

多属性元数据查询系统优化策略

姚念民, 单 颖, 田亚坤

(哈尔滨工程大学 计算机科学与技术系, 黑龙江 哈尔滨 150001)

摘 要: 在 TBF 的基础上提出了用于多属性元数据快速查询的双层体系结构(DBF)和基于分组的延迟更新策略(GDA), GDA 配合 DBF 中的确认证验证机制, 在不会附加错误查询的前提下大幅度的减少了属性更新所需要的操作, 从而能够有效地减小系统更新开销. 边数量更新策略解决了 DBF 中消减操作的更新冲突问题, 同时, 最大限度的减少消减操作次数, 提高了更新效率.

关键词: 元数据快速查询; 延迟更新; 确认证验证机制; 边数量更新

中图分类号: TP333

文献标识码: A

文章编号: 1000-7180(2011)09-0190-03

Inheritance Anomaly in Concurrent Object Orientation

YAO Nian-min, SHAN Ying, TIAN Ya-kun

(Department of Computer Science and Technology, Harbin Engineering University, Harbin 150001, China)

Abstract: A multi-attribute double quick search metadata architecture DBF (Double-level Bloom Filters) and the packet-based update delay strategy GDA (Group-based Delayed-update Approach) is presented on the basis of TBF. GDA combining with the authentication mechanism of DBF can reduce updating operation of object attribute under the premise that the additional error check will not be significantly increased, which can effectively reduce the cost of system updating operation. Updating strategy of edges number proposed reduces operations to minimize the number reduction and improve the updating efficiency while addressing the update operation of the conflict, since the shared edge of DBF ring or chain can not exist simultaneously.

Key words: metadata quick query; delay updating; authentication mechanism; edges updating

1 引言

计算机领域发展的三大阶段分别为计算速度的发展、传输速度的发展、海量数据存储技术的发展^[1-2]. 研究设计简洁的结构对海量数据资源进行表示, 相应的设计高效的查询算法定位数据资源, 成为提升网络软件体系结构、进行高效的大规模数据资源管理的关键, 也成为多种网络应用得以最终推广的基础^[3-7].

文中在 TBF 的基础上提出了用于多属性元数据快速查询的双层体系结构 DBF (Double-level Bloom Filters) 和基于分组的延迟更新策略 GDA

(Group-based Delayed-update Approach), 还提出了边数量更新策略. 在解决 DBF 中由于共享边不能同时为环或链导致的消减操作中更新冲突的同时, 最大限度的减少消减操作次数, 提高了更新效率.

2 设计考虑

2.1 模型分析

针对 TBF 的弊端^[8-10], 文中提出了一种优化体系结构: 双层体系结构 DBF (Double-level Bloom Filters). DBF 包括双层结构: 第一层为 Bloom Filter 阵列, 包含对象的 p 个属性; 第二层为对象 Bloom Filter 阵列, 存储对象的确认值, 用以检查多

收稿日期: 2011-01-05; 修回日期: 2011-01-20

基金项目: 国家自然科学基金项目(61073047); 高效能服务器和存储技术国家重点实验室开放课题基金(2009HSSA08); 中央高校基本科研业务费专项基金(HEUCFT1007, HEUCF100607)

个属性是否属于同一对象^[10]. 当查询对象 X 的请求到来时,对 X 的 p 个属性分别用 k 个哈希函数进行哈希运算,在相应的属性 BF 中进行查询并在存储单元中确认查询. 查询流程如图 1 所示.

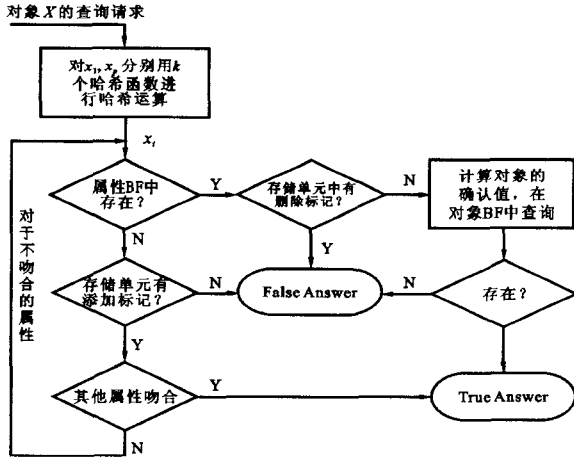


图1 对象查询流程

2.2 基于组的延迟更新策略

为了清楚的表示属性间更新关系,用箭头表示旧的属性到新属性的变化,例如用 $a1 \rightarrow a2$ 来表示旧的属性 $a1$ 被新的属性 $a2$ 替代. 假如存在属性更新关系以 $a2$ 为源头,例如 $a2 \rightarrow a3$,连接两步更新成为一个链. 进一步,如果还有更新为 $a3 \rightarrow a1$ 的话,则三步更新组成一个环,这个环包括更新($a1 \rightarrow a2$, $a2 \rightarrow a3$ 和 $a3 \rightarrow a1$). 显然,并不是所有的更新操作都要一一执行,可以对环或链进行简化:对于环,相当于没有发生属性变化,不需执行任何操作,同理,当有 $a1 \rightarrow a2 \rightarrow a3$ 时,只需执行更新 $a1 \rightarrow a3$ 即可.

基于分组的延迟更新策略 GDA 的基本思想是:在时间段 T 内,属性的更新操作并不立即执行,只是把需要更新的属性信息暂时存储到相应的存储单元中,时间段 T 过后,采用基于分组的策略,集中执行组内对象的所有属性更新操作. GDA 减少大量重复操作,如图 2 所示.

$a_3 \rightarrow a_2$ $b_3 \rightarrow b_2$ $c_2 \rightarrow c_3$
 $a_2 \rightarrow a_3$ $b_2 \rightarrow b_1$ $c_1 \rightarrow c_3$
 $a_3 \rightarrow a_2$ Φ $c_3 \rightarrow c_2$

(a) 基于分组更新的操作

Φ $b_3 \rightarrow b_1$ $c_1 \rightarrow c_3$

(b) 基于分组延迟更新的操作

图2 更新操作

假设在时间段 T 内,对象 $O(a1, b1)$ 中属性 $a1$ 发生变化,变化结果为 $O(a2, b1)$,因为延迟更新在时间段 T 内并没有改变属性 BF 中属性 $a1$ 和 $a2$,只是在相应的存储单元中进行标记,标记属性变化 $a1$

$\rightarrow a2$ 并且标记其所属对象信息 $O(a1, b1)$. 查询请求到来时有以下三种情况:假设查询对象 $O(a1, b1)$ 或是 $O(a1, b2)$,对于属性 $a1$,属性 BF 中有此属性,但是相应的存储单元中有 $a1 \rightarrow a2$ 的标记,则说明属性 $a1$ 已不存在,返回 false;假设查询对象 $O(a2, b1)$,对于属性 $a2$,属性 BF 中无此属性但是相应的存储单元中有 $a1 \rightarrow a2$ 且有 $O(a1, b1)$,则说明由于 $a1 \rightarrow a2$ 的变化对象 $O(a1, b1)$ 转变成 $O(a2, b1)$,返回 true;假设查询对象 $O(a2, b2)$ 的,对于属性 $a2$,属性 BF 中无此属性但是相应的存储单元中有 $a1 \rightarrow a2$ 的标记和 $O(a1, b1)$,对于属性 $b2$,属性 BF 中无此属性并且相应的存储单元中没有标记,返回 false.

由于存储单元中存储有对象信息,部分查询可以从存储单元处直接得出结果,省略计算对象确认值的过程,进一步简化操作. 采用 GDA 不用频繁的计算哈希值,只需暂时存储属性变化及其所属对象的信息到相应的存储单元中,等到时间段 T 过后再进行集中操作. 由于是暂时存储,存储单元只需占用很小的存储空间,并且在存储单元处的确认步骤只是搜索匹配操作,这个操作简单快捷,不需要大量的计算. 每减少一步属性更新操作,相当于用一次复制操作.

替换 $2k$ 次哈希运算和加减运算操作,并且哈希运算的时间复杂度远远高于复制操作的时间复杂度,所以 GDA 在不附加错误查询的基础上极大的减少了重复的更新操作,减少了更新开销,节省了系统资源.

2.3 边数量更新策略

进行属性集中更新可能产生一些被不同的链或环共享的边,当链或环的个数大于共享边的个数时,共享边不能同时为环和链做消减操作,从而导致冲突的产生. 产生冲突时有多种不同的解决方法,各种方法消减掉的边的数量相同,但进行消减操作的次数却不同. 显然,在可消减的边的数量相同的情况下,执行消减操作的次数越少,更新效率越高. 文中提出的边数量更新策略就是本着增加每次消减的边的数量的原则,减少操作次数来提高更新效率.

对于一个多属性对象的更新操作,首先对需要更新的属性进行哈希运算映射到数组中的 k 个位,然后把 k 个位的计数值分别加、减 1,所以属性更新在 BF 中的体现就是位数组中相应的 counter 值的变化,不受更新操作的时间顺序影响.

用 $D[a3 \rightarrow a1]$ 表示 $a3 \rightarrow a1$ 的度, $D[a3 \rightarrow a1] = n$

表示 a_1 替换 a_3 的操作进行 n 次. 将链与环的所有交点中的起始点作为环的起点, 同理, 在环与环相交时, 将共享边的起始点作为各个环的起点. 在链(环)中, 非重复的更新的数目规定为链(环)的长度, 表示为 $LT(LL)$; 将链(环)中所有的度的最小值规定为链(环)的度, 表示为 $DT(DL)$. 链(环)中一次消减的边的数量受限于其中度的最小值, 即 $DT(DL)$, 记 $PT = LT \cdot DT$, $PL = LL \cdot DL$. 显然, $PT(PL)$ 是链(环)中可以一次性消减的更新数量的最大值. 此处对于环或链的各种定义具有等价性, 所以环与环相连、环与链相连和链与链相连三种情况, 本质上是一致的.

为了在解决更新冲突的同时, 最大限度的减少消减操作次数, 提高更新效率, 文中提出边数量更新策略. 当产生更新冲突, 需要在多个包含共享边的环或链中进行选择, 优先使用共享边进行消减操作时, 计算包含共享边的所有的 PT 和 PL 并进行比较, 选取数值最大者优先使用共享边进行消减操作, 保证一次性消减的更新数目最大; 当 PT 和 PL 相等时, 则以环优先, 以上即是“边—数量”更新策略. 该策略可以一次性最大限度的增加消减数量, 减少消减操作的次数, 节省时间, 提高更新效率.

3 结束语

文中在 TBF 的基础上提出了用于多属性元数据快速查询的双层体系结构(DBF)和基于分组的延迟更新策略(GDA), GDA 配合 DBF 中的确认验证机制, 在不会附加错误查询的前提下大幅度的减少了属性更新所需要的操作, 从而能够有效地减小系统更新开销. 文中还提出了边数量更新策略, 解决了 DBF 中由于共享边不能同时为环或链做消减操作而产生的更新冲突, 同时, 最大限度的减少消减操作次数, 提高了更新效率.

参考文献:

- [1] Mesnier M, Ganger G R, Riedel E. Object-based storage [J]. IEEE Communications Magazine, 2003, 41(8): 84—90.
- [2] Mike Mesnier, Gregory R, Ganger, et al. Object-based storage: pushing more functionality into storage [J]. IEEE Communications Magazine, 2005; 31—34.
- [3] Azagury A, Dreizin V, Factor M, et al. Towards an object store [C]// Proc 20th IEEE/11th NASA Goddard Conf Mass Storage Systems and Technologies. Washington, DC, USA; IEEE Computer Society, 2003: 165—176.
- [4] David H C Du, Recent. Advancements and future challenges of storage systems [J]. Proceedings of the IEEE, 2008, 96(11): 1875—1886.
- [5] Bharadwaj Veeravalli. WPAR: a weight-based metadata management strategy for petabyte-scale object storage systems [C]// Proceedings of the fourth international workshop on Storage Network Architecture and Parallel I/Os. San Diego, CA; IEEE, 2007: 99—106.
- [6] Scott A Brandt, Ethan L Miller, Darrell D E Long, et al. Efficient metadata management in large distributed storage systems [C]// Proceedings of the 20th IEEE/11th NASA Goddard Conference on Mass Storage System and Technologies (MSS'03). Washington, DC, USA; IEEE Computer Society, 2003: 290—298.
- [7] LIU Qun, FENG Dan, WANG Fang. Research on metadata server of high reliability [J]. Computer Engineering, 2008, 34(17): 88—90.
- [8] Zhu Y, Jiang H, Wang J. Hierarchical bloom filter arrays (HBA): A novel, scalable metadata management system for large cluster-based storage [C]// Proceedings of 2004 IEEE International Conference on Cluster Computing. US; IEEE, 2004: 165—174.
- [9] Zhu Yifeng, Jiang Hong, Wang Jun, et al. HBA: distributed metadata management for large cluster-based storage systems [J]. IEEE Transactions On Parallel and Distributed Systems, 2008, 19(6): 750—762.
- [10] Hua Yu, Zhu Yifeng, Jiang Hong, et al. Scalable and adaptive metadata management in ultra large-scale file systems [C]// Proceeding of 2004 IEEE International Conference on Cluster Computing System. Beijing: IEEE, 2004: 403—410.

作者简介:

姚念民 男, (1974—), 博士, 教授. 研究方向为网络存储技术.

单颖 女, (1981—), 博士研究生. 研究方向为网络存储技术.

田亚坤 女, (1982—), 硕士. 研究方向为网络存储技术.