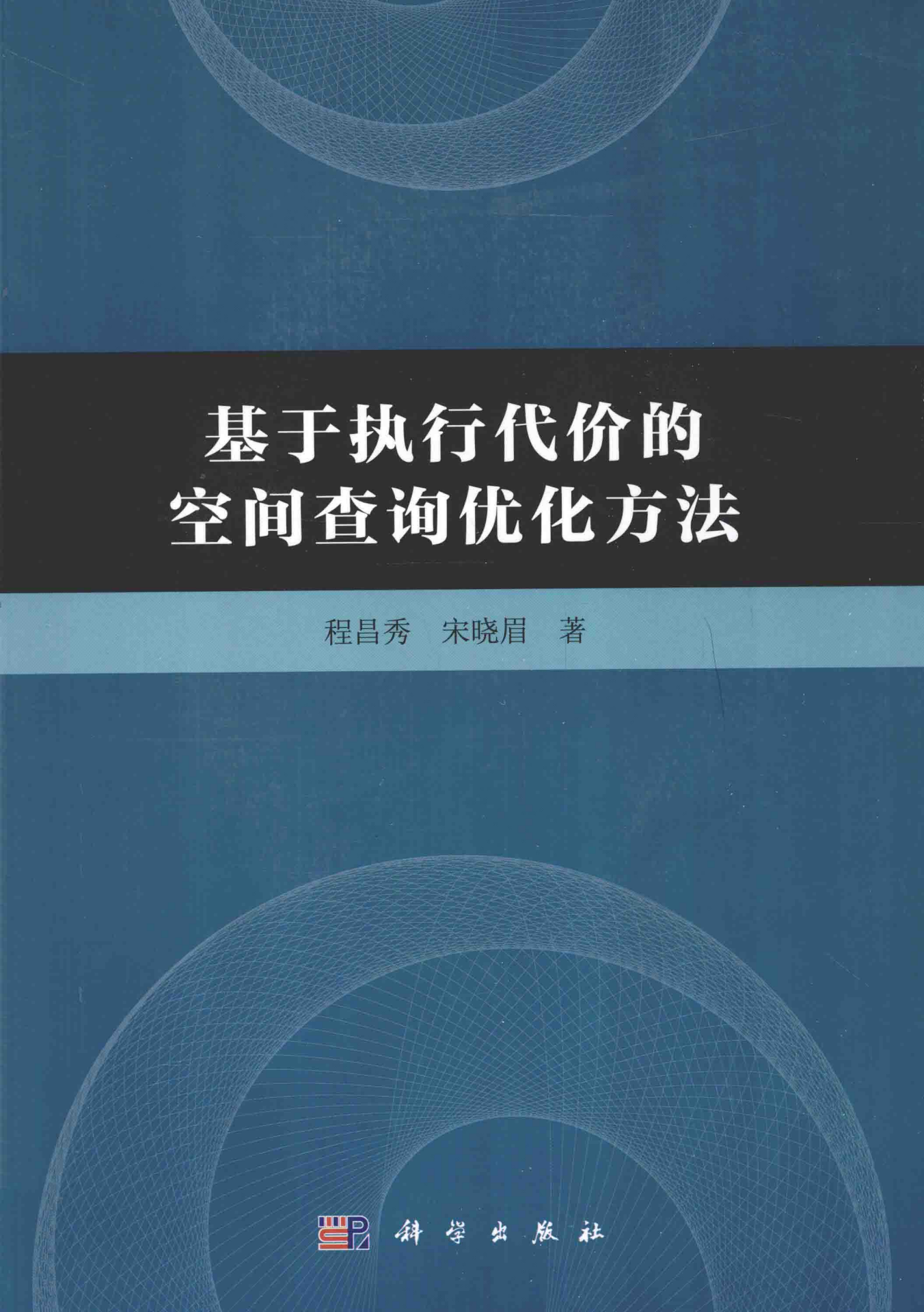




基于执行代价的 空间查询优化方法

程昌秀 宋晓眉 著



科学出版社

基于执行代价的 空间查询优化方法

程昌秀 宋晓眉 著

国家自然科学基金委优秀青年科学基金项目(41222009)

国家自然科学基金委面上项目(41271405)

资助

科技部 863 课题(2007AA120401)

科学出版社

北 京

内 容 简 介

空间查询优化方法研究对丰富空间数据管理的理论体系有重要意义。本书结合空间数据和空间操作的特殊性,系统阐述了作者团队在空间查询优化方面的相关研究成果,主要包括:启发式空间查询计划生成方法、空间属性一体化代价评估模型、累计 AB 直方图及其选择率估计方法,并在开源数据库平台 Ingres 中开展了相关技术实现和验证工作。

本书可作为空间数据库高级管理人员和研发人员的参考用书。

图书在版编目(CIP)数据

基于执行代价的空间查询优化方法/程昌秀,宋晓眉著. —北京:科学出版社,2015.5

ISBN 978-7-03-044450-9

I. ①基… II. ①程…②宋… III. ①地理信息系统-查询优化-最优化算法-研究 IV. ①P208

中国版本图书馆 CIP 数据核字(2015)第 108313 号

责任编辑:彭胜潮 苗李莉 / 责任校对:张小霞

责任印制:张 倩 / 封面设计:铭轩堂

科学出版社出版

北京东黄城根北街 16 号

邮政编码:100717

<http://www.sciencep.com>

文林印务有限公司印刷

科学出版社发行 各地新华书店经销

*

2015 年 5 月第 一 版 开本:787×1092 1/16

2015 年 5 月第一次印刷 印张:9 3/4

字数:228 000

定价:59.00 元

(如有印装质量问题,我社负责调换)

前 言

面对“大数据”时代数据管理的需求，信息领域推出了云存储、NoSQL 数据库、列数据库等多种解决方案。为了切实实现大地理数据的高效管理，地理信息领域还需针对空间数据的特殊性开展数据管理、查询优化等方面的研究。

数据库曾 4 次荣获计算机领域的最高奖(图灵奖)，具有扎实的理论基础。在过去 30 多年间，数据库以其专业的数据管理能力及其出色的性能表现，推动了 Oracle、Sysbase、Ingres 等商业数据库产品的成熟与应用。在体量大、多样性强、速度快、价值密度高的“大数据”时代，关系数据库的相关理论显得有些复杂，但其对数据认识之深入、管理方法之完善还是值得品味和借鉴的。本书传承关系型数据库查询优化的方法与理论，结合空间数据体量大、结构复杂、运算(操作)代价昂贵等特点，阐述了我们在空间查询计划枚举、空间代价计算和空间选择率估计等关键环节的相关研究成果及其在开源数据库 Ingres 中的研发实现与实验，为空间查询优化方法的研究和应用提供新的理论依据。

本书共分 6 章，按下述逻辑撰写。第 1 章主要介绍空间查询优化的研究背景及理论依据，总体概括本书的主要研究目标和贡献。第 2 章介绍后续章节涉及的数据库内核的一些基本概念，介绍空间数据库的基本概念和相关理论，介绍数据库内核中查询优化的全过程，并简单剖析 Ingres 后台与查询优化相对应的各模块。第 3 章分析数据库常用查询计划枚举方法的优缺点，提出一种复合的空间查询计划生成方法，提出三种启发式减枝策略，避免不可行或低效的计划枚举；在 Ingres 内核中实现空间约束对的概念，并通过实验展示空间表与索引放置规则的有效性。第 4 章介绍关系型数据库的代价评估模型，展示空间查询案例在 Ingres 中的代价评估过程，通过实例表明关系代价模型忽略了空间选择率、空间扩展表读取、空间 CPU 代价等因素，直接导致查找最优查询计划的失败；此后，在传统评估模型框架下，第 4 章针对空间扩展表存储模式、空间索引、空间笛卡儿积、空间 KEY 连接以及空间 TID 连接等特殊特性，扩展了相应的空间代价评估模型，形成了一套空间属性一体化代价评估模型；在 Ingres 中加入上述空间代价评估模型后，执行上述空间查询，结果表明，该模型能较为准确地给出空间查询的代价，在该模型的支撑下 Ingres 查到了一个更优的空间查询计划。针对第 4 章代价评估模型的一个重要输入参数——空间选择率，第 5 章首先介绍传统数据库中常见的选择率估计方法(直方图方法)，但由于空间数据的多维性，导致传统属性直方图无法直接使用；目前有关空间直方图的研究尚不成熟，其中基于精细拓扑谓词的选择率估计、空间连接的选择率估计、空间直方图的推演等科学问题制约着空间直方图的发展和成熟；针对上述问题，本章提出了累计 AB 直方图，给出一系列空间查询操作的选择率估计方法以及直方图推演方法，并在 Ingres 中实验，实验表明累计 AB 直方图能有效地实现复杂

空间查询的选择率估计，为空间 I/O 代价的评估奠定良好的基础。第 6 章总结本书的创新之处，展望未来可以继续开展的研究工作。

特别感谢国家自然科学基金委优秀青年科学基金项目(41222009)、面上项目(41271405)以及 863 课题(2007AA120401)的资助。在上述成果的研究过程中，感谢颜勋博士、姜平硕士参与的 Ingres 内核代码解析工作，感谢朱焰炉硕士对空间直方图的探索研究，感谢胡夏天硕士生为 AB 直方图相关实验付出的辛勤劳动，感谢崔珂瑾硕士、李晓岚硕士为整理空间数据模型的相关材料付出的辛勤劳动。本书由程昌秀、宋晓眉执笔，最后感谢杨山力硕士生、杨静硕士、Nikita 博士生为本书的编排和校对付出的辛勤劳动。

本书系统、全面地介绍我们近年来在空间数据库内核中的研究成果，希望能为后续从事空间数据库内核研究的人员奠定一些基础，提供一丝启发，希望国内空间数据库内核研究能走得更远、更坚实。此外，由于作者能力有限，书中难免存在疏漏和不妥之处，恳请读者不吝赐教。

目 录

前言

第 1 章 绪论	1
1.1 背景与意义	1
1.2 研究范畴	2
1.3 研究目标及贡献	3
1.3.1 空间查询计划生成方法	3
1.3.2 空间查询代价评估模型	4
1.3.3 空间直方图选择率估计	5
第 2 章 空间数据库基础知识介绍	6
2.1 关系数据库系统内核基础知识	6
2.1.1 元组标识(TID)	6
2.1.2 物理存储格式	7
2.1.3 主索引与辅助索引	8
2.1.4 查询图	9
2.1.5 连接树	9
2.1.6 等价类	10
2.1.7 执行操作算子	11
2.2 空间扩展的相关基础知识	15
2.2.1 几何数据类型	16
2.2.2 空间操作与函数	19
2.2.3 空间索引	21
2.2.4 空间数据的表结构	27
2.2.5 空间数据库的执行操作算子	28
2.3 数据库查询处理流程与 Ingres 程序框架	31
2.3.1 数据库查询处理流程	31
2.3.2 Ingres 查询处理的程序框架	36
第 3 章 空间查询计划的生成	39
3.1 查询计划生成方法综述	39
3.1.1 穷举法	39
3.1.2 动态规划法	40
3.1.3 贪婪法	43
3.1.4 概率法	45

3.1.5	复合算法	53
3.1.6	小结	54
3.2	一种复合的空间查询计划生成方法	55
3.2.1	连接树形的生成	55
3.2.2	基于分块约束的表排列生成	58
3.2.3	操作枚举	61
3.2.4	空间启发式策略的加入	62
3.3	空间启发式策略的实现与实验	65
3.3.1	空间约束对在 Ingres 中的改进与实现	65
3.3.2	启发式策略在缩小计划枚举空间方面的作用	72
第 4 章	空间代价评估模型	75
4.1	空间代价评估研究综述	76
4.1.1	基于 R 树的空间选择代价	76
4.1.2	空间连接操作及其执行代价	78
4.1.3	空间算子的 CPU 代价	80
4.1.4	小结	81
4.2	基于查询树的 Ingres 代价评估模型	81
4.2.1	代价模型推演框架	81
4.2.2	连接代价的计算	82
4.2.3	计划树的代价	87
4.2.4	Ingres 代价评估示例(改进前)	87
4.3	扩展的空间代价模型	92
4.3.1	空间查询代价估算的特殊性	92
4.3.2	空间扫描代价	93
4.3.3	空间连接代价	94
4.3.4	Ingres 代价评估示例(改进后)	96
第 5 章	空间直方图及其选择率估计	99
5.1	空间选择率估计研究	99
5.2	现有空间直方图综述	101
5.2.1	MinSkew 直方图	101
5.2.2	SQ 直方图	101
5.2.3	CD 直方图及其修正估计方法	101
5.2.4	Euler 直方图及其扩展	103
5.2.5	PH 直方图	105
5.2.6	GH 直方图	106
5.2.7	PostGIS 直方图	107
5.2.8	小结	108
5.3	累计 AB 直方图的相关概念及核心操作	109

5.3.1	AB 直方图	109
5.3.2	累计 AB 直方图	110
5.3.3	核心操作函数	111
5.4	累计 AB 直方图的选择率估算	118
5.4.1	空间选择的选择率估计	118
5.4.2	空间连接的选择率估计	125
5.5	累计 AB 直方图的推演	128
5.5.1	面向空间选择的直方图推演	129
5.5.2	面向空间连接的直方图推演	130
5.6	实现与实验	130
5.6.1	相关系统实现	130
5.6.2	空间选择操作的选择率估计实验	131
5.6.3	空间连接操作的选择率估计实验	133
5.6.4	累计直方图的推演实验	135
5.7	小结	137
第 6 章	总结与展望	138
6.1	内容总结和结论	138
6.2	存在的问题与进一步的工作	139
参考文献		141

第 1 章 绪 论

1.1 背景与意义

位置信息服务是当今社会发展的重要方向(周成虎等, 2011), 大量位置信息数据(空间数据)亟需高效的空間数据管理与查询方案。虽然空間数据管理水平在某种程度上依赖于计算机领域的技术革命, 但是历次革命给我们留下最深刻的经验和教训往往是: 计算机领域的技术革命常常会给 GIS(geographic information science)披上了时尚的外衣, 时尚之后真正能沉淀下来的是地理信息人针对空間数据及其应用的特殊性, 不断发展和积累下来的相关基础理论和方法, 例如空間索引、9-交模型、影像金字塔等。

查询优化器是介于系统用户与操作系统间的一层用于提升查询性能的软件, 是数据库管理系统的灵魂。由于空間数据具有数据量庞大、数据结构复杂、操作代价昂贵等特点, 传统关系数据库的查询优化方法不能适应空間查询的需求, 导致难以实现海量空間数据的高效检索和存取(龚健雅, 2000), 因此, 空間查询优化势必成为空間数据管理的难点和突破点(方裕等, 2001)。近年来, 国内外有些学者开展了一些研究工作, 为空間查询优化的系统研究奠定了良好基础。但是, 计算机领域查询优化理论相当复杂, 其代码实现深入系统后台的内核, 且与各种底层操作交错在一起, 加大了空間查询优化研究的难度。本书结合空間数据、空間操作的自身特点, 在计划枚举、代价计算和选择率估计等关键环节上, 提出一系列基于代价的新优化策略和方法, 发展了空間查询优化方法与理论基础, 并在开源数据库 Ingres 中研发实现, 为空間查询优化方法的研究和应用提供新的理论依据。

也许有读者会质疑: 随着云存储、NoSQL 数据库的出现, 目前数据管理技术已呈现百花齐放的局面, 传统对象关系型数据库何去何从都难以判断, 而基于对象关系型数据库理论的空間查询优化技术又有多长的生命力? 就目前形式来看, 云存储、NoSQL 数据库、传统对象关系型数据库有其各自既定的应用场景, 传统对象关系型数据库依然有其用武之地。云存储、NoSQL 数据库更多是针对检索模式单一、检索速度快、伸缩性强的数据管理; 而传统对象关系型数据库能保证数据完整性与一致性, 在联机事务处理(OLTP)的应用中有较强优势。也许随着计算机领域硬件技术或操作系统软件的大变革, 传统对象关系型数据库可能会退出历史的舞台; 但其完善的基础理论和方法体系难免会被今后的其他数据库管理技术或其他一些不可预知的系统所采用。因此, 本书不是针对具体 Ingres 介绍一个空間查询优化系统, 而是以关系数据库查询优化的理论为基础, 重点介绍空間查询优化方面的相关基础算法和理论。这样不仅可以发展、完善 GIS 的数据管理理论体系, 也期望相关的基础算法和理论可以长远地服务于空間数据管理的研究或其他一些不可预知领域的研究。

1.2 研究范畴

在数据库中，一条查询语句(SQL)往往对应多个空间执行计划。如何准确、快速地从众多可选查询计划中选取执行代价(cost)最小(执行速度最快)的计划，提高空间数据库的查询效率，一直以来都是数据库理论与技术研究的难点与热点。Cost 在本书中多数情况下译为“代价”，根据语境需要有时也会译为“开销”。数据库查询代价是执行查询计划所消耗系统资源，主要是指消耗时间资源。

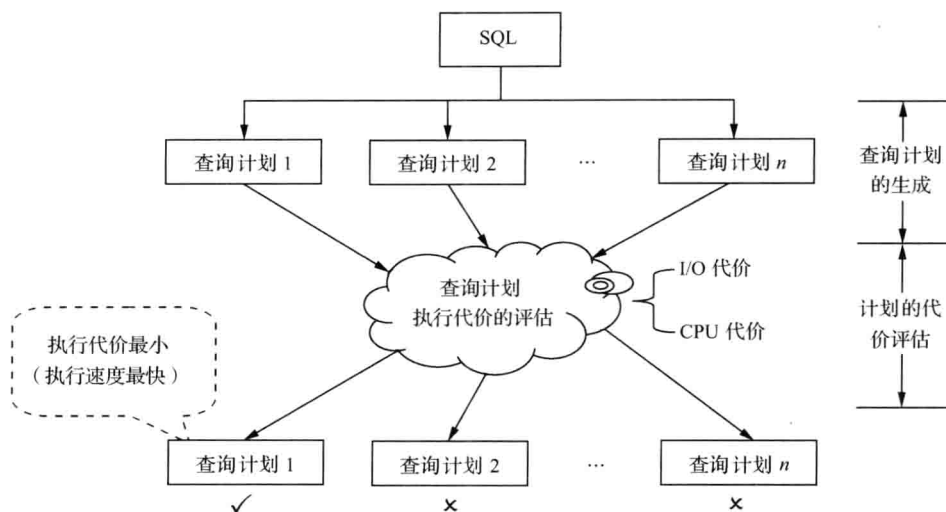


图 1-1 查询优化的过程示意

基于代价的查询优化至少包括查询计划生成、计划代价评估两个阶段(如图 1-1 所示)。空间数据库查询优化的处理流程及框架与图 1-1 基本一致，但由于操作数据类型的本质区别，造成了数据存储方式、索引机制、操作函数和统计信息管理等的巨大差异，因此需要结合空间数据自身的特点研究空间查询优化。

- 查询计划生成：生成所有可能的查询执行计划。在提高最优计划覆盖率的同时，尽可能缩小计划搜索空间。查询计划的生成主要包括连接树形枚举、表(含索引表)排序枚举、操作枚举三个阶段。其研究重点包括：空间索引参与的枚举(因为空间数据比普通数据对索引具有更强的依赖性)、空间操作等价的变换枚举、空间表连接方法的枚举以及启发式空间剪枝策略等；核心研究思路是：结合空间数据类型、数据操作等特殊特性，引导计划搜索器向最(较)优查询计划存在的方向进行，同时摒弃一些不合逻辑的空间查询计划。
- 计划的代价评估：根据上阶段生成的查询计划，评估各计划的执行代价，选择出代价最(较)小的计划。该研究应遵循以下原则：①快速、准确地估算出计划各环节的执行代价；②尽量减少系统在统计数据维护方面的资源开销、减少人工干预。执行代价包括 I/O 代价和 CPU 代价两方面。为了准确地估算计划树中各查

询操作的 I/O 代价,系统往往需要了解待查询表的数据统计、数据分布以及索引等相关信息。直方图是描述数据分布的重要方法;而基于直方图的选择率估计不仅能为查询代价评估提供重要的输入参数,而且能预先给用户演示最可能的查询结果(Liu et al., 2003)。空间直方图是目前研究的热点,涉及的主要科学问题有:多表连接的直方图推演、空间连接的选择率估计、精细拓扑谓词的选择率估计等。除上述第 1 个问题外,国内外已有少量算法可以处理后两问题,但是其估计精度及其可实现性都存在一定问题。至于 CPU 代价,空间查询与传统属性查询的重要区别是:空间查询操作通常是时间复杂度较高、较耗 CPU 资源的运算,而且空间操作函数的数量远比传统函数多,空间数据查询处理的细节也会有所不同(详见 2.2.5 节)。如果与传统数据库一样不重视空间查询 CPU 代价的估计,那么无论上层的枚举算法多么优越,也可能因忽视 CPU 代价而导致错误的评估结果,与最优的查询计划失之交臂(例如,4.2.4 节所示的例子)。

目前国内外在空间索引、查询算法优化、查询代价估算等分支领域有一定进展,但至今尚未提出一套完整的优化系统设计方案(蒋苏蓉等, 2004)。基于传统查询优化的成熟理论体系,研究空间查询计划的生成方法,并将空间操作代价融合到代价模型,发展空间属性一体化的查询优化理论体系,是一件重要且迫切的任务。

1.3 研究目标及贡献

本书继承了传统关系型数据库查询优化的方法与理论,结合空间数据体量大、结构复杂、运算(操作)代价昂贵等特点,对传统属性查询优化关键环节的理论和方法进行探索和完善,形成了空间属性一体化的代价评估模型,并在开源数据库 Ingres 中研发实现与实验,为空间查询优化方法的研究和应用提供新的理论依据。主要研究成果包括空间计划生成方法、空间查询代价评估模型以及空间选择率估计方法。

1.3.1 空间查询计划生成方法

目前查询计划生成方法较多,但在实际应用中常常存在各种各样的问题。穷举法和概率法都是按照既定的规则找到一个完整查询计划,再计算其执行代价。其区别在于:前者能枚举出所有查询计划,从理论上讲能找到最优计划;而后者更为简单、收敛速度更快。穷举法在表数目比较小的情况下,比较容易找到最优执行计划,但当表数目较多时,穷举法往往由于搜索空间过于巨大,导致查询代价预评估的系统开销过大,最终严重降低了系统整体执行的速度。而概率法则是在表数目较多的情况下采用一些算法(例如,快速选择、迭代改进、遗传变异、模拟退火以及禁忌搜索)加快计划查找的收敛速度,但当表的数目较少时,概率法可能会因不全面的搜索导致系统陷入局部最优、而错过全局最优的计划。动态规划法则先按照一定规则枚举出查询计划的所有子树,确定最优子计划,并记录在内存中,其后根据前期存储的最优子计划,直接生成当前的优选计划。动态规划法实际上是穷举法的改进,即通过增加少量的存储空间来加快计划枚举的

速度,故它能在较短时间内找到较优的查询计划。贪婪法是从众多表中选出行数少的表先执行查询操作,行数少的表其中间结果集也会较少,再用较小中间结果去过滤行数多的表。贪婪法的时间复杂度、空间复杂度都很低,甚至低于动态规划法的时间、空间消耗,但是付出的代价是:不能枚举出所有子计划,可能错过最优计划。此外,动态规划法和贪婪法最大的问题在于:尽管它们都在努力构造一个最(较)优计划,但是若在有限的时间内没找到最(较)优计划,其前期构造的次最(较)优计划就会被浪费,返回结果是没有查询计划,即优化器的作用完全消失。

总体看来,当参与查询的表(含索引表)数目不多于 10 时,建议采用完备性较强的穷举法。同时,借鉴动态规划法思想,记录枚举过程中产生的每个子计划和某完整计划的代价,在枚举其他的计划时可以避免重复的枚举和计算。当表(含索引表)数目超过 10 时,建议采用贪婪算法(例如,Ingres)或遗传算法(例如,PostGIS)查找计划。

本书第 3 章提出了一种复合的空间查询计划生成方法,涉及连接树、表排列生成方法以及相应的操作枚举方法;此外,根据空间连接的特点提出了一套空间启发式策略,避免了不必要查询计划的生成,缩小了枚举空间,最后进行了相关实证研究。

1.3.2 空间查询代价评估模型

目前空间查询优化评估模型的研究多集中于基于空间索引(例如 R-树系列)的代价,而空间与属性一体化的查询代价研究尚不成熟。本书的研究思路是在早期关系数据库代价模型的基础上进行调整。早期以 Faloutsos 等为代表的 R-树性能分析研究可以衍生出空间代价估计,但该方法严重依赖于 R-树索引的存在。此外,它们是针对 R-树索引结构,采用自上而下的方法,探讨索引节点的访问次数及其访问代价;而本书则不依赖索引的存在,而是基于查询计划树,采用自下而上的方法估计查询计划代价。Faloutsos 等探讨的 R-树代价可理解为一个仅涉及基表及其索引表的查询计划,或一个空间 KEY 连接操作的代价;而本书的方法则不局限于某种特殊的空间查询操作,而是针对整个复杂的空间查询计划进行评估;因此,基于查询计划树的代价模型具有更强的通用性。最后,基于 R-树的节点访问率的评估只适用于空间相交谓词;而本书采用空间直方图法则能精确估计出各空间拓扑关系的选择与连接操作的元组数,从而增加算法的适用范围、提高空间查询代价的评估精度。第 4 章以查询计划树为框架,以拓扑关系为主要空间操作,在开源 Ingres 上完善空间查询操作的代价估计模型。

Ingres 的代价主要来源于 I/O 代价和 CPU 代价。对数据表而言,I/O 代价主要是用元组个数和元组宽度来估算;对于索引而言,I/O 代价则是根据索引的填充因子来估算。CPU 代价的估算相对较为复杂,主要因为每个操作都要引起 CPU 的消耗,这种消耗主要与参与运算的元组数、扫描谓词(比较和等值等等)、连接语句(左连接\右连接\全连接)、存储方式(ISAM\HASH\B-树\R-树)、排序等多种因素有关。Ingres 中上述参数都是使用的是经验值,即根据数据元组个数和长度乘以经验常数构成。关于空间数据或空间操作的代价 Ingres 几乎没有考虑。第 4 章在详细分析前人研究内容的基础上,分析了 Ingres 代价模型和流程,并基于该框架对空间代价的估算进行了补充和完善。

1.3.3 空间直方图选择率估计

Güting(1994)在 VLDB 上定义了“空间数据库系统”的概念。他认为空间数据库要有独立的数据存储方式、字段、操作(连接)方式、索引等。空间优化方法必定与传统的代价模型有区别,但是流程框架应该是一致的。这个流程中一个很重要的参数是选择或者连接操作结果的元组个数(即选择率),这个参数往往需要直方图技术的支持。目前,传统属性直方图技术应用已经较为成熟,然而空间直方图的研究有待加强。

根据文献可知,除 CD 直方图和欧拉直方图外,其他直方图都不能精确估计空间选择操作的选择率;除改进欧拉直方图、PH 和 GH 外,其他直方图都不能实现空间连接操作的选择率估计;除改进欧拉直方图外,其他直方图都不能估计空间包含操作的选择率估计。为了解决上述问题,第 5 章提出了累计 AB 直方图,给出了一系列空间查询操作的选择率估计方法以及直方图推演方法,并在 Ingres 中实验,实验表明,累计 AB 能有效地实现复杂空间查询的选择率估计,为空间 I/O 代价的评估奠定良好基础。累计 AB 直方图不但可用于各类空间算子的选择率估计(例如包含、分离、北面、南面、距离米等),还可以将估计扩展到三维数据模型中。

第 2 章 空间数据库基础知识介绍

本书主要在对象-关系型空间数据库管理系统 Ingres 的基础上开展研究工作。所谓对象-关系数据库管理系统(ORDBMS)就是在关系型数据库管理系统(RDBMS)之上,通过用户自定义类型、自定义函数、自定义索引等技术,实现空间数据的存储、管理与分析。Oracle 的 Oracle Spatial、IBM 的 DB2 Spatial Extender、微软的 SQL Server Spatial、开源的 PostGIS 等都在不同关系数据库管理系统之上的空间扩展。

2.1 关系数据库系统内核基础知识

Ingres 是较早的数据库管理系统,开始于加利福尼亚大学柏克莱分校 Stonebraker 的一个研究项目。该项目开始于 20 世纪 70 年代早期,在 80 年代早期结束。像柏克莱大学的其他研究项目一样,它的代码使用 BSD 许可证(Berkeley Software Distribution license)。从 80 年代中期,在 Ingres 基础上产生了很多商业数据库软件,例如 Sybase、Microsoft SQL Server、Informix 等。80 年代中期启动的后继项目 Postgres,产生了 PostgreSQL、Illustra。由此可见,Ingres 是对关系数据库管理系统最经典、最全面、最具体的诠释。为此 Stonebraker 于 2014 年获得了计算机领域的最高奖项“图灵奖”。本书将以 Ingres 为例,介绍后续章节涉及的关系数据库内核的一些重要基本概念。

2.1.1 元组标识(TID)

在关系数据库中,每一条元组(记录)都有一个物理地址。TID(Tuple ID)是用来标志该地址的唯一标识。系统可以根据 TID 定位元组的页面及其偏移量。例如,在图 2-1

Empno	Name	Salary	...	TID		
1	Jim	10000	...	0	Page 0	Row 0
2	Nacy	12000	...	1		Row 1
3	James	15000	...	2		Row 2
4	Bill	20000	...	3		Row 3
5	Emily	9000	...	512	Page 1	Row 0
6	Kevin	11000	...	513		Row 1
7	...	...	...	514		Row 2
8	...	...	...	515		Row 3
9	...	...	...	1024	Page 2	...
10	...	...	...	1025		
11	...	...	...	1026		
12	...	...	...	1027		

图 2-1 Employee 表的元组标识 TID

的 Employee 表中, 每个元组都对应一个隐式 TID 值, 其中 Empno 为 6 的元组的 TID 是 513, 意味着该元组存储在第 2 个磁盘块(页面)第 0 行的位置上。因此, TID 仅用于系统内部的管理, 不支持用户的手动修改。

2.1.2 物理存储格式

在数据库中, 当数据表的元组存储到磁盘时, 可以有不同的组织形式。Ingres 提供多种类型的存储结构, 每种存储结构对于特定的查询应用具有不同的效果。具体存储结构如下。

1. HEAP 结构

这种结构的表也称堆(HEAP)组织表, 即按插入的顺序, 元组的存储位置依次增加。其特点在于: ①无索引; ②数据无序; ③查询需要全表扫描。

2. HASH 结构

基于这种结构的表也称为散列表(hash table), 即根据某元组中指定的关键词值通过一个 Hash 函数将其转换成一个整型数字, 然后用该数字对 Hash 桶的个数进行取余, 取余的结果当作数组的下标, 将该元组存储在以该数字为下标的数空间里。当对哈希表进行查询时, 根据条件中的键值通过 Hash 函数可以很快得到所需数据的存储地址, 直接取出数据, 返回给前端。

其特点在于: ①在基于关键字精确匹配中, 使用 Hash 函数能快速定位数据页面及位置; ②对于范围搜索或模式匹配查询, 则需要全表扫描; ③数据无序; ④对于关键字不唯一的情况, 如果散列存在溢出链需要进行溢出链页面的扫描。

3. ISAM 结构

数据存储位置按主键(primary key, PK)进行排序的一种索引结构, 故主键是唯一能够标定一行记录的字段或字段集。ISAM 结构是静态的, 因此当表中记录增删变化时, 可能会影响其他记录的存储位置。

其特点在于: ①可以根据完整关键字或者关键字的一部分(比如: 字符串关键字的左子串)来确定数据页面的位置; ②随机或者模式匹配搜索都可以使用索引; ③数据是物理排序的, 索引在创建时数据项依关键字严格排序, 随着记录的变化索引项将不准确; ④对于关键字不唯一的情况, 需要对溢出链页面扫描。

4. B-树结构

也是基于关键字排序的一种结构。使用多叉 B-树(balanced tree)作为索引, 具有动态增长、动态平衡的特点。使用 B-树获得的元组流总是有序的, 故适合于范围搜索。

这里需要注意: ①上述四种物理存储格式既适用于基表, 也适用于索引表。②在后续的论述中, 还常常提到 SORT 格式。SORT 是 Ingres 中将 HEAP 数据转换为有序数

据表的一种统称，它可能是内存中的一种格式，也可能以临时表的形式存储。Ingres 常采用快速排序法将 HEAP 结构转为 SORT 格式。

2.1.3 主索引与辅助索引

在 ISAM 存储方式中，关键字被称为主键(PK)或主索引，例如图 2-2 中 Employee 基表(EBT)中的 Empno 列。为了检索的需要，常常需要对数据表中的其他字段建立辅助索引(secondary key, SK)。辅助索引是独立于基表的另外一个表(即索引表)，它不影响基表的存储结构，例如图 2-2 中的 EIT 是对 EBT 表中 Salary 列建的索引表；其中，EBT 的 TID 列是隐式的，即物理上不存在该字段的值，其值表征每条记录存储的物理位置，而 EIT 的 TIDP 则是显示的，即实际存储了所指记录的物理存储位置。在有些数据库管理系统中，确定基表存储结构的主键也被称为“聚簇索引”，而辅助索引则被称为“非聚簇索引”。

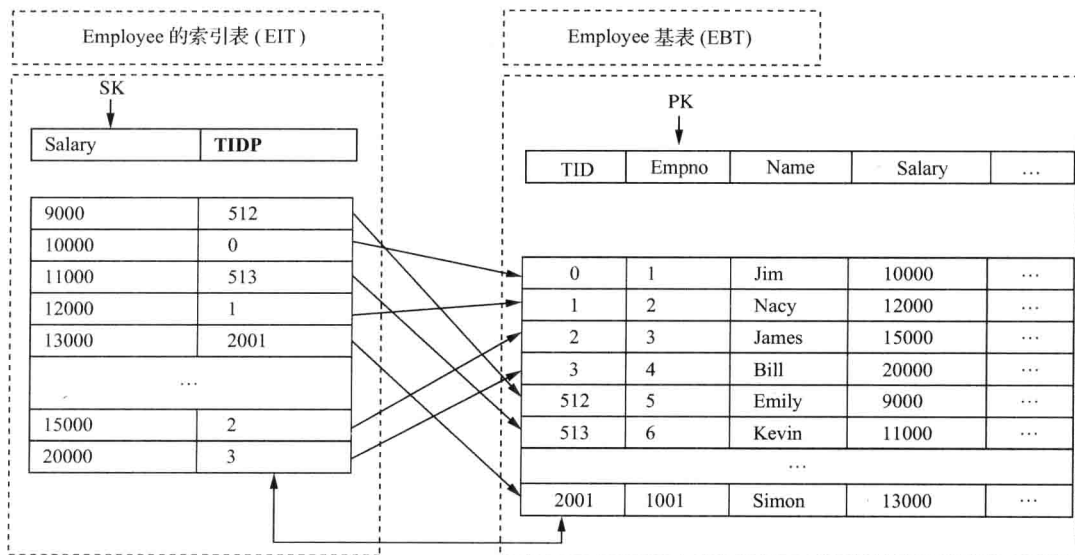


图 2-2 基表与辅助索引表物理数据组织

在查询优化研究中，需要注意：①可以通过 TIDP 与 TID 的连接，强制将辅助索引表加到执行计划的枚举中；例如在图 2-2 中，我们可以使用“EIT.TIDP = EBT.TID”的语句启用辅助索引。②在某些查询中，可采用索引表替换基表的方法来提高检索效率；例如，对于图 2-2 所示的实例，在 select TID from EBT where salary < 10000 的查询语句中，可以用 EIT 表来替换 EBT 表，根据 EIT 表 B-树的检索方式能很快定位出 Salary < 10000 的记录，并返回其 TIDP 值。③辅助索引表是独立于基表之外的另一张表，需要额外的系统开销，故实际应用中的辅助索引不易过多，最好根据查询需要适当建立。

2.1.4 查询图

数据库查询优化器的首要任务是根据各表之间的连接关系(查询图)生成初步的查询计划(连接树)。下面先介绍查询图。

查询图中各节点表示查询语句涉及的关系 $R_1, \dots, R_n$, 连接节点的边则表示查询语句中连接两端节点的连接谓词; 故查询图是包含 $R_1, \dots, R_n$ 的无向图。查询图理论上包括链状、星状、树状、环状以及团状几种不同的类型, 如图 2-3 所示。

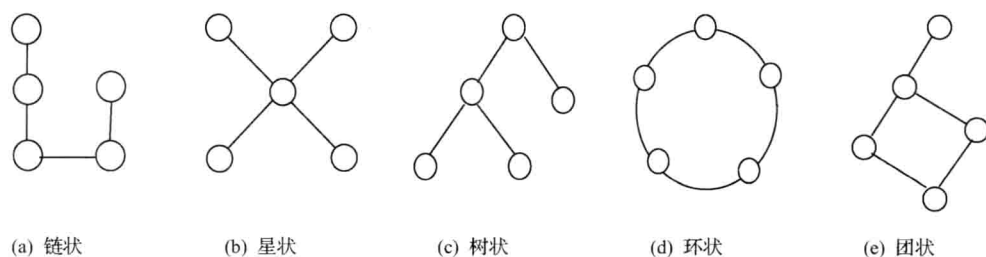


图 2-3 查询图的形状

查询图对查询计划中连接操作的选择有重要的影响, 即通常选用查询图中直接相连的两表参与连接。如果选择了查询图中不直接相连的表参与连接, 其执行代价通常会较大; 因此连接只能采用笛卡儿积生成结果, 算法比较耗时。查询图虽然示出了查询计划中所有可能的连接, 但并不指定连接执行的顺序。

2.1.5 连接树

连接树是数据库的内部连接顺序的一种表达形式; 它作为逻辑和物理优化的基础, 将直接决定查询计划的搜索空间。连接树主要有左深树(left-deep tree)、右深树(right-deep tree)、锯齿树(zig-zag tree)、浓密树(bushy tree)四种树形, 如图 2-4 所示。左深树和右深树两者类似, 每一个连接的左子树或者右子树中, 一定存在一个叶子节点表示关系 R_i 。锯齿树则可看成是左深树与右深树的结合, 即要么左子树是表示关系 R_i 的叶子节点, 要么右子树是表示关系 R_i 的叶子节点。对于浓密树来说, 左右子树没有任何

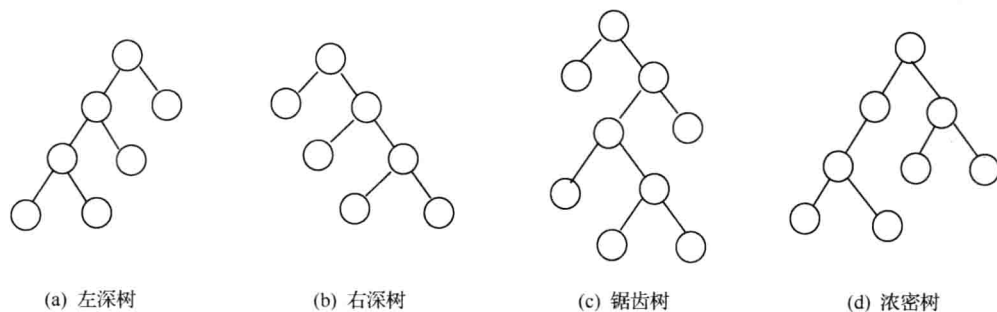


图 2-4 连接树形状