基于并行及异构数据库联合系统

姚领众 杨楠 邢艳辉
郭贵锁 宋瀚涛 （北京理工大学计算中心 100081）
李瑞红 （兰州炼油化工总厂第二中学 730060）

摘要：用 PowerBuilder 开发异构数据库的联合使用是一种好方法，然而它不支持数据库服务器间的并行工作。本文提出了在 PowerBuilder 应用程序与数据库接口之间通过构造“并行桥”的策略解决这一问题的思想，并由此设计实现了一个“异构数据库联合使用系统”。该系统将分布在网络不同节点上的 ORACLE、SYBASE 及 MS SQL SERVER 数据库服务器中的局部库集成为一个全局库并提供全局事务管理和并行查询处理的异构数据库系统。

关键词：异构数据库 PowerBuilder 并行策略 并行桥

1. 引言

近年来异构数据库得到了长足的发展。国外一些著名的数据库公司纷纷扩充了产品性能，以期支持对异种数据的访问。然而，包括著名的象 ORACLE、SYBASE 等在内的这些流行的关系数据库虽然程度不同的都支持对异种数据的访问，但这种支持是极其有限的，因为它们并不是异构数据库管理系统，也不是多数据库系统，比如要将分散在网络不同节点上的异种库集成为一个全局库，它们就难以做到。

我们选用 PowerBuilder 作为主要开发工具，设计并实现了一个由 ORACLE、SYBASE 及 MS SQL SERVER 构成的异构数据库联合使用系统 HDBUS（Heterogeneous DataBase United System）。本文在第 2 节中分析了 PowerBuilder 开发 HDBUS 的优势，同时也指出它的弱点，即 PowerBuilder 不能使多个数据库服务器并行工作。在第 3 节中提出了使多个数据库服务器并行工作的解决方案。第 4 节介绍了 HDBUS 设计原理。第 5 节是对上述工作的总结。

2. 选择 PowerBuilder 作为开发工具

PowerBuilder 是目前最好的数据库前端开发工具之一，它以其面向对象、具有可视化图形界面、符合客户/服务器模式而闻名。就开发异构数据库联合使用系统而言，最重要的一点还在于它支持多种关系数据库管理系统。PowerBuilder 是通过数据库专用接口或 ODBC 接口与后端的数据库服务器进行连接的[1]，如图 1 所示。

虽然 PowerBuilder 有如上所述的一些有利因素，但也存在以下不足：

(1)不支持异库表连接查询。所谓异库表连接指要连接的表不属于同一个库，而是分属于不同的库，如果这些库属同一种 DBMS，则是同构的，否则便是异构的。例如下面的 SQL 语句：

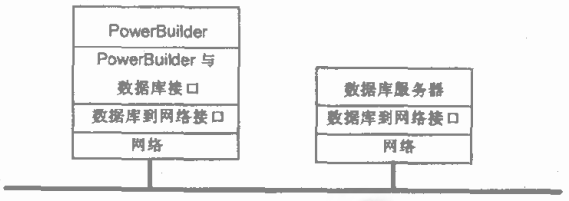

图 1 PowerBuilder 与数据库服务器的连接示意图

SELECT * FROM emp1, emp2 WHERE emp1.no = emp2.no ;

如果 emp1 和 emp2 不属于同一个库，则上述查询便是异库表连接查询。如果 emp1 属于 ORACLE，而 emp2 属于 SYBASE，则准确地说是异构异库表连接查询。

PowerBuilder 操作数据库的方式有两种，一种是通过创建数据窗口（Data Window），另一种是通过嵌入在程序中的 SQL 语句（类似嵌入在 Pro * c 中的 SQL 语句），不管用那种方式，都须首先与数据库建立连接。PowerBuilder 是通过事务对象与数据库通信的，事务对象是 PowerBuilder 和数据库间通信的桥梁。每个事务对象有十五个属性，其中 10 个用于 DBMS 所需的连接信息，如数据库厂商标识、注册到数据库的口令、服务器名等等，另外 5 个用于返回 SQL 语句运行或失败的信息，如数据库厂商提供的错误代码 SQLDBcode、错误信息 SQLEr-

rTextd 等。例如图 2 表示的是用于查询 ORACLE 库和 SYBASE 库的两个数据窗口通过两个事务对象连接到远端库的过程。

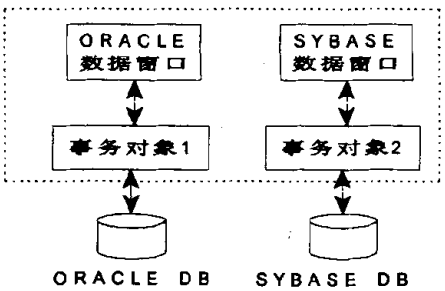

图 2　应用程序通过事务对象与数据库进行连接

虽然 PowerBuilder 应用程序通过创建多个事务对象可以连通多个库，但一个事务对象只能作为一个库的通信桥梁。只要分析一 PowerBuilder 的内嵌 SQL 语句的语法，就知道在一个 SQL 语句中是不可能操作分属在两个库中的对象的(比如表)。内嵌 SQL 语句的格式如下：

＜SQL 语句＞ USING [TransactionObject]

尖括号中的"SQL 语句"是标准的 SQL，方括号中是事务对象名。由于一个语句中只能有一个事务对象，在一个 SQL 语句中是不可能操作两个库的，更谈不上支持异库表连接。

(2) PowerBuilder 不支持多个数据库服务器并行工作。虽然可以通过创建多个事务对象使多个服务器同时处于连通状态，然而同一时刻只能使一个事务对象处于活动状态[1]，换句话说同一时刻只能有一个服务器处于工作状态。为了证明上属的论断，我们做了下述实验：

·实验环境：在一个以太网的 4 个节点上分别单独安装了 ORACLE、SYBASE、MS SQL SERVER 数据库服务器以及前端开发工具 PowerBuilder(操作系统平台为 Windows3.2)，3 个表 emp1、emp2 和 emp3 分别驻留于三个服务器上。现在编写一 PowerBuilder 应用程序，其中含有 3 个数据窗口 dw1、dw2 及 dw3，它们分别用于查询并显示 emp1、emp2、emp3。通过触发数据窗口上的查询事件调用所连的服务器为其查询。我们可以设想，如果三个服务器能够并行服务的话，则依次触发 3 个数据窗口查询事件后，3 个数据窗口必然交替显示查询结果。

·实验现象：触发 dw1 查询事件开始查询 ORACLE 库，数据窗口开始显示查询到的 emp1 记录。在查询未结束之前紧接着触发 dw2 查询事件开始查询 SYBASE 库，此时 dw2 开始显示 emp2 记录，说明 SYBASE 服务器正在查询，然而此时 ORACLE 数据窗口不再显示记录内容，说明已停止了查询。同理，在 SYBASE 库查询未结束之前触发 dw3 查询事件，结果与上述类似，MS SQL SERVER 开始查询，其余两个服务器皆停止查询。等到 MS SQL SERVER 查询结束，我们看到在 dw-2 数据窗口又开始显示查询结果，从查询的数据来看是 SYBASE 服务器原来中断了的查询的继续。SYBASE 查询结束，我们会看到同样的现象，ORACLE 服务器接着原来中断的查询做继续查询，直至结束。

·结果分析：表 1 是上述实验的查询时间记录

	ORACLE	SYBASE	MS SQL SERVER
查询开始	10:20:05	10:21:22	10:22:45
查询结束	10:28:58	10:26:51	10:24:47
表面用时	8 分 53 秒	5 分 29 秒	2 分 02 秒
实际用时	3 分 24 秒	3 分 27 秒	2 分 02 秒

上表中，表面用时 = 查询结束 - 查询开始

之所以称之为"表面用时"是因为它并不代表真正的查询用时，理由可以从上述的实验现象中分析出，在此不在赘述。对于"实际用时"，若用 obt、sbt、mbt 分别表示 ORACLE、SYBASE、MS SQL SERVER 的表面用时，用 ost、sst、mst 依次表示它们的实际用时，则

$$mst = mbt$$

$$sst = sbt - mbt = sbt - mst$$

$$ost = obt - sbt = obt - sst - mst$$

上面查询总用时为 8 分 53 秒，结果就是三个服务器单独查询用时的总和。也可用下面的平面坐标系形象地表示：

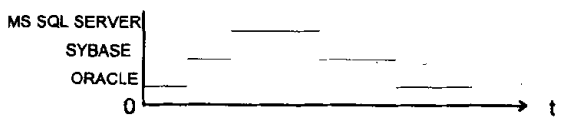

图 3

图中横轴 t 表示时间。从图中可以看出，三个服务器在查询时间上没有任何重迭，这也意味着查询过程完全是串行的。

3. 并行策略

尽管用 PowerBuilder 开发异构数据库系统有很多便利之处，然而它不能够调用多个数据库服务器并行工作是一重大障碍，因而必须设法解决。

既然在一个 PowerBuilder 应用程序中无法调用多个服务器使其并行工作，我们可以设想设计多个 PowerBuilder 应用程序，一个应用程序中有且仅有一个事务对象用以连通一个服务器，如果能使这多个应用程序并行工作(即多进程)，则有可能使多个服务器并行工作，进而使我们开发的异构数据库联合使用系统具有并行性。显而易见，要实现上述设想的一个必要条件是 PowerBuilder 所在的操作系统平台必须支持多进程，从目前支持 PowerBuilder 的操作系统平台来看，Windows 3.x、Windows 95、Windows NT 及 UNIX 等都支持多进程，因而为实现上述设想提供了可能。

为了验证上述思想的可行性，我们编制了三个应用程序，分别对应上述的三个数据库服务器，形式与前述实验的程序相似，可以认为是将其一分为三。

经过与第一个实验类似的多次实验(每个表的数据不变，查询条件不变)，我们得出的结论是用上述方法可实现服务器的并行工作。表 2 是其中的一次实验数据：

	ORACLE	SYBASE	MS SQL SERVER
查询开始	11:01:02	11:01:02	11:01:02
查询结束	11:04:58	11:05:05	11:03:43
用　时	3 分 56 秒	4 分 03 秒	2 分 41 秒

用坐标系表示如下：

图 4

总的查询用时从 11:01:02 至 11:05:10，为 4 分 3 秒。

比较表 1 和表 2 可以看到，就某个表单独的查询用时来看，表 2 中的用时长于表 1，但总的用时却从 8 分 53 秒降到 4 分 3 秒。分析其中的原因，是因为虽然 3 个服务器的查询是完全并行的，但作为客户端的应用程序所在的是一个单处理器、按时间片分配的多进程机制，因而从客户端向服务器发送命令和从服务器接收数据的过程实际上是串行的，即以上程序的执行时间由两部分组成：(1) 服务器的并行工作时间；(2) 客户端的串行执行时间。进一步的实验表明，从服务器查到的数据愈少，则并行性愈好。对于 UPDATE、DELETE 等不回送记录内容的语句，并行性将达到最好。

上述实验表明：在一个应用程序中 PowerBuilder 无法实现服务器间的并行工作，但通过多个并行运行的应用程序却可以实现。这一结果对于我们下面将要介绍的 HDBUS 的并行性设计是至关重要的，因为 HDBUS 的并行性是通过并行桥实现的，而并行桥简单地说就是这样的多个并行应用程序。

4. HDBUS 的总体描述

HDBUS 主要由 3 部分组成，如图 5 所示。图中箭头表示数据流。

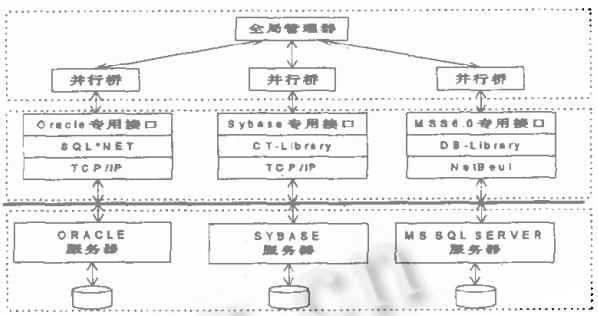

图 5　HDBUS 的系统结构

上述系统是一个客户服务器体现结构，可划分为三个层次，最高层为客户端的全局协调器，我们的设计工作主要集中在这一层。第二层是接口层，包括 PowerBuilder 与数据库的接口、数据库与网络的接口。第三层是服务器端的数据库服务器。

全局查询协调器直接为终端用户提供服务，从功能上看，它由全局事务管理器和并行桥两部分组成。全局事务管理器是在前台运行的独立程序，即前台进程。并行桥是后台独立运行的独立程序，即后台进程。系统中 3 个并行桥分别对应远端的 3 个数据库服务器，服务器正是通过并行桥的驱动获得了并行工作的能力。

5. 全局事务管理器的功能设计

主要由以下功能模块组成：

·全局数据字典。异构数据库中的全局库(Global

DataBase)是由分布在各网络节点上的本地库(Local DataBase)组成,从逻辑上讲存在全局库,从物理上看并不存在,因此数据字典就成了构造全局表的依据。它包括与全局库对应的各本地库的表名、列名、列的类型与长度、本地库在网上的分布情况以及一些与查询优化有关的信息等。数据字典是用 PowerBuilder 的内置数据库 Watcom 构造的[2]。

·全局数据字典输入器。对用户(或全局数据库管理员)来说,做全局查询前的第一步准备工作是构造全局库,即在全局数据字典中输入有关该库的信息。如果本地库不发生变化,则该工作只需做一次,然而如果本地库的结构发生了变化,则这个变化必须要反映到全局数据字典中来,即引起全局库的变化。因此,全局数据字典数据的录入应该说是非常重要的工作。

·全局查询语言分析器。由于 PowerBuilder 支持标准的 SQL 语言,因此用它作为全局查询语言是非常方便的,但必须做一定的修改。如前所述,全局库(在查询时更具体地说是全局表)在物理上是不存在的[3],因此对于一个由三个本地表 emp1、emp2、emp3 组成的全局表 emp,如果发出查询"SELECT * FROM emp ;",则 PowerBuilder 是不接受的,原因很简单,因为在任何一个节点上不存在 emp 表,PowerBuilder 在做语义、语法分析时便会指出不存在 emp 表的错误。因此,我们必须做一个能做全局语义、语法分析的分析器替代 PowerBuilder 的分析器。

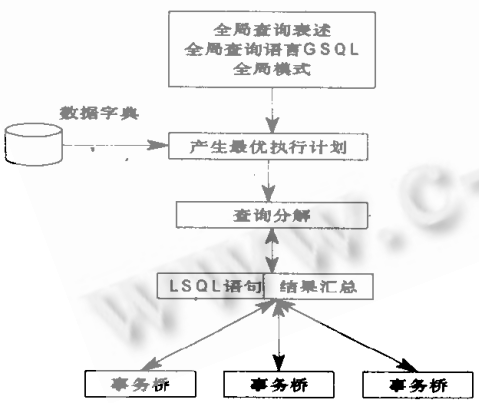

图 6　全局事务管理器的工作流程

·查询优化器。根据数据字典中的信息产生一个最优的查询方案。

·查询分解器。将表示全局查询的 GSQL(Global SQL)语句分解成本地查询的 LSQL(Local SQL)[6][7],然后交给相应的并行桥,再由并行桥启动具体的数据库服务器,执行查询。前述的全局查询语句"SELECT * FROM emp ;",便应分解成如下三个 LSQL 语句:

```
SELECT * FROM emp1 USING oratrans;
SELECT * FROM emp2 USING sybtrans;
SELECT * FROM emp3 USING msstrans;
```

3 条语句由并行桥分别交给数据库服务器 ORACLE、SYBASE、MS SQL SERVER 去执行。

全局事务管理器的工作流程如图 6 所示。

6. **并行桥**

并行桥是为克服 PowerBuilder 不支持服务器并行工作而设计的,其实质是三个各含有一个事务对象的独立程序,一个应用程序在同一时刻仅能使一个数据库服务器工作,而并行运行的多个应用程序使得通过它们驱动的多个服务器能够并行工作。在系统中并行桥作为后台进程运行,前台是全局事务管理器。前台与后台通过消息进行联络,工作过程如图 7。

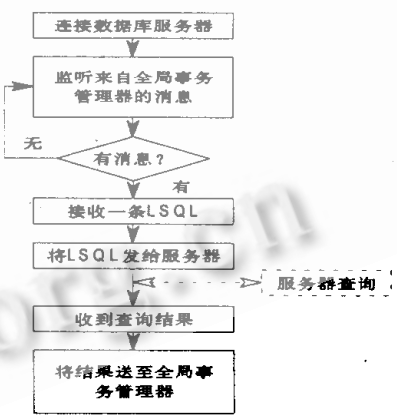

图 7　并行桥的工作流程

参考文献

[1] H. Kuantsae, Distributed Database Management System, AD-A124 921/8

[2] 姚领众,宋瀚涛,郭贵锁.通过网络互联实现校园网新旧 MIS 系统的联合使用.计算机应用研究,14 卷,4 期,1997

[3] Song Shiyu, Song Hantao and Liang Yunrong, The Technology of Data Conversion

(来稿时间:1997 年 5 月)