

文章编号:1001-9081(2010)07-1950-03

移动数据库中改进的 CMIP 数据预取策略

李靖,余建桥

(西南大学 计算机与信息科学学院,重庆 400715)

(queenchill00@163.com)

摘要:数据预取是移动数据库缓存技术中的关键,CMIP 预取策略通过客户端历史访问记录关联规则的挖掘得到预取数据,使系统性能得到了提高。但由于没考虑到数据的更新率及数据大小,将会经常发生缓存失效。在此算法的基础上增加对数据更新率及大小的判断并对所选数据排序,然后进行预取数据的选择。通过改进降低了缓存的失效效率、减少了数据访问的时间及电能的消耗。

关键词:缓存;数据预取;关联规则;更新率

中图分类号: TP311 **文献标志码:** A

Improved CMIP data prefetching strategy in mobile environments

LI Jing, YU Jian-qiao

(School of Computer and Information Science, Southwest University, Chongqing 400715, China)

Abstract: The data cache prefetching is a key technology in mobile environments. Cache-Miss-Initiated Prefetch (CMIP) data can improve the system performance by prefetching strategies to access the client record of the history to mining the association rules and get the prefetch data. However, since it does not take account of the data update rate and data size, cache invalidation often takes place. In this paper, this algorithm was based on the data update rate and it increased the size of the selected data to determine and sort, then made the choice of the prefetching data. Through improving, cache invalidation declines and the data access time and power consumption decrease.

Key words: caching; data prefetching; association rule; update rate

0 引言

移动数据库中数据广播技术^[1]及移动通信带宽的有限性引起较大的数据访问延迟;移动客户机与固定网络频繁(主动或被动)断接将使得事务得不到所需要的数据;移动设备的电能限制,通过无线网络访问远端服务器将消耗大量的电能。正是由于这些移动数据库的限制特点,缓存数据预取技术^[2]的使用能显著提高数据访问速度,减少数据访问时间,降低了电能的消耗。

CMIP^[3] (Cache-Miss-Initiated Prefetch) 预取策略是传统移动数据库中数据预取策略之一,它通过对客户端历史访问记录的关联规则挖掘动态地生成两个集合:总是预取数据集合和失效预取数据集合。对于总是预取集合里的数据只要出现在广播信道中都预取到客户端,对于失效预取数据集合中的数据,如果客户机需要访问的数据在本地找不到时,就要向服务器发送请求,这时不仅请求没有找到的数据,而且要请求预取与该数据相关的其他数据,这样大大提高了缓存的命中率。然而由于该预取策略没有考虑到数据的更新率及数据的大小,可能导致一个频繁更新的数据在客户端访问时经常失效,增加了上行链路请求、访问时间及电能的消耗。基于这两点,本文在该算法的基础上通过引入数据的更新率和数据大小对总是预取数据集合里的数据进行价值评估,然后根据价值大小对这些数据进行排序,预取价值最大的数据作为预取

数据集合,提出了改进的 CMIP 预取策略及其模型。

1 CMIP 预取模型及策略

1.1 CMIP 预取模型

CMIP 预取策略使用的是一个基于请求的广播模型。数据库服务器上的数据库由 N 个数据 $d_1, d_2, d_3, \dots, d_N$ 组成。服务器负责维护数据库和响应客户端的数据请求。当客户端需要访问的数据在缓存中找不到时,就通过上行链路向服务器发送请求,服务器接受到该请求之后就把这个数据通过广播信道广播出去。为了保证缓存数据和数据库数据的一致性,服务器每隔一段时间就广播失效报告 (Invalidation Report, IR),并且在每个广播周期中复制 M 次 IR,即更新失效报告 (Update Invalidation Report, UIR) 报告,该报告中只包括上次更新报告中更新的数据信息。客户端通过接受 IR/UIR 来判断缓存中的数据是否有效。具体的 CMIP 预取模型如图 1 所示。

从图 1 中可以看到,CMIP 预取策略中产生的预取数据来源于通过关联规则挖掘得到的两个数据集合:总是预取数据集合和失效预取数据集合。

1.2 CMIP 预取策略

CMIP 预取策略的核心^[3]是关联规则的挖掘。通过客户端的历史访问记录,利用数据挖掘技术中的关联规则挖掘,得出下一时刻客户事务最可能访问到的数据,然后把这些数据预取到缓存中。该预取策略将动态地构造两个数据集合:总是预取数据集合和失效预取数据集合。总是预取数据集合里包括这些经常访问的数据,只要出现在广播信道中就总是预

收稿日期:2010-01-28;修回日期:2010-03-05。

作者简介:李靖(1984-),女,重庆人,硕士研究生,主要研究方向:移动数据库;余建桥(1957-),男,重庆人,教授,博士,主要研究方向:数据库技术、人工智能。

取到缓存里。而对于失效预取数据集,当某个数据在缓存中不存在或者已经出现在 IR/UIR 报告中,则向服务器发送数据请求,此时不仅只请求该失效数据,同时也请求预取与该数据相关的其他数据。该关联规则的挖掘是基于客户端在某段时间内的访问记录。然而由于客户的访问兴趣总是集中在一段时间里的,所以,对于关联规则的挖掘在一段时间之后要重新挖掘,重新构造新的预取数据集,这样才能保证缓存中数据的有效性。

关联规则是由频繁项集^[5]产生的。频繁项集的产生采用 Aprior 算法^[5],分为连接步和剪枝步。 S 表示客户端的访问记录。输入该访问记录,得到频繁一项集。然后连接频繁一项集得到候选项集,通过剪枝函数,去掉支持度小于最小支持度的项集得到频繁二项集。依次循环下去,从频繁 $K-1$ 项集中产生频繁 K 项集。这里只产生两类关联规则:1) “ $\Rightarrow i_j$ ”; 2) “ $i_j \Rightarrow Y$ ”。 i_j 是一个数据项, Y 是一个数据项集。第一类关联规则用于产生总是预取数据集,第二类关联规则用于产生失效预取数据集。

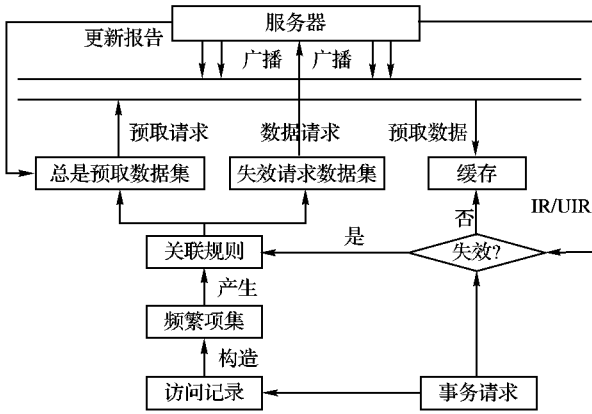


图1 CMIP 预取模型

2 改进的 CMIP 预取策略及模型

2.1 CMIP 预取策略分析

通过前面的描述,分析得到在 CMIP 预取模型中只考虑数据在客户端的访问概率,通过访问记录的数据挖掘得到的数据直接预取到缓存中而忽视了数据在服务器上的更新频率及数据大小。客户端事务每次访问数据时都会先比较 IR/UIR 报告从而判断该数据是否有效,那么如果一个缓存中的数据经常出现在 IR/UIR 报告中,将导致缓存经常失效,事务将需要重新向服务器发送数据请求,增加了上行链路带宽及访问时间,同时也加大了电能的消耗。因此这样经常更新的数据预取到客户端缓存价值不大。

另外,对于失效的数据,将直接向服务器发送与该失效数据相关的所有数据。这里通过一些关联规则得到的一部分数据已经被预取到缓存中,则很有可能通过另一些关联规则得到的与该失效数据相关的数据已经预取到缓存中,那么就没有必要再向服务器发送数据请求。针对上述两个缺点,对该模型作了改进,得到了改进的 CMIP 预取模型。

2.2 改进的 CMIP 预取模型

基于上述观点,对 CMIP 预取模型作了如下的改进。首先,通过关联规则产生两个集合:欲预取数据集和失效欲请求数据集。对于欲预取数据集中的数据进行价值评估(数据预取代价模型如图2所示),该价值评估包括数据的更新率^[4]和数据的价值。然后根据数据的价值排序,选择价值最大的数据进行预取。 $Value(i)$ 表示数据项 i 的预取价值; $Size(i)$ 表示数据项 i 的大小; $U(i)$ 表示数据项 i 的更新率,指在 K 个广

播周期内数据 i 更新的平均次数; $d(i)$ 表示从服务器检索数据项 i 的延迟^[3],即从发送数据请求到广播中出现该对象所需等待的时间。

$$U(i) = \sum_{i=1}^k f_i/k \quad (1)$$

$$value(i) = d(i)/(size(i) \times U(i)) = d(i)/\left(size(i) \times \sum_{i=1}^k f_i/k\right) \quad (2)$$

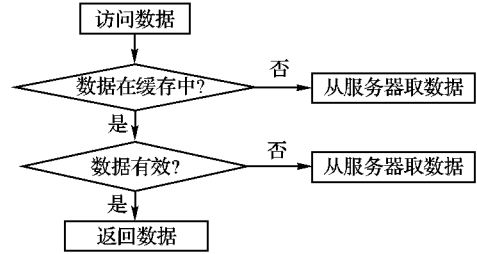


图2 数据预取代价模型

而对于失效欲请求数据集中的数据,优先判断该数据是否已经存在于缓存中,如果存在,则把该数据从集合中删除。改进的 CMIP 预取模型如图3所示。

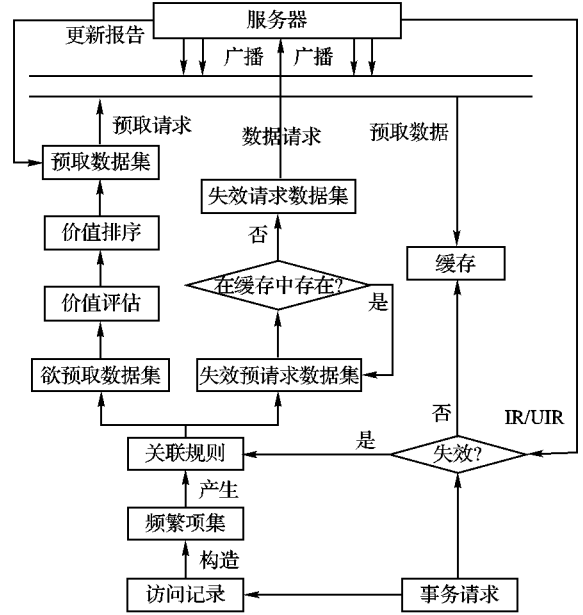


图3 改进的 CMIP 预取模型

2.3 改进的 CMIP 预取算法

由上面图3的预取模型,得到了如下两个预取算法:预取数据集算法和失效请求数据集算法。

1) 预取数据集算法:

```

prefetch( R, M )      // R 为关联规则集; M 为关联规则的数目
F = ∅; P = ∅; Lc = ∅;  // * F 为当前考虑预取的候选数据集; P 为预取数据集; c 为候选数据集 */
i = 0; N = 0;
While( M > 0 )
    If( R1 ∈ “=>fj.item1” ) // fj.itemm 为项集 fj 的第 m 个数据项
        F = F ∪ fj.item1;
        N++; M--;          // N 为 F 中的数据项个数
    End
While( N > 0 )
    Value(i) = d(Fi)/(size(Fi) * U(Fi)); // FK 为频繁 K 项集
    Sort( F );                          // 对 F 中的数据按价值进行排序;
    i++; N--;
End;
while( size(c) > 0 || i > 0 )          // size(c) 为缓存的空闲空间

```

```

if( size(c) > size(Fi) )
P = P ∪ Fi;
Lc = P;           //Lc 为缓存中存放的数据集
size(c) = size(c) - size(Fi);
i - -;
End;               //通过价值评估获得预取数据集 P
Return P;
2) 获得缓存失效请求数据集算法:
Invalidation_prefetch( item)
IP = ∅;             //IP 为失效请求数据集
i = 0;
While( M > 0)
If( Ri ∈ "ff. itemi => {ff - {ff. itemi}} " && item = ff. itemi)
// item 失效数据项
IP1 ← ff - {ff. itemi}; // IP1 为失效预请求数据集
If (ff. itemk ∈ Lc && k ≠ i)
// 判断该数据是否已经存在缓存中
IP = IP1 - ff. itemk;
M - -;
End;
Return IP;

```

3 性能分析

为了评价该缓存预取模型,参照文献[3,4,6-8]提出的一组实验数据,参见表1,并且和CMIP预取模型进行比较,通过对缓存命中率、缓存失效率及电源消耗三个方面进行分析。

在服务器端,为了保证缓存的一致性,服务器周期性地地向客户机广播IR和UIR信息。广播周期时间间隔 L 设置为20s,并且在每个时间间隔内UIR被复制4次。设服务器端有 N 个数据,每个数据大小 s_i 在 S_{\min} 到 S_{\max} 平均分布。在移动客户端,如果该客户机处于连接状态,将一直监听IR和UIR来维护缓存的一致性。

表1 实验关键参数值

参数	缺省值	参数	缺省值
数据服务器中数据个数	1 000	广播窗口大小(w)	10 interval
移动客户机个数	200	UIR 复制次数	4
广播间隔(L)	20 s	最小支持度	60%
广播带宽	10 KBps	最小置信度	80%
缓存大小	10 ~ 300 items		

3.1 缓存失效率比较结果

缓存失效率指一个广播周期内缓存失效的次数与事务访问缓存中数据的次数之比。在本文中通过引入数据更新率,对每个数据项进行价值评估,因此预取到缓存中的数据失效率与CMIP预取策略将减少许多。比较结果如图4所示。

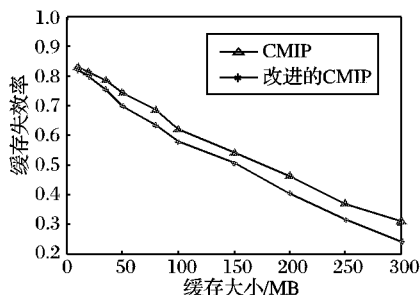


图4 缓存失效率

3.2 电源消耗比较结果

由于客户机电源的消耗主要集中在数据预取上,预取的次数及数据项的大小都将消耗客户机电源能量。对于CMIP

预取策略,将预取所有通过关联规则产生的数据。而本文中通过价值评估,将预取价值相对较大的数据,减小了预取数据集,因此降低了客户机的电源能耗。该实验的结果是基于缓存为150时产生的,比较结果如图5所示。

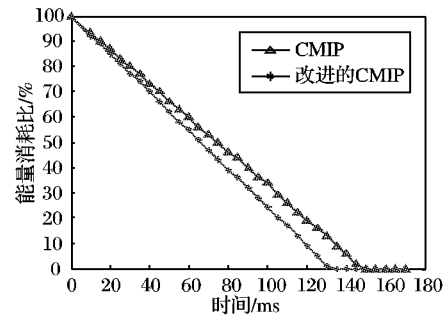


图5 电源能耗

3.3 缓存命中率比较结果

由于CMIP预取策略,预取了所有的通过关联规则挖掘得到的数据,所以缓存的命中率略高于本实验模型。比较结果如图6所示。

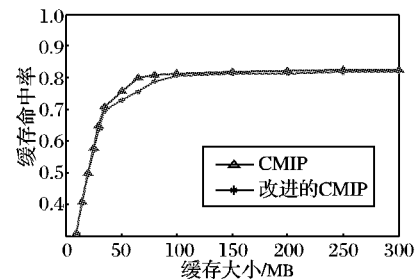


图6 缓存命中率

4 结语

在移动数据库中,数据预取技术是提高数据访问速度和减少数据访问时间的一个有效办法。本文通过在客户端进行关联规则的挖掘来预测将要预取的数据,通过对该数据集中数据进行相关价值评估,增加对数据更新率和数据大小的考虑,最终确定所预取的数据集合。通过对CMIP预取策略的改进,尽管缓存的命中率略有下降,但是明显降低了缓存的失效率,减少了数据访问时间及电能的消耗。

参考文献:

- [1] 刘刚. 移动数据库中数据广播技术的研究[D]. 哈尔滨: 哈尔滨工程大学, 2006.
- [2] 林晨, 黄宇, 金蓓弘. 无线网络环境下的缓存策略研究[J]. 计算机科学, 2009, 36(4): 1-5.
- [3] SONG HUI, CAO GUOHONG. Cache-miss-initiated prefetch in mobile environments[J]. Computer Communications, 2005, 28(7): 370-381.
- [4] YIN LIANGZHONG, CAO GUOHONG. Adaptive power-aware prefetch in wireless networks[J]. IEEE Transactions on Wireless Communications, 2004, 3(5): 1648-1658.
- [5] HAN JIAWEI, KAMBER M. 数据挖掘: 概念与技术[M]. 范明, 孟小峰, 译. 北京: 机械工业出版社, 2006.
- [6] 李国辉, 杨兵, 陈辉, 等. 移动环境下支持实时事务处理的数据预取[J]. 计算机学报, 2008, 31(10): 1841-1847.
- [7] SONG HUI, CAO GUOHONG. On improving the performance of cache invalidation in mobile environments[J]. Mobile Networks and Applications, 2002, 7(4): 291-303.
- [8] SHIM J, SCHEUERMANN P, VINGRALEK R. Proxy cache design: Algorithms, implementation and performance[J]. IEEE Transactions on Knowledge and Data Engineering, 1999, 11(4): 549-562.