

文章编号:1000-5641(2014)05-0340-11

数据库性能测试可视化工具 VisualDBBench 及面向内存数据库的应用

李 梁, 吴 刚, 刘辉林, 王国仁

(东北大学 信息科学与工程学院, 沈阳 110819)

摘要: 从开发自动化数据库基准测试工具的角度, 深入研究了 TPC 组织发布的 TPC-C 测试标准和 TPC-H 测试标准. 在此基础上, 给出了测试模型, 介绍了所开发的自动化测试工具 VisualDBBench 的架构和主要类的功能. 并且针对内存数据库进行测试, 验证了内存数据库的性能优势.

关键词: 基准测试; TPC-C; TPC-H; 内存数据库; VisualDBBench

中图分类号: TP392 **文献标识码:** A **DOI:**10.3969/j.issn.1000-5641.2014.05.031

Design and implementation of a database benchmark visualization tool VisualDBBench and application in main-memory databases

LI Liang, WU Gang, Liu Hui-lin, WANG Guo-ren

(College of Information Science and Engineering, Northeastern University, Shenyang 110819, China)

Abstract: From the view of developing automatic database testing tools, this paper studies TPC Benchmark C and TPC benchmark H standard specification in depth. On the above basis, the test model is presented, and the architecture and major classes of our automated database testing tools, VisualDBBench, are described. Furthermore, we verified that main-memory databases have more advantages than that of traditional databases.

Key words: benchmark; TPC-C; TPC-H; main-memory database; VisualDBBench

0 引 言

从 20 世纪 80 年代以来,数据库技术在商业领域的巨大成功刺激了其他领域对数据库技术需求的迅速增长. 随着用户应用需求的提高、硬件技术以及 Web2.0 等技术的快速发展,促进了很多新型数据库的快速发展. 这里比较有代表性的有内存数据库、非关系型数据库 NoSQL 等. 传统的关系数据库在应付 Web2.0 网站,特别是超大规模和高并发的 SNS 类

收稿日期:2014-06

基金项目:国家自然科学基金(61332006,61370154,61025007,61328202);973 计划(2011CB302200-G);863 计划(2012AA011004)

第一作者:李梁,男,博士研究生,研究方向为内存数据库.

通信作者:吴刚,男,副教授,研究方向为语义 Web,内存数据库等. E-mail: wugang@ise.neu.edu.cn.

型的 Web2.0 纯动态网站时已经显得力不从心,暴露了很多难以克服的问题.新型数据库的快速发展很好地解决了这些问题^[1].

随着各种新型数据库层出不穷的出现,按照统一的基准进行性能测试就显得很有必要.数据库的应用类型主要分为 OLTP 和 OLAP 两种,TPC-C 和 TPC-H 基准分别可以测试数据库在 OLTP 和 OLAP 方面的处理性能.目前,对于数据库性能测试工作大多是基于命令行的方式进行展开,配置和操作过程都比较繁琐.本文将基于 TPC-C 和 TPC-H 基准,开发一套自动化的可视化测试工具 VisualDBBench,直观方便地对各种数据库性能进行结果对比,定性地了解新型数据库,特别是内存数据库,所具有的性能优势.

1 TPC-C 与 TPC-H 测试基准规范

1.1 TPC-C 测试规范

1.1.1 TPC-C 基准介绍

TPC-C^[3,4]是由 TPC^[2](Transaction processing Performance Council,事务处理委员会)组织发布的基准测试标准.TPC-C 是一种旨在衡量 OLTP 系统性能与可伸缩性的行业标准基准测试项目.这种基准测试项目将对包括查询、更新及队列式小批量事务在内的广泛数据库功能进行测试.

1.1.2 TPC-C 商业模型

TPC-C 测试规范中模拟了一个比较复杂并具有代表意义的 OLTP 应用环境:假设有一个仓库和订单管理系统,它拥有若干个分布在不同区域的仓库;通常每个仓库供货覆盖 10 家地区分公司;每个分公司销售点给 3 000 个客户提供服务;平均每个客户的一个订单有 10 项产品;所有订单中约 1%的产品在其直接所属的仓库中没有存货,需要由其他区域的仓库来供货.图 1 描述了 TPC-C 商业模型的逻辑结构.

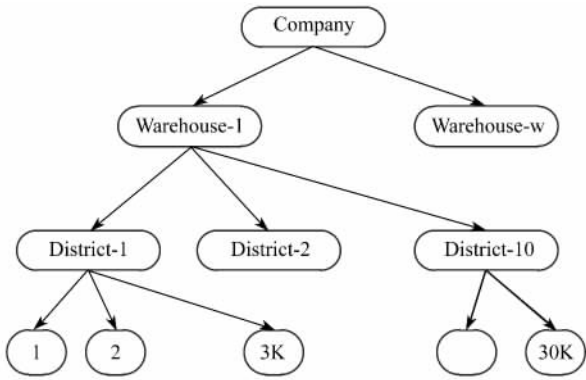


图 1 TPC-C 商业模型
Fig. 1 Business Model of TPC-C

1.1.3 数据库表规划

TPC-C 的事务处理在一个以 9 张表为基础的数据库上实现处理过程,其数据量及其对应关系如图 2 所示.

1.1.4 5 种测试事务

TPC-C 共定义了 5 种测试事务,在这个环境中,以一个新订单事务为核心活动,并辅以

执行其他四种事务:支付操作事务、订单状态查询事务、发货事务、库状态查询事务. 整个测试流程就是不断地执行事务.

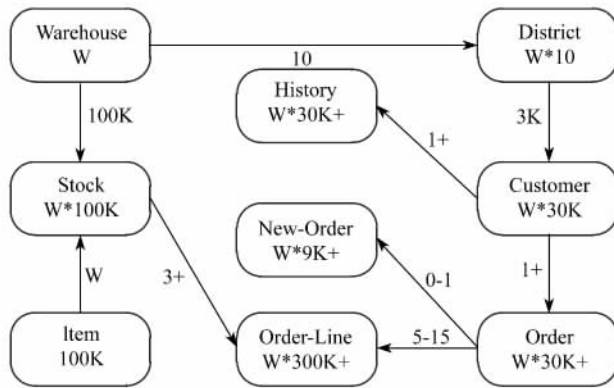


图 2 TPC-C 数据量及其对应关系

Fig. 2 The amount of data about TPC-C and the relations between the tables

1.1.5 度量指标

基准的性能度量用每分钟新增订单事务数(tpmC)表示,为了与 TPC-C 标准一致,TPC-C 结果的所有参考值必须包括 tpmC 值、与之相关的每个 tpmC 的价格和给定系统价格的可用日期.

1.2 TPC-H 测试规范

1.2.1 TPC-H 基准介绍

TPC-H^[5-7]是 TPC 发布的基准测试标准,用于评价数据库系统的复杂查询性能.它的主要目的是评价特定查询的决策支持能力,强调服务器在数据挖掘、分析处理方面的能力.

1.2.2 商业和应用环境

TPC Benchmark H 测试由一系列商业查询组成,这些查询在某种意义上代表复杂的商业分析应用.这些查询给出了一个实际的环境,描绘了批发商的活动以帮助读者将该基准的组件联系起来.图 3 说明了 TPC-H 所对应的商用环境,并且突出了 TPC-H 测试和其他 TPC 测试的根本区别.

TPC-H 不代表任何特定商业领域里的活动,而是可以被应用到任何需要管理、销售或在全球范围销售某种商品的行业(比如汽车租赁,食品销售,供应商等).但 TPC-H 并不是如何构建实际信息分析系统的模型.

TPC-H 数据库由八个单独的表(基本表)组成.这些表的列与列之间的关系如图 4 所示.

1.2.3 测试过程说明

TPC-H 测试主要包括 22 个 SELECT 查询.这些 SELECT 语句的复杂程度超过大多数实际的 OLTP 应用,在 22 个查询执行的同时,还有一对更新操作 RF1 和 RF2 并发地执行.

TPC-H 测试包括 3 个子测试:数据装载测试、Power 测试和 Throughput 测试.建立测试数据库的过程被称为装载数据,装载测试是测试 DBMS 装载数据的能力.测试装载数据的时间,这项操作非常耗时.Power 测试要求 22 个查询顺序执行一遍,与此同时执行一对 RF1 和 RF2 操作.与 Power 测试相比,Throughput 测试对 SUT 系统的压力有非常大的增

加,有多个查询语句组,同时有一对 RF1 和 RF2 更新流.

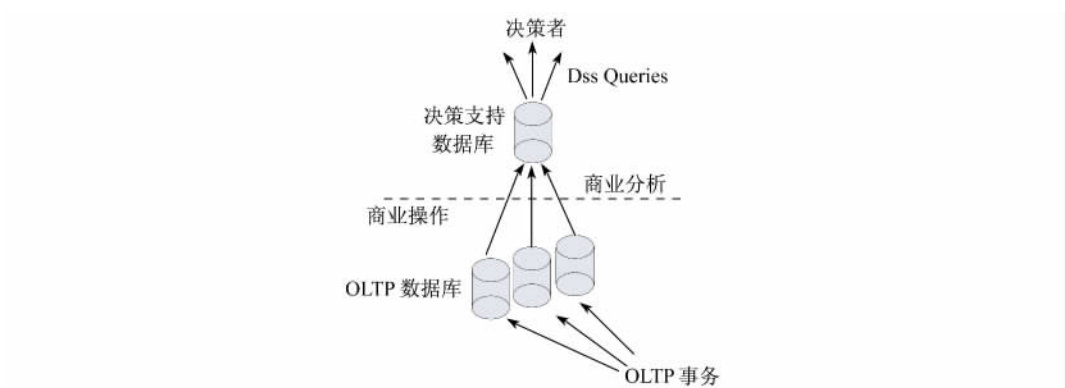


图 3 TPC-H 商业环境

Fig. 3 Business Environment of TPC-H

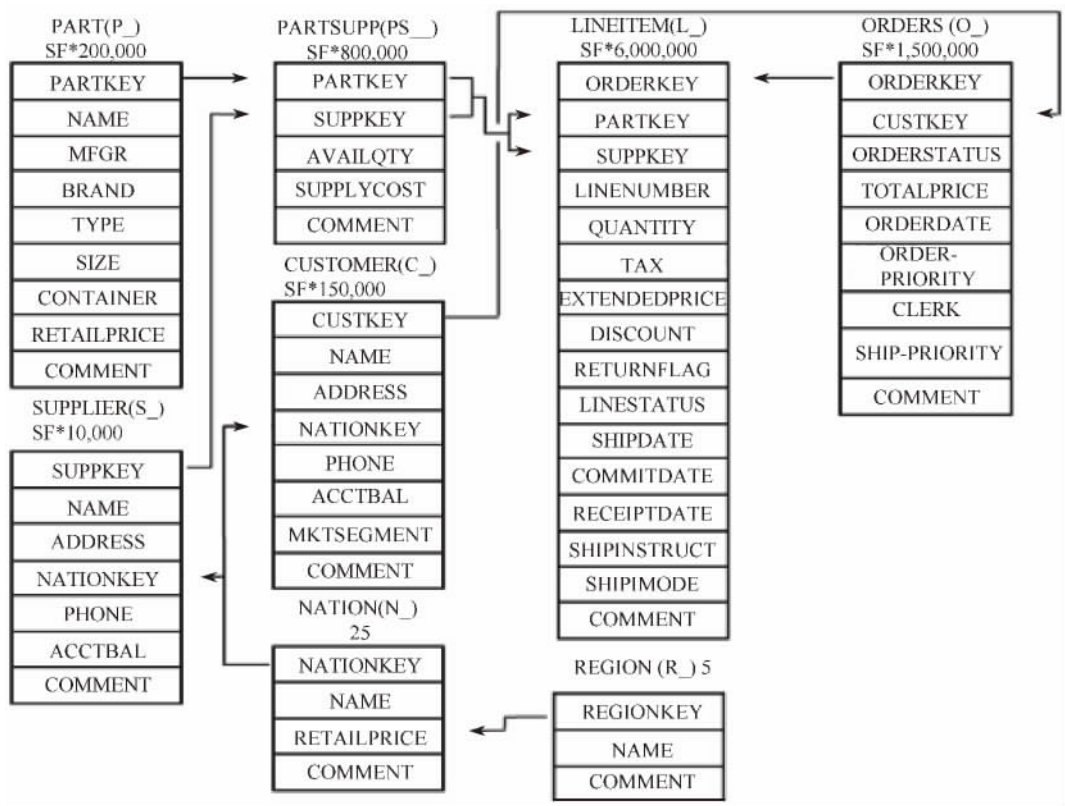


图 4 TPC-H 模式

Fig. 4 TPC-H Schema

1. 2. 4 度量指标

测试时,装载数据的全过程需要测量所用时间,并计入到数据装载时间中. 一般情况下,需要记时的操作有建表、插入数据和建立索引.

(1) 功率测试指标

Power@Size 是 Power 测试的结果,被定义为查询时间和更改时间的几何平均值的倒数,公式如下:

$$\text{Power@Size}(\text{QppH@Size}) = \frac{3600 * SF}{\sqrt[24]{\prod_{i=1}^{i=22} QI(i,0) * \prod_{j=1}^{j=2} RI(j,0)}},$$

其中, $QI(i,0)$ 为功率测试中单一查询流中查询 Q_i 的秒级时间间隔; $RI(j,0)$ 为功率测试中单一查询流中更新流 RF_j 的秒级时间间隔; Size 为用于测量的已选数据库规模, SF 为相应的规模大小.

(2) 吞吐量测试指标

吞吐量测试结果用来计算在已选数据库规模下的 TPC-H 吞吐量,其定义为整个执行的查询数与测量间隔长度的比,需如下计算:

$$\text{Throughput@Size}(\text{QthH@Size}) = \frac{S * 22 * 3600}{T_s} * SF,$$

其中 T_s 为测量间隔; S 为在吞吐量测试中使用的查询流数量; Size 为已选则的数据库规模, SF 为相关的比例因子.

(3) TPC-H 合成的每小时查询性能度量指标

数字值 TPC-H 功率和 TPC-H 吞吐量结合形成 TPC-H 合成的每小时查询性能测试,需如下计算:

$$\text{QphH@Size} = \sqrt{\text{Power@Size} * \text{Throughput@Size}},$$

2 现有的开源数据库测试工具

2.1 Database Benchmark

Database Benchmark^[8] 基准测试程序可以被用来测试不同数据库在大数据工作流负载环境下的处理性能. 工具整体以压力测试的方式随机插入大量记录,按照主键读取插入的记录,测试数据库的负载能力. 程序测试指标主要关注记录的插入速度和读取速度,以及最终数据库的数据空间开销.

工具面向接口设计,可以实现多个数据库的扩展功能,不同数据库只要实现抽象类 Benchmark 并实现以下四个基本函数 Init(), Write(), Read(), Finish() 即可. 工具可视化界面丰富,直观爽朗. 然而工具并不是基于 TPC-C 测试基准设计实现,与本文描述的工具在测试方法上不同.

2.2 BenchmarkSQL

BenchmarkSQL^[9] 是一款经典的开源数据库测试工具,内嵌了 TPC-C 测试脚本,可以对 EnterpriseDB、PostgreSQL、MySQL、Oracle 以及 SQL Server 等数据库直接进行测试. 工具通过配置文件可以插拔式的连接多个不同的数据库,通用性得到了很大的保证. 工具以日志的形式展示测试结果,没有实现可视化展示功能.

2.3 HammerOra

HammerOra^[10] 内嵌了两套标准测试脚本 TPC-C 和 TPC-H,也可以自己开发和扩充测试脚本,开发语言是 Tcl. HammerOra 测试工具相对完善,而且图形界面也比较直观. 但只能对少数数据库可以进行直接测试,对其他数据库则需要用户自行编辑测试脚本,开发语言

Tcl 不易掌握,增加学习成本.

3 VisualDBBench 测试基准设计与实现

3.1 基准测试过程设计

为了实现自动化的数据库基准测试工具,本文设计实现了一个基于 TPC-C 和 TPC-H 测试基准规范的可视化工具, VisualDBBench. 针对 TPC-C 和 TPC-H, 分别有两个测试界面,称为 VisualDBBench-C 和 VisualDBBench-H. 按照规范指出,整个测试的流程如图 5 所示. 程序在启动时可以自动加载用户配置,然后对多个数据库依次加载待测数据,对多个数据库测试结束后,为了方便用户,要能直观地显示测试结果.

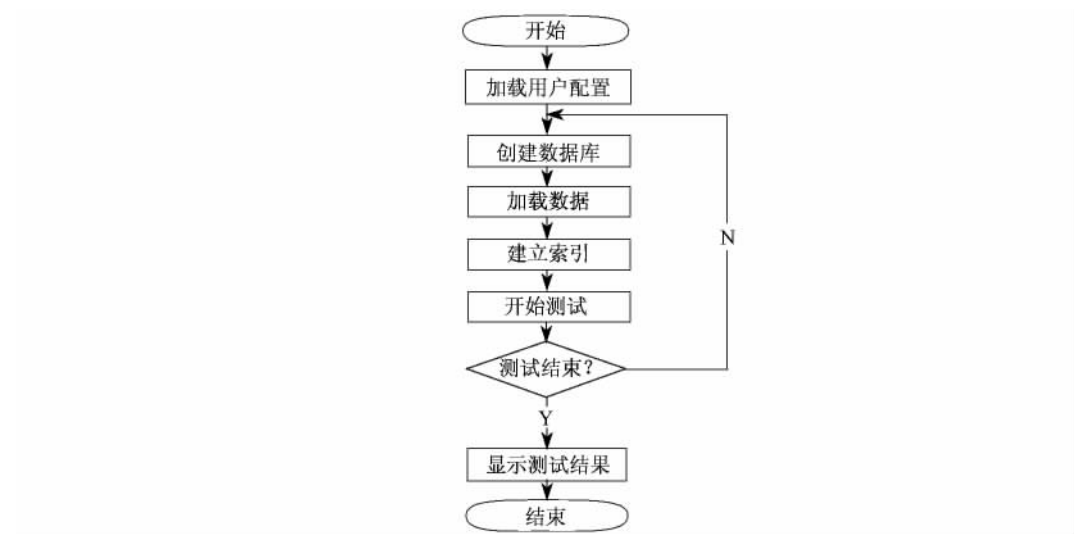


图 5 VisualDBBench 测试流程

Fig. 5 The progress of VisualDBBench

3.2 需求分析过程

仔细分析工具的功能需求,具体来讲,本系统主要有以下几个功能:

- (1) 配置待测数据库的个数;
- (2) 配置单个数据库的参数,主要包括数据库连接所用到的 URL 参数和用户名以及密码等;
- (3) 测试数据库:该模块主要包括创建数据库表,加载基本测试数据和创建数据库索引并开启测试过程;
- (4) 实时显示测试结果:为了直观地反应测试结果,必须有实时的可视化显示界面;
- (5) 加载 SQL 语句:这个模块主要是负责配置系统执行时需要用到的 SQL 语句;
- (6) 生成多个数据库对比结果:以条形图的形式对比展示多个数据库的性能结果.

图 6 展示了工具 VisualDBBench 的系统用例图. 数据库测试人员的需求用例主要包括:完成工具的配置,编辑 SQL 语句和测试数据库. 而测试数据库部分,主要包括创建数据库表,加载测试数据,给表创建索引以及测试过程. 更进一步,工具可以反馈给用户直观可视化的测试结果.

3.3 VisualDBBench 程序架构

虽然 TPC-H 和 TPC-C 标准给出测试方法、度量指标和测量方法,但并不能直接进行测试,必须开发用于测试的工具程序.

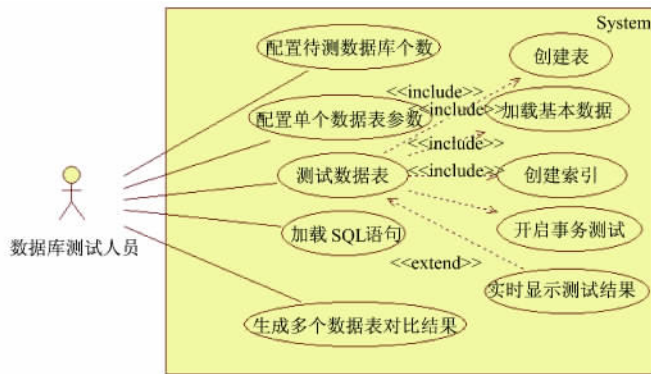


图 6 需求分析

Fig. 6 Requirements analysis

测试工具的体系结构见图 7,它可以通过 GUI 界面配置测试参数,自动完成整个基准测试. 测试驱动模块根据测试基准的不同,还会有各自的不同处,TPC-C 测试驱动主要完成 5 种基本事务的执行,而 TPC-H 测试驱动主要是执行 22 个负责查询语句和两个更新语句.

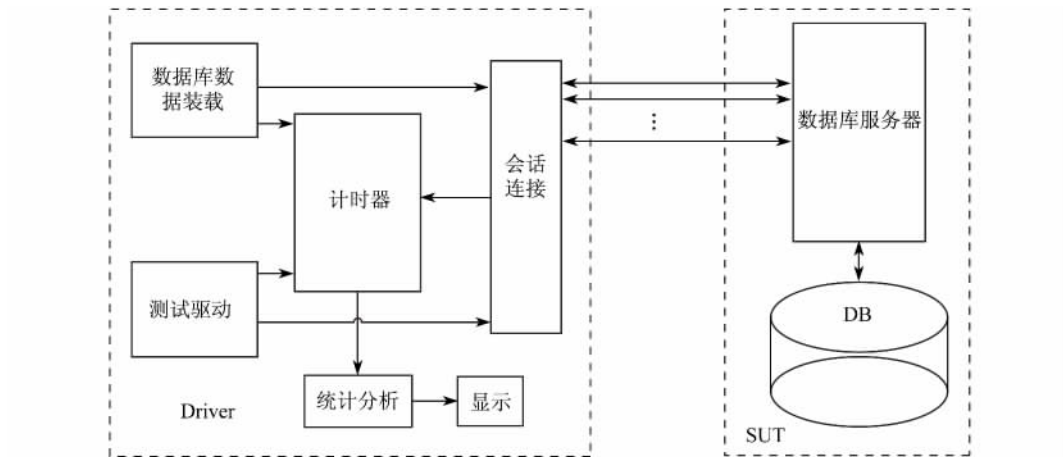


图 7 系统架构图

Fig. 7 System architecture

3.4 VisualDBBench-C 程序类图分析

工具按面向对象设计,采用 Java 编程. 工具的类图设计如图 8 所示.

程序中主要类结构说明:

(1) jTPCC 类:承担用户与 VisualDBBench 的交互,它包括标签、按钮、文本区等及响应事件. 用户可以在此设置参数如 DBMS 的类型、数据库的连接信息、TPC-C 测试终端和数据仓库的个数以及程序测试时间和 5 种测试事务所占权重. 界面中可以动态显示测试进度信息和测试结果信息.

(2) jTPCCTerminal 类:根据用户的参数配置,模拟每个终端开启五种事务的基本测试.

(3) ExecJDBC 类:连接数据库,运行特定 sql 语句文件,考虑到程序通用性的问题,sql 脚本语言定义在工程目录下的文本文件中,对于特定的数据库可以使用不同的脚本.

(4) BarResult 类:以柱状图的形式显示对比每个数据库的性能测试结果.

(5) MemoryPanel 类:以仪表盘的形式实时显示 JVM 内存使用情况.

(6) TpmCchartPanel 类:获取测试过程中 tpmC 的数值,实时显示给用户.

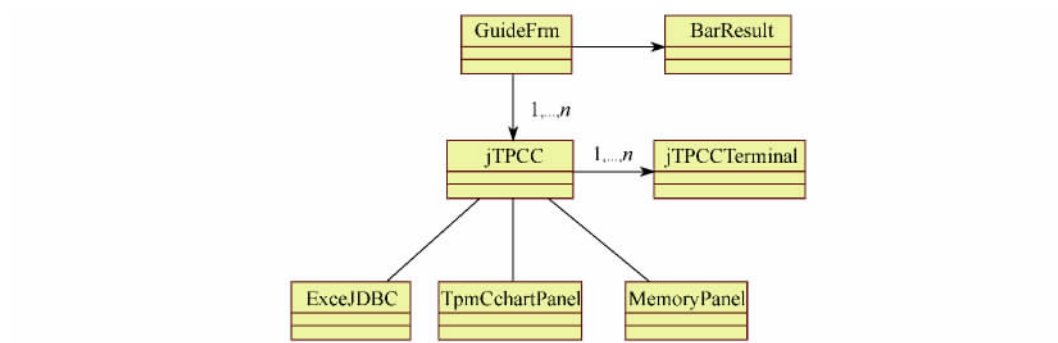


图 8 系统类图

Fig. 8 System class diagram

3.5 VisualDBBench-H 主要类说明

3.4.1 界面类

界面类主要功能有测试前的参数配置和测试过程的过程展示,以图形化界面的方式让用户与程序之间进行交互.可以配置的主要参数有:比例因子 SF 、查询流数目、数据库源、连续测试/单步测试选择等.

3.4.2 查询流类

将 QGEN 产生的查询语句,通过指定的会话发送给数据库服务器.每个查询需要附带查询流编号、查询编号.

3.4.3 更新流类

将更新操作通过指定的会话发送给数据库服务器,注意要保证更新操作成对执行.

3.4.4 计时器类

记录每个查询和更新的执行时间,以秒为单位.根据查询流的个数设置计数器,计录的结果写入磁盘中,该类可以和 VisualDBBench-C 程序部分进行复用.

3.4.5 统计分析类

将计时器类得到结果由统计分析类处理后,结果转换成适合存储的格式和有利于屏幕显示的格式.为了方便用户观察测试的进度和分析变化趋势,将统计结果变换后实时地输出在显示幕上,主要的可视化视图有 Throughput 测试中每个查询流查询完成数目;Power 测试中每个查询执行时间.

3.4.6 会话连接类

负责建立、管理和撤消 Driver 与 DBMS 之间的连接,保证查询流和更新流与 DBMS 连接之间的一对一的对应关系.

鉴于 VisualDBBench-C 和 VisualDBBench-H 程序在很多模块功能上类似,因此可以借助复用的思想,设计类结构.

4 VisualDBBench 面向内存数据库的应用

4.1 工具演示环境

相较于传统数据库,内存数据库抛弃了磁盘数据管理的传统方式,基于全部数据都在内存中重新设计了体系结构,并且在数据缓存、快速算法、并行操作方面也进行了相应的改进,所以数据处理速度比传统数据库的数据处理速度要快很多,正常情况下都在十倍以上. 其中特别值得一提的是,无需再分离运营系统和分析系统^[1]. 内存计算技术支持对运营数据进行分析,从而简化软件和硬件架构,最终降低总成本. 常见的内存数据库产品有 MySQL、H2^[11]、Redis、VoltDB 等. 目前内存数据库领域还没有一套完整的测试方案,我们将以 TPC 组织发布的基准规范对 H2 等常见内存数据库进行测试,定性了解内存数据库对比于传统数据库的性能优越性.

本次实验的主要目的在于测试传统数据库和内存数据库在 OLTP 和 OLAP 方面的性能差异,因此实验中选取了几个被广泛使用且具有代表性的数据库作为测试对象. 表 1 说明了工具中的开发与运行环境.

表 1 开发工具与运行环境

Tab. 1 Development tools and runtime environment	
运行平台或工具	版本号或型号
Windows	XP sp3
CPU	CoreI i3-2120
RAM	DDR3 2G
JDK	1.8.0
NetBeans	7.4
MySQL	5.0
H2	1.3.175
JFreeChart	1.0.13

4.2 配置文件

为了实现工具的通用性、灵活性和可插拔特性,在工具的外部用一个 conf. txt 的配置文件来管理我们需要测试的多个数据库. 另外在每一个数据库对应的配置文件内,记录测试的四个属性,分别是待测数据库的名字、JDBC 连接的驱动类、JDBC 连接数据库对应的 url 和连接数据库的授权用户名和密码.

4.3 TPC-C 测试中实时获取 tpmC 参数值

在对每个数据库的测试过程中,我们可以在 tpmc result 选项卡页面实时观测 tpmC 变化情况,工具可实时显示当前 tpmC 值和 tpmC 的平均值. 图 9 描述了数据库实时 tpmC 的测试结果.

4.4 获取多个数据库的 TPC-C 测试比较结果

向导界面的最后会将三次测试结果做对比. 图 10 展示了 H2 数据库、MySQL 数据库和 H2 内存数据库的 TPC-C 测试结果对比.

对比 H2 数据库和 MySQL 数据库的性能结果并没有显著的差别. 但相比内存模式的

H2 数据库,内存数据库在单位时间内处理事务的能力则出现百倍级的提升.

文献中指出,内存数据库的性能通常为传统基于磁盘的数据库性能的百倍以上^[1],从图 10 可以看出,实验结果基本与其吻合.

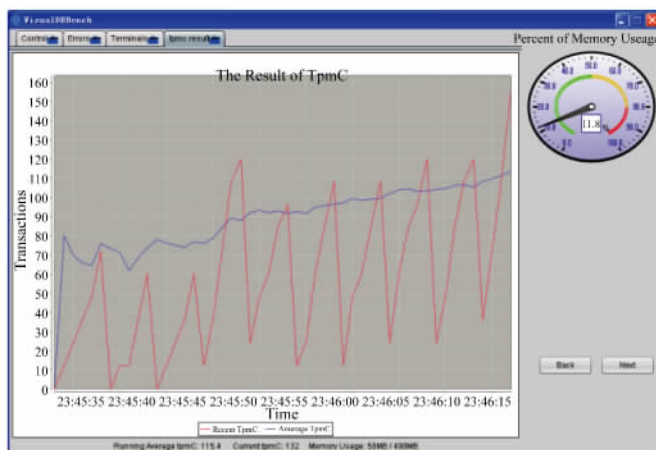


图 9 实时显示 tpmC 数值

Fig. 9 Show tpmC value in real time

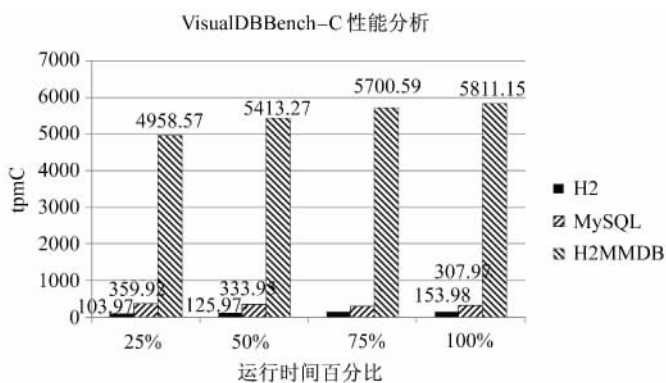


图 10 多个数据库 TPC-C 基准测试结果对比

Fig. 10 TPC-C Benchmark results compared with multiple databases

4.5 VisualDBBench-H 测试比较结果

VisualDBBench-H 测试程序主体按照 TPC-H 基准进行测试. 实验中分别选取 MySQL 磁盘数据库和 MySQL 数据库内存模式作为测试对象,对比分析内存数据库在联机分析处理方面的性能特点. 图 11 是 VisualDBBench-H 程序测试对比结果. 从图中我们可以看出,内存模式下,数据库作复杂语句查询的耗时更少.

5 结论与展望

本文分析了 TPC-C 和 TPC-H 基准规范测试的原理,主要包括对 TPC-C 与 TPC-H 基准的适用范围、内部结构进行了阐述和分析. 针对两个基准对多个数据库系统进行了性能测试,得出性能对比的图形化对比结果. 实现了一个自动化的测试工具 VisualDBBench, 实现

自动化的测试数据库性能,使用户无需深入了解测试规范就能实现测试过程.测试过程表明,内存数据库对比与传统数据库,在 OLAP 和 OLTP 方面都有非常显著的优越性,具有很好的市场潜力.

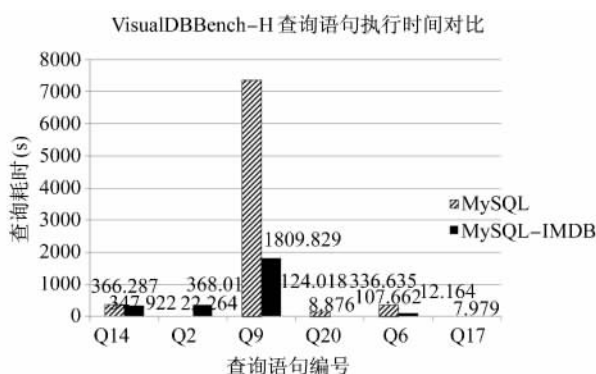


图 11 MySQL 数据库 TPC-H 基准测试结果对比

Fig. 11 MySQL database TPC-H benchmark results contrast

本文的不足之处在于,数据库的性能评测和性能优化是一项复杂而又艰巨的任务,本文只做了很小的一部分工作,其中还有很多地方需要进一步完成.

需要进一步研究和改进的地方有:

1. 在测试中,未包括 GUI 的响应时间,因此测试得到的时延并不是系统真正的时延,只能大体上表示系统的时延.要使测试结果真实,就必须测量 GUI 的响应时间,在测试中如何减少用户时间的干扰,有待于进一步的研究.
2. 跟踪国际最新的评测工具和分析数据库性能的方法,摸索出一套实用的瓶颈测试方法,对应用程序优化、内存优化、I/O 优化等问题进一步讨论,考虑更通用的解决方案.
3. 参数配置也是数据库性能调整的一个大类,涉及的参数很多,原因也很复杂,在研究过程中应该作为重点来规划.

[参 考 文 献]

- [1] 哈索,亚历山大·蔡尔.内存数据管理[M].北京:清华大学出版社,2009.
- [2] TPC. Transaction processing performance council[EB/OL]. [2014-6-30]. <http://www.tpc.org>.
- [3] 孙雪祥,蒋艳凰,张怡. TPC-C 测试系统的实现[J]. 计算机工程, 2006, 32(20): 81-83.
- [4] 王良,蔡荣. 基于 TPC-C 标准的自动化测试工具 TPCCLoader[J]. 计算机工程与应用, 2006, 41(25): 102-105.
- [5] 王良,栗跃鹏,杨政. TPC-H 自动测试工具 TPCHDriver 的研究[J]. 计算机工程, 2009, 35(2): 18-20.
- [6] 康强强,江舟,金澈清,等. TPCHSuite: 一个 TPC-H 自动化测试工具的设计与实现[J]. 计算机研究与发展, 2013, 50(Suppl): 394-398.
- [7] 宋钦宇. 数据库 TPC-H 测试系统的设计与分析[J]. 长春大学学报, 2008, 18(2): 62-65.
- [8] Database benchmark[EB/OL]. [2014-07-02]. <http://database-benchmark.com/>.
- [9] BenchmarkSQL[EB/OL]. [2014-07-02]. <http://benchmarksql.sourceforge.net/>.
- [10] SHAW S. HammerDB[EB/OL]. [2014-06-28]. <http://sourceforge.net/projects/hammerora/>.
- [11] H2 Database Engine[CP/OL]. [2014-07-01]. <http://www.h2database.com/html/main.html>.

(责任编辑 林 磊)