

数据库物理结构优化技术*

崔跃生^{1,2}, 张勇², 曾春², 冯建华¹, 邢春晓²

¹(清华信息科学与技术国家实验室(清华大学) 计算机科学与技术系, 北京 100084)

²(清华信息科学与技术国家实验室(清华大学) 信息技术研究院, 北京 100084)

通讯作者: 崔跃生, E-mail: yuesheng797@163.com

摘要: 面对快速增长的数据量和复杂的查询请求, 关系数据库要满足用户检索的高效性, 不能仅仅依靠 SQL 查询优化, 必须针对不同的应用背景, 对数据库的物理结构进行优化, 从底层提高数据库的检索查询效率. 描述了 4 种已被商业数据库优化的物理结构, 总结了物理结构优化领域的关键技术, 并介绍了商业数据库中使用的数据库物理结构优化推荐工具, 最后展望了未来的研究方向.

关键词: 物理结构; 优化; 推荐; 工作负荷; 抽样; 如果假设分析; 组合爆炸

中图法分类号: TP311 **文献标识码:** A

中文引用格式: 崔跃生, 张勇, 曾春, 冯建华, 邢春晓. 数据库物理结构优化技术. 软件学报, 2013, 24(4): 761-780. <http://www.jos.org.cn/1000-9825/4355.htm>

英文引用格式: Cui YS, Zhang Y, Zeng C, Feng JH, Xing CX. Database physical structure optimization technology. Ruanjian Xuebao/Journal of Software, 2013, 24(4): 761-780 (in Chinese). <http://www.jos.org.cn/1000-9825/4355.htm>

Database Physical Structure Optimization Technology

CUI Yue-Sheng^{1,2}, ZHANG Yong², ZENG Chun², FENG Jian-Hua¹, XING Chun-Xiao²

¹(Department of Computer Science and Technology, Tsinghua National Laboratory for Information Science and Technology (Tsinghua University), Beijing 100084, China)

²(Research Institute of Information Technology, Tsinghua National Laboratory for Information Science and Technology (Tsinghua University), Beijing 100084, China)

Corresponding author: CUI Yue-Sheng, E-mail: yuesheng797@163.com

Abstract: In face of the growing data volume and search requests, to meet the efficiency of users' search requests, the database cannot depend just on the SQL query optimization. Improvement must be made upon the physical structures of databases and the search efficiency from the origin. The paper describes four physical structures optimized by commercial databases, summarizes several key technologies in physical structure optimization sphere and introduces several recommendation tools for physical structure optimization now utilized in commercial database. The research directions are presented at last.

Key words: physical structure; optimization; recommendation; workload; sampling; what-if analysis; combinatorial explosion

关系数据库(后面所有的数据库均指关系数据库)的物理结构、查询优化器和执行引擎是影响数据库查询性能的主要因素. 数据库物理结构的改变虽然不会影响应用的查询结果, 但会影响数据库性能. 早期的关系数据库主要用于联机事务处理, 数据规模较小, 通常进行查询语句和执行引擎的优化, 物理结构设计的重要性没有充分得到体现.

近年来, 随着互联网技术的发展, 企业信息化的不断深入, 信息系统中的数据日益膨胀. 如何从历史数据中获取规律性以指导企业的规划和决策, 促进数据的有效利用, 已成为信息化建设新的发展方向. 因此, 数据库系

* 基金项目: 国家自然科学基金(61170061); 国家重点基础研究发展计划(973)(2011CB302302)

收稿时间: 2012-07-04; 修改时间: 2012-09-12; 定稿时间: 2012-12-03; jos 在线出版时间: 2013-01-17

CNKI 网络优先出版: 2013-01-17 15:23, <http://www.cnki.net/kcms/detail/11.2560.TP.20130117.1523.001.html>

统在大规模的数据处理领域(如多媒体、电子商务、数据挖掘、决策支持系统和地理信息系统等)获得了广泛应用.另外,得益于互联网、云计算、移动计算和物联网的迅猛发展,现在的企业能够源源不断地获得各种各样的数据.面对当前的商业应用,数据管理成为当今软件产业的一个具有挑战性的话题,数据已经不仅仅是驱动企业运行的动力,而且是企业发展和创新的基础,因此,数据已经成为企业的核心资产.2011年,全球知名咨询公司麦肯锡的全球研究所发表研究报告^[1]指出:当前,数据已经渗透到各个行业的业务职能领域,逐渐成为重要的生产要素;人们对海量数据的运用表明,新一波生产率增长和消费者盈余浪潮的到来——“大数据时代”即将到来.通常,我们把超大规模的数据定义为大数据(big data),然而现在我们所要面对的大数据已经超越了传统意义上的海量数据,因为它们不仅在数量上十分巨大,而且还具有更复杂的特点:丰富的数据种类、复杂的数据结构、紧密的数据关联度、快速的数据变化性、需要快速处理保持时效性等.在这样的背景下,数据库系统的应用发生了巨大的变化,传统的数据组织方法在针对某些新特点上遇到了瓶颈,因此我们需要使用一些高效的、灵活的、健壮的和低成本的数据物理结构.

为了适应数据库系统应用的新特点,目前传统关系数据库使用了索引、物化视图、无共享分区和多维群集等物理结构,一定程度上解决了关于海量存储、快速访问和数据的多维性等问题.随着技术的进一步发展,今后或许还会提出更多更优的物理结构.由于这些物理结构在选择上具有多样性和复杂性,而且数据库不可能在有限的存储空间里存储所有的物理结构,只能在满足一定存储约束条件的情况下,选择存储最优的物理结构配置.因此,数据库管理员(DBA)已经不能仅是简单地依赖优化查询语句和执行引擎来进一步提高数据库系统的性能,优化传统数据库系统的物理结构设计(也称为物理数据库设计)^[2-6]变得尤为重要.

物理结构设计是在逻辑结构设计之后进行的.物理结构优化的核心目标是基于逻辑设计确保数据库存储和访问数据表的方法或操作具有高效性,使所有基于关系数据库的应用程序最大限度地提高整个系统的数据访问速度.然而,物理结构设计往往涉及到几十个甚至上百个经常使用的元素或变量,其中大部分变量之间还是相互关联的^[3],存储系统优化或性能分析通常是在不同负载条件下比较哪个物理配置更优,这个过程可以被归纳为已被证明为 NPC 问题的组合优化问题^[5-7].因此,在物理结构推荐优化的过程中,不可能使用穷举法来实现最优物理配置的快速而有效的选择.

如何为数据库选择一个最优的物理结构,已成为数据库领域的一个研究热点.目前,主流商业数据库都已经研究和开发了数据库物理结构的性能调优工具,如 IBM 的 DB2 设计顾问、微软的数据库调优顾问以及 Oracle 的 SQL 访问顾问和调优顾问等,这些工具主要负责数据库的物理结构调优和性能分析.

1 推荐优化模型

数据库的物理结构是整个数据库存储的基础,主要在设计阶段确定而且具有持久性,也就是说,物理结构一旦确定,不会经常变化.一般在以下情况下会发生物理结构的优化推荐^[8]:

- 数据被移植到一个新的存储系统;
- 数据库的应用模式有了较大的变动,主要指的是元数据的变化;
- 数据库的工作负荷(workload)有很大的变动,主要指的是查询语句的使用变化;
- 数据库性能明显降低.

物理结构优化问题就是找到一个物理配置(创建数据库时需要设计的一组物理结构,可以是索引、物化视图、分区和多维群集等,也可以是它们任意的组合配置),使一组已知工作负荷在这个数据库上的执行效率达到最优(或者接近最优),同时,这个配置要满足一定的存储约束.

数据库物理结构推荐优化的数学模型^[4]如下:

已知一个数据库 D , 一个用户给出用于分析的工作负荷 W 和一个存储上限 S , 找出一个物理配置 P , 使其存储空间没有达到 S ($\sum_{Q \in W} F_Q < S$), 并使 $\sum_{Q \in W} Cost(Q, P)$ 是最小的, 同时保证 P 是一致的

在这个模型中,工作负荷 W 主要是指用户访问的 SQL 语句集合,包括插入、更新和查询等语句.存储上限 S 是指创建该物理结构所需要的额外存储空间的上限.物理配置 P 是指推荐产生的物理结构集合.整个模型表示的就是在满足存储约束的情况下找到收益最大(代价最小)的物理结构配置.

2 物理结构优化的元素

数据库物理结构优化是指为数据库系统的数据推荐合适的物理存储位置或存储结构,以及为查询推荐合适的存取路径,进而改进整个系统的性能.在数据仓库还没有发展起来之前,物理结构优化的早期产品一般只考虑索引这一种物理结构的优化.但是,随着互联网等技术的发展,数据仓库技术的广泛应用,为了进一步提高系统的性能,数据库系统逐渐提供了对其他物理结构种类的支持,除了传统的索引之外,还有物化视图、无共享分区和多维聚集等.

在传统关系数据库中,索引是一种与表有关的数据库结构,可以使对应于表的某些查询语句执行得更快.而创建索引需要额外的存储空间,当对表中的数据进行增加、删除和修改时,索引也要动态地维护,降低了数据的维护速度.索引的选择问题(ISP)一直是数据库物理结构优化领域的一个研究焦点.

随着互联网、物联网和云计算等技术的蓬勃发展,数据量呈爆炸式增长,此时,数据仓库技术经过多年的研究与发展,已经广泛应用于各个行业.随着时间的推移和数据的累积,执行查询和分析的连接和聚合等操作变得更加复杂,耗时也越来越多.通过创建索引等简单物理结构已无法有效解决查询响应所需时间越来越长的问题,物化视图技术应运而生.物化视图和索引同样面临着存储和维护的成本问题,由于其选择过程比较复杂,逐渐成为数据库物理结构优化领域的另一个热点.

在数据存储技术高速发展的过程中,持续高并发读写的数据库系统面临的一个主要难点是高可用性.解决这个问题一个办法是使用已经被许多互联网公司采用的 NOSQL 数据库,这种数据库为了增强数据库的可用性和可扩展性,牺牲了交易型事务的 ACID 特性.在一致性需求不高的应用中,这种方法可以给用户一个满意的设计方案;但在面对那些如财务或订单处理的企业高端数据时,NOSQL 数据库无法实现有效支持.因此,一类新型的并行数据库系统 NEWSQL 出现了^[9],这种数据库利用 OLTP 工作负荷的可划分性来实现可扩展性,而且不会牺牲数据的 ACID 特性.这种数据库设计过程中的一个关键技术就是无共享分区.大多数面向数据仓库的数据库系统都支持无共享分区,比如 Greenplum,Vertica,Asterdata,Paracel,Netezza 和 Teradata 等.除此之外,传统关系数据库 DB2 也支持无共享分区^[10].无共享分区可以有效增强系统的并行性和可扩展性,然而解决数据倾斜问题是无共享分区设计的难点,仅依靠数据库管理员确定最优的分区设计是比较困难的.

随着关系数据的增长,促进了对大量数据进行操作和复杂查询的不断研究,特别是联机分析处理 OLAP 和决策支持系统 DSS 在数据挖掘和商业分析领域的广泛应用.OLAP 和 DSS 系统的特点是对企业数据的多维分析以及典型的事务查询应用,如分组、聚合、范围查询、立方体、向上挖掘和向下挖掘等.针对这些特点,利用多维数据群集可以使性能有显著的提升.但是群集维度和群集粒度的选择并不是一件很容易的事情,即使对于有丰富经验的 DBA 或者数据库专家而言也具有一定的难度.

综上所述,几种物理结构面对不同的应用在一定程度上可以解决部分访问的性能问题,由于各种物理结构设计的复杂性,它们都存在如何优化的问题.另外,针对当前数据库系统应用面临的新特点,关于非结构化数据等其他数据种类以及如何保持数据的时效性等方面,还有待提出新的优化存储技术来加以解决.因此,物理结构优化产品要能够支持多种物理结构的优化,而且有必要支持新的物理结构优化的扩展.下面,我们分别总结已有的数据库物理结构的特点及其优化过程与方法.

2.1 索引

索引(index)是数据库系统中用于提高检索查询速度的数据组织,一般情况下可以显著提升数据库系统检索查询的性能,也可以快速检索特定数据和范围数据以及提升数据值唯一性等.通常在索引的选择过程中,除了选择最好的单一属性索引之外,还需要逐渐考虑附加索引,以确定是否有更好的选择方案.有些情况下,虽然创建新的索引提升了查询效率,但却严重降低了更新性能.因此,当判断一个新的索引是否可用时,通常要考虑查

询和更新的性能组合.索引设计过程中需要考虑的关键问题包括:(1) 增加一个索引是否会对检索有益;(2) 使用什么类型的索引;(3) 索引需要包括哪些属性的组合;(4) 考虑索引对存储空间的需求;(5) 在表更新时,索引是否还是有益的.综合考虑上面的几个因素,当一个索引的收益大于其开销时,就应当创建并使用该索引,其中的收益和开销都依赖于响应时间和吞吐量.当一个已用索引的性能变得较差时,需要选择优化新的设计方案.

索引优化是物理结构优化领域最早开始研究的问题.因此,关于 ISP 的算法有很多.文献[11]提出解决 ISP 问题的两种方法:一种是基于背包子问题和基于拉格朗日分解的启发式算法,这种方法适合应用在大规模 ISP 问题上;另一种方法是基于线性规划的分支界限法,这种方法通过预处理过程降低候选索引集的大小来提高索引的选择效率.而文献[12]中提出了改进的分支切面法,把 ISP 问题归约为装箱问题.为了解决空间搜索的效率问题,后来又提出各种基于启发式算法的解决方案,如文献[13]中使用了遗传算法等.在 ISP 问题上,以上方法都存在效率低下和优化质量不高的问题,比如启发式算法在索引方案产生之前,由于候选索引各自收益或者价值的相互作用,导致其最后得到的不一定是最好的优化方案.更重要的是,自动优化技术大都依赖于如果假设分析接口,实现对某个查询的索引集合的收益的精确评价,如果工作负荷集比较大,再加上针对每个查询的巨大的解决方案集合,整个优化过程就会依赖于大量的如果假设调用,显著增加了优化推荐过程的代价或成本.为进一步解决效率问题,文献[14]提出了一种截然不同的优化方法,利用 60 多年前在线性规划领域的研究成果来解决 ISP 问题,证明了基于快速如果假设分析的索引优化问题等价于简洁的二进制整数规划(BIP)问题.BIP 的引出,使优化过程只需进行少量的如果假设调用,利用线性规划现有的方法高效地解决了索引的选择优化问题.基于线性规划的方法解决物理结构优化问题近几年才开始广泛研究和应用,如在文献[15,16]中也使用了该方法.除了大量关于选择算法的研究外,还有很多针对选择过程的研究,目前,索引优化的过程主要分为脱机优化、联机优化和半自动优化这 3 种^[17],适合于不同种类的应用.

当前,大型商业数据库中已有许多索引优化工具,这些工具均是通过评价每一个可用索引的潜在收益的方式来确定最优索引的.然而因可用方案的组合数量庞大,综合评估每一个可能的索引是不现实的,因此,市场上不同产品之间的区别是用于搜索新的索引的复杂程度.用的最多的方法是选择基于工作负荷的谓词以及可能在结构集列上的索引,排除了其他的列,就可以极大缩小搜索的范围,而且推荐优化效果也非常不错.

2.2 物化视图

物化视图(materialized view)是用于保存在基表上进行连接或聚集等耗时较多的操作结果的一种数据库对象.物化视图存储基于原始数据表的检索数据,这样在执行查询时就可以直接访问物化视图,避免重复进行耗时的操作,从而加快得到结果的速度.

利用物化视图能够显著改善磁盘 I/O 和检索查询性能,主要来源于以下 4 方面因素:(1) 利用专用磁盘生成视图,避免 I/O 资源的竞争;(2) 物化视图可能会比基表小很多,这为每次查询的磁盘 I/O 性能带来很大的收益;(3) 降低多表之间的连接成本,减少计算量;(4) 对于经常访问的查询来说,收益倍增.物化视图对应用透明,增加和删除物化视图不会影响应用程序中 SQL 语句的正确性和有效性.物化视图的缺点是需要占用存储空间,当基表发生变化时,物化视图也应刷新.同步更新问题严重降低了物化视图的使用性能,因此在设计和使用时必须考虑更新性能.

在设计物化视图的过程中需要考虑 3 个主要因素:查询成本、存储成本和维护成本^[18],因此要注意以下特点:(1) 不是所有视图都会提高用户的查询效率,通常只需要选择一些关键的视图进行物化,也就是重点关注响应常见查询的视图;(2) 需要存储物化视图的磁盘空间的大小是另外一个显著的约束;(3) 物化视图的数据来源于某些基表,视图的更新具有一定的维护成本,而且需要考虑何时更新以及如何更新.十几年来,很多研究人员对于物化视图选择与管理问题进行了大量深入的研究,具有代表性的有基于多维数据格(multidimensional lattice)的贪婪算法(如 BPUS 算法和 PBS 算法)、遗传算法和模拟退火算法等.关于数据仓库中的视图物化选择问题最早由 Harinarayan 提出,他在文献[19]中对此问题进行了系统阐述,并提出了 BPUS 算法,该算法当时被证明可以取得较好的结果.但该算法的时间复杂度较高,算法的运算时间会随着维数的增加而飞速增加.因此,Shukla 等人于 1998 年在文献[20]中提出 PBS 算法,继承 BPUS 的线性代价模型,但其运行速度要比 BPUS 快

几个数量级.PBS 算法虽然在较大程度上降低了算法的复杂度,但其运行时间随着维度的增加仍然呈指数级增长,不具有实用性.基于 BPUS 算法的一些其他变种的贪婪算法也有很多,比如文献[21,22]等,这些算法都是围绕着 BPUS 进行适当的改进.随着启发式算法的不断发展,有很多文献提出用遗传算法^[18,23]来解决该问题.对于视图选择问题,遗传算法虽然可以提高效率,但其本身并不保证能够获得好的近似最优解,该方法效果的好坏取决于很多因素,比如正确的问题定义、算法的设置以及繁琐的算法参数调整.与遗传算法类似,文献[24]提出采用模拟退火算法来解决视图选择问题.由于模拟退火算法的快速性,因此可以应用在维度较高的物化视图选择问题中.随着数据仓库系统的运行,用户查询请求的动态变化会导致物化视图集的部分物化视图收益下降,部分未被物化的视图收益上升,使得物化视图集的总收益下降,所以必须通过动态选择算法加以解决.为了满足动态优化,文献[25]在 BPUS 算法的基础上提出了基于单位空间上查询频率的即时调整的算法 FPUS,该算法根据收集到的查询分布情况对物化视图进行动态调整,具有一定的创新性.但 FPUS 算法未考虑视图间的依赖关系,忽略了物化视图的维护时间.另外,还需要频繁地进行全体物化效益的比较,特别是对于查询密度很高的情况不能很好地适应.除了上面介绍的重要优化方法之外,还有一些数据挖掘算法^[26]、混合算法^[27]、基于蚁群优化的算法^[28]、基于线性规划的算法^[29]和基于 AND-OR 图的算法^[30]等等,也被应用在解决物化视图的选择问题上.

无论是静态还是动态视图选择算法,它们均存在各自的弊端.对于静态物化视图选择而言,其运行效率低下而且不具有动态自适应性,无法应用于较大的数据维度的选择问题中;对于动态视图选择而言,往往需要耗费昂贵的缓存空间,当缓存的视图片段无法命中时,其性能也会明显下降,这会导致查询时间出现较大抖动.因此,为了解决已有算法的问题,未来的物化视图选择算法要具备易用性、高效性和自适应性这 3 个特点.DB2,Oracle 和 SQL Server 等大部分优化推荐工具都支持物化视图的自动优化.

2.3 无共享分区

无共享分区(shared-nothing partitioning)对于在一个大的数据集上解决计算复杂性的问题是一种分而治之的策略.在并行数据库管理系统中,一个重要的资源管理问题就是数据库的布局,即数据分布.一些早期的研究表明,无共享并行数据库系统的可扩展性和可靠性为系统中多个节点间分发数据提供了有利条件.无共享系统使用一个相对独立的服务器集合在问题域上进行协同工作,每个服务器处理问题域的一个子集,有时,某些服务器之间需要共享数据,那么这些操作可以通过高速网络互连来执行^[31].

使用无共享分区的最终目标就是使大型数据库环境下的所有服务器节点都工作在一种并行的模式下,而且尽量降低在节点之间共享数据的程度.如果数据分布不合理,在查询处理过程中,系统的并行性将不能得到充分的发挥,浪费了系统资源,增加了查询处理时间,从而降低了系统性能^[32].因此,设计无共享分区需要解决以下几个问题:(1) 为了使无共享分区变得有效,数据需要在节点之间被均匀地分布,因为整个系统的运算时间受响应速度最慢的节点所限制.另外,选择使数据在所有节点之间均匀分布的分区列是非常困难的;(2) 为了使无共享分区有良好的并行扩展,节点之间的通信应该尽可能地降低;(3) 针对那些数据量不是很大的关系表采用分区策略不但没有提高效率,反而会因为增加并行化的维护和计算成本而降低了查询性能.无共享分区的大多数方法都是基于数据聚集的概念,依据应用的工作负荷的大规模统计分析,目标是把数据均匀分布在各个节点上,最大限度地提高并行处理能力.无论是在应用还是理论的研究上,自动划分技术都是数据库领域研究的热点.在应用方面,最著名的成就包括 Microsoft's SQL Server 的 AutoAdmin^[33-38]和 IBM 的 DB2 Database Advisor^[8,31,39,40].这两种工具优化推荐的基本结构类似,而不同之处主要包括候选划分属性集的选择方法和寻找最优划分方案的搜索过程.Schism^[41]是近年来出现的划分工具,它利用基于图的划分算法实现分区的最优化.而在理论研究方面,关于搜索过程提出了很多启发式算法,比如遗传算法^[42,43]、爬山算法^[44]和分支界定法^[35]等,也有文献提出使用近似算法^[41].然而针对 OLTP 系统,这些方法并不是很适用,因为它们没有考虑节点间的相互作用,在这种类型的并行系统中,多节点协调工作所需要达到的事务一致性会严重影响整个系统的性能表现.文献[9,10]中指出,即使忽略了锁竞争的影响,这种并行系统和单节点系统相比,一个事务的执行总代价要增加 50%.因此,文献[45]提出了一种自动数据库划分算法,基于 large-neighborhood search 技术和新的分析成本模型,可以根据负载分布实现对节点间同步协调成本的估计.这种算法首次在数据库划分过程中考虑了节点之间的

同步协调成本。

目前,只有 IBM DB2 的产品提供无共享分区的自动化设计工具,为了解决偏斜问题,DB2 依靠哈希函数以及哈希映射来进行平衡,同时关注于那些数据粒度较少的属性.针对数据量较小的关系表,用复制策略代替划分策略,通过冗余的方法实现并行化操作。

2.4 多维群集

多维群集(multidimensional clustering,简称 MDC)是一种利用多维立方体的形式组织数据表,并提供根据多个维对数据进行灵活群集的方法.多维群集主要应用于大型数据库环境,在联机事务处理和决策支持系统等数据挖掘和商业分析领域具有广泛的应用.多维群集允许同时根据多个键或维来对表进行物理群集,通过扫描任意的维索引提供已群集数据的存取,大量减少 I/O 成本和 CPU 成本,使检索性能显著提升^[46-48]。

从一个有限集中选择群集维度的问题,可以很容易地被建模成一个简单的组合问题.但多维群集的问题比这还要复杂,因为每个维度都有不同数量的群集的粒度.比如,一个基于时间的群集可以按照年、月、周、天等等粒度进行设计,那么假设一个设计过程中涉及到 n 个可选维度,而且每个都有 m 个级别的粒度,那么可能群集的选择空间就有 m^n 个,因此,选择群集维度的搜索空间是非常巨大的.Lightstone 与 Bhattacharjee^[46]在 2004 年使用了一个简单模型来应付 MDC 的选择,该模型建立在如下两点上:(1) 当数据库表中仅包含一个元素时,那么 MDC 的设计没有任何价值;(2) 当维度具有很多可能时,随着每个维度的粒度越来越细,MDC 表中的单元数量将会增加,这同时也会造成存储空间增加.对于任何候选维度都存在一个最为有用的维度粒度(finest useful dimension granularity,简称 FUDG),高于这个维度粒度基本不会获得最优的综合收益.这个 FUDG 可以通过 SQL 优化期内的模拟来确定.在 FUDG 确定后,可以构造一个基于在一维和 FUDG 之间推荐的优化模型。

目前,在数据挖掘领域,MDC 技术蓬勃发展,出现了很多优秀的论文和工具.但在商业数据库领域,采用 MDC 技术的只有 IBM 的 DB2,通过显著降低磁盘 I/O,对索引进程和数据转出提供了非常大的性能提升,尤其是对 MDC 表的设计,提高存储块中的数据密度,使表对存储的需求更加合理.MDC 是一种新的、强大的物理数据库设计技术,在未来的数据库应用中会得到更多的发展。

3 物理结构优化的关键技术

前面第 1 节介绍了物理结构优化的数学模型,优化推荐工具或优化推荐算法都是基于这个数学模型展开的,虽然在某些环节有各自不同的处理方法,但总结过去的文献可以发现,一些相对通用的关键技术物理结构优化领域获得了广泛应用,对提高优化推荐的质量和效率具有关键性的作用.下面分别总结几种关键技术的应用背景、技术特点和主要算法及流程。

3.1 工作负荷压缩技术

3.1.1 应用背景

如何使用一个数据库系统,对于数据库系统优化和管理是非常重要的.在传统关系数据库中,数据访问的特征以一种特殊的形式被表示,即工作负荷,它是一组 SQL 语句的集合.在过去的几十年里,出现了很多基于工作负荷驱动的数据库系统优化和管理的应用,比如统计直方图调优(histogram tuning)、改进查询优化(improving query optimization)、索引选择(index selection)、物化视图选择(materialized view selection)等,影响这些应用可扩展性的一个重要因素就是工作负荷的大小^[49].为了解决此问题,出现了关于工作负荷压缩技术的研究.在物理结构优化的数学模型中,工作负荷压缩技术是针对模型输入之一(工作负荷 W)的关键技术。

3.1.2 技术分析

工作负荷作为物理结构优化数学模型中的重要输入,其质量的好坏直接决定推荐优化过程的性能和推荐优化结果的质量.而在数据库的使用过程中,每天可能生成成千上万次 SQL 语句的执行,在这些 SQL 语句中,有些语句之间可能具有非常少的语法差别,有些可能和其他语句相比则完全不同.虽然提供详细的工作负荷是为了使数据库物理配置在今后执行过程中获得更好的执行性能,但是由于工作负荷的大小和优化推荐的性能具

有指数级别的关系,如果工作负荷数量过大,则推荐优化过程很难完成.因此,要处理好工作负荷规模的有效性和高效性之间的矛盾,既要确保压缩后的工作负荷集能够全面代表未被压缩过的工作负荷集,还要确保压缩过的工作负荷集的处理效率明显高于未被压缩过的工作负荷集.图 1 是工作负荷压缩的示意图^[49],如果一个工作负荷压缩是有意义的,那么就需要确保两个条件:第一, W 的压缩时间加上应用程序对 W' 的执行时间小于应用程序直接对 W 的执行时间;第二,最终应用程序执行的结果 R 和 R' 在质量上没有太大的区别.

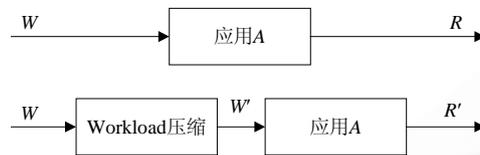


Fig.1 A flow diagram of workload compression

图 1 工作负荷压缩流程图

现在比较流行的工作负荷压缩技术主要有两种:一种是 Chaudhuri 等人^[49]在 2002 年提出的基于距离的工作负荷压缩方法(WC-KMED).该方法通过分析原始工作负荷,基于距离函数(distance function)找到与当前 SQL 语句相似的 SQL 语句类别(不同应用可以设定不同的阈值,若小于该阈值,则认为相似),然后从原始工作负荷中保留具有代表性的一个 SQL 语句并删除其他相似的冗余 SQL 语句.简言之,即合并那些语法相似只是参数不同的语句.另一种是 Zilio 等人^[39]在 2004 年提出的内嵌在优化顾问中的工作负荷压缩方法,当工作负荷较大时,优化顾问才调用工作负荷压缩模块,否则不调用.在该压缩方法中,原始工作负荷经过压缩,最终只保留负载最大且代价最昂贵的 K 个语句而忽略其他语句.其主要思想是,把优化推荐过程中的收益集中体现在代价较大且权重较高的语句的优化上.这种方法虽然比较简单而且可以有效降低工作负荷的数量,但有一个致命的弱点,就是在语句的成本出现倾斜的情况下会出现糟糕的结果.为了解决这个问题,Zilio 提出首先找到类似查询,定义其成本开销为其代价与类似查询个数的乘积.然后再使用上面提出的方法进行压缩.这两种方法中,第 1 种关注于工作负荷中最具有代表意义的子集,第 2 种方法关注于工作负荷中最重要的子集.在压缩过程中,为了降低搜索计算时间,他们还采用了 K -Mediod 和 All-Pairs Greedy 等搜索算法来提高压缩效率.除了上面两种主流方法之外,Chaudhuri 还提出了随机抽样法(WC-PARTSAMP)和贪婪压缩法(WC-ALLPAIRS).随机抽样法不需要输入参数,但这种方法容易剔除重要的 SQL 语句对物理结构优化的结果造成严重影响;而贪婪压缩法虽然不具有随机抽样法的弊端,但其时间复杂度相对较高.

3.1.3 小结

在优化推荐过程中,工作负荷压缩技术是在前端降低优化推荐复杂度的有效手段,对于提高优化推荐性能和质量具有至关重要的作用.但现有的压缩算法要么压缩质量不好,要么压缩效率不高,不同的应用可以根据自身的特点选择合适的算法.另外,针对复杂的决策支持工作负荷(比如 TPCH),由于 SQL 语句中谓词数量较多,基于已有方法会获得非常低下的压缩比.因此,能够提出既满足高效率又达到高质量的压缩算法,是工作负荷压缩技术未来的主要研究方向.

3.2 如果假设分析(What-If analysis)技术

3.2.1 应用背景

在数据库管理或优化的过程中,为了能够预先对需要改动的策略或配置进行评估,决策者或者管理者需要一套有先见之明的分析系统或分析模型.如果假设分析技术可以在某些给定场景下,对复杂系统的未来行为进行模拟和检查,进而可以满足决策者的需求,为决策结果的产生提供依据.在物理结构优化的数学模型中,如果假设分析技术是针对模型中物理配置价值分析的关键技术.

3.2.2 技术分析

在数据结构优化领域,如果假设分析是一种基于查询优化器进行评估的技术,假设如果采取不同的策略或

配置方案,调用查询优化器进行评估会产生何种结果,以便做出最佳的决策^[50].如果假设分析的输入一般是工作负荷和一组数据结构配置,输出基于这组数据结构配置的执行代价或成本.在分析过程中,这组数据结构配置并不是真正的生成,只是虚拟出来,而且没有任何数据,利用已知的统计信息对工作负荷的成本和代价进行评估.例如,如何为表创建一个最合适的索引?首先根据工作负荷分析出所有可能适合的索引配置,然后依次把索引配置作为输入调用如果假设分析,系统会模拟出一个这样配置的索引定义,伴随着相关的统计数据.当处理一个查询时,查询优化器将会把这个虚拟出来的索引当作真实存在的索引,并且将其包含在查询执行计划中进行审议和评估.在查询优化器的评估过程中,索引是否真正地存在是无关紧要的,查询优化器最关心的是索引键的定义和统计信息等.然后返回基于该索引配置的整个工作负荷的成本估计,找到综合成本估计最小的那个索引就可以作为该表的最优索引.

利用如果假设分析技术的物理结构优化的基本结构如图 2 所示.

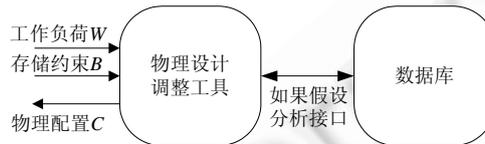


Fig.2 Architecture of what-if analysis

图 2 如果假设分析的基本结构

Chaudhuri 等人^[51]在 1997 年提出利用查询优化器来帮助自动选择索引,之后经过扩展被应用到物化视图、数据划分等其他物理结构的选择上,以索引选择为例,最初描述的方法如下:

- 1) 确定候选索引方案,这个过程主要执行 BEST-CONF(query specific best configuration)算法,针对工作负荷中的每个查询查找最优的索引,构成后面操作的候选索引集;
- 2) 索引枚举,运行贪婪算法 Greedy(m, k),其中, m 是初始化的种子集, k 是目标索引集的最大值,服从单索引结合修整技术(single-join atomic configuration pruning)枚举索引配置,并且选择一个索引集直至达到所需的索引数量或者总体成本不再减少为止;
- 3) 基于在步骤 2)中得到的单属性索引,使用 MC-LEAD 迭代技术选择一组可接受的多属性索引;
- 4) 由步骤 2)中选择的单列索引和步骤 3)中选出的多列索引组成可被接受的索引,再重复步骤 1)和步骤 2).

3.2.3 小结

在大部分商业数据库的物理结构推荐工具中都使用了如果假设分析技术,利用查询优化器的评价功能筛选和评估候选物理结构配置.虽然如果假设分析技术在很多优化推荐工具中获得应用,但其仍然存在严重的弊端.在利用如果假设分析来探测可行解空间的过程中,调优工具等待如果假设调用(访问查询优化器)的返回结果所消耗的时间,占优化过程总时间的 90% 以上^[37].为了解决这个问题, Bruno 等人在 2008 年提出参数可配的如果假设调用过程,但这种可配置查询优化技术过于复杂,应用和实现起来相对比较困难.

3.3 成本估计和数据抽样技术

3.3.1 应用背景

成本估计是物理结构优化工具的核心,也是决定优化质量好坏的最主要的因素.早期文献提出的优化方法都是基于简单的评价模型进行成本评估,也未涉及到统计数据和抽样技术,因此推荐优化的质量不是很好.1997 年, Chaudhuri 等人提出利用查询优化器来帮助自动选择索引,利用统计数据建立良好的评估模型,极大地提高了推荐优化的质量,后来被广泛应用在物理结构优化领域.统计数据是查询优化器进行成本估计的基础,由于当前数据库的数据量较大,为了解决统计数据产生过程中存在的质量和效率的矛盾,数据抽样技术被广泛地加以应用.在物理结构优化的数学模型中,成本估计和数据抽样技术是针对模型中某一工作负荷执行时进行代价评

估的关键技术。

3.3.2 技术分析

为了提高优化推荐的效率,大部分推荐系统中使用的都是基于统计数据和抽样技术的成本估计。随着数据量的日益增长,针对数据库中的全部数据进行分析已经变得非常困难,那么在物理结构优化推荐的过程中,推荐的时间成本也变得非常昂贵。为了降低推荐优化成本,统计数据的获取方法主要包括两种:采用数据抽样技术和提取原始数据库的统计信息。

在大数据集上进行统计操作是非常耗时的,但在大多数情况下,可以采用数据抽样技术来实现近似统计。这时,统计操作的速度明显提高,但会稍微降低统计的准确度。在统计精度要求不是很高的情况下,数据抽样方法是非常高效的。数据库领域应用的抽样技术^[52-54]总结起来主要包括如下4种:

- 1) 简单随机抽样。简单随机抽样要求在抽样过程中,从总体中任意随机抽取个体出来作为样本,任何个体被抽中的概率都相等,样本的每个单位完全独立,彼此间无一定的关联性和排斥性。简单随机抽样具有以下不足之处:
 - (a) 简单随机抽样要求被抽取的样本的总体个数是有限和可知的;
 - (b) 被水平分区的数据表很难对抽样操作进行并行处理。因此,大多数数据库管理系统并不直接支持简单随机抽样;
- 2) 伯努利抽样。伯努利抽样作为 SQL 标准的一部分,已被大多数主流数据库厂商所支持。伯努利抽样是在行级别进行的一种抽样技术,由于抽样粒度较小,所以最终都会获得一个有效的、不依赖数据特性(比如群集等)的随机抽样。但因为抽样操作于行级别,所以这种类型的抽样性能较差。因此,这种抽样技术一般都结合可用索引一起使用,以便提高抽样性能;
- 3) 系统抽样。系统抽样是首先将总体中各单位按一定顺序排列,根据样本容量要求确定抽选间隔;然后随机确定起点,每隔一定的间隔抽取一个单位的一种抽样方式。而在数据库领域,系统抽样是定义在 SQL 标准中的抽样,在实际使用过程中,它在存储页面级别进行抽样,可以显著提高抽样的性能。但是,如果数据具有明显的群集时,系统抽样效果会变得非常差;
- 4) 分层抽样。分层抽样是行级别伯努利抽样和页级别系统抽样的结合,从而实现对抽样速度和抽样精度合适的折中。其过程是:首先以概率 p 进行页级别系统抽样,然后再在被抽中的页内进行概率为 q 的行级别伯努利抽样。分层抽样尽量利用事先掌握的信息,并充分考虑到需要保持样本结构和总体结构的一致性,这对提高样本的代表性是很重要的,避免了在随机抽样过程中由于坏运气所形成的非常差的分布结果。分层抽样未定义在 SQL 标准中,但却被一些商业数据库所支持。

3.3.3 小结

以上总结了4种数据抽样技术的特点以及在数据库领域适合的应用场景,无论是哪种抽样技术,其所获得的统计结果都只是原始数据的近似统计结果。由于数据不可能完全均匀分布在磁盘上,需要统计的数据往往是有偏斜的,因此,抽样技术的方法优劣决定了统计结果与真实结果的偏差大小,进而影响了查询优化策略的生成或者物理结构优化配置的选择。

3.4 组合优化的搜索算法

3.4.1 应用背景

组合优化作为应用数学中最年轻而又至关重要的领域之一,整合了组合数学、线性规划以及算法理论的方法和技巧。由于它在解决从远程通信到超大规模集成电路、从产品运销到航班机组排班等领域内困难问题方面的成功,这一领域在过去的10年里取得了巨大的、超乎寻常的发展。在推荐优化的过程中,系统都会面临物理结构配置的巨大的组合空间的搜索问题,总结起来就是一个组合优化问题。前面提到过,使用枚举法是不现实的,那么就要寻找有效的搜索算法。在物理结构优化的数学模型中,搜索算法是针对模型中物理配置选择过程的关键技术。下面总结几种在组合优化问题上通用的搜索算法,这些算法也经常应用在物理结构的优化过程中。

3.4.2 遗传算法

遗传算法(genetic algorithm,简称 GA)是模拟达尔文生物进化论的自然选择和遗传学机理的生物进化过程的计算模型,是一种通过模拟自然进化过程搜索最优解的方法.虽然该算法随机地进行运算,但却不是一个简单的随机过程,它通过遗传信息来探测搜索空间中新的领域,以求获得更优的方案^[55-58].

遗传算法的基本运算过程如下(如图 3 所示):

- 1) 初始化:设置进化代数计数器 $t=0$,设置最大进化代数 T ,随机生成 M 个个体作为初始群体 $P(0)$;
- 2) 编码:这个过程主要是对所要解决的问题进行编码,每个个体的编码代表问题的一个解;
- 3) 个体评价:计算群体 $P(t)$ 中各个个体的适应度;
- 4) 选择运算:将选择算子作用于群体.选择的目的是把优化的个体直接遗传到下一代,或者通过配对交叉产生新的个体再遗传到下一代.选择操作是建立在群体中个体的适应度评估基础上的;
- 5) 交叉运算:将交叉算子作用于群体.所谓交叉是指把两个父代个体的部分结构加以替换重组而生成新个体的操作.遗传算法中起核心作用的就是交叉算子;
- 6) 变异运算:将变异算子作用于群体.即是对群体中的个体串的某些基因座上的基因值加以变动;
- 7) 群体 $P(t)$ 经过选择、交叉、变异运算之后得到下一代群体 $P(t+1)$;
- 8) 终止条件判断:若 $t=T$ 或者满足一定终止条件,则以进化过程中所得到的具有最大适应度个体作为最优解输出,终止计算;否则,返回步骤 3),继续计算.

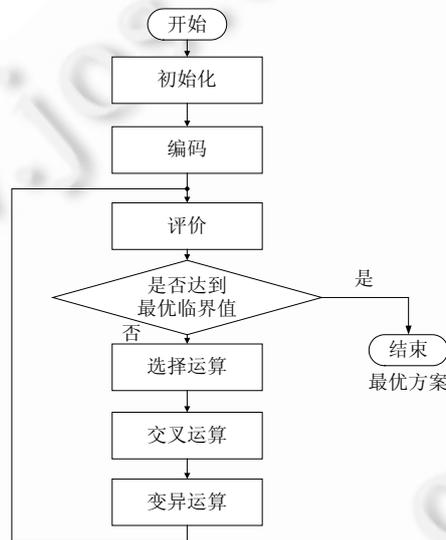


Fig.3 A flow diagram of genetic algorithm

图 3 遗传算法流程图

使用遗传算法解决物理结构优化问题的优点是:使用概率机制进行迭代,具有随机性;从群体开始搜索,覆盖面大,利于全局择优;具有可扩展性,容易与其他算法结合.而缺点是遗传算法的编程实现比较复杂,首先需要对问题进行编码,找到最优解之后还需要对问题进行解码;3 个算子的实现也有许多参数,如交叉率和变异率,并且这些参数的选择严重影响了解的品质,而目前这些参数的选择大部分依靠的是经验;遗传算法在适应度函数选择不当的情况下有可能收敛于局部最优,而不能达到全局最优.

3.4.3 模拟退火算法

模拟退火算法(simulated annealing,简称 SA)是一种通用启发式优化方法,用于在一个大的搜寻空间内找寻命题的最优解.这种算法的思想最早是由 Metropolis 等人提出来的,但把它应用于组合优化问题却是由

Kirkpatrick 等人^[59]于 1983 年提出来的.SA 的思想是基于物理中固体物质的退火过程与一般组合优化问题间的相似性.组合优化问题解空间中的每一个点都代表一个解,不同解有着不同的成本函数值,所谓优化,就是在解空间中找到成本函数值最大(最小)的解.

模拟退火算法新解的产生和接受可分为如下 4 个步骤(如图 4 所示):

- 1) 由一个产生函数从当前解产生一个位于解空间的新解.通常选择由当前新解经过简单变换即可产生新解的方法.注意到产生新解的变换方法决定了当前新解的邻域结构,因而对冷却进度表的选取有一定的影响;
- 2) 计算与新解所对应的目标函数差.因为目标函数差仅由变换部分产生,所以目标函数差的计算最好按增量计算;
- 3) 判断新解是否被接受.判断的依据是一个接受准则,最常用的接受准则是 Metropolis 准则:若 $\Delta E < 0$,则接受 S' 作为新的当前解 S ; 否则,以概率 $\exp(-\Delta E/T)$ 接受 S' 作为新的当前解 S ;
- 4) 当新解被确定接受时,用新解代替当前解,这只需将当前解中对应于产生新解时的变换部分予以实现,同时修正目标函数值即可.此时,当前解就实现了一次迭代,可在此基础上开始下一轮实验.而当新解被判定为舍弃时,则在原当前解的基础上继续下一轮实验.

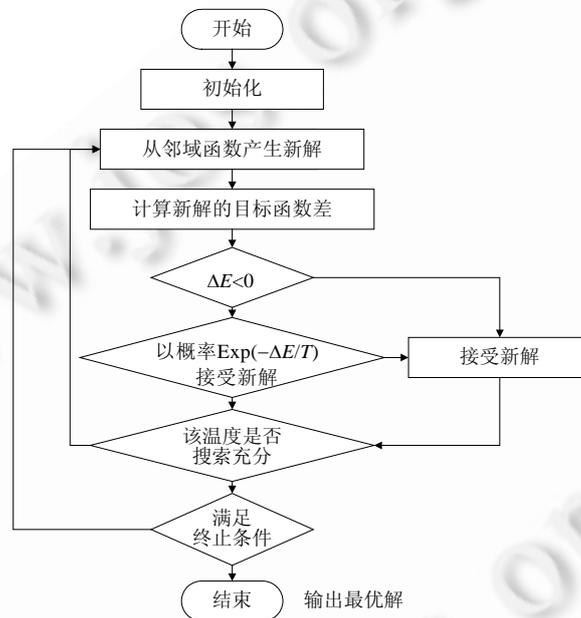


Fig.4 A flow diagram of simulated annealing

图 4 模拟退火算法流程图

模拟退火算法具有如下优点:(1) 模拟退火算法求得的解与初始解状态无关;(2) 模拟退火算法具有渐近收敛性,已在理论上被证明是一种以概率 1 收敛于全局最优解的全局优化算法;(3) 模拟退火算法具有并行性.模拟退火算法在组合优化问题上具有广泛的应用,而且该算法还可以应用到多目标优化环境^[60].但它还有一个主要的缺点就是较高的计算需求,因此,为了提高计算速度可以采用并行化^[61,62].

3.4.4 禁忌搜索算法

禁忌搜索(tabu search,简称 TS)是对局部邻域搜索算法的一种扩展,Glover 在 20 世纪 80 年代^[63,64]提出这个概念,进而形成一套完整算法.它从一个初始可行解出发,选择一系列的特定搜索方向(移动)作为试探,选择实现让特定的目标函数值变化最多的移动.为了避免陷入局部最优解,TS 搜索中采用了一种灵活的记忆技术,对已

经进行的优化过程进行记录和选择,指导下一步的搜索方向.禁忌搜索是一种全局逐步寻优、高效启发式优化技术和对人类智力过程的一种模拟.TS 算法通过引入一个灵活的存储结构和相应的禁忌准则,以避免迂回搜索,并通过藐视准则赦免一些被禁忌的优良状态,进而保证多样化的有效搜索,最终实现全局优化.

TS 算法的主要过程如下(如图 5 所示):

- 1) 以某一随机配置启动,求当前配置下的准则函数(criterion function)的值;
- 2) 按照当前候选集中的活动开始进行操作,如果这些活动中最优的不在禁忌列表中,或者在列表中但满足藐视准则,则将此活动作为下一个当前配置,并将其添加到禁忌列表中;否则,选取不在禁忌列表中的、具有最优效果的活动作为下一个当前配置.这个过程重复一定的次数;
- 3) 当步骤 2)结束时,就获取了当前最优的解决方案.

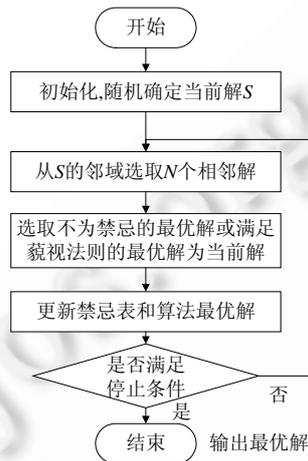


Fig.5 A flow diagram of tabu search

图 5 禁忌算法的流程图

与传统邻域搜索算法有很大区别的禁忌搜索的主要特点是:搜索过程中可以接受劣解,因此具有较强的适应能力;新解不是在当前解的邻域中随机产生,而或是优于当前最优解的解,或是非禁忌的最佳解,因此选取优良解的概率要远远大于其他算法.尽量避免迂回搜索,允许接受邻域中非禁忌的劣解和满足藐视准则的优良解,是 TS 算法实现高效搜索的关键.由于 TS 类似于模拟退火算法,同为串行搜索,因此,提高优化效率也是 TS 面临的重要课题.

3.4.5 小结

除了上面介绍的 3 种算法外,还有一些组合优化的搜索算法,比如神经网络^[65]、蚁群算法^[66,67]和爬山算法^[44]等,也经常用来解决组合优化问题.这些算法在解决组合优化问题上都有自己独特的优点:首先,这些算法的鲁棒性都比较强,对基本算法模型稍加改动就可以应用到其他问题上;其次是具有并发性,适合分布式解决计算量较大的问题.这些算法既相互区别又紧密联系,在一定应用环境下,把它们有机地结合起来,取长补短,可能会取得更加出色的效果.面对物理结构优化过程中庞大的搜索空间,如何选择搜索算法,是优化效率好坏的关键因素.

4 物理结构优化工具

4.1 Database Engine Tuning Advisor 工具

Data Engine Tuning Advisor(简称 DTA)是 Microsoft SQL Server 自带的数据库性能调优工具,支持一个整合的物理设计推荐,包括范围划分、索引和物化视图^[38].这个工具使用的是基于查询优化器的紧耦合优化方法:首

先,在一台机器上以黑盒子的方式模拟物理上数据库的各种优化配置.就像一个真正的数据库,但没有实际数据;然后,利用并行查询处理优化器的简要优化搜索空间(MEMO)结构,推断出部分配置的成本下限,并且利用这一阈值更快地完成物理配置空间的搜索过程(如图 6 所示).

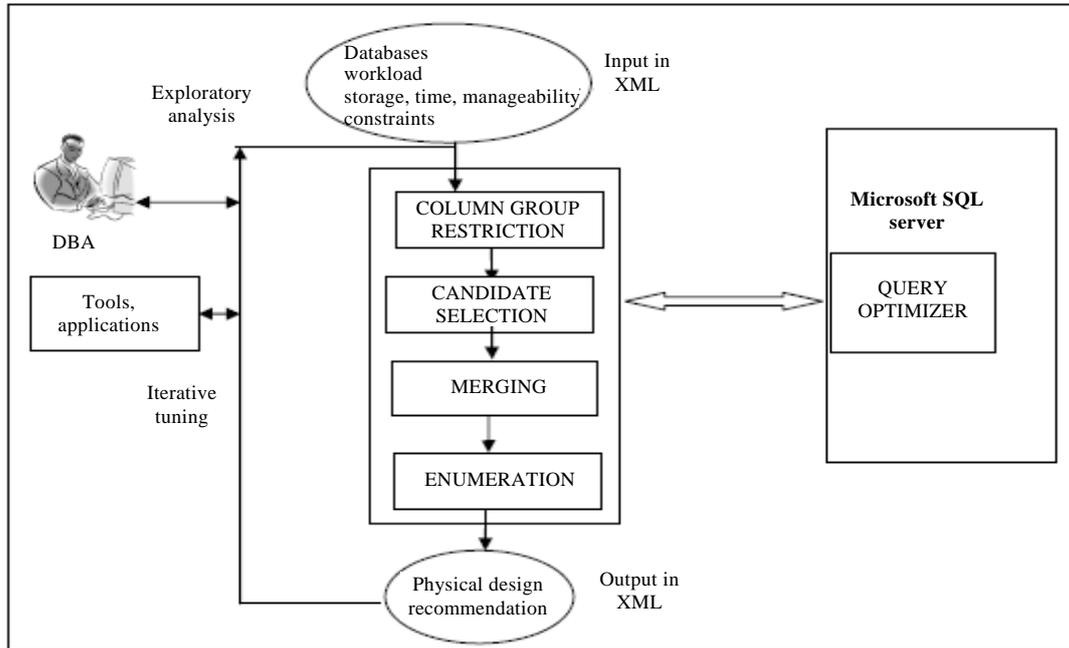


Fig.6 Architecture of Database Engine Tuning Advisor^[36]

图 6 Database Engine Tuning Advisor 体系结构^[36]

DTA 的主要功能包括^[38]:

- 通过查询优化器分析实际工作负荷下的查询,推荐最好的索引组合;
- 基于工作负荷为数据库推荐对齐(aligned)和非对齐(non-aligned)分区;
- 基于工作负荷为数据库推荐带索引的视图;
- 分析工作负荷,对某种变更的效果进行评估,包括索引用法、表间查询分布、查询效果;
- 对一些有问题的查询进行调优推荐;
- 可以通过高级设置,比如磁盘空间大小限制,来影响其做出的推荐方案;
- 提供基于某个工作负荷的推荐方案具体实现的效果报告;
- 可以提供不同假设配置下的替代方案进行分析.

4.2 Design Advisor工具

IBM 的 DB2 Design Advisor 工具可以根据用户的工作负荷,自动地推荐数据库的多个物理特征设计.普通工具和算法一般支持索引和物化视图两个物理特征的优化推荐,而 Design Advisor 工具作为一个商业数据库物理结构优化推荐工具,成功地支持至少 4 种物理特征^[39].Design Advisor 所提供的建议能够与数据库调优专家的建议相媲美.对于非专家来说,该工具的好处是可以获得更好的设计.对于专家来说,Design Advisor 可以节省他们宝贵的时间,因为 Design Advisor 可以提供一个初始的设计,然后由专家进一步加以改进.Design Advisor 还可以提供对专家的设计的独立确认^[40].

DB2 Design Advisor 工具的主要特点是(如图 7 所示):

- 该工具支持多种提供工作负荷的方法.比如在 GUI 中从一个文件装载工作负载,或者从 Event Monitor

获得工作负载等等;

- 采用了工作负载压缩技术,所以运行该工具时所需的资源更少;
- 通过查询优化器推荐候选物理配置方案,有效地降低了推荐过程的组合规模;
- 自动地推荐数据库的多个物理特征的设计,包括索引、物化视图、非共享数据划分和数据表的多维群集,而且支持其他物理特征的扩展;
- 可以生成关于推荐的一个报告,并将其保存到一个文件中.

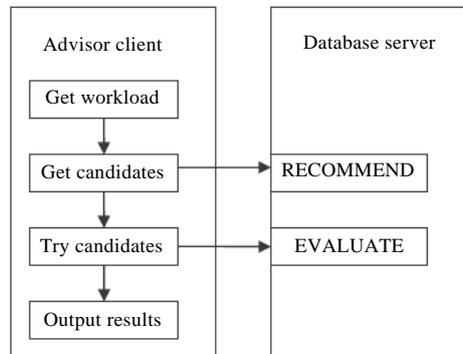


Fig.7 Architecture of Design Advisor^[39]

图 7 Design Advisor 的结构图^[39]

4.3 SQL Access Advisor工具

SQL Access Advisor 工具(SQL 访问顾问)是 Oracle 公司推出的数据库性能优化工具,基于对 SQL 语句的各种谓词和从句的分析,对索引、物化视图的创建和数据表的物理划分等进行推荐^[68,69].当为复杂的、数据密集查询进行最优性能调整时,物化视图和索引是必要的.SQL 访问顾问通过为一个给定的工作量推荐的一个关于物化视图、索引和表划分的适当集合,来帮助使用者达到优化目标.当优化 SQL 时,理解和使用这些结构是必要的,因为它们可以在数据读取方面获得重大的性能提升.但是,有得必有失,创建和维护这些对象需要花费时间,空间的要求也是很重要的.SQL 访问顾问会根据实际的应用需求给出最合理的推荐(如图 8 所示).

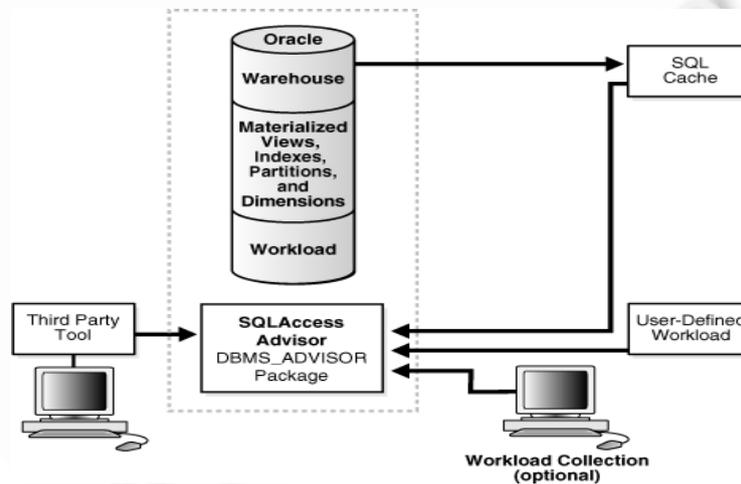


Fig.8 Architecture of SQL Access Advisor^[70]

图 8 SQL Access Advisor 体系结构^[70]

SQL 访问顾问在优化物理结构的过程中通常需要进行许多推荐,每个推荐包括一个或多个独立的操作或动作.一般来讲,每个推荐都是对一个或一些查询语句提供相对不错的优化收益.在一个推荐中的所有动作必须在一起执行才能达到最好的收益效果.这些推荐中的动作可以彼此共享.SQL 访问顾问有两种操作模式——problem-solving 和 evaluation^[70].在模式 problem-solving 中,SQL 访问顾问主要试图通过创建新的索引结构、划分、物化视图和物化视图日志等来提高数据的访问效率;而在 evaluation 模式下,SQL 访问顾问主要评估被应用于工作负荷的那些物理结构.若想清楚地看到每个工作负荷所使用的索引、物化视图的具体情况,SQL 访问顾问工作在 evaluation 模式下是非常有用的,但 SQL 访问顾问不会评估现有基表的划分对性能的影响.

SQL 访问顾问具有如下功能^[70]:

- 基于收集到的或用户输入的理想的的工作量信息推荐物化视图和索引;
- 推荐基表划分、物化视图和索引;
- 标记、更新和删除物理结构推荐;
- 收集和管理工作负荷;
- 执行一个使用单个 SQL 语句的快速优化;
- 展示如何使物化视图快速刷新;
- 展示如何更改物化视图,使一般查询语句的重写成为可能.

4.4 DB Designer工具

HP 的 Vertica 是基于列存储的面向数据分析的数据仓库系统,采用高效的数据压缩机制,同时具有基于 K-Safe 的高可用性.K-Safe 机制简单地讲就是保证在 Vertica 集群内,每一个数据库的每一个表的每一列被存储在至少 $K+1$ 台机器上.这样,当 K 台机器不可用时,仍然有一个完整的数据库备份可用.DB Designer 是 Vertica 开发的物理数据库自动设计工具.通过简化数据库的设计,让设计和管理人员更多地关注业务需求和逻辑模型设计.DB Designer 根据数据样本和查询样本将逻辑 schema 映射为多个物理 schema,即将一个行映射拆分为多个行映射,同时保证每个查询的 from 部分都只来自一个映射(如图 9 所示).

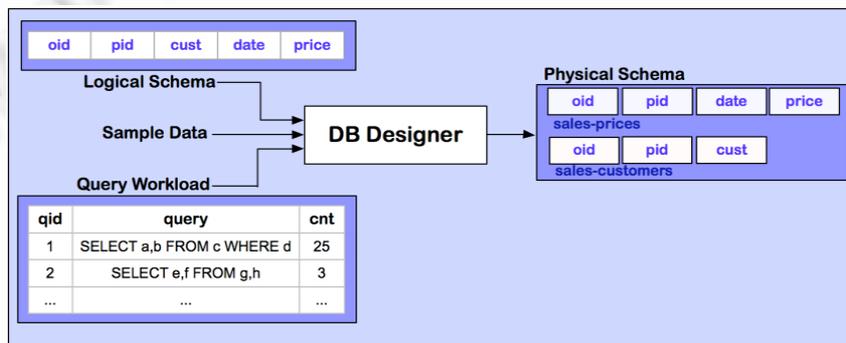


Fig.9 Flow diagram of DB Designer tool^[71]

图 9 DB Designer 工具的流程图^[71]

DB Designer 工具^[71]具有如下特点:

- 在使用 Vertica 时,用户仅需要设计一个逻辑的数据库模式,Vertica 的 DB Designer 工具自动产生一个合适的物理数据库设计,这个设计基于逻辑模式、样列数据集和查询及数据库空间来估算;
- DB Designer 通过确保所有的查询数据至少出现在一个映射内,从而保证任何一个有效的查询都能有效命中;
- DB Designer 根据每个列的数据类型和数据值自动选择优化的压缩算法,并决定为了优化某个查询而如何构建映射,决定包含哪些属性列以及如何排序;

- DB Designer 选择每个映射的合适的段关键字,保证这个映射满足具体的 *K-Safety* 要求.

4.5 小结

上面介绍的几种流行的设计工具都是通过查询优化器从候选配置集中选择那些使用最少的创建、维护以及存储成本而且提供最大收益给整个工作负荷的最优的物理结构配置.然而,各个优化推荐工具因为所属数据库的不同而具有不同的功能组合.其中,有些因数据库产品本身不支持的功能,优化推荐工具也不支持该功能.表 1 是几种流行的设计工具的功能列表.

Table 1 Feature list of design products

表 1 设计工具的功能列表

	SQL server 数据库引擎调优顾问	DB2 设计顾问	Oracle SQL 访问顾问	Vertica 数据库设计工具
索引	✓	✓	✓	
物化视图	✓	✓	✓	✓
组合属性选择	✓	✓	✓	✓
无共享分区		✓	✓	
范围分区	✓			
多维聚集		✓		
工作负荷压缩	✓	✓		
如果假设分析	✓	✓	✓	✓
自动化抽样		✓		✓

5 总结与展望

物理结构优化技术一直都是数据库优化领域研究的一个热点.从 20 世纪 70 年代开始,学者们就开始研究物理结构优化技术,目前已经有了很多物理结构优化推荐方法,但大部分还都只是处于研究阶段,只有少数方法被一些商业数据库采用并推向市场,形成了数据库物理结构优化推荐工具.本文从 3 个方向总结了物理结构优化技术:首先,介绍了物理结构优化涉及的几种基本元素,包括索引、物化视图、无共享划分和多维聚集,分别描述了每种基本元素的特点和相关的优化方法;然后,总结了物理结构优化过程中涉及到的几种关键技术,包括可以提高优化效率的工作负荷压缩技术和启发式组合算法,以及对物理配置进行评价的如果假设分析技术和统计数据抽样技术;最后,介绍了几种商业数据库中使用的数据库物理结构优化推荐工具的特点,并针对它们的功能进行了比较.

在物理结构优化领域,虽然已经研究和开发出很多实用的算法和工具,但仍然有很多方向值得我们进一步研究和总结.归纳起来有以下几点:

- (1) 解决组合爆炸问题.物理结构优化问题实际上就是一个组合优化问题,穷举法无法又快又好地完成组合空间的搜索,因此需要一种较好的启发式组合优化算法.已有的算法要么推荐效率低下,要么推荐质量不高,因此,高效率、高质量的搜索算法还是未来该领域的主要研究方向之一;
- (2) 代价评估方法的研究.优化问题面对巨大的搜索空间,要为每一个配置的好坏进行一个定性的评价,评价标准和评价方法的选择直接关系到优化推荐工具的可用性和有效性,是优化过程的一个重要因素.基于查询优化器的评估方法是当前的主流方法,但其如果假设调用的代价较大,需要进一步加以改善;
- (3) 相关性分析.优化过程中涉及到很多不同的对象,这些对象间都具有一定的相关性,优化推荐过程需要考虑对象间的相关性,否则,在实际应用中无法获得最优性能.过去实现的优化系统大部分都未考虑这些对象间的相关性,这是研究的一个难点;
- (4) 优化推荐过程的自适应性.目前已有的算法和工具通常都需要在数据库管理员(DBA)的干预下来完成物理结构的优化过程,优化推荐的自动化是物理结构优化领域未来的一个发展趋势.

致谢 在此,我们向对本文的工作给予支持和建议的老师和同学表示感谢.

References:

- [1] Manyika J, Chui M, Brown B, Bughin J, Dobbs R, Roxburgh C, Byers AU. Big data: The next frontier for innovation, competition, and productivity. Technical Report, McKinsey Global Institute, 2011. 1–15. http://www.mckinsey.com/Insights/MGI/Research/Technology_and_Innovation/Big_data_The_next_frontier_for_innovation
- [2] Bruno N, Chaudhuri S. Automatic physical database tuning: A relaxation-based approach. In: Özcan F, ed. Proc. of the Special Interest Group on Management of Data (SIGMOD 2005). Baltimore: ACM Press, 2005. 227–238. [doi: 10.1145/1066157.1066184]
- [3] Gebaly KE, Aboulnaga A. Robustness in automatic physical database design. In: Kemper A, Valduriez P, Mouaddib N, Teubner J, Bouzeghoub M, Markl V, Amsaleg L, Manolescu I, eds. Proc. of the Extending Database Technology (EDBT 2008). Nantes: ACM Press, 2008. 145–156. [doi: 10.1145/1353343.1353365]
- [4] Borovica R, Alagiannis I, Ailamaki A. Automated physical designers: What you see is (not) what you get. In: Lo E, Waas F, eds. Proc. of the 5th Int'l Workshop on Testing Database Systems (DBTest 2012). Scottsdale: ACM Press, 2012. 1–6. [doi: 10.1145/2304510.2304522]
- [5] Zilio DC, Lightstone S, Lohman GM. Trends in automating physical database design. In: Proc. of the IEEE Int'l Conf. 2003. India: IEEE Press, 2003. 441–445. [doi: 10.1109/INDIN.2003.1300376]
- [6] Rozen S, Shasha D. A framework for automating physical database design. In: Lohman GM, Sernadas A, Camps R, eds. Proc. of the Very Large Data Bases (VLDB'91). Barcelona: Morgan Kaufmann Publishers, 1991. 401–411.
- [7] Zilio DC. Physical database design decision algorithms and concurrent reorganization for parallel database systems [Ph.D. Thesis]. Toronto: University of Toronto, 1998.
- [8] Rao J, Zhang C, Lohman G, Megiddo N. Automating physical database design in a parallel database. In: Franklin MJ, Moon B, Ailamaki A, eds. Proc. of the Special Interest Group on Management of Data (SIGMOD 2002). Madison: ACM Press, 2002. 558–569. [doi: 10.1145/564691.564757]
- [9] Aslett M. How Will the Database Incumbents Respond to NoSQL and NewSQL? San Francisco, The 451 Group, 2011. 1–5.
- [10] Stonebraker M, Cattell R. 10 rules for scalable performance in 'simple operation' datastores. Communications of the ACM, 2011, 54(6):72–80. [doi: 10.1145/1953122.1953144]
- [11] Caprara A, Fischetti M, Maio D. Exact and approximate algorithms for the index selection problem in physical database design. IEEE Trans. on Knowledge and Data Engineering, 1995,7(6):955–967. [doi: 10.1109/69.476501]
- [12] Caprara A, Salazar Gonzalez SJJ. A branch-and-cut algorithm for a generalization of the uncapacitated facility location problem. TOP, 1996,4(1):135–163. [doi: 10.1007/BF02568608]
- [13] Kratica J, Ljubic I, Tosic D. A genetic algorithm for the index selection problem. In: Raidl GR, Meyer JA, Middendorf M, Cagnoni S, Cardalda JJR, Corne D, Gottlieb J, Guillot A, Hart E, Johnson CG, Marchiori E, eds. Proc. of the Applications of Evolutionary Computing, EvoWorkshops 2003. Berlin, Heidelberg: Springer-Verlag, 2003. 280–290. [doi: 10.1007/3-540-36605-9_26]
- [14] Dash D, Polyzotis N, Ailamaki A. CoPhy: A scalable, portable, and interactive index advisor for large workloads. Proc. of the VLDB Endowment (PVLDB), 2011,4(6):362–372.
- [15] Kimura H, Huo G, Rasin A, Madden S, Zdonik SB. CORADD: Correlation aware database designer for materialized views and indexes. Proc. of the VLDB Endowment (PVLDB), 2010,3(1):1103–1113.
- [16] Jimenez I, Sanchez H, Tran QT, Polyzotis N. Kaizen: A semi-automatic index advisor. In: Candan KS, Chen Y, Snodgrass RT, Gravano L, Fuxman A, eds. Proc. of the Special Interest Group on Management of Data (SIGMOD 2012). Scottsdale: ACM Press, 2012. 685–688. [doi: 10.1145/2213836.2213932]
- [17] Schnaitter K, Polyzotis N. Semi-Automatic index tuning: Keeping DBAs in the loop. Proc. of the VLDB Endowment (PVLDB), 2012,5(5):478–489.
- [18] Zhou LJ, Geng HJ, Xu MS. An improved algorithm for materialized view selection. Journal of Computers, 2011,6(1):130–138. [doi: 10.4304/jcp.6.1.130-138]
- [19] Harinarayan V, Rajaraman A, Ullman JD. Implementing data cubes efficiently. In: Jagadish HV, Mumick IS, eds. Proc. of the Special Interest Group on Management of Data (SIGMOD'96). Montreal: ACM Press, 1996. 205–216. [doi: 10.1145/235968.233333]
- [20] Shukla A, Deshpande P, Naughton JF. Materialized view selection for multidimensional datasets. In: Gupta A, Shmueli O, Widom J, eds. Proc. of the Very Large Data Bases (VLDB'98). New York: Morgan Kaufmann Publishers, 1998. 488–499.
- [21] Kumar TVV, Ghoshal A. A reduced lattice greedy algorithm for selecting materialized views. In: Prasad SK, Routray S, Khurana R, Sahni S, eds. Proc. of the Information Systems, Technology and Management (ICISTM 2009). Berlin, Heidelberg: Springer-Verlag, 2009. 6–18. [doi: 10.1007/978-3-642-00405-6_5]

- [22] Kumar TVV, Haider M. A query answering greedy algorithm for selecting materialized views. In: Pan JS, Chen SM, Nguyen NT, eds. Proc. of the Int'l Conf. on Computational Collective Intelligence (ICCCI 2010). Berlin, Heidelberg: Springer-Verlag, 2010. 153–162. [doi: 10.1007/978-3-642-16732-4_17]
- [23] Kumar TVV, Kumar S. Materialized view selection using genetic algorithm. In: Parashar M, Kaushik D, Rana OF, Samtaney R, Yang YY, Zomaya AY, eds. Proc. of the Int'l Conf. on Contemporary Computing (IC3 2012). Berlin, Heidelberg: Springer-Verlag, 2012. 225–237. [doi: 10.1007/978-3-642-32129-0_26]
- [24] Derakhshan R, Stantic B, Korn O, Dehne FKHA. Parallel simulated annealing for materialized view selection in data warehousing environments. In: Bourgeois AG, Zheng SQ, eds. Proc. of the Int'l Conf. on Algorithms and Architectures for Parallel Processing (ICA3PP 2008). Berlin, Heidelberg: Springer-Verlag, 2008. 121–132. [doi: 10.1007/978-3-540-69501-1_14]
- [25] Tan HX, Zhou LX. Dynamic selection of materialized views of multi-dimensional data. Ruanjian Xuebao/Journal of Software, 2002,13(6):1090–1096 (in Chinese with English abstract). <http://www.jos.org.cn/1000-9825/20020608.htm>
- [26] Zhang DZ, Huang ZY, Xue YS. NDSMMV—A new dynamic selection strategy of materialized views for multi-dimensional data. Journal of Computer Research and Development, 2008,45(5):901–908 (in Chinese with English abstract).
- [27] Shah B, Ramachandran V, Ramachandran K. A hybrid approach for data warehouse view selection. Int'l Journal of Data Warehousing and Mining, 2006,2(2):1–37. [doi: 10.4018/jdwm.2006040101]
- [28] Drias H. Generating materialized views using ant based approaches and information retrieval technologies. In: Proc. of the Computational Intelligence and Data Mining (CIDM 2011). Paris: IEEE Press, 2011. 276–282. [doi: 10.1109/CIDM.2011.5949302]
- [29] Yang J, Karlapalem K, Li Q. Algorithms for materialized view design in data warehousing environment. In: Jarke M, Carey MJ, Dittrich KR, Lochovsky FH, Loucopoulos P, Jeusfeld MA, eds. Proc. of the Very Large Data Bases (VLDB'97). Athens: Morgan Kaufmann Publishers, 1997. 136–145.
- [30] Gupta H. Selection of views to materialize in a data warehouse. In: Afrati FN, Kolaitis PG, eds. Proc. of the Int'l Conf. on Database Theory (ICDT'97). Berlin, Heidelberg: Springer-Verlag, 1997. 98–112. [doi: 10.1007/3-540-62222-5_39]
- [31] Baru CK, Fecteau G, Goyal A, Hsiao H, Jhingran A, Padmanabhan S, Wilson WG. An overview of DB2 parallel edition. In: Carey MJ, Schneider DA, eds. Proc. of the Special Interest Group on Management of Data (SIGMOD'95). San Jose: ACM Press, 1995. 460–462. [doi: 10.1145/223784.223876]
- [32] Mehta M, De Witt DJ. Data placement in shared-nothing parallel database systems. The Very Large Data Bases (VLDB) Journal, 1997,6(1):53–72. [doi: 10.1007/s007780050033]
- [33] Agrawal R, Chaudhuri S, Das A, Narasayya VR. Automating layout of relational databases. In: Dayal U, Ramamritham K, Vijayarman TM, eds. Proc. of the Int'l Conf. on Data Engineering (ICDE 2003). Bangalore: IEEE Press, 2003. 607–618. [doi: 10.1109/ICDE.2003.1260825]
- [34] Agrawal S, Narasayya VR, Yang B. Integrating vertical and horizontal partitioning into automated physical database design. In: Weikum G, König AC, DeBloch S, eds. Proc. of the Special Interest Group on Management of Data (SIGMOD 2004). Paris: ACM Press, 2004. 359–370. [doi: 10.1145/1007568.1007609]
- [35] Nehme R, Bruno N. Automated partitioning design in parallel database systems. In: Sellis TK, Miller RJ, Kementsietsidis A, Velegrakis Y, eds. Proc. of the Special Interest Group on Management of Data (SIGMOD 2011). Athens: ACM Press, 2011. 1137–1148. [doi: 10.1145/1989323.1989444]
- [36] Agrawal S, Chaudhuri S, Kollar L, Marathe A, Narasayya V, Syamala M. Database tuning advisor for microsoft SQL server 2005. In: Nascimento MA, Özsu MT, Donald Kossman D, Miller RJ, Blakeley JA, Schiefer KB, eds. Proc. of the Very Large Data Bases (VLDB 2004). Toronto: Morgan Kaufmann Publishers, 2004. 1110–1121.
- [37] Bruno N, Nehme IV. Configuration-Parametric query optimization for physical design tuning. In: Wang JT, ed. Proc. of the Special Interest Group on Management of Data (SIGMOD 2008). Vancouver: ACM Press, 2008. 941–952. [doi: 10.1145/1376616.1376710]
- [38] Database engine tuning advisor white paper. [http://technet.microsoft.com/en-us/library/ms173494\(v=SQL.105\).aspx](http://technet.microsoft.com/en-us/library/ms173494(v=SQL.105).aspx)
- [39] Zilio DC, Rao J, Lightstone S, Lohman GM, Storm AJ, Garcia-Arellano C, Fadden S. DB2 design advisor: Integrated automatic physical database design. In: Nascimento MA, Özsu MT, Kossman D, Miller RJ, Blakeley JA, Schiefer KB, eds. Proc. of the Very Large Data Bases (VLDB 2004). Toronto: Morgan Kaufmann Publishers, 2004. 1087–1097.
- [40] IBM design advisor. <http://www.ibm.com/developerworks/data/library/techarticle/dm-0508mcinerney/index.html>
- [41] Curino C, Zhang Y, Jones EPC, Madden S. Schism: A workload-driven approach to database replication and partitioning. Proc. of the VLDB Endowment (PVLDB), 2010,3(2):48–57.

- [42] Du J, Alhajj R, Barker K. Genetic algorithms based approach to database vertical partition. *Journal of Intelligent Information Systems*, 2006,26(2):167–183. [doi: 10.1007/s10844-006-0242-2]
- [43] Song SK, Gorla N. A genetic algorithm for vertical fragmentation and access path selection. *The Computer Journal*, 2000,34(1): 81–93.
- [44] Hammer M, Niamir B. A heuristic approach to attribute partition. In: Bernstein PA, ed. *Proc. of the Special Interest Group on Management of Data (SIGMOD'79)*. Boston: ACM Press, 1979. 93–100. [doi: 10.1145/582095.582110]
- [45] Pavlo A, Curino C, Zdonik S. Skew-Aware automatic database partitioning in shared-nothing, parallel OLTP systems. In: Candan KS, Chen Y, Snodgrass RT, Gravano L, Fuxman A, eds. *Proc. of the Special Interest Group on Management of Data (SIGMOD 2012)*. Scottsdale: ACM Press, 2012. 61–72. [doi: 10.1145/2213836.2213844]
- [46] Lightstone SS, Bhattacharjee B. Automated design of multidimensional clustering tables for relational databases. In: Nascimento MA, Özsu MT, Kossman D, Miller RJ, Blakeley JA, Schiefer KB, eds. *Proc. of the Very Large Data Bases (VLDB)*. Toronto: Morgan Kaufmann Publishers, 2004. 1170–1181.
- [47] Bhattacharjee B, Padmanabhan S, Malkemus T, Lai T, Cranston L, Huras M. Efficient query processing for multi-dimensionally clustered tables in DB2. In: Freytag JC, Lockemann PC, Abiteboul S, Carey MJ, Selinger PG, Heuer A, eds. *Proc. of the Very Large Data Bases (VLDB 2003)*. Berlin: Morgan Kaufmann Publishers, 2003. 963–974.
- [48] Padmanabhan S, Bhattacharjee B, Malkemus T, Cranston L, Huras M. Multi-Dimensional clustering: A new data layout scheme in DB2. In: Halevy AY, Ives ZG, Doan A eds. *Proc. of the Special Interest Group on Management of Data (SIGMOD 2003)*. San Diego: ACM Press, 2003. 637–641. [doi: 10.1145/872757.872835]
- [49] Chaudhuri S, Gupta AK, Narasayya V. Compressing SQL workloads. In: Franklin MJ, Moon B, Ailamaki A, eds. *Proc. of the Special Interest Group on Management of Data (SIGMOD 2002)*. Madison: ACM Press, 2002. 488–499. [doi: 10.1145/564691.564747]
- [50] Chaudhuri S, Narasayya V. AutoAdmin “What-If ” index analysis utility. In: Haas LM, Tiwary A, eds. *Proc. of the Special Interest Group on Management of Data (SIGMOD'98)*. Seattle: ACM Press, 1998. 367–378 [doi: 10.1145/276305.276337]
- [51] Chaudhuri S, Narasayya V. An efficient cost-driven Index selection tool for microsoft SQL server. In: Jarke M, Carey MJ, Dittrich KR, Lochovsky FH, Loucopoulos P, Jeusfeld MA, eds. *Proc. of the Very Large Data Bases (VLDB'97)*. Athens: Morgan Kaufmann Publishers, 1997. 146–155.
- [52] Haas PJ, Swami AN. Sequential sampling procedures for query size estimation. In: Stonebraker M, ed. *Proc. of the Special Interest Group on Management of Data (SIGMOD'92)*. San Diego: ACM Press, 1992. 341–350. [doi: 10.1145/141484.130335]
- [53] Haas PJ, König C. A bi-level Bernoulli scheme for database sampling. In: Weikum G, König AC, Deßloch S, eds. *Proc. of the Special Interest Group on Management of Data (SIGMOD 2004)*. Paris: ACM Press, 2004. 275–286. [doi: 10.1145/1007568.1007601]
- [54] Haas PJ, Kandil M, Lerner A, Markl V, Popivanov I, Raman V, Zilio DC. Automated statistics collection in action. In: Özcan F, ed. *Proc. of the Special Interest Group on Management of Data (SIGMOD 2005)*. Baltimore: ACM Press, 2005. 933–935. [doi: 10.1145/1066157.1066293]
- [55] Zhou M, Sun SD. *Genetic Algorithms—Theory and Applications*. Beijing: National Defense Industry Press, 1999. 1–202 (in Chinese).
- [56] Goldberg DE. *Genetic Algorithms in Search Optimization and Machine Learning*. Boston: Addison-Wesley Longman Publishing Co. Inc., 1989. 1–372.
- [57] Fotouhi F, Galarce CE. Genetic algorithms and the search for optimal database index selection. In: Sherwani NA, Doncker ED, Kapenga JA, eds. *Proc. of the Great Lakes Computer Science Conf.* 1989. New York: Springer-Verlag, 1991. 249–255. [doi: 10.1007/BFb0038500]
- [58] Hung WL, Xie Y, Vijaykrishnan N, Kandemir MT, Irwin MJ. Total power optimization for combinational logic using genetic algorithms. *The Journal of Signal Processing Systems*, 2010,58(2):145–160. [doi: 10.1007/s11265-009-0338-3]
- [59] Kirkpatrick S, Gelatt Jr. CD, Vecchi MP. Optimization by simulated annealing. *Science*, 1983,220(4598):671–680. [doi: 10.1126/science.220.4598.671]
- [60] Suman B. Study of simulated annealing based algorithms for multiobjective optimization of a constrained problem. *Computers & Chemical Engineering*, 2004,28(9):1849–1871. [doi: 10.1016/j.compchemeng.2004.02.037]
- [61] Knopman J, Aude JS. Parallel simulated annealing: An adaptive approach. In: *Proc. of the Int'l Parallel Processing Symp. (IPPS'97)*. Geneva: IEEE Press, 1997. 522–526. [doi: 10.1109/IPPS.1997.580950]

- [62] Yeh JY, Fu JC. Parallel adaptive simulated annealing for computer-aided measurement in functional MRI analysis. *Expert Systems with Applications*, 2007,33(3):706–715. [doi: 10.1016/j.eswa.2006.06.018]
- [63] Glover, F. Tabu search—Part I. *ORSA Journal on Computing*, 1989,1(3):190–206. [doi: 10.1287/ijoc.1.3.190]
- [64] Glover, F. Tabu search—Part II. *ORSA Journal on Computing*, 1990,2(1):4–32. [doi: 10.1287/ijoc.2.1.4]
- [65] Han JW, Kamber M. *Data Mining: Concepts and Techniques*. 2nd ed., San Francisco: Morgan Kaufmann Publishers, 2006. 1–743.
- [66] Dorigo M, Gambardella LM. Ant colonies for the traveling salesman problem. *BioSystems*, 1997,43(2):73–81. [doi: 10.1016/S0303-2647(97)01708-5]
- [67] Gambardella LM, Taillard E, Dorigo M. Ant colonies for the quadratic assignment problem. *Journal of the Operational Research Society*, 1999,50(2):167–176.
- [68] Dageville B, Das D, Dias K, Yagoub K, Zait M, Ziauddin M. Automatic SQL tuning in oracle 10g. In: Nascimento MA, Özsu MT, Donald Kossmann D, Miller RJ, Blakeley JA, Schiefer KB, eds. *Proc. of the Very Large Data Bases (VLDB 2004)*. Toronto: Morgan Kaufmann Publishers, 2004. 1098–1109.
- [69] Dias K, Ramacher M, Shaft U, Venkataramani V, Wood G. Automatic performance diagnosis and tuning in Oracle. In: Ailamaki A, Bernstein P, eds. *Proc. of the Conf. on Innovative Data Systems Research (CIDR 2005)*. Asilomar, 2005. 84–94.
- [70] Chan I, Ashdown L. *Oracle Database Performance Tuning Guide*, 11g Release 2 (11.2). E16638-07, 2012. 451–486.
- [71] The Vertica Analytic Database Technical Overview White Paper. A DBMS architecture optimized for next-generation data warehousing. Vertica Systems Inc., 2010. 1–13.

附中文参考文献:

- [25] 谭红星,周龙骧.多维数据实视图的动态选择. *软件学报*,2002,13(6):1090–1096. <http://www.jos.org.cn/1000-9825/20020608.htm>
- [26] 张东站,黄宗毅,薛永生.NDSMMV——一种多维数据集物化视图动态选择新策略. *计算机研究与发展*,2008,45(5):901–908.
- [55] 周明,孙树东. *遗传算法原理及应用*.北京:国防工业出版社,1999.1–202.



崔跃生(1982—),男,吉林舒兰人,硕士,主要研究领域为数据库技术.
E-mail: yuesheng797@163.com



冯建华(1967—),男,博士,教授,博士生导师,CCF高级会员,主要研究领域为数据库管理系统,数据安全与隐私保护,信息检索.
E-mail: fengjh@tsinghua.edu.cn



张勇(1973—),男,博士,副研究员,CCF高级会员,主要研究领域为数据库技术.
E-mail: zhangyong05@tsinghua.edu.cn



邢春晓(1967—),男,博士,教授,博士生导师,CCF高级会员,主要研究领域为数字图书馆,数据库技术.
E-mail: xingcx@tsinghua.edu.cn



曾春(1976—),男,博士,主要研究领域为个性化服务,数据库,内容管理技术.
E-mail: zengchun@tsinghua.org.cn