vol. 2, No. 1, pp. 21-46, 2009.

5. Khoshgozaran, A. and C. Shahabi, "Blind Evaluation of Nearest Neighbor Queries Using Space Transformation to Preserve Location Privacy" , SSTD 2007.

6. Konan, A. R., T. . Gndem and M. E. Kaya, "Assignment Query and Its Implementation in Moving Object Databases" , *International Journal of Information Technology & Decision Making* , Vol. 9, No. 3, pp. 349-372, 2010.

7. U, L. H., M. L. Yiu, K. Mouratidis and N. Mamoulis, "Capacity Constrained Assignment in Spatial Databases" , SIGMOD, 2008.

8. U, L. H., K. Mouratidis and N. Mamoulis, "Continuous Spatial Assignment of Moving User" , VLDB, 2010.

9. Bertsekas, D. P., and D. A. Castanon, "A Generic Auction Algorithm for the Minimum Cost Network Flow Problem, Alphatech Report, Burlington, MA, Sept. 1991.

10. Bertsekas, D. P., "Auction algorithms for network flow problems: A tutorial introduction,

Computational Optimization and Applications , Vol. 1, pp. 766, 1992.

11. Cormen, T. H., C. E. Leiserson, R. L. Rivest, and C. Stein, *Introduction to Algorithms, Second Edition* , MIT Press and McGraw-Hill, ISBN 0-262-03293-7, Section 26.3: Maximum bipartite matching, pp. 664669, 2001.

12. Karp, R., U. Vazirani and V. Vazirani, "An Optimal Algorithm for On-Line Bipartite Matching" , *Proceedings of the* 22^{nd} *ACM STOC* , pp. 352-358, 1990.

13. Kuhn, H. W., "The Hungarian method for the assignment problem" , *Naval Research Logistics* , Quarterly 2, pp. 83-97, 1955.

14. Munkres, J., "Algorithms for the Assignment and Transportation Problems" , *Journal of the Society of Industrial and Applied Mathematics* , Vol. 5, No. 1, pp. 32-38, March 1957.

15. Derigs, U., "A shortest augmenting path method for solving minimal perfect matching problems" , Networks Vol. 11, No. 4, pp. 379390, 1981

16. Bertsekas D. P. and D. A. Castanon, "Parallel Synchronous and Asynchronous Implementations of the Auction Algorithm" , *Parallel Computing* , Vol. 17, pp. 707-732, 1991.

17. Ghinita G., P. Kalnis, M. Kantarcioglu and E. Bertino, "A Hybrid Technique for Private Location-Based Queries with Database Protection" , SSTD 2009.

18. Mokbel, M. F., C. Y. Chow and W. G. Aref, "The new casper: Query processing for location services without compromising privacy" , VLDB 2006.

19. Gruteser, M. and D. Grunwald, "Anonymous usage of location-based services through spatial and temporal cloaking" , MobiSys 2003, San Francisco, CA 2003.

20. Gedik, B. and L. Liu, "A customizable k-anonymity model for protecting location privacy" , ICDCS 2005, Columbus, OH, 2005.

21. Li, C. and B. Iyer , "Executing SQL over Encrypted Data in Database Service Provider Model", *In Proceedings of ACM Sigmod* , USA, 2002.

22. Indyk, P. and D. P. Woodruff, "Polylogarithmic Private Approximations and Efficient Matching" , Halevi, S., Rabin, T. (eds.) TCC 2006.

23. Khoshgozaran, A., C. Shahabi and H. Shirani-Mehr, "Location Privacy; Moving Beyond K-anonymity, Cloaking and Anonymizers" , Technical report, University of Southern California 2008.

24. Ghinita, G., P. Kalnis, A. Khoshgozaran, C. Shahabi and K. L. Tan, "Private Queries in Location Based Services: Anonymizers are Not Necessary" , SIGMOD 2008, Vancouver, BC, Canada 2008.

25. Yiu, M. L., C. S. Jensen, X. Huang and H. Lu, "Spacetwist: Managing the Trade-offs among Location Privacy, Query Performance, and Query Accuracy in Mobile Services" , ICDE 2008.

26. Lung Y. M., G. Ghinita, C. S. Jensen and P. Kalnis, "Outsourcing Search Services on Private Spatial Data" , ICDE 2009.

27. Yiu, M. L., G. Ghinita, C. S. Jensen and P. Kalnis, "Enabling Search Services on Outsourced Private Spatial Data" , *VLDB Journal* 2009.

28. Guttman, A., "R-trees: A Dynamic Index Structure for Spatial Searching, *Proc. of the ACM SIGMOD* Intl. Conf., 1984.

29. Gunther, O. and J. Bilmes, "Tree-Based Access Methods for Spatial Databases: Implementation and Performance Evaluation" , *IEEE Transactions on Knowledge and Data Engineering* , Vol. 03, No. 3, pp. 342-356, Sept., 1991.

30. R-tree, Wikipedia, http://en.wikipedia.org/wiki/R-tree, 2010.

31. Agrawall, R., J. Kiernan and R. Srikant, "Order Preserving Encryption for Numeric

Data" , *ACM Sigmod Record* , France, 2004.

32. Lawder, J. K. and P. J. H. King, "Querying Multi-dimensional Data Indexed Using the Hilbert Space-filling Curve" , *SIGMOD Record* , Vol. 30, No. 1, 2001.