计算机科学 2005Vol. 32№. 11

基于 DPB$^+$-Tree 的索引复制策略研究

唐继勇　白新跃　杨　峰　何　建
(电子科技大学自动化学院　成都 610054)

摘　要　索引复制是分布并行数据库提供并行性和提高可用性的一个重要手段。本文提出一种适合于索引复制的树结构——DPB$^+$-Tree,在此基础上研究了相关的索引复制策略,其中副本复制原则考虑了更新/检索比、节点机负载和可靠性需求;索引副本建立允许一个新的副本学习先前的副本;而索引副本更新基于搜索更新机制来完成。对DPB$^+$-Tree 索引复制策略的仿真实验结果表明,副本对查询的响应性能和负载均衡度有明显改善。

关键词　索引复制,DPB$^+$-Tree,复制原则,副本建立,副本更新

Index Replication Strategy Study Based on DPB$^+$-Tree

TANG Ji-Yong　BAI Xin-Yue　YANG Feng　HE Jian
(School of automation , University of Electronic Science and Technolog, Chengdu 610054)

Abstract　Index replication is an important approach that provides parallel and improves usability of distributed parallel database. This paper presents a new tree structure — DPB$^+$-Tree, which is fit for index replication. Then we research the index replication strategy based on DPB$^+$-Tree. The replica duplicating principle includes update/search ratio, machine load and reliability requirement. The replica producing can learn from an old one and the replica updating based on search and update mechanism of DPB$^+$-Tree. The simulation results of index replication strategy demonstrate that replica can improve searchresponse characteristic and load balance.

Keywords　Index replication, DPB$^+$-Tree, Duplicating principle, Replica producing, Replica updating

1　引言

分布并行数据库中的索引复制加强了并行访问数据和利用处理机集的能力,其多余的索引结构增强了数据读能力,使得数据的有效性不会变得依赖于一些可能存储在某个无效处理机上的单个索引副本的有效性。同时由于索引文件的大小通常为数据文件的 1%左右,索引复制和数据复制相比开销很小,但却能够很好地提高系统的性能,因此人们对索引复制[1~9]进行了大量的研究。

这些研究大多侧重于如何改进存储结构来提供对索引复制的支持,而对索引复制的具体策略和相关过程未进一步考虑。针对这种情况,我们首先提出一种适合于索引复制的树结构——DPB$^+$-Tree,该索引树以 B$^+$ 树和 hash 结构为基础,其叶子结点被组织为有 n 个散列表元的 hash 表链,从树的根结点到叶子结点,结点副本数量逐渐减少并动态变化。然后在 DPB$^+$-Tree 的基础上,研究相关的副本复制原则、建立过程以及更新机制。

2　DPB$^+$-Tree 结构

基于 B$^+$ 树特性和分布并行的需求,我们在 B$^+$ 树的基础上构造一种适合于索引复制的树结构——DPB$^+$-Tree(Distributed & Parallel B$^+$-Tree)。

设树的每个结点用结点标识符 i 来区别,结点 i 的级表示为 $L(i)$。当结点 i 是结点 j 的父亲时,$P(j)=i$ 并且 $L(j)=L(i)+1$。对于根结点 r 则 $L(r)=1$,对于叶子结点则 $L(l)=H$,其中 H 为树的高度。

定义 1　设 S_i 为存储结点 i 的节点机集,$F(key)$为一选定的 hash 函数。一个索引树结构满足以下条件:

(1) 如果 $i=P(j)$,则 $S_i \supseteq S_j$;

(2) 如果 $\forall i \exists L(i)=H$,则 i 结点以 hash 方式存储,对应的 hash 函数为 $F(key)$,则称为 DPB$^+$-Tree。

由以上定义

$$S_i=\bigcap_{j=C_i} S_j$$

这里 $c_i=\{k|P(k)=i\}$。因此,对于根结点

$$S_r=\bigcup_{i\in C_i} s_i=\bigcup_{i\in C_i}\bigcup_{j\in C_i} s_j=\cdots=\bigcup_{i\in F_2} s_i=\bigcup_{j\in F_3} s_i=\cdots=\bigcup_{l\in F_h} s_l$$

这里 $F_h=\{k|L(k)=H\}$。

DPB$^+$-Tree 的叶子结点采用 hash 结构来存储数据页的地址指针,以减小索引结点的拆分和更新的开销,和 B+树相比,其空间利用率无明显下降。Hash 函数 $F(key)$的选定通常根据处理冲突的哈希表查找成功时的平均查找长度来确定。平均查找长度是装填因子 $\alpha=\frac{\text{表中填入的记录数}}{\text{哈希表的长度}}$ 的函数,而不是记录数 n 的函数。因此,不管 n 多大,总可以选择一个合适的装填因子将平均查找长度限定在一个范围内。

3　索引复制策略

DPB$^+$-Tree 的索引结点副本进行动态分配,其根据系统的访问统计信息,触发对副本的添加或者是减少,或者是副本的迁移。

3.1　副本复制原则

下面计算副本复制时更新和检索的代价和收益。设 i 为

唐继勇　硕士,副教授,主要研究领域为模式识别与人工智能。

索引数据的标志；j 为节点机的标志；k 为应用的标志；f_{ij} 为索引数据 i 在节点机 j 上出现的概率；f_{kj} 为应用 k 在节点机 j 上出现的频率；r_{ki} 为应用 k 对索引数据 i 进行检索访问的次数；U_{ki} 为应用对索引数据 i 进行更新访问的次数，则对于应用 k 的检索收益为：

$$B_{ijr}=\sum_k f_{kj}f_{ij}r_{ki}$$

增加的更新开销为：

$$B_{ij'u}=\sum_k\sum_{j'\neq j}f_{kj'}f_{ij'}u_{ki}$$

总的收益以检索与更新之差额来考虑为：

$$B_{ij}=B_{ijr}-CB_{ij'u}$$

其中，C 是一常数，即 U/Q。

上面是更新和检索的代价和收益，再考虑到节点机的负载 L_i，可靠性要求 R_i，则判断索引数据 i 是否应该增加或者删除副本的衡量标准是：

$$C_i=a_1B_{ij}+a_2L_i+a_3R_i$$

式中，a_1，a_2 和 a_3 代表各种因素的影响大小。通常 C 设置一个上限和一个下限，当 C_i 超过上限时，其创建一个新的副本，而当 C_i 超过下限时并且副本数≥2，则删除一个副本。

3.2 副本建立

假定索引数据拥有一个或多个索引结点，我们将这些结点通过索引边指针[10]连接起来。同时产生一个起始指针(prime pointer，PP)，PP 负责整个搜索空间，并通过直接的后续和间接的边指针连接到所有别的索引结点。

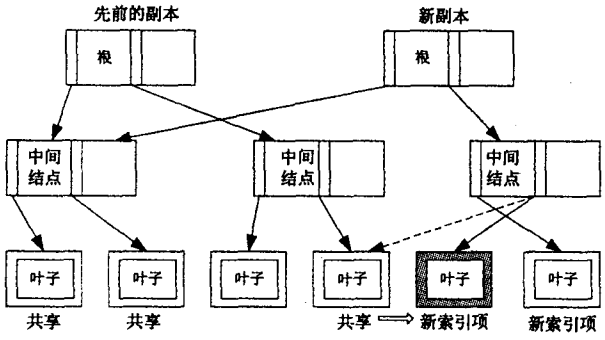

图 1 索引副本的学习和建立

DPB$^+$-Tree 中的新副本可以学习先前的副本，从而利用已经被另一个索引副本发现的索引项(称为 basis)。索引副本的学习和建立过程如图 1 所示，在建立副本的时候，首先将 basis 索引的根拷贝传递给一个要求建立新索引副本的节点机，新的副本收到一批关于访问基本数据的信息，其包括新的索引项以及和 basis 共享的一部分索引结构。新的索引项是从 basis 外的索引数据遍历得来，此遍历通过边索引指针进行。

当副本处理搜索请求的时候，其遍历它的私有结点就像它是一个完全的独立树结构一样。当对某个索引项的搜索涉及到 basis 的一个索引结点时(或者是别的一些副本，因为共享是可以递归的)，其要求获得共享索引中的那个结点。当这个结点被请求节点机收到后，它被复制到本地，并且原来指向远程共享结点的指针改为指向结点的本地拷贝，如图 1 中的阴影叶子结点所示。在这种方式中，新副本以结点大小为单位成批获得索引项，从而提高索引复制的效率。

3.3 副本更新

副本更新基于 DPB$^+$-Tree 的搜索更新机制，该机制通过消息来传递数据组织发生改变的信息。当 DPB$^+$-Tree 结点被遍历的时候，其会积累相关的搜索路径，将它传递给共享副本后，就能够建立新的结点副本。图 2 是从 DPB+-Tree 中返回结点和它们相关搜索路径的搜索过程。

1. 副本 A 在处理一个搜索请求的时候，遇到一个对共享结点 N 的引用，副本 A 就发送一个消息到 B(结点 N 的拥有者)；
2. 副本 B 受到从 A 到来的请求，然后根据此请求访问 N；
   - 如果此搜索导致一个数据结点引用，副本 B 返回结点 N 的内容给副本 A(结点被复制的地点)，在副本 A 中，结点 N 的指针(原来定位在 B 中)变成了指向副本 B 的共享指针；
   - 如果搜索导向副本 B 上的结点 M，那么搜索在 B 上继续，并且添加在搜索过程中所遇到的本地结点到将要返回给 A 的结点集，此结点集包括了通过边遍历所到达的子结点和兄弟结点；
   - 如果搜索导向在另一个副本 C 上的一个共享结点 M，那么 B 从 C 请求 M 并且当 C 返回了对应的结点或者是索引树中更远的结点时，B 收集这些结点，并加上其在本地遍历的结点后返回给 A。同时 B 也将从 C 中返回的结点添加到它的索引副本中。
3. 副本 A 将返回的索引结点序列复制到它的本地索引副本中，并将相关的共享指针改为对应的私有指针。
4. 副本 A 添加返回给对应索引结点的索引项，如果必要的话，进行结构的调整(如结点拆分)。

图 2 副本搜索过程

4 仿真实验

在仿真实验中，每个节点机包含独立的 CPU 和磁盘，通过交换网络相连；并假定 CPU 处理每一类请求的时间和事务所等待的时间成负指数分布，输入请求使用先来先服务方式进行处理；在有副本存在时，任务通过 round-robin 算法在多个副本间进行调度。我们通过改变叶子副本的数量，测试不同操作的响应时间(Response Time)、资源利用率(Resource Utilization)和负载均衡度(Load Balance)来获得相关的性能表示。

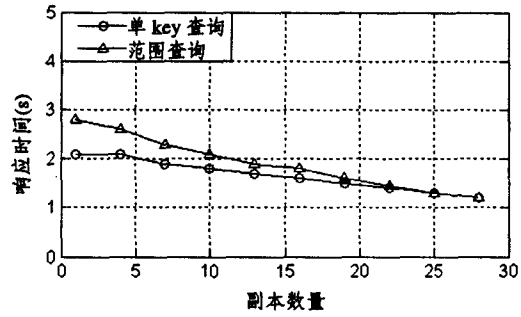

图 3 查询操作中副本数量对响应时间的影响

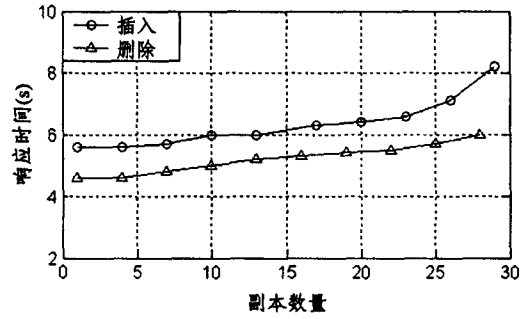

图 4 插入和删除操作中副本数量对响应时间的影响

5.1 叶子副本数量 Ncopy 对响应时间的影响

如图 3 所示，在单 key 查询操作中，随着 N_{copy} 增加，响应性能逐渐变好。这是因为 N_{copy} 增加使得在本地进行操作的可能性增大，减少了通信开销。而且查询任务能够在各个副本之间调度，改善了负载均衡性，使得响应的延迟时间减小。

而对范围查询，其不但使得在本地进行操作的可能性增大，而且查询的并行度也相应增加，因此变化体现得更为明显。图4表明，对于插入和删除操作，随着 N_{copy} 增加，响应性能变差，这是因为 N_{copy} 增加使得副本更新开销及相关的通信开销也相应增加。

5.2 副本数量 N_{copy} 对资源利用率的影响

资源利用率通过综合读写操作来进行模拟。图5显示，对于综合查询操作，随着 N_{copy} 的增加，CPU利用率、磁盘利用率和网络利用率都增加，这是因为 N_{copy} 增加使得副本更新开销随之增加，此外 N_{copy} 的增加还会使得任务并行可能性增大，从而增加相应的启动开销。同时还可看出，网络利用率的增加幅度比CPU利用率和磁盘利用率大，这是由于任务的启动和数据的更新都是通过信息来传递，随着 N_{copy} 的增加，其几乎呈指数上升，因而网络利用率的增加幅度最大。

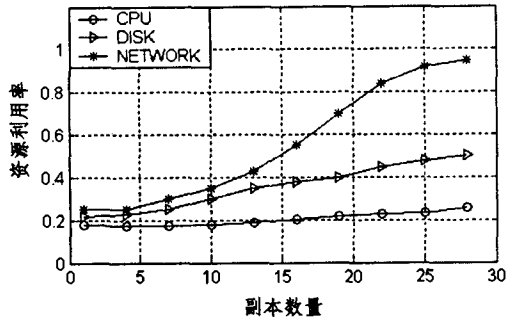

图5 副本数量对资源利用率的影响

5.3 副本数量 N_{copy} 对负载均衡度的影响

负载均衡度也通过综合读写操作来模拟。图6表明，对于综合查询操作，随着副本数量 N_{copy} 的增加，负载均衡度有较为明显的提高，特别是在副本间采用任务调度机制后。

结束语 索引复制是分布并行数据库提供并行性和提高可用性的一个重要手段，本文在 DPB$^+$-Tree 的基础上提出了相关的索引复制策略，包括副本复制原则、建立过程以及更新机制。仿真结果表明：副本对查询的响应性能有明显提高，但也相应增加了更新操作的开销；副本数量对CPU利用率、磁盘利用率和网络利用率有一定的影响；副本复制及其任务调度能够有效改善负载均衡度。

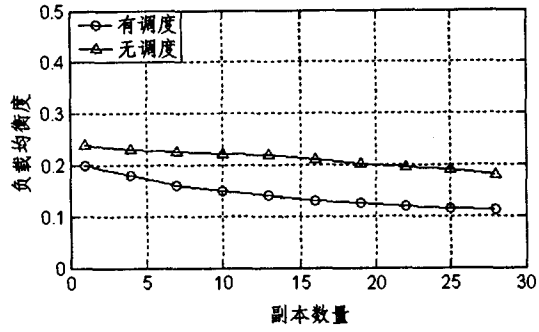

图6 副本数量对负载均衡度的影响

参 考 文 献

1 Yokota H, Kanemasa Y, Miyazaki J. Fat-Btree: An update-conscious parallel direc tory structure. In: 15th Int. Conf. on Data Engineering. Sydney, Australia, 1999. 448～457
2 Lomet D. Replicated Indexes for Distributed Data. In: Proc. of the Fourth Intl. Conf. on Parallel and Distributed Information Systems. Miami Beach, Florida, USA, 1996. 108～119
3 Devine R. Design and Implementation of DDH: Distributed Dynamic Hashing. In: Proc. of the 4th Int. Conf. on Foundations of Data Organization on Algorithms (FODO'93). Chicago, Illinois, 1993. 101～114
4 Litwin W, Neimat M-A, Schneider D. Linear Hashing for Distributed Files. In: Proc. ACM SIGMOD Conf. Washington, D. C, 1993. 327～336
5 Vingralek R, Breitbart Y, Weikum G. Distributed File Organization with Scaleable Cost/Performance. In: Proc. ACM SIGMOD Conf. Minneapolis, MN, 1994. 253～264
6 Kroll B, Widmayer P. Distributing a Search Tree Among a Growing Number of Processors. In: Proc. ACM SIGMOD Conf. Minneapolis, MN, 1994. 265～276
7 Seeger B, Larson P. Multi-Disk B-trees. In: Proc. of ACM SIGMOD Conf. 1991. 436～445
8 Litwin W, Neimat M A, Schneider D A. RP*: A Family of Order-Preserving Scalable Distributed Data Structures. In: Proc. of VLDB'94. 1994, 342～353
9 Johnoson T, Krishna P. Lazy Updates for Distributed Search Structure. In: Proc. ACM SIGMOD Conf. Washington, D. C, 1993. 337～346
10 Lomet D, Salzberg B. Access Method Concurrency with Recovery. In: Proc. ACM SIGMOD Conf. San Diego, CA, 1992. 351～360

(上接第24页)

- $p|\sim X$：P 曾发布过 X，并且 P 在发布 X 时相信 X。
- $p|\equiv X$：P 相信 X，P 认为 X 为真。
- $\#(X)$：p 表示语句 X 是新的，以前从未出现过。
- $p|\equiv\phi(X)$：P 相信 X 是可认知的。
- $\xrightarrow{k} p$：P 拥有公钥 K，相应的私钥为 K^{-1}，且不会被任何其它人知道。
- $p \Rightarrow X$：P 对 X 有控制权，P 是 X 的权威机构。
- 横线代表"推导出"，意即横线上的公式可以推导出横线下的结论。

参 考 文 献

1 Clausen T, Jacquet P. Optimized Link State Routing Protocol (OLSR). RFC 3626, 2003
2 Johnson D B, Maltz D A, Hu Y C. The Dynamic Source Routing Protocol for Mobile Ad Hoc Networks (DSR). Internet-Draft, draft-ietf-manet-dsr-10. txt, July 2004
3 Perkins C, Belding-Royer E, Das S. Ad hoc On-Demand Distance Vector (AODV) Routing. RFC 3561, 2003
4 Zhou L, Hass Z J. Securing Ad Hoc Networks. IEEE Network, 1999, 13(6): 24～30
5 Papadimitratos P, Hass Z J. Secure Routing for Mobile Ad Hoc Networks. In: SCS Communication Networks and Distributed Systems Modeling and Simulation Conf. (CNDS 2002). San Antonio: SCS Press, Jan. 2002. 1～10
6 Dahill B, Levine B N, Royer E, Shields C. A secure routing protocol for ad hoc networks: [Technical Report UM-CS-2001-037]. University of Massachusetts, Department of Computer Science, 2001
7 Hu Y C, Perrig A, Johnson D B. Ariadne: A Secure On-Demand Routing Protocol for Ad Hoc Networks. In: Proc. 8th Ann. Int'l Conf. Mobile Computing and Networking (MobiCom2002). New York: ACM Press, 2002. 12～23
8 Hu Y C, Johnson D B, Perrig A. SEAD: Secure Efficient Distance Vector Routing in Mobile Wireless Ad Hoc Networks. In: Proc. 4th IEEE Workshop on Mobile Computing Systems and Applications (WMCSA 02). IEEE Press, 2002. 234～244
9 Marti S, et al. Mitigating Routing Misbehavior in Mobile Ad Hoc Networks[A]. In: Proc. 6th Ann. Int'l Conf. Mobile Computing and Networking (MobiCom 2000). New York: ACM Press, 2000. 255～265
10 Hu Y C, Perrig A, Johnson D B. Packet Leashes: A Defense against Wormhole Attacks in Wireless Ad Hoc Networks. In: Proc. 22nd Ann. Joint Conf. IEEE Computer and Communications Societies (INFOCOM 2002). IEEE Press, 2003. 1976～1986
11 Li Xiaoqi, Lyu M R, Liu Jiangchuan. A Trust Model Based Routing Protocol for Secure Ad Hoc Networks. In: 2004 IEEE Aerospace Conf. Montana: IEEE press, 2004
12 Xiong Yan, Miao Fu-you, Zhang Wei-chao, Wang Xing-fu. Secure Distributed Authentication Based on Multi-Hop Signing with Encrypted Signature Functions in Mobile Ad Hoc Networks. ACTA ELECTRONICA SINICA, 2003, 31(2): 161～165
13 Capkun S, Buttyan L, Hubaux J P. Self-Organized Public-Key Management for Mobile Ad Hoc Networks. IEEE Transactions On Mobile Computer, 2003, 2(1): 52～63
14 Burrows M, Abadi M, Needham R. A Logic of Authentication [A]. In: Proc. of the 12th ACM Symposium on Operating System Principles. Arizona, Dec. 1989
15 Gong L, Needham R, Yahalom. Reasoning about Belief in Cryptographic Protocols. In: Proc. of the 1990 IEEE Symposium on Research in Security and Privacy. IEEE Press, 1990. 234～248