移动计算环境下路网上移动对象的位置更新

唐 蔚 张栋梁 范媛媛
(同济大学计算机科学技术系 上海 201804)

摘 要 针对移动网络环境下移动对象的位置更新、断连检测和处理的问题,提出了现有位置更新策略的改进算法。采用固定时限策略进行断连检测,虽然能够获取快速而准确的断连检测,但是直接增加了位置更新的次数,降低了原有位置更新策略的性能。为了改正该缺点,提出了自适应策略,即根据移动对象的位置估计一个最晚的更新时限,实现强制更新。此外,还提出了群组更新策略中断连的处理算法。实验仿真结果表明,新算法既能减少位置更新次数,又能实现有效合理的网络处理。

关键词 移动网络环境,位置更新,断连检测,断连处理
中图法分类号 TP3-0 **文献标识码** A

Research on Location Updates of Network-constrainted Moving Objects in Mobile Computing Environment

TANG Wei ZHANG Dong-liang FAN Yuan-yuan
(Department of Computer Science and Technology, Tongji University, Shanghai 201804, China)

Abstract New algorithms were proposed in this paper based on the segment-based policy, in order to detect disconnection in mobile computing environment and update location efficiently. First algorithm uses fix-time policy, can detect disconnection quickly and efficiently, but cause more numbers of location updates. Second algorithm uses adaptive time limit policy and finds a good balance. At last group-based policy will be modified to detect and deal with disconnection. Our experiments show that these algorithms not only decrease numbers of location updates, but also can detect network disconnection efficiently.

Keywords Mobile computing environment, Location update, Disconnection detection, Disconnection handling

1 引言

随着具有自定位能力的移动计算设备的大量普及,基于位置服务(Location-Based Service,LBS)成为近两年发展最快的产业之一。空间中的移动对象可以根据所在位置做出相应的计算,以提供相应的信息和服务。基于位置服务能提供军事指挥、交通监管、广告推送、小区域信息发布和自动签到等服务。

LBS应用的难点在于大量移动对象的管理,而主动的位置更新策略是基础。具体的应用都是对大量的移动对象提供服务,单纯以一定频率更新位置,所需的带宽和计算能力只有超级计算机才能应付。此外,GPS位置精度一般在一定范围内,服务器只能知道其历史位置信息。

目前位置更新策略的研究,大多基于有限带宽的假设,努力在保证位置精度的情况下,更新对象位置。对于一般低速的移动对象来说,这是非常成功的。但是如果对高速移动物体提供LBS应用,移动网络的频繁断连和不稳定性就会对系统的性能造成很大的影响。

在一些实时性要求较高的场景,特别是军事指挥系统、检测系统中,用户希望知道网络连接是否正常及其对位置的影响。下面两种形式是必需的:

1)网络连接正常。A现位于 x 点,偏移为 d。

2)网络连接已中断。A在 t_1 至 t_2 时刻位于 x 点和 y 点之间,偏移为 d。

如果网络已中断,2)所能提供的信息比1)更有用。但是为了判断网络是否连接正常,必然会占用一定的网络资源,潜在地增加了更新次数。

第2节阐述了相关工作;第3节对问题涉及的指标做了详细的分析;第4节则介绍了基本的概念和框架;第5节则提出了改进算法;第6节给出了实验仿真结果;最后总结全文。

2 相关研究

在位置更新的同时,考虑断连检测和处理,研究工作不是很多。Wolson的dtdr[1](the disconnection detection dead reckoning policy)策略是第一个明确在更新中对断连进行检测的策略。通过使用变动的偏移阈值,dtdr策略巧妙地通过周期减少更新的阈值上限来进行断连检测。一开始更新时,dtdr策略选择了偏移阈值 K,经过了 t 个时间单位后,使用 K/t 来作为 t 时刻的更新阈值。因而,如果客户端网络连接正常,经过的时间越长,更新阈值越小,从而发起更新的概率就越

到稿日期:2011-07-03 返修日期:2011-09-30 本文受937课题(2010CB328101),国家自然科学基金青年基金项目(61003089)资助。
唐 蔚(1986—),男,硕士,主要研究方向为智能交通,E-mail:alburthoffman@gmail.com;张栋梁 男,硕士,主要研究方向为交通仿真;范媛媛 女,硕士,主要研究方向为软件可靠性。

大。如果客户端长时间没有更新，客户端网络断连的可能性也就随之增加。dtdr 策略在每次更新时，会通过预测，计算出下一个更新阈值 K，使得总的信息代价最小。

dtdr 策略的优点是避免了服务器端对客户端的查询操作。网络断连的概率越大，客户端发起更新的概率越大。而且一旦达到阈值而网络断接，客户端就会不断尝试更新位置，直到网络连接正常。由于客户端只要有一次成功更新，就会进入下一轮更新周期，因此并没有过多增加上行的流量。dtdr 从某种意义来说是一个概率算法，客户端和服务端都不知道是否真的断连，而只有一个断连的概率。

实验表明，dtdr 策略的更新代价优于采用固定偏移策略，却要比自适应偏移策略差，其差距可以看成是断连的检测代价。

最近的 GBL[2]（Group-Based Location updating）策略提供了一个位置更新和断连检测的新视角。与尝试减少个体的更新次数不同的是，GBL 尝试对个体进行分组，然后以组群为单位对位置进行更新。因而 GBL 重点是寻找有效的分组算法，从而减少分组数目。理论和实验都表明，GBL 策略极大地减少了更新的次数。

GBL 的位置更新要先报告给组中的中心点，再由中心点将整组的位置报告至服务端。其实网络流量并没有减少，只是转移到了客户端之间。客户端要能自动计算分组的中心点，必须具有较强的计算能力。此外网络断连会改变 GBL 的分组状态，导致动态的中心点变化。GBL 采用时限法作为断连的判断依据。如果一个位置更新没有在固定的时限内到达，GBL 将判断为断连。这种策略会大大增加更新的次数，导致不必要的更新，从而影响 GBL 的性能。

3 指标分析

位置更新策略的共同目标就是减少更新次数，减轻服务器端压力，同时保证位置查询的精度[3,4]。减少更新次数分为两种：个体和分组，这两种策略都是可行的，不管何种策略，总的更新次数都是算法性能的重要指标。

其次，考虑用户位置的固有不确定性。无论采用何种更新策略，服务器预测的位置和用户实际位置之间必然存在偏移。在一定的偏移范围内，用户的实际位置是不确定的，可能是偏移区域中随机的一点。位置更新策略中，必须要保证位置查询的精度尽可能小，体现为与实际位置的偏移尽可能小。

为了对网络断连进行有效的检测，还必须保证系统的反应时间。反应时间具体是指从客户端网络断连到服务端检测出来所经过的时间。反应时间越短，系统越能提供一个更精确的位置估计。在军事指挥系统中，反应时间时常关乎一次行动的成败。但是，为了提高系统的反应时间，势必要增加客户端位置更新的次数。

断连检测的误判率也是系统要关心的。所谓误判率，就是在特定网络连接的状况下客户端连接正常时服务端对客户端连接状态所做的错误判断的概率。这个概率越大，断连检测的结果越不可信。在成组更新策略中，服务器会根据网络的断连尝试重新计算群组的中心点，错误判断率越高，服务器端所要做的无用功就越多。

综上，位置更新策略，必须要能有效减少更新次数，保证位置查询精度；同时对网络断连进行有效检测，反应时间合理，误判率越小越好。由于断连检测和减少更新次数之间存在内在矛盾，这个目标是不容易达到的。

4 概念和系统框架

4.1 基本概念

假设 1 物体在一个二维平面上运动。二维平面有 x 和 y 两个正交的实数坐标轴，可以看成是数学中的二维平面。

定义 1（点） 二维平面上的一点，坐标 (x,y) 标示其位置，用 p 来表示。

定义 2（直线） 两个点之间的直线，可以使用 (p_0,p_1) 来标记，用 l 来表示。

定义 3（向量） 有方向的直线，其方向从 p_0 指向 p_1，用 $\vec{v}$ 来表示。

定义 4（折线） 点集合 $(p_0,p_1,\cdots,p_{n-1})$，且对 $0\leqslant i<n-1$，(p_i,p_{i+1}) 是一个向量。整个折线用 pl 来表示，其方向从 p_0 指向 p_{n-1}。

定义 5（移动对象） 一个三元组 $(p,\vec{v},t)$，用 ml 表示。其中，p 表示移动对象在 t 时刻的位置，$\vec{v}$ 表示移动对象在 t 时刻的速度和方向，t 是一个时间点。

定义 6（路） 路是一条折线，用 r 来表示。

定义 7（路口） 路的交点称为路口，用 j 来表示。

定义 8（路网） 有向图，一个路和路口的集合 (R,J)。对于任意一条路 r，其只有起点 p_0 和终点 p_{n-1} 属于集合 J。

定义 9（路网上的移动对象） 一个四元组 $(rid,rdis,rv,t)$，记作 $mlor$，其中，rid 表示移动对象 t 时刻在路网中路的编号，$rdis$ 表示移动对象所在位置到路起始点的距离，rv 表示移动对象 t 时刻在路上的移动速度（正数表示运动方向和路的方向一致），t 表示一个时间点。

4.2 系统架构

图 1 描述了一个 LBS 系统中位置更新的典型过程。客户端通过 GPS 传感器获取当前位置 ml，由位置预测函数 lpf 获得一个预测位置 p。如果偏移 $(ml.p-p)$ 超过了阈值 th，则客户端发起一次位置更新。

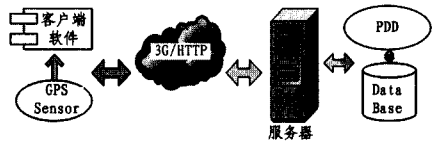

图 1 典型的 LBS 系统结构

服务器获取到客户端的位置更新后，会计算出相应的 $mlor$，更新位置数据库，计算出新的预测函数 lpf，并将 lpf 传回客户端。在对客户端位置的查询中，服务器会判断当前查询时间是否在有效时间内。如果是，则表示网络连接正常，服务器计算出客户端预计的位置并返回给用户；否则服务器提示用户客户端已断接，并给出一个大致的范围。

5 算法与分析

5.1 位置预测函数

Civilis 等人详细描述了段策略[5]（segment-based policy）的实现原理，并对其进行仿真，得出段策略相对优于点策略和向量策略的结论。Civilis 等人的段策略采用的预测函数是基于上次更新的速度向量的时间函数。这在位置预测的研究中又被称为延时预测模型。在延时预测中，未来时刻的速度标

量值和方向都与最后一次更新的标量值和方向一致。

$v_n = v_{n-1}$

延时模型[6,7]很好地适应了移动对象的速度变化，但是却不能提供很好的预测值。因而，对 Civilis 的段策略做了如下改进。

假设已知 t_0 时刻的速度 v_0，t_1 时刻的速度 v_1，且两个时刻间的距离是 d。设速度是时间的函数：

$v(t) = at$

则有：

$$d = \int_{t_0}^{t_1} v(t)\mathrm{d}t$$

取下时刻的速度预测函数为：

$$v'(t) = \frac{d}{2(t_1 - t_0)^2}t$$

式中，t 为自 t_1 时刻开始的时间数。

据此给出位置预测函数 lpf 的定义。

定义 10（位置预测函数 lpf） 一个三元组（$mlop$，a，$threshold$）中，$mlop$ 是路网上的移动对象，a 是速度预测函数的系数值，$threshold$ 是更新的偏移阈值。

5.2 FTLA（Fixed Time Limit Algorithm）

客户端在每次更新时除了检查位置偏移是否超过一定阈值，还判断距离上次更新的时间间隔是否超过固定的时限。如果超过时限，则客户端会强制执行一次更新。服务端在收到一次更新后，会计算出下次更新的最晚更新时间。最晚更新时间后的查询结果都会被认为是网络断连。

考虑到网络延迟，最晚更新时间的计算必须减去网络传输的时间。设 ft 表示固定的时限，而 D 表示平均网络延迟，则最晚更新时间 wt(waiting until time)为：

$wt = ml.t + ft - D$

平均网络延迟的计算可以使用如下的递归式：

$D_0 = 0, D_1 = d_1$

$$D_i = \frac{(i-1)D_{i-1} + d_i}{i}$$

式中，d_i 表示第 i 次更新时的网络延迟，D_i 是前 i 次更新的平均网络延迟。

客户端和服务器端的算法描述如下。

算法 1（客户端）

```
get ml from GPS sensor
update to Server and get prediction function lpf
repeat
    get ml from GPS sensor
    l=lpf.mlop.rdis+lpf.a*2*(Now()-lpf.mlop.t)^2
    get point p which is l far away from lpf.mlop.rid's p0
    if |p-ml.p|>lpf.threshold or Now()-lpf.mlop.t>ft then
    update to server and get lpf
while not finished
```

（服务器端）

```
k=0
D=0
repeat
    get ml from Client
    get relative mlop with ml
    get last update object mlop'
    k=k+1
    D=((k-1)D+Now()-ml.t)/k
    lpf.mlop=mlop
    lpf.a=(mlop.rdis-mlop'.rdis)/(2(mlop.t-mlop'.t)^2)
    lpf.threshold=threshold
    store lpf,wt to the database
    return lpf to Client
while not finished
```

算法 1 中，服务器只对位置进行更新，没有任何断连检测的逻辑。这是因为断连检测要等到查询用户位置时才会处理。之后的算法都使用了如下的位置查询算法。

算法 2 QDDA(Query with Disconnection Detection Algorithm)

```
get lpf,wt from database
if Now()<wt then
    caculate future location by lpf and return to user
else
    Output 'Mininum disconnection time is',Now()-wt
    caculate the possible locatioin by lpf within time region [lpf.mlop.t,wt]
```

FTLA 算法使用固定的时限来实现断连检测，其性能的好坏决定于 ft 的选择。ft 越小，反应时间越小，但是客户端的更新次数的下限也就越大；反之亦然。

由于网络的不稳定，单纯使用网络延迟的平均来预测消息接收时间，FTLA 算法的误判率会比较大。

5.3 ATLA(Adaptive Time Limit Algorithm)

ATLA 算法不再采用固定的时限，而是估计下一时刻更新的时间，并作为位置更新的下限。在段策略中，物体到达路口时会有一次更新，因而可以使用上次更新位置到达路口预计的时间作为时限。这样，由于路口的更新是必要的，这个方法对更新次数的影响可以忽略不计，从而克服 FTLA 算法的更新下限问题。

在服务器端，ATLA 算法与 FTLA 的区别是使用了一个修正值 a 来修正预计的消息接收时间。这个修正值通常为 10 秒左右，可以使用动态的计算方法来计算。具体的算法描述如下。

算法 3（客户端）

```
get ml from GPS sensor
update to Server and get prediction function lpf
repeat
    get ml from GPS sensor
    l=lpf.mlop.rdis+lpf.a*2*(Now()-lpf.mlop.t)^2
    get point p which is l far away from lpf.mlop.rid's p0
    if |p-ml.p|>lpf.threshold or p=pn or p=p0 then
      update to server and get lpf
while not finished
```

ATLA 算法的客户端实现没有对时限的判断，而服务器端代码也只增加了 wt 的修正计算。可以看出，ATLA 算法的更新次数必然少于 FTLA，但是其反应时间要略大于 FTLA。同时，为了有效处理错误判断问题，wt 的计算增加了一个偏移 a，它能有效降低误判率。

5.4 群组更新中对断连的检测和处理

群组更新 GBL 与个体的更新不同，群组更新致力于减少更新的个体的数目，同时对个体使用有效的更新策略，使得总

体上极大降低位置更新的次数。

中国人民大学的陈继东等在吸收 GBL 算法的基础上，提出 GSP[8,9]（Group location update strategy based on the SP model）算法，即只对一条路上面的移动对象进行分组。GSP 不要求客户端具有自组网能力，且每次只更新中心点附近元素的位置。GSP 对移动对象的分组是服务器端的计算任务，而不是客户端自组网的结果。

GSP 算法的预测模型是基于孟小峰教授提出的元胞自动机理论，因而从理论上具有更精确的位置预测结果。元胞自动机理论在交通流仿真中有较好的表现，但是用来对移动对象进行建模，还需要进一步的验证。而且由于 GSP 中一旦分好组，到达下一个路口期间就不会发生变化，每次更新也是由群组的中心点来更新，因此其并不能很好反映交通路况的变化。本文的研究借鉴了 GSP 的思想，只对同一条路线上的移动对象进行分组，但不采用元胞自动机作为预测的理论模型，将重点放在群组更新中对断连的检测和处理上。

5.4.1 *移动对象的分组*

因为只对同一条路线上的对象进行分组，所以将运动方向相同且将来运动轨迹相近的移动物体分到同一组。由第 4 节的描述可知，移动物体的距离预测函数可以表示成：

$$d(t) = \int_{t_0}^{t} v(t)\mathrm{d}t = \frac{1}{2}at^2$$

式中，a 为更新时的计算值。

这里的预测函数是一个二次函数。只要两条抛物线满足在当前时刻以及到达路口处的偏移不超过给定的阈值，就可以将对应的移动对象分到同一组。

5.4.2 *群组更新*

移动对象在某次更新中会被分到一个群组中去。此后直到中心点到达路口为止，该中心点将一直作为群组的代表，向服务器更新位置。对于断连的检测，群组更新中采用类似 FTLA 算法的做法，但是其他点的更新阈值 $threshold_m$ 要比中心点的阈值 $threshold_l$ 大一些，同时更新时限 ft_m 也比中心点 ft_l 大。这样，在中心点失效后的一段时间内，群组内其他的移动对象将会发起更新，导致重新分组。

服务器接收到移动对象的更新时，如果发现中心点已经失效，并且移动对象又不是中心点，则会计算该对象和群组中其他对象的粘合度。根据其粘合度，移动对象决定是否成为调整后的群组的中心点。群组中其他对象的偏移阈值都是一样的，一旦中心点不能更新，它们大致就会在一个较短时间内主动更新，从而完成新的分组计算。

算法 4 GUDD（Group Update with Disconnection Detection）

```
get ml from Client
get relative group information g
if obj not enter the new edge then
    if obj is not group's leader then
    make obj become group's leader
    cacluate mlop, threshold_l and ft_l, save to database and return to client
    else
        just do learder's job
else
    just adjust group's structure as normal
```

上述算法的性能受到 $threshold_m$ 和 ft_m 的影响，如不考虑断连处理，则可以认为两个变量都是无限大的，即 GSP 算法变形了。

6 实验与仿真

实验数据来自同济大学网格计算中心智能交通项目。该数据为上海市 3 个月内 2300 多辆出租车的 GPS 原始数据。民用 GPS 的精度大致为 25 米，实验所用的位置更新阈值最小为 25 米。FTLA 算法需要预先确定固定的更新时限，车载 GPS 的更新频率为每秒 3 次，所以实验选择的时限为 10 秒、30 秒和 1 分钟，分别标记为 FTLA1、FTLA2 和 FTLA3。GUDD 策略中，非中心点的时限和阈值都要比中心点大，实验中取中心点的 1.5 倍。

实际位置更新时，每个移动对象的网络状态都是不同的，在实验中很难做到这一点。为了比较 3 种算法（FTLA、ATLA 和 GUDD），需要在相同网络环境下测试各种数据，因而实验对移动网络的仿真做了简化，每次实验都让所有的移动对象大致具有相同移动网络状态。

由图 2 知，FTLA1 的位置更新率最高，这是由于更新时限太短的缘故。其次是 FTLA2 和 FTLA3，更新时限设得越大，算法的更新率越接近于 ATLA 算法。GUDD 的更新率要好于 ATLA 算法，但由于采用了时限，因此没有达到 GSP 算法那么好的性能。

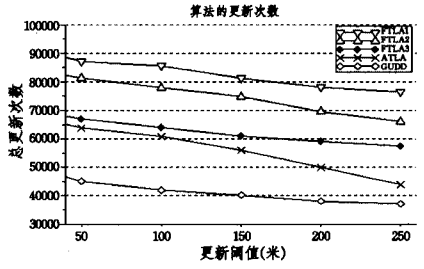

图 2 算法的更新次数比较

表 1 显示了 3 种算法的平均反应时间和误判率。FTLA 算法的平均反应时间都差不多，这主要是受到网络延迟的影响。ATLA 算法的反应时间比较大，但仍能在合理时间内完成。GUDD 算法的反应时间也差不多。ATLA 的误判率比较低，这主要是由于引入了一个修正因子。

表 1 反应时间（ms）和误判率（%）

	FTLA1	FTLA2	FTLA3	ATLA	GUDD
ms	5237	5201	5235	7692	5245
%	1.21	1.09	1.17	0.79	1.13

结束语 FTLA 算法使用了传统固定时限模型。其固定时限的选择成为位置更新次数、位置精度和反应时间的重要影响因子。ATLA 策略则摆脱了这种极端矛盾的关系，使用路口更新和修正值对 FTLA 进行改进。GUDD 算法尝试在 GSP 思想的基础上加以改进。通过增加断连检测和处理，GUDD 不但能做出断连检测，还可以进行重新分组。

实验表明，ATLA 算法相对优于 FTLA 算法，能在不增加段策略更新次数和位置精度的情况下，获得较低的误判率和反应时间。GUDD 和 GSP 相比，所增加的更新次数都是由

（下转第 138 页）

200 次估计中，采用原始操作剖面 P 时可靠性估计的平均值为 0.9977，估计方差的平均值为 4.7213×10^{-3}；采用由模拟退火算法得出的测试剖面 T 时的可靠性估计的平均值为 0.9981，估计方差的平均值为 1.8412×10^{-3}；采用由交叉熵方法得出的最优测试剖面 Q^* 时的可靠性估计的平均值为 0.9975，估计方差的平均值为 7.9444×10^{-4}。采用最优测试剖面可以显著降低估计的方差，且交叉熵方法要略优于模拟退火算法。由盒状图可知，采用最优测试剖面时数据最集中，而采用标准方法时，数据波动最大，模拟退火算法次之。

在 200 次仿真模拟过程中，遍历路径数为 1000 条时，由操作剖面、模拟退火算法得出的测试剖面、最优测试剖面分别生成的测试路径中关键操作累计遍历次数被记录，各关键操作的平均遍历次数如表 1 所列。

表 1　200 次仿真模拟中各关键操作平均遍历次数

	操作 6	操作 9	操作 10
标准方法	86.5	28.66	105.3
模拟退火方法	158.3	120.3	230.6
交叉熵方法	235	95	350

由此可见，采用交叉熵方法和模拟退火算法都可明显提高稀有操作的遍历机会。交叉熵方法对关键操作 6 的遍历机会要大于模拟退火算法，而对关键操作 8 和 9 的遍历机会要低于模拟退火算法。

结束语　本文基于重要抽样技术，利用交叉熵通过一种修正机制调节操作剖面，在保证可靠性估计是无偏估计的前提下，降低估计的方差，充分发挥重要抽样技术的优点，增加使用概率小的关键操作的测试机会，加速软件测试，在提高软件系统质量的同时降低软件总费用。该方法是为实现测试目标而采用的一种导向性测试数据集模拟生成方法，它利用“功效”值最好的测试数据集包含的失效信息，把修正测试剖面归结为一个迭代优化问题迭代求解，估计安全关键软件可靠性，加速软件测试。

参 考 文 献

[1] 赵亮，王建民，孙家广. 统计测试的软件可靠性保障能力研究[J]. 软件学报，2008，19(6)：1379-1385

[2] 杨善林，丁帅，褚伟. 一种基于效用和证据理论的可信软件评估方法[J]. 计算机研究与发展，2009，46(7)：1152-1159

[3] 覃志东，雷航，桑楠，等. 安全关键软件可靠性验证测试方法研究[J]. 航空学报，2005，26(3)：334-339

[4] Gutjahr W J. Failure risk estimation via Markov software usage models[C]//Schoitsch E. ed. SAFECOMP 96，Proc. of the 15th International conference on computer safety，reliability and security. Springer，1997：183-192

[5] Gutjahr W J. Importance sampling of test cases in Markovian software usage models[J]. Probability in the Engineering and Informational Sciences，1997，11：19-36

[6] Gutjahr W J. Software dependability evaluation based on Markov usage models[J]. Performance Evaluation，2000，40(4)：199-222

[7] Doerner K，Laure E. High performance computing in the optimization of software test plans[J]. Optimization and Engineering，2002，3：67-87

[8] Doerner K，Gutjahr W J. Extracting test sequences from a Markov software usage model by ACO[C]//LNCS. Springer Verlag，2003，2724：2465-2476

[9] 颜炯，王戟，陈火旺. 基于重要抽样的软件统计测试加速[J]. 计算机工程与科学，2005，27(3)：64-66

[10] Yan J，Zhou K P，Deng C H，et al. Importance Sampling Based Safety-Critical Software Statistical Testing Acceleration[C]// International Conference on Computational Intelligence and Software Engineering (CiSE). 2010：1-4

[11] 徐云青，徐义峰，李舟军. 基于使用模型的软件可靠性加速测试[J]. 计算机应用与软件，2009，26(3)：147-148

[12] 张德平，聂长海，徐宝文. 软件可靠性评估的重要抽样方法研究[J]. 软件学报，2009，20(10)：2859-2866

[13] 张德平，聂长海，徐宝文. 测试资源受约束的安全关键软件加速测试方法[J]. 计算机科学，2009，36(5)：138-141

[14] 张德平，查日军. 基于 Markov 链使用模型的加速统计测试方法[J]. 东南大学学报，已录用

[15] Chari K，Hevner A. System test planning of software：an optimization approach[J]. IEEE transactions on software engineering，2006，32(7)：503-509

[16] Orsak G C，Aazhang B. Constrained solutions in importance sampling via robust statistics[J]. IEEE Trans. Inform. Theory，1991，37：307-316

(上接第 109 页)

额外的断连检测和处理所引起的。

参 考 文 献

[1] Wolfson O，Chamberlain S，Dao S，et al. Cost and Imprecision in Modeling the Position of Moving Objects[C]//Proceedings of the Fourteenth International Conference on Data Engineering (ICDE). 1998

[2] Lam G H K，Leong HV，Chan SC. GBL：Group-based location updating in mobile environment[C]//Proceedings of the 9th International Conference on Database Systems for Advanced Applications. Jeju Island，Korea，2004：762-774

[3] Wolfson O，Sistla A P，Chamberlain S，et al. Updating and Querying Databases that Track Mobile Units[J]. Distributed and Parallel Databases，1999，7：257-287

[4] Wolfson O，Xu B，Chamberlain S，et al. Moving Objects Databases：Issues and Solutions[C]//Proceedings of the Tenth International Conference on Scientific Database Management (SSDBM). 1998

[5] Civilis A，Jensen C S，Nenortaite J，et al. Effient Tracking of Moving objects with Precision Guarantees[C]//Proceedings of the First Ann. International Conference on Mobile and Ubiquitous Systems：Networking and Services (Mobiquitous). 2004

[6] Lam K Y，Ulusoy O，Lee T S H，et al. An efficient method for geneating location updates for proceeding of location-dependent continuous queries[C]//Proceedings of the 6th International Conference on Database Systems fro Advanced Applications. Hong Kong，China，2001：218-225

[7] Wolfson O，Yin H. Accuracy and resource consumption in tracking and location prediction[C]//Proceedings of the 7th International Symposium on Spatial and Temporal Databases. Santorini Island，Greede，2003：325-343

[8] 胡志智，孟小峰，郭妍妍，等. 基于模拟预测的移动对象位置主动更新策略[J]. 计算机研究与发展，2004，41(增刊)：43-49

[9] Chen J D，Meng X F，Li B，et al. Tracking network-constrained moving objects with group updates[C]//Proceedings of the 7th International Conference on Web-Age Information Management (WAIM 2006). Hong Kong，China，2006：158-169