Scalable Spatial Information Discovery over Distributed Hash Tables

Faraz Memon, Daniel Tiebler,
Frank Dürr, Kurt Rothermel
IPVS – Distributed Systems Department
Universität Stuttgart
Universitätsstraße 38, 70569 Stuttgart, Germany
firstname.lastname@ipvs.uni-stuttgart.de

Marco Tomsu, Peter Domschitz
Bell Laboratories Alcatel–Lucent
Lorenzstraße 10, 70435 Stuttgart, Germany
firstname.lastname@alcatel-lucent.de

ABSTRACT

In this paper, we present a Peer-to-Peer (P2P) spatial information discovery system that enables spatial range queries over Distributed Hash Tables (DHTs). Our system utilizes a less-distorting octahedral map projection in contrast to the quadrilateral projections used by majority of the previously proposed systems, to represent the spatial information. We also introduce a Space-Filling Curve (SFC)-based data placement strategy that reduces the probability of data hot-spots in the network. Moreover, we show that our system achieves scalable resolution of location-based range queries, by utilizing a tree-based query optimization algorithm. Compared to the basic query resolution algorithm, the query optimization algorithm reduces the average number of parallel messages used to resolve a query, by a factor of 96%.

Categories and Subject Descriptors

C.2.1 [**Computer-Communication Network**]: Network Architecture and Design; H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval—*Search process*

General Terms

Algorithms, design, performance

Keywords

Peer-to-Peer, overlay networks, spatial information discovery, location-based queries, space-filling curves, distributed hash tables

1. INTRODUCTION

Over the past few years, extensive research in the area of structured peer-to-peer (P2P) systems has paved the way for scalable, efficient, and fault-tolerant overlay networks. Naturally, these overlay networks have been adopted by several application areas for sharing diverse information. One such application area is spatial information discovery, which is the focus of this paper.

Typically, spatial information discovery systems require support for location-based range queries. Queries, such as *"Retrieve information about all hotels within 1 km of current location"* and *"Retrieve all the video streams within area X"* are examples of location-based range queries. The support for such queries can be provided either by constructing a unique P2P overlay network or by utilizing a Distributed Hash Table (DHT)-based P2P overlay network.

P2P spatial information discovery systems, such as Globase.KOM [13], GeoPeer [2], and [11] construct a unique network topology by organizing peers in a tree or a lattice structure to support location-based range queries. Due to complex structural requirements, the network construction and maintenance overhead for these systems is in general higher than that of a DHT with a simple basic structure such as a *ring* or a *torus*.

Conventional DHT-based P2P overlay networks, such as Chord [24], CAN [20], and Pastry[21] used to support only exact-match queries. However, recently they have been extended to support multi-attribute and range queries [1, 4, 8, 22, 23]. P2P spatial information discovery systems, such as [5, 9, 12, 25, 26] leverage the DHT-based systems by implementing a layer on top of them to support location-based range queries. This layer consists of either a tree-based data index or a locality-preserving mapping of the data to the peers. Such a layered architecture is attractive because it allows any 3^{rd} party DHT implementation to be used without modifications. However, the layered architectures have some shortcomings discussed below.

One of the most desirable property of the P2P spatial information discovery systems is the locality preservation of the data, i.e., data objects that lie close on the surface of the earth, should be placed on the same peer or a group of neighboring peers in the network. This allows for the efficient retrieval of the data objects, without flooding the whole network with a query. P2P spatial information discovery systems that use only a tree-based data index on top a DHT, do not preserve the locality of the data well. Moveover, P2P spatial information discovery systems that utilize a locality-preserving mapping, do not scale because of the large number of messages generated by location-based range queries.

In this paper, we present a P2P spatial information discovery system that performs a scalable resolution of location-based range queries, in spite of using a locality-preserving mapping of the data to the peers. Our system has a layered architecture to allow any DHT to be used as a network structure. The only requirement is that the DHT provides the standard *lookup(key)* and *notify()* methods for communication. The key innovations of our system include: (1) use of a less-distorting octahedral map projection to represent

the spatial data, (2) utilization of the Sierpinski Space-Filling Curve (SFC) for locality-preserving mapping of the data to the peers, (3) a data placement strategy that reduces the probability of the data hot-spots in the network, (4) and finally a scalable implementation of range queries by utilizing just the basic DHT *lookup(key)* method.

The rest of the paper is organized as follows: In Section 2, we present a review of the related work in the area of P2P spatial information discovery. In Section 3, we discuss the architecture of our system along with its components, in detail. Results from the system evaluations are presented in Section 4. Finally, we wind up the paper with conclusion and a notion of the future work in Section 5.

2. RELATED WORK

Generally, P2P spatial information discovery systems could be divided into two major categories: ones that built a unique network topology (Non-DHT based Systems) and others that use a DHT-based P2P overlay network to support location-based range queries (DHT based Systems). In this section, we briefly discuss these systems in relation to our spatial information discovery system.

2.1 Non-DHT based Systems

The Globase.KOM system [13] enables location-based range queries by organizing peers in a tree-based overlay network. The system utilizes the Plate Carrée map projection to index spatial data on the surface of the earth. Such a map projection introduces more distortion compared to the octahedral map projection utilized by our system. Moreover, the search efficiency of the Globase.KOM system is influenced by the amount of stored cache information which is not the case for our system.

The GeoPeer system [2] organizes peers to form a Delaunay triangulation, with each peer being responsible for holding information about a certain area on the globe. Unlike a DHT, the search algorithm opted by the GeoPeer is not as efficient, because each peer maintains a list of only few neighboring nodes.

The spatial information discovery system introduced by Kang et al. [11] also uses Delaunay triangulation to construct a network of peers. Similar to the GeoPeer system, the network coverage of a spatial query is not limited, i.e., a query could get flooded throughout the network before arriving at the destination peer. In contrast to that, our system uses DHT for routing queries. DHT has a known bound of $O(logN)$ steps to route a message from a source node to a destination node.

2.2 DHT-based Systems

DHT-based P2P overlay networks have evolved over a period of time. As a result, scalable implementations of DHTs, such as Open Chord [15], are widely available, which raises the desire to utilize them for spatial information discovery.

Harwood et al. [9] and Tanin et al. [25] extend the Chord [24] protocol for spatial information discovery using a quadtree and an octree, respectively. Their systems recursively sub-divide the virtual world into regions and assign these regions to a network of Chord peers. A location-based range query is first translated into a set of identifiers, then a lookup operation is performed for each of these identifiers to resolve the query. The shortcoming of these systems is that the assignment of the regions to the peers is not locality-preserving, leading to longer path lengths in the DHT for the location-based range queries.

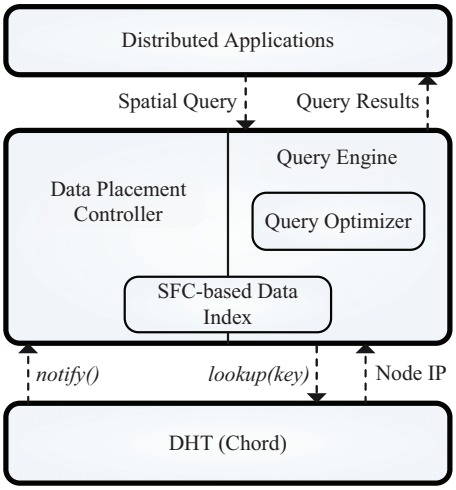

Figure 1: System Architecture

P2P spatial information discovery systems proposed by Chawathe et al. [5], Wang et al. [26] and Knoll et al. [12] are the ones that come closest to our system.

Chawathe et al. [5] use the Z-curve (aka Morton-order or z-order) [17] to reduce the multi-dimensional location data to a single dimension. This 1-dimensional data is then distributed over the Chord ring using a Prefix Hash Tree (PHT) [19]. The use of PHT introduces extra maintenance overhead for the system because the information about the PHT nodes has to be maintained at each peer in the network. In contrast to their system, the only information maintained by our system is the one maintained by the routing tables of the DHT nodes.

Wang et al. [26] sub-divide the space of a CAN [20] DHT into scatter regions. A scatter region is further partitioned into zones. A spatial data object is then mapped to a single scatter region and hashed to a random zone within the scatter region. The hashing is done to achieve better load balancing, but the locality of the data is lost, particularly if the scatter region is large.

Knoll et al. [12] compare the performance of the Peano [18], Z-Curve, and the Hilbert [10] SFC-based indices over a Pastry [21] network. They analyze the locality preservation properties of each SFC, but do not focus on location-based range queries. Moreover, they utilize a quadrilateral map projection in contrast to the less-distorting octahedral projection used by our system.

Although general DHT-based P2P information discovery systems, such as [8], and [22] could be extended to support location-based range queries, such an extention would require a map projection with only small distortion, like the octahedral projection utilized by our system. Moreover, related systems based on DHTs and SFCs could benefit from the optimized query processing mechanisms proposed in this paper.

3. SYSTEM ARCHITECTURE

The architecture of our system is a layered architecture, shown in Fig. 1. The layers are represented as boxes with thicker border. The top layer is the application layer that consists of large-

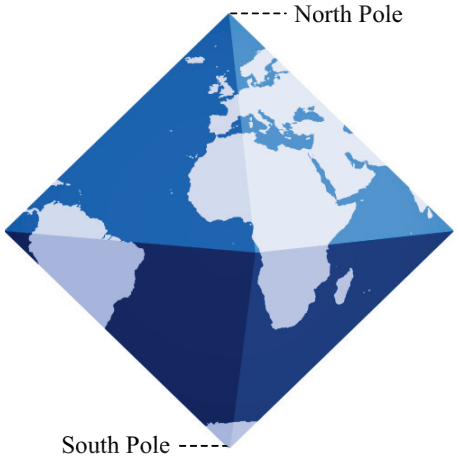

Figure 2: Octahedral Map Projection

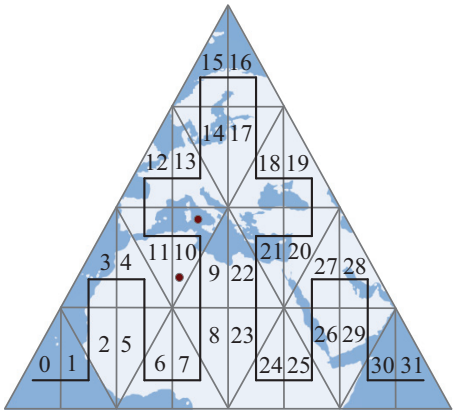

Figure 3: Sierpiński Space-Filling Curve

scale distributed applications that rely on spatial range queries, e.g., location-based services or other context-aware applications.

The middle layer is the framework layer. The framework layer consists of four modules: SFC-based data index, data placement controller, the query engine, and the query optimizer. The SFC-based data index is used to generate identifiers for the spatial data objects. These data objects are then placed in the network by the data placement controller. The query engine of the framework layer is responsible for the query resolution. In order to achieve scalable resolution of spatial range queries, the query optimizer performs the routing optimization.

The bottom layer is the DHT layer. The DHT layer provides the methods *lookup(key)* and *notify()*. Given a key, the *lookup(key)* method returns the IP address of the node responsible for it, while *notify()* is a call-back method that indicates a change in the key set of a node. The following sections discuss each of the framework modules in detail.

3.1 Data and Query Model

The spatial data objects in our system are defined as points on the surface of the earth. Each data object consists of two primary attributes, namely `Latitude` and `Longitude`. Apart from these primary attributes, a data object can accommodate several other attributes that describe its properties. For example, a data object can be defined as ($\texttt{Latitude} = 48.745^\circ$, $\texttt{Longitude} = 9.106^\circ$, $\texttt{Building} = University$, $\texttt{Department} = CS$, ...). However, it would be indexed only using the primary attributes `Latitude` and `Longitude`.

A spatial range query in our system is defined as a polygon on the surface of the earth. The query polygon can be described as a conjunction of tuples of the form ($\phi = value, \lambda = value$) where ϕ and λ represent `Latitude` and `Longitude` respectively. For example, a spatial range query could be defined as $\{(\phi = 48.747^\circ, \lambda = 9.106^\circ)$ AND $(\phi = 48.743^\circ, \lambda = 9.101^\circ)$ AND $(\phi = 48.744^\circ, \lambda = 9.111^\circ)\}$. All the spatial data objects held by the peers that are responsible for the areas corresponding to the query polygon, are returned. A spatial range query can also include other attributes to allow local filtering of data on the queried peers.

3.2 SFC-based Data Index

As discussed earlier, the locality preservation of the spatial data is the most desirable property of spatial information discovery systems. Locality preservation could be achieved by utilizing a SFC-based mapping. Since SFCs work with Cartesian coordinate system, the surface of the earth has to be projected from geographical coordinate system to the Cartesian coordinate system. We utilize the octahedral map projection [7], in contrast to the quadrilateral projections, for this purpose. Fig. 2 shows the octahedral map projection of the surface of the earth. The Octahedral map projection achieves good polar symmetry and the important global features (poles, equator and meridians) are mapped to triangular faces and vertices of the projection [14]. To the best of our knowledge, such a map projection has not been used by any P2P-based spatial information discovery system.

Although quadrilateral maps are easier to work with, they tend to have greater distortion as we start to move away from the equator towards the poles of the earth [3]. Octahedral map projection, however, is less distorting. This means that if the surface of an octahedral globe is sub-divided into smaller regions, each region would be of approximately equal areal shape and size. If these regions are uniformly distributed over a network of peers, each peer would hold information about almost an equal area of the earth, which would not be the case if a quadrilateral map projection is used.

We utilize Sierpiński SFC [3] to map the spatial data objects on the surface of the octahedral map to the DHT while preserving data locality. Sierpiński SFC is chosen over Hilbert [10], Peano [18], Z [17], and other SFCs because it achieves regular sub-division of the triangular faces of the octahedral map.

A Space-Filling curve (SFC) is essentially a mapping from a multi-dimensional space to a 1-dimensional line. The SFC-based data index in our system is created by defining eight Sierpiński SFCs [3] on each face of the octahedral map.

Each face of the octahedral map is a triangle. The construction of the Sierpiński SFC on a triangular face of the octahedral map is a two step process. The first step divides the triangle into smaller triangles (which we call zones). This division step can be viewed as a recursive process, where each run of the process divides each triangle into two equal triangles. The process continues k times

yielding 2^k zones. In the next step, a line is drawn that enters and leaves the triangular face of the map, passing through each of the 2^k zones once. The line imposes an order among the zones as it walks through them. As a result, a k^{th} order SFC-based mapping is defined for the triangular face of the earth. Fig. 3 shows a 5^{th} order Sierpinski SFC on a triangular face of the octahedral map that covers parts of Asia, Africa, and Europe.

The part of the SFC line that passes through a zone is an approximation of all the points in that zone. Consider the example of the spatial data objects shown as dots in Fig. 3. Although the data objects belong to separate locations on the map, both are indexed by the 10^{th} zone on the Sierpinski SFC, i.e., both the data objects have an identifier of 10.

Even though a single continuous Sierpinski SFC could be constructed on the octahedral map that indexes all the spatial objects, we utilize a separate SFC-based data index for each face of the map. This is done to achieve better distribution of data over the DHT, which we discuss in the next section.

3.3 Data Placement Controller

The data placement controller in the framework layer of our system is responsible for placing the spatial data objects over a DHT. Without loss of generality, we utilize the Chord [24] DHT to place spatial data objects in the network.

Chord provides the functionality for the look-up of keys in a network. The protocol constructs a network of peers by arranging them in an identifier circle modulo 2^m. Each peer is assigned an m-bit identifier from an identifier space of $[0, 2^m)$. The identifier is generated by hashing the IP address of the peer using a base hash function such as SHA-1. The Chord protocol provides the methods *lookup(key)* and *notify()* for peer communication. The *lookup(key)* method returns the IP address of the peer whose identifier is equal to or follows the identifier of the key in the circle. This peer is called the successor of the key. The *notify()* method indicates a change in the key set of a peer [24].

For efficient routing, a Chord peer maintains m entries in its routing table (called the finger table). For a peer n in the system, the i^{th} entry in its finger table is the identity of the peer that is successor of $n + 2^{i-1}$ on the identifier circle. The finger table ensures that a query is routed to the destination peer in $O(logN)$ overlay hops, in the worst case, where N is the total number of peers in the network.

The keys in our system are the identifiers generated by the SFC for the spatial data objects. When the framework layer of a peer receives a new data object from an application, the data placement controller generates an identifier for it using the Sierpinski SFC (c.f. Sec. 3.2). Once the identifier for the data object is known, the data placement controller performs a DHT lookup using this identifier in order to determine the peer that is responsible for the data object. Finally, the spatial data object is directly transferred to the peer responsible for hosting it.

The DHT layer of a peer reports any changes in the data set of the peer to the data placement controller using the *notify()* method call. These changes occur when the DHT performs a repair operation after a peer joins or leaves the network. After reception of a notification, the data placement controller redistributes the data among the neighbors accordingly.

If a single Sierpinski SFC is used to index all the spatial data objects on the surface of the octahedral map, the result could be a non-uniform distribution of the data over the Chord ring. For example, if a system indexes all the restaurants in the world, then the peers responsible for holding information about the oceans would contain no data. Since oceans are the largest parts of our planet, the majority of the peers would not be utilized for storage. To reduce the degree of non-uniform distribution of data, we use a separate SFC for each face of the octahedral map. This way, a peer is responsible for eight different regions on the map and the probability that each of these region contains no data, is reduced.

Using a separate SFC for each face of the octahedral map comes with a trade-off, i.e., the locality of the data along the edges where two faces meet, is disturbed. For example, if a single SFC is used, two spatial data objects that lie along the edge where two faces meet, would receive data identifiers that are close to each other. On the contrary, the same data objects would receive very different identifiers if two separate SFCs are used to index the two faces. Therefore, it is not straight-forward to extend this approach for map projections with larger number of faces, e.g., dodecahedral and icosahedral map projections.

Typically, the identifier space of the Sierpinski SFC, $[0, 2^k)$, is smaller than the identifier space of the Chord ring, $[0, 2^m)$. Therefore, if the spatial data objects from a face of the octahedral map are directly placed on the Chord ring, the peers with identifier values greater than 2^k, would contain no data. To avoid this, we scale up the identifier space of the SFC by a factor of 2^{m-k}. Details and an example of up-scaling could be found in our previous work [16].

3.4 Query Engine

Once the spatial data objects are placed in the network, they could be retrieved by any peer using a location-based range query. As described earlier, a location-based range query is defined as a polygon on the surface of the earth and all the spatial data objects covered by the area of this polygon are retrieved. The query engine that is a part of the framework layer is responsible for performing the query resolution.

After a location-based range query is received by the query engine from an application, the following sequence of operations are performed by the query engine. First, the queried area is mapped onto the octahedral map in order to determine the triangular faces of the map that are covered by the query. These triangular faces are then further refined using the Sierpinski SFC up till the highest approximation level k. During the refinement process, the SFC zones that do not match the query polygon are pruned away, resulting in a set of matching zone identifiers. The matching zone identifiers, form clusters of zones where a single cluster contains a sequence of continuous zone identifiers. Once the identifiers of the zones matching the query are determined, the peer performs the following sequence of operations:

- A DHT lookup is performed for the first zone identifier in each zone cluster, in order to determine the peers responsible for these zones.
- The query, along with the cluster and the identity of the query initiator, is transferred directly to the peer responsible for the first zone in that cluster.

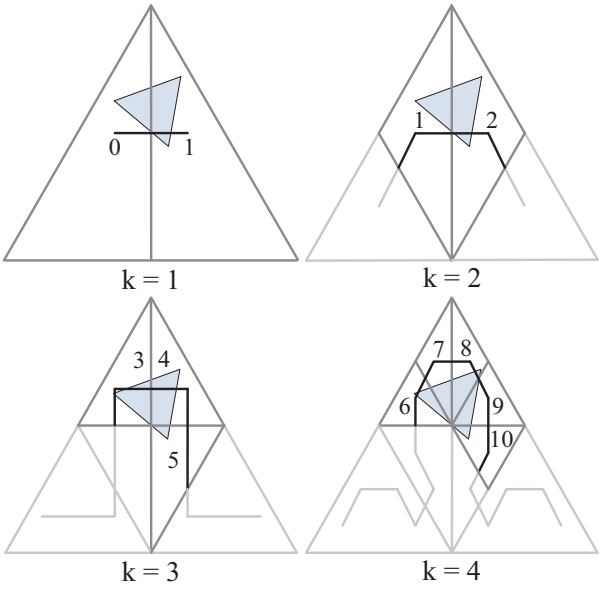

Figure 4: Successive Query Refinement

A peer that receives a query and a cluster of zones from some other peer in the network, performs the following set of operations:

- A local repository lookup is performed and the matching data objects are directly transferred to the query initiator.
- The zone identifiers that the peer itself is responsible for, are removed from the cluster.
- A DHT lookup is performed for the first zone identifier in the remaining cluster.
- The query, along with the remaining cluster and the identity of the initiator, is transferred directly to the peer responsible for the first zone in the remaining cluster.
- The above four steps are repeated until all clusters are completely resolved.

Fig. 4 shows an example of successive refinement of a query by the Sierpinski SFC with the maximum approximation level of 4. The query is shown as a shaded triangle (3-sided polygon) and the grayed out areas represent the zones that are pruned away by the refinement process. For the 1^{st} approximation level, the query matches the zones 0 and 1. Therefore, both of these zones are further refined to get four zones in the next approximation level. Note that for $k = 2$, the identifiers for the non-matching zones are not generated by the refinement process, which indicates that these zones are pruned away. For $k = 3$, only zones 1 and 2 from the previous step are further refined, since only these zones match the query. Finally for the last approximation level, zones 3 to 5 from the previous step are refined further to get zones 6 to 10 as a single cluster.

Now consider a Chord identifier ring shown in Fig. 5 (long distance links have been removed for clarity). Physical peers are shown as circles with darker borders. The query in the example above would be resolved by the peers 6, 7 and 12.

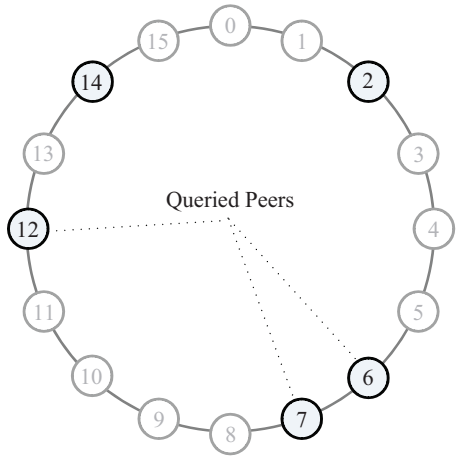

Figure 5: Chord Ring with $m = 4$ and $N = 5$

3.5 Query Optimizer

The query resolution approach, discussed in the section above is a basic one. There are several shortcomings of such an approach. Typically a location-based range query results in a large number of clusters on the Sierpinski SFC. Performing a DHT lookup for the first zone identifier in each cluster generates a high message load at the query initiator. Moreover, due to locality preservation property of the SFC, a single peer is typically responsible for one or more continuous clusters. Since a query is also composed of clusters, such a peer would receive multiple copies of the same query for each zone cluster. A simple approach to overcome this problem is to sequentially forward the query along the ring, starting from the peer that is responsible for the first zone identifier in the first cluster. However, due to the lack of parallelism, such an approach would increase the latency of the query resolution, drastically.

In this section, we discuss a tree-based query optimization strategy that reduces the message load of the query initiator with a slight increase in the latency of the query resolution. This query optimization strategy has been previously proposed by us for the OID system [16].

In order to perform the query optimization, our system defines two parameters, f (fanout) and l (depth-level). Parameter f, limits the number of messages forwarded by a peer in the network, while parameter l, limits the total number of parallel messages in the network. After the query initiator has calculated the identifiers of the SFC zones matching the query, it performs the following sequence of operations:

- The zone identifiers are ordered in an ascending order and divided into f sets.
- A DHT lookup is performed for the first zone identifier in each of the f sets.
- Once the identity of the peers responsible for these identifiers is known, the sets, along with the query, the identity of the initiator, and the values of the parameter f and l, are directly transferred to the corresponding peers.

On receiving a set of identifiers along with other query parameters,

Figure 6: NASA Visible Earth Light Distribution

a peer performs the following set of operations:

- A local repository lookup is performed against the received query and all the matching spatial data objects are directly transferred to the query initiator.
- The zone identifier that the peer itself is responsible for, are removed from the received set.
- If there are still some identifiers left in the set and the value of l is greater than zero, the remaining set is further divided into f smaller sets.
  - A DHT lookup is performed for the first zone identifier in each of the new sets.
  - Each set, along with the query, the identity of the initiator, and the values of the parameter f and l, are directly transferred to the responsible peers.
- If l is zero, the remaining set is sequentially forwarded through the responsible peers without increasing the number of parallel messages.
- The above set of operations are repeated until all clusters are completely resolved.

For more details on the routing optimization see [16].

4. SYSTEM EVALUATIONS

In this section, we present evaluations of our system. We simulated the use of Sierpinski SFC over the Chord DHT. The simulations were performed over a cluster of 32 nodes with each node having a 2.7 GHz dual-core AMD Opteron processor and 8 GB of RAM.

The data set for our simulations is generated using the NASA Visible Earth [6] image of the city light distribution over the globe (Fig. 6), where the probability of bright pixels is higher than the probability of the dark pixels. This image, for example, represents the distribution of developed areas around the world.

For performance comparison, we simulated the following two configurations of our system:

Configuration A (Multiple SFCs): A separate Sierpinski SFC is used for each face of the octahedral map to index the spatial data.

Configuration B (Single SFC): A single continuous SFC is used to index the spatial data on the octahedral map.

Param	Value(s)	Description
N	$10^2, 10^{2.5}, 10^3 \ldots 10^5$	# of peers
m	128	# of Chord identifier bits
O	$10 * N$	# of spatial data objects
k	32	SFC approximation level
f	10	fanout
l	$\lfloor log(N) \rfloor - 1$	depth-level

Table 1: Performance Evaluation Parameters

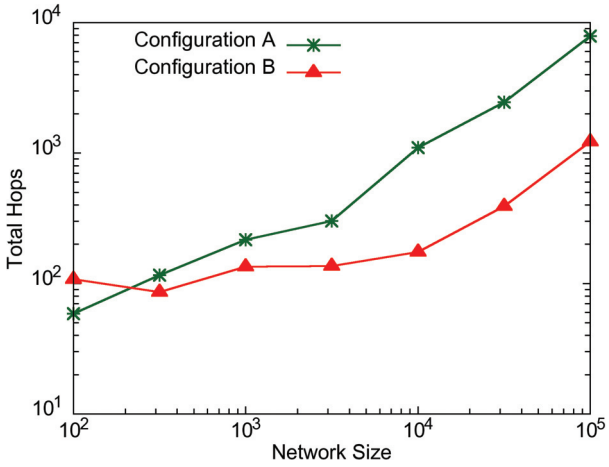

Figure 7: Total Hopes

4.1 Performance Evaluation

For evaluating the performance of our spatial information discovery system, we perform a location-based range query over Germany with the parameters shown in Table 1. The peer identifiers are uniformly distributed over the identifier space of $(0, 2^m)$. Like a typical P2P system, the total number of data objects is increased with the network size. The approximation level k is chosen such that a zone of the SFC represents an area that is approximately equal to the size of a FIFA football field (approx. 8000 m^2). The values for the query parameters f and l are chosen such that the number of parallel messages is restricted to 10% of the network size in the worst case. The following performance metrics are measured for the two types of system configurations described above:

- **Total Hops** - Total number of hops of all messages needed to resolve a query. This includes the hops of the DHT lookup messages as well.
- **Processing Peers** - Number of peers that perform a local repository lookup to evaluate the query.
- **Data Peers** - Number of peers containing the data objects that match the query (subset of processing peers).

Fig. 7 to Fig. 9 show the performance of both system configurations with respect to the performance metrics defined above. For each of the network sizes shown in the graphs, the location-based range query is performed 10 times starting from a random peer in the network. The values are then averaged to produce a point on a

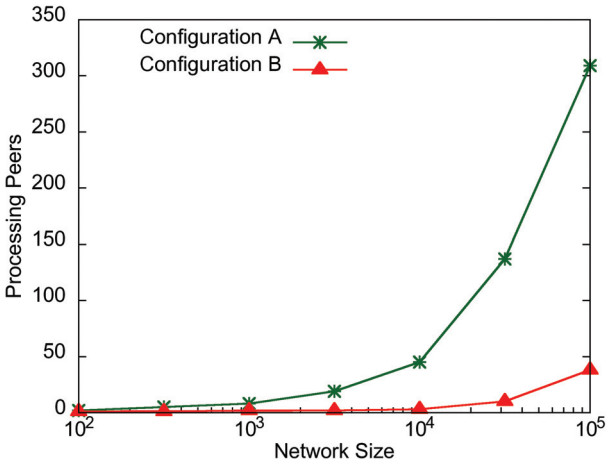

Figure 8: Processing Peers

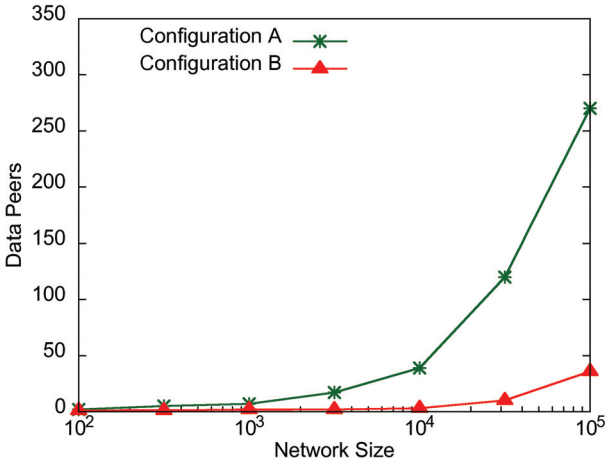

Figure 9: Data Peers

graph. Note that the x-axis of Fig. 7 to Fig. 9 and the y-axis of Fig. 7 are on a log scale.

The total number of hops, the number of processing peers, and the number of data peers increase with the network size in the case of both system configurations (Fig. 7 to Fig. 9). With the increasing network size, each peer is responsible for smaller range of identifiers. Hence, in order to resolve a location-based range query, the query has to be evaluated at a larger number of peers.

Param	Value(s)	Description
N	10^3	# of peers
O	10^4	# of spatial data objects
k	32	SFC approximation level

Table 2: Data Distribution Evaluation Parameters

For each of the performance metrics, system Configuration B (Single SFC) performs better than Configuration A (Multiple SFCs), because the spatial data is better distributed over the network in

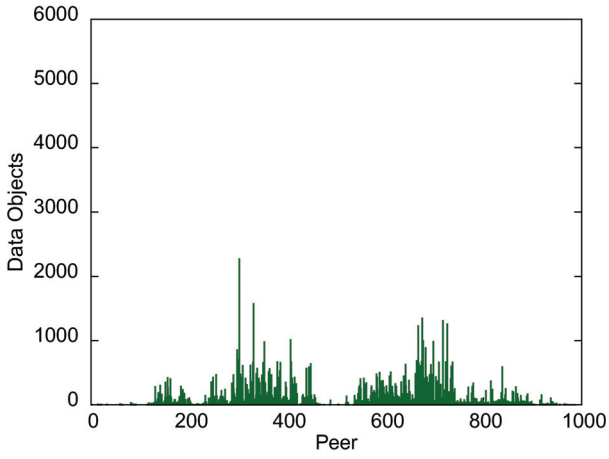

Figure 10: Data Distribution for System Configuration A

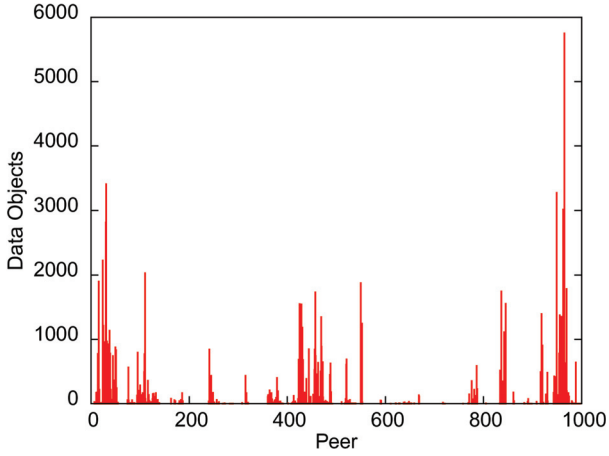

Figure 11: Data Distribution for System Configuration B

the case of system Configuration A. Therefore, the query has to be evaluated at a larger number of peers compared to the system Configuration B. However, as discussed earlier (c.f. 3.3), there is a trade-off between the performance and the degree of data distribution. The degree of data distribution for both system configurations is analyzed in the next section.

4.2 Data Distribution

In this section, we analyze the data distribution properties of the two different configurations of our spatial information discovery system defined above. The simulations are performed using the parameters shown in Table 2. For each peer in the network, the following metric is measured in the case of both types of system configurations:

- **Data Objects** - Number of data objects that a peer stores in its local repository.

Fig. 10 and Fig. 11 show the degree of data distributions achieved by the two system configurations. Clearly, there are several data

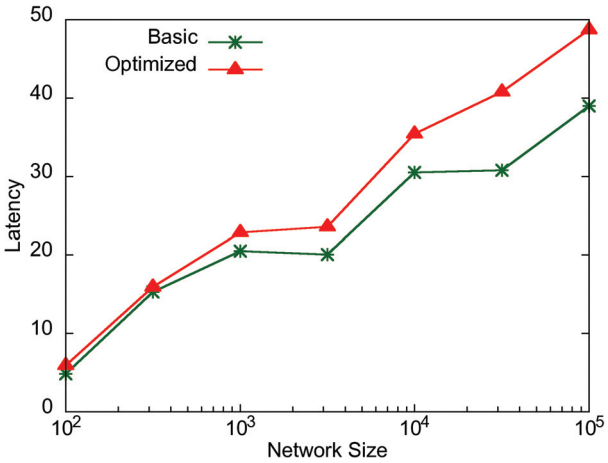

Figure 12: Latency for System Configuration A (Experiment 1)

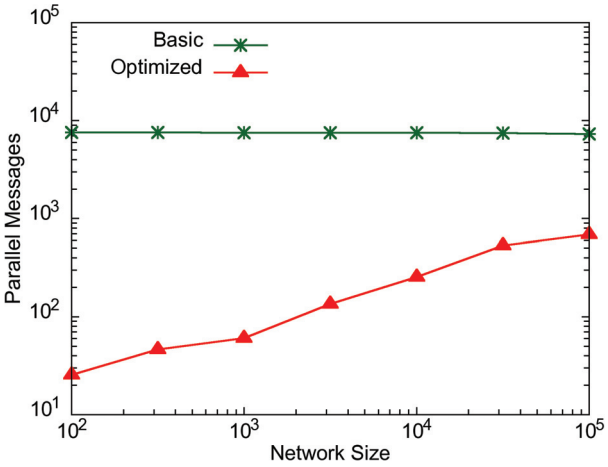

Figure 13: Parallel Messages for System Configuration A (Experiment 1)

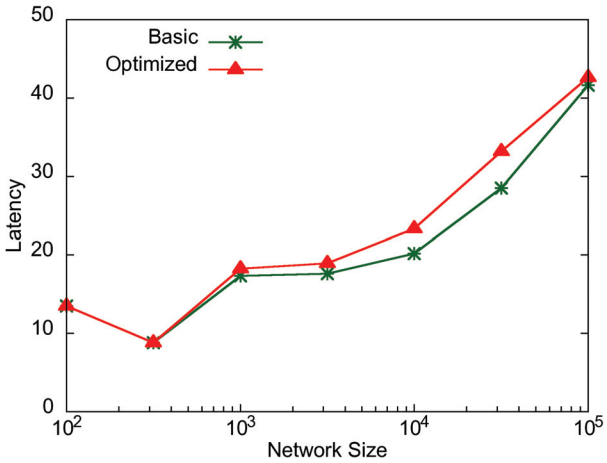

Figure 14: Latency for System Configuration B (Experiment 1)

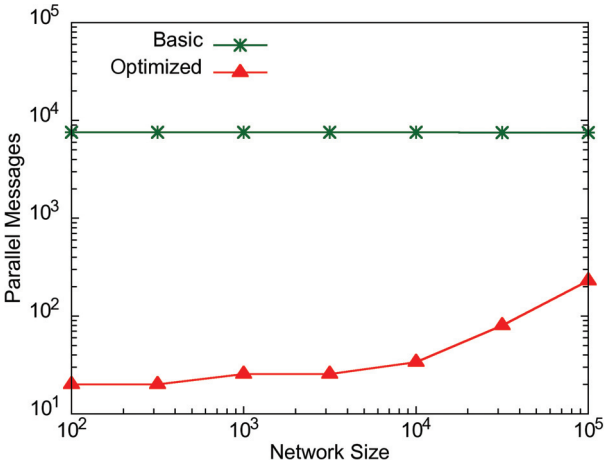

Figure 15: Parallel Messages for System Configuration B (Experiment 1)

hot-spots in the case of system Configuration B (Single SFC) (Fig. 11). However, Fig. 10 shows that the spatial data is better distributed in the case of system Configuration A (Multiple SFCs). For example, the number of peers responsible for 1000 or more data objects is 7 for system configuration A, compared to 24 for system configuration B.

The difference in the data distribution of the two system configurations is due to the fact that the peers responsible for holding information about the oceans contain no data in the case of system Configuration B (as there is hardly any light in oceans Fig. 6). However, in the case of system Configuration A, a peer is responsible for holding information about 8 different regions of the world. Therefore, the probability of a peer containing no spatial data is reduced drastically.

4.3 Performance vs. Data Distribution

As shown by the simulation results in Sec. 4.1 and Sec. 4.2, there is a trade-off between performance and the degree of the data distribution over the network. If the spatial data is non-uniformly distributed over the ranges of latitude and longitude (like with our data set), the system configuration with a separate SFC for each face of the octahedral map achieves better distribution of the data over the network, compared to the system configuration with a single continuous SFC. However, the performance deteriorates.

The query used for the performance evaluation is over the area of Germany. However, location-based range queries are typically over smaller areas, e.g., a single city. Since the performance of the system configuration with a separate SFC for each face of the octahedral map is not extremely bad, it is still feasible to utilize such a system configuration. For example, a query over the city of Stuttgart in Germany is resolved in 230 overlay hops over a network of 10^5 peers by the same system configuration.

4.4 Query Optimization

In this section we discuss the results of several experiments performed in order to compare the performance of the query optimization algorithm (c.f. Sec. 3.5) with basic query resolution (c.f. Sec. 3.4). Each experiment is performed on both types of system con-

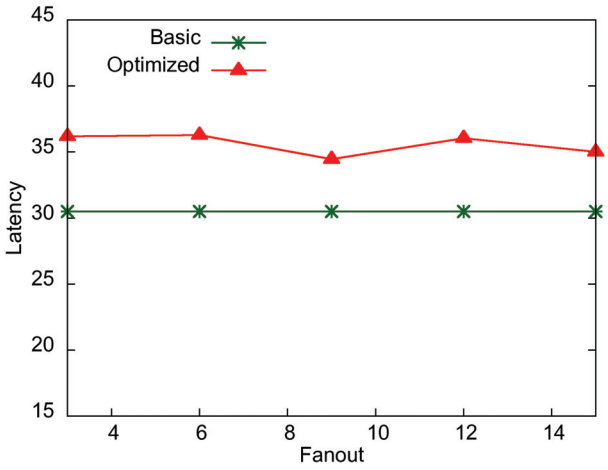

Figure 16: Latency for System Configuration A (Experiment 2)

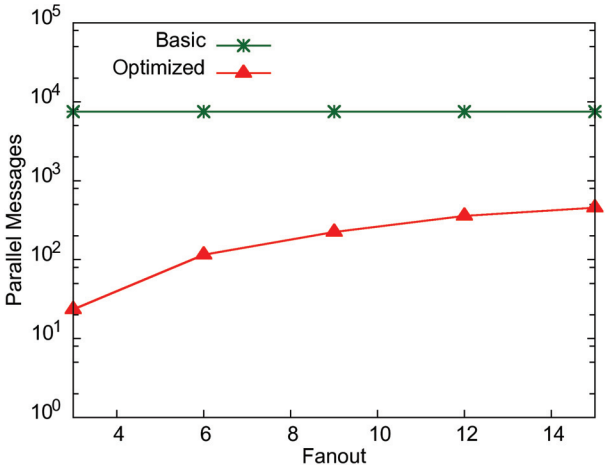

Figure 17: Parallel Messages for System Configuration A (Experiment 2)

figurations defined above. The following performance metrics are monitored during the experiments:

- **Latency** - Number of hops of the longest message path.
- **Average Number of parallel messages** - Total number of forwarded messages divided by the number of levels of the search tree.

The first experiment portrays the effects of varying network size on the performance of the two query resolution algorithms with respect to the metrics defines above. The simulation parameters are shown in Table 3.

Fig. 12 and Fig. 14 show that for each of the system configurations, the latency of the query resolution increases with the network size, in the case of both query resolution algorithms. Since with the increasing number of peers in the network, the range of identifiers that a single peer is responsible for decreases, the same query has to be evaluated at a larger number of peers.

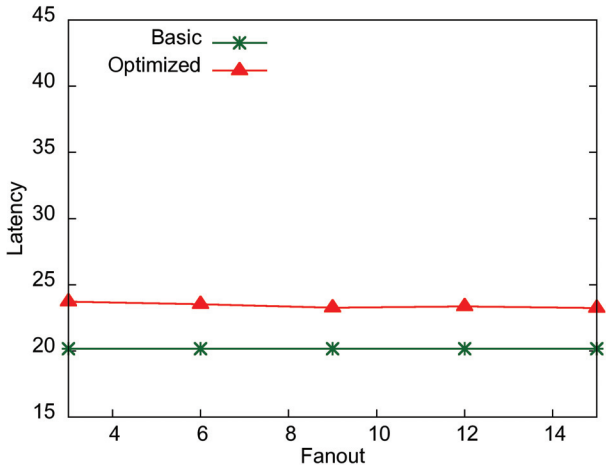

Figure 18: Latency for System Configuration B (Experiment 2)

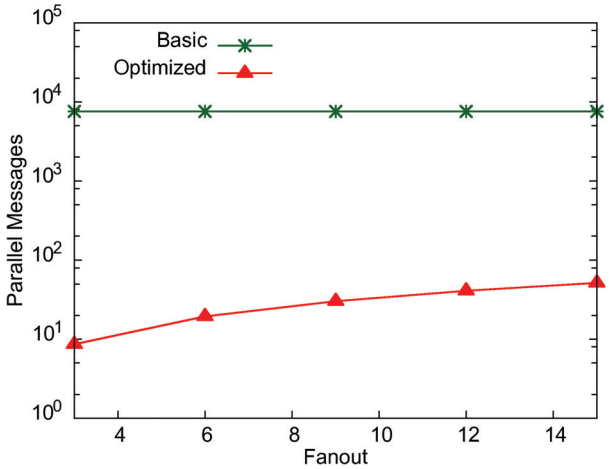

Figure 19: Parallel Messages for System Configuration B (Experiment 2)

The difference between the latencies of the two query resolution algorithms is very small for both types of system configurations (Fig. 12, 14). However, there is a significant difference between the average number of parallel messages produced by the two query resolution algoritms (Fig. 13, 15). The optimized query resolution algorithm produces fewer parallel number of messages than the basic query resolution algorithm, even with the network size as high as 10^5.

Param	Value(s)	Description
N	$10^2, 10^{2.5}, 10^3 \ldots 10^5$	# of peers
O	$10 * N$	# of spatial data objects
k	32	SFC approximation level
f	10	fanout
l	3	depth-level

Table 3: Optimization Evaluation Parameters (Experiment 1)

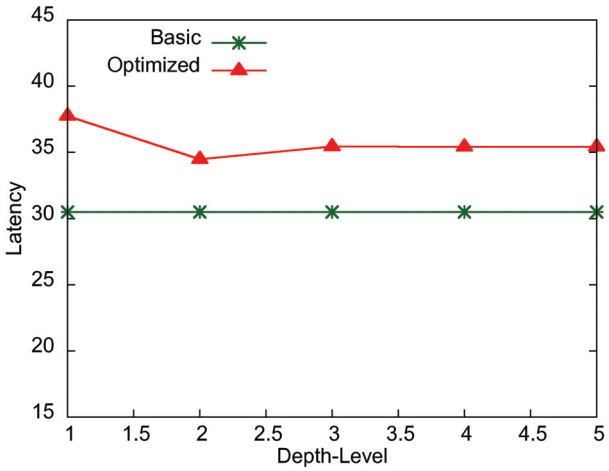

Figure 20: Latency for System Configuration A (Experiment 3)

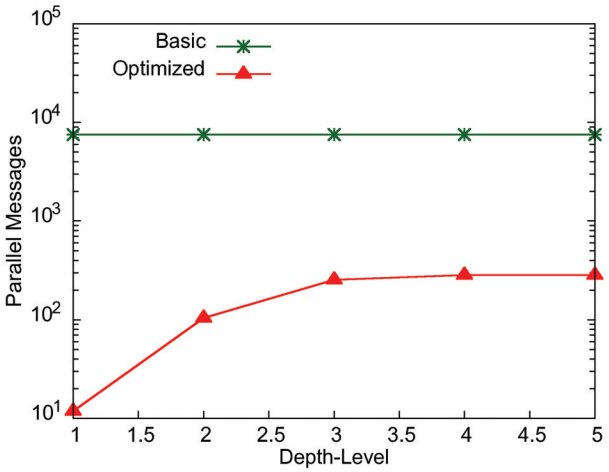

Figure 21: Parallel Messages for System Configuration A (Experiment 3)

The performance of system Configuration B (Single SFC) is in general better than the performance of system Configuration A (Multiple SFCs) because of the same reasons discussed in Sec. 4.1.

Param	Value(s)	Description
N	10^4	# of peers
O	10^5	# of spatial data objects
k	32	SFC approximation level
f	3, 6, 9 . . . , 15	fanout
l	3	depth-level

Table 4: Optimization Evaluation Parameters (Experiment 2)

The second experiment shows the effects of varying values of parameter f on the query optimization algorithm, with respect to the performance metrics defined above. For comparison, the performance of the basic query resolution algorithm is also plotted. The second experiment is performed using the simulation parameters

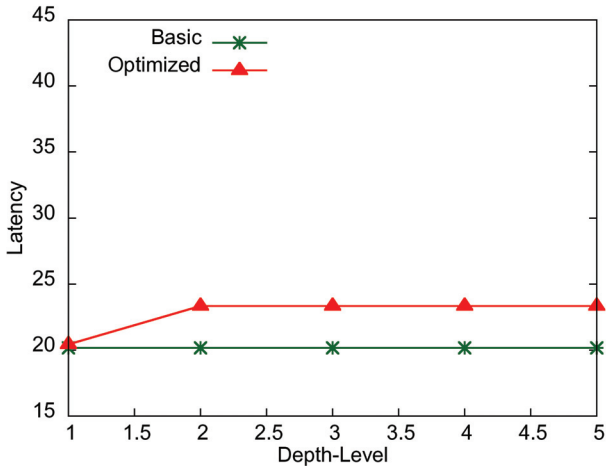

Figure 22: Latency for System Configuration B (Experiment 3)

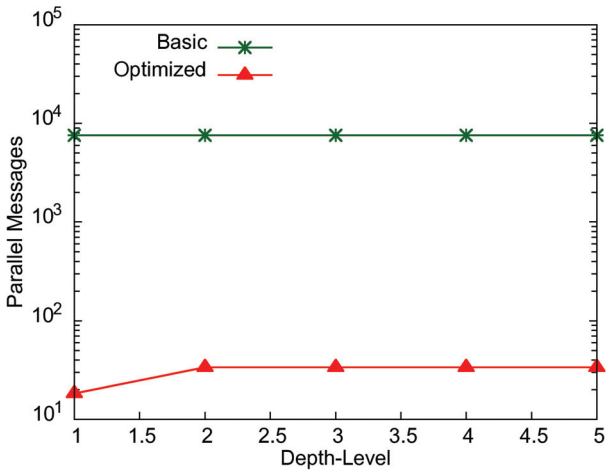

Figure 23: Parallel Messages for System Configuration B (Experiment 3)

show in Table 4 for both types of system configurations.

The latency for the query resolution remains constant for the basic query resolution (Fig. 16, 18) throughout Experiment 2, because all the query clusters are immediately forwarded by the query initiator to the responsible peers. Due to the same reason, the basic query resolution has slightly lower latency compared to the optimized query resolution.

Although the number of parallel messages for the optimized query resolution algorithm increases with increasing fanout values (Fig. 17, 19), it still remains below the number of parallel messages transmitted by the basic query resolution algorithm, even for the fanout value as high as 14. The number of parallel messages increases with the increasing fanout because the query clusters are divided more at each level of the query tree.

The third and the final experiment shows the effects of varying depth-level parameter l on the query resolution latency and the parallel number of messages in the network. The third experiment is

performed using the parameters shown in Table 5 for both types of system configurations.

Param	Value(s)	Description
N	10^4	# of peers
O	10^5	# of spatial data objects
k	32	SFC approximation level
f	10	fanout
l	$1, 2, \ldots, 5$	depth-level

Table 5: Optimization Evaluation Parameters (Experiment 3)

For the system Configuration A (Multiple SFCs), the query resolution latency first decreases, and then it remains almost constant for the optimized query resolution algorithm (Fig. 20). In the beginning the latency decreases as more parallel messages are used to resolve the query. Therefore, each message has to travel less through the network. Later, it seems that the whole query is resolved within two levels of the search tree. Therefore, starting from the depth-level 3 onwards, the query resolution latency remains almost constant.

Similarly, for the system Configuration B (Single SFC), the latency remains constant throughout the experiment for the optimized query resolution (Fig. 22). For $l = 1$, the value for the latency is an outlier because the query initiator was itself responsible for resolution of some query clusters and therefore, did not transmit them through the network.

The latency of the basic resolution algorithm is better than the latency of the optimized algorithm for the same reasons discussed in the second experiment.

The parallel number of messages first increases then remains constant with increasing values of parameter l for the optimized query resolution algorithm (Fig. 21, 23). As the value of parameter l increases, more number of messages are generated in order to resolve the query, until a certain point is reached. After this point, more messages cannot be generated because the query gets resolved with fewer messages already.

The parallel number of messages generated by the optimized resolution algorithm is significantly lower than the messages generated by the basic resolution algorithm, because of the same reasons discussed in the first experiment.

5. CONCLUSION AND FUTURE WORK

In this paper, we presented the system design and evaluation of a DHT-based spatial information discovery system. Our system is the first one to utilize the less-distorting octahedral map projection compared to the quadrilateral projections utilized by the majority of the previously proposed spatial information discovery systems. Our system uses Sierpinski SFC in order to map the spatial data on the surface of the octahedral map to a network of Chord peers, while preserving the data locality. We proposed and evaluated two different types of system configurations for utilizing the Sierpinski SFC. The first configuration uses a separate SFC for each face of the octahedral map and therefore achieves a better distribution of the data over the peers. The second configuration uses a single continuous SFC for the complete octahedral map and therefore achieves better query resolution performance.

We increased the scalability of our system by utilizing an optimized query resolution algorithm. According to our simulations, the optimized query resolution algorithm achieves up to 96% reduction in the average number of parallel messages used to resolve a query, equal to the area of Germany, over a network of $100,000$ peers.

For the future work, we shall investigate the effects using a polyhedral map projection that has a larger number of faces than the octahedral map projection, e.g., dodecahedral or icosahedral map projections. Furthermore, the development of a locality-aware load-balancing algorithm is also a part of the future agenda. Using locality-aware load-balancing it would be possible to use a single continuous SFC over the octahedral map and still achieve a better distribution of the data over the peers.

6. ACKNOWLEDGMENTS

Special thanks to Manuel Gonzalo and Lars Geiger of the Distributed Systems Department at Universität Stuttgart, for their helpful comments.

7. REFERENCES

[1] A. Andrzejak and Z. Xu. Scalable, Efficient Range Queries for Grid Information Services. In *Proceedings of the 2nd International Conference on P2P Computing*, page 33. IEEE Computer Society, 2002.

[2] F. Araújo and L. Rodrigues. GeoPeer: A Location-Aware Peer-to-Peer System. In *Proceedings of the 3rd International Symposium on Network Computing and Applications*, pages 39–46. IEEE Computer Society, 2004.

[3] J. J. Bartholdi and P. Goldsman. Continuous Indexing of Hierarchical Subdivisions of the Globe. *International Journal of Geographical Information Science*, 15(6):489–522, 2001.

[4] M. Cai, M. Frank, J. Chen, and P. Szekely. MAAN: A Multi-Attribute Addressable Network for Grid Information Services. In *Proceedings of the 4th International Workshop on Grid Computing*, page 184. IEEE Computer Society, 2003.

[5] Y. Chawathe, S. Ramabhadran, S. Ratnasamy, A. LaMarca, S. Shenker, and J. Hellerstein. A Case Study in Building Layered DHT Applications. In *Proceedings of the 2005 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications*, pages 97–108. ACM, 2005.

[6] N. V. Earth. *Earth's City Lights*. http://visibleearth.nasa.gov/, 2008.

[7] C. A. Furuti. *Polyhedral Map Projections*. http://www.progonos.com/furuti/MapProj/Normal/ProjPoly/projPoly.html, 1996-97.

[8] P. Ganesan, B. Yang, and H. Garcia-Molina. One Torus to Rule Them All: Multi-dimensional Queries in P2P Systems. In *Proceedings of the 7th International Workshop on the Web and Databases*, pages 19–24. ACM, 2004.

[9] A. Harwood and E. Tanin. Hashing Spatial Content over Peer-to-Peer Networks. In *Proceedings of the 2003 Australian Telecommunications, Networks and Applications Conference*, 2003.

[10] D. Hilbert. Über die stetige Abbildung einer Linie auf ein Flächenstück. In *Mathematische Annalen*, 1891.

[11] H.-Y. Kang, B.-J. Lim, and K.-J. Li. P2P Spatial Query Processing by Delaunay Triangulation. In *Proceedings of the 4th International Workshop on Web and Wireless*

Geographical Information Systems, pages 136–150. Springer, 2004.

[12] M. Knoll and T. Weis. Optimizing Locality for Self-organizing Context-Based Systems. In *Proceedings of the 1st International Workshop on Self-Organizing Systems*. Springer, 2006.

[13] A. Kovacevic, N. Liebau, and R. Steinmetz. Globase.KOM - A P2P Overlay for Fully Retrievable Location-based Search. In *Proceedings of the 7th International Conference on Peer-to-Peer Computing*, pages 87–96. IEEE Computer Society, 2007.

[14] R. Laurini and D. Thompson. *Fundamentals of Spatial Information Systems*. Academic Press Ltd., 1992.

[15] K. Loesing and S. Kaffille. *Open Chord*. http://sourceforge.net/projects/open-chord/, 2006.

[16] F. Memon, D. Tiebler, F. Dürr, K. Rothermel, M. Tomsu, and P. Domschitz. OID: Optimized Information Discovery using Space Filling Curves in P2P Overlay Networks. In *Proceedings of 14th IEEE International Conference on Parallel and Distributed Systems*, pages 311–319. IEEE Computer Society, 2008.

[17] J. A. Orenstein and T. H. Merrett. A Class of Data Structures for Associative Searching. In *Proceedings of the 3rd SIGACT-SIGMOD Symposium on Principles of Database Systems*, pages 181–190. ACM, 1984.

[18] G. Peano. Sur une courbe, qui remplit toute une aire plane (On a Curve Which Completely Fills a Planar Region). In *Mathematishe Annalen*, 1890.

[19] S. Ramabhadran, J. Hellerstein, S. Ratnasamy, and S. Shenker. Prefix Hash Tree - An Indexing Data Structure over Distributed Hash Tables. In *23rd Annual SIGACT-SIGOPS Symposium on Principles of Distributed Computing*. ACM, 2004.

[20] S. Ratnasamy, P. Francis, M. Handley, R. Karp, and S. Schenker. A Scalable Content-Addressable Network. In *Proceedings of the 2001 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications*, pages 161–172. ACM, 2001.

[21] A. Rowstron and P. Druschel. Pastry: Scalable, Decentralized Object Location and Routing for Large-scale Peer-to-Peer Systems. In *Proceedings of IFIP/ACM International Conference on Distributed Systems Platforms*, pages 329–350. ACM, 2001.

[22] C. Schmidt and M. Parashar. Flexible Information Discovery in Decentralized Distributed Systems. In *Proceedings of the 12th International Symposium on High Performance Distributed Computing*, page 226. IEEE Computer Society, 2003.

[23] Y. Shu, B. C. Ooi, K.-L. Tan, and A. Zhou. Supporting Multi-dimensional Range Queries in Peer-to-Peer Systems. In *Proceedings of the 5th International Conference on P2P Computing*, pages 173–180. IEEE Computer Society, 2005.

[24] I. Stoica, R. Morris, D. Karger, F. Kaashoek, and H. Balakrishnan. Chord: A Scalable Peer-to-Peer Lookup Service for Internet Applications. In *Proceedings of the 2001 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications*, pages 149–160. ACM, 2001.

[25] E. Tanin, A. Harwood, H. Samet, S. Nutanong, and M. T. Truong. A Serverless 3D World. In *Proceedings of the 12th International Workshop on Geographic Information Systems*, pages 157–165. ACM, 2004.

[26] H. Wang, R. Zimmermann, and W.-S. Ku. ASPEN: An Adaptive Spatial Peer-to-Peer Network. In *Proceedings of the 13th International Workshop on Geographic Information Systems*, pages 230–239. ACM, 2005.