

一种分布式环境中的近邻检测方法

徐 建¹, 郭 鸣², 徐 冲¹, 徐 明¹, 郑 宁¹

(1. 杭州电子科技大学, 浙江杭州 310037; 2. 浙江大学城市学院, 浙江杭州 310015)

摘 要: 为了提高分布式环境中近邻检测的效率, 本文提出了一种基于近邻框的检测方法. 近邻就是在用户周围一定物理距离范围内其所关心的朋友. 在大规模近邻检测中, 一般的检测方法研究都关注减少系统内用户客户端和服务端之间位置更新消息的数目, 以降低服务器的负担. 本文使用近邻框概念, 通过移动用户之间位置关系的简单判断来取代欧氏距离或者最短距离的计算, 来提高系统的处理效率. 同时在用户客户端结合地图信息对自身的移动区域进行自适应地预测, 减少客户端和服务端之间的消息交互. 论文讨论了近邻框检测的一般步骤, 检测过程中疑似近邻用户的处理, 并对系统性能展开了分析, 通过实验验证了近邻框检测的可行性. 实验结果表明本文的方法在不同环境下都能较好地完成近邻查询, 方法中使用的优化技术可以显著提高系统的整体效率.

关键词: 基于位置的服务 (LBS); 分布式; 近邻检测; 近邻框

中图分类号: TP391 **文献标识码:** A **文章编号:** 0372-2112 (2013)05-1001-06

电子学报 URL: <http://www.ejournal.org.cn> **DOI:** 10.3969/j.issn.0372-2112.2013.05.027

A Proximity Detection Algorithm in Distributed Environment

XU Jian¹, GUO Ming², XU Chong¹, XU Ming¹, ZHENG Ning¹

(1. Hangzhou Dianzi University, Hangzhou, Zhejiang 310037, China; 2. City College, Zhejiang University, Hangzhou, Zhejiang 310015, China)

Abstract: A proximity box based detection algorithm is presented to improve the efficiency of proximity detecting in distributed environment. Proximity detection is to find each pair of friends such that the distance between them is within a given threshold. Servers in Location-Based Services are likely to be the bottlenecks in large scale proximity detecting, therefore, the state-of-art-of proximity detection methods are designed with the goal to reduce the server load. In this paper, we propose a detecting method based on proximity box instead of the Euclidean distance or the shortest distance, thereby improving the processing efficiency of the system. At user client, an adaptive method with the road map information is introduced to predicting the user's security moving zone for reducing the message exchange between the client and server. The detailed steps of the proximity detection algorithm are given and approaches to deal with suspected proximity user are discussed. An indexing algorithm for moving users and proximity relations between friends is also discussed. The experimental results indicate the algorithm is effective in a real city map. The results also prove that proposed index algorithm can significantly improve the overall efficiency of the system.

Key words: location-based services; distributed environment; proximity detection; proximity rectangle

1 引言

在日常生活中, 随着 GPS 和无线通信网络的普及, 基于位置的服务 LBS (Location-Based Services) 已经成为现实. 在众多 LBS 服务中, 路网环境中移动物体的近邻检测是许多应用的基础, 因此它是一个重要的研究话题.

给定一组用户 U , 一个路网环境 G , 以及某用户朋友对 (u_i, u_j) 的近邻距离标准 $\epsilon_{i,j}$, 近邻检测问题^[1]被定义为寻找这样的朋友对 (u_i, u_j) , 首先 u_i, u_j 是相邻的, 其次是 u_i, u_j 之间的欧几里德距离 $\text{DistO}((u_i, u_j)) \leq$

$\epsilon_{i,j}$. 在一个路网环境中的近邻检测问题^[2]就是 $\text{DistD}((u_i, u_j)) \leq \epsilon_{i,j}$, 其中 DistD 表示两个节点之间的 Dijkstra 最短距离. 目前大部分研究都在欧几里德空间讨论这个问题^[3~5], 以减少服务器处理用户更新消息数量为目的, 并没有考虑单次位置更新/近邻查询的开销. 而在用户数量规模较大的路网环境中, 邻近查询中欧几里德距离或者 Dijkstra 最短距离的计算, 随着位置更新消息数目的增加将会带来巨大地计算开销.

本文介绍了一种适用于分布式路网环境的连续近邻检测方法. 文章对提出的方法进行了理论分析, 介绍了若干优化的方法, 最后通过实验验证了它的可行性.

2 系统模型

系统的组成包括移动用户和 LBS 服务器,假设两者对所处地理环境具有相同的地图信息.移动用户通过无线通信网络将自身位置等信息发送给服务器.在收到移动用户发来的更新消息后,服务器会更新移动用户的位置,当前移动速度等参数,并进行跟踪.在保证跟踪精度的前提下,为了减少服务器对移动用户更新信息的处理压力,移动用户和服务器运行相同的前向位置预测算法,服务器根据移动用户当前的位置、其朋友列表、近邻距离,进行近邻检测,并将结果通告给所有相关的移动用户.

假设系统中存在 n 个移动用户 $u = \{u_1, u_2, \dots, u_n\}$, 用户 u_i 的一个移动轨迹是一个三元组的时空序列,

$$S = ((t_0, l_0, v_0), (t_1, l_1, v_1), \dots, (t_k, l_k, v_k)) \quad (1)$$

其中 $t_j (j = 0, 1, \dots, k)$ 是时间戳,并且 $\forall 0 \leq j < k, t_j < t_{j+1}$; $l_j = (u_i \cdot l_j \cdot x, u_i \cdot l_j \cdot y)$ 是 t_j 时刻移动用户在 R^2 空间中的坐标;类似的, $v_j = (u_i \cdot v_j \cdot x, u_i \cdot v_j \cdot y)$ 是 t_j 时刻移动用户的速度.

显然,对于线性的运动模型,对于对象 u_i 在某个时间 $t_e > t$,在该时刻它的物理位置是

$$u_i \cdot l_{t_e} \cdot x = u_i \cdot l_t \cdot x + u_i \cdot v_t \cdot x(t_e - t) \quad (2)$$

$$u_i \cdot l_{t_e} \cdot y = u_i \cdot l_t \cdot y + u_i \cdot v_t \cdot y(t_e - t) \quad (3)$$

定义 1 移动区域预测

用户不间断地进行移动区域预测.移动区域预测是指对于一个特定的移动模型,以时间 t_r 为起始,在未来某个时间 t_e 内,移动用户 u_i 可能所在的区域 $PB(\text{PredictedBox}) = (t_r, t_e, LB, VB)$,其中 t_r 是预测区域的引用时间, t_e 是预测区域的过期时间, LB 是对象在这一时间段内的物理移动范围, VB 是对象在这一时间段内的速度变化范围.

本文没有对移动用户的速度进行预测,因此, LB 范围限定 $[LB \cdot x^{\uparrow}, LB \cdot x^{\downarrow}]$, $[LB \cdot y^{\uparrow}, LB \cdot y^{\downarrow}]$ 的取值范围是

$$LB \cdot x^{\uparrow} = l_{t_r} \cdot x, LB \cdot x^{\downarrow} = l_{t_r} \cdot x + v_{t_r} \cdot x(t_e - t_r) \quad (4)$$

$$LB \cdot y^{\uparrow} = l_{t_r} \cdot y, LB \cdot y^{\downarrow} = l_{t_r} \cdot y + v_{t_r} \cdot y(t_e - t_r) \quad (5)$$

定义 2 超限移动区域更新

移动用户运动范围超越预测区域,其真实位置在预测区域外,移动用户向服务器发送位置更新信息.服务器使用移动用户发送的位置更新信息进行位置更新操作,作为移动用户的当前位置.移动用户这种位置更新称为超限移动区域更新.

定义 3 基于近邻框的近邻检测

对于移动用户 u_i, u_j 的近邻距离标准为 $\epsilon_{i,j}$.在某一时刻检测其朋友列表中的用户 u_j 是否在其近邻范围

内,基于近邻框的近邻检测就是判断他们的坐标关系是否满足 $|u_i \cdot x - u_j \cdot x| \leq \epsilon_{i,j}$,且 $|u_i \cdot y - u_j \cdot y| \leq \epsilon_{i,j}$.

给定一个长宽分别为 A, B 的预测区域,两个用户 u_i, u_j 之间的 XY 轴方向距离关系有式.

$$\text{MinDX}(u_i, u_j) = \max\{|u_i \cdot x - u_j \cdot x| - u_i \cdot A - u_j \cdot A, 0\} \quad (6)$$

$$\text{MaxDX}(u_i, u_j) = |u_i \cdot x - u_j \cdot x| + u_i \cdot A + u_j \cdot A \quad (7)$$

$$\text{MinDY}(u_i, u_j) = \max\{|u_i \cdot y - u_j \cdot y| - u_i \cdot B - u_j \cdot B, 0\} \quad (8)$$

$$\text{MaxDY}(u_i, u_j) = |u_i \cdot y - u_j \cdot y| + u_i \cdot B + u_j \cdot B \quad (9)$$

u_i, u_j 之间确定具有近邻关系的条件是 $\text{MaxDX}(u_i, u_j) \leq \epsilon_{i,j}$ 且 $\text{MaxDY}(u_i, u_j) \leq \epsilon_{i,j}$.

u_i, u_j 之间确定不具近邻关系的条件是 $\text{MinDX}(u_i, u_j) > \epsilon_{i,j}$ 或 $\text{MinDY}(u_i, u_j) > \epsilon_{i,j}$.

显然,系统对在路网上上下文环境中运动的移动用户进行连续近邻检测时,使用移动区域预测算法就应充分考虑路网的特点,进行合理的预测,可以提高系统的总体效率.

3 近邻框检测方法分析

3.1 近邻框检测方法概述

服务器端记录有每个用户的朋友列表以及该用户的近邻标准.移动用户的索引使用 TPR-tree(time-parameterized R-tree)^[6].客户端负责发送包含自身当前位置信息和速度信息的移动区域预测 PB (update 阶段);接受服务器端邻近朋友的位置信息,根据自身的定义对近邻进行重新筛选和展示(refinement 阶段).服务器端接收移动用户的 PB 信息,更新该用户的跟踪记录,在必要时查询用户当前的位置信息(probe 阶段)重新计算其近邻列表,发送其邻近朋友的位置信息(notification 阶段).整个过程由以下三个步骤组成.

Step1 客户端移动区域更新

移动用户周期性的对自身位置进行定位,采集移动速度和位置信息.取得上一次发往服务器的预测 $PB = (t_r, t_e, LB, VB)$.首先判断当前时间是否已经超过 t_e ;然后判断当前位置与预测位置的距离是否超过安全阈值.然后向服务器发送一个新的预测.如果两种情况都不是,移动用户就等待下一个更新周期.

为了减少位置更新消息数量,用户 u 需要结合当前位置,以及所处的路网上上下文环境对预测的超时时间 t_e 进行优化. t_e 优化过程是一个迭代的过程.图 1 是一个自适应更新的例子.

Step2 服务器端基于近邻框的检测和通告

服务器对于每个超限移动区域更新进行近邻检测.本文假设移动用户之间具有相互的朋友关系,但是

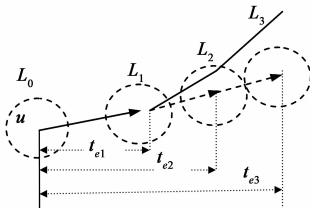


图1 自适应移动区域预测

不同用户可以定制近邻标准,也即不同用户可能具有不同的近邻标准.例如 u_j 是 u_i 的朋友,那么 u_i 也是 u_j 的朋友,但是他们具有不同的近邻标准 $\epsilon_{i,j}$ 和 $\epsilon_{j,i}$.

检测完成后,用户 u_i 的朋友就分为了三类.在近邻

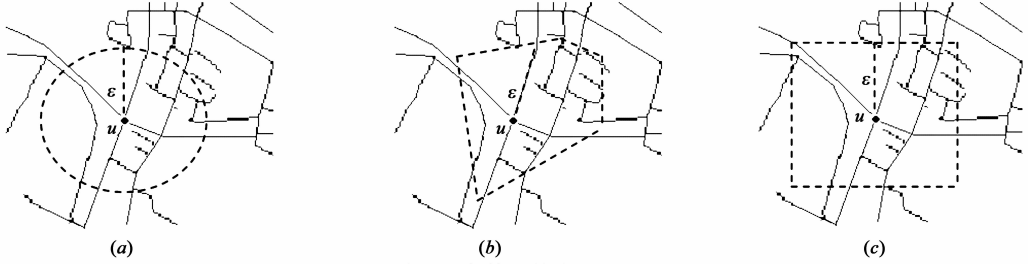


图2 近邻检测策略

服务器在得到用户 u_i 的检测结果后,将其近邻范围内的所有其朋友 ID、当前位置以增量方式通告给用户 u_i . 同时也将结果通告给用户 u_i 相关的朋友. 增量方式通告也即仅通告用户近邻关系的变化,而对没有变化的近邻关系不予重复通告.

算法 1 服务器端近邻检测

Algorithm 1 Proximity detection on server

```

1: initialize two-level heaps
2: receive updates from the user  $u_i$ 
3: for each friend pair  $(u_i, u_j)$  in  $u_i$ 's friend list do
4:   if  $(\epsilon_{i,j} < \epsilon_{j,i})$ 
5:     if proximity  $(u_i, u_j)$  then
6:       refresh two-level heaps with  $u_i$  and  $u_j$ 's trigger time
7:     else
8:       if proximity  $(u_j, u_i)$  then
9:         refresh two-level heaps with  $u_j$ 's trigger time
10:    else
11:     if proximity  $(u_j, u_i)$  then
12:       refresh two-level heaps with  $u_i$  and  $u_j$ 's trigger time
13:    else
14:     if proximity  $(u_i, u_j)$  then
15:       refresh two-level heaps with  $u_i$ 's trigger time
16: upon every time stamps:
17:   check global_queue
18:   notify users  $u_i$  with a proximity message if necessary

```

Step3 客户端近邻的展示

移动用户接收到服务器的近邻列表更新信息后,

框以外的,他们就不具有近邻关系,在近邻框以内的就具有近邻关系.由于服务器保持的是某个用户的移动预测区域,通过其速度只能粗略确定其所在位置.当用户移动区域刚好跨越用户 u_i 的近邻框时,系统就不能判断他们两者的近邻关系.这个时候就需要向该用户发送查询信息,通过其最新的移动区域预测了解当前的位置信息,进行近邻与否检测.

基于近邻框的检测过程仅通过判断两个移动用户的坐标关系,在 x 、 y 两个轴向任何一个方向距离超过用户的近邻标准定义即停止检测.图 2 是欧几里德距离、Dijkstra 最短距离和近邻框三种检测策略的示意图.

根据用户的设置进行近邻展示.

3.2 触发时间

服务器端使用了触发时间^[7]的概念,来提高移动用户之间位置关系查询结果之间的连续性.在触发时间的计算过程中,本文使用一种过滤方法简化触发时间的处理,同时采用一种双层优先级队列^[8]数据结构索引系统中用户的触发事件和近邻关系.

两个移动用户的触发时间 $T(u_i, u_j)$,在本文中是指 u_j 移动进入 u_i 近邻框的最早时间.假设 $(u_i.x, u_i.y)$ 和 $(u_j.x, u_j.y)$ 是在当前时间 t_c 后的某个 t 时刻 u_i 和 u_j 的位置,那么

$$T(u_i, u_j) = \min(t \mid t \geq t_c \wedge |u_i.l_t.x - u_j.l_t.x| \leq \epsilon_{i,j} \wedge |u_i.l_t.y - u_j.l_t.y| \leq \epsilon_{i,j}) \quad (10)$$

触发时间的计算需要同时考虑两个 u_i, u_j 的当前位置,移动速度.

例如用户 u_i 在 t 时刻的位置为

$$l_t.x = l_{t_c}.x + v_{t_c}.x \times t \quad (11)$$

$$l_t.y = l_{t_c}.y + v_{t_c}.y \times t \quad (12)$$

在基于近邻框的检测中,计算 u_i, u_j 触发时间就需要计算

$$\begin{cases} |u_i.l_{t_c}.x - u_j.l_{t_c}.x| \\ + (u_i.v_{t_c}.x - u_j.v_{t_c}.x) \times t \leq \epsilon_{i,j} \\ |u_i.l_{t_c}.y - u_j.l_{t_c}.y| \\ + (u_i.v_{t_c}.y - u_j.v_{t_c}.y) \times t \leq \epsilon_{i,j} \end{cases} \quad (13)$$

在基于欧几里德距离的近邻检测中,触发时间的需要计算

$$\begin{aligned} & (u_i \cdot l_{i_c} \cdot x - u_j \cdot l_{j_c} \cdot x) + (u_i \cdot v_{i_c} \cdot x - u_j \cdot v_{j_c} \cdot x) \times t)^2 \\ & + ((u_i \cdot l_{i_c} \cdot y - u_j \cdot l_{j_c} \cdot y) \\ & + (u_i \cdot v_{i_c} \cdot y - u_j \cdot v_{j_c} \cdot y) \times t)^2 \leq \epsilon_{i,j}^2 \end{aligned} \quad (14)$$

式(10)经过整理,是求解关于 t 的一元二次方程组.

对触发时间的过滤是指仅处理下次更新之前可能会发生近邻位置关系变化的朋友对.

3.3 性能分析

假设移动用户在 R^2 的平面空间运动,以时间 T 为周期检测其当前位置、速度等信息.同时假设用户每次预测移动区域为长宽 $A \times B$ 的矩形,预测移动距离为 $L = \sqrt{A^2 + B^2}$.所有用户使用相同的近邻定义 ϵ .假设系统中用户数为 n ,每个用户的朋友数为 m .

本节讨论系统中服务器处理移动区域预测消息,查询消息,通告消息的开销以及近邻检测的开销,假设每个消息处理的开销为 C_{message} ,近邻检测的开销为 $C_{\text{proximity}}$.

移动区域更新消息处理开销取决于客户端的移动区域更新次数和服务器单次消息处理的开销.客户端的移动区域更新次数与用户的运动速度 V ,所处的地理环境是相关的.每次预测的移动距离与用户的运动速度比之即为两次更新之间的平均间隔时间.那么周期 T 时间内的每个移动用户更新次数为 VT/L .对于 n 个移动用户,其更新开销为

$$P_{\text{update}} = \min\{VT/L, 1\} \quad (15)$$

$$C_{\text{update}} = n \times \min\{VT/L, 1\} \times C_{\text{message}} \quad (16)$$

对于查询消息的处理开销,需要分析移动区域刚好位于某个用户近邻框周围的概率.如图3所示,用户 u 周围的三个用户 u_a, u_b, u_c 分别位于 u 近邻框的内部,过渡位置和外部.需要查询确定最新的位置用户是 u_b . u_b 的分布范围如图中阴影部分所示.

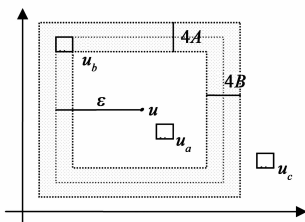


图3 probe概率分析

$$\begin{aligned} P_{\text{probe}} = & 4((\epsilon + 2A) \times (\epsilon + 2B) \\ & - (\epsilon - 2A) \times (\epsilon - 2B)) / R^2 \end{aligned} \quad (17)$$

由于假设每个用户平均都有 m 个朋友,该用户如果参与其 m 个朋友中任何一个的查询过程,他将收到

查询消息的概率为 $(1 - (1 - P_{\text{probe}})^m)$.另外,只有该用户当前没有发送移动区域更新消息的条件下才会收到查询消息,这种概率为 $(1 - P_{\text{update}})$.因此

$$\begin{aligned} C_{\text{probe}} = & 2n \times (1 - P_{\text{update}}) \\ & \times (1 - (1 - P_{\text{probe}})^m) \times C_{\text{message}} \end{aligned} \quad (18)$$

在通告阶段,使用增量的通告方式仅对有变动的近邻关系进行通知,通告消息的数量取决于单位时间内近邻朋友的变动速度.

实验研究发现,在预测移动区域与近邻区域比值较小的情况下,基于近邻框检测与其它两种方法相比 probe 阶段消息总数没有明显变化.实验同时也证实,在使用通用处理器的条件下,通过采用近邻框的检测方法,可以显著的减少单次近邻检测计算时间 $C_{\text{proximity}}$.

4 实验模拟

在本部分,对论文提出的近邻框检测方法进行了模拟实验.

实验使用两幅地图 Oldenburg (Old) 和 San Joaquin (San),如图4所示.在 Brinkhoff^[9]研发的模拟器基础上,使用 Java 语言开发实现本文的算法,并模拟不同交通状况条件下的位置更新和查询.模拟硬件环境为 Pentium Dual-Core CPU E5300 @2.60GHz,2G 内存.



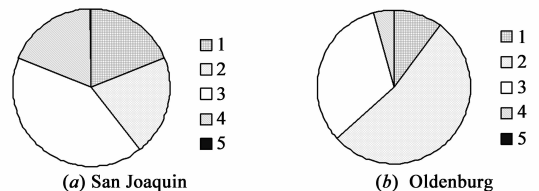
(a) San Joaquin

(b) Oldenburg

图4 实验地图

4.1 自适应移动区域预测效果

实验统计了两幅地图的节点度数分布如图5所示.统计了节点总数,与每个节点相连边的数目(度数),边的平均长度,以及自适应更新的平均距离 L ,结果如表1所示.



(a) San Joaquin

(b) Oldenburg

图5 路网节点度数分布

从表1中可知,两个实验中得到的移动区域平均距离都大于边的平均长度,显示了自适应策略的有效性.这里平均长度和道路的弯曲度两个因素对移动区域距

离预测的影响都得到了体现。

表 1 移动区域预测结果

地图名称	节点		边长平均	自适应预测平均距离
	总数	度数为 2		
Oldenburg	6105	3232	184	479
San Joaquin	18496	3806	2387	3277

4.2 近邻检测结果比较

在本实验,通过统计单位模拟时间内一个用户的近邻朋友和消息数的变化来说明近邻框检测的有效性.实验设置为总共 1020 个移动用户。

如图 6、8 所示,在两次实验中,近邻数量都随近邻标准的增大而增大.近邻框的检测方法(DistR)在每组统计数据里面都较欧几里德距离检测(DistO)和最短路径检测(DistD)大,显示出规律性的变化.因为路网中的移动用户数目相同,在较小规模的 Old 地图中平均近邻朋友数要比 San 大 11.4% 左右.图 7、9 是单位时间消息总数的变化,三种方法之间没有明显区别.表明在本文实验条件下,基于近邻框的检测对消息数量没有显著影响。

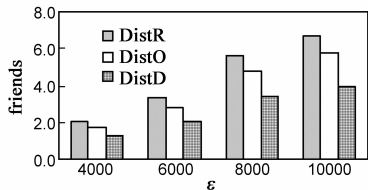


图6 单位时间邻近朋友数(Oldenburg)

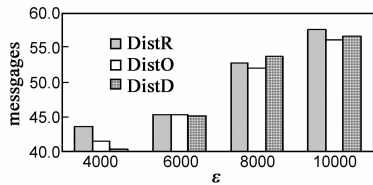


图7 单位时间消息数(Oldenburg)

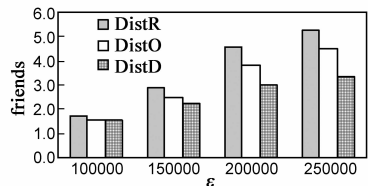


图8 单位时间邻近朋友数(San Joaquin)

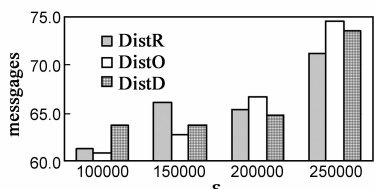


图9 单位时间消息数(San Joaquin)

4.3 基于近邻框检测效率的提高

使用 3.2 相同的实验过程,利用 JDK1.6 的 nanoTime()函数采集单次近邻查询的 CPU 开销,结果如图 10、11 所示。

图中横坐标是实验中每用户平均朋友数目,纵坐标是 CPU 时间.结果显示基于近邻框的检测 CPU 时间大约是基于欧几里德距离方法的 20.5%,是基于 Dijkstra 最短距离方法的 4.0%.而随着平均朋友数目的增加,CPU 时间呈线性增加趋势.近邻框和欧几里德距离方法的计算时间在两幅地图间没有显著区别,但是 Dijkstra 最短距离方法在较大规模的 San 地图中,计算时间增加了 28.1% 左右。

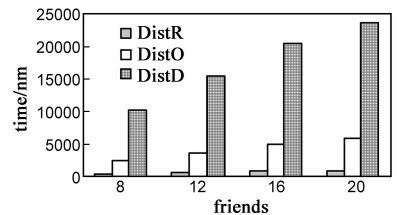


图10 近邻检测时间(Oldenburg)

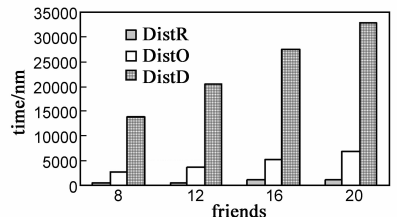


图11 近邻检测时间(San Joaquin)

4.4 优化效果

实验 4.4 对采用了两层优先级队列等方法优化的系统和没有优化的原始系统进行比较.实验对不同移动用户规模和平均朋友数目进行模拟,统计单次更新消息系统花费的 CPU 时间.结果如图 12 所示,平均优化程度约为 23.3%。

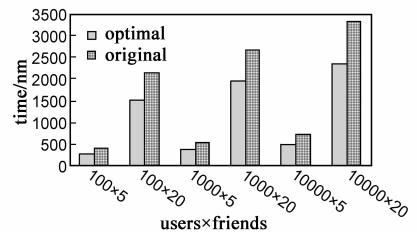


图12 系统优化效果

5 相关研究

Efrat 等^[1]提出了适用于移动用户近邻检测的一种分布式解决方案.Treu 等^[3]介绍了一种 C/S 模式的近邻检测方法,提出了移动扇区的概念,但 Man^[5]的分析显

示移动扇区策略在各种情况下对通信开销的优化都不特别明显。

Hans 等^[2]讨论了路网环境中的近邻检测问题. Xu 等^[4]将近邻检测问题一般化为受限检测问题. Shen^[10]和 Yin^[11]分别提出了对于移动用户基于草图和网格空间的查询. Civilis^[12]研究了矢量跟踪的方法. Chen^[13]提出了一种适用于欧式空间的自适应跟踪策略.

6 结论

在用户数较大的近邻检测中,中心服务器常常成为系统的瓶颈.本文提出了近邻框的概念,以提高服务器的处理效率.在客户端,辅以结合路网信息的自适应移动区域预测,减少总的通信消息数量,提高系统性能.文章介绍了近邻框检测的基本步骤,优化方法,对算法进行了理论分析,并通过实验验证了其可行性.下一步项目组将对算法进一步完善,并对其在实际应用中可能产生问题的进行研究.

参考文献

- [1] Amir A. et al. Buddy tracking-efficient proximity detection among mobile friends [A]. IEEE INFOCOM Proceeding [C]. Hong Kong: Institute of Electrical and Electronics Engineers Inc, 2004. 298 – 309.
- [2] Kriegel H. P. et al. Continuous proximity monitoring in road networks [A]. GIS Proceeding [C]. Irvine: Association for Computing Machinery, 2008. 85 – 94.
- [3] Treu G. et al. Efficient proximity detection among mobile targets with dead reckoning [A]. MobiWAC Proceeding [C]. Terromolinos: Association for Computing Machinery, 2006. 75 – 83.
- [4] Xu Z. et al. Adaptive location constraint processing [A]. SIGMOD Proceeding [C]. Beijing: Association for Computing Machinery, 2007. 581 – 592.
- [5] Man L. Y. et al. Efficient proximity detection among mobile users via selftuning policies [A]. VLDB Proceeding [C]. Singapore: Association for Computing Machinery, 2010. 985 – 996.
- [6] Tao Y. et al. The TPR * -tree: an optimized spatio-temporal access method for predictive queries [A]. VLDB Proceeding [C]. Berlin: Association for Computing Machinery, 2003. 323 – 335.

- [7] Iwerks G S. et al. Maintenance of K-nn and spatial join queries on continuously moving points [J]. ACM TODS, 2006, 31 (2): 485 – 536.
- [8] Subramani K. et al. Two-level heaps: a new priority queue structure with applications to the single source shortest path problem [J]. Computing, 2010, 90(3 – 4): 113 – 130.
- [9] Thomas B. Generating network-Based moving objects [A]. International Conference on Scientific and Statistical Database Management Proceeding [C]. Berlin: Springer, 2000. 253 – 256.
- [10] 申世群, 等. 基于草图的空间数据检索研究 [J]. 电子学报, 2010, 38(8): 1819 – 1824.
SHEN Shi-qun. et al. Research on spatial data retrieval based on sketch. [J] Acta Electronica Sinica, 2010, 38(8): 1819 – 1824. (in Chinese)
- [11] 殷晓岚. 动态网络空间中的 k – NN 查询 [J]. 电子学报, 2011, 39(2): 389 – 395.
YIN Xiao-lan. k-nearest neighbors query in dynamic spatial network databases [J]. Acta Electronica Sinica, 2011, 39(2): 389 – 395. (in Chinese)
- [12] Civilis A. et al. Techniques for efficient road-network-based tracking of moving objects [J]. IEEE TKDE, 2005, 17(5): 698 – 712.
- [13] Chen S. et al. An adaptive updating protocol for reducing moving object database workload [A]. VLDB Proceeding [C]. Singapore: Association for Computing Machinery, 2010. 735 – 746.

作者简介



徐 建 男, 1975 年出生于浙江, 杭州电子科技大学计算机学院教授. 研究方向为计算机网络, 分布式计算, 智能控制等.
E-mail: jian. xu@hdu. edu. cn

郭 鸣 男, 1972 年出生于江苏, 浙江大学城市学院计算机系副教授. 研究方向为分布式计算, 语义 web 等.
E-mail: guom@zucc. edu. cn