

空间事件的检测及优化策略*

胡佳锋^{1,2}, 金蓓弘¹⁺, 褚伟^{1,2}, 陈海彪¹, 张利锋^{1,2}

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)

Spatial Event Detection and Optimization

HU Jia-Feng^{1,2}, JIN Bei-Hong¹⁺, ZHUO Wei^{1,2}, CHEN Hai-Biao¹, ZHANG Li-Feng^{1,2}

¹(Institute of Software, The Chinese Academy of Sciences, Beijing 100190, China)

²(Graduate University, The Chinese Academy of Sciences, Beijing 100190, China)

+ Corresponding author: E-mail: Beihong@iscas.ac.cn

Hu JF, Jin BH, Zhuo W, Chen HB, Zhang LF. Spatial event detection and optimization. *Journal of Software*, 2011, 22(Suppl. (2)): 147-156. <http://www.jos.org.cn/1000-9825/11035.htm>

Abstract: Currently, various location-based applications, such as dynamic location alarm services and location-based shopping promotion services, etc. emerge with bright prospects. Among these applications, the kernel technique is spatial event detection. The paper adopts Pub/Sub middleware to detect spatial events, presenting the basic detecting method. Moreover, the paper presents the optimization strategy of detecting the events which are matched with unary location subscriptions. Specifically, the paper explores the relations between the regions covered by events and builds the multilevel indexes to improve the processing of spatial events. On the other hand, it utilizes the computational capacities of client computers and calculates and maintains the safe regions on the clients. Through this system, the events can be filtered on the clients, and the workloads on the server can be reduced. The paper has also conducted the simulation experiments on the system which implements the proposed speed-up strategies to evaluate its performance and costs. The experimental data show the speed-up strategies can efficiently accelerate the processing of spatial event detection.

Key words: spatial event; event detection; pub/sub; safe region; spatial index

摘要: 当前,各种基于位置的应用,如用户动态位置提醒服务,商家基于位置的促销服务等不断出现,应用前景广泛.这些应用的关键是空间事件的检测.采用发布/订阅中间件检测空间事件,给出了空间事件检测的基本方法,并且针对一元位置订阅,给出了相应的事件检测优化策略,即为提升空间事件的处理速度,挖掘了事件发生区域之间的关系,为其建立多级索引,另一方面,利用客户端的处理能力,构造并维护安全区域,通过在客户端过滤事件,减轻了服务器的事件检测负载.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,上述加速策略能够有效地加速空间事件的检测.

关键词: 空间事件;事件检测;发布/订阅;安全区域;空间索引

随着定位技术的成熟和定位设备(如 GPS 接收器、RFID 设备等)的普及,用户可以很方便地观察到周围环境中发生的带有时间和位置标记的事件(我们称为原子事件).但仅获得这些事件还不足以满足应用的需求,在

* 基金项目: 国家自然科学基金(60970027)

收稿时间: 2011-07-20; 定稿时间: 2011-12-01

不少实际应用中,常常需要捕获时空事件,并根据这些时空事件决定其下一步的行为.这里的时空事件是指一个由若干事件(包含原子事件)组成的复杂事件,而且这些组成复杂事件的成员事件需满足用户事先指定的时空约束.例如,在物流系统中,经常需要观察下列事件:指定车辆是否按时离开某指定的仓库,指定车辆是否在指定时间段内到达指定区域,车队是否按指定线路依次到达各送货地等等.这些事件都是时空事件.同时我们发现,不涉及时序关系的空间事件的检测需求也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,方便出行.又例如,在商家基于位置的促销服务中,商家需要观察用户在其商铺附近出现这样的事件.总之,时空事件的有效检测已成为许多应用能否成功部署和推广的关键.发布/订阅中间件^[1]内在的特性正好适合提供此类服务,可以在上述应用的构建中扮演重要的时空事件检测和通知的角色.

在发布/订阅中间件中,事件反映了事物的状态,通常用属性-值对方式描述,订阅描述了订阅者感兴趣的事件,在基于内容的发布/订阅中间件中,原子订阅是一组谓词的合取,用于指定对事件属性的约束.而复合订阅是将多个订阅(也称成员订阅)用操作符连接起来的订阅,反映了多个事件之间要满足的约束.订阅者提交订阅,声明他们感兴趣的事件,发布/订阅中间件中的服务器接收原子事件,并负责检测原子事件及复杂事件是否与订阅匹配,然后,将满足订阅的事件推送给订阅者.

由于当前已有的发布/订阅中间件对事件的时间和空间关系的支持都比较弱,不能满足实际应用对时空事件检测的需求,为此,在调研、分析了多种应用需求的基础上,我们设计、实现了能够检测时空事件的发布/订阅中间件 OPS4ST 标准版.同时,我们针对空间事件检测,设计、实现了 OPS4ST 优化版,优化版提供了面向一元位置订阅的加速策略,加速了空间事件的检测.具体而言,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以及服务器的负载;在服务器端引入了多级索引机制,提高了服务器针对一元位置订阅的处理性能.本文主要叙述在 OPS4ST 中间件中实现的空间事件检测及优化策略.

第 1 节介绍相关工作.第 2 节给出空间事件的订阅表达.第 3 节描述空间事件的检测方法,包括事件之间空间关系的判断方法、面向一元位置订阅的事件检测优化策略.第 4 节评估空间事件检测优化方法的性能和开销.最后是全文的总结.

1 相关工作

当前已有的发布/订阅中间件对事件的时序、空间关系的支持都比较弱,也就是说,用户提交的订阅所能表达的时空事件模式种类很少.

早期系统如 Hermes^[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,它们都不能满足目前应用对空间事件检测的需求.我们以前构建的系统^[6]挖掘了事件之间的时序关系特别是有严格偏序的时序关系,能检测非触发式事件,这里,非触发式事件是指无法通过某个事件的发生来通知系统自己已经发生的事件.不过,该系统未考虑事件的空间关系.

在事件的空间关系表达上,文献[7]为基于位置的服务设计、实现了两个订阅谓词 *within* 和 *distance*,但它提供的功能仅限于此.在空间事件处理方面,文献[8]是 IBM 关于它开展空间事件处理的早期报告,文献[9]讨论了空间位置提醒(*spatial alarm*)需求,即用户在他提交的空间警报(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.空间位置提醒提供对一类空间事件的检测,所以,可以看成是发布/订阅机制的变种.文献[9,10]给出了对空间位置提醒的解决方案,特别地,引入了安全时间区间和安全区域的概念来加速空间位置提醒的处理.文献[11]提供了 SQL 风格的语言用于检测传感网中的事件.该语言能够描述非触发式事件,但提供的时空约束比较有限.例如,只有一个指定位置约束的手段,即指定两个事件的位置之间的距离.

时空事件除了通过发布/订阅方式检测之外,数据流系统以另一种方式即通过执行连续查询来识别所需的时空事件.

文献[12]为移动对象上的连续静态范围查询设计了一个查询索引机制.查询索引机制是为查询而不是为对象建立索引,因为增删查询没有对象移动频繁,所以,这种索引的更新量能够显著减少,该文还最早引入安全区域概念,通过计算查询不关心的移动对象的位置,快速过滤掉无用的对象移动事件,从而提高数据流处理性能.SINA^[13]在时空数据流上实现了对连续范围查询的增量式求解.它使用共享执行图进行并发查询的增量式求解.SINA 的缺点在于它不允许用户动态增加或取消查询.文献[14]把连续约束查询建模成二元约束满足问题(CSP 问题),并用 CSP 求解器求解,但受已有 CSP 求解器效率不高的限制,该方法无法有效应对大量的连续查询.

本文提出了一套完整的空间事件检测技术,并借鉴数据流连续查询处理技术,针对一元位置订阅,在服务器端建立多级索引,在客户端通过安全区域过滤事件,提升了空间事件的处理速度.

2 空间事件的订阅表达

我们首先给出 OPS4ST 中事件发生位置的表示方法.对于事件发生的位置,如果该事件覆盖一定的地理范围,那么我们用描述该范围边界的地理坐标点的集合表示该事件.如果该事件的发生位置足够小,仅用一个地理坐标点表示就足够了,那么上述点集就仅包含一个点.除点集外,我们同时记录包含该点集的最小凸多边形,也就是说,我们用一个点集和相应的凸包来表示事件发生的位置.

OPS4ST 采用基于内容的发布/订阅模式,也就是说,原子订阅是一组谓词的合取,这里的谓词采用如下格式:“属性类型:属性名字 操作符 值”,而复合订阅是若干成员订阅通过复合操作符连结的订阅,用于表达复杂事件的模式.

空间事件是有空间关系约束的事件,属于复杂事件,相应的订阅需要用空间操作符来表示.OPS4ST 提供下列空间操作符,以表示事件位置之间的拓扑关系^[15]和度量关系^[16]:

- (1) 之内(happen-in)操作符,基本形式为 $@(R)$,变体为 $IN(A)$;
- (2) 覆盖操作符,基本形式为 $OVLP(R)$,变体为 $OVLP(A)$;
- (3) 相同地点操作符,基本形式为 $SPL(R)$,变体为 $SPL(A)$;
- (4) 距离操作符,基本形式为 $DIST(R;<;l)$ 、 $DIST(R;=;l)$ 和 $DIST(R;>;l)$,变体为 $DIST(A;<;l)$ 、 $DIST(A;=;l)$ 和 $DIST(A;>;l)$.

这里, R 表示一个用凸多边形表示的区域, l 是整数,表示距离, A 表示订阅,在上述操作符中,用来传递发生满足 A 的事件的位置信息.我们把仅涉及单个空间操作符基本形式的订阅称为一元位置订阅.同时,为了表述的方便,下文都用大写字母表示订阅,用相应的小写字母表示满足该订阅的事件.例如, A 、 B 表示订阅, a 、 b 表示满足订阅 A 、 B 的事件.

基于以上的空间操作符,用户可以用订阅 $@(R)A$ 观察发生在区域 R 的事件 a ;而订阅 $OVLP(A)B$ 可以表示用户需要在事件 a 、事件 b 的发生地点有重叠的时候被通知;订阅 $SPL(A)B$ 表示用户关注在相同地点发生的事件 a 和事件 b ;订阅 $DIST(R;<;l)A$ 、 $DIST(R;=;l)A$ 、 $DIST(R;>;l)A$ 分别用于观察事件发生地与区域 R 距离小于 l 的事件 a 、与区域 R 距离 l 处的事件 a 和与区域 R 距离大于 l 的事件 a .

下面通过具体应用场景来说明如何用订阅表达空间事件.

(1) 出租车服务场景

出租车服务为普通的出行者提供服务.用户需要在需要外出的时候,可以提交相关订阅,以期告之附近有空载出租车的信息.例如,用户对他所在位置附近约 2 000m 范围内的空载出租车信息感兴趣.所以,他可以提交下面的订阅.

$$DIST((x,y));<;2000)(STRING:type="TAXI";STRING:state="uncarry") \quad (\text{订阅 1})$$

订阅 1 中的 (x,y) 表示用户当前的位置; $STRING:type="TAXI";STRING:state="uncarry"$ 表示用户对空载的出租车感兴趣.

(2) 商家基于位置的促销服务

商家发现其周围有顾客出现,就可以给其发送商品广告.商家为了发现商铺附近的顾客,可以提交这样的

订阅:

$@((PointA.x,PointA.y),(PointB.x,PointB.y),(PointC.x,PointC.y),(PointD.x,PointD.y))(ARITH:ageRange=v-range)$
(订阅 2)

订阅 2 中的 $PointA$ 、 $PointB$ 、 $PointC$ 、 $PointD$ 表示订阅区域的 4 个顶点, $v-range$ 是一个整数,从集合 {15,25,40,65} 中取值,分别代表 4 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.

(3) 好友到达通知场景

用户希望在好友到达的时候自动获得通知,那么,他可以提交这样的订阅:

$SPL(Circle(x,y,500))(ARITH:group=v-group)$ (订阅 3)

订阅 3 中的 $Circle(x,y,500)$ 表示以用户当前的位置坐标为圆心、半径为 500 米的圆形区域; $v-group$ 是一个整数,表示用户感兴趣的组别,这里需要设置为好友所在组别的编号.

3 空间事件的检测

OPS4ST 采用客户/多服务器体系结构.在 OPS4ST 中,空间事件检测的基本流程是:当订阅者提交的复合订阅从客户端到达 OPS4ST 的服务器 X 时,会在服务器 X 上建立相应的匹配结构,同时以服务器 X 为订阅者,向其他服务器提交组成该复合订阅的各原子订阅.当原子事件到达某服务器后,若与该服务器上所保存的以 X 为订阅者的订阅相匹配,那么,将该事件转发给 X,由 X 进一步负责对复杂事件的匹配,即进行空间约束关系的检测.若有复合订阅被满足,那么给订阅者发送通知.OPS4ST 标准版实现了上述检测流程.

在第 2 节提到的应用场景中,用户对服务的实时性要求很强,若单纯依靠服务器进行事件检测,那么,在原子事件被大量发布的场景下,服务器需要基于被匹配的原子事件逐一检测空间事件,所以,会出现即时通信量较大,事件检测的效率不高,用户体验较差的状况.同时,OPS4ST 标准版没有充分挖掘事件覆盖区域间的关系.为此,针对第 2 节定义的一元位置订阅,我们提出了空间事件检测的加速策略,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.该策略实现在 OPS4ST 优化版中.

3.1 空间关系的判定

在 OPS4ST 中,为了判断事件是否满足在订阅中表达的空间关系,需要进行以下几种计算:(1) 求点集的凸包,我们用 Graham 扫描算法^[17]求解.在两个或多个事件形成一个复杂事件时,我们把根据构成各成员事件的点集构造出的最小凸多边形,作为该复杂事件的位置属性.(2) 判断点与凸多边形的拓扑关系.二维空间上一个点和凸多边形的关系主要有 3 种:点在凸多边形内、点在凸多边形上、点在凸多边形外.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.(3) 判断凸多边形之间的拓扑关系.我们关注两个凸多边形的“内含(in)”、“重叠(overlap)”、“分离(disjoint)”3 种关系,图 1 给出了这 3 种拓扑关系的示意图.我们通过逐个计算点与凸多边形关系来得出凸多边形间的拓扑关系,从而获得事件之间的拓扑关系.(4) 求解两分离的凸多边形的距离即最近点对的距离,并把它作为两个事件之间及事件与区域之间的距离.我们采用旋转卡壳寻找对踵点算法^[18]计算.(5) 计算凸多边形的中心点.我们将包含 n 个顶点的凸多边形分解为 $n-2$ 个三角形(它们以凸多边形的某个顶点为其公共顶点),分别求这 $n-2$ 个三角形的重心和面积,得到总面积的同时按照三角形的面积加权相加得到凸多边形的重心,并将该重心作为凸多边形的中心点.我们通过求两凸多边形中心点之间的距离是否小于设定阈值来判断两事件是否发生于同一个地点.上述方法构成空间事件检测的基本方法,已实现在 OPS4ST 标准版中.

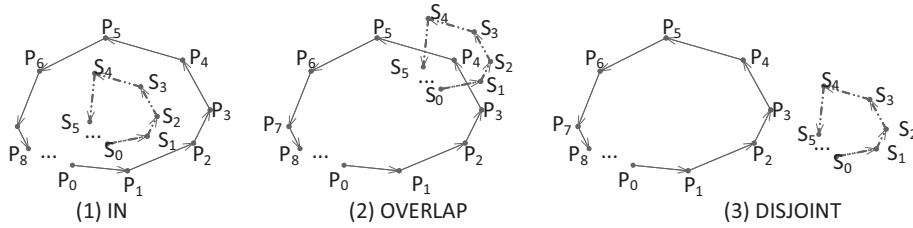


Fig.1 Topological relationships between two convex polygons
图 1 两个凸多边形的空间拓扑关系示意图

3.2 安全区域的识别

我们定义安全区域为这样的一个区域,它在所有订阅关注的区域外,如果到达事件的位置在这个区域的内部,那么该事件不会触发任何一个订阅.也就是说,在安全区域内的位置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.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、实现了一种表示和计算安全区域的算法,即贪心最大矩形(greedy largest rectangle,简称 GLR)算法,如图 2 所示.

```

//Input: Location Point  $p_s$ , Corresponding Grid  $C_{k,l}$ ;
//Output: SafeRegion  $\phi_s$ .
1 safeRegion GLRSafeRegionAlgorithm(Position  $p_s$ , Grid  $C_{k,l}$ )
2 BEGIN
3   FOR each  $sub$  in  $A_s$ 
4      $R_{s,k,l} = R_{s,k,l} \cup \{Loc(sub) \cap C_{k,l}\}$ 
5   IF  $R_{s,k,l}.isEmpty() = true$ 
6      $\phi_s = C_{k,l}$ 
7   ELSE
8      $regionQuads = assignRegionToQuads(R_{s,k,l})$ 
9      $canPointsQuads = getQuadrants(regionQuads)$ 
10     $canPointsQuads = generateCandidatePoints(canPointsQuads)$ 
11     $canPointsQuads = generateTensionPoints(canPointsQuads)$ 
12     $\phi_s = getFinalSafeRegion(canPointsQuads)$ 
13 RETURN  $\phi_s$ 
    
```

Fig.2 GLR algorithm
图 2 GLR 算法

首先,我们将整个地图 U 抽象成一个大的矩形区域 $Rect(x,y,w,h)$,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,网格的大小要不小于任何一个移动对象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 个网格区域便可以定义为 $C_{ij} = Rect(x + i \times \alpha, y + j \times \beta, \alpha, \beta)$, $M = \lfloor w/\alpha \rfloor$, $N = \lfloor h/\beta \rfloor$.有此定义后,每个点 $p(p_x, p_y)$ 到网格的映射 $f(p)$ 即是

$$f(p) = C_{\left\lceil \frac{p_x - x}{\alpha} \right\rceil \left\lceil \frac{p_y - y}{\beta} \right\rceil}$$

设当前位置点为 p_s ,以 p_s 为中心的最大的安全区域为 $\phi_s = C_{kl} - \bigcup_{i=1}^{|A_{s,k,l}|} R(A_i)$,其中, $C_{k,l}$ 为 p_s 所在的

的网格, $R(A_i)$ 为当前网格中一个与 p_s 相关的订阅的区域, $A_{s,k,l}$ 为当前网格中与 p_s 相关的订阅集.

GLR 算法的输入是移动对象的当前位置点 p_s 和通过映射函数 $f(p)$ 得到的网格 $C_{k,l}$.设 A_s 为整个地图中与 p_s 相关的一元位置订阅集.算法首先得到网格 $C_{k,l}$ 内一元位置订阅区域 $R_{s,k,l}$,即,对 A_s 中的订阅 sub ,通过 $Loc(sub)$ 取得订阅 sub 关注的位置(拓展为最小外接矩形区域),这里,如果订阅关注的区域在 $C_{k,l}$ 内部,就将区域放入 $R_{s,k,l}$ 中;如与 $C_{k,l}$ 重叠,则首先将订阅关注的区域与 $C_{k,l}$ 求得交集后再放入 $R_{s,k,l}$ 中.如果 $R_{s,k,l}$ 为空,则说明该网格内部没有订阅关注的区域,整个 $C_{k,l}$ 即为所求的安全区域.接着,调用 $assignRegionToQuads()$ 方法,以 p_s 为象限中心,

将 $R_{s,k,l}$ 对应到每个象限中.该方法的返回值为数组 *regionQuads*,它的下标为象限名,每一项为对应象限的区域集.然后进入算法的核心步骤:首先,对每个象限内的每个订阅关注的矩形区域,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* 作进一步处理,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.

客户在获取移动对象的位置事件后,首先判断是否有该移动对象位置的历史记录,如果有,则意味着已为该移动对象建立了安全区域,如果没有,则向服务器发送位置事件,并向服务器请求获得移动对象的位置事件所报告的当前位置在地图对应网格所包含的订阅区域信息,并调用 GLR 算法,建立该移动对象的安全区域;如果安全区域已经建立,并且当前的位置在这个安全区域里面,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,并且调用 GLR 算法重新计算该移动对象的安全区域.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.图 3 给出了一个安全区域的示意图. P_s 是客户保存的移动对象的位置. $A_1 \sim A_5$ 所在的矩形为相应订阅所关注的区域,阴影矩形为以 P_s 为中心点的安全区域.

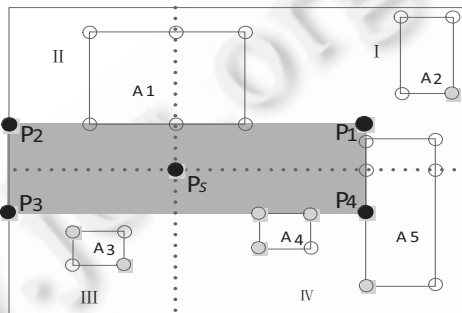


Fig.3 An example of safe region

图 3 安全区域示意图

3.3 空间索引的建立

为了提升空间事件检测的性能,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 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、中间节点和叶节点这 3 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,我们能够快速地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,如图 4 所示,第 1 级为位置订阅操作符(如 @, DIST)索引,第 2 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,先找到与此相匹配的原子订阅(通过索引第 1 层及第 2 层),再到该原子订阅所对应的 R-tree 中(索引第 3 层)找到那些包含该位置点的区域.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.我们发现在某些场景,如商家促销场景,同类订阅对应的订阅区域比较多,对应的原子订阅比较单一,采用上述多级索引结构能够大大加快空间事件与一元位置订阅的匹配过程.

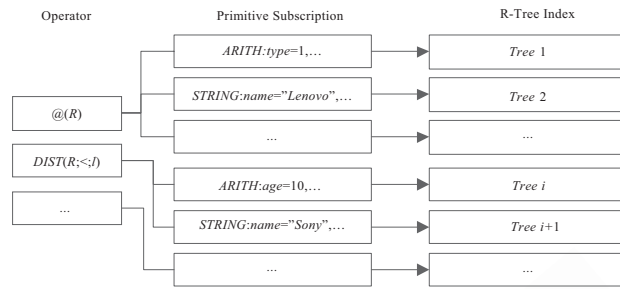


Fig.4 An example of the multi-level index

图 4 多级索引示意图

第 3.2 节和第 3.3 节的优化机制实现在 OPS4ST 优化版中.

4 实验评估

为了评估 OPS4ST 优化版的性能和开销,我们将它与 OPS4ST 标准版作对比,进行了若干实验.我们以出租车服务和商家促销服务为背景设计实验,对一元位置订阅与事件的匹配进行了评估,以验证 OPS4ST 优化版的优化效果.

4.1 出租车服务场景

该实验模拟多个用户提交第 2 节中的订阅 1.同时,模拟空载出租车在移动中不断发送如下格式的事件:

$ARITH: id=v-id, STRING: type="TAXI", STRING: state="uncarry", systemTime, x, y$

其中, $v-id$ 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.

具体实验过程如下:首先向 OPS4ST 标准版、优化版中添加一定数目的订阅 1,然后不断地发布事件.事件是模拟出租车行驶路线生成的,具有连续性.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.在总共发布的 20 000 个事件中,有 50%的事件能与订阅匹配上.图 5 显示了 OPS4ST 标准版及优化版在该实验中的时空开销,图 5 中, X 轴表示订阅的数目,从 1 000 开始,每次增加 1 000,一直增长到 10 000, Y 轴表示服务器所花的匹配时间或内存开销.

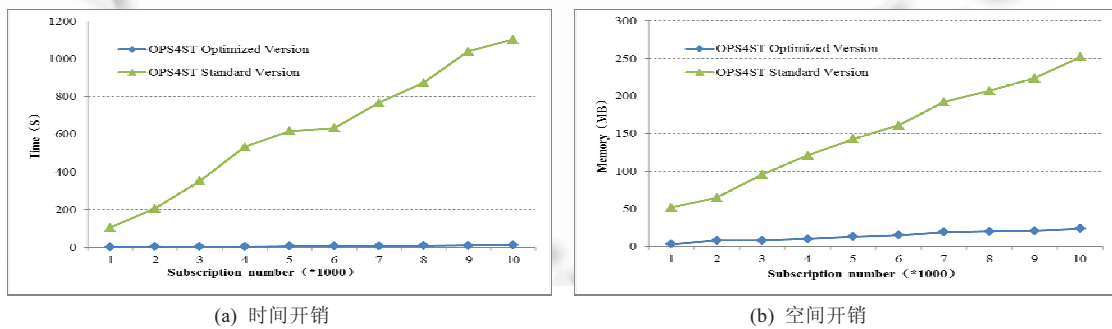


Fig.5 For taxi scenario

图 5 出租车服务场景

实验结果表明,与 OPS4ST 标准版相比,OPS4ST 优化版在 $DIST(R; \leq; L)$ 操作符匹配上具有较大的优势.从图 5 中可以看到匹配 20 000 个事件,OPS4ST 优化版 10 次实验的平均时间仅为 OPS4ST 标准版的 1.02%,平均内存消耗为 OPS4ST 标准版的 9.32%.

4.2 商场促销场景

该实验模拟多个商家提交第 2 节的订阅 2.同时,模拟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* 含义与订阅中的 *v-range* 相同;*systemTime* 表示这个事件的发生时间;*(x,y)*表示用户的当前位置.

具体实验过程:首先模拟商家向系统添加一定数目的订阅 2,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 20 000 个事件(每人 4 000 个连续的位置事件).这 20 000 个事件中,有 50%的事件能与已添加的订阅相匹配.图 6 显示了系统不同版本在该实验中的时间和空间开销,图 6 中 *X*轴表示订阅的数目,从 1000 开始,每次增加 1 000,一直增长到 10 000,*Y*轴表示服务器的时间或空间开销.

从实验结果可以看出,OPS4ST 优化版在商场促销场景下,相对于 OPS4ST 标准版,在@(*R*)操作符匹配上有较大的优势.从图 6 中可以看出,匹配 20 000 个事件,OPS4ST 优化版的平均时间开销仅为 OPS4ST 标准版的 1.24%,内存占用的平均开销为 OPS4ST 标准版的 7.61%.

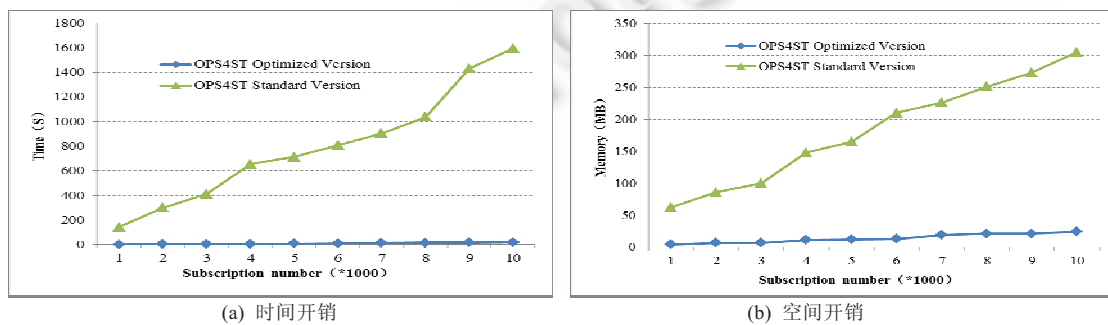


Fig.6 For market promotion scenario

图 6 商场促销场景

5 结束语

本文关注利用发布/订阅中间件进行空间事件检测.本文提出了一套空间事件检测机制,包括空间关系判定机制、面向一元订阅的检测优化策略,从而能够正确、高效地完成空间事件的检测.特别地,本文针对基于一元位置订阅的应用场景,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.上述机制已实现在我们开发的发布/订阅中间件 OPS4ST(标准版和优化版)中.实验结果表明,针对空间事件检测,OPS4ST 标准版与优化版相比,后者能够更有效地检测到空间事件.下一步的工作包括进一步提高系统的可伸缩性以及提供对移动订阅者的支持.

References:

- [1] Eugster PT, Felber PA, Guerraoui R, Kermarrec AM. The many faces of publish/subscribe. *ACM Computing Surveys*, 2003,35(2):114–131. [doi: 10.1145/857076.857078]
- [2] Pietzuch PR, Bacon JM. Hermes: A distributed event-based middleware architecture. In: *Proc. of the 22nd Int'l Conf. on Distributed Computing Systems*. Washington: IEEE Computer Society Press, 2002. 611–618. [doi: 10.1109/ICDCSW.2002.1030837]
- [3] Carzaniga A, Rosenblum DS, Wolf AL. Design and evaluation of a wide-area event notification service. *ACM Trans. on Computer Systems*, 2001,19(3):332–383. [doi: 10.1145/380749.380767]

- [4] Li GL, Jacobsen HA. Composite subscriptions in content-based publish/subscribe systems. In: Alonso G, ed. Proc. of the 6th ACM/IFIP/USENIX International Middleware Conf. LNCS 3790, Heidelberg: Springer-Verlag, 2005. 249–269. [doi: 10.1007/11587552_13]
- [5] Wang FS, Liu SR, Liu PY, Bai YJ. Bridging physical and virtual worlds: Complex event processing for RFID data streams. In: Ioannidis Y, *et al.* eds. Proc. of the 10th IEEE Int'l Conf. on Extending DB Technology (EDBT 2006). LNCS 3896, Heidelberg: Springer-Verlag, 2006. 588–607. [doi: 10.1007/11687238_36]
- [6] Jin BH, Zhao XC, Long ZY, Qi FL, Yu S. Effective and efficient event dissemination for RFID applications. *The Computer Journal*, 2009,52(8):988–1005. [doi: 10.1093/comjnl/bxn063]
- [7] Chen XY, Chen Y, Rao FY. An efficient spatial publish/subscribe system for intelligent location-based services. In: Jacobsen HA, ed. Proc. of the 2nd Int'l Workshop on Distributed Event-Based Systems (DEBS 2003). New York: ACM Press, 2003. 1–6. [doi: 10.1145/966618.966625]
- [8] Etzion O, Zolotarevsky N. Spatial perspectives in event processing. In: Sachs K, Petrov I, Guerrero P, eds. From Active Data Management to Event-Based Systems and More. LNCS 6462, Heidelberg: Springer-Verlag, 2010. 85–107. [doi: 10.1007/978-3-642-17226-7_6]
- [9] Bamba B, Liu L, Yu PS, Zhang G, Doo M. Scalable processing of spatial alarms. In: Sadayappan P, Parashar M, Badrinath R, Prasanna VK, eds. Proc. of the 15th Int'l Conf. on High Performance Computing (HiPC 2008). LNCS 5374, Heidelberg: Springer-Verlag, 2008. 232–244.
- [10] Bamba B, Liu L, Iyengar A, Yu PS. Distributed processing of spatial alarms: A safe region-based approach. In: Proc. of the 29th IEEE Int'l Conf. on Distributed Computing Systems (ICDCS 2009). Montreal: IEEE Computer Society, 2009. 207–214. [doi: 10.1109/ICDCS.2009.25]
- [11] Mao J, Jannotti J, Akdere M, Cetintemel U. Event-Based constraints for sensor network programming. In: Baldoni R, ed. Proc. of the 2nd Int'l Conf. on Distributed Event-Based Systems (DEBS 2008). New York: ACM Press, 2008. 103–113. [doi: 10.1145/1385989.1386003]
- [12] Prabhakar S, Xia Y, Kalashnikov DV, Aref WG, Hambrusch SE. Query indexing and velocity constrained indexing: Scalable techniques for continuous queries on moving objects. *IEEE Trans. on Computers*, 2002,51(10):1124–1140. [doi: 10.1109/TC.2002.1039840]
- [13] Mokbel MF, Xiong X, Aref WG. SINA: Scalable incremental processing of continuous queries in spatio-temporal databases. In: König AC, Dessloch S, eds. Proc. of the 2004 ACM SIGMOD Int'l Conf. on Management of data (SIGMOD 2004). New York: ACM Press, 2004. 623–634. [doi: 10.1145/1007568.1007638]
- [14] Hadjieleftheriou M, Mamoulis N, Tao Y. Continuous constraint query evaluation for spatiotemporal streams. In: Papadias D, Zhang D, Kollis G, eds. Proc. of the 10th Int'l Conf. on Advances in Spatial and Temporal Databases (SSTD 2007). LNCS 4605, Heidelberg: Springer-Verlag, 2007. 348–365.
- [15] Randell DA, Zhan C, Anthony GC. A spatial logic based on regions and connection. In: Proc. of the 3rd Int'l Conf. on Knowledge Representation and Reasoning. San Mateo: Morgan Kaufmann Publishers, 1992. 165–176. [doi: 10.1.1.35.7809]
- [16] Gutting RH. An introduction to spatial database systems. *The VLDB Journal*, 1994,3(4):357–388. [doi: 10.1007/BF01231602]
- [17] Cormen TH, Leiserson CE, Rivest RL, Stein C. *Introduction to Algorithms*. 2nd ed., Cambridge: MIT Press, 2001. 946–953.
- [18] Toussaint G. Solving geometric problems with the rotating calipers. In: Protonotarios EN, ed. Proc. of IEEE Mediteranian Electrotechnical Conf. (MELECON 1983). New York: IEEE Press, 1983. 1–4.



胡佳锋(1988-),男,浙江上虞人,硕士生,主要研究领域为分布式计算,普适计算.



陈海彪(1986-),男,工程师,主要研究领域为分布式计算,普适计算.



金蓓弘(1967-),女,博士,研究员,主要研究领域为分布式计算,普适计算.



张利锋(1977-),男,博士生,主要研究领域为分布式计算,普适计算.



榘伟(1986-),男,硕士生,主要研究领域为分布式计算,普适计算.

www.jos.org.cn

www.jos.org.cn