基于DPB$^+$-Tree的数据迁移策略研究

黄克军[1]，杨 峰[1]，熊 梅[2]，李毅超[1]

(1. 电子科技大学计算机科学与工程学院 成都 610054; 2. 四川日报报业集团印务中心 成都 610016)

【摘要】提出了一种适合于数据迁移、又能保证分布并行特性的树结构DPB$^+$-Tree，讨论了基于DPB$^+$-Tree的数据迁移策略，其中数据节点迁移采用分布式提交协议来保证原子性，索引重构通过对溢出链的hash重排来实现，迁移算法则通过设置负载系数的两个阈值来对负载倾斜进行判断。经模拟实验结果表明，该数据迁移策略能够有效改善系统的负载均衡和吞吐率特性。

关 键 词 并行特性树结构; 数据节点迁移; 索引重构; 迁移算法

中图分类号 TP311.13 文献标识码 A

Data Migrating Strategy Study Based on DPB$^+$-Tree

Huang Kejun[1]，Yang Feng[1]，Xiong Mei[2]，Li Yichao[1]

(1. School of Computer Science and Engineering, UEST of China Chengdu 610054; 2. Printing Center of Sichuan Daily Group Chengdu 610016)

Abstract This paper presented a new tree structure DPB$^+$-Tree, which suitable for data migrating and distributed and parallel. We have studied the data migrating strategy based on DPB$^+$-Tree in which include data node migrating, index restructuring and migrating algorithm. The data node migrating utilise distributed commit protocol to ensure the atom-operation characteristic. Index restructing rearrange overflow chain list following hash rule. According to set two thresholds, a recursive migrating processing is triggered when load coefficient exceed limit and there is another machine can receive data. The simulation results demonstrate: data migrating strategy can improve load balance and throughput characteristic.

Key words distributed and parallel b$^+$-tree; data node migrating; index restructuring; migrating algorithm

在分布并行数据库系统中，随着每个节点机中局部数据的插入和删除，数据的划分长度和访问频度将发生改变，从而发生数据倾斜，为了在查询操作时达到好的负载均衡，必须对数据进行迁移。数据迁移是一种非常耗费资源的操作，包括时间、空间和对外界的服务能力等，因此数据迁移策略一直是分布并行数据库系统中的研究重点。文献[1]使用一种简单分配算法来实现数据迁移，当选中一些候选关系后通过最初的安排策略将其重新分配给各节点机，这种方法没有利用已分配给各节点机并且已经平衡了的那部分数据。文献[2]提出一种自适应的迁移算法，该算法计算出节点机在迁移后应该保留的平均元组数，然后每个节点机保留和平均元组数相等的本地单元数目，剩下的单元合入有序表中并使用最初的安排方法将其分配到各个节点机。此外，许多并行联结算法也可以用于动态数据迁移[3]。文献[4]提出了一种动态hash联结方法，文献[5]改进该方法并提出了三种具有动态负载均衡能力的并行联结算法，这些算法采用了文献[6]所介绍的划分协调概念。本文提出一种适合于数据迁移又能保证分布并行特性的树结构DPB$^+$-Tree(Distributed & Parallel B$^+$-Tree)，讨论了基于DPB$^+$-Tree的数据结点迁移策略，并通过仿真实验验证该策略的有效性。

收稿日期：2004 – 02 – 23

作者简介：黄克军(1966 –)，男，学士，讲师，主要从事计算机网络及信息安全方面的研究；杨 峰(1972 –)，男，博士，讲师，主要从事分布并行处理方面的研究；熊 梅(1969 –)，女，学士，工程师，主要从事网络数据库方面的研究；李毅超(1969 –)，男，硕士，副教授，主要从事计算机网络及信息安全方面的研究

1 DPB$^+$-Tree结构

设树的每个结点用结点标识符i来标识，结点i的级为$L(i)$。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。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。

DPB$^+$-Tree的叶子结点采用hash结构来存储数据页的地址指针，以减小数据迁移时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。具体的结构组织如下：叶子结点采用半分类排序的key值，即对于任意两个叶子结点L_i和L_j，L_i在L_j的左边，则在L_i中的所有key值都小于L_j中的key值。如图1所示，有效的key存储空间在叶子结点被组织为多个散列连接链组成的数组，变量freelist作为有效空间链的头，其中的每一项有一个头记录指针h-ptr，该指针指向每个散列表元链的顶部。字段num用来统计散列表元中的key数量，字段link-r是指向右边下一个散列表元结点的连接指针。当一个结点满时，下一个插入将到一个溢出结点，此时通过一个溢出指针of-ptr连接到当前结点。在DPB$^+$-Tree中的叶子节点还有两个指向左右兄弟的指针，该指针可能指向同一节点机的叶子节点，也可能指向别的节点机上的叶子节点，其主要用于副本的更新和数据重构。

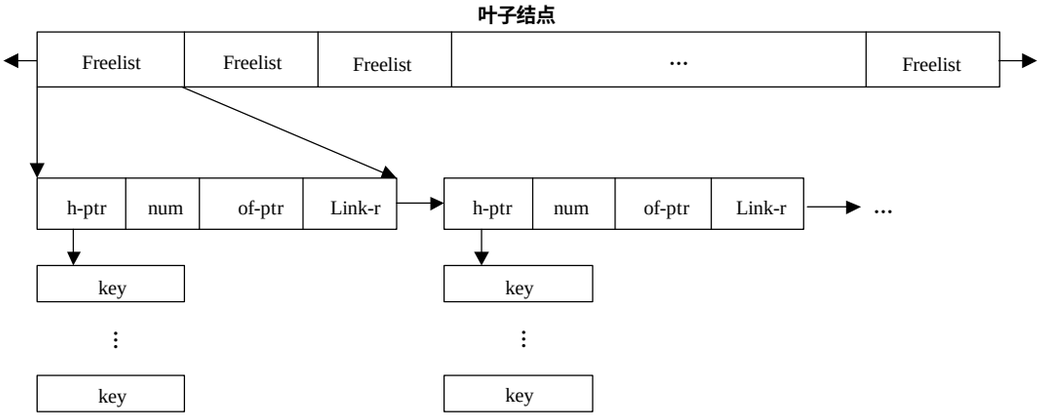

图1　DPB$^+$-Tree中的叶子结点结构

2 数据迁移策略

2.1 数据结点迁移

索引数据结点的迁移活动必须是原子性，在这一原子活动中，需要进行如下两个操作：

1) 分配新结点并将旧结点的内容拷贝到其中；

2) 删除旧结点并产生一个标识新结点的兄弟索引链接作为搜索继续开始的位置。

由于新结点和旧节点处于不同节点机上，因此采用分布式提交协议来提供原子性。假设新结点所处的节点机为N_1，旧结点所处的节点机为N_2。首先节点机N_1上的新结点拷贝节点机N_2上旧结点的内容，然后N_1通知N_2，其通知信息包括相关key值和新结点的地址。N_2收到此消息后删除旧结点，并在相关位置中插入一个兄弟索引链接，此链接指向节点机N_1上的新结点。最后N_2提交本地的原子活动，在N_2提交期间，N_1有可能等待太长，此时它询问N_2是否已经提交。当提交消息丢失时，节点N_2通过简单检查新结点的兄弟索引链接是否已经被更新，来获得提交状态并且回答询问。这里兄弟索引链接用于保持被迁移数据的搜索访问，同时提供搜索更新所需要的信息。

原子操作的结点删除过程如下：

1) 设被删除结点为N，当一个删除请求到达某节点机时，被请求者首先需要移去引用结点N的索引项和相关的数据，当没有索引共享被涉及时，结点的删除很简单，但支持索引共享的删除则需要修改相应的共

享指针；

2) 删除操作必须保证被删除结点的包含结构完整，即被删除结点N的父结点应包含先前由结点N包含的空间，使得原来对结点N空间的搜索转移到对N的父结点的搜索。最后将结点N相关的内容插入其父结点，即结点N的儿子作为其父结点的儿子，结点N的兄弟作为其父结点的兄弟。在上述产生新节点和删除旧结点的过程中，为了保证数据副本一致性，对该结点其他副本(如果存在)的相关内容也同时进行更新。

2.2 索引重构

当数据迁移使得节点机上存在太多的溢出结点时，此节点机上的索引树需要进行重构。索引重构的条件通过对索引的更新事务的连续统计进行检查，对根结点、内部索引结点和叶子结点采用不同的统计函数。

为了确定重构的对象，一个重构进程通过链接指针检查每个叶子结点及其父结点，直到找到需要进行重构的结点为止。具体的重构处理过程从叶子结点的父结点开始，首先检查父结点下的每个叶子结点，如果其没有溢出链，则保持不变。否则，建立一个新的父结点ptemp，并将原父结点的key值和指针拷贝到ptemp。假定有q(q>1)个结点在溢出链中，就将原叶子结点和q个溢出结点中key进行hash重排，从而产生多个水平链接的新结点，这些新结点中最左边的结点链接到原叶子结点的左邻居，最右边结点链接到原叶子结点的右邻居，最后删除原父结点，整个过程如图2所示，其中移动的结点为图中的阴影部分，旧指针用虚箭头表示。

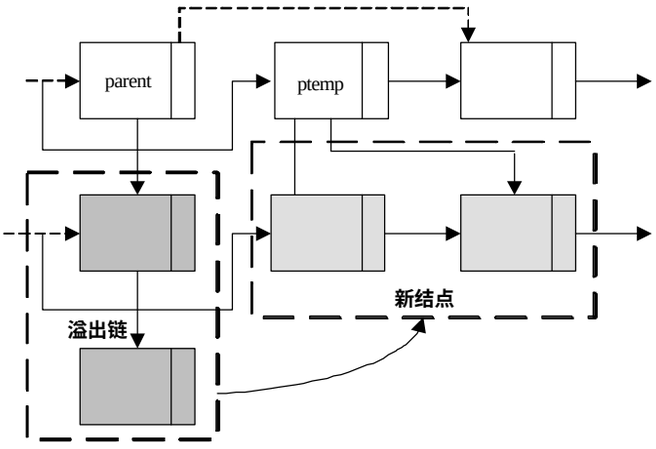

图2 索引结点重构

2.3 数据迁移

数据迁移的前提是副本策略已经不能够保证系统的负载均衡，或副本策略的开销大于数据迁移的开销。为进行数据迁移，需要在节点机间发现索引数据倾斜的程度，本文采用一种基于阈值的方法来进行判断[7~9]。在此方法中，设定节点机的负载系数P=数据量/服务能力，通过对负载系数的两个阈值设置，使每个节点机属于以下三个链之一：

1) P值较小的节点链Enable_list，即能够接收数据迁移的节点机链；

2) P值较大的节点链Disable_list，即需要迁出数据的节点机链；

3) 正常P值节点链Normal_list，即不迁出也不接收数据的节点机链。

在数据迁移条件的判断过程中，每个节点机根据当前负载系数大小对应放入上述各个链中，然后将链发送到下一个节点机，如此不断地循环。如果发现某节点机处于Disable_list，而Enable_list又不为空，同时副本策略已不能满足负载均衡的需要，则此节点机触发数据迁移进程。

定义 2 设系统中有n(n>1)个节点机，记为S_1，S_2，…，S_n。对于每一个节点机S_i(i=1,2,…,n)，其服务能力为C_i，所分配的数据量为I_i，则其负载系数P_i= I_i /C_i。

定义 3 设定两个根据整个系统的平均负载系数来进行动态调整的负载阈值TH_1和TH_2，且TH_1<TH_2，则节点机负载系数P_i< TH_1时为Enable_list，TH_1 < P_i< TH_2时为Normal_list，大于TH_2时为Disable_list。

定义 4 数据迁移的过程如下：在副本策略已不能满足负载均衡需要的情况下，对于Enable_list≠f，且Disable_list≠f，可从Disable_list中节点机向Enable_list中节点机迁移数据，此迁移数据的原则是从Disable_list

向临近(左边或右边)的节点机上迁移数据，其临近的节点机再向下一个临近的节点机迁移数据，以此类推，直到到达Enable_list中的节点机。

如上所述，数据迁移不改变树结构，而且其值范围的分布维持得很好。由于数据的迁移可能会带来其父结点的更新，偶尔会影响更远的祖先结点。如果父结点已经在目的节点机上存在，则仅要求更新对应项，否则父结点将被转移到目的节点机。如果源节点机还包含父结点别的儿子，则源节点机必须保留父结点的一个拷贝。

3　仿真实验

数据迁移对系统性能的影响主要体现在如下两个方面：

1) 负载均衡度和系统开销，其中系统开销可表现为吞吐率变化，仿真实验使用综合读写操作来进行模拟，其中每个节点机包含独立的CPU和磁盘，通过交换网络相连；

2) 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，节点机数量S_{qty}设为30、缓存深度C_{size}设为4 000、任务到达率l_{task}设为200、网络速度设为10 Mb/s。

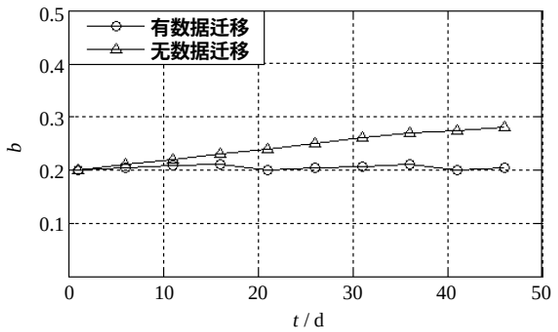

图3　数据迁移对负载均衡度的影响

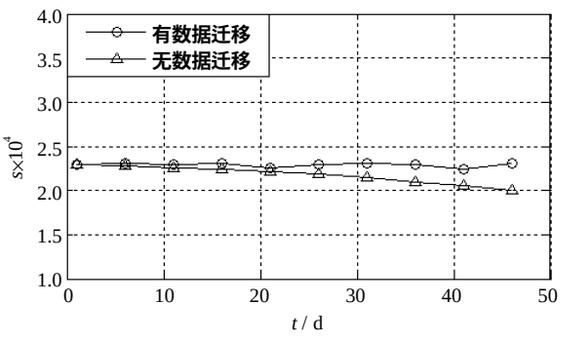

图4　数据迁移对吞吐率的影响

数据迁移对负载均衡度b的影响如图3所示，从图中可以看出，对于综合查询操作，随着运行时间的增长，无数据迁移的负载均衡度逐渐变差，对于采用数据迁移策略的情况，由于每隔一定时间数据进行调整，因此其负载均衡度维持在一定的范围之内。图4是数据迁移对吞吐率的影响，由于无数据迁移时负载均衡度会随运行时间的增长而变差，因此其相应的吞吐率下降。而有数据迁移时，在数据迁移前，其吞吐率也有一定的下降，但一旦数据进行调整，吞吐率就恢复正常。从图3、4还可看出，由于数据迁移是在副本策略不能满足负载均衡的情况下才会发生，因此出现的几率较小。

4　结 束 语

随着计算机应用领域的不断扩大，数据的规模亦越来越大，查询也越来越复杂，分布并行数据库以其高性能已逐渐成为解决这类复杂问题的有效手段，而数据迁移策略一直是分布并行数据库系统的研究重点。本文提出了一种适合于数据迁移又能保证分布并行特性的树结构DPB$^+$-Tree，所讨论的基于DPB$^+$-Tree的数据迁移策略，包括节点迁移、索引重构和迁移算法，是改善系统的负载均衡和吞吐率特性的有效手段。

（下转第224页）

4 结 束 语

用传统方法所判定出元素之间的关系为：Pb、Zn之间的关系最为密切，Ag、Sb之间的关系次之，而Cu元素的分布与以上元素关系不密切，表明Cu元素和其他元素具有不同的地球化学背景和地球化学异常；Pb、Zn、Ag、Sb具有相似的成矿作用和成矿背景，而铜的成矿作用与成矿背景与前者显然是不同的。

从用多重分形方法处理出来的D_q拟合图也可以看出，Pb、Zn之间的关系最为密切，Ag、Sb之间的关系次之，而Cu元素的分布似乎与以上元素关系不密切。本文的结论和传统方法的结论非常接近。

1) 结果和用传统的逐步回归、正交因子分析和聚类分析等常用的统计分析所得到的结果非常相近。

2) 通过作D_q拟合图，只是定性的判定元素之间的关系，完全可以根据D_q寻找一种定量的方法来刻画元素之间的关系，并判定它们的共生组合性，这是可以改进的地方。

参 考 文 献

[1] 赵鹏大, 陈永清, 刘吉平, 等. 地质异常成矿预测理论与实践[M]. 北京: 中国地质大学出版社, 1999

[2] 胥泽银, 郭 科. 多元统计及其程序设计[M]. 成都: 四川大学出版社, 1999

[3] 袁慰平, 孙志忠, 吴宏伟, 等. 计算方法与实习[M]. 南京: 东南大学出版社, 1999

[4] 张济忠. 分形[M]. 北京: 清华大学出版社, 1995

编 辑 漆 蓉

(上接第183页)

参 考 文 献

[1] Copeland G, Alexander W. Boughter E, *et al.* Data placement in Bubba[C]. In Proceedings of ACM SIGMOD Conference, 1988

[2] Hua K, Lee C, Young H. An efficient load balancing strategy for shared-nothing database systems[C]. In Proceedings of DEXA' 92 Conference. Valencia, Spain, 992, 69-474

[3] Lakshmi M, Yu P. Limiting factors of join performance on parallel processors[C]. In Proceedings of the 5th International Conference on Data Engineering. Los Angeles, California, 1989, 488-496

[4] Kitsuregawa M, Yasushi O. Bucket spreading parallel hash: a new, robust, parallel hash join method for data skew in the super database computer(sdc)[C]. In Proceedings of the 16th VLDB Conference. Brisbane, Australia, 1990, 210-221

[5] Hua K, Lee C. Handling data skew in multiprocessor database computers using partition tuning[C]. In Proceedings of the 17th VLDB Conference. Barcelona, Spain, 1991, 525-535

[6] Hua K, Lee C. An adaptive data placement scheme for parallel database computer systems[C]. In Proceedings of the 16th VLDB Conference. Brisbane, Australia,1990, 493-506

[7] Boral H. Parallelism and data management[C]. In Proceedings of 3rd International Conference on Data and Knowledge Bases. Jerusalem, 1988, 362-373

[8] 杨 峰, 刘心松, 左朝树, 唐 续. 分布式并行服务器透明性及任务调度研究[J]. 计算机研究与发展, 2003, 40(9): 1 319-1 325

[9] Hua K A, Su J X W. Dynamic load balancing in very large shared-nothing hypercube database computers[J]. IEEE Transactions on Computer, 1993, 42 (12): 1 425-1 439

编 辑 徐培红