A NEW FULLY DECENTRALIZED SCALABLE PEER-TO-PEER GIS ARCHITECTURE

S.H.L. Liang

Department of Geomatics Engineering, University of Calgary, Calgary, Alberta, CANADA T2N 1N4 – steve.liang@ucalgary.ca

Commission IV, WG IV/5

KEY WORDS: GIS, Internet, Scale, Web Based, Distributed

ABSTRACT:

Today's Internet GIS's centralized components have limited the system scalability in that they are single points of failure and system performance bottleneck. As a solution, we propose a new fully decentralized Internet GIS architecture, called P2P Spatial Access Method (P2PSAM). P2PSAM removes the centralized components by integrating dynamic spatial indexing algorithms (e.g., QuadTree) with fully decentralized P2P networks (e.g., Distributed Hash Table). Our experiments demonstrated that P2PSAM is able to scale to several millions of spatial data objects in a P2P environment without significantly increasing computational and networking overhead.

1. BACKGROUND

In this paper, we propose a new fully decentralized scalable Peer-to-Peer GIS architecture. The number of Internet GIS users and spatial data sources (e.g., geo-sensor networks) are increasing rapidly. It is challenging today's Internet GIS architectures in terms of their scalability. From literature and existing systems, there are two Internet GIS architectures: (1) Client-Server (C/S) GIS architecture, and (2) Distributed GIS architecture. Both architectural designs have centralized components, *i.e.*, the central server of the C/S GIS and the central directory/registry of the Distributed GIS. In distributed systems, centralized designs are cited as being vulnerable and consequently prevent the system to scale (Ratnasamy, 2001; Tewari, 1998).

When a system scales in users and data sources, its centralized components makes the system unreliable in that they are single points of failure. They also become system performance bottlenecks in that all additional system loads are added to the central components. Furthermore they result in poor flexibility of administration in that the system is controlled by the providers of these central components. For the above reasons, we need a new "fully" decentralized architecture (*i.e.*, without any centralized components) for the Internet GIS.

2. INTERNET GIS SCALABILITY CHALLENGE

2.1 Scalability: Definition

First, we start with a definition of the word: Scalability. Neuman defines (Neuman, 1994): *"A system is said to be scalable if it can handle the addition of users and resources without suffering noticeable loss of performance or an increase in administrative complexity."*

In the context of distributed systems, informally a system can be said to be scalable if it can support millions of users (Ratnasamy, 2002). Moreover, if a system has the following three characteristics, it is considered un-scalable (Tewari, 1998):

1. It is built using centralized components.
2. It has single point of failure.
3. It needs system-wide synchronization requirements.

2.2 A Scalability Analysis of the Current Internet GIS Architecture

In this section, we analyze the scalability of current Internet GIS architectures. From literature and existing systems, there are two existing Internet GIS architectures: (1) Client-Server (C/S) GIS Architecture, and (2) Distributed GIS Architecture. We examine the scalability of the two existing architectures using the following three dimensions because the three dimensions are all affected by the scale of the system. The three dimensions are: (1) Reliability; (2) System Load; and (3) Administration.

2.2.1 Client-Server(C/S) GIS Architecture

In the C/S GIS architecture, multiple clients issue requests by sending requests to a single server. The provider of the service (server) allocates physical resources such as storage, memory and CPU in order to satisfy incoming requests. In this architecture, multiple clients issue requests to the server in order to access the spatial data hosted by the server. Systems using the C/S GIS architecture range from historical systems, such as XEROX PARC Map Viewer (Putz, 1994) (one of the earliest prototypes of web-based GIS) and GRASSLinks (http://ippc2.orst.edu/glinks/), to today's ESRI ArcIMS (http://www.esri.com/arcims) and Autodesk MapGuide (http://www.mapguide.com). Below, we use the metric of three dimensions to examine the scalability of this C/S GIS architecture.

(1) Reliability: Reliability is related to system failure, network failure, disconnection, and availability of resources, etc. In the C/S GIS architecture, one server receives multiple serves multiple clients' requests. As a result, the server in this model is a single point of failure. In addition, as the number of the users increases, it becomes less likely that the single server will be available to all of the users all of the time (e.g., a large number of requests induces a server crash or a large geographical

distance between server and user leads to a network timeout). Thus, the C/S GIS architecture becomes less reliable as the system scale (e.g., population of users using the service increases).

(2) Load Balance: In this C/S GIS architecture, when the amount of spatial objects hosted by the server increases, the central server needs to locate more resources (e.g., CPU, storage, bandwidth, etc.) in order to manage the system. The central server needs to locate even more resources if the centralized server is hosting dynamic datasets (e.g., real-time sensor feeds). In addition, as the total number of users increases, the number of requests for service will also increase. As a result, in the C/S GIS architecture, as the size of data or the number of user increases, all of the additional system loads will be added onto the single central server. Consequently, the C/S GIS architecture has poor load scalability.

(3) Administration: In the C/S GIS architecture, all of the system information (e.g., a list of available datasets, their metadata, etc.) is stored centrally at the server. This central administration leads to two issues: (1) Frequent up-date: When the system grows (e.g., the number and type of spatial datasets increases), information about the system changes more frequently. All system loads caused by the frequent up-to-date operations will be added onto the central server. (2) Poor flexibility in administration: The C/S GIS architecture provides poor flexibility across administrative boundaries because the service is controlled solely by the organization that owns the central server. In a large scale Internet GIS system, it is less likely that all spatial datasets will be provided by a single organization or individual. Therefore the system has to cross administrative boundaries. As the system crosses administrative boundaries, organizations want more autonomy and control over their own part of the data and system. Consequently, the poor flexibility in administration makes the C/S GIS architecture not a suitable architecture to build large scale Internet GIS.

2.2.2 Directory-based Distributed GIS Architecture

In order to help the C/S GIS scale to a larger set of users and resources, some Internet GIS systems evolve to use the directory-based distributed GIS architecture. In this distributed architecture, clients firstly ask a centralized directory service to determine which service provider it should contact. Clients then contact the service provider independently. Example systems include ESRI ArcWeb Services (http://www.esri.com/arcwebservices/), UMN MapServer (http://mapserver.gis.umn.edu/), and GeoServer (http://geoserver.org/). Next, again we use the metric of three dimensions of scalability to examine this distributed GIS architecture.

(1) Reliability: In this distributed architecture, there are multiple servers that serve multiple clients' requests for the spatial datasets. The centralized directory service distributes client traffic across resources (e.g., multiple servers). This architecture improves the reliability of the C/S GIS architecture because the servers are no longer a single point of failure. However, even though the directory service can handle a large number of requests, it is still a central point of failure in this architecture. From the above introduced definitions of scalability, a system is considered as un-scalable if it has a single point of failure. Therefore, although the centralized directory service distributes client traffic, the distributed GIS architecture still becomes less reliable as the system scales.

(2) Load Balance: Compared to the C/S GIS architecture, the distributed GIS architecture has more severs in the system and allows new datasets to be added onto the server experiencing the lowest system load. As a result the distributed GIS architecture balances the system storage resources much better than does the C/S GIS where a single server is responsible for managing all datasets in the system. Moreover, the central directory service distributes the users' requests across multiple servers. Thus it also better balances system load than does the C/S architecture, where a single server is responsible for dealing with all user requests. However, when the total amount of users increases, the number of requests sent to the directory service in order to locate the desired sensors still increases. All of the additional system loads resulting from locating the desired spatial data will be added onto the single central directory service. For this reason, although the distributed GIS architecture provides better load balance, the centralized directory service is still the system load bottleneck.

(3) Administration: In the directory-based distributed GIS architecture, most of the system information (e.g., a list of available spatial datasets and their metadata) is still stored centrally at the directory service. The central administration of the directory service leads to the same two issues similar to the C/S GIS architecture. (1) Frequent up-date: The directory-based distributed GIS architecture does not solve the C/S architecture's issues of frequent up-dating. In the case of connecting to highly dynamic data sources (e.g., wireless sensor networks), the directory service needs to keep the highly dynamic system status up-to-date, and all system loads caused by up-to-date operations will be added onto the central directory service. (2) Poor flexibility on administration: Although in this model different spatial data providers have the control over their own services and datasets, the discovery of the services and datasets is still controlled by the central directory service. Service providers still do not have direct control over making their spatial data resources published or unpublished.

2.3 Scalability, an Unavoidable Challenge for Today's Internet GIS

Thus far, we can argue that scalability is of significant importance and is an unavoidable challenge to a successful Internet GIS architecture, especially for today's highly dynamic spatial data sources, such as sensor networks. After examining the scalability of existing GIS architectures, we found that they are not able to fulfill the scalability challenges raised by the enormous amount of users and heterogeneous data sources. A new Internet GIS architecture that is able to scale up in order to accommodate this is very much needed. As a solution, we propose a fully decentralized architecture to build an Internet GIS. In the next section we introduce the core of this decentralized architecture, *i.e.*, our proposed Peer-to-Peer (P2P) spatial lookup framework, called the P2P Spatial Access Method (P2PSAM).

3. METHODOLOGY: THE POWER OF DECENTRALIZATION

Since the architecture's centralized components are the root of the above issues, we are motivated to remove these and to use a fully decentralized architecture to build an Internet GIS. Figure

1 shows a conceptual diagram of the fully decentralized Internet GIS architecture.

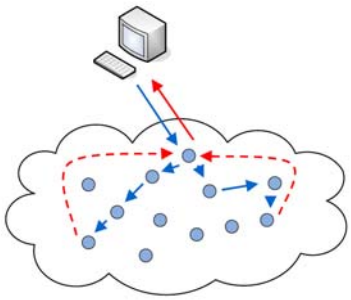

Figure 1. A conceptual architectural diagram of a fully decentralized Internet GIS.

Instead of depending on a centralized server to discover the available spatial datasets and the servers hosting the datasets, this decentralized architecture operates on a cooperative model, where the system's participating service nodes leverage each others' available resources (*i.e.*, CPU, storage, bandwidth, etc.) to perform the service discovery task.

From a client perspective, all participating service nodes, as a whole, form a single global directory service. At a conceptual level, compared to the centralized directory service, a decentralized design allows this fully-decentralized system to potentially scale to a very large amount of users, services, and datasets. This is because: (1) it is more reliable due to the fact that it has no single point of failure; (2) it achieves better load balancing by distributing the system load; and (3) it is more flexible because it takes discovery control away from the central service directory provider.

In this cooperative architecture, the participating nodes are peers in functionality and each participating node is potentially both a server and a client. Therefore this decentralized architecture can also be referred to as the peer-to-peer (P2P) architecture. Today P2P systems are emerging as a new paradigm for constructing large-scale distributed systems. Existing commercial P2P systems (e.g., Napster, Kazza, Skype) have proved P2P to be a useful architecture for building large scale (*i.e.*, millions of users) wide-area information systems (Guha, 2006; Milojicic, 2002). Next, we review the existing P2P architectures.

3.1 Review of Existing P2P Architectures

Our goal here is to find out whether existing P2P architectures are applicable and useful to build a fully decentralized directory service for an Internet GIS. According to their lookup mechanisms, existing P2P architectures fall into two categories: (1) an un-structured overlay network and (2) a structured overlay network. In the next section we review the two types of lookup mechanisms with a focus on its spatial lookup mechanisms, because a P2P-based spatial lookup mechanism is the core to build our target application, a fully decentralized GIS directory service.

3.1.1 Unstructured Overlay Networks

An un-structured overlay network is a P2P network where participating nodes perform actions for each other, where no rules exist to define or constrain connectivity between nodes. In this type of network, because of the lack of structure, maintaining full connectivity between nodes generally means that each node must be aware of, and maintain a route to, each and every node (*i.e.*, all possible destinations) in the network (Zhao, 2004). For this reason, the size of the routing information, which each node needs to keep, scales linearly with the size of the network. This is clearly a factor that limits the size of these networks.

In order to avoid this limitation, some un-structured overlay networks, such as Gnutella (Ripeanu, 2001) uses a controlled flooding mechanism to distribute query messages. The flooding search mechanism is simple. A message representing a query is sent to all neighbours of a participating node, and the node's neighbours again forward the query to all of their neighbours. As soon as a node receives a query matching its local data, it directly replies to the node that issued the original query message. To avoid infinitely circulating queries, a query is limited to being able to make only a certain number of hops.

In our Internet GIS context, we are concerned with whether the un-structured overlay network supports spatial query. These un-structured overlay network protocols support spatial query, because the lookup mechanism is not dependent on the nature of the query. The nodes flood the query messages without considering what is contained in the query message. When a node finds that its local data's spatial property fulfils the received spatial query constraints, it replies the result to the node that issued the spatial query.

Although the un-structured overlay network supports spatial query, it is not suitable as the underlying P2P infrastructure for building a scalable Internet GIS. We can summarize that the flooding mechanism induces two limitations: (1) It generates enormous amounts of network traffic due to its flood-based approach and as a result, the un-structured overlay networks are not scalable (Ritter, 2001) and (2) It does not guarantee that a request will be spread across the entire network due to its "controlled" flooding mechanism. Therefore, there is no guarantee as to the lookup result. Both of these critical limitations prevent us from using the un-structured P2P network as the underlying P2P infrastructure for an Internet GIS.

However, while we study the un-structured overlay network, there is one key observation worth noting. Using a flooding mechanism to perform spatial queries on a P2P network is similar to using sequential scanning (*i.e.*, without using spatial indexes) to perform spatial queries on a local disk. On a local disk, without using spatial indexes, in order to find all data items that fulfil the spatial query constraints, the query engine needs to sequentially scan and check all disk pages that store the records. In an un-structured overlay network, in order to find all data items that fulfil the spatial query constraints, the query engine needs to flood the query to all P2P nodes that store the records. The difference between the above two cases is that without employing spatial indexes, in order to answer the spatial query, on a local machine the disk pages are scanned; and on an un-structured overlay network the nodes are scanned. On a local machine, avoiding the scanning of disk pages can reduce disk Input/Output (I/O) numbers, and in a P2P environment, avoiding the scanning of nodes can reduce the number of messages being exchanged between the P2P nodes.

3.1.2 Structured Overlay Networks and Distributed Hash Tables

In order to avoid flooding the network with query messages, structured overlay networks have been proposed. These structured overlay networks use hash functions to build distributed indexes for their stored data items. The hash tables, like distributed indexes, successfully reduce the nodes to be scanned per query. Examples of such a P2P network include Chord (Stoica, 2001), Pastry (Rowstron, 2001), Tapestry (Hildrum, 2002), and CAN (Ratnasamy, 2001). Since they support distributed hash table (DHT) functions, they are referred as DHTs (Ratnasamy, 2002). DHTs support storing, updating, and retrieving [*key*, *value*] pairs across the P2P network.

DHTs provide two major advantages: (1) They offer simple hash table interfaces to manage communication between P2P nodes, and (2) They algorithmically scale up to millions and billions of nodes. For the above two reasons, DHTs have been considered suitable as the building block for the development of large scale network application infrastructures (Zhao, 2004). Example applications that use DHT as the underlying P2P infrastructure include event notification services, such as Scribe (Castro, 2002); Internet telephony system, such as Skype (Guha, 2006); and cooperative web caching systems, such as SQUIRREL (Iyer, 2002).

Although DHTs have the advantage of simple interfaces and scalability, they have one major limitation. They do not support complex queries. The DHTs' simple hash table interfaces only support exact-match [*key*, *value*] pair queries, that is, queries where the exact key (*i.e.*, a string) of the requested data object is known. That means DHTs do not support spatial query. However, in our Internet GIS context, we are especially interested in a P2P network for spatial query. Thus, in order to build an Internet GIS using a P2P network, the P2P network's support of spatial query is necessary.

In this section, we have reviewed two existing P2P architectures: (1) un-structured overlay networks and (2) structured overlay networks. The result of this review shows that neither of these existing P2P architectures can be considered suitable for building an Internet GIS. The unstructured overlay networks do not meet our requirements, because: (1) their flooding-based lookup mechanism is not scalable, and (2) the controlled flooding mechanism does not guarantee the query results. Although the structured overlay networks are scalable and their query results are guaranteed, they only support exact-match [*key*, *value*] pair queries. For an Internet GIS, we need the P2P network to support the processing of spatial queries.

4. PEER-TO-PEER SPATIAL ACCESS METHOD (P2PSAM)

In this paper, we propose Peer-to-Peer Spatial Access Method (P2PSAM) as our solution for a P2P spatial lookup mechanism. The goals of the P2PSAM, as described in the previous section, are (1) scalable on the number of spatial data objects, (2) load-balancing, (3) reliable, and (4) administratively flexible. In order to achieve these goals in a fully decentralized P2P environment, we must reduce the number of P2P nodes to be visited (*i.e.*, to avoid flooding query messages) while performing a spatial query in the P2P network.

Our proposed P2PSAM solution is motivated by the aforementioned key observation: Using a flooding mechanism to perform spatial query on a P2P network is conceptually similar to using a sequential scanning mechanism (*i.e.*, without using spatial indexes) to perform spatial query on a local spatial database.

In the case of the traditional spatial databases, it largely relies on Spatial Access Methods (SAM) (Seeger, 1988) as the primary mechanism for efficient search and retrieval. The core of the SAMs is a local spatial index, an auxiliary data structure that can be implemented based on other primitive data structure, such as B-tree. Thus, motivated by the SAM's spatial index approach, we have the following research question to answer:

Local spatial databases use a primitive data structure, such as B-tree, to build and maintain a spatial index in order to reduce the disk pages to be sequentially scanned per spatial query. Can we thus use a similar concept, such as a primitive distributed data structure, such as the DHT, to build and maintain a distributed spatial index, in order to reduce the P2P nodes to be visited per spatial query?

P2PSAM is our proposed solution to the above research question. P2PSAM solves the research question by building, accessing, and maintaining spatial indexes using the DHT. In P2PSAM, a distributed spatial index is built on a collection of entries. Each entry is an [*mbbox*, *endpoint*] pair. The pair's endpoint is a URL endpoint linking to a P2P node. The P2P node hosts a collection of spatial data objects, and the spatial extent of the data objects can be approximated by the pair's *mbbox*. Knowing the *endpoint* allows us to directly access, via the Internet, the physical P2P node that maintains the physical representation of the spatial data objects.

Next, we introduce the P2PSAM spatial query framework, which consists of three steps, in order to locate the spatial objects stored in the P2P network.

4.1 Three Steps of the P2PSAM Search Operation

Here we introduce the P2PSAM. P2PSAM is a multi-step spatial query processing mechanism. It consists of three steps: (1) Filter Step, (2) Forward Step, and (3) Refinement Step.

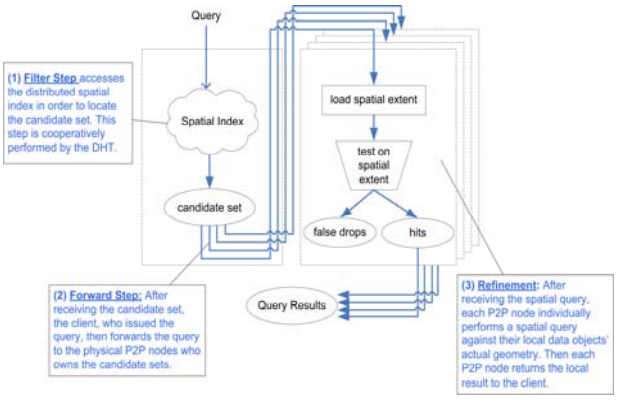

Figure 2 Three steps of the P2PSAM search operation

4.1.1 Filter Step

The purpose of the filter step is to find a set of entries in which each entry's mbbox satisfies the spatial query constraints. This

filter step consists of two sub-steps: (1) using the DHT interfaces to retrieve the distributed index and find the entries whose mbboxes potentially fulfilling the spatial query constraint, and (2) applying the spatial query constraint on the mbbox of each entry.

The result of the filter step is called the candidate set. The candidate set is a collection of endpoints, whose corresponding mbboxes satisfy the spatial query constraints. These endpoints are links (e.g., a URL) pointing to the physical P2P nodes hosting some data objects that "potentially" satisfy the spatial query constraints.

The filter step is performed cooperatively by the underlying DHT. By using DHT to access the distributed spatial index, we can exploit the advantages of the DHT, such as load balancing, resilience, and scalability.

4.1.2 Forward Step

In this forward step, the query initiator, according to the candidate set's endpoints, directly forwards the spatial query message to the candidate P2P nodes. The candidate P2P nodes, who received the forwarded queries, then perform the refinement step (introduced below) independently and in parallel.

4.1.3 Refinement Step

In the refinement step, candidate P2P nodes, after receiving the spatial query message, execute the spatial query locally against the actual geometries of their hosting data objects. The refinement step is needed because a P2P node's mbbox is the approximation of the data objects hosted by the P2P node. The P2P node's mbbox might satisfy the query, but its spatial data objects' exact geometries might not. Those data objects, which pass the filter step but not the refinement step, are called false drops. After finishing the refinement step, each P2P node then returns the results to the query initiator. It is worth noting that each physical P2P node may maintain a secondary local spatial index for their hosting data objects in order to accelerate the locally performed refinement step.

The above introduced P2PSAM spatial query framework can be applied to most existing spatial indexes, such as Quad-Tree family, R-Tree family, Hierarchical Triangular Mesh (HTM), etc (Samet, 2006). As a proof of concept, we implemented P2PSAM using Linear Quad-Tree (LQT) and an open source DHT, called Pastry (Rowstron, 2001). We call this distributed LQT in the P2PSAM setting: P2PLQT. In next section, we present a preliminary result of a P2PLQT scalability experiment.

5. RESULTS

Our interest in P2PSAM is based on the belief that a P2P spatial lookup mechanism, such as the P2PSAM, would provide a new architectural framework for building an Internet GIS. As a proof of concept, we designed and implemented GeoSWIFT 2.0 as an example of how to use P2PSAM to build a fully decentralized Internet GIS. Details about GeoSWIFT 2.0 can be found in (Liang, 2008). In this section, we present the result of our scalability experiment.

We use the computational and networking overhead associated with each spatial query operation as the scalability measure of P2PLQT. Since all of the P2PLQT queries are based on DHT's hash table interfaces, our metric in measuring this overhead uses the number of DHT operations per spatial query.

As input for the experiment, we use the five digits US ZIP code boundary data set (Figure 3). In order to establish the environment for the experiment, we repeatedly inserted the bounding box of each polygon using a random displacement of 1~5 km until the data size reached our target size. After inserting enough bounding boxes into the system, we then submit 1,000 window queries against the system. Each query window size is 5 km x 5 km. The geographical distribution of the window queries follows the US population distribution (*i.e.*, the areas with higher population density gets more queries than the areas with lower population density).

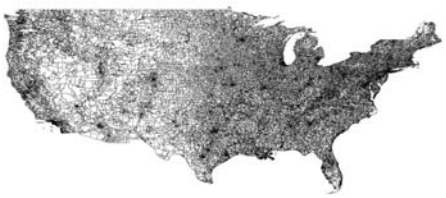

Figure 3 Test datasets: five digits US ZIP code

Figure 4 shows the result of the experiment. We vary the number of polygons inserted into the P2P network from 1 million to 3 million. The result shows that the number of the DHT operations grows consistently and fluidly with the number of polygons in the system. Regardless of the number of polygons in the system, most of the window queries can be answered within 30 DHT operations.

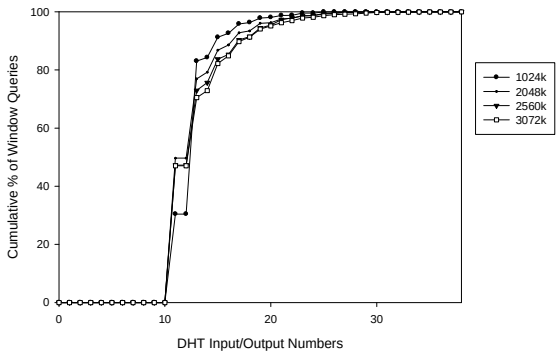

Figure 4 A cumulative distribution function (CDF) plot of the percentage of window queries as a function of the DHT I/O numbers

6. CONCLUSIONS

The major contribution of this paper is the P2PSAM, a new fully decentralized Peer-to-Peer GIS architecture. P2PSAM is a new architectural framework for building an Internet GIS.

In this paper, we have reviewed the two existing Internet GIS architectures, namely C/S GIS architecture and directory-based distributed GIS architecture. After examining the scalability of

the two architectures, we found that they cannot meet the scalability challenges raised by today's enormous amount of GIS users, the heterogeneous spatial data, and the highly dynamic data sources (e.g., wireless sensor networks).

In our review, we found the architectures' centralized components (e.g., the central GIS directory service) are the root that limits their scalability. We are motivated to remove these centralized components and to use a fully decentralized architecture to build an Internet GIS. In the second part of this paper, we reviewed the existing P2P architecture with the goal to find out whether existing P2P architectures are applicable and useful to build a fully decentralized directory service for an Internet GIS. We have concluded that both the un-structured and structured overlay networks are not suitable for building a scalable Internet GIS. Then as a solution, we have proposed the P2PSAM for dealing with scalability challenges. In the later section, we've described the P2PSAM framework. P2PSAM removes the centralized components by integrating dynamic spatial indexing algorithms (e.g., QuadTree) with fully decentralized P2P networks (e.g., Distributed Hash Table). Our experiments demonstrated that P2PSAM is able to scale to several millions of spatial data objects in a P2P environment without significantly sacrificing system performance.

REFERENCES

Castro, M., Druschel, P., Kermarrec, A.-M. and Rowstron, A., 2002. *SCRIBE: A Large-scale and Decentralised Application-level Multicast Infrastructure*. IEEE Journal on Selected Areas in Communication (JSAC), 20(8).

Guha, S., Daswani, N. and Jain, R., 2006. *An Experimental Study of the Skype Peer-to-Peer VoIP System*, The 5th International Workshop on Peer-to-Peer Systems, Santa Barbara, CA.

Hildrum, K., Kubiatowicz, J.D., Rao, S. and Zhao, B.Y., 2002. *Distributed Object Location in a Dynamic Network*, 14th ACM Symposium on Parallel Algorithms and Architectures (SPAA), Winnipeg, Manitoba, Canada, pp. 41-52.

Iyer, S., Rowstron, A. and Druschel, P., 2002. *SQUIRREL: A Decentralized, Peer-to-Peer Web Cache*, 12th ACM Symposium on Principles of Distributed Computing (PODC 2002), Monterey, California, USA.

Liang, S.H.L., 2008. *A New Peer-to-Peer-based Interoperable Spatial Sensor Web Architecture*, The XXIth ISPRS Congress, Beijing, China.

Milojicic, D.S. et al., 2002. *Peer-to-Peer Computing*. Technical Report HPL-2002-57, HP Lab.

Neuman, B.C., 1994. Scale in Distributed Systems. In: T. Casavant and M. Singhal (Editors), *Readings in Distributed Computing Systems*. IEEE Computer Society Press, Los Alamitos, California, pp. 463-489.

Putz, S., 1994. *Interactive Information Services using World-Wide Web Hypertext*, First Conference on World-Wide Web, Geneva, Switzerland, pp. 273-280.

Ratnasamy, S.P., 2002. *A Scalable Content-Addressable Network*. Ph.D. Thesis, University of California at Bekeley, Berkeley, 93 pp.

Ratnasamy, S.P., Francis, P., Handley, M., Karp, R. and Shenker, S., 2001. *A Scalable Content-Addressable Network*, ACM SIG-COMM, San Diego, California.

Ripeanu, M., 2001. *Peer-to-Peer Architecture Case Study: Gnutella Network*, First International Conference on Peer-to-Peer Computing (P2P'01), Linkoping, Sweden, pp. 0099.

Ritter, J., 2001. *Why Gnutella can't scale. No, Really*. http://www.darkridge.com/~jpr5/doc/gnutella.html (accessed 10 May 2008).

Rowstron, A. and Druschel, P., 2001. *Pastry: Scalable, Decentralized Object Location and Routing for Large-Scale Peer-to-Peer Systems*, 18th IFIP/ACM International Conference on Distributed Systems Platforms (Middleware 2001), Heidelberg, Germany.

Samet, H., 2006. *Foundations of Multidimensional and Metric Data Structures*. Morgan Kaufmann, San Francisco, California, USA, 993 pp.

Seeger, B. and Kriegel, H.-P., 1988. *Techniques for Design and Implementation of Efficient Spatial Access Methods*, Proceedings of the 14th International Conference on Very Large Data Bases. Morgan Kaufmann Publishers Inc., pp. 360-371.

Stoica, I., Morris, R., Karger, D., Kaashoek, M.F. and Balakrishnan, H., 2001. *Chord: A Scalable Peer-to-Peer Lookup Service for Internet Applications*, ACM SIGCOMM 2001, San Diego, California, pp. 149-160.

Tewari, E., 1998. *Architectures and Algorithms for Scalable Wide-area Information Systems*. Ph.D. dissertation Thesis, The University of Texax at Austin, Austin, 171 pp.

Zhao, B.Y., 2004. *Decentralized Object Location and Routing: A New Networking Paradigm*. Ph.D. Thesis, University of California, Berkeley, 216 pp.