

空间事件检测的加速策略研究

胡佳锋^{1,2} 金蓓弘¹ 陈海彪¹

¹中国科学院软件研究所, 北京 100190

²中国科学院研究生院, 北京 100190

摘 要: 随着物联网的发展, 基于位置的物联网应用, 如用户动态位置提醒服务、商家基于位置的促销服务等, 备受关注, 应用前景广泛。这些应用的关键是空间事件的检测。本文针对这些场景, 挖掘了区域之间的关系, 为其建立多级索引, 提升了空间事件的处理速度, 另一方面, 利用用户所持有的智能终端的处理能力, 构造并维护安全区域, 从而通过在客户端过滤事件, 减轻了服务器的事件检测负载。论文通过实验展示了采用上述加速策略的系统在若干应用场景中的性能和开销, 实验数据显示上述加速策略能有效地加速空间事件的检测。

关键词: 空间事件; 事件检测; 安全区域; 空间索引

1. 引言

物联网的发展给现有的互联网尤其是移动互联网带来了新的发展契机。随着带 GPS 等定位设备的智能终端的普及, 带有位置属性的事件即空间事件可以方便地形成, 部分移动应用通过 LBS (Location Based Service) 给用户带来了全新的互联网体验, 如 Google 通过 Google Map 和 GPS 给用户精确的定位; 大众点评给用户附近餐饮场所等的查询。然而, 传统的 LBS 往往采用拉模式或者说用户发起的模式, 这就意味着用户为获得相应的服务, 需要先发送一个请求, 然后等待服务器的响应。比如当一个用户驾车时, 需要寻找附近的加油站的信息, 这时, 他需要停车并打开相应的应用进行查询。为了能够得到更好的用户体验, 推送模式或者服务发起的模式逐渐为业界所推崇, 而发布/订阅系统本身的特性正好适合完成此类物联网服务, 可以在其中扮演重要的空间事件检测和通知的角色。在发布/订阅系统中, 事件给出了订阅者感兴趣的事物的状态, 订阅描述了订阅者的兴趣。系统负责检测事件是否与订阅匹配, 并将满足订阅的事件推送给订阅者。但当前已有的发布/订阅系统对事件的空间关系的支持都比较弱, Hermes[1]、Gryphon[2]等只提供了检测以属性-值对方式描述的原子事件的功能, 而 Siena[3]、REBECA[4]、PADRES[5]等系统提供了对事件的逻辑关系和时序关系的简单支持, 都不能满足实际物联网对空间事件检测的需求。

本文在调研了多种应用需求的基础上, 针对一些基于一元位置订阅的物联网应用场景, 提出了空间事件检测的加速机制, 即, 通过客户端的安全区域管理和服务器端的空间索引, 减轻服务器的事件检测负载, 提高空间事件检测的处理速度。

下文按如下方式组织: 第二节概述了空间事件的检测需求、基本方法和加速思路, 第三节讲述了空间区域的识别, 第四节描述了空间索引的建立, 第五节给出了对加速策略的性能和开销评估。最后一部分是全文的总结。

资助项目: 国家自然科学基金项目(60970027)

联系作者: 金蓓弘, E-mail: Beihong@iscas.ac.cn

2. 空间事件的检测

空间事件的检测需求广泛存在且多种多样。例如，城市生活中，乘坐出租车是必不可少的，出行的用户希望能及时获取附近空载出租车的信息。一旦附近有空载出租车出现，用户就能得到通知的话，可以方便用户的出行。[6]分析了与位置相关服务的应用需求，介绍了两个具体的应用场景：商场的电子广告（E-coupon）与移动好友列表（Mobile Buddy-List），这里，E-Coupon 是指商家基于位置的促销服务，商场发现其周围有顾客出现，就可以给其发送商品广告；Mobile Buddy-List 是一个附近好友通知服务，用户可以及时获得在其附近出现的好友的列表。[7]讨论了空间位置提醒（Spatial Alarm）需求，即，用户在他提交的空警报（类似订阅）中指定他所关心的空间区域，当有移动目标例如用户自己进入该空间区域时，给出报警通知。[8]是 IBM 关于它开展空间事件处理的早期报告，其中介绍了车队管理中的空间事件检测需求。

我们在调研了多种应用需求的基础上，认为一个事件与一个区域间的关系最为常用，这包括事件是否在某区域内以及事件与某个区域的距离。我们用 $@(R)$ 、 $DIST(R;<;L)$ 、 $DIST(R;=;L)$ 、 $DIST(R;>;L)$ 空间操作符来描述上述关系， R 表示一个区域， L 是距离常数。若用 A 表示一个订阅，用 a 表示满足订阅 A 的事件，那么，订阅 $@(R)A$ 表示用户关心在区域 R 内发生的满足订阅 A 的事件；订阅 $DIST(R;<;L)A$ 表示用户关心满足订阅 A 的事件 a ，且事件 a 的发生在区域 R 之外且与 R 的距离小于 L 。这几类订阅也称为一元位置订阅。

我们用点表示事件的发生位置，用凸多边形表示区域，那么，为了判断事件发生的位置是否属于所指定的区域，可以将事件与区域的关系判断转化为下列三种判断：点在区域内、点在区域上、点在区域外。具体而言，如果某条边 P_iP_{i+1} 包含了点 X ，即 $\overrightarrow{P_iP_{i+1}} \times \overrightarrow{P_iX} = 0$ 、 $(P_i.x - X.x) \times (X.x - P_{i+1}.x) \geq 0$ 、 $(P_i.y - X.y) \times (X.y - P_{i+1}.y) \geq 0$ 成立，那么点在凸多边形上；如果对于凸多边形的任意一条边 P_iP_{i+1} 都有 $\overrightarrow{P_iP_{i+1}} \times \overrightarrow{P_iX} > 0$ ，那么点 X 在凸多边形内部；如果存在某条边 P_iP_{i+1} 有 $\overrightarrow{P_iP_{i+1}} \times \overrightarrow{P_iX} < 0$ ，那么，点 X 在凸多边形外。另一方面，为了获得事件发生地与某指定区域的距离，可以通过先计算点到凸多边形 n 个点的距离以及点到凸多边形 n 条边的距离，然后取其中最短的一个即是所需的距离。上述方法构成空间事件检测的基本方法。

在物联网应用场景中，如互联网位置提醒服务、商家基于位置的促销服务中，用户需要的服务实时性很强，若单纯依靠服务器进行事件检测，那么，服务器需要连续不断地接收位置发布信息并逐一检测事件，这样，即时通信量较大，事件检测的效率不高，用户体验比较差。

注意到客户所持有的智能终端其本身也具有一定的处理能力，让智能终端分担部分事件检测工作，可以减轻服务器的处理负载，减少客户与服务器之间的通信。为此，我们引入下列处理流程：在服务器收到一元位置订阅后，就把它转发到扮演事件发布者的客户上，客户收到事件发布者发布的事件后，就地与位置订阅进行匹配，若有订阅与之匹配，那么再发送通知给服务器，否则，直接丢弃该事件。同时，我们参考了空间位置提醒解决方案[7]中的安全区域的概念，将安全区域思想应用到空间事件检测的加速上，用于快速过滤无人感兴趣的事件。物联网下的应用所涉及的位置订阅经常与区域相关，我们参考[9]、[6]

中对空间数据的处理方法，在服务器端挖掘区域之间的关系，为其建立多级索引，用以提高事件的匹配效率，进而改善用户体验。

3. 安全区域的识别

我们定义安全区域为一个用户在当前位置点附近的区域，它在所有订阅关注的区域外，如果用户在这个区域的内部，则不会触发任何一个订阅。换句话说，安全区域是这样的一个区域：用户在这个区域内部移动的时候，他进入一个有订阅关注的区域的概率为零。这样，该用户在安全区域内的位置移动事件可以被直接过滤掉，无需发送到服务器去与订阅进行匹配。对一个用户而言，理想的安全区域是整个地图去除所有订阅关注的区域后剩下的部分。很明显，按这样的计算方式，在整个地图很大或者相关的订阅区域非常多的情况下，计算量十分巨大。由于理想状况下安全区域的求解难以实现，所以，我们需要用一个合理的可计算的区域来表示安全区域。为此，我们设计、实现了一种安全区域表示和计算的算法，即贪心最大矩形算法（Greedy Largest Rectangle，简称 GLR 算法）。

首先，我们将整个地图 U 抽象成一个大的矩形区域 $\text{Rect}(x,y,w,h)$ ，其中， (x,y) 为地图的左下角顶点， w 为地图的宽， h 为地图的高。然后，把地图按照应用需求分为 $M \times N$ 网格，也就是说，网格的大小要不小于任何一个用户在任意时刻的安全矩形区域。这样，横向从左至右的第 i 个、纵向自下而上的第 j 个网格区域定义为 $C_{i,j} = \text{Rect}(x+i \times \alpha, y+j \times \beta, \alpha, \beta)$ ， $M = \lceil w/\alpha \rceil$ ， $N = \lceil h/\beta \rceil$ 。有此定义后，每个点 $p(p_x, p_y)$ 到网格的映射 $f(p)$ 即是

$f(p) = C_{\left\lfloor \frac{p_x - x}{\alpha} \right\rfloor, \left\lfloor \frac{p_y - y}{\beta} \right\rfloor}$ 。同时，用户 s 的最大的安全区域为 $\phi_s = C_{k,l} - \bigcup_{i=1}^{|A_{s,k,l}|} R(A_i)$ ，其中， $C_{k,l}$ 为用户 s 所在的网格， $R(A_i)$ 为当前网格中每个与 s 所在位置相关的订阅区域， A_s 为整个地图中与 s 所在位置相关的订阅集， $A_{s,k,l}$ 为当前网格中与 s 所在位置相关的订阅集。

GLR 算法的输入是用户当前位置点 p_s 和通过映射函数 $f(p)$ 得到的网格 $C_{k,l}$ 。算法首先得到该网格内位置订阅区域 $R_{s,k,l}$ ，即，通过 $\text{Loc}(\text{sub})$ 取得订阅 sub 关注的位置（矩形区域），接着确定 $R_{s,k,l}$ 所在的象限（其象限中心为 p_s ）。然后进入算法的核心步骤：首先，对每个象限内的每个区域，去掉远离 p_s 的点，返回初步的候选点集合 canPointsQuads 。其次，按象限对 canPointsQuads 进行进一步处理，去掉能被同象限其他点支配（dominate）的点。然后，对现有的 canPointsQuads 做进一步处理，返回每个象限内的候选顶点。最后，通过候选点集 canPointsQuads 贪心构建一个最大矩形安全区域。

客户在获取自己的当前位置后，首先判断是否已建立安全区域，如果没有建立，则向服务器发送位置事件，并向服务器请求获得当前位置在地图对应网格所包含的订阅区域信息，用以建立安全区域；如果安全区域已经建立，并且当前的位置在该安全区域里面，则忽略该位置事件；如果安全区域已经建立，并且当前的位置不在该安全区域里面，则向服务器发送位置事件，并且重新计算安全区域。在客户端计算安全区域，可以减少客户与服务器的通信量，但需要在客户端存储订阅信息并且进行计算。

4. 空间索引的建立

为了提升事件检测的性能，我们挖掘了区域之间的关系，为其建立了空间索引。我们把这些区域以 R-Tree 方式存储。R-Tree[9]是一种经典的存储空间对象数据的数据结构，它

是 B+树在 K 维空间上的自然扩展，是一种高度平衡树。R-Tree 由根节点、中间节点和叶节点三类节点组成，中间节点代表数据集空间中的一个矩形，该矩形是包含了所有孩子节点的最小外接矩形，叶节点存储的是实际空间对象的外接矩形。借助 R-Tree 对空间区域建立索引，我们能够快速的实现从一个已知位置查找到包含该位置的（矩形）区域。

我们采用多级索引机制对订阅中的空间信息进行管理，第一级为位置订阅操作符索引，第二级对下层的订阅（一般为原子订阅）进行索引，最后是 R-Tree 索引。特别的，R-Tree 记录的信息不但包括了空间区域的信息，而且还包含了提交订阅的订阅者信息。这样，位置事件到达后，若在 R-Tree 索引找到订阅者非空的空间区域，那么表示可能有订阅与此位置事件相关，进一步判断，若位置事件与订阅中的区域约束关系成立，则可根据订阅者信息直接进行事件通知。注意到在某些场景中，如商家电子促销场景，订阅对应的订阅区域比较多，采用上述多级索引结构能够大大加快匹配过程。

5. 加速策略评估

加速策略已在我们开发的发布/订阅系统 OPS4ST 中实现。为了验证采用加速策略的 OPS4ST 的高效性，我们编写了一个采用普通检测策略的系统（记为 Naïve Strategy）作为对比系统，在该系统中，所有空间事件都被送到服务器按照第 2 节所述的基本方法逐一与订阅进行匹配。我们针对两个具体应用场景，即，出租车服务（E-Taxi）场景和商场电子促销（E-Coupon）场景，对涉及 $\text{DIST}(R, <, l)$ 和 $@(R)$ 操作符的空间事件检测做了相应的实验设计并进行了实验，在实验中记录了两个系统的性能和开销数据。

场景 1: E-Taxi 场景实验

用户即普通的出行者，对他所在位置附近约 200 米范围内的空载出租车信息感兴趣。所以，他提交的订阅为： $\text{DIST}((x,y)); < 200)(\text{STRING:state} = \text{"uncarry"})$ ，其中， (x,y) 表示用户当前的位置； $\text{state} = \text{"uncarry"}$ 表示用户对状态为空车的出租车感兴趣。出租车在移动中不断发送事件，事件的格式如下： $\text{ARITH:id} = v\text{-id}, \text{STRING:state} = \text{"uncarry"}, \text{systemTime}, x, y$ ；其中， $v\text{-id}$ 是一个正整数，表示出租车编号； systemTime 表示这个事件的发生时间； (x,y) 表示该出租车的当前位置。

我们首先向两个系统添加一定数目的订阅，然后不断的发布事件。事件是模拟出租车行驶路线生成的，具有连续性。在总共发布的 20000 个事件中，有 50% 的事件能与订阅匹配上。图 1 和图 2 显示了不同系统在该实验中的时空开销，其中，X 轴表示订阅的数目，从 1000 开始，每次增加 1000，一直增长到 10000，Y 轴表示服务器的时间或空间开销。

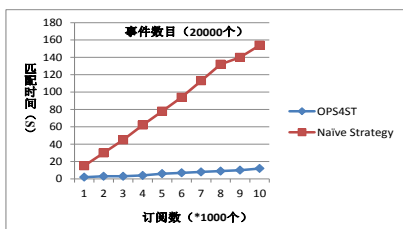


图 1 E-Taxi 场景中的时间开销

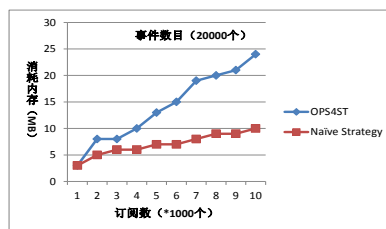


图 2 E-Taxi 场景中的空间开销

从图 1 和图 2，我们可以看出，与 Naïve Strategy 相比，OPS4ST 在 $\text{DIST}(R; <, l)$ 操作

符匹配效率上具有较大的优势，但由于服务器端增加了索引机制，这导致所需的内存开销有所增加。

场景 2: E-Coupon 场景实验

商家提交的订阅为 $@((PointA.x,PointA.y),(PointB.x,PointB.y),(PointC.x,PointC.y),(PointD.x,PointD.y))(ARITH:ageRange=v-range)$ ，其中 PointA、PointB、PointC、PointD 表示订阅区域的四个顶点；v-range 是一个整数，从集合{15,25,40,65}中取值，可分别代表四种年龄段人，即少年、青年、中年、老年。该订阅表示对出现在自己商铺位置附近（用矩形表示）的用户感兴趣，并且对用户的年龄段有所限制。用户移动时发布报告自己年龄和位置的事件，该事件的格式为 `STRING:username=v-name ,ARITH:ageRange=v-range,systemTime,x,y`，其中 v-name 表示用户名，仅限于从集合{"a","b","c","d","e"}中取值，分别代表了 5 个不同的用户；v-range 含义同上；systemTime 表示这个事件的发生时间；(x,y)表示用户的当前位置。

在这组实验中，首先模拟商家向系统添加一定数目的订阅，然后模拟 5 个用户各自不同的行走路线，从而向系统中连续发布 20000 个事件（每人 4000 个连续的位置事件）。位置事件的连续性使得客户端可以通过安全区域机制有效过滤那些一定不会被匹配上的事件。这 20000 个事件中，有 50%的事件能与已添加的订阅相匹配。图 3 和图 4 显示了不同系统在该实验中的时间和空间开销，其中 X 轴表示订阅的数目，从 1000 开始，每次增加 1000，一直增长到 10000，Y 轴表示服务器的时间或空间开销。

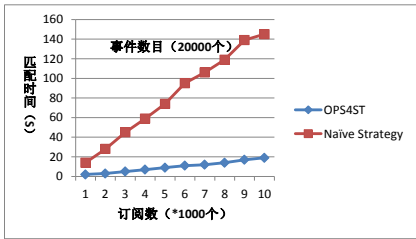


图 3 E-Coupon 场景中匹配实验的时间开销

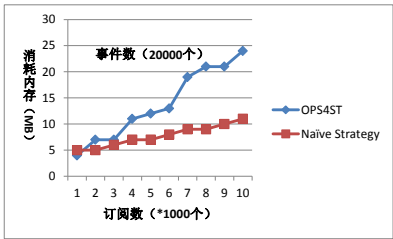


图 4 E-Coupon 场景中匹配实验的空间开销

实验结果表明，OPS4ST 在 E-Coupon 场景下，相对于 Naive Strategy 在@ (R)操作符匹配效率上有较大的优势，同时，由于服务器端增加了索引机制导致增加了一些内存开销。

6. 结束语

我们分析了一些常见的物联网下时空事件检测的应用场景，如位置提醒服务，还有商家基于位置的促销服务等，为加速空间事件检测，设计、实现了服务器端和客户端结合的加速算法。我们在客户端引入了安全区域机制，通过对客户端处理能力的合理利用，减少了客户端与服务器的通信量；在服务器端引入了多级索引机制，提高了服务器针对特定场景所涉及的空间操作符的处理性能。实验结果表明，采用加速策略的系统，其空间事件处理速度相比与采用普通匹配策略的系统有着显著的提高。

参考文献

[1] PIETZUCH P. R.and HERMES J. B.. Hermes: A distributed event-based middleware architecture,

- Proceedings of the 22nd International Conference on Distributed Computing Systems (ICDCS 2002). Washington, IEEE Computer Society Press, 2002:611-618.
- [2] AGUILERA M. K, STROM R. E, STURMAN D. C, etc. Matching events in a content-based subscription system. Eighteenth ACM Symposium on Principles of Distributed Computing. New York, NY, USA. 1999:53-61.
 - [3] CARZANIGA A, ROSENBLUM D, WOLFAL. Design and Evaluation of a Wide-Area Event Notification Service. ACM Transaction on Computer Systems, 2001: 19(3), 332-383.
 - [4] G. MÄUHL. Large-scale content-based publish/subscribe systems. PhD thesis, Department of Computer Science, Darmstadt University of Technology, 2002.
 - [5] Li GL, JACOBSEN HA. Composite Subscriptions in Content-based Publish/Subscribe Systems. the 6th ACM/IFIP/USENIX International Middleware Conference. 2005: 249-269.
 - [6] X.Chen, Y.Chen and F.Rao. An Efficient Spatial Publish/Subscribe System for Intelligent Location-Based Services. DEBS. 2003.
 - [7] B.Bamba, L.Liu, A.Iyengar and P.S.Yu. Distributed Processing of Spatial Alarms: A Safe Region-based Approach. International Conference on Distributed Computing Systems, ICDCS. 2009.
 - [8] Nir Zolotarevsky, Opher Etzion, Yuri G Rabinovich, Spatial Perspectives in Event Processing, DEBS 2009, Proceedings of the Third International Conference on Distributed Event-Based Systems, July 6-9, 2009, Nashville, TN, USA
 - [9] A.Guttman. R-Tree: A Dynamic Index Structure for Spatial Searching. ACM SIGMOD. 1984.

On Speed-up Strategies for Detecting Spatial Events

Jiafeng Hu^{1,2}, Beihong Jin¹⁺, Haibiao Chen¹

¹Institute of Software, Chinese Academy of Sciences, Beijing 100190, China

²Graduate University of Chinese Academy of Sciences, Beijing 100190, China

⁺ Beihong Jin: E-mail: Beihong@iscas.ac.cn

Key words: Spatial Event, Event Detection, Safe Region, Spatial Index

Abstract: With the development of the Internet of Things (IoT), the location-based applications, such as dynamic location alarm services, and location-based business promotion services, etc. emerge with bright prospects. Among these applications, the kernel technique is spatial event detection. After investigating multiple application scenarios, the paper explores the relations between the regions and builds multilevel indexes to improve the processing of spatial events. On the other hand, the paper utilizes the computational capacity of smart terminals which are held by clients, and calculates and maintains safe regions on the clients. By this way, the events can be filtered on clients, and the workloads on the server can be reduced. The paper also has conducted the simulation experiments on the system which adopts our speed-up strategies, so as to evaluate its performance and costs. The experimental data show our strategies can effectively accelerate the processing of spatial event detection.