小型微型计算机系统
Journal of Chinese Computer Systems
DOI:10.20009/j.cnki.21-1106/TP.2025-0105
2026年4月 第4期
Vol.47 No.4 2026

基于最小化 AES 的鲁棒计划选择方法

段坤仁,黄煜坤,方焱志,彭煜玮,彭智勇

(武汉大学 计算机学院,武汉 430072)
E-mail:ywpeng@whu.edu.cn

摘 要: 鲁棒计划选择通过计划图简化和计划切换能够减少由基数估计误差带来的数据库查询优化器计划决策错误.现有方法在计划图简化过程中仅关注了局部选择率空间内的计划鲁棒性,忽视了全局抗误差能力.针对该问题,本文提出平均误差比指标用于全面评估计划在整个选择率空间内的鲁棒性,并设计了最小化平均误差比计划图简化算法,在减小计划图规模的同时,确保简化后计划集合的全局鲁棒性.实验结果表明,相比于现有方法,本文方法生成的计划图在简化效率和计划集合质量上都有明显提升,能在统计信息滞后、数据更新频繁或复杂连接查询的场景下有效减少因优化器计划决策错误导致的性能退化,为数据分析型应用提供了可靠保障.

关 键 词: 鲁棒计划选择;计划图简化;平均误差比
中图分类号: TP392 **文献标识码:** A **文 章 编 号:** 1000-1220(2026)04-0776-08

Robust Plan Selection Method Based on Minimized AES

DUAN Kunren, HUANG Yukun, FANG Yanzhi, PENG Yuwei, PENG Zhiyong
(School of Computer Science, Wuhan University, Wuhan 430072, China)

Abstract: Robust plan selection mitigates erroneous query optimization decisions caused by cardinality estimation errors through plan diagram reduction and plan switching. Existing methods focus only on the robustness of query plans within a local selectivity space during plan diagram reduction, neglecting their global error resistance. To address this issue, we propose the average error scale as a novel metric to comprehensively evaluate the robustness of plans across the entire selectivity space. Accordingly, we design a plan diagram reduction algorithm that minimizes the average error scale while reducing the size of the plan diagram, ensuring the global robustness of the retained plan set. Experimental results demonstrate that, compared to existing methods, our approach significantly improves reduction efficiency and the quality of the resulting plan set. It effectively mitigates performance degradation caused by erroneous optimizer decisions in scenarios involving outdated statistics, frequent data updates, or complex join queries, thereby providing reliable support for data analytics applications.

Keywords: robust plan selection; plan diagram reduction; average errorscale

0 引 言

查询优化器作为数据库系统的核心组件,其决策质量直接影响查询执行效率.自 System-R 论文[1]发表以来,基于代价的优化器已被广泛采用,该架构通过枚举候选计划、估算执行代价并选择最优方案.其中基数估计(即预测查询操作中间结果集规模)的准确性尤为关键,研究表明[2]基数估计误差对执行计划质量的负面影响显著高于代价模型偏差.这类误差通常源于统计信息陈旧、属性独立性假设失效、复杂谓词解析误差等多维度因素,更值得注意的是误差在查询计划树中具有传播放大特性[3].尤其在多表连接场景中,初始阶段的微小估计偏差可能通过查询计划树被逐级放大,最终导致执行计划质量的显著退化.上述分析表明,基数估计的准确性已成为制约优化器决策质量的重要瓶颈.

为突破这一瓶颈,让优化器做出更好的计划选择,学界主要从两个方向展开探索:其一是探索各种提升估计准确度的方法(如直方图[4,5]、数据摘要[6]、采样[7]、基于模型的估计[8-12]等).然而,在数据分布倾斜、复杂谓词解析及统计信息缺失等场景下,此类方法难以有效提升基数估计的准确性,其适用性存在显著局限;另一类研究思路则接受基数估计不准这一现实,通过其他方式[13,14]减少次优化风险.例如,鲁棒查询处理[15-18](Robust Query Processing,RQP)技术通过构建误差容忍机制为解决该问题提供了系统性框架.

本文聚焦鲁棒查询处理(RQP)中的鲁棒计划选择方法[15],其核心目标是筛选在基数估计存在误差时仍能保持执行时间与资源消耗稳定的执行计划.N. Reddy 等人提出的计划图(Plan Diagram)[19]作为该领域的重要解决方案,通过离散化选择率空间,将多维选择率空间划分为互斥子区域,并为每个区域绑定具有鲁棒性的执行计划,获取鲁棒计划集合.在此基础上,计划切换(Plan-switching)方法[20]通过运行时反馈机制实现集合内自适应调整,从而确保最终执行计划的性能下限.然而,随着查询谓词复杂度的增加,计划图的维度扩展

收稿日期:2025-03-07 收修改稿日期:2025-04-15 基金项目:国家重点研发计划项目(2023YFB4503604)资助. 作者简介:**段坤仁**,男,1997年生,硕士研究生,CCF会员,研究方向为数据库系统;**黄煜坤**,男,1999年生,研究方向为数据库系统;**方焱志**,男,1998年生,研究方向为数据库系统;**彭煜玮**(通信作者),男,1980年生,博士,副教授,CCF会员,研究方向为数据库系统、大数据、数字水印;**彭智勇**,男,1963年生,博士,教授,博士生导师,CCF会士,研究方向为大数据安全、数据管理、数据库、数据挖掘与分析.

导致候选集合规模过大,显著影响优化器决策效率.为应对此问题,计划图简化技术[21]通过合并相邻子区域压缩集合规模,提升鲁棒计划选择效率.其合并依据为替换计划在对应区域内比被替换计划有更好的鲁棒性指标(Robustness Metrics).但现有方法提出的指标只考虑了局部空间的鲁棒性对比,可能导致整个选择率空间内潜在的风险被忽视,从而无法保障简化后集合的整体鲁棒性.

基于上述研究中存在的问题,本文提出基于最小化平均误差比(Average Error Scale,AES)的鲁棒计划选择方法,该方法能提升鲁棒计划选择方法在复杂连接查询和统计信息滞后等场景下的性能表现.本文的主要贡献如下:

1)提出了 AES 鲁棒性指标,用以量化计划在整个选择率空间内受基数估计偏差影响的性能降级程度,相比于局限于局部区域评估的方法更全面地评估了计划的鲁棒性;

2)构建了基于 AES 最小化准则的计划图简化算法(Plan Diagram Reduction Based on Minimized AES,PBMA),在压缩计划图规模的同时有效平衡鲁棒性保障,为计划切换方法提供更轻量化且高可靠的候选集合;

3)在 TPC-H 和 JOB 基准数据集测试中,特别是在模拟高频更新导致的统计信息滞后场景下,本文方法在计划图简化效率与质量上均展现出优势,有效提升了鲁棒计划选择机制的性能.

1　相关工作

1.1　计划图与计划图简化

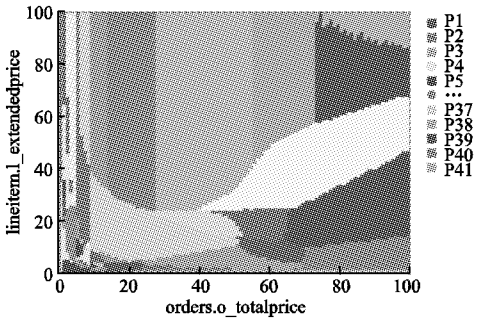
计划图是一种使用颜色编码的用于查询优化的执行计划图示,为优化器提供预计算决策支持[19].一个具有 n 个参数的查询模板的计划图是一个 n 维空间,其中的每一个维度对应查询模板的一个参数的选择率(从 0 ~ 100%),因此计划图也常被称为选择率空间(Selectivity Space,SS).查询模板的 n 个参数值确定后形成的查询可以被映射为该查询模板的选择率空间中的一个点,这个点被称为该查询在选择率空间中的查询点.对于一个确定的查询点,由于其各个参数值的选择率确定,因此该查询点的最优执行计划也能被确定.在计划图中会为每一个查询点按照其最优执行计划着色,如果两个不同的查询点具有相同的最优计划,则它们会被使用相同的颜色.

```
select *
from part, lineitem, partsupp, orders
where
ps_partkey = l_partkey
and p_partkey = l_partkey
and o_custkey = p_partkey
and |orders.o_totalprice|
and |lineitem.l_extendedprice|;
```

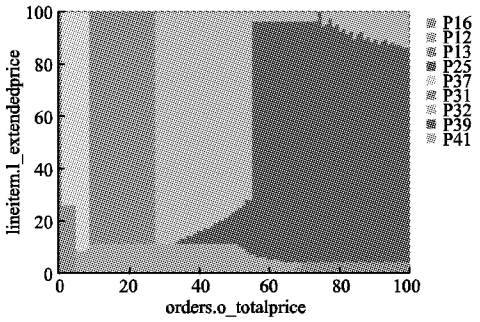
图 1　示例查询模板 *QT*

Fig. 1　Example query template *QT*

计划图的构建通常采用离线方式,根据设定的分辨率参数 *res*(详见 3.2 节)将原本连续的选择率离散化成一定间隔的选择率,从而在选择率空间中生成一系列的离散查询点.遍历这些查询点并调用查询优化器生成每个点对应的执行计划集合.例如考虑图 1 所示的基于 TPC-H 数据集的查询模板 *QT*,其中参数 o_totalprice 和 l_extendedproice 构成了二维选择率空间.在分辨率参数值取 100 时,*QT* 对应的计划图实例(基于 PostgreSQL 生成)如图 2(a)所示,其中每个色块代表特定选择率区间内的查询点对应的最优计划,共有 41 个计划,可以看到计划图可能是极其复杂且密集的,这使得鲁棒计划的选择面临计算复杂度和决策质量的挑战.



(a)Plan diagram



(b)Reduced plan diagram

图 2　*QT* 对应的计划图和简化计划图

Fig. 2　Plan diagram and reduced plan diagram of *QT*

Harish 等人[21]提出了计划图简化方法来压缩计划集合规模.该方法基于一个考虑:若原计划 P_a 和替换计划 P_b 在所有被覆盖区域内查询点的执行代价增量 $cost(P_b) - cost(P_a)$ 不超过用户可以接受的阈值 α,则在该区域内用 P_b 替代 P_a.例如 *QT* 的计划图可以简化为图 2(b)所示,只剩下 9 个计划.也就是说,大多数原始计划都被完全覆盖,计划集合规模大大降低.基于此,学界陆续发展了代价贪心算法[21]、阈值贪心算法[21]以及抗选择率误差算法(SEER)[22]等改进方法.其中代价贪心算法在计算复杂度上有较大改进;阈值贪心算法对合并效率进行改进,使得同一查询在相同的代价阈值范围内,简化得到的计划集合内的计划个数更少;SEER 算法则提升了决策质量,其保障计划图简化后,剩余计划相对原有计划在代价上的单调递减性.然而,现有方法仅考虑了替换计划在局部选择率空间内的性能降级,缺乏对计划在选择率空间全局内抵抗误差能力的考虑,会造成计划集合鲁棒性的下降.

1.2　鲁棒计划选择框架

实际使用中,计划图简化算法常用于为计划执行期的计划切换方法提供稳定计划集合,减少计划切换中使用不稳定计划带来的额外代价.基于计划切换[20]方法的鲁棒计划选择框架如图 3 所示,向计划图生成器中预先输入查询模板 *QT*,确定 *QT* 中的参数谓词并离散化谓词选择率空间 *SS*.将空间中的查询点 q_i 输入数据库优化器获取该查询点对应最优计

划 p_i,生成计划图 P. 随后通过计划图简化算法,获得简化后的鲁棒计划集合对应的计划图 P'.

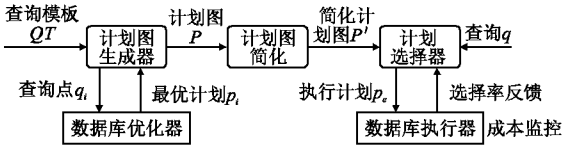


图 3 鲁棒计划选择框架

Fig. 3 Robust plan selection framework

当用户在线提交查询实例 q 后,计划选择器从简化后的候选集合中选取初始执行计划 p_e. 若运行时监控模块检测到其实际执行代价超过预定义代价约束,系统将排除当前计划并从候选集合依据监控模块反馈的实际选择率选择新的执行计划,直到满足约束条件,最终保障在基数估计存在误差的情况下仍能在代价约束范围内完成查询.

2 基于最小化 AES 的计划图简化

本章聚焦于鲁棒计划选择框架中的计划图简化模块,其生成的候选计划集合是决定计划选择器效率的关键. 通过对示例二维计划图的分析,揭示现有简化方法的鲁棒性指标只考虑了替换计划在局部选择率空间内的性能降级的问题,提出使用平均误差比对计划在选择率空间的鲁棒性进行评估,并给出基于平均误差比的计划图简化算法 PBMA.

2.1 问题描述

定义 1. (计划图简化). 对给定的计划图 P,给定最大代价阈值 $\alpha(\alpha\geqslant 0)$,生成最小计划基数的简化计划图 P',对于 P 中的任意计划 p_i 满足以下条件:

$$p_i\in P' \text{或} \forall q\in P, \frac{c_j(q)}{c_i(q)}\leqslant(1+\alpha) \tag{1}$$

其中 q 表示计划图中的查询点,$c_i(q)$ 代表计划 p_i 在该点的查询代价,$c_j(q)$ 表示 P' 中替换 p_i 的计划 p_j 在该点的查询代价,式(1)即为判断两个计划是否能够在查询点 q 发生替换的替换规则. 因此,在选择率空间 SS 中,原有计划图中的区域由以下两类构成:

1)稳定区域 SO:该区域的查询点未被其他计划覆盖,该区域内的计划称为稳定计划.

2)不稳定区域 SOE:该区域的查询点被其他计划覆盖,该区域内的计划称为不稳定计划.

通过对计划图简化,可以将不稳定区域内的计划由稳定区域内的计划进行替代.

定义 2. (误差比)(Error Scale,ES). 误差比的计算公式如下所示:

$$ES(q_e,q_a)=\frac{c_e(q_a)}{c_a(q_a)}, ES\in[1,+\infty) \tag{2}$$

其中,$c_e(q_a)$ 表示优化器使用统计信息估算的查询点 q_e 对应的最优计划 p_e 执行实际选择率查询点 q_a 所用代价;$c_a(q_a)$ 表示 q_a 对应的最优计划 p_a 的执行代价. 通过误差比可知优化器当前执行计划产生的性能降级比例.

根据上述定义,考虑计划图简化后如图 4 所示的计划替换情况,当优化器估计的查询点 q_e 落在 p_r 替换 p_e 的区域内时,替换前,误差比为:

$$ES(q_e,q_a)=\frac{c_e(q_a)}{c_a(q_a)} \tag{3}$$

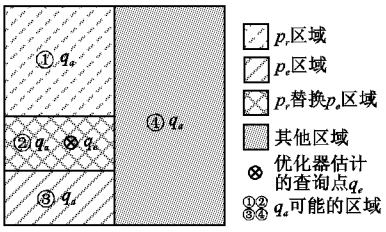


图 4 计划图覆盖示意图

Fig. 4 Illustration of plan coverage in plan diagram

替换后,误差比为:

$$ES'(q_e,q_a)=\frac{c_r(q_a)}{c_a(q_a)} \tag{4}$$

考虑实际查询点 q_a 所在位置的情况,计划替换后有如表 1 所示的 4 种情况. 可以看出,在整个选择率空间内,计划图简化过程中的查询点计划替代不一定保证替代计划的性能降级程度优于替换前计划. 因此,目前的计划图简化算法在计划替换过程中除满足覆盖条件外,还需限制其他条件. 例如文献[22]引入平均选择率估计错误抵抗因子(Average Selectivity Error Resistance Factor, AvgSERF),提出的 SEER 算法要求 $AvgSERF$ 大于 0,$AvgSERF$ 计算公式如下:

表 1 计划覆盖后的误差比分析表

Table 1 Error scale analysis table after plan coverage

q_a 所在位置	误差比变化
p_r 区域	$\frac{ES'(q_e,q_a)}{ES(q_e,q_a)}<1$
p_e 区域	$\frac{ES'(q_e,q_a)}{ES(q_e,q_a)}<1+\alpha$
p_r 替换 p_e 区域	$\frac{ES'(q_e,q_a)}{ES(q_e,q_a)}<1+\alpha$
其他区域	$\frac{ES'(q_e,q_a)}{ES(q_e,q_a)}$ 不确定

$$AvgSERF=\frac{\sum_{q_e\in SOE}\sum_{q_a\in SO}SERF}{\sum_{q_e\in SOE}\sum_{q_a\in SO}1} \tag{5}$$

其中,分子部分 $SERF$ 为:

$$\begin{aligned}SERF&=1-\frac{c_r(q_a)-c_a(q_a)}{(1+\alpha)\times c_e(q_a)-c_a(q_a)}\\&=1-\frac{ES'(q_e,q_a)-1}{(1+\alpha)\times ES(q_e,q_a)-1}\end{aligned} \tag{6}$$

当 $SERF$ 大于 0 时,表示替换后的误差比小于替换前误差比的 $1+\alpha$ 倍,小于 0 时表示替换后的计划带来了性能降级,$SERF$ 越大,替换后的计划稳健性就越好. 当计划 p_r 满足替换规则时,SEER 算法会计算该计划对应的 $AvgSERF$,只有当其大于 0 时才用 p_r 替换 p_e,否则不进行计划替换. 然而,$AvgSERF$ 的定义表明,该值计算过程中考虑的是图 4 中的 q_a 位于标号为 4 的位置的情况,也就是实际查询点的最优计划是稳定计划而优化器估计的查询点的最优计划是不稳定计划的情况,没有考虑图 4 所示的所有情况,也就是说 $AvgSERF$

大于 0 时无法保障替换计划在整个选择率空间都不会带来性能下降,因此需要设计新的限制条件来保障计划替换的有效性.

2.2 平均误差比

计划图中计划的鲁棒性指标可视为发生计划替换后计划的性能降级比,其应当考虑计划简化过程在整个选择率空间内引入的误差比影响. 在上述概念的基础上,本文提出计划鲁棒性可由选择率空间内的平均误差比衡量. 其定义如下:

定义 3. (平均误差比 Average Error Scale,AES):

$$AES = \frac{\sum_{q_e \in SS} \sum_{q_a \in SS} ES(q_e, q_a)}{\sum_{q_e \in SS} \sum_{q_a \in SS} 1} \tag{7}$$

AES 从整体上评估优化器处理查询模板 QT 的性能降级比例. 对于选择率空间内任意一个优化器的估计选择率对应的查询点 q_e,该定义考虑了实际选择率对应的查询点 q_a 可能位于选择率空间中的任意位置. 覆盖了大批量的 UPDATE 操作或者统计信息远远落后等极端情况.

对于选择率空间中的任一查询点 q_e,该点对于 AES 的贡献 $AESC(q_e)$ 为:

$$AESC(q_e) = \frac{\sum_{q_a \in SS} ES(q_e, q_a)}{\sum_{q_a \in SS} 1} = \frac{\sum_{q_a \in SS} \frac{c_e(q_a)}{c_a(q_a)}}{\sum_{q_a \in SS} 1} \tag{8}$$

查询点的 $AESC(q_e)$ 仅与该点选择率下的最优计划 p_e 有关,与该查询点对应的选择率空间位置无关. 因此,$AESC(q_e)$ 可替换为计划 p_e 对 AES 的贡献,即 $AESC(p_e)$. 当查询点 q_e 发生计划替换时,计划 p_e 被替换为 p_r,该点的贡献 $AESC(p_r)$ 为:

$$AESC(p_r) = \frac{\sum_{q_a \in SS} \frac{c_r(q_a)}{c_a(q_a)}}{\sum_{q_a \in SS} 1} \tag{9}$$

由公式(7)、公式(8)可得:

$$AES = \frac{\sum_{q_e \in SS} AESC(p_e)}{\sum_{q_e \in SS} 1} \tag{10}$$

定义 4. (最小化 AES):

$$min(AES) = \frac{\sum_{q_e \in SS} min(AESC(p_e), AESC(p_r))}{\sum_{q_e \in SS} 1} \leqslant AES \tag{11}$$

即,对于一个查询点来说,选择满足替换规则的计划中平均误差比贡献最小的计划进行替代,可保证替换后的计划平均误差比小于等于替换前计划.

2.3 PBMA 算法

结合前述分析,本文基于最小化 AES 提出了 PBMA 计划图简化算法. 算法的流程如算法 1 所示:

算法 1. PBMA(基于最小化 AES 的计划图简化算法)

输入: 原始计划图 P,代价阈值 α,选择率空间 SS

输出: 简化计划图 P'

1. 查询点数目 $m = |SS|$,原计划图 P 计划数目 $n = |P|$
2. 初始化计划 $AESC$ 贡献数组 $AESC[n] \leftarrow \{0\}$
3. 初始化计划替换查询点集合 $planSet[n][\,] \leftarrow \varnothing$
4. **for** 查询点 $q_i \in SS(i = 1, \cdots, m)$ **do**
5. **for** 计划 $p_j \in P(i = 1, \cdots, n)$ **do**
6. $AESC[j] += c_j(q_i)/c_a(q_i)$
7. **endfor**
8. **endfor**
9. **for** 查询点 $q_i \in SS(i = 1, \cdots, m)$ **do**
10. **for** 计划 $p_j \in P(i = 1, \cdots, n)$ **do**
11. **if** $c_j(q_i)/c_a(q_i) \leqslant (1+\alpha)$ 且 $AESC[j] < AESC[a]$ **then**
12. 覆盖计划 $p_r = p_j$
13. **else then** $p_r = p_a$
14. **endif**
15. $planSet[r] \cup q_i$
16. **endfor**
17. **endfor**
18. $P' \leftarrow \{p_i \mid planSet[i] \neq \varnothing (i = 1, \cdots, n\}$
19. return P'

算法 1 首先计算各个计划在选择率空间 SS 中的平均误差比贡献 $AESC[i]$. 在计划替换阶段对于符合替换规则的计划进行平均误差比贡献的比较,选择平均误差比最小的计划作为该点的最优计划,若无计划满足替换规则,则该点选择执行代价最小的计划 p_a,并加入到对应计划替换查询点集合 $planSet$ 中. 最后去除 $planSet$ 中为空的集合,剩余集合即为简化后的计划图 P'.

如 1.2 节研究现状所述,鲁棒计划选择框架在动态切换执行计划时只考虑计划图简化后生成的计划集合,其计划集合的规模应该尽可能小. 考虑该方面要求,本文改进 PBMA 算法,尽可能保证生成的计划图的最优计划集合最小. 最小集合 PBMA 算法如算法 2 所示:

算法 2. Min-PBMA(最小集合 PBMA 算法)

输入: 原始计划图 P,代价阈值 α,选择率空间 SS

输出: 简化计划图 P'

1-17. 与算法 1 相同

18. 初始化最优集合 $Opt[\,][\,] \leftarrow 0$
19. **while** $planSet$ 不为空 **do**
20. 初始化 $maxPoints \leftarrow 0$,$maxPlanIdx \leftarrow 0$
21. **for** 计划集合 $plan \in planSet$ **do**
22. $maxPoints = min(maxPoints, |plan|)$
23. 若 plan 的集合更大则更新 $maxPlanIdx$
24. **endfor**
25. $Opt = Opt \cup plan$
26. **for** 计划集合 $plan \in planSet$ **do**
27. **for** 查询点 $point \in planSet[maxPlanIdx]$ **do**
28. $plan.\ remove(point)$
29. **endfor**
30. 清除 $planSet$ 中的空集
31. $P' \leftarrow \{p_i \mid Opt[i] \neq \varnothing (i = 1, \cdots, n\}$
32. return P'

最小集合的 PBMA 算法将满足算法 1 第 11 行中关系的查询点归属到对应计划的集合中,然后使用集合合并算法,以贪心思路剔除 $planSet$ 中的最大集合元素,被剔除元素归入计划图的最优集合中,直至 $planSet$ 为空,所得最优集合即为 PBMA 算法下的最小最优计划集合.

2.4 高维度扩展

本文以二维谓词查询模板为例进行说明,但 PBMA 算法不仅适用于低维场景,其对于高维度查询模板也具有普适性.

由 2.3 节算法伪代码可知,PBMA 的实现与维度大小无直接关联. 算法实现过程中仅依赖离散选择率空间 SS 对应的查询点集合 $q_i \in SS(i=1,\cdots,m)$ 作为输入,选择率空间构建方法详见 3.2 节所述的选择率空间离散化流程. 谓词维度 d 只会影响选择率空间 SS 生成的查询点集合规模,也就是 PBMA 在 $q_i \in SS(i=1,\cdots,m)$ 上的计算量,不会影响算法流程. 综上,PBMA 算法适用于任意维度的查询模板.

3 实 现

3.1 选择率注入

为了构造计划图中的查询点,本研究在 PostgreSQL 中实现了对 SQL 查询语句中谓词选择率的注入,从而获得优化器在该点的最优计划决策. 通常选择率注入有两种方法:模拟注入和优化器注入. 模拟注入主要针对过滤谓词,利用直方图数据构造布尔表达式,从而人为设定不同选择率场景,但该方法无法支持连接谓词且依赖数据库直方图统计信息. 相比之下,优化器注入方法不仅精度更高,而且对数据库统计信息依赖性较低,因此更适合于复杂场景. 本文主要通过以下 3 个部分实现 PostgreSQL 的优化器注入:

1)语法扩展:修改 scan.l 和 gram.y 文件,在 raw_parser 函数中利用 Lex 与 Yacc 生成的代码实现词法和语法分析,扩展查询语句语法以支持注入表达式;

2)系统表调整:解析器根据操作符类型选择对应操作函数,而操作符与选择性计算函数的关联由 pg_operator 系统表管理,尤其是 oprrest 字段. 因此需修改 pg_proc.dat 和 pg_operator.dat 文件,重新编译生成 BKI 文件,并在 initdb 初始化过程中生成新的系统表;

3)函数实现改造:在函数管理器(fmgr)中,通过将注入的选择率(作为第 2 个操作数)包装成 Datum 类型,直接返回预设值用于选择率估计. 对于连接谓词,由于涉及两个关系属性且现有查询语法不支持三元操作符,本方案在 Join_Expr 结构中新增 injectsel 属性,并在语法解析时引入"joinexpr:numeric"表达式,使得注入的选择率在后续的连接选择率计算中被识别并应用.

设计的注入语法如下所示:

SELECT *

FROM t1 *JOIN* t2 *ON* t1.*id* = t2.*id*:0.03

WHERE t1.*id* ~ 0.04;

假设 t1 和 t2 的元组数分别为 200 和 300,在对连接谓词注入了选择率 0.03 之后,其结果集的基数可计算为 $200\times300\times0.03=1800$,过滤谓词的注入结果则为 $200\times0.04=8$. 通过上述方法,实现了对选择率估计过程的控制,从而可以获取优化器在该点的最优计划决策.

3.2 选择率空间离散化

选择率空间 SS 的离散化通过对各选择率谓词维度进行等比例切割方式完成. 决定谓词选择率切割之后份数的参数为分辨率 res,其选择将在 3.3 节详细讨论. 将分割后的选择率代入查询模板 QT,得到对应的查询点 q,查询点集合即为离散化选择率空间 SS. 具体过程如算法 3 所示.

3.3 参数选择

分辨率 res 直接决定了选择率空间离散化的粒度. 分辨率越大,离散化的选择率空间就越逼近真实的连续选择率空间,但同时会增大计算量,影响计划图生成的效率. 因此,分辨率值的设置应权衡模型精度和计算成本,确保其在合适的范围内.

具体来说,当选择率空间的分辨率较小时,可能使得离散化的选择率空间与真实的选择率空间相差较大. 若查询模板对应执行计划的搜索空间比较大,部分计划只出现于特定的选择率条件下,会造成计划图中缺失这些计划. 因此当查询中包含的表连接越多、谓词的维度越高,则可能生成的执行计划数量越多,此时需要设置更大的分辨率. 选择率空间的分辨率应当综合当前查询的复杂程度以及谓词的维度数量考虑. 本文给出二维查询模板且表连接数量不超过 6 个时分辨率的最佳值. 以查询模板 QT 为例,其计划图种类个数随分辨率的变化,如图 5 所示.

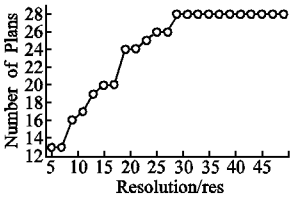


图 5 QT 的计划数量随分辨率 res 的变化

Fig. 5 Number of plans for QT varying with resolution (res)

图中曲线达到稳定状态(即曲线斜率趋近于零)时对应的分辨率 res 称为临界分辨率. 实验表明,在二维查询模板场景下,临界分辨率实测值普遍低于 35. 基于上述分析,本文提出分辨率参数配置原则:对于二维模板且表连接数不超过 6 的典型查询负载,将分辨率设置为 35 可有效平衡计算效率与离散化精度需求.

算法 3. 选择率空间离散化算法

输入:分辨率 res,谓词维度 d,查询模板 QT

输出:离散化选择率空间 $SS=\{q_1,\cdots,q_m\}$

1. 初始化 $SS\leftarrow\varnothing$,$m\leftarrow1$
2. **for** $i\in\{1,\cdots,n\}$ **do**
3. $m* = res$
4. **endfor**
5. **for** 索引 $index=\{1,\cdots,m\}$ **do**
6. 初始化选择率数组 $sel\leftarrow\varnothing$,临时变量 $t\leftarrow index$
7. **for** 维度 $i=\{1,\cdots,d\}$ **do**
8. $j = t$ **mod** res
9. 中值选择率 $sel[i]=(j+0.5)/res$
10. $t/=res$
11. **endfor**
12. 生成查询点 $q=Compose(QT,sel)$
13. $SS[index]=q$
14. **endfor**
15. **return** SS

代价阈值 α 是计划图简化的核心参数,该参数保证了在估计准确场景下,替换计划对现有查询的性能降级的边界值. 该参数越小,简化后的计划图引入的额外代价越小,同时,原计划图中的可替换关系越少;该参数越大,则引入的额外代价越大,原计划图中的可替换关系越多,更容易满足计划替换条

件,发生计划替换,但也会带来更大的性能降级比. 但当代价阈值增长到一定程度时,图中已无新的替换关系,此时,继续增大代价阈值已无效果. 因此需要选取合适的代价阈值,平衡简化后的计划图规模和性能降级比.

首先定义计划图性能降级比,当 AES 为 1 时,则表明该空间内计划图中的计划种类个数为 1,不存在计划选择带来的性能降级问题,该情况即为 AES 理论最小值. 因此,计划图性能降级程度(Performance Degradation,PD)即可定义为:

$$PD = AES - 1 \tag{12}$$

而计划图简化算法在稳定性上的提升可以通过比较简化之后的计划图与简化前的计划图的 PD 比值,该值即为计划图性能降级比(Performance Degradation Scale,PDS),定义为:

$$PDS_{algorithm} = \frac{PD_{algorithm}}{PD} \tag{13}$$

该值 <1 时,则说明简化算法生成计划图相比简化前的计划图性能降级程度更低;该值 >1 时,说明简化算法在原计划图基础上降低稳健性. 该值越小,则说明该计划图对于估计误差的抗干扰能力更强.

本文选取多个二维查询模板,研究在分辨率为 35 条件下,PBMA 算法的 PDS 值随代价阈值 α 的变化关系. 该关系如图 6 所示.

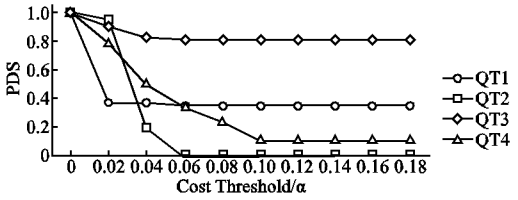


图 6　PDS 随代价阈值的变化

Fig. 6　PDS variation with cost threshold

实验表明,PDS 随代价阈值增大而减小,同时,当代价阈值增大到一定值后,PDS 趋于稳定. 可以看出,多数查询模板在代价阈值达到 0.1 时,PBMA 算法的 PDS 值已经达到稳定,可以认为该情况下的计划图中,满足最小化 AES 的代价替换关系已经完全包含代价覆盖关系. 综上所述,本文建议使用 PBMA 简化算法时,对于二维查询模板使用场景,将代价阈值设定为 0.1 较为合适.

4　实　验

4.1　实验环境

本文基于 PostgreSQL 11 进行了实现和实验,所有实验均在配备有 Intel(R) Xeon(R) CPU E5-2603 v4 和 16GB 内存的服务器上进行,操作系统为 Ubuntu 22.04. 实验采用 TPC-H[23] 和 JOB[2] 两个基准测试的数据集,规模分别为 1GB 和 4GB. 计划图简化算法相关的分辨率和代价阈值等参数参照 3.3 节设置.

4.2　实验结果

如图 7 所示,本文比较了 PBMA 算法和 SEER 算法简化后计划图的 PDS 值. PBMA 算法简化后的计划图的 PDS 总是小于 SEER 算法得到的计划图的 PDS. 并且,对于 JOB 中 Q1a、Q2a 以及 Q4b 等几个查询,SEER 算法简化后的计划图相比原有计划图鲁棒性降低,而 PBMA 算法仍能保证对计划图鲁棒性的提升. 分析这些查询发现,SEER 算法失效通常是因为查询模板的易错谓词中存在 EXIST/IN 运算符. 其根源是 SEER 算法假设查询执行代价随关系选择率的增大而增大,从而保证计划代价函数在选择率空间内的非递减属性,但该假设在包含 EXIST/IN 运算符的查询中并不满足,当关系选择率变大时,EXIST/IN 运算符对应的操作算子的代价并不一定变大[14] 在此类场景下,PBMA 算法相较 SEER 算法有更强的适用性.

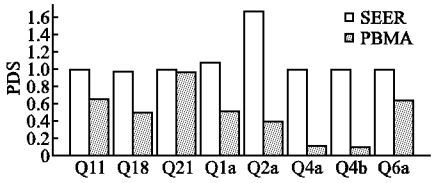


图 7　简化计划图 PDS 对比

Fig. 7　Comparison of PDS of reduced plan diagrams

计划图简化算法的另一目的,是为了减少计划图中计划数量. 更少的计划数量可以减少后续使用计划图时的计算量. 例如,鲁棒计划选择框架执行层面中的 Plan Bouquet 算法[13] 需要利用计划图生成计划代价曲面,用于后续的动态计划切换,若计划图中的计划种类过多,则动态切换过程对于选择率的变更非常敏感,带来不必要的计算量. 简化后的计划图包含计划数量越少,则说明简化算法的效率越高. 简化算法的简化效率(Simplification Efficiency,SE)可定义为:

$$SE = \frac{\text{简化前计划图计划数量} - \text{简化后计划图计划数量}}{\text{简化前计划图计划数量}} \tag{14}$$

本文还对比了 SEER 算法、PBMA 算法以及最小集合 PBMA 算法在 SE 上的表现情况.

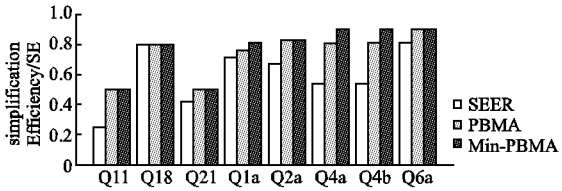


图 8　计划图简化算法 SE 对比

Fig. 8　SE comparison of plan diagram reduce algorithm

如图 8 所示,PBMA 算法及最小集合 PBMA 算法在简化计划图的效率上普遍优于 SEER 算法. 最小集合 PBMA 在部分查询模板上的 *SE* 表现要好于 PBMA,但多数情况下,两者表现一致. 同时可以看到,JOB 基准测试上的 PBMA 算法大多高于 TPC-H 基准测试上的表现. 其原因一方面是 JOB 基准测试的查询多以表间连接为主,通常一个查询中包含 4 个以上的表参与连接,而多表连接使得 PostgreSQL 中的优化器在进行计划搜索时的范围更大,从而使得原有计划图中的计划种类更多,因而可合并简化的部分更多;另一方面,尽管可以控制部分谓词的选择率变化,从而使得计划中的部分子树结构变更,增加计划图的计划种类,但是 TPC-H 基准测试查询通常嵌套多个子查询以及较多的过滤谓词,因此最终计划图

的计划种类并不多,可简化合并空间更少.

基于前述的计划图简化算法,本文在 PostgreSQL 中实现了类似于图 3 的鲁棒计划选择框架. 比较了鲁棒计划选择框架与原始的 PostgreSQL 优化器的最大次优比(Maximum Sub-Optimality,MSO)[20],其定义如下:

$$MSO(QT) = \max_{q_e,q_a \in SS}(ES(q_e,q_a)) \tag{15}$$

对于给定的查询模板 QT,该值表示的是实际执行代价与最优计划代价的最大比值,反映了计划执行的性能下限. 测试结果如图 9 所示,其中纵坐标为对数尺度的 MSO 值. 在 TPC-H 测试集上,鲁棒计划选择框架的鲁棒性相比原生优化器并不明显,在 Q3、Q10 等查询中相比原生优化器产生了降级. 分析 TPC-H 部分查询的执行计划后可以发现,由于多表连接中存在规模较小的表,导致连接顺序较为固定(如在 Q10 中小表 nation 总会被先连接),导致不同计划之间的代价差异较小,此时鲁棒计划选择中计划切换的额外开销就造成了 MSO 增大,性能下限变低.

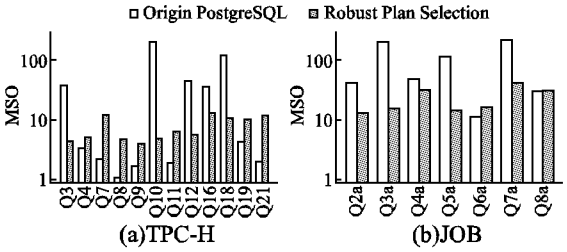


图 9 最大次优比对比

Fig. 9 Comparison of MSO

而在 JOB 测试集上,由于数据集大小远大于 TPC-H,且连接模型更复杂,因此在计划切换带来的额外开销相对于选择错误连接顺序产生的开销几乎可以忽略不计,切换为更优的执行计划后能带来显著的性能提升(例如 Q7a 提升了 81.31%),所以鲁棒计划选择框架整体表现更优.

因此,如果想在数据分析型场景中使用鲁棒计划选择框架来保障查询的鲁棒性,还需要在离线阶段对查询的相关特性进行提前分析,结合实验结果来看,当查询的数据集较大、涉及的连接表多且复杂时,使用鲁棒计划选择框架可以有效地防止查询性能急剧下降.

表 2 被更新字段说明

Table 2 Updated field description

数据表名称	字段名	字段解释
SUPPLIER	S_ACCTBAL	供货商支付
PART	P_RETAILPRICE	零件单价
PARTSUPP	PS_SUPPLYCOST	供货商零件出价
CUSTOMER	C_ACCTBAL	顾客支付
ORDERS	O_TOTALPRICE	订单总价
LINEITEM	L_EXTENDEDPRICE	在线商品价格

为了验证本文方法在实际场景中的表现,我们对 TPC-H 数据集进行随机更新,并测试更新后的查询执行情况. 本文选取的更新字段以及对应解释如表 2 所示.

实验中使用 Shuffle 算法[24]随机打乱表中数据,随机抽取表中占比 γ 的元组,并对这些元组的目标更新字段值进行正态分布随机处理,从而模拟实际情况下的数据变动. 其中 γ 的值从 10% 开始以 10% 为增量递增到 100%. γ 值反映了实际场景中统计信息的落后程度,γ 值小表明统计信息更新间隔小,统计信息新鲜度高;反之,则说明统计信息更新间隔大,统计信息更加陈旧. 对于每种 γ 值,实验中执行查询 3 次取执行时间平均值以减小误差,实验结果如图 10 所示.

从图 10 可以看出,在 γ 值较小时,PBMA 算法的执行耗时相较于 SEER 算法和不使用简化计划图算法相比要稍高一些. 这是由于本文方法实现过程中,使用 pg_hint 插件指定计划来完成计划的选择与切换,引入了额外的开销,使得总执行时长在数据库优化器估计准确度较高时,没有优势,图 10 中 PBMA_WITHOUT_HINT_COST 曲线与 PBMA 曲线的差值显示了计划切换带来的额外开销. 随着 γ 值的增大,优化器估计准确度逐渐降低,选择的执行计划并非最优计划,此时 PBMA 算法与 SEER 算法对于查询的优化提升逐渐显现. 并且,在 γ 值超过 50% 时,PBMA 算法的优化效果消除了使用 pg_hint 插件进行计划指定和切换带来的额外开销,在平均执行时间上由于原始优化器和 SEER 算法. 而且,对于 2.2 节所述的大批量更新或者统计信息远远落后的极端情况,本文方法的鲁棒性优势也较为明显.

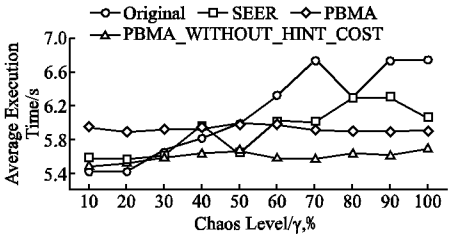


图 10 查询执行平均时长对比

Fig. 10 Comparison of average query execution time

从实验结果可以看出,PBMA 算法对于未构建统计信息或统计信息已过时的场景具有较好表现. 而当统计信息较新或数据集不频繁改变时,SEER 算法效果更好. 由于 PBMA 在统计信息较新或不频繁更新场景下表现一般,在下一步工作中考虑设计合理的机制在 PBMA 和 SEER 算法之间进行切换. 例如结合机器学习的手段根据用户的查询或者工作负载智能的判断统计信息的变化并自动的切换到相应的算法,从而提高鲁棒选择框架在统计信息变化过程中的性能下限.

5 总 结

本文围绕鲁棒查询处理中的计划图简化问题展开研究,提出了一种基于最小化平均误差比的计划图简化方法. 为鲁棒计划选择提供了新的解决方案,在数据分析型场景中,尤其是涉及大规模数据集和复杂连接时,本文的方法能够显著提升查询执行性能的稳定性. 未来我们还将聚焦于进一步优化算法的扩展性,并探索其在动态环境下的自适应调整机制.

References:

[1] Selinger P G, Astrahan M M, Chamberlin D D, et al. Access path selection in a relational database management system [C]//Proceedings of the ACM SIGMOD International Conference on Management of Data (SIGMOD), 1979: 23-34.

[2] Leis V, Gubichev A, Mirchev A, et al. How good are query optimizers, really? [J]. Proceedings of the VLDB Endowment, 2015, 9(3): 204-215.

[3] Ioannidis Y E, Christodoulakis S. On the propagation of errors in the size of join results[C]//Proceedings of the ACM SIGMOD International Conference on Management of Data (SIGMOD), 1991: 168-177.5.

[4] Ioannidis Y. The history of histograms(abridged)[C]//International Conference on Very Large Data Bases(VLDB), 2003: 19-30.

[5] ZHANG Y, YANG Z S, WANG H Z, et al. Compressed histogram based similarity join size estimation for dirty database[J]. Journal of Chinese Computer Systems, 2012, 33(10): 2113-2120.

[6] Cai W, Balazinska M, Suciu D. Pessimistic card-inality estimation: tighter upper bounds for intermediate join cardinalities[C]//Proceedings of the ACM SIGMOD International Conference on Management of Data(SIGMOD), 2019: 18-35.

[7] Hu P, Motik B. Accurate sampling-based cardinality estimation for complex graph queries[J]. ACM Transactions on Database Systems, 2024, 49(3): 1-46.

[8] Lan H, Bao Z F, Peng Y W. A survey on advancing the DBMS query optimizer: cardinality estimation, cost model, and plan enumeration[J]. Data Science and Engineering, 2021, 6(1): 1-16.

[9] LI G L, ZHOU X H, SUN J, et al. A survey of machine learning based database techniques [J]. Chinese Journal of Computers, 2020, 43(11): 2019-2049.

[10] Liu H H, Peng Z Y, Zhang Z, et al. MSP: learned query performance prediction using metaInfo and structure of plans[C]//Asia-Pacific Web (APWeb) and Web-Age Information Management (WAIM) Joint International Conference on Web and Big Data (APWeb-WAIM), 2023: 3-18.

[11] Li Y, Wang L W, Wang S, et al. A learned cost model for big data query processing [J]. Information Sciences, 2024, 670 (C): 120650-120668.

[12] YU Y, PENG Y W. A query optimization algorithm based on learning to rank[J/OL]. Computer Science, 1-12, http://kns.cnki.net/kcms/detail/50.1075.TP.202503 21.1437.022.html, 2025-03-25.

[13] GU J G, WANG Y S, ZHU T T, et al. SPES: SPARQL query optimization based on predicate selectivity estimation [J]. Journal of Chinese Computer Systems, 2017, 38(9): 1983-1987.

[14] ZHOU Y, WEI X F, XIONG H, et al. CSPRJ: MapReduce join query algorithm based on data skew[J]. Journal of Chinese Computer Systems, 2018, 39(2): 367-371.

[15] Yin S, Hameurlain A, Morvan F. Robust query optimization methods with respect to estimation errors[J]. ACM SIGMOD Record, 2015, 44(3): 25-36.

[16] Xiu H, Agarwal P, Yang J. PARQO: penalty-aware robust plan selection in query optimization[J]. Proceedings of the VLDB Endowment, 2024, 17(13): 4627-4640.

[17] YU X, CHAI C L, ZHANG X N, et al. AlphaQO: robust learned query optimizer[J]. Journal of Software, 2022, 33(3): 814-831.

[18] Wolf F, Brendle M, May N, et al. Robustness metrics for relational query execution plans[J]. Proceedings of the VLDB Endowment, 2018, 11(11): 1360-1372.

[19] Reddy N, Haritsa J R. Analyzing plan diagrams of database query optimizers[C]//International Conference on Very Large Data Bases(VLDB), 2005: 1228-1239.

[20] Dutt A, Haritsa J R. Plan bouquets: afragrant approach to robust query processing[J]. ACM Transactions on Database Systems, 2016, 41(2): 1-37.

[21] Harish D, Darera P N, Haritsa J R. On the production of anorexic plan diagrams[C]//International Conference on Very Large Data Bases(VLDB), 2007: 1081-1092.

[22] Harish D, Darera P N, Haritsa J R. Identifying robust plans through plan diagram reduction[J]. Proceedings of the VLDB Endowment, 2008, 1(1): 1124-1140.

[23] TPC-H[EB/OL]. http://www.tpc.org/tpch, 2024-07-05.

[24] Ellis J, Krahn T, Fan H. Computing the cycles in the perfect shuffle permutation[J]. Information Processing Letters, 2000, 75(5): 217-224.

附中文参考文献:

[5] 张　岩,杨忠胜,王宏志,等.基于压缩直方图的劣质数据库上相似连接结果大小估计[J].小型微型计算机系统,2012,33(10):2113-2120.

[9] 李国良,周煊赫,孙　佶,等.基于机器学习的数据库技术综述[J].计算机学报,2020,43(11):2019-2049.

[12] 余　阳,彭煜玮.基于学习排序的查询优化算法[J/OL].计算机科学, 1-12, http://kns.cnki.net/kcms-/detail/50.1075.TP.20250321.1437.022.html,2025-03-25.

[13] 顾进广,王岩松,朱婷婷,等.SPES:基于谓词选择率估计的SPARQL查询优化方案[J].小型微型计算机系统,2017,38(9):1983-1987.

[14] 周　娅,魏夏飞,熊　晗,等.CSPRJ:基于数据倾斜的MapReduce连接查询算法[J].小型微型计算机系统,2018,39(2):367-371.

[17] 余　翔,柴成亮,张辛宁,等.AlphaQO:鲁棒的学习型查询优化器[J].软件学报,2022,33(3):814-831.