

文章编号: 1001-7011(2001)04-0049-04

PostgreSQL 分析器的研究

郭龙江, 李金宝

(黑龙江大学 计算机科学技术学院, 黑龙江 哈尔滨 150080)

摘要:介绍了对象关系数据库系统 PostgreSQL 的内部结构、查询处理的过程及 PostgreSQL 语法/词法分析器的实现,并通过一个 SQL 语句实例,分析了 PostgreSQL 语法/词法分析器的内部结构,最后给出 Query 树的详细数据结构图示。

关键词: PostgreSQL; 分析器; Parse 树; Query 树

中图分类号: TP393.03 **文献标识码:** A

1 PostgreSQL 简介

PostgreSQL 是一个面向公众的,开放源码的,对象关系数据库管理系统。它是从美国伯克利大学开发的 Postgres 软件包发展而来的。经过十几年的发展,PostgreSQL 是世界上可以获得的最先进的开放源码的数据库系统。它提供了多版本并行控制,支持几乎所有 SQL 构件并且可以获得广阔的开发语言绑定。PostgreSQL 的开发始于 1986 年,第一个演示性的系统在 1987 年推出,1988 年在 ACM-SIGMOD 大会上演示。1990 年 6 月发布了使用新规则系统的版本 2。1991 年,PostgreSQL 已经在许多领域中得到了应用,如财务数据分析系统、小行星跟踪数据库、医疗信息数据和地理信息系统等。1992 年末 PostgreSQL 成为 Sequoia2000 科学计算项目的首要数据管理器。到了 1996 年,正式改名为 PostgreSQL,它的开发重点也转到了一些有争议的特性和功能上。目前,欧美及日本一些大学的开发者通过互联网对 PostgreSQL 进行开发和维护。

2 PostgreSQL 的查询路径

当应用程序向 PostgreSQL 系统提交一个查询时,一般要经过五个阶段^[1]:

(1) 联接阶段:建立从应用程序到 Postgres 服务器的联接,应用程序向服务器发送查询然后接收服务器返回的结果。

(2) 分析阶段(parser stage):检查从应用程序(客户端)发送过来的查询,核对语法并创建一个查询树(query tree)。

(3) 重写阶段:以分析器输出的查询树,以及从 pg_rewrite 表里来的应用到查询树上的重写规则作为输入,传送给重写系统(重写系统是位于查询分析器和优化器之间的一个模块),并根据给出的规则体进行转换,然后创建零个或者多个查询树作为结果。例如,当一个查询访问一个视图时,重写系统改写用户的查询,使之成为一个访问带有视图定义的基本表的查询。查询树经过重写后还是一个查询树,但增加了一些扩展信息。经过重写的查询树被传送给规划器/优化器(planner/optimizer)。

(4) 优化阶段:规划器/优化器(planner/optimizer)接收(可能是改写过的)查询树,然后创建一个查询规则(queryplan)。规划器/优化器首先创建所有得出相同结果的可能的路径(paths)。例如,如果待扫描

收稿日期: 2000-12-05

基金项目: 黑龙江大学校科研基金项目资助

作者简介: 郭龙江(1973-),男,河北丰宁人,黑龙江大学计算机科学技术学院讲师,硕士,主要研究方向:数据库,并行计算,计算机应用

的关系上有一个索引,并且 where 条件从句中的某一属性恰好是索引关键字,那么扫描的路径就有两个,一个可能是简单的顺序查找,而另一个可能就是使用索引查找,下一步是计算出每个索引执行的开销,并且选择和返回开销最少的那个。另外,如果要对两个关系进行 join,那么 join 的方式有多种(nested iteration join, merge sort join, hash join),规划器/优化器负责选择出开销最小的 join 算法。查询规划被传送给执行器(executor)。

(5) 执行阶段: 执行器递归地扫描规划树(plan tree),并且要在这个过程中检索规划所代表的记录。执行器在对关系进行扫描时使用存储系统(storage system)进行排序(sort)和联接(join),计算条件(qualification),并且最终返回生成的记录。

3 前端和后端的连接

PostgreSQL 采用“每用户——进程”的 client/server 模型^[2]。在这种模型里一个客户端进程只与一个服务器进程联接。因为不知道具体要建立多少个联接,所以不得不利用一个主进程在每次联接请求时派生出一个新的服务器进程来。这个主程叫做 postmaster,它监听着一个特定的 TCP/IP 端口(缺省为 5432)等待联接。每当检测到一个联接请求时,postmaster 进程派生出一个新的叫 postgres 的服务器进程。postgres 进程相互之间使用信号灯和共享内存进行通讯,以确保在并行的数据访问过程中的数据完整性。图 1 显示了主进程 postmaster,服务器进程 postgres 和客户端应用之间的相互关系。

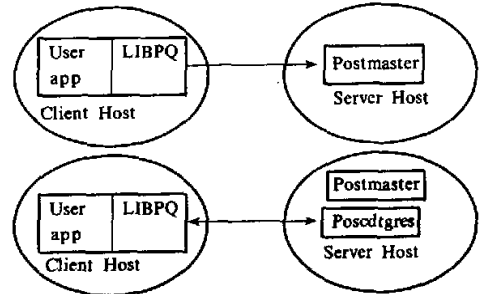


图 1 连接建立过程

一旦建立起来一个联接,客户端进程就可以向后端服务器进程发送查询了。查询是通过纯文本传输的,也就是说在前端不做任何分析处理。服务器分析查询,创建执行规划,执行该规划并且通过已经建立起来的联接把检索出来的记录返回给客户端。

4 PostgreSQL 的语法分析器

PostgreSQL 的分析器包括词法分析器和语法分析器。若将 PostgreSQL 源码安装在/usr/src/pgsql/postgresql/src,可用一个环境变量 PGROOT 代替上述目录。PostgreSQL 的分析器定义在 SPGROOT/backend/parser 目录下。其工作是分析用户前端发送来的 SQL 语句或 PostgreSQL 命令,建立 Query 树。此目录所有函数集成为一个函数 List*parser(char*str, Oid*typer, int nargs)。它是分析器的入口函数,放在 parser.c 中。对用户命令或查询语句的分析从此函数开始。因此,parser 函数实际上就是 PostgreSQL 的分析器。词法分析器定义在 scan.l 转换分解成 tokens,scan.l 使用 Unix/Linux 工具 Lex 制作。用 lex 可把 scan.l 转换成 scan.c。PostgreSQL 的语法分析器定义在 gram.c,其中定义了一个重要的函数 yyparse()。yyparse()调用 scan.c 中的 yylex()分析用户的查询,然后建立 Query 树。分析器(parser 函数)必须检查以纯 ASCII 文本方式的查询字符串的语法。如果语法正确,则创建一个 Parse 树回,否则,返回一个错误。词法分析器(yylex 函数)在文件 scan.l 里定义,负责识别标识符,SQL 关键字等。对于发现的每个关键字或者标识符,yylex()都会生成一个记号并且通过变量 yylval 传递给语法分析器。yylvar 是由 yacc 把 gram.y 转换成 gram.c 后自动定义的 YYSTYPE 类型的变量。YYSTYPE 是在 gram.y 中定义的联合体。经过 yacc 转换后,YYSTYPE 定义在 parse.h 中。语法分析器(yyparse 函数)在文件 gram.y 里包含了一套语法规则和触发规则时执行的动作。动作代码(实际上是 C 代码)用于建立分析树(parse tree)。PostgreSQL 分析器内部调用顺序如图 2 所示。

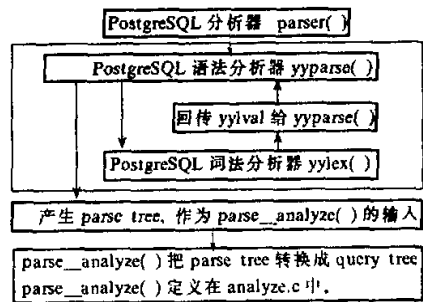


图 2 PostgreSQL 分析器内部调用顺序

为了更好地理解 PostgreSQL 分析器, 我们用一个查询例子作为分析器的输入, 看一看 PostgreSQL 分析器的输出。

假设 SUPPLIER 是存储供应商编号 (SNO), 名称 (SNAME) 和城市 (CITY) 的表, PART 是存储部件的编号 (PNO), 名称 (PNAME) 和价格 (PRICE) 的表, SELLS 是供应商和部件数据库:

| SUPPLIER | | | | SELLS | | | PART | | | |
|----------|-----|-------|--------|-------|-----|-----|------|-------|-------|-------|
| | SNO | SNAME | CITY | | SNO | PNO | | PNO | PNAME | PRICE |
| | 1 | Smith | London | 1 | 1 | 1 | | | | |
| | | | | 1 | 2 | | 1 | Screw | 10 | |
| | 2 | Jones | Paris | 2 | 4 | | 2 | Nut | 8 | |
| | | | | 3 | 1 | | | | | |
| | 3 | Adams | Vienna | 3 | 3 | | 3 | Bolt | 15 | |
| | | | | 4 | 2 | | | | | |
| | 4 | Blaks | Rome | 4 | 3 | | 4 | Cam | 25 | |
| | | | | 4 | 4 | | | | | |

如果我们想知道那些供应商号大于 2 且销售超过一个部件的供应商, 使用下面查询:

```
SELECT S. SNO, S. SNAME
FROM SUPPLIER S, SELLS SE
WHERE S. SNO > 2 AND S. SNO = SE. SNO
```

输入上述查询字符串, 经过 `yyparse()` 函数将构建 Parse 树, 由于树很大, 本文只给出分析器的最终结果 Query 树 (参见附录)。

5 结论

PostgreSQL 系统中的设计思想、实现算法有较高的学术研讨价值, 对于我们研究系统软件有重要的学术参考价值, 在今后的 PostgreSQL 研究中, 我们还将进一步研究 planer/optimizer, executor 及 storage management system。

参考文献:

- [1] The PostgreSQL Global Development Group. PostgreSQL Developer's guide[S]. 1998.
- [2] STERFAN SIMKOVICS. Enhancement of the ANST SQL Implementation of PostgreSQL[D]. Department of Information Systems, Vienna University of Technology, November 29, 1998.

Study of the PostgreSQL's parser

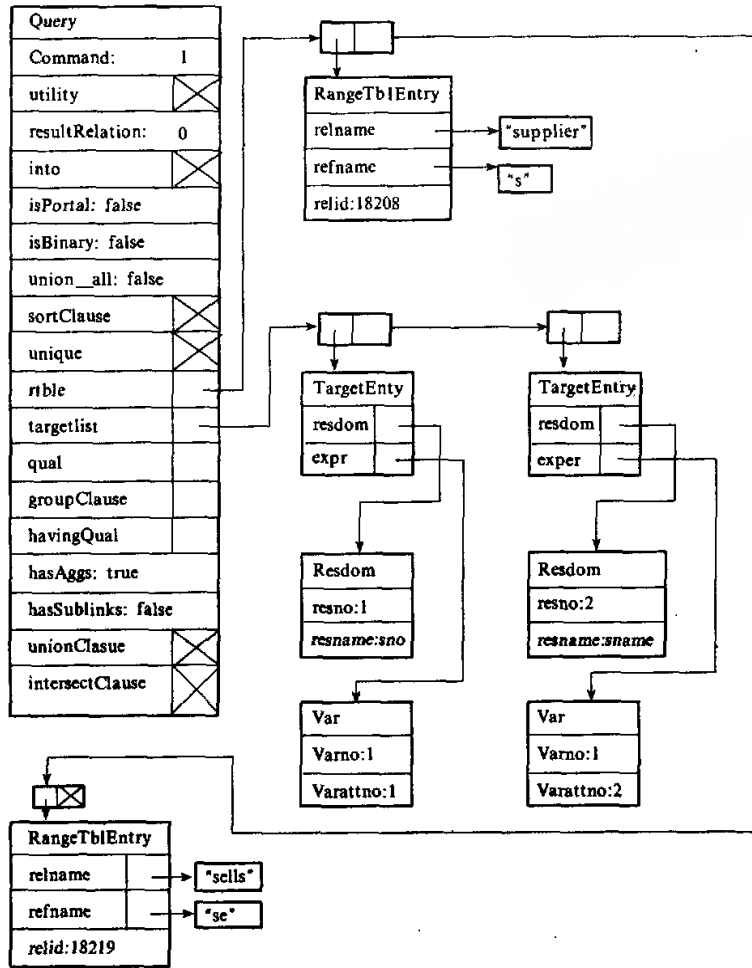
GUO Long-jiang, LI Jin-bao

(College of Computer Science and Technology, Heilongjiang University, Harbin 150080, China)

Abstract: The inner structure of the PostgreSQL-Object relation database system and query process and implementation of the PostgreSQL's parser are proposed. The authors also gave a example and analyzed the inner structure of the PostgreSQL's parser. Finally, they showed the chart of the Query tree which is generated by the PostgreSQL's parser.

Key words: PostgreSQL; parser; Parse tree; Query tree

附录: Query 树



(上接第 48 页)

基于多重加权树的并行数据查询处理方法


李建中

(黑龙江大学 计算机科学技术学院, 黑龙江 哈尔滨 150080)

摘要: 研究了并行数据库系统的优化问题, 提出了一个基于多重加权树的查询处理方法。这种方法由基于多重加权树的并行查询计划模型、并行查询计划的代价模型和一个并行查询优化器等三部分组成。并行查询计划模型首次模拟了所有关系操作、查询执行的三种并行性、查询操作的处理器和存储器分配、按流水线方式并行执行的操作间的存储器分配以及数据重分布。查询代价模型充分考虑了按流水线方式并行执行的操作间的等待时间, 并给出了一种自底向上的查询代价计算方法。查询处理器以最经常使用的选择-投影-连接查询为背景。查询优化器使用了一组确定启发式规则为数据操作分配处理器, 并且很好地利用了存储器资源。此外, 查询优化器支持多种连接操作实现算法, 能够根据不同情况为查询中每个连接操作选择优化的连接操作实现算法。该查询处理方法已经用于作者自行设计的并行数据库管理系统, 取得了良好的效果。

关键词: 并行数据库; 并行查询计划模型; 并行查询处理

PostgreSQL分析器的研究

作者: 郭龙江, 李金宝
作者单位: 黑龙江大学计算机科学技术学院,
刊名: 黑龙江大学自然科学学报 
英文刊名: JOURNAL OF NATURAL SCIENCE OF HEILONGJIANG UNIVERSITY
年, 卷(期): 2001, 18(4)
引用次数: 0次

参考文献(2条)

1. The PostgreSQL Global Development Group PostgreSQL Developer's guide 1998
2. STERFAN SIMKOVICS Enhancement of the ANST SQL Implementation of PostgreSQL 1998

相似文献(2条)

1. 学位论文 王静 数据库安全增强系统模型的研究 2006

随着我国信息化的不断发展,数据库得到日益广泛的应用。同时,开放复杂的网络环境,使数据库中的数据受到了越来越多攻击威胁,所以,数据库安全技术也就成为了计算机技术研究领域中一个重要的分支。目前,我国数据库安全技术研究主要有安全数据库管理系统和数据库安全增强系统两个方向,本文主要从数据库安全增强系统的角度出发,参考经典的TCSEC监视器模型,在数据库客户端和服务器端之间建立一个代理系统,来完成数据库的安全通信、SQL分析和访问控制等功能。本文主要在以下几个方面开展了研究:一、从TCSEC准则出发,根据数据库系统的安全需求,构建了一种数据库安全增强系统模型,从结构上分为客户端接口和安全数据库代理两部分。二、进行了系统的实现方案的设计,应用标准ODBC规范实现客户端接口,应用SSL安全协议,保证客户端接口与安全数据库代理的通信安全。三、参考PostgreSQL分析器,提出了基于LEX和YACC的SQL语句分析算法,应用该算法建立SQL查询树,并应用了一种查询树的优化思想——下推思想。四、为数据库主、客体分配安全级,并定义了安全级操作和访问控制规则;引入BLP模型,通过对数据字典、DDL、DML的改造,提出了一套安全增强方案。在研究过程中遇到了很多问题,在以后的工作中我们将解决这些问题,并逐步探讨解决数据库安全的有效途径。

2. 学位论文 虞俊 GEO-RBAC中角色层次关系模型的改进与实现 2008

安全空间数据库是当前信息安全研究的一个重要分支,具有广泛应用前景。该领域的研究具有较强的保密性,信息技术发达国家对我国一直施行尖端安全产品禁止输出策略,数据库安全产品亦在其列,因此,研究和开发自主安全数据库产品是进行信息保护的一个重要手段。基于角色的访问控制RBAC(Role-Based Access Control)模型的出现方便了系统权限管理。同时,随着基于角色访问控制应用领域的扩大,在基于角色的访问控制模型中需要考虑空间特性。本文以开源数据库PostgreSQL为平台,研究带有空间特性的角色访问控制模型(GEO-RBAC)中的角色层次关系特性。角色层次通过角色与角色之间的关系,定义与权限继承和角色激活相关的语义。本文首先详细分析了现有的角色继承机制的不足之处,对角色继承语义作了深入分析,将原有的角色继承语义作了进一步细分,并且在现有的带空间特性的基于角色访问控制模型GEO-RBAC中引入混合角色层次关系。本文主要工作在如下四个方面:第一,引入了“用户可激活集”这一概念,用以界定分配了层次中某个角色的用户的访问能力。以形式化的方式,论述了如何在带空间特性的混合模型中判断可激活集,并阐明了如何通过限制与可激活集相关联的角色,实现对用户权限分配的控制。第二,引入了一组推导规则。运用这些推理规则,我们可以推断出两个无直接关联的角色之间的层次关系,同时证明了这套推理规则是正确且完备的。第三,从角色添加、角色删除以及角色划分等方面,探讨了层次转换问题。第四,在完成了以上三个方面的理论研究的基础上,本文以PostgreSQL数据库为平台,在PostgreSQL的分析器模块中插入带空间特性的基于角色的访问控制模型,实现对数据库的安全访问控制和对系统中角色层次的管理。本文对GEO-RBAC模型中的角色层次关系作了详细分析的基础上对原有的角色层次关系作出合理的改进,并且在PostgreSQL数据库平台上实现了带空间特性的访问控制模型及角色层次管理工具。

引证文献(2条)

1. 付国良 基于企业模型的制造业信息系统动态重构研究[学位论文]硕士 2004
2. 叶志坤 国产数据库应用增值平台研究与开发[学位论文]硕士 2004

本文链接: http://d.g.wanfangdata.com.cn/Periodical_hljdxzrkxxb200104011.aspx

下载时间: 2009年12月29日