基于半连接的并行查询处理算法的研究*

王意洁，王勇军，卢锡城

(国防科学技术大学 计算机学院 并行与分布处理国家重点实验室，湖南 长沙 410073)

E-mail: wwyyjj1971@sina.com

http://www.nudt.edu.cn

摘要：多元连接查询的并行执行是并行数据库的研究重点，传统的并行查询处理算法没有利用面向对象数据库及其查询的特点，算法效率较低.借鉴分布式数据库查询处理中基于半连接的优化思想，提出了基于半连接的并行查询处理算法.性能评价表明了其实用性和有效性.

关键词：面向对象数据库；多元连接查询；并行；半连接；性能评价

中图法分类号：TP311 文献标识码：A

多元连接查询的执行效率直接反映了数据库系统的性能，它始终是并行数据库领域的研究热点.多元连接查询的并行执行可以分为两步：(1) 对象匹配：利用并行查询处理算法得到查询结果，这些查询结果分布于各个处理机结点上；(2) 结果回收：回收分布于各结点上的查询结果，得到最终完整的查询结果.

传统的并行查询处理算法[1~3]以基于树的查询执行计划模型为基础，在算法执行过程中产生大量的中间结果，从而导致大量的系统开销，抵销了并行性带来的效率的提高，以致影响并行面向对象数据库的性能.针对传统并行查询处理算法的不足，提出了基于半连接的并行查询处理算法[4]，有效地实现了面向对象数据库中的并行查询处理.

在内涵级和外延级两个层次上都可以用图来表示面向对象数据库.在内涵级，数据库可以定义为一个由相互关联的类构成的集合，采用模式图来表示.在外延级，数据库可以视为由属于不同类的相互关联的对象构成的网络，采用对象图来表示.可以将面向对象数据库中的查询转化为查询图.查询图是模式图的进化子图，具体而言，查询图是由模式图的了图依据类与类之间的继承关系进化而成.

1 研究基础

分阶段执行策略[5]的主要思想是：第1步，通过在对象类之间处理和传播对象标识Oid，以Oid的形式确定“合格”的对象，也就是查询的前期结果；第2步，将系统定义的或用户定义的函数作用于第1步得到的前期结果，从而产生最终结果.

基于对象类的混合式数据放置策略[4]包括混合式数据划分策略和基于对象类的数据分配策略.混合式数据划分策略的具体思想是：首先，对数据库中的对象进行垂直划分，产生数据子集；然

* 收稿日期：1999-05-17；修改日期：1999-12-03

基金项目：国家自然科学基金资助项目(69903011,68933030)；部委级基金资助项目

作者简介：王意洁(1971—)，女，江苏镇江人，博士，副研究员，主要研究领域为数据库，并行分布处理技术；王勇军(1971—)，男，江西高安人，博士，副研究员，主要研究领域为虚拟现实，数据库，并行分布处理技术；卢锡城(1946—)，男，上海人，教授，博士生导师，主要研究领域为大规模并行处理技术，先进网络技术.

后,对数据子集基于Oid进行水平划分,得到混合式数据子集.基于对象类的数据分配策略对混合式数据划分策略的结果进行合理分配.

基于合格标记的数据操作并行执行算法[4]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.

2 并行查询处理算法

传统的并行查询处理算法以连接操作为基础,通过宏观控制各连接操作的执行次序来实现查询的有效处理.它的基本思想可以归结如下:

(设在查询图中,类C的相邻类数目为AN(C),与其进行过连接操作的相邻类数目为JAN(C))

依次对每个类进行如下处理,直到所有类的AN等于JAN:

· IF (AN(C)-JAN(C))==1

THEN (设类D是类C的相邻类,且类C没有与类D进行过连接操作)

类C与相邻类D进行连接操作,即:C×D;

JAN(C)++;

JAN(D)++.

针对传统并行查询处理算法的不足,我们采用基于图的查询执行计划模型,在分阶段执行策略、基于对象类的混合式数据放置策略和基于合格标记的数据操作并行执行算法的基础上,进一步将一个连接操作转化成两个半连接操作,并通过宏观控制所有半连接操作的执行次序来实现查询的有效处理,这就是我们提出的基于半连接的并行查询处理算法.对于连接操作来说,"合格"不仅意味着对象满足选择条件,而且对象之间还要存在一定的关系.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.对于多元连接查询来说,用作合格标记的关联模式应为对象多元组(Oid,...,Oid),它由查询涉及的各类对象的Oid构成,随着查询处理的不断进行,关联模式中的Oid数目也由2增加至Num_Of_Class(Num_Of_Class为查询涉及的类的数目).

基于半连接的并行查询处理算法的基本思想是:

(设在查询图中,类C的相邻类数目为AN(C),类C向JANS(C)个相邻类发送过消息并与它们进行半连接操作,JANR(C)个相邻类向类C发送过消息并与类C进行半连接操作)

· IF (AN(C)-JANR(C))==1 AND JANS(C)==0

THEN

(设类D是类C的相邻类,且类D没有与类C进行过半连接操作,即D→C)

类C向相邻类D发送消息并与其进行半连接操作(即C→D),

其目的是进一步判定D类对象的合格性,更新D类的关联模式集;

JANS(C)++;

JANR(D)++;

· IF AN(C)==JANR(C) AND JANS(C)==0

THEN 类C向所有相邻类发送消息并与它们进行半连接操作;

(即

FOR 类C的每一个相邻类D

C向D发送消息并与其进行半连接操作(C→D),

其目的是进一步判定D类对象的合格性,

更新D类的关联模式集;

JANS(C)++;
JANR(D)++;
ENDFOR)

· IF AN(C)==JANR(C) AND JANS(C)==1
THEN (说明类C仅向一个相邻类发送过消息并与其进行了半连接操作)
类C向所有其他相邻类发送消息并与它们进行半连接操作;
(即
FOR 类C的每一个相邻类D且C未与其进行过半连接操作
C向D发送消息并与其进行半连接操作(即C→D),
其目的是进一步判定D类对象的合格性,
更新D类的关联模式集;
JANS(C)++;
JANR(D)++;
ENDFOR)

分析算法的执行过程,可以发现:

(1) 算法结束后,各类的最终关联模式集是相同的;

(2) 当类C收到来自其所有相邻类的消息后(即AN(C)==JANR(C)),就已经得到了类C的最终合格对象和反映查询结果的最终关联模式集.

所以,当AN(C)==JANR(C)时,类C与其相邻类的半连接操作可以简化,即无需进行繁琐的建表操作和探询操作,只需利用类C的关联模式集去更新其相邻类的关联模式集,简化后的半连接操作被称为伪半连接操作.

鉴于上述考虑,对基于半连接的并行查询处理算法进行改进,改进后的算法的基本思想如下:

· IF (AN(C)-JANR(C))--1 AND JANS(C)==0
THEN
(设类D是类C的相邻类,且类D没有与类C进行过半连接操作,即D→C)
类C向相邻类D发送消息并与其进行半连接操作(即C→D),
其目的是进一步判定D类对象的合格性,更新D类的关联模式集;
JANS(C)++;
JANR(D)++;

· IF AN(C)==JANR(C) AND JANS(C)==0
THEN 类C向所有相邻类发送消息并与它们进行伪半连接操作;
(即
FOR 类C的每一个相邻类D
C向D发送消息并与其进行伪半连接操作(C→D),
其目的是进一步判定D类对象的合格性,
更新D类的关联模式集;
JANS(C)++;
JANR(D)++;
ENDFOR)

· IF AN(C)==JANR(C) AND JANS(C)==1
THEN(说明类C仅向一个相邻类发送过消息并与其进行了半连接操作)
类C向所有其他相邻类发送消息并与它们进行伪半连接操作;

```
(即
  FOR 类 C 的每一个相邻类 D 且 C 未与其进行过半连接操作
      C 向 D 发送消息并与其进行伪半连接操作(即 C→D),
      其目的是进一步判定 D 类对象的合格性,
      更新 D 类的关联模式集;
      JANS(C)++;
      JANR(D)++;
  ENDFOR)
```

3 性能评价

多元连接查询的并行执行包括对象匹配和结果回收两部分,在对象匹配中采用的并行查询处理算法不同,相应的结果回收算法也不同.为了对基于半连接的并行查询处理算法进行较全面的评价,我们对查询的整个并行执行过程进行了性能分析,既考虑了对象匹配,也考虑了结果回收.由于受到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半连接的并行查询处理算法与传统的并行查询处理算法的性能.由于篇幅所限,这里只给出性能评价的设计和结果,详细的说明参见文献[4].

我们设计的模拟测试程序接收查询图作为输入,分别采用传统方法和基于半连接的方法对其进行模拟处理,计算它们的执行时间.我们分别产生了 16 个涉及 4 个类的树型查询(最多产生 16 个)、100 个涉及 8 个类的树型查询、100 个涉及 12 个类的树型查询和 100 个涉及 16 个类的树型查询.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.测试参数取值参见表 1.

Table 1 Parameters

表 1 测试参数

Parameter①	Value②	Parameter	Value	Parameter	Value
N	8	The size of page③	16KB	t_{opera}	0.005ms
m	4	ts	0.05ms	t_{insert}	0.002ms
JS	10^{-8}	t_{io}	0.05ms	t_{hash}	0.003ms
size_of disk_array	10	t_{comp}	0.008ms	F	1.4
S_{small}	16B	t_{disk}	5ms		
S_{big}	8KB	t_{net}	0.2ms		

①参数,②取值,③页面大小.

在表 1 中,t_s 表示启动互联网络发送一页数据或启动互联网络接收一页数据所用的 CPU 时间;t_{io} 表示启动一次顺序磁盘 I/O 所用的 CPU 时间;t_{disk} 表示磁盘传送一页数据的时间;t_{net} 表示网络传送一页数据的时间;t_{hash} 表示计算键字的散列函数所花费的 CPU 时间;t_{insert} 表示往散列表中插入一项所花费的 CPU 时间;t_{comp} 表示执行一个比较操作所花费的 CPU 时间;t_{mark} 表示对标记子表中的一项进行修改所花费的 CPU 时间;F 表示散列表的平均检索长度;t_{opera} 表示构造一个关联模式(对象二元组)所花费的 CPU 时间;N 表示处理机结点数目;JS 表示连接选择率;m 表示每个类所含的大性质数目;size_of disk_array 表示 Master 结点上的磁盘数.

类的性质按照其大小可分为小性质和大性质,小性质主要是指关系和简单的属性(如:整数),大性质主要是指复杂的属性(如图形、文本等).设大性质的平均大小为 S_{big},小性质(二元组)的平均大小为 S_{small}.

通过改变 N,m,S_{big} 等重要参数的取值,观察两种方法的执行效率的变化.

*结点数目 N 的变化对执行时间比值的影响

·两种算法的执行时间都随着结点数目的增加而减少,传统算法的执行时间的减少幅度略大一点.

·两种方法的执行时间都是由并行查询处理算法的执行时间和结果回收时间组成,而且结果回收时间占有很大份额;随着结点数目的增加,两种算法的执行时间都随着结点数目的增加而减少,且减少幅度相近;两种方法的结果回收时间都随着结点数目的增加而减少,且传统方法的结果回收时间的减少幅度更大,但是,当结点数目达到一定的数值时,Master 结点成为结果回收的瓶颈,导致两种方法的结果回收时间都不随结点数目的增加而发生变化.

*每个类所含大性质数目 m 的变化对执行时间比值的影响

·随着每个类所含大性质数目 m 的增加,传统算法的执行时间逐渐增加,基于半连接的算法的执行时间不变.

·两种方法的执行时间都是由并行查询处理算法的执行时间和结果回收时间组成,而且结果回收时间占有很大份额;随着每个类所含大性质数目 m 的增加,传统算法的执行时间逐渐增加,基于半连接的算法的执行时间不变;两种方法的结果回收时间都随着每个类所含大性质数目 m 的增加而增加,且增加的幅度相当.

*大性质的大小 S_{big} 的变化对执行时间比值的影响

·随着大性质的大小 S_{big} 的增加,传统算法的执行时间逐渐增加,趋于线性增加,基于半连接的算法的执行时间不变.

·两种方法的执行时间都是由并行查询处理算法的执行时间和结果回收时间组成,而且结果回收时间占有很大份额;随着大性质的大小 S_{big} 的增加,传统算法的执行时间逐渐增加,基于半连接的算法的执行时间不变;两种方法的结果回收时间都随着大性质的大小 S_{big} 的增加而增加,且增加的幅度相当.

4 结 论

从模拟测试结果可以看出,在一般情况下,基于半连接的并行查询处理算法优于传统的并行查询处理算法.类的大性质数目越多,大性质占用的存储空间越多,基于半连接的并行查询处理算法就越优于传统的并行查询处理算法.这充分说明了基于半连接的并行查询处理算法抓住了面向对象数据库及其查询的本质特点.基于半连接的并行查询处理算法是针对面向对象数据库及其查询的本质特点而提出的,具有一定的实用性和有效性.

References:

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

[2] Haddleton, R. F. Parallel set operations in complex object oriented queries [Ph. D. Thesis]. University of Virginia, 1998.

[3] van den Berg, C. A. Dynamic query processing in a parallel object-oriented database system [Ph. D. Thesis]. Twente University, 1994.

[4] Wang, Yi-jie. Parallel query processing and transaction management in the object-oriented database [Ph. D. Thesis]. Changsha: Graduate School of National University of Defence Technology, 1998 (in Chinese).

附中文参考文献:

[4] 王意洁.面向对象数据库的并行查询处理与事务管理[博士学位论文].长沙:国防科学技术大学研究生院,1998.

Research on the Semi-Join-Based Parallel Query Processing Algorithm*

WANG Yi-jie, WANG Yong-jun, LU Xi-cheng

(*National Laboratory for Parallel and Distributed Processing, School of Computer, National University of Defence Technology, Changsha 410073, China*)

E-mail: wwyyjj1971@sina.com

http://www.nudt.edu.cn

Abstract: Parallel execution of the multi-join query is one of the research emphases of the parallel database. The traditional parallel query processing algorithm can not capture the intrinsic characteristic of object-oriented database and its query, and the efficiency of the traditional algorithm is low. Hence, the semi-join-based optimization method of query processing in the distributed database is utilized, a semi-join-based parallel query processing algorithm is proposed, and the performance evaluation results show that the semi-join-based parallel query processing algorithm is an efficient practical algorithm.

Key words: object-oriented database; multi-join query; parallel; semi-join; performance evaluation

* Received May 17, 1999; accepted December 3, 1999

Supported by the National Natural Science Foundation of China under Grant Nos. 69903011, 69933030; the Ministry&Commision-Level Research Foundation of China