移动计算环境中缓存失效策略的归类研究法*)

吴 劲 卢显良 任立勇 周 旭
(电子科技大学计算机科学与工程学院 成都610054)

摘 要 缓存失效报告法是在移动计算环境中普遍采用的缓存方案,即服务器定期广播缓存失效报告,该报告中包含了最近被更新的对象,根据报告内容,客户可使缓存中被更新对象失效,以保证缓存的有效性。缓存失效报告技术可采用多种策略,本文在研究多种缓存策略的基础上,提出了一种针对缓存失效方案的归类研究法,可对具体策略加以分析和评估,在此基础上对其性能进行改良,以形成更适应特定移动计算环境要求的缓存方案。

关键词 缓存失效,移动计算环境,移动主机,归类研究

A Categorical Research Method for Cache Invalidation Strategies in Mobile Computing Environments

WU Jing LU Xian-Liang REN Li-Yong ZHOU Xu
(College of Computer Science & Engineering, UESTC of China, ChengDu 610054)

Abstract Cache invalidation schemes are an attractive approach for mobile computing environment. To affirm the validity of mobile host' cache content, servers periodically broadcast cache invalidation reports that contain information of data that has been updated. In this paper, we reexamine the issue of designing cache invalidation strategies and propose a categorical research method for cache schemes. Based on this method, many improving schemes can be constructed.

Keywords Cache invalidation, Mobile computing environments, Mobile host, Categorical research

1. 引言

随着移动网络技术的发展,移动用户能不受时空限制地访问定位在静态网络上的服务器,但该项技术的推广受到无线带宽和移动计算机电源能力的限制。通过在移动主机上缓存经常被访问的数据,能有效减少带宽需求,并能节省移动计算机的能耗。但移动主机具有移动性和频繁断接性,使传统的缓存策略无法适应移动计算环境的需求,必须采用新的缓存策略。

Imielinski 和 Barbara 等人,提出了一种新的缓存技术,称为缓存失效报告广播技术:即服务器定期或异步地广播缓存失效报告,该报告中包含了最近被更新的对象,根据报告内容,客户可使缓存中被更新对象失效,以保证缓存的有效性。本文在研究多种缓存策略的基础上,提出了一种缓存失效方案的归类研究法,可对具体策略加以分析和评估,在此基础上对其性能进行改良,以形成更适应特定移动计算环境要求的缓存方案。

2 缓存失效方案的归类研究法

2.1 缓存失效法的分类

在传统的客户/服务器系统中,可有两种方法维护客户缓存的有效性:

1. 服务器发送缓存失效报告给客户机。服务器了解客户机的缓存状态,即客户机缓存了哪些数据对象,当某个数据对象被更新时,服务器直接给缓存该对象的客户机发送一个失效(或更新)信息。

2. 客户机通过查询服务器来验证其缓存的有效性。

这两种方法都不适合应用于移动计算环境。若采用方法1,由于移动主机经常断连,断连期间客户机无法接收缓存失效信息,因此重新连接后,无法判断哪些缓存内容有效,只能令整个缓存失效,失去缓存应有的作用;若采用方法2,虽可避免断连引起的问题,但又会严重浪费有限的网络带宽,达不到节省网络带宽的目的。

因此,必须研究新的缓存技术,适应移动计算环境的需求,而缓存失效报告广播技术可以较好地解决这些问题。在参考文[2]中提出了两种基本的缓存方案:

1. "基于状态的(stateful-based)",服务器了解移动客户机的缓存内容,当数据库被更新时,服务器发送失效报告给相应的客户。

2. "无状态的(stateless-based)",服务器无需了解移动客户机的缓存状态,服务器广播被更新的对象,客户机收听报告,根据报告内容更新缓存。

由于基于状态的方法太过复杂,服务器必须定位客户机,一旦客户机开始移动,还必需把重定位的信息发给服务器,因此很少采用这种方式。

而无状态的缓存失效策略又分为两种方式:

1. 异步方式。在这种情况下,一旦记录被更新,服务器立即广播更新值。异步方式对于连接客户非常有效,可以及时更新缓存;但对于断连一段时间后重新连通的客户,就无法判断缓存的有效性。为了挽救缓存,Barbara 和 Imielinski 提出失效报告可以捎带以前一段时间(T)内所有的失效通知,若断连时间大于 T,则令整个缓存失效;若小于 T 则可根据报告

*)本文由国防跨行业预研基金和电子科技大学校青年基金资助,项目编号分别为:51406070201DZ0211和 YF020803.吴 劲 讲师,博士生,主要研究方向为计算机网络及分布式数据库技术;卢显良 教授,博士生导师,主要研究方向为计算机网络技术及应用。

更新缓存。但若客户断连一段时间再重连后，就不能保证客户需要等多久才能等到下一个异步报告。

2. 同步方式。在这种方式下，服务器跟踪最近被更新的对象，定期向客户广播失效报告。根据报告，客户可决定缓存是否有效，若有效则用缓存内容回答查询；若无效，则把查询提交给服务器。因为缓存报告定期广播，它提供了客户等待缓存报告时间的上限。

通过分析比较发现同步方式能更好地适应移动计算环境的特殊要求，因此，本文将重点研究无状态的同步缓存报告法。

2.2 技术指标分析

分析缓存失效策略可从以下几个方面入手：

2.2.1 *颗粒度* 是指报告中每条记录信息的详细程度。报告中的记录可表示为(id,TS)，id 为更新对象的标识符，TS 是更新时戳；也可表示为(object,TS)，object 是指实际对象本身。前一种方式称为"失效更新(update invalidation)"，客户删除缓存中失效的内容；后一种方式称为"广播更新(update propagation)"，客户可根据报告内容更新其缓存。在失效更新方式下，客户需提交请求，检索被更新记录；而广播更新方式可使客户立即更新其缓存内容，但其更新报告较大要占用较多的下行带宽，会牺牲宝贵的网络带宽。

和这两种方式相对，还可以有两种方式，即报告以记录清单(list of ids,TS)或对象清单的形式组织(list of objects,TS)。每个(list of ids/objects,TS)表示在时戳 TS 后被更新的 ids 或 objects 的清单。例如：对象 O_1、O_2和 O_3在 TS_1、TS_2和 TS_3时刻被更新，且 $TS_1<TS_2<TS_3$，在报告中的记录为：(O_1,O_2,O_3,TS_1)。这种方式可减少报告的大小，但也会带来一定的问题。假设，一个客户在 TS_d 时刻访问这三个对象，$TS_2<TS_d<TS_3$，而且在 TSr 时刻重新连接，$TS_r>TS_3$，虽然 O_1、O_2仍然有效，实际上它们已经失效了。这个问题在文[5]中有详细的描述。

还有一种减少失效报告大小的方式，就是以组的方式广播信息，即数据库对象以组的形式组织，报告中的记录为(group-id,TS)，它表示在 TS 时刻一组对象已经被更新。与清单组织方式类似，这种方式也存在错误失效的问题，我们在下面的例1中会说明组的错误失效问题。文[6]对基于组的方法有详细的描述。

2.2.2 *报告大小和期限* 失效报告的大小可以是固定或变化的，更新期限的长短也可以是固定或变化的。如果报告大小是固定的，那更新期限的长短无法确定，即是可变的；若更新的期限固定，则报告大小是可变的。例如：在时刻 T 广播一个更新报告，该报告反映在[T-ωL,T]时间间隔内的所有更新，其中 ω 表示广播窗口的大小，L 表示广播报告的时间间隔。固定失效报告大小的缓存方式在文[5,9]中有相应的描述，固定更新历史长度的缓存方式在文[1,6]中也有相应的描述。

2.2.3 *失效机制* 当一个客户收到失效报告，可以有两种方式更新其缓存内容：缓存级失效法和查询级失效法。缓存级失效法是指对缓存中的所有对象进行有效性验证，这种方法需要扫描整个或大部分失效报告，它的缓存内容的时戳与已经处理的最近失效报告的时戳一致。而查询级更新法，是当有查询发生时，才进行缓存有效性验证，所以每个缓存对象都需有相应的时戳，该时戳表示该对象最后被验证有效的时刻。文[9]采用了查询级更新法，而文[2,5,6]都采用了缓存级更新法。

3 具体策略分析

在这一小节，将用上一小节的方法分析两种具体的缓存失效策略：一种称为"双报告缓存失效法"(Dual-Report Cache Invalidation，DRCI)；另一种称为"位序列法"(Bit-Sequences，BS)。为了说明这两种策略，我们使用一个有16个对象的数据库作为运行例子，数据对象的 ID 和相应的更新时戳如表1所示。

表1 例子数据库

ID	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
TS	30	22	16	12	28	24	32	42	8	26	20	36	14	10	18	34

3.1 双报告缓存失效法(DRCI)

双报告缓存失效法 DRCI 的技术指标如下所示：①颗粒度：报告采用了两种颗粒度，(id,TS)和(group-id,TS)；②失效机制：采用缓存级更新法；③报告大小和期限：报告的大小和期限都可变。报告包含[T-wL,T]期间内被更新的(id,TS)记录，还包含[T-WL,T]期间内被更新的(group-id,TS)记录，且0<w<W。

采用双报告缓存失效法，服务器每隔 L 时间单位，发送一双失效报告：对象失效报告(Object Invalidation Report，OIR)和组失效报告(Group Invalidation Report，GIR)。当移动主机收到报告，其更新方式如下：

步骤1 在[T-wL,T]期间内被更新的包含在 OIR 中的对象，使用 OIR 中的内容更新，忽略 GIR 中其 group-id 的时戳。

步骤2 对组中的没包含在 OIR 中其它对象，使用 GIR 中的内容进行更新。

DRCI 法可以使用两种失效处理方法：若客户只断连了很短的一段时间，可以使用 OIR 更新缓存；若断连时间在 T-wL 之前，要使用 OIR 和 GIR 两个报告一起更新缓存。OIR 可以补充 GIR 的不足，减少错误失效的发生；而 GIR 可以使经过较长时间断连后再重连，也能验证其缓存的有效性。其具体算法如下：

```
//T is timestamp of current report
//T* is timestamp of last valid report by mobile host
for every pair[Oid,tid]∈OIR {
    if(Oidis in the cache){
        if(T*<tid)
            remove object from the cache; }
}
if(T-T*>wL){
    for every group in the client cache{
        check the pair[Gid,Tid] in the GIR;
        if(T*<Tid)
            remove all object in group from the cache;
}
for every object O∈Database{
    if(O is in the cache)
        use the cache's content to answer the query;
    else
        submit request to server;
}
T*=T;
```

下面我们以表1的数据库为例，说明 DRCI 的应用。

例1 假设 T=44，L=4，w=2，W=6；对象的分组情况如下：

$G_1=\{O_1,O_2,O_3,O_4\}$，$G_2=\{O_5,O_6,O_7,O_8\}$，

$G_3=\{O_9,O_{10},O_{11},O_{12}\}$，$G_4=\{O_{13},O_{14},O_{15},O_{16}\}$。

在 T 时刻，移动主机收到了失效报告，其内容如下：

·OIR 报告:{(O_7,32), (O_8,42), (O_{12},36), (O_{16},34), 44}

·GIR 报告:{(G_1,30), (G_2,28), (G_3,26), (G_4,18),44}

假设移动主机最后一次收到报告的时戳为28,在时刻29断连,在时刻44重新连,若要查询对象 O_1,O_3,O_5,O_8,O_9,O_{13},O_{15}。

首先使用 OIR,删除缓存中的 O_7,O_8,O_{12},O_{16}四个对象;组中剩下的对象为 $G_1=\{O_1,O_2,O_3,O_4\}$,$G_2=\{O_5,O_6\}$,$G_3=\{O_9,O_{10},O_{11}\}$,$G_4=\{O_{13},O_{14},O_{15}\}$,再使用 GIR,可知 G_2,G_3,G_4是在断连时刻29之前更新的,所以仍旧有效,而整个 G_1被删除(我们可以看出 O_2,O_3,O_4被错误失效)。

因此对象 O_5,O_9,O_{13},O_{15}可以在缓存中提取,而对象 O_1,O_3,O_8需向服务器提出请求。

3.2 位序列法(BS)

位序列法 BS 的技术指标如下所示:①颗粒度:报告记录的形式为(list of ids,TS);②失效机制:采用缓存级更新法;③报告大小和期限:报告的大小固定,但期限可变。

采用位序列法,服务器广播位序列报告,即报告中的"位"对应数据库中的数据对象,而"位"的位置表示数据对象的索引。如大小为 N 的序列的第 n 位表示,数据对象 On。在失效报告中,每个序列和唯一的时戳相关,"位"的值为"1"表示在时戳时刻之后该位代表的数据对象被更新了;"位"的值为"0"表示在时戳时刻之后该位代表的数据对象都没被更新。

为了减小报告的大小,可采用一种"更新聚簇"的技术。假设数据库中数据对象的个数为 $N=2^n$,在 BS 算法中,报告可反映 n 个不同时刻的更新序列,T_n,T_{n-1},…,T_i,…,T_1,且 $T_i<T_{i-1}$,$1<i\leqslant n$。对于序列 B_n,当置为"1"的位达到一半时,形成(B_n,T_n)序列;下一个序列 B_{n-1}只有 N/2位,只包含那些被置为"1"的位,在 B_{n-1}中的第 k 位,对应在 B_n 中的第 k 个为"1"的位;对于其它的 i,$0\leqslant i\leqslant n-1$,$B_{n-i}$是有 $N/2^i$ 位的序列,它表示在 T_{n-i}时刻有 $N/2^{i+1}$个对象被更新;序列 B_{n-i}的第 k 位对应于序列 B_{n-i+1}中第 k 个"1"所在的位;以此类推直到形成(B_1,T_1)序列为止。还可以有一个附加的时戳为 T_0的空序列 B_0,表示从时刻 T_0开始没有对象被更新。其算法如下:

```
//T is timestamp of current report
// T* is timestamp of last valid report by mobile host
if(T0≤T*)
    all objects are valid in the cache;
else{
    if(T*<Tn)
        remove all object in the cache;
    else{
        using the bit sequence Bi such that Ti≤T*≤Ti-1,1≤i≤n
        invalidate all the objects marked "1" in Bi
    }
}
for every object O∈Database{
    if(O is in the cache)
        use the cache's content to answer the query;
    else
        submit request to server;
}
T*=T;
```

例2 下面我们以表1给出的例子数据库的数据为依据,形成一个 BS 报告,其结构如图1所示。

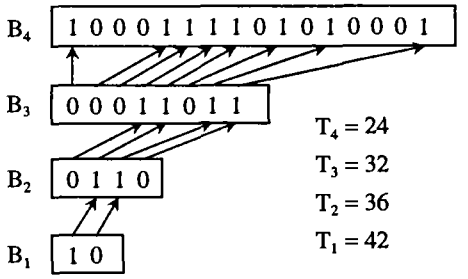

图1 位序列结构

假设客户最后一次收到失效报告的时刻是37,因为$36<37<42$,所以使用 B_2来更新缓存,为了定位为"1"的位,必须检测 B_2到 B_4序列,其分析如下:B_2的第1个"1"是第2位;查找 B_3的第2个"1",处于第5位;查找 B_4的第5个"1",处于第8位;由此可知 O_8对象失效;同理可知 O_{12}对象失效。

总结 本文提出了一种研究缓存失效策略的分类研究法,该方法主要能对各种策略进行定性分析,第3节分析的两种策略是目前大多数缓存失效方案的基础,许多改良算法都是对第2节所提的技术指标进行改进而形成的。我们下一步的工作是在分类研究法的基础上,以访问时间、带宽大小和能量消耗等为具体性能指标,对移动计算环境中的缓存方案进行定量分析。

参 考 文 献

1 Barbara D, Imielinski T. Sleepers and Workaholics: Caching Strategies in Mobile Distributed Environments. In: Proc. 1994 ACM-SIGMOD Int'l Conf. Management of Data, 1994. 1～12

2 Barbara D, Imielinski T. Sleepers and Workaholics: Caching Strategies in Mobile Environments (Extended Version). MOBIDATA: An Interactive J. Mobile Computing, 1994,1(1)

3 Imielinski T, Badrinath B R. Mobile wireless computing: challenges in data management. Communication of ACM, 1994

4 Barbara D, Imielinski T. Sleeper and Workaholics: Caching Stragegies in Mobile Enviroments. Very Large Database J., Dec. 1995

5 Jing J, Elmagarmid A, Helal A, Alonso R. Bit-Sequences: An Adaptive Cache Invalidation Method in Mobile Client/Server Environments. Mobile Networks and Application, 1997,2(2)

6 Tan K L, Cai J. Broadcast-Based Group Invalidaion: An Energy Efficient Cache Invalidaion Schema. Information Sciences, 1997, 100

7 Liu G Y, McGuire Jr G Q. A Mobility-Aware Dynamic Database Caching Scheme for Wirless Mobile Computing and Communication. Distributed and Parallel Database, 1996,4:271～288

8 Sistla A P, Wolfson O, Huang Y. Minimization of Communication Cost Through Caching in Mobile Enviroments. In: Proc. ACM Special Interest Group on Management of Data, May 1994

9 Cai J, Tan K L. Energy-Efficient Selective Cache Invalidaion. Wireless Networks, 1999,5(6)