An Efficient Spatial Publish/Subscribe System for Intelligent Location-Based Services

Xiaoyan CHEN
IBM China Research Laboratory
Beijing, 100085, China
xiaoyanc@cn.ibm.com

Ying CHEN
IBM China Research Laboratory
Beijing, 100085, China
yingch@cn.ibm.com

Fangyan RAO
IBM China Research Laboratory
Beijing, 100085, China
raofy@cn.ibm.com

ABSTRACT

The advance in wireless Internet and mobile computing brought the booming of intelligent Location-Based Services(LBS), which can actively push location-dependent information to mobile users according to their predefined interest. The successful development of push-based LBS applications relies on the existence of a publish/subscribe middleware that can handle spatial relationship. This paper presents an efficient spatial publish/subscribe system that can serve as the middleware for intelligent LBS applications. The basic models, including spatial event model, spatial subscription model and notification model, are introduced and the overall architecture is presented. Two kinds of spatial predicate that can meet most common requirement of intelligent location aware applications are also discussed. Furthermore, we propose a novel spatial event processing approach that dispatches the spatial subscriptions to self-positioning mobile devices. By leveraging client-side computing resource and decreasing the communication times, the server-side workload is relieved and the communication cost is reduced. Experimental results clearly demonstrate the efficiency of our approach.

Keywords

Spatial Publish/Subscribe, Location-based Service, Event-based Systems

1. INTRODUCTION

Recent years have seen rapid growth in the area of Location-based Services (LBS). The computing or service model for current LBS is based on pull model or user-initiated model, which means a user sends requests to a server which replies location dependent answers. Traditional location aware services, such as mobile yellow page service (nearby service), are in the scope of this model. For example, a user driving a car requests the location of the nearest gas station.

With advances in mobile Internet technologies and increasing competitive pressure, more and more service providers are beginning to deploy intelligent LBS (iLBS) based on push model or service-initiated model. In service-initiated model, a service provider actively pushes the location dependent information to mobile users via different channels according to their predefined interest in the system. Service-initiated model brings a new user experience to mobile users. It can facilitate mobile users in retrieving information without explicitly sending a request each time. For example, two interesting iLBS applications could be:

- E-coupon. A shopping mall sends a promotion message to nearby mobile users who are potential consumers. The E-coupon service attracts more customers and increases the revenue.
- Mobile Buddy-List. User gets notification when a friend in his buddy list is nearby. Mobile buddy-list attracts more users for wireless carriers because it brings their customers a new experience with cell phones.

Applications listed above could be implemented based on the publish/subscribe paradigm (hereafter referred to as pub/sub), where information providers publish units of information called events, and information consumers issue subscriptions specified by predicates on the content of the events. The pub/sub middleware ensures the timely delivery of published events to all interested subscribers [1]. From iLBS perspective, events are the stream of location of mobile users or devices and subscriptions are spatial predicates on location events. An example of spatial subscription is *"Send e-coupon to potential customers while they are within 200 meters from McDonald today"*. In this example, the spatial condition is "location of user within 200 meters of location of McDonald". Most existing pub/sub systems such as Gryphon [2], Siena [3] can match address information in text format. Spatial matching needs extra mechanism such as spatial index to handle 2 or 3 dimension information, which is not well addressed by current pub/sub systems. This motivates us to provide a spatial pub/sub system that can match user pre-defined spatial condition (subscription) against events received by the system and take appropriate action whenever there is a match.

In the context of intelligent location-aware applications, a person's current location is obtained by positioning his/her mobile device. There are a number of positioning technologies that can be classified into two categories: terminal-based and network-based solutions. The former builds significant intelligence into the handset for positioning while the latter builds more intelligence into the mobile network infrastructure. For network-based positioning approaches, a central tracking service is usually provided to continuously retrieve location of mobile devices or receive location reported from the mobile network infrastructure. Then the tracking service publishes location as events to the spatial pub/sub system. For terminal-based positioning approaches, mobile devices, such as a PDA with a GPS receiver, continuously publish their own location as events to the spatial pub/sub system. While supporting a large number of mobile users, performance is a big challenge for the spatial pub/sub system. A novel spatial event processing method is proposed to improve performance. The basic idea behind the method is that the spatial pub/sub server dispatches spatial subscriptions to mobile devices when mobile devices have the capability of self-positioning, computing and networking. We call this kind of devices as intelligent devices. Intelligent devices process spatial subscriptions based on their own location and report matching spatial events to the server.

In this paper, an efficient spatial pub/sub system is presented. The system has been implemented in Java at IBM China Research Lab as a prototype. Our main contributions focus on: (1) a spatial pub/sub middleware that supports spatial matching for push-based location aware applications and (2) a high performance spatial event processing mechanism based on close cooperation between intelligent devices and the spatial pub/sub server.

This paper is organized as follows. Section 2 introduces three models: spatial event model, spatial subscription model and notification model. Overall architecture of the spatial pub/sub system and components are presented in section 3. Section 4 focuses on the performance analysis. Section 5 briefly surveys related work. Finally we conclude the paper and propose further work.

2. MODELS

The models used by Pub/Sub engine include Spatial Event Model, Spatial Subscription Model and Notification Model. This section details these models and gives their solid semantic in the context of a spatial pub/sub system.

2.1 Spatial Event Model

A spatial event is described by a set of properties specified by name-value (NV) pairs. The three mandatory properties for a spatial event are:

- *Object identifier (OID)* is a unique identity indicating the owner of the location. The object is either a person, a device, or anything that can be located.
- *Timestamp (TS)* is the time when the object is positioned.
- *Location (LOC)* is the geographical location specified by predefined spatial reference system (SRS) or textual description of location that could be translated into geographical location via geocoding.

Three properties above describe three most important dimensions pertained to a spatial event: Who, When and Where. Other optional properties could also be introduced to facilitate the processing of spatial events, such as

Table 1: A Spatial Event Sample

Name	Value
OID	foo@acme.com
TS	2002-05-01-13:20:05
LOC	(119.323,38.465)
LPID	CMCC
Priority	High

Table 2: A Notification Example

Name	Value
OID	foo@acme.com
TS	2002-05-01-13:20:05
LOC	(119.323,38.465)
LPID	CMCC
Priority	High
Mode	Enter
Target	Time Square

- *Uncertainty (UNC)* describes the uncertainty of the location detected.
- *Location provider identifier (LPID)* is the provider of the location information.

Besides the predefined mandatory and optional names for properties, other properties could be attached to the spatial event to describe domain or application specific information. Usually this information is intended for the subscription application and not processed by the Pub/Sub engine.

Table 1 is a sample instance of spatial event. The spatial event describes that the mobile user foo@acme.com is at longitude 119.323 and latitude 38.456 at time 2002-05-01-13:20:05, and the location information is provided by CMCC (a mobile carrier). There is also a domain-specific property called Priority with value High to indicate that it is an important event.

2.2 Spatial Subscription Model

Spatial subscription is used by subscribers to express their interest in spatial events. In the spatial pub/sub system, a spatial subscription is defined as (SP,ToS) tuple, where SP is spatial predicate defined upon location of objects and ToS stands for Type of Services, which will be discussed later. The semantic for a spatial subscription is: a notification (based on notification model) is send to a subscriber when an incoming spatial event (based on spatial event model) meets SP and ToS requirement. Currently two kinds of SP are supported in the spatial pub/sub system:

- *Within* predicate has the syntax *(oid-1,oid-2,,oid-n) within (zone-1,zone-2,zone-n)*. The predicate is true if and only if one of the location of the mobile user (*oid-i*) is within one of the zone (*zone-j*). A zone is used to represent an interested region. Subscription using within predicate can be used to support E-coupon case mentioned before.
- *Distance* predicate has the syntax *(oid) distance (D, oid-1,oid-2,,oid-n)*. The predicate is true if and only if distance between oid and oid-i ($1 \leq i \leq n$) is less than D. Subscription using distance trigger can support services like Mobile Buddy List through which a user can

define his buddy list and ask for an alert when any of his buddy is near to him.

We believe that the two SPs could meet large amount of requirement of LBS applications and we are investigating other SPs for constructing more complex location-aware application. Traditional pub/sub systems are based on the "publish-match-notify" operational cycle, therefore, a spatial event is always sent to subscribers when a spatial subscription is satisfied by the event. In the context of location aware applications, however, the event filter mechanism could sometimes confuse the user. For example, a user defines a *within* subscription "(oid1) within (zone1)", where zone1 is the predefined area of a shopping mall. While the user (oid1) is entering the zone (zone1), he receives a promotion message. The *within* subscription is always evaluated to be true when the user stays in the shopping mall, so that he continuously receives the promotion message. Obviously in this case only one promotion message makes sense both to the user and the promotion provider. One-time semantic of subscription specified by ToS is introduced to guarantee that the user receives only one promotion message. While the ToS is set to once, one-time semantic of subscription is applied by spatial matching engine.

2.3 Notification Model

Notification takes the same NV pairs format as event model and includes two parts:

1. *Original event part* copied from the event information.
2. *Subscription-specific part* derived from the process of matching subscriptions with events.

Take the *within* subscription with One-Time semantic as an example, a notification is described in Table 2. The first five items are copied from the event and the remaining two items are subscription-specific properties. Property Mode with value "enter" indicates that the user is currently entering the defined zone, if the user is leaving the zone, the property has the value "leave". Property Target with value "Times Square" indicates that the user is entering the zone Times Square. The Target property is used to distinguish the triggered zone (Times Square) from other zones defined in the subscription. For the distance subscription, two properties named "Target" and "Target Location" are introduced to express the identifier of a matching user and her/his current location.

3. ARCHITECTURE

This section details the overall architecture of the spatial pub/sub system. The pub/sub system adopts a novel client-side event processing approach to improve performance. To begin with, we introduce this approach.

3.1 Client-side Event Processing Approach

Due to the intrinsic constraints of mobile network, the bandwidth of wireless communication is limited. How to utilize bandwidth of wireless communication in an economic way is a challenge for the system design. When spatial event matching is handled in the central pub/sub server, intelligent devices should continuously publish their own location to the server. This consumes the bandwidth of wireless communication and restricts the concurrency of the server. A novel client-side event processing approach is presented to solve this problem.

This approach takes full advantage of the resource capabilities (such as computing, storage and positioning) in intelligent mobile devices. Its work flow is shown below:

1. The spatial pub/sub server dispatches *within* subscriptions to a related intelligent device;
2. The intelligent device gets its location from the embedded positioning module and performs spatial matching instead of reporting the location to the pub/sub server;
3. If the spatial subscription is satisfied, for example, if a user is entering a predefined zone, a notification is sent to the pub/sub server. Otherwise, the location is discarded.

Our event processing approach relieves the workload of the pub/sub server for spatial matching is handled in client sides. Only matching spatial events are sent to the pub/sub server, so the bandwidth of wireless communication is saved and the concurrency of the pub/sub system is improved.

3.2 Overall Architecture

Figure 1 depicts the overall architecture for the spatial pub/sub system. The system is highly modular with clearly defined interfaces. The key components include Spatial Pub/Sub Manager, Spatial Matching Engine, Zone Definition Engine, Location Agent Controller and Mobile Location Agent. Location Agent Controller and Mobile Location Agent work together to implement the proposed approach. In addition, privacy protection is considered to be the critical issue in LBS. The architecture also provides Privacy Manager component that provides Privacy API to handle privacy issues and allows users to control who and what applications can access their location. The discussion of privacy protection is beyond the range of this paper and it can be referred to in [4,5].

Spatial Pub/Sub Manager is responsible for managing event subscription/publish and exposing JMS (Java Messaging Service [10]) interface. Message selectors from JMS are extended to support spatial subscription, whose syntax was outlined in Section 2. For a subscription request, privacy check should be done first. Instead of forwarding all subscriptions to Spatial Matching Engine, the Pub/Sub Manager exploits the sematic of the subscriptions and sends *within* subscriptions to Location Agent Controller. As a result, the workload of Spatial Matching Engine is relieved.

Spatial Matching Engine takes charge of filtering spatial events. For achieving high performance in spatial matching, the engine uses spatial index technique to accelerate the matching process. For *within* predicate, R-tree is employed to index predefined zones based on their MBR (Minimum Bounding Rectangle) and transform subscription evaluation to R-tree searching. Each zone maintains a hash table to record the list of interested users. When a mobile user's location is within a zone and the mobile user is in the hash table of the zone, a notification is sent out. *Distance* evaluation involves more than one mobile user; the location cache is provided to store latest location data of those users

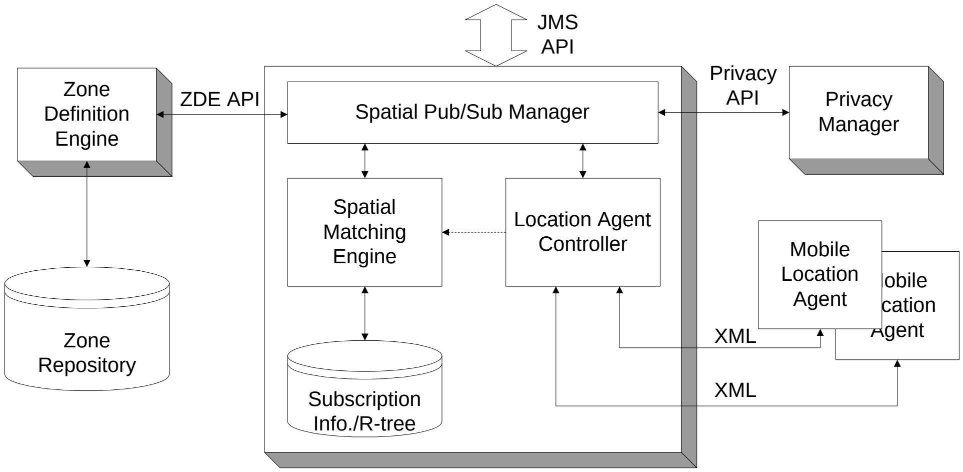

Figure 1: Spatial Publish/Subscribe System Architecture

and an object trigger graph mechanism is employed to handle the case. Spatial Match Engine reuses the module called Trigger Handler from CAMEL project [6]. The detailed algorithms and performance comparison result can be found in [6]. The result shows that the matching engine has good performance and scalability.

Zone Definition Engine is used by the system administrator and/or end users to define well-known ZOI (Zone Of Interest) or user-specific ZOI, which could be a polygon (including rectangle) and a circle. For example, in Beijing, hot places such as Xidan can be predefined in system as a polygon based on its geographic coordination (latitude, longitude), and each user can define his/her home according to its geographic location. Predefined ZOI can be referred in system as a symbol name, like SYSTEM.Xidan and Mike.HOME, where the prefix SYSTEM stands for system defined ZOI and otherwise is the name of the user who defines the ZOI. Zone Definition Engine exposes ZDE API for client applications and Spatial Pub/Sub Manager to manipulate zones.

Location Agent Controller manages all Mobile Location Agents running on intelligent devices. It provides authentication mechanism for these Agents, sends related spatial subscriptions to them, receives the matching spatial events from them and forwards those matching events to Spatial Pub/Sub Manager. When the mobile user involves *distance* predicates, the Controller sends periodical location report command to the Agent, receives location of the mobile user periodically and passes them to Spatial Matching Engine. The interface between to the Controller and the Agent is base on XML.

Mobile Location Agent runs on intelligent devices. It focuses on obtaining its location information from embedded positioning module such as GPS and handling *within* predicates sent by the Controller. It evaluates spatial predicates based on the current location of the mobile device. When a spatial predicate is evaluated to be true, the matching spatial event will be sent to the Controller. Occasionally Mobile Location Agent can send the location of the mobile device periodically according to instructions from the Controller.

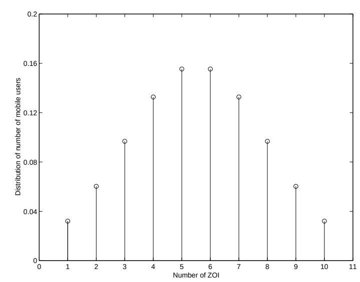

Figure 2: Distribution of ZOIs for Mobile Users

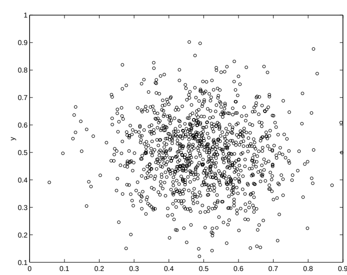

Figure 3: Centers of ZOIs

4. PERFORMANCE

Since the bandwidth of wireless communication is restricted, communication times is an important measurement factor for performance of the spatial pub/sub system. Additionally, workload of CPU-intensive matching influences performance of a spatial pub/sub server. In our proposed method, matching operation is handled in the client sides so the matching cost of the server is zero. Therefore only the communication times is taken into consideration in following experiments. Taking *within* predicate as an example, we use simulated data to obtain the communication times. The scenario is: mobile users with various speed walks randomly

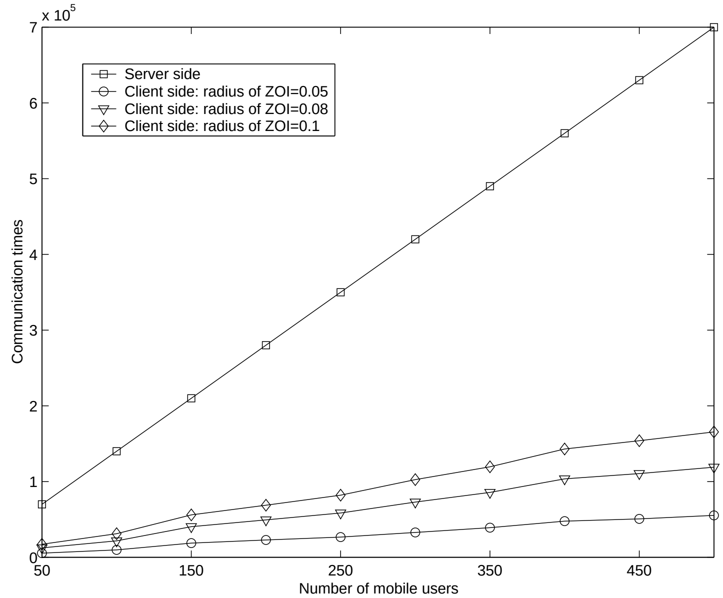

Figure 4: Communication times vs. Number of Mobile Users

in the fixed area normalized with size 1. Each mobile user could define one or more of ZOIs, and the number of ZOIs defined by each user follows the normal distribution as illustrated in Figure 2. All ZOIs are circles and the centers of circles follow spatial normal distribution depicted in Figure 3. Figure 4 shows how the performance of the system in terms of the communication times changes with the number of mobile users grows. In this case the total sample times for each mobile user is 1400. Using server-side matching approach, the communication times increases linearly against the number of mobile users, for each spatial event is sent to Spatial Matching Engine for evaluating. Using client-side matching approach, the communication times decreases dramatically compared with server-side approach, because only the matching events are sent to the server. Also from the performance chart we observe that the size of ZOI also influences the matching events, and when the radius of ZOIs grows, the matching events increase, so does the communication times in our experiment.

By leveraging computing resource of intelligent devices, the proposed novel event processing approach not only relieves server-side workload, but also reduces communication times and saves bandwidth of wireless communication, thus make the system support more concurrent mobile users with less cost.

5. RELATED WORKS

Gryphon [2] and Siena [3] are content-based pub/sub systems. Gryphon provided an efficient and scalable filtering algorithm to handle event matching. Siena provided expressive subscription language for subscribers to select interested events. Both of them supported primitive data types while our spatial pub/sub system can handle complicated spatial data type.

Ivana Podnar et al. presented an architecture to deliver content to mobile users based on the publish/subscriber paradigm [7]. Their publish/subscribe system can work together with location management to deliver location-aware message to mobile users. However, the spatial pub/sub system proposed in this paper focuses on spatial-related information matching and performance improvement.

An event specification language that can be used to express spatial events was presented in [8]. Semantics of basic spatial events is the same as ours. They paid more attention to event definition and event composition while we took a great effort on spatial subscription and spatial event matching.

Work at Cambridge has investigated services that, based on registrations of interest in user locations and proximities, notifies clients when changes occur [9]. Their system architecture CALAIS was based on distributed events technology and used a dynamically modifiable R-tree index, fed with a stream of location events, to monitor locations and proximities. CALAIS was suitable for the support of context-aware applications operating within a typical indoor, office domain while our spatial pub/sub system is more suitable for outdoor environment by leveraging intelligent devices to provide a high performance spatial event processing.

6. CONCLUSIONS AND FUTURE WORK

Push-based location-based services bring new user experience to mobile users and provide new revenue opportunities to carriers. Spatial pub/sub paradigm is a key enabler to support push-based LBS applications, where event providers publish spatial events that are location of mobile users and event consumers subscribe their interest in spatial events. Most existing pub/sub systems cannot handle such spatial subscriptions directly. In this paper, we present an efficient spatial pub/sub system that can match user pre-defined spatial condition against spatial events based on spatial index technique. The system provides subscription mechanism for location of mobile users. A novel client-side spatial event processing method is proposed by dispatching the spatial subscription to intelligent mobile devices and leveraging their self-positioning and computing resource. The advantage of our method resides on reduced communication times and improved system concurrency. Experimental results clearly demonstrate the efficiency of our approach.

Our future work includes building spatial Pub/Sub system that supports more advance spatial predicates and functions, as well as aggregating non-spatial Pub/Sub (such as Gryphon) and spatial Pub/Sub to support subscription composing by both non-spatial and spatial condition.

7. ADDITIONAL AUTHORS

Additional authors: Xiulan YU (IBM China Research Laboratory, email: `yuxl@cn.ibm.com`) and Dong LIU (IBM China Research Laboratory, email: `liudong@cn.ibm.com`).

8. REFERENCES

[1] Lukasz Opyrchal, et.al., Exploiting IP Multicast in Content-based Publish-Subscribe Systems, IFIP/ACM International Conference on Distributed systems platforms, 2000 , New York, New York, United States.

[2] Marcos K. Aguilera, Robert E. Strom, Daniel C. Sturman, Mark Astley, and Tushar D. Chandra, Matching Events in a Content-Based Subscription System, Proceedings of Principles of Distributed Computing (PODC '99), Atlanta, GA, May 1999.

[3] A.Carzaniga, D.S.Rosenblum, A.L.Wolf, Achieving Scalability and Expressiveness in an Internet-Scale Event Notification Service, Nineteenth ACM Symposium on Principles of Distributed Computing (PODC2000), Portland OR. July, 2000.

[4] Narayanan, A., Realms and States: A Framework for Location Aware Mobile Computing, WMC 01 Rome Italy, 2001

[5] Einar Snekkenes, Concepts for Personal Location Privacy Policies, EC 01, October 14-17, 2001, Tampa, Florida, USA.

[6] Ying CHEN, Fangyan RAO, Xiulan YU and Dong LIU, CAMEL: A Moving Object Database Approach for Intelligent Location Aware Services, Proceedings of International Conference on Mobile Data Management, Jan.21-24 2003, Melbourne, Australia.

[7] Ivana Podnar,Manfred Hauswirth and Mehdi Jazayeri. Mobile Push: Delivering Content to Mobile Users. In Proceedings of the 22nd International Conference on Distributed Computing System Workshops (ICDCSW'02)

[8] Martin Bauer and Kurt Rothermel, Towards the Observation of Spatial Events in Distributed Location-Aware Systems, Proceedings of the 22nd International Conference on Distributed Computing Systems Workshops (ICDCSW'02)

[9] G. Nelson, Context-Aware and Location Systems. PhD thesis, University of Cambridge Computer Laboratory, 1998

[10] Java Messaging Service, http://java.sun.com/products/jms/