

数据仓库的可扩展性研究与设计

秦学勇^{1,2}, 刘 栋³

(1. 中国人民大学 信息学院, 北京 100872;

2. 安徽建筑工业学院 电子与信息工程学院, 安徽 合肥 230022;

3. 北京理工大学 软件学院, 北京 100081)

摘 要:随着海量异构的数据不断进入数据仓库和系统用户的大量增加,在数据质量、可用性和查询等方面的因素将严重影响数据仓库的性能,所以数据仓库必须设计成可扩展的体系结构。文中采用可扩展的软件并行和硬件并行相结合的方法进行数据仓库的性能扩展,在数据仓库初建时采用 SMP 结构,当数据仓库增长到一定的时候采用高速缓存相关的非一致性内存访问结构,并且较好地利用 I/O 并行性,取得较好效果。使得当大量异构数据涌入可扩展数据仓库中时系统性能不会下降,很好地满足决策支持。

关键词:可扩展性;数据仓库;并行;NUMA

中图分类号:TP311.5

文献标识码:A

文章编号:1673-629X(2009)05-0065-03

Research and Design on Scalability of Data Warehouse

QIN Xue-yong^{1,2}, LIU Dong³

(1. Information School of Renmin University of China, Beijing 100872, China;

2. School of Electronic and Info. Eng., Anhui Inst. of Architecture and Industry, Hefei 230022, China;

3. Software College of Beijing Industry University, Beijing 100081, China)

Abstract: Along with mass heterogeneous data and system users getting into data warehouse, its performance begins to decline on data quality, usability and query aspects, thus data warehouse should be designed with scalable architecture. Scalable software and hardware technologies are used to extend test data warehouse in this paper, and in the beginning SMP is used in data warehouse. CC-NUMA is employed for extending data warehouse when it increases to a certain scale that SMP can not satisfy system performance need, and at the same time I/O parallel is used fully. In this case, a good effect on system performance is obtained when a great deal of heterogeneous data swarming into data warehouse, and meets the needs of decision support better.

Key words: scalability; data warehouse; parallel; NUMA

0 引言

数据仓库在创建、维护到更新扩展的过程之中,也就是在其不断成长的过程之中,性能不可避免地成为问题。一旦哪里有少量的数据,那里不久就会有海量的数据;哪个数据仓库环境中少量的用户,那里不久就会出现大量的用户群。数据质量、可用性和查询等方面的因素将影响数据仓库的性能,使得用户感觉数据仓库的性能逐渐下降。所以数据仓库必须设计成可扩展的体系结构,包括软件的并行和硬件的并行。文中建立了考试数据仓库平台,在数据集成^[1,2]阶段采

用 SQL Server 2000 的 DTS 对异构的数据源进行转换,而对文本文件等不是非常规范的数据文件采用装载前预处理的方法。最后将硬件平台扩展为基于 SMP 的 CC-NUMA 的结构,从而更好地提高考试数据仓库的性能。

1 数据仓库简介

数据仓库已经在很多行业广泛使用,商务管理人员通过收集、集成、预处理和存储数据仓库中的数据进行分析 and 商务决策。在这里引用一个权威公认的由数据仓库之父 W. H. Inmon 所下的定义^[3]:数据仓库是一个面向主题的、集成的、非易失的且随时间变化的集合,用来支持管理人员的决策。美国斯坦福大学计算机科学系的一个专门小组通过对数据仓库技术的研究,提出了数据仓库技术的一个比较完善的体系结构。

收稿日期:2008-08-26

基金项目:安徽省自然科学基金项目(2006KJ066B)

作者简介:秦学勇(1974-),男,安徽合肥人,博士研究生,讲师,研究方向为数据库与多媒体软件、分布式计算。

数据仓库总体结构如图 1 所示。

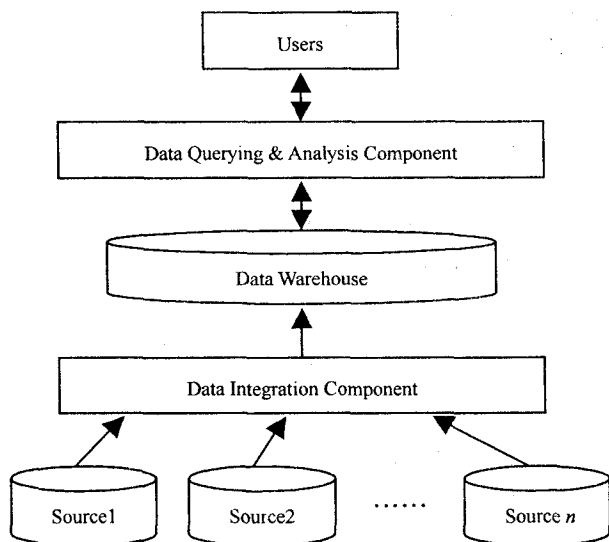


图 1 数据仓库总体结构

2 数据仓库扩展性相关技术

数据仓库^[4]所要管理的数据量要远远多于传统系统所管理的数据量,在最初配置数据仓库平台时无论多么大,随着时间的推移数据仓库平台必须支持更多的数据和用户、更多的和复杂的查询处理,所以数据仓库平台必须是有机的,也就是说随着时间的推移它们自身可以改变以适应业务需求的变化。并行处理对数据仓库来说是特别重要的,这是由数据仓库自身的内在特征决定的。数据仓库平台主要是 OLAP 应用,在 OLAP 环境之中充满了大量的复杂的查询,为了满足快速的查询需要和数据仓库自身发展的需要,需要采用更多和更高级的方法来应付这样的查询需要。数据仓库本身是非常庞大的,而且数据仓库更倾向于快速增长,从而一个可扩展的数据仓库体系结构是非常重要的。除了较小的数据仓库平台以外,传统的硬件和软件技术不能够很好地满足数据仓库平台的性能和可扩展的需要,那么必须采用并行技术^[5,6]。所谓并行技术就是通过将一个大的问题或任务分割成小型的、更易于管理的块,并同时进行这些小块的处理。通过利用并行性原理所获得的性能远远超过运用传统技术所达到的性能等级,通过并行技术可以得到二种性能增加:扩增和加速。

并行技术的实现可以分为硬件并行和软件并行。硬件并行是体系结构的改善,包括对称多处理器(SMP)、群集(Cluster)、海量并行处理器(MPP)、非均匀内存访问(NUMA)等结构。软件并行是指数据仓库管理员理解软件开发者提供的不同类型的并行性,这是非常重要的。软件的开发者提供三种类型的并行

性:数据并行性、功能并行性和流水线并行性。可扩展的硬件结构都有其强项和不足之处,必须根据自身的实际情况来选择哪种硬件结构,去选择最适合你的系统而不是去选择最时尚的或最贵的,应该把软件并行和硬件并行结合起来使用,从而达到更好的扩展效果。

3 可扩展数据仓库设计

3.1 数据仓库的设计

在信息技术高速发展的今天,如果不会使用计算机去解决实际工作和学习上的问题将是一件非常可怕的事情,所以现在社会上和学校里面有很多的电脑培训机构。作为这些电脑培训机构的管理部门,对这些培训机构的管理包括培训的内容、教学的质量和结业学生的素质等方面进行考察和监督,而衡量一个学校的教学质量和学生素质的一个很重要的方面是该学校的学生在各类计算机的考试中通过率的情况。通过对各个学校和培训机构中学生考试成绩的分析可以做出很多的决策支持,如果某一个培训机构经常在某种考试之中排名靠前,可以通过分析数据发现其原因,到底是由于其教师素质较高还是学生的起点较好或者是该培训学校在这个模块中有其独特的特色等,可以将该培训机构的经验加以普及从而提高整个行业的教育水平。在文中构造的数据仓库平台中,你可以很轻松并且很方便地查询出你所需要的信息,例如你可查询某个城市某所学校的考试通过率和每个具体的课程的通过率,也可以查询某年某个地区的考试情况等方面的问题。

在文中就是构造这样一个数据仓库平台以供决策支持,实验的操作系统和数据库平台为:操作系统, Windows 2000 Advanced Server; 数据库, SQL Server 2000。先经过数据集成将数据转载入数据仓库中去建立考试数据仓库平台,在数据集成的时候采用 DTS 来进行装载,然而对于像文本文件和 E-Mail 等文件采用预先处理的方法,避免装入数据仓库之后再进行处理。在数据仓库平台中采用的是星型模型或雪花模型进行建模,而在传统的 OLAP 环境中采用 E-R(实体-关系)图,数据仓库平台采用星型模型^[7]。为考试数据仓库设计如下的事实表和维表:

事实表:

exam_fact, 考试事实表

(student_id, school_id, module_id, examsite_id, time_id, mark)

维表:

student, 学生维表

(student_id, name, gender, age, graduate_level, idcard_num,

which_school, hometown, examsite)

school, 学校维表

(school_id, school_name, city, section, teacher_num, size, property, student_num, school_master, phone_num)

time, 时间维表

(time_id, year, quarter, month)

module, 考试模块维表

(module_id, module_name, module_description, difficulty)

examsite, 考点维表

(examsite_id, city, section, street_num, name, examsite_num)

通过在 Analysis Manager 中设计事实表和各个维表,并显示它们之间的关系。

然后建立多维数据集 exam, 再对多维数据集 exam 进行处理, 在进行完处理和编辑以后通过在 Analysis Manager 中进行数据浏览, 如图 2 所示, 在多维数据浏览器中可以通过双击带有 + 号的成员进行详细数据的浏览, 可以在浏览器的上方下拉列表中取选择你所需要的信息, 其结果将显示在下面的数据集表中, 从而你可以很快得出一些聚集的信息以助于你的决策支持。你可以单击维表中的总体数据, 还可以单击维表中的具体和轻度概括对数据, 结果将显示在下方的表格之中, 可以很快地查看到你所需要的信息, 从而帮助企业的决策者做出决策分析。

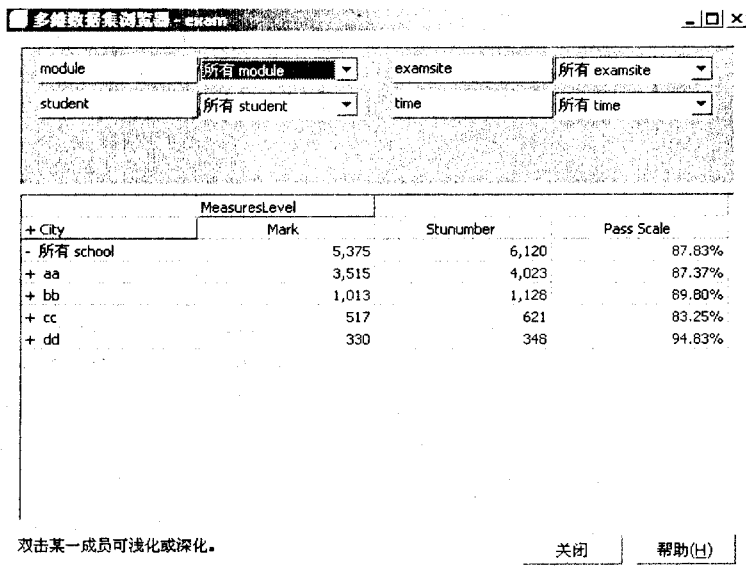


图 2 浏览多维数据集图

3.2 数据仓库的扩展

在考试数据仓库平台中采用分割和布纹相结合的方法去充分地利用 I/O 的并行性, 因为在磁盘的读取过程中有一个称为扫描进程(或线程)负责从磁盘读取数据并将数据传递给处理器去处理, 在一个特定的时刻一个扫描线程只能从一个磁盘中读取数据而且只能将这个磁盘中的数据传递给一个处理器。当使用多

处理器系统的时候要使数据总是不断地流入到所有的处理器中, 如果把所有的数据都放在一个磁盘中, 那么每次只有一个扫描线程是活动的, 并不能利用任何一种 I/O 并行性^[8], 所以应该将数据分布到多个磁盘上。

在考试数据仓库平台之中, 将四个城市字段通过以下方法进行分区:

PARTITION BY VALUE

city = 'aa' IN partition_1;

city = 'bb' IN partition_2;

city = 'cc' IN partition_3;

city = 'dd' IN partition_4;

当分区以后你要访问 aa 中的数据你就可以在 partition_1 中去访问它, 而不需要去访问另外的三个分区。这种分割语法告诉数据库在建立这种表格的时候(和随后对表格数据的任何更改时)把所有 aa 数据放到一个称为分块 1 的分块之中, 这样数据库引擎就知道对于上面的查询, 可以排除分块 2, 3 和 4。这样做可以释放更多的机器资源, 从而使得可以同时处理其他的查询, 因而这将极大地节省处理能力。但是它并不能提高查询时间, 必须要把这些数据布纹到不同的磁盘上或者逻辑卷组之上。这样当你使用多处理器进行查询的时候可以很大地改善你的数据仓库平台的性能, 从而满足可扩展的需求。

在这里采用这样的一种架构: 在数据仓库初建时采用 SMP 结构, 当数据仓库增长到一定的时候采用高速缓存相关的非一致性内存访问, 每个节点为 SMP 结构, 这种结构具有很好的可扩展性。CC-NUMA^[9]是 NUMA 的一种类型, 在 CC-NUMA 系统中, 分布式内存相连接形成单一内存, 内存之间没有页面复制或数据复制, 也没有软件消息传送。CC-NUMA 只有一个内存映像, 存储部件利用铜缆和某些智能硬件进行物理连接。Cache Coherent 是指不需要软件来保持多个数据拷贝的一致性, 也不需要软件来实现操作系统与应用系统的数据传输。如同在 SMP 模式中一样, 单一操作系统和

多个处理器完全在硬件级实现管理。

CC-NUMA 与 SMP 类似, 能够处理多个连接起来的处理器, 每个处理器能存取一个公共的存储器组。这种结构把处理器分成几个节点, 在每个节点中所有处理器互连在一起, 互相通信, 并可与节点内的本地存储器通信以减轻 SMP 的总线阻塞情况。例如, 一个 64 处理器的服务器可以分成 2 个大节点, 每个节点有

(下转第 71 页)

文中的主要内容是以讨论基于 UML 状态图的测试技术研究为主线,在介绍统一建模语言(UML)技术的基础上,阐述了基于 UML 状态图的测试方法。在分析加载时的 Compiler 对象模型的过程中,运用了 UML 的建模思想为其构造了 UML 状态图,并在 UML 状态图模型的基础上,生成扩展的有限状态机,构造了测试策略,产生测试路径,生成较少的测试用例。

2 结束语

面向对象的建模语言 UML 是一种全程建模语言,运用于从需求分析至测试分析的软件开发全过程,有力地推动了面向对象的软件开发。基于 UML 建模技术的测试方法,促使软件测试技术更加成熟和高效。状态图模型是当今软件开发的一种重要的建模手段,被广泛应用于描述软件结构和指导软件开发。基于状态图的测试技术研究是软件测试领域的一个热点,在实际中有着广阔的应用,发展潜力巨大,而对它的研究也在不断完善之中。为了达到故障的覆盖率,需要生成较多的测试用例,如何减少测试用例生成的数量是基于 UML 状态图测试方法研究的一个方面,同时,今后可以深入研究综合利用待测试系统的各种模型图的

测试用例生成方法。

参考文献:

- [1] 宫云战. 软件测试[M]. 北京:国防工业出版社,2006.
- [2] 董 威,王 戟,齐治昌. UMLStatecharts 的模型检验方法[J]. 软件学报,2003,14:750-756.
- [3] 张术梅,孙 辉. 基于 UML 的面向对象的软件静态测试方法的研究[J]. 计算机技术与发展,2007,17(1):125-127.
- [4] Offutt A J, Abdurazik A. Using UML Collaboration Diagrams for Static Checking and Test Generation[C]//Proc. 3rd International Conference on the Unified Modeling Language (UML'00). York, UK: [s. n.], 2000:383-395.
- [5] 江 曼,王天青,潘金贵. 基于 UML 状态图的面向对象软件测试用例生成[J]. 计算机科学,2006,33(6):284-290.
- [6] 章 涛,顾 庆,陈道蓄. 基于 UML 状态图的测试技术研究[J]. 计算机科学,2007,34(10):264-267.
- [7] 徐宝文,周 明,卢红敏. UML 与软件建模[M]. 北京:清华大学出版社,2006.
- [8] Booch G, Rumbaugh J, Jacobson I. The Unified Modeling Language User Guide [M]. Boston, MA: Addison Wesley Longman Inc., 1998.
- [9] 徐 博,樊晓光,田 涛. 扩展 UML 状态图在软件测试中的应用[J]. 计算机工程,2007,33(24):78-82.

(上接第 67 页)

32 个处理器,并有其自己的存储器组。处理器也可存取其它所有节点中的存储器组,这种存取时间随着节点的距离远近而异,CC-NUMA 相对于 SMP 扩展性更强,只需一个操作系统,管理相对容易。与群集不同的是,群集采用一种松散的结合方式,在几台机器之间相互通信,内部交换时间长,消耗大。而且把几台机器作为一个系统管理,势必增加了管理上的难度。而 CC-NUMA 计算机则不同,无论它内部有多少个处理器,对用户来说,它也仅仅是简单的一台计算机。总而言之,CC-NUMA 克服了 SMP 和群集的某些弊端,在它们不能施展才能的地方发挥了作用。

4 结束语

数据仓库不是一日建成的,是通过不断的迭代方式来建立它,当大量的数据开始在数据仓库中堆积时,数据仓库的性能就开始成为一个重要的问题。所以数据仓库必须设计成为一种可扩展的结构,以便适应将来的数据环境。笔者提供采用软件和硬件的并行技术来扩展数据仓库平台,从而满足数据和用户日益增长的需要。

参考文献:

- [1] Strum J. Microsoft SQL Server7 数据仓库技术指南[M]. 刘汉宇,等译. 北京:机械工业出版社,2000.
- [2] 秦学勇,姚燕生. 可扩展数据仓库若干关键问题研究与分析[J]. 计算机技术与发展,2006,16(12):136-138.
- [3] Inmon W H. Building the Data Warehouse[M]. 王志海,等译. 北京:机械工业出版社,2000.
- [4] Widom J. Research Problems in Data Warehousing[C]//In Proceedings of the 4th Int'l Conference on Information and Knowledge Management (CIKM). Baltimore, Maryland, USA: [s. n.], 1995.
- [5] Tamura T, Nakamura M, Kitsuregawa M, et al. Implementation and performance evaluation of the parallel relational database server SDC-II[C]//In Proceedings of International Conference on Parallel Processing. Ithaca, NY, USA: [s. n.], 1996:212-221.
- [6] Inmon W H, Rudin K, Buss C K, et al. 数据仓库管理[M]. 王天佑,等译. 北京:电子工业出版社,2000.
- [7] Shapiro J R. SQL Server 2000 参考大全[M]. 周 之等译. 北京:清华大学出版社,2002:610-635.
- [8] 陈国良. 并行计算——结构、算法、编程[M]. 北京:高等教育出版社,1999.
- [9] 吴少刚. 机群 OpenMP 系统的设计与实现[J]. 计算机学报,2004,27(7):904-911.