

TPC-H 自动测试工具 TPCHDriver 的研究

王 良, 栗跃鹏, 杨 政

(中国人民大学信息学院计算机科学与技术系, 北京 100872)

摘 要: TPC-H 基准测试可以比较不同数据库系统之间的查询性能差别, 它的测试结果已经成为评价 DBMS 和服务器的一个重要标准。该文开发了自动化的 TPC-H 测试工具 TPCHDriver, 该工具应用面向对象设计、Java 语言开发, 图形用户界面, 实现了从数据装载到 Power 测试和 Throughput 测试、生成测试报告整个过程的自动化。简述了 TPC-H 基准的测试模型、TPCHDriver 程序的结构和主要功能, 阐述了 TPCHDriver 的关键实现技术。

关键词: TPC-H 基准; 性能测试; 查询响应时间

Research on TPCHDriver of Automatic Test Tool in TPC-H

WANG Liang, LI Yu-peng, YANG Zheng

(Department of Computer Science and Technology, School of Information, Renmin University of China, Beijing 100872)

【Abstract】 The TPC-H benchmark test is applied to contrast difference of select performance between database systems. Results of TPC-H test have become important standard evaluating performance of DBMSs and database servers. From the point of view of a user, a TPCHDriver of automatic software test tools are developed, designed with object-oriented modeling, programmed in Java. Test process from a load test to a power test and a throughput test, building a test result report are automatically implemented. This paper depicts test model of TPC-H benchmark, TPCHDriver's structure and primary functions, emphasizes the pivotal technology of implementing a TPCHDriver.

【Key words】 TPC-H benchmark; performance test; selecting respond time

1 概述

TPC-H(商业智能计算测试)由TPC(Transaction Processing Performance Council)发布,是在TPC-D上发展起来并取代了TPC-D^[1]。它模拟决策支持系统中的数据库操作,测试数据库系统复杂查询的响应时间,以每小时执行的查询数(TPC-H QphH@Siz)作为度量指标。

在 TPC-H 模型中,定义了 8 张表,22 个复杂查询(SELECT)和 2 个更新(带有 INSERT 和 DELETE 的程序段)的操作。被测试数据库的数据量从 1 GB~10 000 GB,有 8 个级别供用户选择。测试时,将 22 个查询随机组成查询流,2 个更新操作组成一个更新流,查询流和更新流并发执行数据库访问,查询流数目随数据量增加而增加。Power 测试只有一个查询流和一个更新流,Throughput 测试有多个查询流和一个更新流。

2 TPC-H 实现分析

TPC-H 基准测试可以比较不同数据库系统之间的查询性能差别,测试结果已经成为评价 DBMS 和服务器的一个重要标准。为了提高测试效率和简化测试过程,开发了自动测试工具 TPCHDriver,用户利用这个工具可以自动地完成 TPC-H 所要求的全部测试。

2.1 测试模型

TPC-H 以决策支持系统为模型,总体框架见图 1,测试模型见图 2。为了保证每次查询的结果的不同,SELECT 语句带有变量。为了防止缓冲功能被放大,22 个 SELECT 语句随机编排成一个查询流后,再逐个执行。TPC-H 提供了查询流生成工具 QGEN 的源代码,供测试者生成查询流。

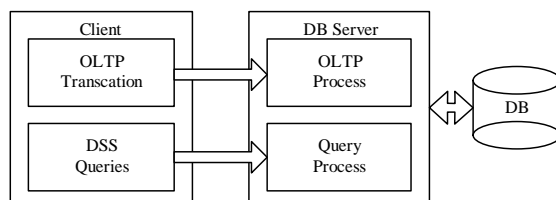


图 1 TPC-H 总体框架

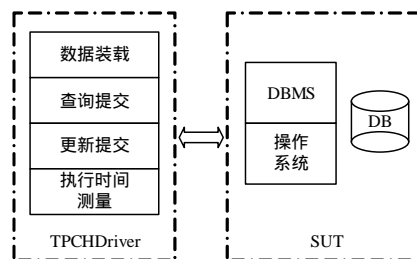


图 2 TPC-H 测试模型

TPC-H 测试的一个重要思想就是独立于 SUT(System Under Test)而设计一个驱动(DRIVER),负责查询流与更新流的提交。DRIVER 是一个数据库的客户端,向数据库服务器

基金项目: 教育部“211”工程基金资助项目“中国人民大学信息科学技术与信息管理学院学科建设”;中国人民大学科学研究基金资助项目(734)

作者简介: 王 良(1963-),男,副教授,主研方向:软件工程,数据库;栗跃鹏,硕士研究生;杨 政,硕士

收稿日期: 2008-11-14 **E-mail:** wangliang@ruc.edu.cn

提交并发的 SELECT, INSERT 和 DELETE 操作。SUT 包括了硬件设备上的服务器和网络设备,软件上有操作系统、DBMS 以及连接 DRIVER 和服务器的系列软件。

2.2 度量方法

完整的 TPC-H 测试由负载测试(load test)加 2 轮性能测试组成,性能测试分为 Run1 和 Run2,每个 Run 又有 1 个 Power 测试和 1 个 Throughput 测试。

TPC-H 度量的基础是 SELECT 语句和更新流执行时间,由执行时间计算出 Power@Size 和 Throughput@Size,再由 Power@Size 和 Throughput@Size 两者的几何平均值求得 QphH@Size。QphH@Size 是 TPC-H 的代表性指标,TPC-H 要求每 Run 都计算出 QphH@Size,以较低值作为测试结果。TPC-H 测试结果之间的可比性指标是性能价格比 Price-per-QphH@Size。

3 TPCDriver 设计

3.1 功能

为了实现自动的 TPC-H 测试,TPCHDriver 需要具有配置参数、装载数据、提交数据库事务和测量响应时间的功能。

TPC-H 的数据库定义和数据操作遵循 SQL92 标准^[2],而有些 DBSM 并不完全支持 SQL92,这要求在保证语义不变的前提下,TPCHDriver 具有转换数据类型和修改 SQL 语句的功能。测试工具需要建立被测试数据库中的 8 张表,装载用户指定的数量的数据。在重复测试时,还要有清理表中数据的功能。

TPCHDriver 可以按用户的要求建立查询流,并为查询流编号,同时控制查询流数不低于 TPC-H 规范的最低要求,多个查询流和更新流同时向被测数据库发出访问请求,DBMS 并发执行这些请求。每个流中的每个查询和更新操作的执行时间被测量和记录下来,以 XML 格式存在文件中。在测试完成时,TPCHDriver 能够自动计算测量指标 Power@Size, Throughput@Size, QphH@Size 和 Price-per-QphH@Size。并按按照 TPC-H 规范附录 E 的格式生成测试报告。

用户可以通过 GUI 界面配置参数,如 SF 或数据量、查询流个数、数据库服务器地址、被测数据库名称等,以文字和图形方式动态显示测试进度。用户还可以选择单步测试和自动测试。前者以向导方式引导用户进行测试;后者用户只需要设置完测试参数,TPCHDriver 自动完成全部测试,中间不需要用人工干预。

3.2 结构

TPCHDriver 结构上分为 6 部分,按面向对象设计,类及类之间的关系见图 3,采用 Java 编程。在类的设计上力求功能的独立性和可扩展性,为程序的升级和扩展做准备。

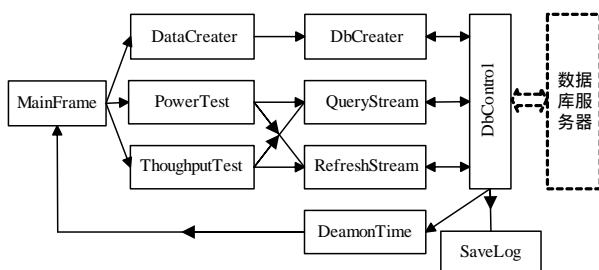


图 3 TPCDriver 类结构

具体类结构如下:

(1)MainFrame 类:承担用户与 TPCHDriver 的交互,它包括标签、按钮、文本区等及响应事件。操作界面见图 4,用户可以在此设置参数如 DBMS 的类型、数据库的连接信息、数据量的大小(SF)。界面中可以动态显示测试进度信息和测试结果信息。

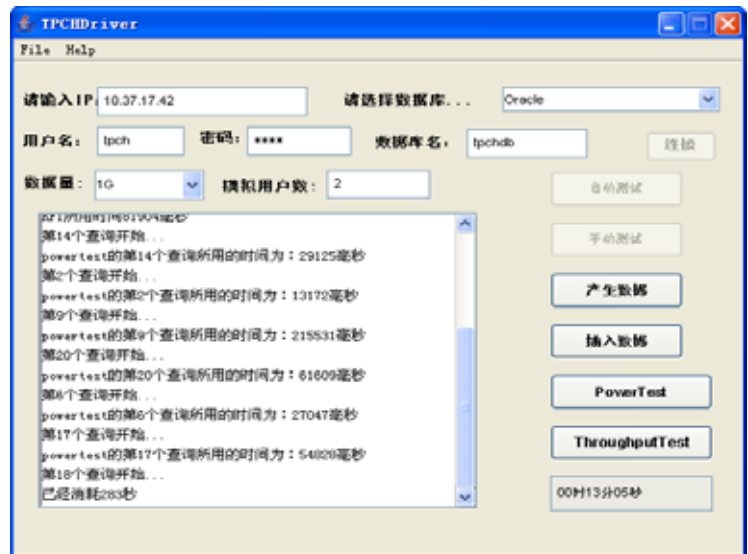


图 4 TPCDriver 界面图

(2)DataCreator 类:根据用户指定的数据量级别 SF,检查硬盘空间是否满足装载数据的需求,然后调用 DBGEN 产生数据,加载数据。

(3)DbControl 类:实现与数据库相关的操作,如数据库连接、DML 和 DDL 操作等。TPCHDriver 通过 JDBC 连接 DBMS。

(4)DbCreator 类:初始化数据库,包括建表和装载数据。TPC-H 标准建议每做一次测试都要清空数据库,重新装载数据。

(5)QueryStream 类:调用 QGEN 产生查询流,查询流中包含随机编排顺序的 22 个 SELECT 语句。

(6)RefreshStream 类:调用 QGEN 产生更新流,流中的 INSERT 语句和 DELETE 语句嵌在循环程序中。

(7)PowerTest 类:模拟单个查询流和单个更新流,实现 Power 测试。

(8)ThroughputTest 类:模拟 n 个并发用户,即 n 个查询流和 n 个更新流,完成 Throughput 测试。

(9)AutoTest 类:调用 DataCreator, DbCreator, QueryStream 和 RefreshStream 自动完成整个测试流程,包括装载测试、Run1 和 Run2 测试。

(10)DeamonTime 类:执行后台操作,测量和记录各操作执行的时间,并显示各个阶段消耗的时间。

(11)SaveLog 类:以 XML 文件形式记录测量的查询操作和更新操作消耗的时间。

3.3 DBGEN 和 QGEN 处理

DBGEN 和 QGEN 作为 TPC-H 规范的附件,是 C 源程序,缺省运行在 Unix 操作系统上,访问 pad 数据库。它们可以很容易地移植到提供 ANSI C 编译器的机器上。DBGEN 和 QGEN 在使用前需要以 makefile.suite 为模板,创建一个 makefile 文件,并填写 makfile 文件中的选项,如编译器 CC、

数据库 DATABASE、机器平台 MACHINE、负载类型 WORKLOAD、目标文件 OBJ、可执行文件 EXE 和链接库 LIBS 等。

DBGEN 是用于产生测试库中的数据，生成的数据存储在文本文件中，DataReady 类读取文本文件中的数据，拼装成 INSERT 语句，向表中插入数据。DBGEN 可以带参数，如 -s 参数调整数据规模、-C 参数控制产生数据的次数、-S 参数选择执行第几次产生数据、-U 参数调整更新流的数目。例如 dbgen -s100 -S 1 -C 100(产生第 1 个 1 GB 的文件)，dbgen -s 100 -S 2 -C 100(产生第 2 个 1 GB 的文件)，dbgen -s 100 -U 5(产生 SF 为 100 GB 的 5 用户的更新流)。

QGEN 用于产生 SELECT 语句，生成的 SELECT 语句保存在文本文件。TranQuery 类读取文本文件中的 SELECT 语句，组成查询流。

QGEN 相当于一个文件过滤器，将 \$DSS_QUERY/<name>.sql 模块中的带有“:”的变量替换为自动产生的参数。

QGEN 也有很多参数，但一般情况下，不需要使用这些参数。

4 TPCDriver 应用

TPC-H 的 22 个查询在大数据量情况时耗时巨大，如果被测 DBMS 的查询计划不恰当，耗时成倍增加，甚至得不到查询结果^[3-4]。

TPCHDriver 记录的测量数据为分析复杂查询的表现提供了辅助手段。

图 5 是 3 个测试案例，即在 1 GB 数据量下，3 个版本的 DBMS 操作时间的对比，RF1 和 RF2 为插入和删除操作， $Q_i(i=1,2,\dots,22)$ 为 SELECT 操作。从图 5 中可以发现 Q1, Q9, Q18 和 Q21 用时较多，约占总 SELECT 时间的 38.29%~46.37%，用时多的主要原因是这些查询全表扫描的次数多，有的查询中还有大量的聚集函数操作。

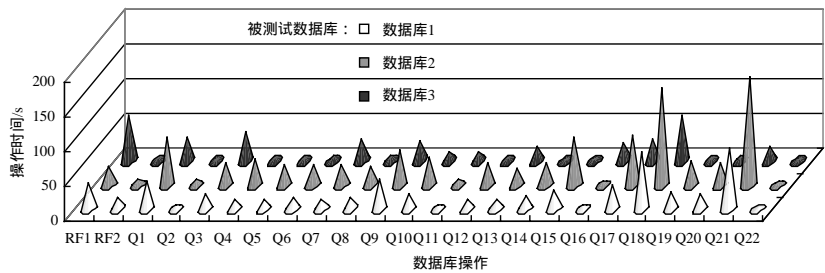


图 5 3 个 DBMS 的 TPC-H 查询执行时间比较

TPCHdriver 已经应用到国产的金仓数据库的 TPC-H 测试中，也对部分主流商用 DBMS 进行了测试，运行效果良好。

5 结束语

本文的研究目的就是开发一个自动化的测试工具，测试者不需要深入地研究 TPC-H 测试规范，不需要开发测试程序，测试过程也无需用户干预。

从初步的研究成果看，TPC-H 的 22 个 SELECT 语句对测试结果的贡献率差异较大，反映了 DBMS 的查询能力的不同方面。在后续研究中，将深入对 SELECT 语句的分类或增加新 SELECT 语句，使 TPCHDriver 能更加细致测试 DBMS 的查询能力，为测试者提供更多的查询性能分析手段。

参考文献

- [1] Transaction Processing Performance Council(TPC). TPC Benchmark TM H(Decision Support) Standard Specification Revision 2.3.0[EB/OL]. (2005-08-11). <http://www.tpc.org/tpch/default.asp>.
- [2] Hector G M, Ullman J D, Widom J. 数据库系统实现[M]. 北京: 机械工业出版社, 2002.
- [3] Trancoso P, Adamou C, Vandierendonck H. Reducing TPC-H Benchmarking Time[C]//Proc. of the 10th Panhellenic Conference on Informatics. Volos, Greece: [s. n.], 2005: 641-650.
- [4] Vandierendonck H, Trancoso P. Building and Validating a Reduced TPC-H Benchmark[C]//Proc. of MASCOTS'06. Monterey, California, USA: [s. n.], 2006.

(上接第 14 页)

4 实验

为了验证算法的可行性，本文在连通图问题和 gcd 函数问题上测试了 GILP 算法的性能。考虑到遗传算法的收敛性，实验中的 GILP 内层算法中采用了精英保留策略。规则性能 F 评估中的 β 设为 0.5。其他参数设置如下：

种群规模	交叉概率	变异概率	最大演化代数
300	0.60	0.05	40

有向图数据挖掘涉及 2 个谓词：目标谓词 $\text{can_reach}(X, Y)$ 表示 X 可达 Y ，背景知识谓词 $\text{linked_to}(X, Y)$ 表示 X 直接连接到 Y 。GILP 从连通图数据中挖掘出如下的一阶规则：

$\text{can_reach}(X_1, X_2) :- \text{linked_to}(X_1, X_2)$
 $\text{can_reach}(X_1, X_2) :- \text{linked_to}(X_1, X_3), \text{can_reach}(X_3, X_2)$

这 2 条一阶规则共同描述了有向图中任意点 X_1 可连通到另一点 X_2 的目标概念 $\text{can_reach}(X_1, X_2)$ 。上述第 2 条规则的含义是，若点 X_1 直接连接到点 X_3 ，且点 X_3 可连通到点 X_2 ，则点 X_1 可连通到点 X_2 。这种递归规则是很难用决策树或其他命题级表示方法表达的。

gcd 问题是从 2 个整数最大公约数的数据中挖掘出最大公约数的一般函数定义。问题涉及 2 个谓词：目标谓词 $\text{gcd}(X,$

$Y, Z)$ ，表示 Z 是 X 和 Y 的最大公约数；背景知识谓词 $\text{plus}(X, Y, Z)$ ，表示 Z 等于 X 加 Y 。从 1~20 这 20 个整数两两之间最大公约数的 400 个例子中，利用相应的背景知识，挖掘出下列规则，该规则集正确描述了 gcd 函数关系：

$\text{gcd}(X_1, X_1, X_1) :-$
 $\text{gcd}(X_1, X_2, X_3) :- \text{plus}(X_2, X_4, X_1), \text{gcd}(X_2, X_1, X_3)$
 $\text{gcd}(X_1, X_2, X_3) :- \text{plus}(X_1, X_4, X_2), \text{gcd}(X_1, X_4, X_3)$

5 结束语

基于一阶谓词逻辑的一阶规则挖掘是数据挖掘技术中的新方法。在研究中认识到，一阶规则挖掘实际上是目标谓词和背景知识谓词构成的各种原子的组合优化问题。本文提出了一阶规则的变长位串编码和 GILP。在连通图问题和 gcd 问题上验证了新方法的可行性。

参考文献

- [1] Dzeroski S, Lavrac N. Relational Data Mining[M]. Berlin, Germany: Springer-Verlag, 2001: 105-139.
- [2] Tom M. Machine Learning[M]. Singapore: The McGraw-Hill Companies, Inc., 1997: 65-66.
- [3] Divina F. Evolutionary Concept Learning in First Order Logic: An Overview[J]. AI Communications, 2006, 19(1): 13-33.