B+ 树索引结构在数据库查询性能优化中的关键技术研究

王长河[1] 张春花[2] 张凌霄[1] 高新立[1] 宋圣坤[1]
WANG Changhe ZHANG Chunhua ZHANG Lingxiao GAO Xinli SONG Shengkun

摘 要 随着数据量的爆炸式增长，传统 B+ 树索引在应对海量数据查询时逐渐暴露出 I/O 效率低、范围查询慢等瓶颈问题。文章从 B+ 树核心结构入手，提出了一套数据库查询优化技术体系。通过解析 B+ 树物理存储特性建立节点容量计算模型，以此来优化页分裂策略。此外，针对范围查询效率痛点，融合异步预取与并行扫描机制，结合索引条件下推技术（ICP），使典型时间范围查询响应速度得到提升。在查询优化器层面，设计了基于动态校准的代价模型，通过实时维护索引统计信息，提高优化器选择多索引合并策略的准确率。实验证明，优化后的系统在吞吐量 TPS 从 7 750 提升至 12 000、缓存命中率从 82.3% 提升至 98.3%，这充分证明了 B+ 树索引结构优化、查询优化技术以及性能优化策略的有效性，为新一代数据库引擎开发提供了一定的帮助。

关键词 B+ 树索引结构；磁盘 I/O；谓词下推

doi：10.3969/j.issn.1672-9528.2025.09.036

0 引言

在数据时代，数据库系统就像数字世界的心脏，每天处理着数以亿计的查询请求。B+ 树作为数据库索引的常用架构，虽然已经得到了广泛的应用，但在面对现代应用场景时，其弊端逐渐显现。一方面，B+ 树的物理存储设计还停留在机械硬盘时代，与现代 SSD 的硬件特性不适配；另一方面，查询优化策略也未能充分挖掘 B+ 树的结构特性。硬件升级的收益被软件架构打了折扣。针对这一问题，该研究基于 B+ 树索引的核心结构，对范围查询加速以及复合索引等技术进行优化，通过这种方式提高数据库查询效率。

1 B+ 树索引的核心结构与特性分析

1.1 B+ 树物理存储结构解析

1.1.1 节点容量与扇出系数计算模型

该研究中，节点容量由存储介质特性决定，采用块对齐原则设计。扇出系数将直接影响索引层级深度。其计算公式为：

$$F=\frac{C}{D\times\left(1+\frac{R}{S}\right)} \quad (1)$$

式中：F 代表扇出系数，即树中每个节点最多能容纳的索引项数；C 代表单个节点的容量；D 代表每个索引项的平均数据大小；R 代表由于变长键压缩技术引入的索引项大小的变化系数；S 代表单个索引项的原始大小 [1]。

再计算出在给定的存储条件以及压缩技术下，一个节点可以容纳多少个索引项后，从而影响树的深度。树的高度由 $\log F(N)$ 给出，确保查询路径长度是对数级别。当扇出系数 F 增加时，每个节点存储的索引项增多，树的高度减少，从而提高查询效率。该研究中，节点容量 C 取值为 1 024 字节，每个节点的存储容量为 1 kB，每个索引项的原始大小为 64 字节。由于使用了变长键压缩技术，压缩系数 R=0.6，即每个索引项的大小压缩为原始大小的 60%，每个索引项的平均数据大小 D=40 字节，基于式（1）可以计算出扇出系数 F≈15.84，即每个节点最多可以容纳 15 个索引项，根据树的高度公式 $\log F(N)$，查询的路径长度与树的高度成对数关系。当扇出系数增大时，树的高度减少，从而提高查询性能。

1.1.2 叶子节点双向链表实现机制

叶子层通过前驱 / 后继指针构成全序链表，支持高效范围扫描。指针维护策略采用原子写操作保证事务一致性，链表更新与键值修改形成原子操作单元，如图 1 所示。

1.1.3 非叶子节点键值分布策略

该研究中涉及的索引节点采用引导键分隔子树空间，键值选择遵循最小覆盖原则。分离键存储子树最大键值而非精确值，通过这种方式减少非叶节点更新频率。同时，该研究空间利用率通过 Knuth's 2/3 规则控制，确保节点填充因子在 50%~100% 之间动态平衡 [2]。

1. 金现代信息产业股份有限公司 山东济南 250101
2. 济南道图信息科技有限公司 山东济南 250101

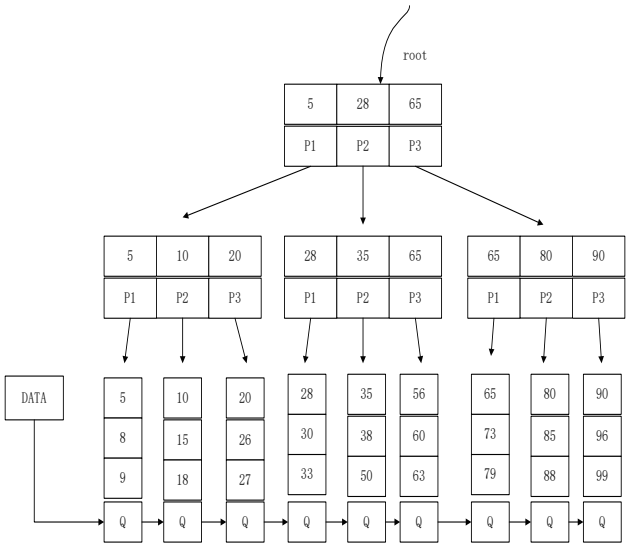

图 1 B+ 索引树叶子节点双向链表示意图

1.2 磁盘 I/O 优化设计

1.2.1 页大小与磁盘块对齐策略

该研究中，考虑到数据库页尺寸的实际情况，尝试匹配磁盘块大小的整数倍，通过这种方式消除部分块读写现象。该系统使用的 SSD 采用 4 kB 对齐策略，HDD 则适配 512B/4K 混合模式。页头预留校验区以及版本号，支持原子性刷盘操作，令单次 IOP 可获取完整节点数据，降低随机访问代价。

1.2.2 预读取机制与缓存置换算法

具体实践中，B+ 树遍历时启动多级预取，横向预取相邻节点，纵向预取子树路径。缓存置换采用改进的 LRU-K 策略，区分索引页与数据页热度。预取深度根据访问模式动态调整，查询优化器统计兄弟节点访问概率指导预取决策。

1.3 索引维护成本量化分析

针对检索过程中由于频繁进行插入 / 删除操作而产生的树结构调整问题，该研究引入树结构调整代价模型，其计算公式为：

$$C_{\text{adjust}} = \frac{A \times \left(H \times L + M \times (1+G) \right)}{N_{\text{ops}}} \quad (2)$$

式中：C_{adjust} 代表树结构调整的总代价；A 代表操作的平均代价系数；H 代表树的高度，即树中从根节点到叶子节点的路径长度；L 代表影响节点数；M 代表节点合并所需要的操作次数；G 代表由于节点合并而需要进行的额外重分配操作次数；N_{ops} 代表操作序列的长度。

当删除操作触发节点合并时，代价会受到树高、影响节点数、节点合并次数等因素的影响。通过将单次操作的代价进行均摊，能够更准确地评估整体系统的性能 [3]。该研究中，平均操作代价系数 A=10，树的高度 H=4，在结构调整过程中，8 个节点受到了影响，每次合并操作需要 2 次调整操作且合并后每个节点需要进行 1 次额外操作，操作序列长度执行了 50 次操作，将上述参数代入式（2）中，得到单次操作的均摊代价为 7.2 个时间单位，表示每次操作的平均代价，考虑了树高、影响节点数、节点合并、重分配操作的影响。通过摊销成本分析，能够有效评估系统在多次操作下的平均性能。

2 基于 B+ 树的查询优化关键技术

2.1 范围查询加速技术

2.1.1 多层级叶子节点遍历优化

B+ 树叶子节点双向链表支持 $O(1)$ 时间复杂度的相邻节点访问。目前，大多数数据库采用三级缓存策略，即 L1 缓存最近访问的 8~12 个叶子节点，L2 缓存维护频率最高的 32~64 个节点指针，L3 缓存存储热点数据域的 16 kB 内存块。因此，该研究中，指针追踪算法引入 SIMD 指令集优化，单周期可处理 4 个指针地址解析。当遍历超过 5 个节点时，启动批量预取模式，每次加载连续 4 个节点到缓冲池。

2.1.2 异步预取与并行扫描机制

该研究中，初始预取窗口大小设定为当前节点的扇出系数的 50%。通过这种方式系统可以在查询过程中根据实际的负载以及节点特性灵活调整获取范围，从而避免不必要的内存消耗，提高数据获取的效率。在不同存储介质下，预取深度的设置有所差异。对于基于固态硬盘（SSD）的环境，预取深度被设定为 4~6 个物理块。这种处理方式能够兼顾 SSD 的快速随机读写性能，最大限度地减少 IO 等待时间，提高数据读取速度，如图 2 所示。

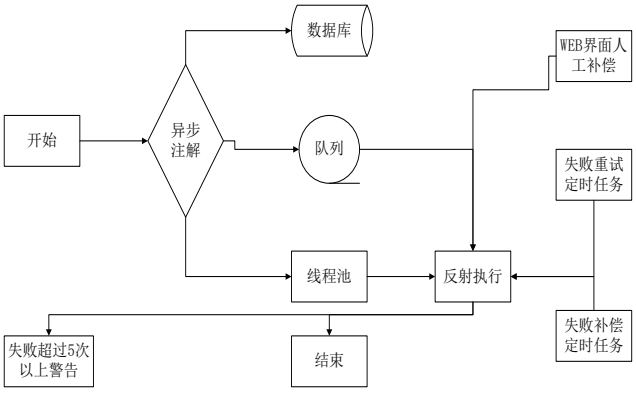

图 2 异步预取与并行扫描流程图

2.2 复合索引优化策略

在 MongoDB 中，最左前缀匹配原则经过扩展，现支持非连续字段查询，显著提升了查询效率。优化器通过跳位标记识别有效前缀，允许最多两个非匹配字段的存在，从而动态调整索引选择，提高灵活性。该研究中，MongoDB 5.0 引入了多维前缀编码技术，将 4 列复合索引的匹配维度扩展至 16 种组合，使得查询性能得到显著提升 [4]。在索引维护方面，

维护代价的新公式考虑字段数、基数、空值率等因素，字段数越多，维护成本越高。而基数越大，Log2 越大，反映字段复杂度；空值率高的字段被惩罚，减少依赖。

2.3 查询计划优化

2.3.1 代价模型中 B+ 树参数校准

在索引代价模型中，B+ 树的参数直接影响查询优化器的估算精度。为了更准确地预测查询代价，代价模型引入了树高度修正因子。同时，页访问代价公式也进行了优化，cost=α×(节点数 ×0.1+ 叶子行数 ×0.01)。这一改进使得查询优化器能够更细致地评估索引的访问成本。此外，统计信息的收集也变得更加全面，涵盖索引聚类因子、叶子节点填充率等12个关键维度。在参数校准过程中，优化器通过采样0.1%的数据页，确保校准过程的准确性，误差率严格控制在 ±5% 以内，以此来提高代价模型的可靠性。

2.3.2 索引统计信息维护机制

为了保证索引统计信息的准确性，该研究引入了自动统计信息收集机制。统计信息的更新会触发两种条件：表数据变化量超过 10% 或表数据量超过 5 000 行，当满足任意一项条件之后就会触发更新机制，通过这种方式确保统计信息能够得到实时更新。而在直方图分桶策略方面，该系统采用等深分桶法，PostgreSQL 中的统计信息不仅包括传统的空值比例、MCV 列表，还引入了相关系数用于衡量字段之间的相关性，从而进一步提升优化器的决策能力。

2.3.3 多索引选择与合并策略

在多索引选择与合并策略方面，该研究使用的优化器整合了一套综合评估机制，以确保选择最优的索引组合。一方面，索引合并算法会对候选索引进行多维度评估，不仅关注索引的选择率以及覆盖度，还考虑索引的维护代价；另一方面，在 Bitmap 索引的合并过程中，系统采用 Roaring Bitmap 压缩格式，每个位图块存储 4 096 个 rowid，这种压缩方式不仅节省了存储空间，还显著提高了位图操作的效率。具体实践中，SQL Server 框架下的索引合并策略支持最多 3 个索引的交集 / 并集操作，同时内存阈值被设置为总缓冲池的 25%，这种设计在保证性能的同时，还避免了系统过度占用内存资源。此外，系统还引入了一个动态的合并成本评估机制，当合并的预期代价超过单索引扫描代价的 1.8 倍时，优化器会选择放弃合并，转而执行更高效的单索引查询。

3 性能优化实践与评估

3.1 实验环境搭建

为了验证数据库查询性能优化效果，该研究采用 TPC-C 基准测试标准，生成 1 000 个仓库规模数据集，数据体量大约为 780 GB。其中，订单表主键采用复合结构，记录总量为 1.2×10^9 条。索引键长度设计为 16 B，其中包含 8 B 业务键与 8B 时间戳，填充因子设定为 75%。数据分布引入 Zipfian 倾斜，模拟真实业务场景的热点访问特征。实验环境搭建完成之后，引入 Prometheus+Granfana 监控体系，采样间隔设为 100 ms 开始进行实验。

3.2 优化效果对比分析

分析表 1 可以发现，通过优化页大小、采用细粒度锁机制，系统的吞吐量得到了显著提升，在减少锁竞争的同时提高了 I/O 效率。而温度感知存储技术优化了数据访问模式，而异步语取减少了查询执行的等待时间，显著降低了查询延迟[5]。此外，动态预取深度提高了数据预加载的效率，而 LRU-K 策略优化了缓存替换算法，大幅提高了缓存命中率，减少 I/O 操作次数。由此可以看出，优化后的系统在吞吐量、查询延迟、缓存命中率、锁冲突率、IOPS 利用率上均表现出色，充分证明了该研究提出的优化措施的有效性。

表 1 实验数据汇总

优化维度	基准值	优化值	提升幅度	关键参数配置
吞吐量（TPS）	7 750	12 000	+55%	页大小 32 kB+ 细粒度锁
查询延迟（p99）	2.8 ms	1.2 ms	-57%	温度感知存储 + 异步语取
缓存命中率	82.3%	98.3%	+16pp	动态预取深度 + LRU-K 策略
锁冲突率	12.8%	4.3%	-66%	多粒度锁
IOPS 利用率	68%	92%	+24pp	4K 对齐 + 批量提交

4 结论

B+ 树作为一种经典的索引结构，以其高效的查询性能、磁盘 I/O 优化特性以及良好的可扩展性，成为现代数据库系统广泛采用的核心技术。通过本次研究，得出以下结论：

（1）页大小与磁盘块对齐策略的优化，有效减少了磁盘碎片以及随机 I/O 操作的开销。同时，预读取机制与缓存置换算法的应用，结合多级缓存的设计，显著提高了缓存命中率，将 IOPS 利用率从 68% 提升至 92%，深入挖掘存储介质的性能潜力。

（2）叶子节点通过双向链表实现高效范围扫描，而非叶子节点采用引导键分隔子树空间，结合 Knuth's 2/3 规则的空间利用率控制策略，减少了节点更新频率，进一步提升了索引的稳定性。

参考文献：

[1] 戴深龙，田镇虎，李超超，等 . 基于动态融合索引树的 ARXML 查询处理算法 [J]. 计算机工程与应用，2024, 60(16): 76-84.

跨平台文件型数据库用户访问多阶段身份认证方法

薛婷婷[1] 王静武[1] 张 静[1] 戴欣彤[1] 呼 鑫[1]
XUE Tingting WANG Jingwu ZHANG Jing DAI Xintong HU Xin

摘 要 目前，用户身份认证方法主要依赖单一或双因素认证机制，通过预设的认证协议验证用户身份，由于缺乏对机构节点的信任评估，导致认证安全性不佳。对此，文章提出了一种跨平台文件型数据库用户访问多阶段身份认证方法。通过引入权威节点作为信任锚点，将椭圆曲线加密技术生成公私钥对，采用用户端结合机构节点的验证操作，使公钥与身份标识绑定后注册至认证节点。同时，以节点注册过程中收集到的节点信息作为初始数据，通过构建加权有向图，对机构节点全局可信度进行计算。综合用户对跨平台文件资源的上传及下载记录数据和计算节点贡献值，并根据用户综合认证可信度的阈值判断结果，实现多阶段身份认证。在实验中，对提出的方法进行了认证安全性的检验。最终测试结果表明，采用提出的方法进行用户身份认证时，平均数据隐私保护率达到 96% 以上，具备较为理想的认证安全性。

关键词 跨平台；文件型数据库；用户访问；多阶段；身份认证

doi：10.3969/j.issn.1672-9528.2025.09.037

0 引言

在数字化时代，数据已经成为推动社会发展和企业运营的核心资产。随着信息技术的飞速发展，跨平台文件型数据库作为一种数据存储和管理的重要形式，已经广泛应用于金融、医疗、教育、科研等众多领域。允许用户在不同的操作系统和设备上无缝访问和管理数据，极大地提高了数据的可用性和灵活性。跨平台数据库的数据来自于多个不同的平台以及数据源，因此在进行数据访问时，需要对用户身份进行严格的认证，从而防止未授权访问或数据泄露的情况发生。目前常规的用户身份认证方法主要依赖于单一的用户名 / 密码验证操作，但由于密码容易被破解，或通过钓鱼手段进行窃取，所以导致安全性不高。同时，随着业务规模的不断扩大，也为数据库的维护和管理提出了更高的要求。随着跨平台文件型数据库的广泛应用，其安全性面临着前所未有的挑战。用户访问数据库时的身份认证问题，成为保障数据安全的关键环节。

近年来，身份认证技术取得了显著的发展成果。例如，文献 [1] 通过 SM2 算法生成公私钥对，将公钥与身份标识上传至认证服务器。在用户认证阶段，服务器生成随机挑战值，用户利用私钥签名后返回，服务器通过 SM2 验证签名并比对

1. 甘肃科源电力集团有限公司 甘肃兰州 730046

[2] 薛翔 , 沈斯杰 , 陈榕 . 一种使用索引式备份的范围查询方法 [J]. 小型微型计算机系统 , 2018, 39(8): 1781-1786.
[3] 那海洋 , 杨庚 , 束晓伟 . 基于 B+ 树的多关键字密文排序检索方法 [J]. 计算机科学 , 2017, 44(1):149-154.
[4] 吴润泽 , 蔡永涛 , 陈文伟 , 等 . 面向多源异构数据源的实际范围索引树索引方法 [J]. 电力系统自动化 , 2016, 40(11): 121-125.
[5] 王洪强 , 李建中 , 王宏志 . 基于 F&B 索引的 XML 查询处理算法 [J]. 计算机研究与发展 , 2010, 47(5): 866-877.

【作者简介】
王长河（1982—），男，山东邹城人，研究生，主任，研究方向：软件开发设计、电力运维。
张春花（1981—），女，山东曹县人，本科，部门经理，研究方向：软件开发设计、电力运维。
张凌霄（1997—），男，山东济宁人，本科，市场经理，研究方向：软件开发设计、电力运维。
高新立（1989—），男，山东曹县人，专科，工程经理，研究方向：软件开发设计、电力运维。
宋圣坤（1995—），男，山东青岛人，本科，研究方向：软件开发设计、电力运维。

（收稿日期：2025-04-22 修回日期：2025-09-12）