

数据库物理自调优研究技术综述^{*}

曹巍

(中国人民大学信息学院, 北京 100872)

摘要: 数据库物理自调优是自管理自调优数据库中一项重要的内容,为厘清这一领域的重要研究工作,需要对关系数据库物理自调优的背景、问题界定、经典技术和研究的新问题等方面进行综述。总结了数据库物理自调优的经典技术,并从优化程度、可扩展性、可用性、可管理性及测试基准四个角度重点阐述了现今数据库物理自调优的关键技术和作为研究开发热点的一些问题,指出不同的物理自调优工具在技术上的共同基础,分析评价了这些关键技术及其方法的各自特点。最后还总结了该领域近几年出现的新的研究问题,并展望了未来的研究方向。

关键词: 物理自调优; 查询优化器; what-if 分析; 物理结构; 工作负载; 物理空间搜索

中图分类号: TP73.9 **文献标志码:** A **文章编号:** 1001-3695(2012)05-1606-07

doi:10.3969/j.issn.1001-3695.2012.05.002

Survey on automatic physical database design

CAO Wei

(School of Information, Renmin University of China, Beijing 100872, China)

Abstract: Automatic physical database design is an important part in self-tuning and self-managing databases. To reflect upon the main research work on this topic, this paper gave a brief survey on the classic technology in physical database design tools in terms of background, problem definition, techniques, and potential future research directions. Specifically, this paper analyzed the common techniques from four dimensions, namely, cost optimality, scalability, high usability and benchmarking. And by comparing different strategies, it also spotted and evaluated their technical differences in detail. The paper discussed new research endeavors in recent years which went beyond the limits of the classic physical design problem. In the end, it gave the conclusion and the vision of the future research directions.

Key words: automatic physical design; query optimizers; what-if analysis; physical structures; workload; search in physical space

0 引言

近十年来,自管理的数据库系统越来越被学界和业界所重视。主要是因为随着计算机硬件单位成本的降低、Internet 应用的普及带来的海量数据环境及其新的应用模式的需求,数据库系统实现的功能越来越多,其自身也越来越复杂。而依靠有经验的数据库管理员(DBA)对如此功能丰富、结构复杂的数据库系统进行配置、管理、调优等日常工作所带来的成本,会使系统的总体拥有成本(total cost of ownership, TCO)变得很高。因此,为了有效地降低系统的总体拥有成本,一个行之有效的方向就是使数据库系统具有自管理的特性。

数据库系统的自管理是相对于以前的人工管理和调优的方式提出来的。在人工管理和调优的阶段,数据库管理员参与完成数据库系统的逻辑和物理设计,配置数据库的重要参数,实施数据库系统的运行维护,以及进行数据装载与数据重组织重构等重要的设计、管理、日常维护等工作。这些在很大程度上依赖于数据库管理员的技术和经验,同时也使人员投入的成本加大。

自管理的数据库系统不需要或很少需要外部的干涉,能够自动识别并记录数据库系统运行的状况,敏锐地感知系统内部

状态及其变动,同时根据外部应用需求的变化,自动地调整自身以适应新的应用环境和应用。具体说来,数据库系统的自我管理功能集中体现在以下几个方面^[1]:

a) 数据库系统的自我配置,包括数据库系统的初始安装,系统的启动和参数配置等方面实现自主管理和配置。

b) 数据库系统的自动设计,包括自动的数据库逻辑设计和自动的数据库物理设计。自动的数据库物理设计就是在给定工作负载和空间限制的条件下,自动地给出较优化的物理结构的推荐方案,并可以依照用户的选择自动地生成相应的物理结构。

c) 数据库系统自动运行维护和调优。它既包括维护描述系统和数据状态的重要统计值的准确性,也包括利用这些统计参数,作为形成正确判断和决策的重要依据。

d) 数据库系统自动故障检测、发现、修复。系统能够随时检测自身运行的状态,及时发现异常情况,并根据自身积累的知识对异常的来源进行定位,给出解决异常的建议或者自动地修复异常。

e) 数据库系统的自我保护和自我恢复。数据库系统能自动识别非法的数据访问和恶意的数据破坏,并自动地采取措施屏蔽或避免非法的恶意访问;在数据发生了损坏的情况下,自动地利用日志或者备份进行数据恢复。

收稿日期: 2011-10-25; 修回日期: 2011-12-06 基金项目: 国家自然科学基金资助项目(61170013)

作者简介: 曹巍(1975-),女,辽宁沈阳人,讲师,博士,主要研究方向为高性能数据库、数据库自我管理自调优、闪存数据库等(caowei@ruc.edu.cn)

数据库系统的自管理、自调优、自组织、自我治愈和自我保护等方面的功能,都对传统的数据库实现技术提出了新的要求和挑战。这不仅仅体现在向传统的 DBMS 各模块中加入自管理自调优的功能,或者在数据库系统设计、开发、调试、维护的各阶段加入自管理自调优的功能,更主要的是,数据库系统的自管理自调优是实现信息系统自主计算这一更高目标的关键一步,它将数据库系统本身作为一个自管理自适应的整体来考虑,综合各部分的功能并协调内部各功能模块之间的关联,实现真正意义上的自管理数据库系统。

本文讨论的是数据库物理自调优的问题。严格地按数据库系统的规范设计法来说,数据库物理自调优既包括在数据库设计阶段根据系统的工作负载自动地设计物理结构的推荐方案,也包括在数据库系统的运行阶段,根据系统的运行状态适时地自动调整数据库的物理存取路径。前者可称为自动的数据库物理设计,后者可称为自动的数据库物理调优。本文将这两种情况统称为数据库物理自调优。在结构上,本文给出了研究问题的界定,论述自动数据库物理设计的四个要求,并分别从多个角度综合分析评价经典数据库物理自调优技术的历史和现状,以及自动数据库物理设计新的研究课题。

1 研究问题的定义和划分

数据库物理自调优是指自动地为数据库系统的查询推荐合适的物理存储结构和存取路径,从而改进系统的性能。文献[2]最早进行关于数据库物理自调优的研究,提出了用查询优化器来评价索引质量的方法,并在 System R 上实现了一个自动物理设计的工具。本文从以下几个角度来划分历史上曾经出现过的数据库物理自调优技术。

1.1 数据库物理自调优实现的形式

大多数 DBMS 商家或者第三方的商家都是以客户端工具的形式提供自动数据库物理设计的功能,如 Microsoft SQL Server 2005 的 Database Tuning Advisor、IBM 的 DB2 Designer Advisor、Quest Central for Sybase SQL Tuning 等。也有的 DBMS 厂商以服务器端解决方案的形式提供自动的数据库物理设计的功能,如 Oracle 10g 的 SQL Access Advisor 就是作为一个服务器端的任务来运行以实现物理自调优的功能。这两种方式都需要从外部调用查询优化器的代价分析模块用于支持对虚拟的物理存取路径导致的代价进行估算。还可以将数据库物理自调优的一部分功能直接放在查询优化器中实现。比如 Oracle 10g 的查询优化器有普通工作模式和调优模式两种,在后一种模式下,查询优化器可以实现统计数据的调优、参数设置的调优和物理存取路径的调优等功能,只是这里的物理存取路径的调优针对某个 SQL 语句,而不是针对工作负载的。

1.2 面向整个负载与面向单个语句

工作负载指的是经常执行的 SQL 语句的集合。很多物理自调优的客户端工具是面向一个工作负载集合的,如 Microsoft SQL Server 2005 的 Database Tuning Advisor、IBM 的 DB2 Designer Advisor、Oracle 10g 的 SQL Access Advisor 等。这样做的好处就是可以针对系统比较重要或者关键的典型应用,设计实现优化的物理存取路径,特别是需反复运行应用时,能大幅度地提高其性能。而如前所述,在 Oracle 10g 查询优化器的调优模式下进行的物理自调优是面向单个的 SQL 语句,而这个语句

被认为可能对整个系统的性能影响较大。这种方法比较灵活,可以随时根据检测到的系统运行状况,对有问题的语句或者按照用户的要求进行物理自调优。

1.3 处理不同物理结构的类别

早期数据库物理自调优工具大多针对单一物理存取结构,如文献[2]中的 DBDSGN、Microsoft SQL Server 7.0 的 Index Tuning Wizard、IBM DB2 UDB 6 的 Index Advisor、Oracle Index Tuning Wizard、BMC 的 Index Advisor 等,它们都只是对最常见的物理结构——索引进行物理自调优。Oracle 9i 的 Summary Advisor 则只针对物化视图进行物理自调优。还有针对另外一种物理结构——水平划分进行物理自调优的研究,如 IBM 的 Partition Advisor^[3]。稍后一些的数据库物理自调优的产品考虑将索引和物化视图结合起来进行自动数据库物理设计,如 Microsoft SQL Server 2000 的 Index Tuning Wizard、IBM DB2 UDB 的 Design Advisor 等。

随着技术的发展,数据库引擎提供了越来越多的物理结构以提高系统的性能,因此当前数据库物理自调优的产品考虑了更多种类的物理结构,包括索引、物化视图、水平划分、垂直划分,还有面向多维分析的多维聚簇的结构等,这些工具产品充分考虑了不同的物理结构之间的相互关系,使推荐的自动数据库物理设计方案日臻完善。典型的代表有 Microsoft SQL Server 2005 的 Database Tuning Advisor、IBM DB2 UDB 8.2 的 Design Advisor 等。

当前的数据库物理自调优的技术是在以往的数据库物理自调优技术基础上积累和创新的结果,其显著特点是负载驱动的、面向多种物理结构,大多以客户端工具的形式来实现。本文的研究侧重在作为客户端工具的自动数据库物理设计的关键技术。该问题的经典描述为^[4]:给定数据库系统工作负载集 W (通常是若干 SQL 查询和更新的集合) 和一定的限制条件 (通常是对存储空间的限制), 找出一个最优的数据库物理配置 (不同存取结构如索引、物化视图、聚簇索引、划分等的集合), 使工作负载集 W 在查询优化器的代价模型估算下代价最小, 达到提高系统运行工作负载性能的目的。

这一问题的直观描述如图 1 所示。图中 client 和 server 端之间的实线界限对应着物理自调优工具的离线工作方式, 两者界限明显。

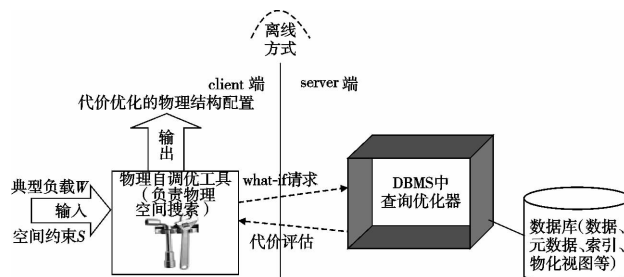


图 1 数据库物理自调优工作示意图

通常物理自调优的工作在客户端,作为一个客户端工具以离线方式完成。它接收的输入有两个,一个是数据库系统典型的工作负载,另一个是空间约束,要求输出的物理自调优配置中所有的物理结构 (如索引、物化视图等) 所占的空间不能超出某一阈值。在工作过程中,它按照一定的物理结构搜索算法与关系数据库的查询优化器进行通信,获得候选物理存储结构的优化器代价,输出具有最优代价的一组物理数据库结构设

计。输出的形式可以是一组 SQL 语句^[5],也可以是一段 XML 脚本^[6],总之可由 DBA 直接运行以实现物理设计。

将数据库物理设计的工作交由工具自动地完成,可以减轻数据库管理员的负担,将其从繁琐复杂的物理存取路径选择的工作中解放出来,进一步减少整个系统的整体拥有成本。

2 自动数据库物理设计的要求

虽然目前不同的数据库厂商提供了不同的数据库物理调优的产品工具,但是在数据库物理设计的自动调优技术方面都有如下一些共同点^[4]:

a) What-if API 及 what-if 分析^[5,7]。因为调优时考虑的物理结构很有可能在数据库中并不存在,而临时创建它们代价太大不可行。所以一个有效的方法就是对查询优化器进行扩充,提供 what-if 的应用编程接口,使其一方面能够模拟并不存在的物理结构,另一方面能够基于这些虚拟的物理结构对查询进行代价估算,就好像它们在数据库中真实地存在一样。

b) 利用系统基于代价的查询优化器并与之保持同步。利用查询优化器估算的工作负载的代价来评价和选择数据库的物理结构。这样做的好处是^[6]:(a)被推荐的数据库物理设计方案一旦被用户接受并实现,还是需要被查询优化器使用;(b)当查询优化器的代价模型改进时,可以自动地直接加以利用;(c)查询优化器的代价模型中考虑的性能因素(如多处理器、服务器内存容量等)均可数据库物理设计自调优所利用。但是这种方法依赖于查询优化器能够准确地估算查询和工作负载的代价,而通常查询优化器估算的代价与工作负载基于实际的物理结构运行时的代价是有所出入的。

自动数据库物理设计的关键技术要满足以下几方面的要求:a)最终推荐的数据库物理设计方案,应该使工作负载的总体代价最小,这个代价通常是指查询优化器估算的代价,即最优代价的要求;b)应该充分适应将来加入新的物理结构或者工作负载容量非常庞大、其中涉及的属性列的个数非常多的情况,在这种情况下,自身的性能不能有大幅下降,即可扩展性的要求;c)自动数据库物理设计的功能一般是通过工具来实现的,作为工具除了实现最基本的功能外,还必须要给用户友好友好的界面,方便用户的使用和管理,即可用性的要求。

另外,随着越来越多的数据库物理自调优工具的出现,势在必行地需要提出比较可行的、体现上述几方面要求的物理自调优工具的性能目标,并且还要有比较合理的评测基准来评测不同的物理自调优工具。

3 实现最优代价的关键技术

要为工作负载中的 SQL 语句集合找到代价最优的物理结构设计,紧密相关的两个问题就是支持多种物理结构,以及多种物理存储结构之上的物理设计方案的搜索算法。

3.1 自动数据库物理设计考虑的结构

在物理设计自调优的早期产品中,一般只简单考虑了索引这一种物理存取结构。但是随着当前数据库引擎能够提供的物理结构种类的增加,除了传统的索引之外,还有物化视图、水平划分、多维聚簇表等,这就要求物理设计的自调优能够既包括数据库现有的多种物理结构,同时又支持将来新的物理结构的加入。

自动的数据库物理设计在考虑多种数据库物理结构时,大致有三种方法:

a)分阶段的方法。即一个阶段只考虑一种物理结构,如针对某一个查询,先考虑对这个查询来说最优的索引,然后考虑可能的水平划分,依次是其他的物理结构。

b)整合的方法。为工作负载中的查询进行自调优时,综合考虑所有种类的物理结构,包括索引、物化视图和水平划分,给出候选的物理设计。Microsoft SQL Server 2005 的 Database Tuning Advisor 采用的就是整合的方法。

c)混合的方法。它先对多种物理结构的相互关系进行分析,根据不同的物理结构之间相互关系强弱程度的不同,在有强依赖关系的物理结构之间采用整合的方法搜索,在依赖关系较弱的物理结构之间采用近似于分阶段的方法。IBM 的 DB2 Design Advisor 采用的是混合的方法。

DB2 Design Advisor 支持的物理结构有四种:索引、物化视图、水平哈希划分、多维聚簇表。通过对这四种物理结构的相互关系进行分析,得到不同物理结构的相互关系表,如表 1 所示^[5]。

表 1 DB2 Design Advisor 对不同物理结构之间的相互依赖关系

A\B	索引	物化视图	水平划分	多维聚簇表
索引		强	弱	弱
物化视图	强		弱	强
水平划分	弱	强		弱
多维聚簇表	弱	强	弱	

基于不同的物理结构之间相互关系强弱程度的不同,DB2 Design Advisor 被分成 IM、P 和 C 三个组件。其中,IM 专门负责推荐索引和物化视图;P 用于专门推荐划分;C 用于专门推荐多维聚簇。这样在同一个组件内部,采用整合的方法考虑多种物理结构,而在不同的组件之间,则采用的是近似于分阶段的方法。比如,索引和物化视图放在一个组件中综合考虑,是因为两者相互之间有着较强的依赖关系(表 1)。而当两个物理结构之间只有单方面较强的依赖关系(如物理结构 A 强烈依赖于物理结构 B,但反过来却不成立)时,在采用分阶段方法的同时,还要保证物理结构 B 在 A 之前进行考虑。因为水平划分较强地依赖于物化视图,但是反过来物化视图较弱地依赖于水平划分,所以针对物化视图的 IM 组件在针对水平划分的 P 组件之前执行。

3.2 负载驱动的物理设计方案的搜索算法

在 3.1 节中针对多种数据库物理结构的不同方法,决定了在物理结构空间中的搜索方法也不同,主要有逐步整合、逐步释放和混合搜索方法。

1)逐步整合的方法 整合考虑多种数据库物理结构时,一种逐步整合的方法采用如下步骤^[8,9]:

a)候选(candidates)物理设计方案的生成。为工作负载中的每一条 SQL 语句进行物理优化,为其中相关的列组合提供多种物理结构的优化存取路径作为候选。一般基于启发式的猜想找出这些候选存取路径。

b)合并候选存取路径。考虑到存储空间的限制和维护的代价,将前一阶段得到的对于单个语句来说优化的存取路径进行合并^[9],得到对于负载中多个 SQL 语句来说较优的物理存取路径。

c)得到最优的物理设计方案。利用一个枚举的算法,从前两步得到的全部候选物理存取路径的集合中,找出针对整个

工作负载来说最优的物理存取路径的集合,即数据库物理设计自调优的推荐结果——一个数据库物理设计方案。

Microsoft SQL Server 2005 的 Database Tuning Advisor 所采用的是这种搜索算法。这是一种自底向上的搜索方法,好处是尽可能保证“优化”的要求,但是随着工作负载中查询越来越复杂,而且新的物理结构的加入使算法的复杂性增大,可扩展性不好。

2) 逐步释放的方法^[4] 逐步释放 (relaxation-based) 的搜索方法是一种自顶向下的策略,它解决了自底向上方法的复杂度高、可扩展性不好的问题。这种方法的改进在于:

a) 候选物理结构的生成阶段不采用类似启发式的猜想或者尝试法生成候选物理结构,而是通过截获查询优化器的优化信息,直接有效地找到能确保为代价最优的物理结构的集合。只是这样的物理结构通常会占用很大空间,超出了物理自调优给出的空间限制条件。

b) 与枚举算法采用的搜索方向相反,新的搜索策略开始于一个代价上最优但是占据很大空间的物理配置,通过有效地转换算法,不断地压缩物理配置所占用的空间,但同时最大程度地减少对期望性能的负面影响。而且作为副产品,在不断转换搜索的过程中,还可以为 DBA 提供不同的物理配置随着空间大小的不同呈现不同性能的分布状态。

3) 混合的搜索方法 IBM 的 DB2 Design Advisor 在用混合的方法考虑多种物理结构的同时,为了得到最终的数据库物理设计方案,针对整个工作负载及其涉及的数据,按照 IM、P、C 的先后顺序,逐个组件地搜索其所代表的全部物理存取路径的设计方案,而且不同组件的搜索算法可以是不同的。

3.3 评价

分阶段这种随机的、临时的方法忽略了多种物理结构之间可能存在的相互依赖关系,另外阶段的分配(即确定先考虑哪种物理结构后考虑哪种物理结构)也是一个重要并关键的问题,如果分配得不合适,很可能会得不到最优的物理设计方案。但是正如文献[5]所提到的,这种方法可扩展性较好,便于将来加入其他种类的物理结构。

整合的方法考虑到多种物理结构之间的相互关系,因此可以保证得到优化的物理设计方案。但是自底向上的逐步整合方法的缺点是,随着系统支持的物理结构种类的增多,或者在工作负载中需要被整合考虑的列的数目越多,都会使自调优时需考虑的搜索空间以指数级增长,因此可扩展性不太好。自顶向下的逐步释放的方法利用查询优化器的介入工作模式,解决了逐步整合方法可扩展性不好的问题,而且可以减少从外部调用查询优化器的次数,优化自调优的性能。但该方法需要依赖查询优化器提供一种新的介入工作模式加以支持。

上述两种方法各有千秋。分阶段的方法便于扩展,但是忽略了不同的物理结构之间可能会有较强的相互依赖关系这一重要特点;整合的方法虽然考虑了不同特性的物理结构之间有较强的相互依赖关系,得到的物理设计方案更接近最优,但是要么由于搜索空间太大,算法的复杂度高,使得可扩展性受到影响,要么需要改动查询优化器的代码以支持介入工作模式。

混合方法的好处是只在相互之间有强依赖关系的有限种物理结构之间采用整合的方法搜索空间,可以使搜索空间不至于过大;在相互之间依赖关系较弱的物理结构之间采用相互独立的调优方法,具有较好的灵活性和可扩展性;另外由于考虑

了多种物理结构之间的相互关系,得到的物理自调优方案是较优的。

为表述清楚起见,将 3.1 和 3.2 节的内容从优化程度、算法复杂度和可扩展性的角度进行评价,如表 2 所示。

表 2 多种物理结构的选择和最终物理设计方案

搜索算法的比较			
方法	优化程度	复杂度	可扩展性
分阶段方法及其搜索算法	可能不优化	低	好
整合方法	逐步整合	高	不好
	逐步释放	最高 需改动查询 优化器代码	好
混合方法及其搜索算法	较高	中等	既具有一定的 可扩展性,又 较灵活

4 实现可扩展性的技术

可扩展性要求物理自调优工具能够在工作负载的规模很大、数据量规模很大、系统负载很重的情况下,进行自动数据库物理设计时,工具的运行不会影响系统正常工作的性能,并且生成的推荐物理配置是代价优化的。常见的技术有负载压缩、生成精简统计信息、系统/测试双服务器工作模式等。

4.1 负载压缩

影响可扩展性的一个重要方面是工作负载。当工作负载中的查询数量呈线性增长时,物理自调优工具自身的执行时间会呈现指数级增长的趋势^[5]。因此为了在自调优本身的性能和质量之间进行权衡,有很多研究围绕负载压缩 (workload compression) 展开。负载压缩需要考虑:对负载的一个规模更小的子集进行自调优与对整个负载进行自调优所达到的优化效果应该接近,或者性能下降不明显。

负载压缩算法要找到满足该条件的规模更小的负载子集。Microsoft DTA 采用的方法是^[6,10]:考虑到负载中查询模板化的内在特点,将整个工作负载基于特征 (signature) 进行划分,只有当两个查询除了语句中的常数不相同其余全相同时,就说它们具有相同的特征;然后基于数据挖掘中聚类的方法从负载的每个划分中提取一个子集,完成负载压缩。

IBM 的 DB2 Design Advisor 采用的方法是基于代价的^[5],其压缩的结果只保留负载中代价最昂贵的 K 个查询,它们的代价总和不超过整个负载在初始状态下总代价的 $X\%$ 。具体方法是根据自调优开始时获得的负载中每个语句的代价估算对语句进行降序排列,直到前 m 个语句的代价达到总代价的 $X\%$,这前 m 个语句构成压缩后的负载集。其中参数 X 采用系统预先提供的值 60、25 和 5,分别对应低、中、高三个不同的压缩级别,通常系统默认为中级的压缩级别。

负载压缩一方面可以从分析负载中的语句特征入手(如 Microsoft DTA),另一方面可以单纯从语句的优化器估算代价角度进行(如 DB2 Design Advisor)。相比较而言,从语句特征入手的方法,可以保证对压缩后的负载进行物理自调优的质量,但它需要借助类似数据挖掘的方法进行聚类分析,计算量比较大^[5],会使物理自调优工具本身的性能受到影响。从语句代价角度压缩的方法比较直接易行,同时压缩后负载中的语句就是占整个工作负载中代价最大的语句集合,而且不会影响物理自调优工具的运行性能,但因为参数 X 只是预先设定的几个固定值,不具有灵活性,所以无法保证负载压缩的优化。

4.2 精简统计信息的创建^[6]

进行 what-if 分析需要获得作为分析对象的物理结构的统计信息,这样才可以不用创建实例化的物理结构而获得查询计划的代价估计。Microsoft DTA 利用 SQL Server 的特点,在为一组列(A, B, C)创建统计信息时,为起始列 A 创建一个直方图(histogram),然后为每一个前缀即(A)、(A, B)、(A, B, C)计算它们的密度(density)信息(即在相应的列组合上,出现的重复值的平均个数)。基于这个特点,在需要为(A)、(B)、(A, B)、(B, A)、(A, B, C)等不同的列的组合考虑 what-if 索引时,只需要对(A, B, C)和(B)创建统计信息即可,不必为所有五种组合都创建索引。

这是一种减少统计信息重复创建的方法,还有一种方法是研究针对某一个统计信息,判断它是否对查询优化器最终选择的查询计划的代价有实际影响。这两种方法可以同时用来精简统计信息的创建。

利用统计信息可以使查询优化器的 what-if 分析避免真正创建数据访问的物理结构,这在数据量很大的情况下非常有效,但是统计信息的创建同样需要付出采样、处理、计算等代价,因此 Microsoft SQL Server 2005 采用设计优化的统计信息和减少重复创建统计信息的方法,对于提高数据库物理自调优的效率很有意义。

4.3 产品服务器/测试服务器的调优方法^[6]

在产品服务器上对大型的工作负载进行自调优,会使产品服务器负担过重。Microsoft DTA 采用测试服务器进行物理自调优来减轻产品服务器的负担,而且只需要先将产品服务器上相关数据库的元数据拷贝到测试服务器,在测试服务器上对工作负载运行 DTA 进行物理自调优。在调优过程中当查询优化器需要用到某些统计信息时,DTA 可将这些统计信息从产品数据库再导入测试数据库,如果产品数据库上没有所需的统计信息,则可以临时从产品数据库上的实际数据创建统计信息。

另一个关键的问题是,产品数据库与测试数据库的硬件配置很可能是不同的。一般说来,产品数据库的配置级别要高一些,而通常查询优化器的代价模型要考虑到底层硬件的因素,如 CPU 数目、内存容量等。测试数据库上运行的查询优化器利用代价模型时不能以测试服务器的硬件配置为准,而必须能够模拟产品服务器相关的硬件配置。SQL Server 2005 扩展了这项功能,使得 DTA 使用 what-if 接口调用查询优化器以获得优化器估计的代价时,将 CPU 数目、可用内存容量等因素作为接口的参数传入,这样就可以实现对产品服务器相关硬件配置的模拟。接下来就可以将测试服务器上运行 DTA 得到的数据库物理设计方案再用于产品服务器上。

产品服务器专门运行系统日常应用,测试服务器专门支持物理自调优工具的运行,可以减少产品服务器的负载压力。测试服务器只需从产品服务器拷贝元数据信息,无须拷贝实际数据,因此启动成本低。Microsoft DTA 只在必要的时候从产品服务器拷贝(或者必要时创建)统计信息到测试服务器,进一步降低了测试服务器的运行成本,也减少了对产品服务器的调用干扰。同样地,这种方法需要对查询优化器的 what-if 接口进行更改,以支持模拟产品服务器与测试服务器的硬件区别。

5 数据库物理自调优工具的可用性与基准测试

1) 高可用性

从数据库物理自调优工具的角度,除了要考虑到基于的核心技术和算法以及工具的可扩展性之外,还要考虑到作为一个工具给用户提供的可用性^[6]。如为 Microsoft DTA 的输入和输出定义一个公共的 XML 模式(XML schema),可以方便地为工具编写脚本,并在其上开发新的工具,也有利于在不同的用户和工具之间交流数据库物理设计信息。Microsoft DTA 还允许用户定制自己想要的那些物理设计结构,DTA 通过自调优生成推荐的物理设计方案时,将用户事先定制的物理设计结构考虑在内。Microsoft DTA 还支持 DBA 用户在循环多次使用自调优工具进行推荐和评价的过程中,不断完善数据库物理设计方案,直到用户最终满意为止。

2) 测试基准和目标研究

文献[11]提出了数据库物理自调优工具应关注的目标和测试基准(benchmarks),提出使用实际执行时间的累积频率(cumulative frequency of the elapsed times of the workload)作为指标来衡量物理自调优工具推荐的物理配置方案的好坏。这个指标表示为 $CF_c(x) = \text{count}(\{q_k : A(q_k, C) < x\}) / \text{size}(W)$ 。其中:W 是工作负载集合, $q_k \in W$; C 代表某个物理配置方案; $A(q_k, C)$ 代表查询 q_k 在配置 C 下的实际执行时间。该指标表示实现了推荐的物理配置方案后的执行工作负载,并统计在某一段时间内完成的语句数目占整个负载的比例。

文献[11]比较物理自调优工具时采用的参照配置是包含了所有单列索引的配置方案。通过实验表明,这种配置方案的实际执行代价要大大优于其他的物理自调优工具推荐的物理配置。另外,文献[11]对物理自调优工具进行基准测试时选择了两个数据库:现实数据库——蛋白质序列数据库和生成的数据 TPC-H 数据库。在设计工作负载时文献[11]采取了模板的形式,根据设计查询的原则设计了五类查询模板用于评测。

文献[11]的实验结论验证了提出的指标 $CF_c(x)$ 的有效性,并且发现全部单列索引的配置方案是性能最优化的方案。

文献[12]对文献[11]的结论进行了质疑,首先是针对指标 $CF_c(x)$ 中采用实际运行时间却忽略了查询优化器的估计运行时间,进行了实验验证,结果表明两个指标的评价能力是非常接近的。针对文献[11]中的 $CF_c(x)$ 指标具有聚集性却忽略了单个查询在不同物理配置下的性能变化,文献[12]又提出一个指标 v_i , 定义为:为 W 中每一个查询 q_i 定义 $v_i = \text{cost}(q_i, C_1) - \text{cost}(q_i, C_2)$, 其中 $\text{cost}(q_i, C)$ 为 q_i 在配置 C 下的代价,将所有查询 v_i 按照取值进行排序并作图,可以看出单个查询在两个不同的物理配置下的性能变化。指标 I 最多只能支持比较两个配置下的查询性能,因此文献[12]建议将 $CF_c(x)$ 指标与 I 结合起来使用。

文献[12]质疑了文献[11]中关于单列索引是最优物理配置的结论,分析了文献[11]中适宜该结论的数据库(小表,每个表的列数中值为 5、多值主索引等)及其查询(无更新)等特点,并通过实验验证了单列索引在很多查询情况下劣于物理自调优工具推荐的配置。

物理自调优工具的基准测试中,数据库和工作负载的生成仍然是一个开放领域^[11]。在这方面,还需要科学的方法和准则来指导选择数据库和工作负载,以便对物理自调优工具的有效性和性能得出客观公正的评价。

6 数据库物理自调优的新问题

最近几年,学者们在数据库物理自调优领域突破了经典问

题框架的限制,进行了探索和创新。

1) 对查询优化器的介入(instrumentation)工作方式^[4]

传统的物理自调优工具借助调用查询优化器的 what-if 接口,无须真正地创建物理结构,通过在服务器端系统目录创建统计信息,利用查询优化器的代价模型,计算并返回并不存在的物理结构对于查询的潜在利益,有助于物理自调优工具推荐真正优化的物理设计。但是当考虑的物理存储结构越来越多,它们之间由于相关性使得物理存储结构的搜索空间越来越大时,会导致频繁地调用查询优化器的 what-if 接口,影响查询优化器的工作性能。

为此一个更简单直接的方法就是查询处理器的介入工作方式^[4]。原因是查询优化器在对查询进行优化时,为查询表达式考虑了现有的物理结构搜索空间上的候选集。新的工作方式在查询优化器发出查询转换的请求时,截获该请求并中断查询优化,通过分析请求中所有有关列上的候选索引,找到组成该请求的最优子查询计划的最优物理结构,并在系统目录中模拟实现它(生成它的元数据和统计信息);之后继续查询优化器的运行,这时查询优化器在优化过程中会考虑新创建的物理结构,保证为查询找到最优的查询执行计划。查询优化的过程中,在系统目录中所有模拟生成的物理结构的集合构成最优的物理优化配置,它可以作为物理空间搜索的起点,采用 3.2 节的逐步释放(自顶向下)的搜索,通过转换算法进行空间搜索。其目标是减少物理优化配置的空间大小,同时保持其性能下降尽可能小,最终返回在空间和性能两方面均比较优化的物理配置。

这种方法的好处是通过截获查询优化器的请求,不会遗漏一些重要的候选物理结构^[13],并且也不需要超出查询优化器的搜索空间考虑一些优化器不会利用的物理结构^[13],减少了对查询优化器的调用,等等。当然,查询优化器需要改动,普通模式之外还要增加一种介入工作模式。

2) 将工作负载从集合扩展为序列^[14]

以往的数据库物理自调优工具将工作负载看做一个由多条查询语句和更新语句组成的集合,但这并不符合实际的情况。如在 OLAP 的应用中,很多情况是白天支持大量的查询操作,在夜晚执行数据的更新操作。这种性质更适合将工作负载看做具有时序性的序列,特别是查询语句或更新语句集合的序列,并对物理数据库调优提出了新的要求,如在查询开始前创建索引提高查询的性能,而在更新语句之前删除索引,以避免数据更新带来的索引维护代价。

这种扩展给数据库物理自调优带来了新的机会,物理自调优的输出也从原来的物理结构的创建/删除语句集合扩展为一个序列,即将这些创建/删除语句夹杂在工作负载的语句序列中。

3) 将物理自调优的约束从单一的空间约束 S 扩展为复杂约束^[15]

传统的物理自调优工具面向的约束非常简单,只有空间约束 S。这种单一的约束不具有灵活性,无法表达更丰富的调优约束,比如要求生成的聚簇索引以指定的列为前缀,或者要求工作负载中每一个查询在经过物理自调优后,比原来配置下的性能降低不超过 10%^[15]等。为了支持这些物理自调优中更丰富的约束表达,文献[15]通过一个专门的约束表达语言,提供简单约束、生成器、过滤器、聚集、嵌套约束、弱约束等结构,具有足够的约束表达能力,并使表达出来的约束很容易与物理

结构空间搜索的转换算法相结合,从而在搜索的过程中将这些约束转换为目标函数,用多目标优化的方法找到物理设计配置的帕累托最优解。

4) 对物理自调优的启动时机进行了深入的研究^[16]

因为物理自调优的工具启动运行很耗系统资源,频繁地启动物理自调优会占用大量系统资源,而得到的性能提高相对有限,稀疏地启动物理自调优又会丧失一些本可以提高负载性能的调优机会。因此一个有效的方法是提供一个轻量级的预警器(alerter)^[17],它提供如下功能:a) 低开销地收集工作负载的信息及准确地诊断当前配置是否需要启动新一轮物理调优;b) 这种诊断是准确量化的,在预警器得出需启动物理自调优的诊断时,同时给出调优后工作负载性能提高的最下限,并给出相应的物理自调优推荐作为该下限的证据,以降低误报率,还可以给出工作负载性能提高的最上限,以降低漏报率。其中预警器通过在查询优化过程中搜集关于工作负载的查询优化信息,可以避免后期诊断时对查询优化器的频繁调用。DBA 根据预警器提供的充分有效的信息,可以根据需要决定何时启动物理自调优工具。

5) 将物理自调优从离线工作方式扩展为在线工作方式^[18]

传统的物理自调优工具以离线的方式工作,需要 DBA 手工获得系统的典型离线工作负载,并手工启动物理自调优工具。这种方法在复杂应用的环境中查询不断变化时操作复杂,很难保证物理调优结果的有效性,并且不易判断何时启动离线的物理自调优。因此专门考虑在线工作方式的物理自调优成为必要。

文献[18]依赖查询优化器的介入工作方式,一方面收集截获在线工作负载的访问请求,采用查询计划局部转换方法考察候选的物理结构对性能改进的空间。在在线的方式下,当这些候选的物理结构在工作负载集上的性能改进足够显著,可以证明物理结构的改变使查询性能提高时,则改变当前的物理配置创建或删除相关的物理结构。这种在线的物理自调优要求其开销尽可能小,不影响 DBMS 的正常工作^[18];同时既要考虑物理配置对在线工作负载的性能影响,又要考虑物理配置变化对性能的负面影响,如创建删除索引均需有一定代价。

6) 多目标优化的数据库物理自调优^[16]

前述物理自调优技术均采用单一目标来为工作负载找到推荐的数据库物理配置——即查询优化器估计的代价最小。文献[16]分析了单一目标的脆弱性:查询优化器选择率估计值不准确时导致的风险,以及运行时工作负载与训练物理自调优的工作负载不一致时导致的不通用性,提出了数据库物理自调优健壮性的目标,将自动物理设计改为多目标(查询优化器的估计代价、风险性与通用性)优化的问题。

文献[16]针对多列上合取谓词的独立性假设考察查询优化器选择率估计值错误导致的风险,提出将查询优化器的普通估计值与最坏情况下的估计值结合起来定义风险。这种方法同样需要修改查询优化器的代码,使其在原来仅返回估算代价之外,还要返回独立性假设不成立时最坏情况下的估计值。

以上这几个新的研究问题为经典的数据库物理自调优研究框架中的元素带来了改变。图 2 总结了在图 1 的研究框架中更新的元素(在图 2 中以 * 开头),可以看出这些新的研究问题并未改变已有的研究框架。图 2 中 client 和 server 端之间的虚线表示通过在线(而非离线)物理自调优,client 和 server 端的界限变得模糊,双方联系更加紧密(在线物理自调优、介

入工作方式和获取访问路径请求均用实线表示这种更紧密的联系)。

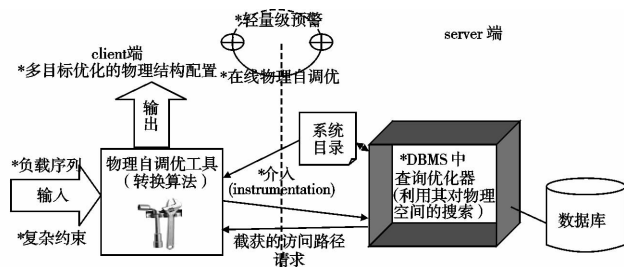


图2 经典的数据库物理自调优研究框架下新的研究问题

7 相关工作和背景

自动的数据库物理设计技术和工具的研究与开发一直是各大数据库厂商和数据库学术界关注的要点。在这方面,Microsoft 和 IBM 的相关研究人员做了很多开创性的工作^[4-6],研究了实现自动数据库物理设计的重要技术,并推出了具有各自特色的数据库物理设计自动调优工具,如 Microsoft 的 SQL Server 2005 的 Database Tuning Advisor、IBM 的 DB2 Design Advisor,其他如 Oracle 的 SQL Access Advisor、BMC 的 Index Advisor 等一些专门的数据库管理系统厂商和第三方的工具提供商,也都提供了相应的数据库物理设计的自动调优工具。

Microsoft 的 AutoAdmin 项目^[19]开始于1997年,目标是使数据库实现自调优和自管理。目前该项目致力于自动的数据库物理设计和关系数据库统计信息的自动管理。在数据库物理设计自调优方面,他们和 Microsoft 的 SQL Server 产品小组紧密合作,将其研究成果与 SQL Server 产品结合,分别成就了 Microsoft SQL Server 7.0 和 SQL Server 2000 的 Index Tuning Wizard 以及 Microsoft SQL Server 2005 的 Database Tuning Advisor。

IBM 的 DB2 Designer Advisor 是 IBM SMART 计划^[1]中的一个研究专题及其成果,专注于数据库物理设计的自调优。SMART (self-managing and resource tuning, 自管理及资源调优项目) 正式开始于2000年春,旨在数据库方面研究研制自调优自管理的 DB2 产品。SMART 是 IBM 更大的自主计算发起活动 (automatic computing initiative) 的一个组成部分。

8 结束语

本文综述了传统的关系数据库物理自调优的研究工作,总结了数据库物理自调优的经典技术,并从优化程度、可扩展性、高可用性以及测试基准四个方面对物理自调优产品的经典技术进行了比较和分析。通过总结分析最近几年来物理自调优领域的新的研究工作可以看出,在新的云计算、社会计算和大数据的环境下,以及新的网络应用模式(如社会网络、电子商务等)下,这一领域未来的研究方向面临着更多的机会和更新的挑战^[13],如轻量级的自调优工具、互联网分布式体系结构(包括分布式存储、分布式计算和分布式数据分析^[13])下的物理自调优等。这些新的环境需要将数据库物理自调优问题进行重新定义,推动研究人员在新的框架下研究关键技术。

参考文献:

- [1] LOHMAN G M, LIGHTSTONE S. SMART: making DB2 (more) automatic [C]// Proc of the 28th International Conference on Very Large Data Bases. [S. l.]: VLDB Endowment, 2002: 877-879.
- [2] FINKELSTEIN S, SCHKOLNICK M, TIBERIO P. Physical database

design for relational databases [J]. ACM Trans on Database Systems, 1988, 13(1): 91-128.

- [3] RAO Jun, ZHANG Chun, MEGIDDO N, et al. Automating physical database design in a parallel database [C]//Proc of ACM SIGMOD International Conference on Management of Data. New York: ACM, 2002: 558-569.
- [4] BRUNO N, CHAUDHURI S. Automatic physical database tuning: a relaxation-based approach [C]//Proc of ACM SIGMOD International Conference on Management of Data. New York: ACM, 2005: 227-238.
- [5] ZILIO D C, RAO Jun, LIGHTSTONE S, et al. DB2 design advisor: integrated automatic physical database design [C]//Proc of the 30th International Conference on Very Large Data Bases. 2004: 1087-1097.
- [6] AGRAWAL S, CHAUDHURI S, KOLLAR L, et al. Database tuning advisor for Microsoft SQL Server 2005: demo [C]//Proc of the 30th International Conference on Very Large Data Bases. 2004: 1110-1121.
- [7] CHAUDHURI S, NARASAYYA V. AutoAdmin "what-if" index analysis utility [C]//Proc of ACM SIGMOD International Conference on Management of Data. New York: ACM, 1998: 367-378.
- [8] AGRAWAL S, NARASAYYA V, YANG B. Integrating vertical and horizontal partitioning into automated physical database design [C]//Proc of ACM SIGMOD International Conference on Management of Data. New York: ACM, 2004: 359-370.
- [9] AGRAWAL S, CHAUDHURI S, NARASAYYA V R. Automated selection of materialized views and indexes for SQL databases [C]//Proc of the 26th International Conference on Very Large Data Bases. San Francisco: Morgan Kaufmann Publisher, 2000: 496-505.
- [10] CHAUDHURI S, GUPTA A K, NARASAYYA V. Compressing SQL workload [C]//Proc of ACM SIGMOD International Conference on Management of Data. New York: ACM, 2002: 488-499.
- [11] CONSENS M P, BARBOSA D, TEISANU A, et al. Goals and benchmarks for automatic configuration recommenders [C]//Proc of ACM SIGMOD International Conference on Management of Data. New York: ACM, 2005: 239-250.
- [12] BRUNO N. A critical look at the TAB benchmark for physical design tools [J]. ACM SIGMOD Record, 2007, 36(4): 7-12.
- [13] CHAUDHURI S, NARASAYYA V. Self-tuning databases systems: a decade of progress [C]//Proc of the 33rd International Conference on Very Large Data Bases. [S. l.]: VLDB Endowment, 2007: 3-14.
- [14] AGRAWAL S, CHU E, NARASAYYA V. Automatic physical design tuning: workload as a sequence [C]//Proc of ACM SIGMOD International Conference on Management of Data. New York: ACM, 2006: 684-694.
- [15] BRUNO N, CHAUDHURI S. Constrained physical design tuning [J]. Proceeding of VLDB Endowment, 2008, 1(1): 4-15.
- [16] GEBALY K E, ABOULNAGA A. Robustness in automatic physical database design [C]//Proc of the 11th International Conference on Extending Database Technology. New York: ACM, 2008: 145-156.
- [17] BRUNO N, CHAUDHURI S. To tune or not to tune? A lightweight physical design alerter [C]//Proc of the 32nd International Conference on Very Large Data Bases. [S. l.]: VLDB Endowment, 2006: 499-510.
- [18] BRUNO N, CHAUDHURI S. An online approach to physical design tuning [C]//Proc of the 23rd International Conference on Data Engineering. Washington DC: IEEE Computer Society, 2007: 826-835.
- [19] Microsoft Research. AutoAdmin [EB/OL]. (2011-09). <http://research.microsoft.com/en-us/projects/autoadmin/>.