XML 数据库的并行 RPE 查询*)

胡军安　于亚新　王国仁　于　戈
(东北大学信息科学与工程学院　沈阳110004)

Parallel RPE Query in XML Database System

HU Jun-An　YU Ya-Xin　WANG Guo-Ren　YU Ge
(College of Information Science and Engineering, Northeast University, Shenyang 110004)

Abstract　Existing query methods on XML documents are usually designed for centralized environments. As the amount of documents in Web applications is increasing very quickly, the existing query methods can not meet the needs of new Web applications. By extending the query processing strategy for centralized environments, we design and implement a parallel query processing method based on RPE. The experimental results show that the method has good speedup and scaleup performance in the case of heavy workload.

Keywords　XML documents, Parallel query, Index, Orient-object databases

1. 引言

随着 Internet 应用的不断发展和日益普及，Internet 上信息的定义和表示成为了重要的技术标准。目前，HTML 标记语言由于其简单、易用等特点，已经成为广泛采用的一种标记语言。但是随着 Internet 上的信息量迅速地增长，在浩如烟海的信息中检索需要的内容变得越来越困难。HTML 只是简单标记文档的展示格式，而不能把文档中的语义结构有效地表示出来，因而诸如 Yahoo，Google 等搜索引擎只能用关键字对全文进行匹配的方法来找到用户所需要的内容。这种方法不仅速度慢，而且准确率也比较低，很难满足用户的需求。

在这种情况下，W3C 提出了新的 Internet 标记语言—XML (eXtensible Markup Language)[1]，它不但可以用来创建标记语言，而且能够通过它来创建高度结构化的标记语言。因此它可以在最大程度上提供文档的语义结构信息，从而使搜索引擎可以变得"聪明"起来，而且可以进行基于文档结构而不仅仅是正文的搜索。Web 上的数据变成了自我描述，这可以使程序更加充分地利用它们。自提出之日起，XML 有了快速的发展，许多相关的标准相继问世，各个主要的业界公司都纷纷宣布支持 XML。可以预见 XML 在不久的将来将成为 Internet 上主要的标记语言而取代现在 HTML 的位置。

XML 的存储和查询是目前亟待解决的问题之一，虽然许多公司都宣布支持 XML，但是由于推出的时间比较短，各种标准并不十分完善。许多机构都在研究这方面的内容并推出了自己的一套系统。有基于文件系统的，有基于关系数据库的，也有基于对象数据库的查询系统。在存储和查询方面有很多研究成果。

另一方面，Internet 上的信息量是非常大的，同时用户对查询速度也十分敏感。在结果准确的前提下，要求响应速度尽可能快。在当今信息作为重要的社会资源以及信息"爆炸"的时代，大量的各种类型的数据都存贮在数据库之中。其中不仅包括简单的基本数据类型，还包括声音、图像、视频及超大文本等等复杂的数据类型。虽然硬件的发展解决了存储容量的问题。但是原有的体系结构却限制了性能的提高。为了提高数据处理的吞吐量，降低查询相应时间，并行技术是一条必由之路[4]。

在目前分布式并行对象数据库以及单机 XML 文档查询研究的基础之上，我们开发了并行 XML 数据库查询系统。该系统将 XML 文档以 DOM 树的形式存放在分布式并行对象数据库中，并提供了对 RPE 路径的并行查询功能。将大文档分布到多个处理机之上后，可以有效地降低响应时间，提高查询效率。

XML 本身是一种半结构化文档的数据交换标准，W3C 提出了若干 XML 文档的表示模型，如 OEM (Object Exchange Model) 模型和 DOM (Document Object Model) 模型[2]。前者表现为一种图的形式，后者表现为一种树的形式。无论是哪一种数据模型，对 XML 文档的查询都是基于一种正则路径表达式(Regular Path Expression, RPE)的查询。在 RPE 查询的基础上我们可以进行其他相关的操作，如排序、计数等等。

例如，对于查询

```
For $b IN document("auction.xml")/site/regions
RETURN COUNT($b//item)
```

该查询可以看作是在路径/site/regions//item 查询的基础之上再做一个 COUNT 函数计算的工作后得到的。由于在 XQuery 查询语言中 RPE 是一种重要的查询设施，因此在本文中我们考虑的是路径表达式的并行处理问题。

2. 并行 XML 文档查询系统的体系结构

本系统是在分布式对象数据库 FISH 之上的 XML 文档查询系统。FISH[2]是我们与日本九州大学共同开发的分布式面向对象数据库。图1是本系统的体系结构图。

整个系统体系可以分为三层，最上层是控制层，在前端机(Front End)上进行 XML 文档数据的生成、分片、索引、生成物理查询计划和结果合并及处理，以及同步与控制、物理查询计划的优化等工作。中间层是用于传输同步控制消息以及数据的通信层，主要采用 FISH 提供的 DSVM 机制和 Socket 接口进行。下层是实现层，即在各个站点上利用已经建立好的索引，执行前端机返送来的物理查询计划，然后将结果返回给前端机。下面详细介绍系统中的主要模块：

*)本文受高等学校优秀青年教师教学科研奖励计划、国家自然科学基金(60173051)资助。

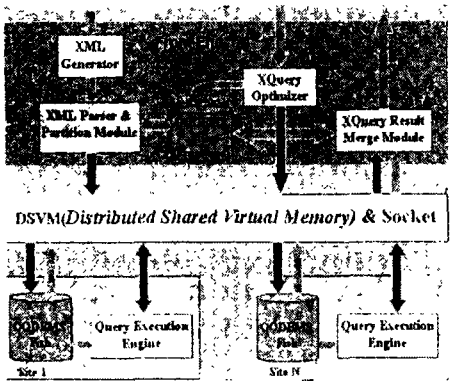

图1 系统的体系结构图

(1)XML 语法分析和文档分片模块:能够对 XML 文本进行语法分析,能够进行元素的定位和遍历。负责对 DTD 文件,XML 文件进行语法分析,负责按照文件类型定义生成数据库的模式。将 XML 文档按照一定的要求快速准确地转换成为一棵 DOM 树,然后再将 DOM 树分布到各个站点之上。可以查看各个站点文档 DOM 树内容,DTD 等模式信息,以及用户所需要的其他信息。

并行数据库的关键技术之一在于数据的分片,我们采用路径实例轮循分布法。即对全局 DOM 树上每一条从根节点到叶子节点的路径实例,参照 DTD 所提供的路径模式信息,将每种模式的每个路径实例均以轮循的方式分布到各个站点之上。这样就可以保证每种模式的路径实例在各个站点上的分布是均衡的,有效地防止了数据原始倾斜。分片完毕后,再在各个站点上按照 XML 文档本身的 DTD 模式信息重新组成与全局 DOM 树同构的一棵子 DOM 树。即二者符合同样的模式,但后者只是前者的部分子集。各个站点上的子 DOM 树合并在一起就构成了原有的全局 DOM 树。保证了查询的完备性。

设有全局 DOM 树 D,经过分片后每个站点上都有一个子 DOM 树 D_i($1 \leqslant i \leqslant n$,其中 n 为站点的个数)。有:

$$D=\sum_{i=1}^{n} D_i$$

其中,$\sum$ 表示将所有的子 DOM 树去重合并。

设 R 为查询 Q 在全局 DOM 树 D 上的查询结果,R_i 为查询 Q 在分片后各个站点的子 DOM 树上的查询结果。有:

$$R=\sum_{i=1}^{n} R_i$$

其中,$\sum$ 表示将所有的子查询结果去重合并。

(2)索引模块:分片之后需要对每个站点上的子 DOM 树建立各自的索引,以提高查询的速度和效率。根据不同的查询,建立的索引包括父子对索引(PnameIndex)、属性值索引(AttributeValueIndex)、引用索引(ReferenceIndex)、全文索引(ElementTextIndex)等[6]。这些索引被存放在各个站点之上,同对应的子 DOM 树在一起。查询的时候则通过这些索引而不是直接遍历 DOM 树本身。每个索引结构都是基于 B^+ 树的,能够理解结构化置标树,提供全文索引,以及对元素和属性进行索引,对结构和内容进行索引,并提供按照元素和属性定位并检索元素的功能。图2表示了查询和索引的关系。

(3)消息传递和数据通讯模块:数据库本身提供的 DSVM(Distribute Shared Virtual Memory,分布式共享虚拟内存)机制可以将分布在多个站点上的数据库对用户屏蔽,在

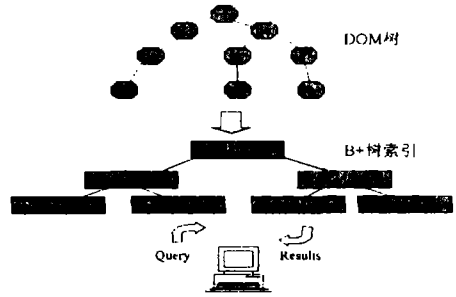

图2 索引与查询的关系

使用中可以不必考虑各个站点之间的通信。通过 DSVM 提供的一个虚拟的全局共享存储器,任何一个站点上的处理器都可以任意访问其中的数据,就好像访问本地数据库中的数据一样。DSVM 软件实现了所有的消息传递,使得各个独立的存储器变得对编程透明,简化了编程的复杂性。可以方便地进行并行化程序的设计和实现。对于分片的工作,由于数据量较大,数据结构比较复杂。采用 DSVM 机制直接将前端机上分片的结果写入远程的站点。

对于控制信息和查询结果的传输,由于数据量较少,数据类型简单,所以采用了更加灵活的 Socket 传输机制,以减少系统的通信开销。

(4)控制、查询优化和物理查询执行机模块:控制模块用于同步各个处理机的查询和结果合并。在前端机上利用多线程机制,对每个站点都单独生成一个线程用于发送指令和接受查询结果,在接收查询结果的同时调用结果合并模块进行合并。而不需要等待所有站点的查询全部结束。查询优化位于前端机之上,可以将用户的查询命令转换成为物理查询计划,传送到各个站点上的执行机模块。可以根据存放在前端机上,语法分析和分片时得到的模式信息来对物理查询计划进行优化。各个执行机模块则根据物理查询计划,通过索引来找到路径上的每个对象,最后将结果的 OID 传送回前置机上的结果合并模块。

(5)结果合并模块:在对象数据库中每个对象都有一个唯一的 OID 标示,该标示为对象的物理标示。在分片的过程中为了保证子 DOM 树的完整性,有很多节点对象在不同站点上是重复出现的。因此全局 DOM 树上的许多节点在整个数据库中具有不同的 OID。对于这种情况,我们对应全局 DOM 树,在分片过程中为每个对象生成了一个 GOID(Global OID,全局 OID),这是对象的逻辑标示,全局 DOM 树上每个对象的 GOID 都是唯一的。在每个节点上的查询结果返回给前端机以后,进行结果合并的时候根据 GOID 可以很容易地去掉重复的查询结果。

3. 测试结果分析

我们使用5台 PC 机组成一个同构的无共享 NOPC 环境进行测试。其 CPU 为 PIII 800MHz,硬盘为 20G,内存为 128M,10/100M 自适应网卡。操作系统均为 Solaris 8,交换区(swap)大小为 1.5G。通过一个高速 HUB(100Mbps)连接在一起,系统运行控制和结果合并都是用前置机来完成。

测试的数据集以德国 CWI 所提出的 XML Benchmark Project 为基础[5],按照固定 DTD 信息生成不同大小的 XML 文档,生成的文档从10M 到100M 不等,用于不同的测试。查询使用标准的 Xquery1.0语言,目前只提供在路径查询基础之上的几种简单的查询操作。所有的文档都符合下面图3中的

DTD。

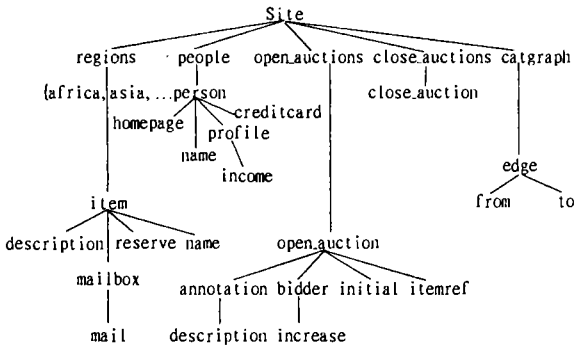

图3 文档所符合的 DTD

在10M 和50M 两种负载下进行测试，RPE 查询如下：

Query：site/regions/Australia/item/name/text()

(1)加速比的测试结果 图4给出了在两种负载下的查询响应时间，对每种情况，每个站点上的查询重复测试3次，取3次查询平均值的响应时间作为平均查询响应时间。

对于10M 的 XML 文档，经过解析后生成的 DOM 树一共有167000多个节点对象，每个站点上的数据库大小为200M。查询在一个站点上的平均响应时间为：0.774635S。在5个站点上的平均响应时间为：Query，0.254527S。对于50M 的 XML 文档，生成的 DOM 树上共有953000多个节点对象，每个站点上的数据库大小为500M。查询 Query 在一个站点上的平均响应时间为：11.47887S。在5个站点上的平均响应时间为：2.54649s。

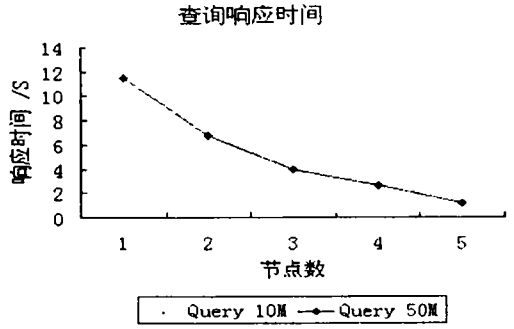

图4 两种负载下的查询响应时间

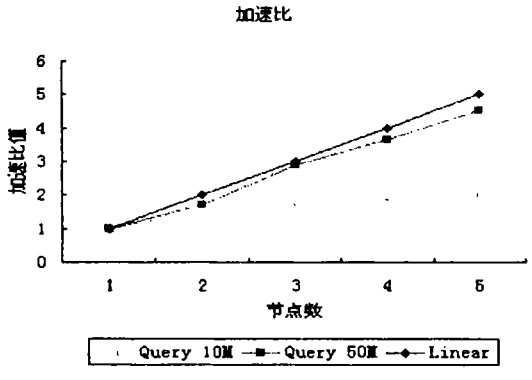

图5 两种负载下的性能加速比曲线

图5是两种负载下的性能加速比曲线。加速比随着站点数的增长而增长。轻负载情况下5个站点时加速比值为2.288008。而重负载的情况下，5个站点时的加速比值为4.507721。

由于目前的分片策略强调 DOM 树上路径实例的均衡，因此各个站点上的重复率比较高，在查询中系统的开销也随着站点个数的增加而变大。而且，在轻负载情况下。系统的通讯开销和查询的初始化工作占的比例相对较大。在重负载的情况下，查询性能有很大的改善。从两种负载下的查询响应时间和性能加速比曲线可以清楚地看到这一点。

(2)性能缩放比的测试结果 我们分别测试 Query 这个查询在10M/1站点，20M/2站点，…，50M/5站点情况下的查询响应时间。测试的结果如下表(单位 s)：

	1	2	3	4	5
Query	0.775	0.828	0.941	1.086	1.147

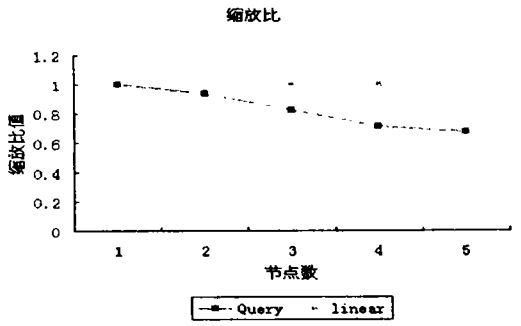

图6 性能缩放比曲线

图6是缩放比性能的测试曲线，由于测试用 XML 文档符合同样的 DTD，可以近似地认为任务量随着文档大小的增加而线性增加。

对于这个查询来说，在5个站点的时候响应时间分别比1个站点上响应时间下降了32.4%，性能的可扩展性是比较好的。

结论 本文主要介绍了 XML 数据库的并行 RPE 查询系统的体系结构，并对其性能做了简单的定量分析。从性能分析的结果可以看到并行查询是提高大 XML 文档查询响应速度的一种有效的方法。

参 考 文 献

1 Bray T, Paoli J, Sperberg-McQueen C M. Extensible Markup Language (XML) 1.0 (2nd Edition). W3C Recommendation, 6 October 2000. at: http://www.w3.org/ TR/ REC-xml

2 Robie J, LeHors A. Document Object Model level 2. W3Crecommendation. Available at: http://www.w3c.org/ TR/ 2000/REC-DOMLevel-2, May 2000

3 Bai G, Makinouchi A. WAJASHI: a distributed paged- object server for storage management of new generation databases. Proc. Of the Intl. Symposium on ADTI, Nara, Japan, Oct. 1994. 137～144

4 DeWitt D, et al. Parallel database systems: The future high performance database systems. Communication of ACM, 1986:75(1)

5 Schmidt A R, Waas F, Kersten M L. The XML Benchmark Project:[Report of CWI,2001]

6 吕建华，周巍等. XML 查询中 RPE 索引技术研究. 计算机科学，2001:28(增刊)