文章编号: 1001-2486 (2000) 04-0061-04

分阶段执行策略及其查询处理技术的研究*

王意洁, 胡守仁

(国防科技大学并行与分布处理国家重点实验室, 湖南 长沙 410073)

摘 要: 针对面向对象数据库及其查询的特点, 提出了查询处理的分阶段执行策略及其数据操作并行执行算法, 理论分析和模拟结果都验证了它们的实用性和有效性。

关键词: 面向对象数据库; 查询处理; 并行
中图分类号: TP311.132.4 文献标识码: A

Research of Stage-by-Stage Executing Strategy and Query Processing Technology

WANG Yi-Jie, HU Shou-Ren
(National Key Laboratory for Parallel & Distributed Processing, National Univ. of Defense Technology, Changsha 410073, China)

Abstract: According to the features of the object-oriented database and its query, the stage-by-stage executing strategy and the data operation parallel executing algorithm are proposed. The theoretical analysis and simulation results show that they are efficient and practical.

Key words: object-oriented database; query processing; parallel

目前, 国际上已经有几个研究机构涉及了并行面向对象数据库 (POODB) 这个研究领域, 但还没有研制出比较成熟的原型系统。从现在的研究情况[1~3]来看, 虽然研究者们的思路各不相同, 但他们的研究存在一些共同的问题, 基本上是借鉴并行关系数据库 (PRDB) 的思想和技术, 并根据面向对象数据库 (OODB) 的特点进行适当的修改和扩充。通过深入分析, 我们发现, OODB 与关系数据库 (RDB) 之间存在着本质区别: ① OODB 处理的是复杂对象而不是规格化的关系元组。④RDB 只处理数据的检索、修改、插入和删除; OODB 除了处理这些传统数据库操作外, 还处理用户定义的操作 (即对象的方法)。因此, 对 PRDB 的思想和技术加以“改造利用”的方法是不能充分利用 OODB 及其查询的本质特点的。

针对 OODB 及其查询的本质特点, 本文提出了查询处理的分阶段执行策略和基于合格标记的数据操作并行执行算法, 并给出了性能评价的结果。

1 分阶段执行策略

由于面向对象数据库本身的特点, 面向对象查询往往很复杂, 所转化成的查询图的结构有可能是线型的、树型的或网络型的, 查询的范围常常是较大规模的外延数据库。这样的查询时间开销大, 尤其是包含多个连接操作的查询。为了避免访问和处理大量不必要的数据, 我们在对面向对象查询处理的本质特点进行分析的基础上, 提出了面向对象查询的分阶段执行策略, 它将查询处理的过程分成两个阶段: 首先找出符合全部查询要求的对象, 然后对这些对象进行性质提取或其它操作以满足用户的需求。因此, 查询处理的关键是找出所有“合格”的对象。我们把这些“合格”的对象以及它们之间的关系构成的集合称为查询的前期结果。查询的前期结果可以用前期结果图来表示, 它是对象图的进化子图, 是查询对象图的子图。

具体而言, 分阶段执行策略的主要思想是: 第一步, 通过在对象类之间处理和传播 Oid, 以 Oid

* 收稿日期: 1999-12-29
基金项目: 国家自然科学基金资助项目 (69903011); 国家部委项目资助 (15.4.1)
作者简介: 王意洁 (1971-), 女, 助理研究员, 博士。

的形式确定“合格”的对象，也就是查询的前期结果；第二步，将系统定义的或用户定义的函数作用于第一步得到的前期结果，从而产生最终结果。在分阶段执行策略中，把对大量描述数据的检索推迟到确定“合格”的对象之后，从而避免了对大量不必要数据的访问和处理；而且，为了减少处理和存储中间结果所需的时间和空间，通过对数据库中符合条件的对象进行标记，来确定“合格”的对象，从而避免产生大量的中间结果。

2 基于合格标记的数据操作并行执行算法

为了开发操作内的并行性，POODB 的研究者提出了基于传统数据放置策略的并行数据操作算法[1~3]，为了便于区分，我们称其为传统的并行数据操作算法。虽然传统的并行数据操作算法比较有效地开发了操作内的并行性，但是，对 OODB 本质特点的考虑不够充分，而且磁盘 I/O 量和网络传输量都较大。针对传统并行数据操作算法的不足，提出了基于合格标记的数据操作并行执行算法，它与传统并行数据操作算法的主要区别是：利用“合格标记”来记录数据操作的结果，而不是利用数据拷贝来记录操作的结果。

对于选择操作来说，“合格”意味着对象满足选择条件，那么，如何有效地进行“合格标记”呢？在一个由 N 个处理机结点通过高速互联网连接的无共享结构中，每个结点的磁盘中都有一张用于记录对象的“合格”标记的表格——标记表，存储于该结点的磁盘上的每一个对象都与标记表中的一项相对应，标记表中的每一项由对象的 Oid 和相应的标记构成。这个标记表由若干个子表构成，每个子表对应于存储在该结点的磁盘上的一个混合式数据子集。

从查询优化的角度考虑，在查询处理时，优先进行选择操作和投影操作，所以我们对连接操作的讨论也是在选择操作和投影操作的基础上进行的。对于连接操作来说，“合格”不仅意味着对象满足选择条件，而且对象之间存在一定的关系。所以，我们利用由两个对象的 Oid 构成的二元组（Oid, Oid）作为连接操作的“合格标记”。我们称这样的二元组为关联模式，所有的关联模式构成关联模式集，它也像标记表一样分布于各个处理机结点上。

查询的基本数据操作有很多，其中，选择（selection）、投影（projection）和连接（join）是三个比较重要的操作，其中，连接是一个既常用又开销大的操作，因此，主要研究连接操作的并行执行。为便于比较，同时给出传统数据操作并行执行算法和基于合格标记的数据操作并行执行算法的描述。

目前，基于无共享结构所进行的有关 POODB 的研究都较充分地考虑了 CPU 处理、磁盘 I/O 和互联网络通信三者的重叠，并进一步考虑了网络发送与网络接收的重叠。为了有效实现这种重叠，数据操作并行执行算法中在每个结点上分配一个接收缓冲区，为每个磁盘分配一个读盘缓冲区和一个写盘缓冲区，为每个远程结点分配一个发送缓冲区。

给定类 C 和类 D，每个类都有 m 个性质，一个无共享结构由 N 个处理机结点通过高速互联网连接。若采用传统的数据放置策略，那么 C 和 D 在所有结点的磁盘上均匀划分，C_i 和 D_i 分别表示存储于结点 i 的磁盘上的 C 和 D 的对象集合，i= 1, 2, …, N；若采用基于对象类的混合式数据放置策略，那么 C 和 D 也在所有结点的磁盘上均匀划分，C_i 和 D_i 分别表示存储于结点 i 的磁盘上的 C 和 D 的混合式数据子集，i= 2, …, N，那么，C_i= {C_{ij}| j= 1, 2, …, m}，D_i= {D_{ij}| j= 1, 2, …, m}，其中 C_{ij}和 D_{ij}分别表示 C 和 D 的对应于性质 j 的数据子集（属性值集或关系集）中存储于结点 i 的磁盘上的部分，i= 1, 2, …, N, j= 1, 2, …, m。不论采用哪种数据放置策略，操作的执行结果可以分散于各结点中，不必收集于某一个结点。

我们首先给出连接操作的问题描述：给定类 C 和类 D，找出所有满足连接条件的对象二元组（CO_i, DO_j）。其中，CO_i 和 DO_j 分别表示类 C 和类 D 中的对象。OODB 与 RDB 的许多本质区别导致了 OODB 中的连接与 RDB 中的连接也不尽相同。根据连接所涉及的两个类之间的关系，将 OODB 中的连接分为两类：隐式连接和显式连接。隐式连接中的两个类之间存在关系（relationship）。显式连接中的两个类之间不存在关系。

现有的 OODB 并行连接算法[1~3]大多是基于散列的算法，我们提出的基于合格标记的连接操作并

行执行算法也是以散列为基础的。下面，我们分别讨论隐式连接和显式连接的并行执行。

2.1 隐式连接

传统的隐式连接并行执行算法的具体思想是:

Step-1 建表 (building) 阶段。具体而言，主要是指每个结点将本结点磁盘上的 C 类对象的有关信息 (包括选择操作的结果，连接性质值，检索性质值等) 传送到与其相关联的 D 类对象所在的结点上，并根据接收到的信息为以后的连接建必要的表格。

Step-2 探询 (probing) 阶段。具体而言，主要是指根据所建表格匹配符合连接条件的 C 类对象和 D 类对象，并将结果存入磁盘。

传统的隐式连接并行执行算法的执行过程中，结点之间不仅传送 Oid 而且传送投影后的大量描述信息，同时大量的描述信息不仅要从磁盘读出而且最后又要存入磁盘。通过分析，我们发现算法的网络传输量大，磁盘 I/O 量大，磁盘占用空间大，这些都直接影响了连接操作的执行效率。基于合格标记的隐式连接并行执行算法通过关联模式 (即由两个对象的 Oid 构成的二元组，如 (c2, d5)) 来记录连接结果，这就避免了大量描述信息从磁盘读出和写入以及在网络上传送。在网络上只需传送 Oid 构成的关联模式，同时也减少了磁盘占用空间，由于连接结果是通过关联模式记录的，而相关联的两类对象往往位于不同的结点上，所以既需要 C 类对象向 D 类对象所在结点发送 Oid 信息，并在 D 类对象所在结点上确定关联模式，又需要 D 类对象向 C 类对象所在结点发送 Oid 信息，并在 C 类对象所在结点上确定关联模式。选择操作的结果是一组 Oid，可将其视作选择结果表。下面，给出基于合格标记的隐式连接并行执行算法的描述:

Step-1 建表阶段 。具体而言，每个结点从磁盘读出类 C 的选择结果表，以 Oid 为关键码建选择结果 Hash 表；每个结点从磁盘读出类 D 的选择结果表，以 Oid 为关键码建选择结果 Hash 表。

Step-2 探询建表阶段。具体而言，每个结点从磁盘读出类 C 的关于连接性质的部分数据子集，通过探询类 C 的选择结果 Hash 表，判断类 C 部分数据子集中的二元组的合格性，并将合格的二元组传送到指定的结点上；同时，根据接收到的信息，为类 D 建立必要的表格。

Step-3 探询阶段。具体而言，每个结点从磁盘读出类 D 的关于连接性质的部分数据子集，通过探询类 D 的选择结果 Hash 表，判断类 D 部分数据子集中的二元组的合格性，若二元组合格，则继续探询连接 Hash 表，对符合连接条件的 C 类对象和 D 类对象进行连接，形成关联模式集，并将类 D 的关联模式集写入磁盘，同时发送到 C 类对象所在结点；每个结点将接收到的关联模式写入磁盘。

2.2 显式连接

传统的显式连接并行执行算法的具体描述如下:

Step-1 建表 (building) 阶段。具体而言，主要是指每个结点将本结点磁盘上的 C 类对象的有关信息 (包括选择操作结果，连接性质值，检索性质值等) 传送到所有其它 N- 1 个结点上，并根据本结点上的 C 类对象的有关信息和接收到的信息，为以后的连接建立必要的表格 (如以连接性质为关键码建 Hash 表)。

Step-2 探询 (probing) 阶段。具体而言，主要是指根据所建表格匹配符合连接条件的 C 类对象和 D 类对象，并将结果存入磁盘。

基于合格标记的显式连接并行执行算法与基于合格标记的隐式连接并行执行算法很相似，它的具体描述如下:

Step-1 建表阶段。每个结点从磁盘读出类 C 的选择结果表，并以 Oid 为关键码建选择结果 Hash 表；每个结点从磁盘读出类 D 的选择结果表，并以 Oid 为关键码建选择结果 Hash 表。

Step-2 探询建表阶段。每个结点从磁盘读出类 C 的关于连接性质的部分数据子集，通过探询类 C 的选择结果 Hash 表，判断类 C 部分数据子集中的二元组的合格性，并将合格的二元组传送到其它所有 N- 1 个结点上；同时，根据接收到的信息，为类 D 建立必要的表格

(如以连接性质为关键码建连接 Hash 表)。

Step-3 探询阶段。每个结点从磁盘读出类 D 的关于连接性质的部分数据子集，通过探询类 D 的选择结果 Hash 表，判断类 D 部分数据子集中二元组的合格性，若合格，则继续探询连接 Hash 表，对符合连接条件的 C 类对象和 D 类对象进行连接，形成关联模式集，并将类 D 的关联模式集写入磁盘，同时发送到 C 类对象所在结点；每个结点将接收到的关联模式写入磁盘。

3 性能评价

由于实验环境等客观条件的限制，我们采用理论分析与模拟实验相结合的方法来对比分析基于合格标记的数据操作并行执行算法与传统的数据操作并行执行算法的性能。

我们编制了模拟测试程序，利用前面推导的代价公式计算了传统的数据操作并行执行算法和基于合格标记的数据操作并行执行算法的响应时间。模拟的工作环境是：多台 SGI Challenge 服务器 (MIPS R4400 芯片，128MIPS) 通过网络互联，网络启动时间为 0.05ms，传输率为 75Mb/s。

通过改变重要参数的取值，观察两种算法响应时间比值 ratio (传统算法的响应时间 / 基于合格标记的隐式连接并行执行算法的响应时间) 的变化。由模拟测试结果可以看出:

(1) 在一般情况下，基于合格标记的连接并行执行算法明显优于传统的连接并行执行算法;

(2) ratio 与大属性数目 P 成正比，趋于线性关系，原因是：随着 P 的增加，传统的连接并行执行算法的磁盘 I/O 量和网络传输量都迅速增长，而基于合格标记的连接并行执行算法的磁盘 I/O 量和 CPU 开销基本不变;

(3) ratio 与选择率 sel 成正比，原因是：随着 sel 的增加，传统的连接并行执行算法的磁盘 I/O 量和网络传输量都迅速增长，而基于合格标记的连接并行执行算法的磁盘 I/O 量和 CPU 开销增长不明显;

(4) ratio 与大性质的大小 S_{big} 成正比，趋于线性关系，原因是：随着 S_{big} 的增加，传统的连接并行执行算法的磁盘 I/O 量和网络传输量都迅速增长，基于合格标记的连接并行执行算法的磁盘 I/O 量和 CPU 开销基本不变;

4 结论

分阶段执行策略和基于合格标记的数据操作并行执行算法利用“合格标记”来记录数据操作的结果，而不是利用数据拷贝来记录操作的结果，从而有效地降低了磁盘 I/O 量和网络传输量。它比较依赖于最初的数据放置，而且不涉及结果数据的放置，这不仅减少了对后续的数据操作的影响，而且减少了数据偏斜产生的可能性。

模拟测试结果表明，在一般情况下，基于合格标记的数据操作并行执行算法优于传统的数据操作并行执行算法。分析基于合格标记的数据操作并行执行算法的模拟测试结果，我们发现：当数据操作涉及的对象的大性质数目越多，大性质占用的存储空间越多，基于合格标记的数据操作并行执行算法就越优于传统的数据操作并行执行算法 (反映在算法的响应时间比值 ratio 越大)。这充分说明了分阶段执行策略和基于合格标记的数据操作并行执行算法抓住了 OODB 及其查询的本质特点，具有一定的实用性和有效性。

参考文献:

[1] Haddleton R F. An Implementation of a Parallel Object Oriented Database System [R]. Technical Report, University of Virginia, 1995.

[2] Bassiliades N, Vlahavas I. Hierarchical query execution in a parallel object-oriented database system [J]. Parallel Computing, 1996, 22 (3): 1017-1048.

[3] Lieuwen D F. Dewitt D J. Mehta M. Parallel Pointer-based Join Techniques for Object-Oriented Databases [C]. In Proc. of the Second Int' l Conf. on Parallel and Distributed Information Systems, 1993: 172-181.