

# 移动计算环境中基于 Agent 技术的语义缓存一致性验证方法

梁茹冰<sup>1</sup> 刘琼<sup>2</sup>

(华南农业大学理学院 广州 510642)<sup>1</sup> (华南理工大学软件学院 广州 510006)<sup>2</sup>

**摘要** 回调算法是一种由服务器驱动的缓存管理算法,存在“写延迟”和终端断接重连时需重新验证缓存的问题,为此提出利用代理技术验证缓存一致性的方法。首先,给出 Client/MSS/Server 结构,设计并说明各层代理的功能;其次,从客户端的数据访问操作和服务器端的写操作两方面讨论缓存一致性验证方法。利用代理管理客户端缓存和转发失效数据,既可满足终端断连的需要,也不致产生服务器端的写延迟,并能够保持数据的强一致性。实验表明,所提方法使终端断接重连的查询响应时间更快,并能弥补回调算法的不足,更加适用于终端频繁移动、断接的无线网络环境。

**关键词** 移动计算,代理,语义缓存,一致性验证,写延迟

**中图分类号** TP311 **文献标识码** A

## Method of Semantic Cache Consistency Checking in Mobile Computing Environments Based on Agent Technology

LIANG Ru-bing<sup>1</sup> LIU Qiong<sup>2</sup>

(College of Science, South China Agricultural University, Guangzhou 510642, China)<sup>1</sup>

(School of Software Engineering, South China University of Technology, Guangzhou 510006, China)<sup>2</sup>

**Abstract** Callback algorithm is a cache management method which is driven by server, but there are some problems in this approach such as writing delay, and cached data needs to be revalidated when network is reconnected. This paper proposed a novel semantic cache scheme using agent technology to check cache consistency. Firstly, the Client/MSS/Server architecture was presented and the functions of agents were given. Secondly, we discussed cache consistent checking method which is aroused by data query operations from terminals and write operations from server. Using agents to manage terminal's local cache data records and forward server's invalidating data, this scheme can not only satisfy terminal's disconnection, but also reduce writing delay time, and maintain cache strong consistency. The results of experiments show that the proposed cache maintenance method can speed up the query response time and overcome the disadvantage of callback algorithm mentioned above, thus is better adapted for the frequent mobility and disconnect network environments.

**Keywords** Mobile computing, Agent, Semantic cache, Consistency checking, Write delay

### 1 引言

移动计算是消息传递技术和无线通信技术的融合,它将使计算机或其它信息智能终端设备在无线环境下实现数据传输及资源共享。在这种计算环境中,有很多亟待解决的问题:比如移动设备本身局限性带来的能量问题、无线通信下的低带宽和带宽利用问题、用户及设备的移动性和频繁断接带来的一系列问题等。因此,移动计算模型的核心问题就是要确定移动终端、服务器的功能如何分配及如何根据需要动态进行调整。

移动计算模型大体上分为5种,它们与移动计算环境的适应情况如表1所列。可见,代理(Agent)技术与移动环境有天然的匹配性<sup>[1]</sup>, Agent在减少网络延迟、支持轻载移动设备、异步信息搜索、数据访问能力等方面的优势是其他方法所

不具备的。

表1 移动计算模型与移动环境适应情况一览表

移动计算模型	适合移动环境的特征 适合连接 不稳定的 无线网络	适合低带 宽的无线 网络	适合计算 能力弱的 移动终端	终端与网 络断开仍 能计算
传统的客户/服务器模型	弱	弱	弱	不支持
客户/Agent/服务器模型	中	中	高	不支持
客户/intercept/服务器模型	高	高	弱	支持
移动 P2P 模型	高	中	弱	不支持
基于移动代理的计算模型	高	高	高	支持

在数据通信方面,采用“广播”和“缓存”技术是解决移动计算难题的有效方法。缓存技术可以缓解数据访问压力、减少网络通讯流量、缩短用户访问延迟,已在传统分布式系统中得到大范围的成功应用。语义缓存由查询描述和查询结果组成,相对页缓存和元组缓存,语义缓存能支持关系型数据库上

到稿日期:2013-04-10 返修日期:2013-08-25 本文受国家“973”计划项目(2007CB07100, 2007CB07106), 国家自然科学基金(61171141)资助。  
梁茹冰(1980—),女,博士,讲师,主要研究方向为移动数据处理、语义缓存、智能代理技术, E-mail: liang\_ru\_bing@scau.edu.cn; 刘琼(1959—),女,博士,教授,主要研究方向为计算机网络应用、移动计算。

的查询<sup>[2]</sup>。利用缓存,用户下次访问相同信息时可避免重新访问网络,从而提高了用户的数据访问性能,节省了带宽资源,提高了数据可用性。然而,为了维护缓存数据的一致性而产生的验证消息也是导致通信开销和操作延迟的重要因素。如果控制得不好,一致性验证的开销可能会抵消缓存带来的好处。

缓存一致性验证算法可以分为客户驱动和服务器驱动两类。Howard等<sup>[3]</sup>提出的回调算法是一种由服务器驱动的缓存管理算法,服务器在修改数据前向每个缓存该数据的客户发送失效通知,客户收到失效通知后将数据从缓存中删除,然后再向服务器发送失效确认通知。Cao等<sup>[4]</sup>证明了回调算法的最大优点是可以用最少的通信开销来维护数据的强一致性。但其缺点也是很明显的,比如服务器需要等待收到所有客户失效确认后才能进行写操作,从而产生了“写延迟”;在移动环境中,由于客户的断连性,重新连接时需要重新验证缓存中的数据的有效性。李冬等<sup>[5]</sup>分析了语义缓存段与更新语句的条件谓词及投影属性的关系,将更新粒度裁剪至属性级,有效减少了通信开销。针对当前存在的错误失效判断问题,Haidar等<sup>[6]</sup>提出基于无状态广播失效报告方式的SAS(Selective Adaptive Sorted)缓存失效策略,即通过对失效报告中的数据按修改时戳降序排序来减少终端查询延迟并克服错误失效判断问题。Bo Yang等<sup>[7]</sup>对相似查询进行了研究,用语义缓存来反映网络中的数据内容分布,通过分析以前缓存的查询结果,试图用部分缓存内容相似的结点来处理新提交的查询请求。Po-Jen Chuang等<sup>[8]</sup>研究并提出应用在MANET环境中的两种缓存失效方案:ADD(Adaptive Data Dividing)和DVD(Data Validity Defining)。ADD根据利用率对数据进行分组,不同组采用不同的广播时间间隔,从而达到减少数据访问时间和带宽消耗的目的。DVD主要解决终端与服务器断接后,再次重连时的缓存数据有效性判断问题。Mershad等<sup>[9]</sup>提出一种适用于MANET的缓存一致性维护方案SSUM。

在代理技术应用方面,Sergio Harri等<sup>[10]</sup>提出一种利用Agent来最优化无线通信代价的算法。其通过生成监视代理、跟踪代理和更新代理,进行位置相关查询的提交、执行,并能够较好地支持移动性。胡海洋等<sup>[11]</sup>提出一种基于代理的无线Web访问机制MAWA,该机制将代理作为移动设备与Web服务器之间交互的中介,有效避免了移动设备不必要的Web页面更新提交操作,从而节省了无线网络连接开销,同时该机制还可有效支持移动设备进行无线Web访问时在基站间的切换,及故障后的状态恢复,从而确保了移动设备运行过程的可靠性。Steven K. C. Lo<sup>[12]</sup>还将合作多代理机制应用于智能交通系统,即通过分布式通信,共享驾驶信息,获得驾驶建议,有效改善交通情况。

上述研究成果为研究语义缓存一致性维护方法提供了理论基础。但它们多是基于传统客户方、服务方两层结构,不利于缓存的统一管理,一致性维护开销较大,难以达到强一致性,并且没有与代理技术相结合。本文提出一种基于代理的移动计算方案,即利用代理技术管理和执行缓存一致性验证

方法来解决回调算法在进行语义缓存一致性维护时存在的“写延迟”问题,有效减少了“写延迟”时间,并能够较好地支持断连操作。本文从客户端的数据访问操作和服务端端的写操作两方面讨论如何进行缓存一致性验证;最后通过实验验证了所提方法的有效性。

## 2 基于 Agent 技术的语义缓存一致性维护

### 2.1 结构及代理的功能设计

基本的移动计算网络环境中,移动终端(MT: Mobile Terminal)无线接入到移动支持站点(MSS: Mobile Support Station),MSS是支持移动计算的固定站点,具有无线通信接口,再由各MSS接入到有线的高速骨干网络中。这里MSS起到了连接移动客户端和服务器的网关作用,从MSS发送的信号可以覆盖的范围称为一个小区(Cell)。

移动计算系统(蜂窝通信系统)结构如图1所示,下图是对上图的一个抽象表示。值得说明的是,MSS与基站(BS: Base Station)的概念较容易混淆,当前很多文献中把BS和MSS认为是一个概念,其实两者有区别。笔者认为,BS是安装在固定位置的无线收发器,是无线通信网络的一部分,BS可以连接其覆盖区域的任何蜂窝电话并将电话转接到有线网络;而MSS具有无线通信接口,是支持移动计算的固定站点。BS只对应一个小区,而MSS可以管理一个或多个小区中的通信业务。可以把MSS认为是无线网络与有线网络之间通信的一个网关,而BS是移动终端与MSS之间连接的桥梁。

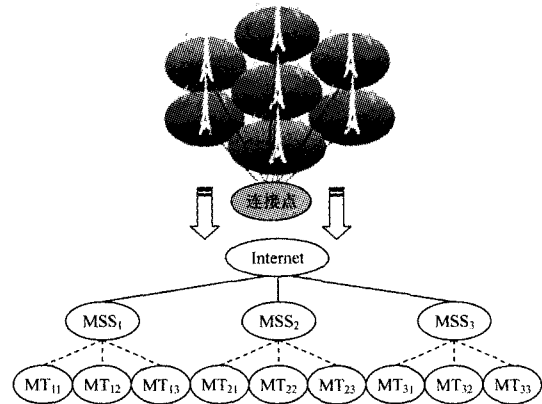


图1 移动计算系统结构

可见,BS的功能较MSS相对简单,MSS作为无线到有线网络之间的一个网关,是一个固定站点,上行与高速有线网络连接,下行与相应的BS连接,本身有一定的数据处理能力和存储能力。因此,可以在MSS中运行一些管理软件与算法。例如,基于代理的移动计算平台中,就要把Agent放置在MSS中,而不能放置在BS中;设计实现的缓存维护策略可以封装在Agent中,作为代理的功能。

然而,在当前移动计算模型中,MSS功能相对简单,网络负载和算法执行主要封装在MT和Server中,给服务器和终端带来了复杂的控制。基于原有网络基础设施,本文设计Local Server用于管理所在小区中的移动终端,即“Client/MSS/Server”结构,如图2所示。

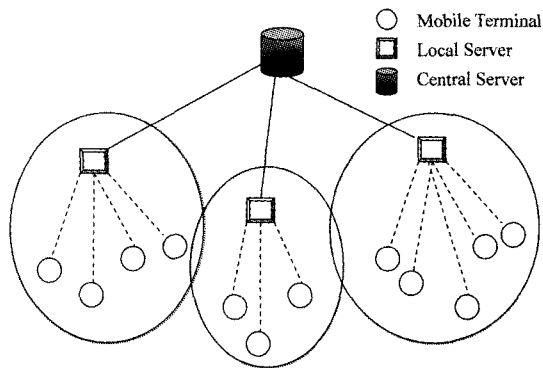


图2 Client/MSS/Server结构

在 MT、Local Server 和 Central Server 各层的功能模块中,设计代理进行缓存管理。MT 层代理有 Service Agent 和 Monitor Agent; Local Server 层和 Central Server 层代理有 Process Agent 和 Monitor Agent。各层的 Agent 及主要功能描述如表 2 所列。

表 2 代理功能描述

层名	Mobile Terminal	Local Server (MSS)	Central Server
功能描述	GPS 定位; 浏览器; 客户端缓存; 提交查询请求。	提供 MT 接入, 接收服务器的广播数据,管理语义缓存备份; 缓存备份;所在小区中的各 MT 的缓存。	广播“热点”数据; 数据库查询操作。
代理功能设计	Service Agent: 提交查询请求,接收来自 MSS 或 Server 的请求信息。 Monitor Agent: 接收来自 MSS 的广播数据并且更新缓存。	Process Agent: 处理来自移动客户端的查询请求。 Monitor Agent: 接收服务器的广播数据并且更新缓存;向客户端广播更新数据。	Monitor Agent: 对“热点”数据进行周期性的广播,由 MSS 接收该广播数据。 Process Agent: 接收来自 MSS 的查询请求并将结果返回给 Process Agent。

## 2.2 语义缓存一致性验证策略

由于移动终端经常有断连操作和位置变化,固定网络和分布式系统下的算法已不再适用于移动计算环境。传统回调查证算法存在写延迟、客户端连接时需要重新验证缓存数据等问题。本文提出基于代理的回调查证策略,即利用代理层管理终端缓存和转发失效数据,协调移动终端与固定服务端的关系,这既不影响客户端断连的需要,也不致产生服务器的写延迟,并能够维持数据的强一致性。

语义缓存数据结构的表示方法采用谓词编码方案<sup>[13]</sup>。这种 0、1 和 \* 表示的缓存数据能够有效地减少网络数据传输开销,并可以利用差(−)运算和交(∧)运算高效地判断需要更新的缓存数据,即文献<sup>[13]</sup>中所提到的剩余谓词(Remainder predicate)。加入代理技术后,缓存一致性验证过程如图 3 所示。

下面从两个方面说明移动终端语义缓存一致性验证方法。

(1)移动终端的数据访问操作引起的缓存一致性验证方法

当 Client 1 提交数据访问请求(步骤①和①′)时,由 Client 1 中的 Service agent 将该请求以缓存验证消息的形式发送给 Client 1 所在小区的 Local Server 1; Process agent 接收该消息并验证该缓存数据的时间戳,如果仍是 TS1,则返回命中消息(步骤②′),Client 1 直接访问本地缓存;如果时间戳是 TS2 或 TS3,表示客户端的数据已经无效,此时需要进行一致性维护(步骤③),即将 Local Server 1 中的相应缓存数据返回(步骤②),更新 Client 1 缓存。如果客户端没有缓存该数据,则直接向 Central Server 请求该数据访问服务。

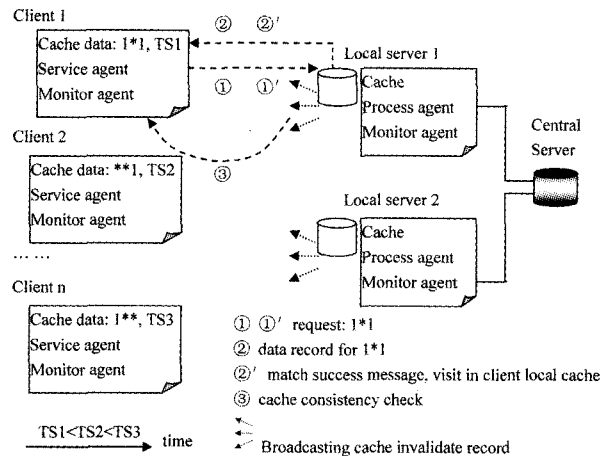


图3 使用代理的缓存一致性验证过程示意图

### (2)服务器端的写操作引起的缓存一致性验证方法

当服务器执行写操作时,传统回调算法存在“写延迟”问题,主要原因是需要等待每个缓存了该数据的客户端的确认后才能修改,而移动环境下,客户端由于带宽限制和频繁断连加剧了“写延迟”。基于代理的缓存一致性验证方法能有效地缓解这种延迟,具体过程如下:

服务器在修改数据前,修改通知不再发送给移动终端,而是发送给缓存了该数据的 Local Server,由其向服务器发送确认。由于这一段的连接是有线的,因此可靠性高、延迟小。服务器修改完数据后,再将更新的数据及时间戳发送给 Local Server。也就是说,服务器写操作由 Central Server 与 Local Server 交互完成,更新的数据和时间戳保存在 Local Server 的缓存中,此过程不需要终端参与,能够适应移动环境的需求,有效地避免了回调算法的写延迟和终端断接重连时重新验证缓存数据的问题。

缓存技术能减少网络拥塞、查询延迟和电能消耗,从而有效提高了网络性能和数据可用性。然而使用缓存最大的挑战就是维护缓存数据与服务器中相应数据的一致性(同步问题)。改进后的代理回调查证方法,以“Client/MSS/Server”3层结构为前提,MSS(即 Local Server)中为终端提供缓存备份并协助客户方进行缓存一致性维护。MSS 与服务器是始终保持连接状态的,服务器的数据变更可以及时发送给 MSS,因此 MSS 中的各语义缓存项在状态上可以与服务器数据保持强一致性,从而保证数据实时、有效。利用文献<sup>[14]</sup>中的方法,MSS 还可以把缓存数据更新操作细化为更新序列,当终端需要恢复缓存有效性时,只需访问 MSS 中的更新序列并执行更新操作即可达到本地缓存与服务器数据同步。利用这一中间层,既避免了写延迟,又简化了维护过程,减少了网

络通信量,并能够适应频繁断接情况下的一致性维护。

### 3 实验及性能分析

#### 3.1 参数设置及实验平台

为对所提方法性能进行数据分析,我们编程实现了文中算法和查询模拟器,并设计了几组实验。表 3 列出了实验中所用的主要参数。编程使用 J2SE 平台的 JDK 1.6 开发包,运行环境是双核 Pentium P6100 2.00GHz 处理器,内存 2GB,操作系统 Windows XP Professional。

表 3 实验参数

参数名	取值	含义
BInterval	60(s)	服务器广播时间间隔
BIntervalMSS	60(s)	MSS 广播失效报告间隔
UpInterval	10(s)	服务器数据平均更新时间间隔
CacheSize	1024(kB)	移动终端缓存空间大小
Data	512(B)	申请访问的数据量
CacheCheck	128(B)	验证缓存项有效与否的请求大小
ProbOffLine	0.1	移动终端与网络断接几率
OffTime	50~600(s)	移动终端与网络断接时长
WBandwidth	256(kB·s <sup>-1</sup> )	无线网络带宽
Bandwidth	2(MB·s <sup>-1</sup> )	有线网络带宽

算法模拟实验执行的平台包括的模块有:数据库生成器、查询产生器、语义缓存生成器和算法模拟器。各模块及其关系如图 4 所示。

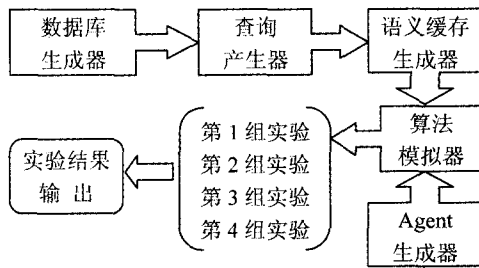


图 4 实验平台中的模块及关系图

**数据库生成器:**基本功能是产生关系数据库中各关系表的内容,并对各关系表中的属性分配选择概率、数据热区范围等进行定义。

**查询产生器:**基本功能是在关系数据库中,根据热点属性分布、元组数据热区、投影属性选择概率和查询语义相关性等参数随机产生 500 个可缓存查询。

**语义缓存生成器:**基本功能是执行随机产生的 500 个查询,利用查询结果形成终端的语义缓存。由于缓存空间大小是有限的,构建缓存过程中还涉及与已有的相关缓存项合并和缓存替换过程。

**算法模拟器:**基本功能是根据不同的实验场景,设置实验参数,对文中算法性能进行数据分析,例如计算查询响应时间、数据通信开销、缓存一致性恢复时间等。

**Agent 生成器:**将表 2 中列出的功能封装在类中,利用 java 语言编程实现,通过实例化类的对象来模拟一个具体的 Agent。在移动终端上生成 Service Agent 和 Monitor Agent;在 MSS 和服务端上生成 Process Agent 和 Monitor Agent。

#### 3.2 实验结果分析

首先,比较文中提出的缓存一致性验证算法(记为“Agent 方法”)与回调验证算法(记为“Callback 方法”)在终端断接重

连后提交数据访问请求时的响应时间。横轴代表“终端断网时长”,纵轴代表“访问缓存数据的平均响应时间”。

如图 5 所示,移动终端离线时长从 30s 增加到 300s, MSS 广播失效报告的时间间隔为 60s。对于 Agent 方法来说,只要终端离线时长在广播周期内,都认为缓存数据有效,因此可以直接由本地缓存响应。当终端离线时长大于 60s,则需要对缓存数据进行有效性验证,该验证过程由 MSS 来完成,无需服务器参与,从而可节省有线部分的带宽。对于 Callback 算法,终端一旦离线,不论多短,再次连接时必须对缓存数据进行一致性验证,而验证时要访问远程服务器,因此无线和有线部分的网络带宽都有所消耗,时间耗费也多。可见,Agent 方法的查询响应时间明显短于 Callback 方法,是一种更加适合在移动环境中使用的方法。

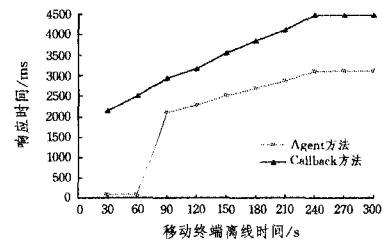


图 5 断连后的查询响应时间

其次,对 Agent 方法和 Callback 方法在“写延迟”方面的性能进行了比较和分析,相关比较结论如表 4 所列。

表 4 两种方法的“写延迟”性能

比较项	写延迟					
	终端通信开销	终端允许断连	终端允许移动	终端断连增加写延迟	终端移动增加写延迟	
Agent 方法	无	无通信开销	允许	允许	无影响	无影响
Callback 方法	有	通信开销大	不允许	允许	有很大影响	无影响

由于回调算法很大程度上都依赖于终端的操作,因此终端的能量消耗较大,同时移动环境中终端的频繁断连也会导致写延迟大幅增加。相比较而言,Agent 方法利用中间层(即 MSS)对缓存数据进行备份和一致性维护,既减轻了终端的负担(节省能量),又能达到数据强一致性,同时还支持终端频繁断接与移动。当终端再次与网络连接时,其缓存一致性恢复可通过访问 MSS 达到,具体方法参见文献[14]中介绍。

第三,比较 Agent 方法与传统方法(记为 Tradition)和文献[5]中提出的两层结构粒度细化算法(后面称两层算法,记为 Two-Tier)在网络通信量方面的性能。

模拟依次对缓存中 50~400 个待更新的元组进行一致性维护,3 种方法所产生的网络通信量如图 6 所示。当客户方始终与网络保持连接时,可以持续接收失效报告进行缓存一致性维护。此时,文中算法与两层算法的通信开销接近,它们较传统方法有更低的网络通信量,且增长幅度也较小。这是因为传统方法对待 UPDATE 更新操作,通常是先在缓存中执行 DELETE 操作,再执行 SELETE 操作以获取更新后的元组,再将它们插入到缓存中。因此,有时只需要修改个别属性值,却变成了整个元组的操作,从而增加了通信量。发生断接操作时,传统算法与两层算法在断接时间小于广播时间间

隔情况下缓存恢复较易;一旦断接时间大于广播时间间隔,再次连接就无法根据当前失效报告来判断缓存的有效性,两层算法对此则没有研究。

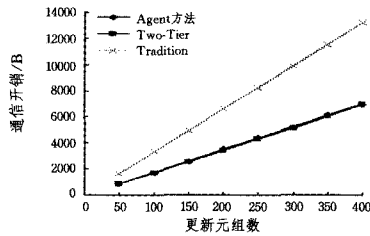


图6 3种方法的数据通信开销

最后,比较基于代理的缓存验证方法与一般基于 Agent 的方法(记为 T-Agent)以及不采用代理的方法(记为 Without-Agent),这3种方法在断接时间变化的情况下再次连接时将缓存恢复有效而产生的时间开销如图7所示。这里,广播失效报告周期是60s,当断接时间小于60s时,缓存维护耗时都很短(Agent方法:4.2ms, T-Agent:14.5ms, Without-Agent:0ms)。当时间大于60s时,Without-Agent策略无法再按照失效报告来维护缓存,因此会认为其缓存全部无效,从而需要重新验证或删除,这无疑抵消了缓存带来的好处。为画图比较,我们给其设置了一个相对较大的数1000来表示,其实1000并不是真正的恢复时间(也许 $+\infty$ 更合适)。另外,Agent方法利用MSS助其维护更新队列,即使在客户方离线情况下,缓存一致状态也可以得到有效维护。客户方再次连接上网络时,只需执行出队列操作,并依次执行队列中的更新操作即可将其缓存恢复有效。而一般基于Agent的算法思想上也是用代理替用户进行缓存管理,但是处理时是基于全属性的,且没有进行更新裁剪,这会导致一致性维护时数据传输开销大的问题。

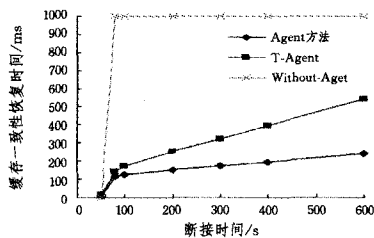


图7 3种方法的缓存一致性恢复时间

**结束语** 本文首先给出 Client/MSS/Server 移动计算架构,其通过代理管理客户端缓存和转发失效数据,来协调移动客户端与固定服务器端的关系,并对回调验证策略进行了改进,既不影响客户端断连的需要,也不致产生服务器的写延迟,且能保持缓存数据的强一致性。所提方法的主要特点归纳如下:

(1)对传统的 Client/Server 两层移动计算平台功能进行扩展,引入代理技术,并利用了 MSS,提出了 Client/MSS/Server 移动计算结构。在 MT, MSS 和 Server 各层设计代理,通过其进行缓存数据访问和响应服务器修改数据消息。

(2)在客户端数据访问服务方面,保持了回调算法低通信开销和数据强一致性的优点。缓存数据结构采用谓词形式,

能够适应与内容相关的缓存管理算法。

(3)代理响应服务器发送的数据修改通知,比较适用于客户端频繁断连的移动计算环境,并使得回调算法的“写延迟”问题得以解决。

实验结果表明,基于代理技术、利用 MSS 协助进行缓存备份和一致性维护的方法,其断接重连的查询响应时间更快,缓存更容易恢复一致,并且能够克服回调算法在写延迟方面的不足,从而更加适用于终端频繁移动、断接的无线网络计算环境。下一步我们将研究移动多代理技术,以适应终端频繁越区移动情况下语义缓存一致性维护的需要。

## 参考文献

- [1] Lange D B, Oshima M. Seven good reasons for mobile agents [J]. Communications of the ACM, 1999, 42(3): 88-91
- [2] Dar S, Franklin M, Jonsson B, et al. Semantic data caching and replacement [C] // Proc the 22nd VLDB Conference, Mumbai (Bombay), India, 1996; 330-341
- [3] Howard J, Kazar M, Menees S, et al. Scale and Performance in a Distributed File System [J]. ACM Transaction on Computer Systems, 1988, 6(1): 51-81
- [4] Cao P, Liu C. Maintaining strong cache consistency in the World Wide Web [J]. IEEE Transactions on Computers, 1998, 47(4): 445-457
- [5] 李东,袁应化,叶友,等.基于属性更新的语义缓存一致性维护算法 [J]. 华南理工大学学报:自然科学版, 2009, 37(5): 139-144
- [6] Safa H, Artaïl H, Nahhas M. A cache invalidation strategy for mobile networks [J]. Journal of Network and Computer Applications, 2010, 33: 168-182
- [7] Yang Bo, Mareboyana M. Similarity search in sensor networks using semantic-based caching [J]. Journal of Network and Compute Applications, 2012, 35: 577-583
- [8] Chuang Po-Jen, Chiu Yu-Shian. Efficient cache invalidation schemes for mobile data accesses [J]. Information Sciences, 2011, 181: 5084-5101
- [9] Mershad K, Artaïl H. SSUM: Smart Server Update Mechanism for Maintaining Cache Consistency in Mobile Environments [J]. IEEE Transactions on Mobile Computing, 2010, 9(6): 778-795
- [10] Harri S, Mena E, Illarramendi A. Monitoring Continuous Location Queries Using Mobile Agents [C] // Manolopoulos Y, Návrat P, eds. ADBIS 2002. LNCS 2435: 92-105
- [11] 胡海洋,胡华,凌云. MAWA: 基于移动 Agent 的一种高效可靠无线 Web 访问机制 [J]. 计算机研究与发展, 2008, 45(4): 636-645
- [12] Steven K, Lo C. A collaborative multi-agent message transmission mechanism in intelligent transportation system-A smart freeway example [J]. Information Sciences, 2012, 184: 246-265
- [13] Chung Y D. A cache invalidation scheme for continuous partial match queries in mobile computing environments [J]. Distrib Parallel Databases, 2008, 23: 207-234
- [14] 梁茹冰,刘琼.一种使用 MSS 维护语义缓存一致性的方法 [J]. 华南理工大学学报:自然科学版, 2011, 39(7): 127-133