

数据库参数配置智能调优研究综述

李奕言^{1),3)} 田季坤^{2),3)} 蒲 照^{1),2)} 李翠平^{1),3)} 陈 红^{1),2)}

¹⁾(中国人民大学信息学院 北京 100872)

²⁾(数据库与商务智能教育部工程研究中心 北京 100872)

³⁾(数据工程与知识工程教育部重点实验室 北京 100872)

摘 要 数据库系统具有大量的参数,这些参数控制了系统的内存分配、I/O优化、备份与恢复等诸多方面,极大地影响着数据库的性能。随着数据库和应用程序的规模和复杂性的增长,传统依靠数据库管理员手动配置参数的方式已经越来越难以满足用户需求。数据库参数配置智能调优将机器学习技术应用到参数调优领域,依据负载信息、数据库参数和性能,借助机器学习算法推荐一组最优的参数。本文针对现有参数配置智能调优技术,从调优方法、应用情况和未来挑战三个方面依次进行梳理和总结。首先将现有参数调优方法依据所用算法不同分为五类,从原理、技术、优缺点等方面对各类方法进行详细介绍和总结。之后介绍当前工业界主流的参数调优工具,分析参数配置智能调优在实际应用过程中遇到的问题及原因。最后,本文对数据库参数配置智能调优的未来研究方向进行了展望。本文旨在帮助研究者掌握当前数据库参数配置智能调优领域主流方法及面临的问题,以推动后续研究工作的开展。

关键词 机器学习;参数调优;贝叶斯优化;强化学习;智能数据库

中图法分类号 TP18 **DOI号** 10.11897/SP.J.1016.2024.01901

Research Survey on Intelligent Tuning of Database Knobs Configuration

LI Yi-Yan^{1),3)} TIAN Ji-Kun^{2),3)} PU Zhao^{1),2)} LI Cui-Ping^{1),3)} CHEN Hong^{1),2)}

¹⁾(School of Information, Renmin University of China, Beijing 100872)

²⁾(Engineering Research Center of Database and Business Intelligence, MOE, Beijing 100872)

³⁾(Key Laboratory of Data Engineering and Knowledge Engineering, MOE, Beijing 100872)

Abstract The database system encompasses numerous configuration knobs that govern various aspects of its operations. These knobs cover a wide spectrum of database functionalities including memory allocation, I/O optimization, backup and recovery processes. The performance of a database system is heavily influenced by how these knobs are tuned, making their optimization a matter of significant importance. As databases and their associated applications continue to grow in scale and complexity, the conventional approach of manually adjusting these configuration knobs by database administrators is proving to be increasingly inadequate. In response to this emerging challenge, the realm of intelligent database knob tuning has surfaced as a promising solution. This innovative approach leverages machine learning techniques to automate the optimization of database knobs. By analyzing workload information, the settings of the database knobs, and performance metrics, intelligent tuning technologies are capable of recommending an optimal set of knobs that enhance database performance. This paper endeavors to conduct a

收稿日期:2023-08-08;在线发布日期:2024-04-30。本课题得到国家重点研发计划(2023YFB4503600)、国家自然科学基金(U23A20299,62072460,62172424,62276270,62322214)资助。李奕言,博士研究生,主要研究领域为数据库参数调优、机器学习。E-mail:liyiyang@ruc.edu.cn。田季坤,硕士研究生,主要研究领域为参数调优、强化学习。蒲 照(共同第二作者),硕士研究生,主要研究领域为数据库自调优、贝叶斯优化。李翠平(通信作者),博士,教授,主要研究领域为社交网络分析、社会推荐、大数据分析 & 挖掘。E-mail:licuiping@ruc.edu.cn。陈 红,博士,教授,主要研究领域为数据库技术、新硬件平台下的高性能计算。

comprehensive review and synthesis of the current methodologies employed in the intelligent tuning of database knobs. The discourse is structured around three primary focal points: the categorization of existing intelligent knob tuning methods, the examination of prevalent tools within the industry, and an exploration of the future challenges and research directions in this domain. The review begins with a systematic categorization of the existing methods of intelligent knob tuning into five distinct groups, each defined by the underlying machine learning algorithms they employ, including bayesian optimization, reinforcement learning, deep learning, searching based and rule based methods. For each category, a detailed exposition is provided, encompassing the principles, techniques, advantages, and limitations inherent to each method. This categorization not only elucidates the current landscape of intelligent knob tuning but also facilitates a deeper understanding of its theoretical and practical aspects. Furthermore, the paper delves into the practical application of intelligent knob tuning within the industry. It presents an overview of the mainstream database knob tuning tools currently in use, coupled with an analytical discussion of the practical challenges and pitfalls encountered in the deployment of these technologies. This segment aims to bridge the gap between theory and practice, shedding light on the real-world implications of intelligent knob tuning. Finally, the paper looks ahead to the future, identifying and discussing potential research directions in the field of intelligent knob tuning for databases. This prospective analysis is aimed at equipping researchers with a clear understanding of the current challenges, thereby inspiring and guiding future research endeavors in this dynamic and evolving field. In summary, this paper aspires to serve as a valuable resource for researchers and practitioners alike, offering a comprehensive overview of the state-of-the-art in intelligent database knob tuning. Through its detailed examination of existing methods, practical applications, and future research directions, it seeks to contribute to the ongoing advancement of research in this important area of database optimization and management.

Keywords machine learning; knob tuning; bayesian optimization; reinforcement learning; intelligent databases

1 引 言

数据库管理系统(DataBase Management System, DBMS)在现代社会中发挥着非常重要的作用. 它能安全高效地存储和管理大量数据, 促进数据共享与协作, 为决策制定提供实时洞察, 支持应用程序的高效运行, 是政府和企业不可或缺的关键基础设施. 现代数据库管理系统往往有数百个参数, 这些参数控制了系统的内存分配、I/O 优化、备份与恢复等诸多方面, 并极大地影响了数据库的吞吐量、延迟等性能. 比如 MySQL 数据库有大约 260 个参数, 调整其中 InnoDB 缓冲池大小和临时表大小可以使数据库高效地处理查询. 类似这种通过调整数据库参数配置来优化性能的一类技术被称作数据库参数配置调优技术. 该技术旨在提升数据库的吞吐量(Throughput), 或降低查询执行延迟(Latency)等,

在数据库系统中发挥着重要作用.

传统数据库参数配置调优方法是由经验丰富的数据库管理员(DataBase Administrator, DBA)借助历史经验和反复测试通过人工进行调整. 与此同时, 一些数据库厂商会提供辅助 DBA 进行调优的工具, 例如 PG Tune^[1]可以根据 PostgreSQL 数据库所处的不同硬件环境来推荐默认参数配置. 但这种依赖于人工的方法面临着许多困难, 主要体现在以下几方面: (1) 参数数量众多, 往往数以百计, 作用各不相同, 而人工调整时往往只关注其中少数几个参数; (2) 参数间不独立^[2], 更改一项参数可能会影响其他参数, 人工难以捕捉参数之间的联系; (3) 配置不通用, 不同工作负载以及不同应用程序下的最优参数配置并不相同, 人工经验往往并不可靠. 因而随着数据库系统和应用程序规模的扩大, 仅凭 DBA 和这些简单的辅助工具进行调优难以达到预期效果, 还会带来极高的人力和时间成本. 这导致当前

数据库参数配置调优应用并不广泛,绝大多数数据库仍使用默认配置运行,难以发挥出最佳性能。

随着人工智能和机器学习的不断发展和广泛应用,人们将它们与数据库系统结合,通过机器学习技术替换或者增强传统数据库的组件来提升数据库系统的性能,提出了AI4DB这一新的研究方向^[2]。作为数据库系统的重要组件,学术界和工业界结合机器学习、深度学习等方法提出了数据库参数配置智能调优,简称为参数调优。该类方法利用工作负载和历史数据来训练模型,根据当前负载自动推荐最优参数,能有效减少人力消耗,提升数据

库系统性能。

近年来该领域关注度不断上升,相关文献数量显著增加,各大数据库厂商也相继推出各自的调优工具。如图1所示是近些年关于数据库参数配置智能调优的相关工作时间轴展示。但目前研究仍存在如下问题:(1)现有工作多为对具体调优算法及架构的改进,缺少综述性质的文章,研究人员难以快速掌握当前最新的研究进展;(2)学术界论文与工业界实际需求之间存在一定差异,实际落地困难;(3)部分工作研究动机不够明确,没有针对亟待解决的问题开展研究。

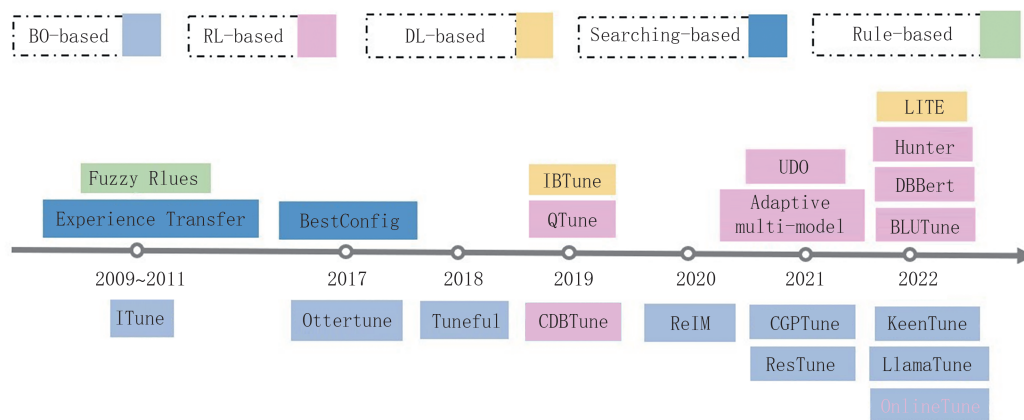


图1 数据库参数配置智能调优方法发展时间轴

基于此,本文对相关工作进行综述,主要包括:

(1)从算法角度详细介绍了当前研究现状。调研了近年来相关工作,依据调优算法将现有研究分为五类:基于贝叶斯优化的、基于强化学习的、基于深度学习的、基于启发式搜索的和基于规则的参数调优算法。本文针对每一种方法分别展开详细介绍,并对比了各自优缺点;

(2)介绍了参数配置智能调优在工业界应用情况。逐一分析了主流数据库厂商相应的参数调优工具,对比其与学术界研究之间存在的差异,并分析原因;

(3)依据对现状的分析,总结出参数配置智能调优当前面临的主要问题及未来发展方向,并提出了可能的解决方案。

确立了研究目标后,通过关键词检索谷歌学术等学术搜索引擎、查找近五年相关领域期刊会议发表论文、查看已有论文的参考文献及相关工作等方法,查阅参数调优相关资料,按照引用量和相关性进行收集和筛选。大量阅读SIGMOD、VLDB、计算机学

报等国内外主流期刊会议论文(具体标准参考了《中国计算机学会推荐国际学术会议和期刊目录》),并运用归纳和演绎、分析与综合以及抽象与概括等方法进行分类整理,紧跟本领域的研究进展,围绕研究主题进行深入思考,提出论点,为本研究实施奠定坚实的基础。

与已有工作相比,文献[3]介绍了主流数据库参数配置调优技术,但是它的侧重点是现有开源方法的性能对比,仅关注了基于强化学习的和基于贝叶斯优化的经典方法,缺乏对其他调优方法及工业界调优工具的关注,且未涵盖许多最新工作。本文以调优算法(模型)类型以及优化策略来分类和总结当前参数配置智能调优技术,包括工业界主流调优工具,相较于上述文献更加综合全面。文献[2]和文献[4]作为智能数据库的综述文章,在一些章节中也简要介绍了参数配置智能调优技术。但仅作为整个智能数据库中的一部分出现,并未详细介绍具体细节。文献[5]和文献[6]分别总结了现有黑盒优化问题算法及超参数优化问题算法,但这些工作关注点

在于更具一般性的算法上,并没有考虑和数据库的结合.综上所述,本文首次对数据库参数配置智能调优方法进行综述,以便为后续该领域的研究人员提供帮助和引导.

文章后续结构如下:第二节和第三节中分别介绍了基于贝叶斯优化的和基于强化学习的参数调优方法.包括基本原理、架构、调优流程和各自优缺点,并从参数空间缩减、模型自身和知识迁移三个改进方向对现有工作及彼此之间的联系展开详细介绍.在第四节中介绍了其余三种调优方法,基于规则的、基于启发式搜索的和基于深度学习的.它们由于自身种种限制,相关工作相对较少,但仍是数据库参数配置智能调优领域重要组成部分,故合为一章介绍.第五节则总结了工业界主流的调优工具,与学术界诸多方法进行了对比.最后,第六节依据当前面临的主要问题,对数据库参数配置智能调优未来研究方向进行了展望.

2 基于贝叶斯的参数调优技术

考虑一个具有参数 $\theta_1, \dots, \theta_m$ 的数据库系统,其参数配置空间为 $\Theta = \Theta_1 \times \dots \times \Theta_m$,其中 $\Theta_1, \dots, \Theta_m$ 代表其各自的值域.参数 θ 可以是连续的,也可以是离散的.将数据库性能指标表示为 f ,它可以是任何要优化的指标,例如吞吐量、延迟等.假设目标是最大化问题,数据库配置调优的目标就是找到一个配置 θ_* 使得

$$\theta_* = \operatorname{argmax}_f(\theta), \theta \in \Theta$$

因此参数优化问题可看作是一类配置空间上的黑盒优化问题.现有的基于贝叶斯优化(Bayesian Optimization, BO)的系统采用BO模型来建模配置与数据库性能之间的关系.它们遵循BO框架来搜索最优的DBMS配置:(1)拟合概率代理模型;(2)通过最大化采集函数来选择下一个配置进行评估.

2.1 参数调优中的贝叶斯优化

贝叶斯优化是超参数调优领域最先进的优化框架之一,它使用贝叶斯定理来指导搜索以找到目标函数的最小值或最大值^[7].在每次迭代时,贝叶斯优化利用之前观测到的历史信息(先验知识)来进行下一次优化.其主要由三个部分组成:贝叶斯优化框架、代理模型(Surrogate Function)以及采集函数(Acquisition Function).其中代理模型用于模拟未知的目标函数 f ,采集函数用于选择下一个用于观测的

点 x .

在参数调优领域应用贝叶斯优化时,将数据库的参数配置作为输入 x ,外部指标如吞吐量(Throughput)、延迟(Latency)等作为输出 y .贝叶斯优化参数调优的一般流程为:(1)负载特征化;(2)数据规范化;(3)选择调优参数(Knob);(4)训练代理模型并生成推荐参数配置;(5)运行配置获得观测值.

重复(4)与(5)直到调优过程结束,整体流程如图2所示.其中(1)、(2)、(3)为前期准备工作,(4)、(5)为调优过程.

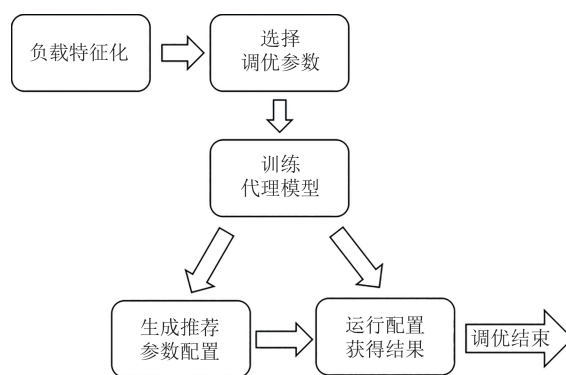


图2 基于贝叶斯优化的参数调优方法基本框架

贝叶斯优化在参数调优领域具有以下优势^[3]: (1)在优化过程中贝叶斯优化考虑到了对已有知识的利用和对未知区域的探索,因此可以在有限的时间里获得一个较好的结果,提高优化效率;(2)可以使用历史数据初始化贝叶斯优化模型,进一步加速调优.

以典型的贝叶斯调优工具OtterTune^[8]为例,首先通过执行查询一组或一个特定的负载,获得该数据库指标的初始观测值,接着与历史数据中的负载进行比较,找到与当前待调优负载最相似的负载.然后使用该负载的相关数据,即参数配置和与之对应的数据库表现,来初始化代理函数.由于用来初始化的负载并不完全等价于当前负载,因此并不能完全“信任”此时生成的模型,作者通过向核函数和被选择的参数配置中添加脊项(Ridge Term)来使虚拟环境噪声化.在选择下一个运行的点时,通过选择在当前表现最优点附近的点或者方差较大的点,来实现利用和探索的权衡.经过一段时间的训练后,未知区域越来越少,因此接下来探索表现较好点周围的区域.在参数空间内随机选择合法点,使其数量与表现较好的点为10:1,对这两部分点使用梯

度下降的方法寻找更优点,直到收敛或达到执行的上限,从中选择最优点作为下一次运行的点.经过上述设计后,调优程序便可以开始自动调优,完成选点-运行-更新-选点的迭代过程.

2.2 相关参数调优算法及改进

宋等人^[9]在2009年第一次将贝叶斯优化引入到参数调优领域,提出了名为iTuned的参数调优器.2017年,Aken等人^[8]将负载匹配与贝叶斯方法结合,使用历史经验来加速调优过程,设计了OtterTune.在此基础上,研究人员又做了诸多改进,近年来各类基于贝叶斯方法的参数调优器层出不穷.图3中展示了基于贝叶斯优化的参数调优主要技术路线.

根据参数调优的流程,本文将整个调优技术划分为三部分:特征空间缩减、BO模型及知识迁移.特征空间缩减模块通过识别重要的参数、降维等方

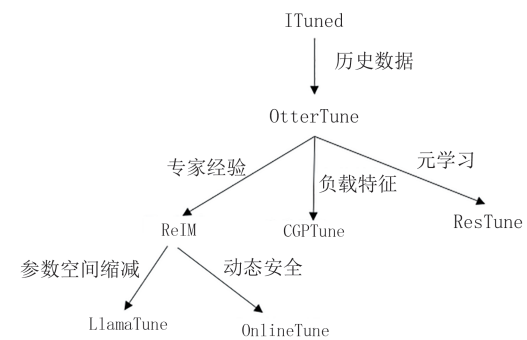


图3 基于贝叶斯优化的参数调优主要技术路线图

法剪枝优化配置空间.修剪后的配置空间被传递给调优器,调优器在每次迭代中在修剪的空间上使用BO模型推荐一个理论较优的配置.此外,知识迁移模块通过借用数据存储库中存储的类似任务加快当前调优任务.表1展示了主要的基于贝叶斯优化的参数调优方法及其所作的改进,下面将依次介绍.

表1 基于贝叶斯优化的参数调优方法

调优方法	缩减特征空间	BO模型	知识迁移	特点
ITuned ^[9]	Sensitivity Analysis(SA)	VanillaBO	无	首次采用BO方法进行参数调优
OtterTune ^[8]	Lasso	VanillaBO	workload mapping	加入历史数据,重用历史观察结果
Tuneful ^[25]	Gini score	BO+costmodel	workload mapping	提出了一种调优成本摊销模型;可直接支持Spark
ResTune ^[11]	Expert experience	SingleTarget ConstraintBO	RGPE	使用预训练的调优模型提高迁移能力
ReIM ^[12]	Expert experience	GuidedBO	workload mapping	只关注内存分配;利用专家经验优化调优
CGPTuner ^[13]	无	Adopting Contextual BO	workload mapping	适应工作负载变化
OnlineTune ^[14]	Safety Assessment Expert experience	Composite Kernel BO	Environment mapping	在线动态调优,关注调优的安全性
LlamaTune ^[15]	High-dimensional to low-dimensional projection,Expert experience	SMAC/Vanilla BO	无	借助专家知识及降维操作,提高采样效率
KeenTune ^[16]	XSen Expert experience	BGCS	workload mapping	集成SHAP方法用于参数选择

2.2.1 特征空间缩减

特征空间缩减模块通过识别重要的参数、降维等方法剪枝优化配置空间,以缩小搜索范围,提高参数调优效率.常用的技术包括重要参数选择、参数取值范围限制以及参数空间映射等.

(1)重要参数选择

尽管数据库系统有数百个参数,但并非所有参数都对数据库性能有显著影响.选择重要的参数可以精简配置空间,进一步加速配置优化.最早调优的参数是由经验丰富的DBA来进行指定的,这样做的好处在于能够快速选出对数据库性能影响重大的重要参数,但一方面,面对不同的负载,同一参数的重要性会有所变化,固定的调优参数集会影响调优

器的拓展性;另一方面,这种方法非常依赖人工经验,自动化水平低.为了进行合理的参数选择,首先需要收集一组不同配置下的观测值,根据观察结果,采用一种算法对参数的重要性进行排序(也称作“重要性测量”).最后,根据重要性排序选择top-k参数来确定构型空间.重要性度量有多种选择,不同度量方式会导致不同的配置空间进而影响后续优化.它们可以分为两类:基于方差的和基于可调性的^[17].基于方差的方法选择对数据库性能影响大的参数;基于可调性的方法^[18]量化参数的可调性,测量从参数的默认值调整参数所能获得的性能增益.

OtterTune使用Lasso^[19]识别重要参数.Lasso是一种基于线性回归系数来评估参数重要性的方法,

是典型的基于方差的方法. 它对系数施加 L1 范数惩罚, 从而使冗余参数的系数置 0. L1 范数惩罚使得 Lasso 在训练样本中有许多不相关的参数时有效. 然而, Lasso 假设参数空间是线性的, 不能捕捉从参数到性能度量的非线性依赖关系.

Tuneful 使用基尼指数^[10]来进行参数选择, 来源于基于树的模型 (如随机森林模型), 同样属于基于方差的方法. 每个参数的基尼系数被定义为给定参数在所有树的树分割中的使用次数, 因为重要参数区分更大数量的样本, 并在树分割中使用更频繁.

基于方差的方法关注的是参数的全局效应, 而基于可调性的方法关注的是空间中比默认配置更好的“好”区域. 它可以直接用于确定从给定的默认值调整参数的必要性. KeenTune^[16]使用 SHAP^[20] (SHapley Additive exPlanations), 它是一个统一的框架, 用于解释合作博弈论中由经典 SHapley 值估计导出的性能变化. 性能变化在参数之间进行相加分解. SHAP 计算每个参数的贡献 (即 SHAP 值), 以将默认性能调整到目标性能. 给定一组观察值, 每个参数的可调性通过平均正 SHAP 值来计算.

(2) 参数取值范围限制

数据库包含大量的参数, 而每个参数的取值范围又有着显著的区别. 一些参数仅有数个可能的值, 而另一些则以百万为单位. 对于许多具有较大值范围的参数, 小的更改不太可能显著影响 DBMS 的性能, 而这种细粒度控制会大大增加配置搜索空间的大小. 基于这一观察, LlamaTune^[15]尝试了以固定间隔桶化参数值空间. 它们通过限制参数取值个数来实现这一点, 任何参数最多只能有 K 个不同的取值. 拥有多于 K 个取值的参数将其取值范围离散化到 K 个, 这些新的离散值统一分布在整个范围内 (例如对于参数 `commit_delay`, 它控制已提交事务刷新到磁盘之前的延迟, 通常取值范围是 $[0, 100\ 000]$, 如果 $K=100$, 新的离散值集将是 $(0, 1000, 2000, \dots, 100\ 000)$).

LlamaTune 同时注意到, 虽然数值参数的值具有自然的顺序, BO 方法在进行局部搜索时可以积极地利用这一顺序 (当调优器确定了一个性能良好的配置时, 它通常会选择探索附近的区域, 以改进当前的最佳), 但一些数值参数有特殊的取值 (如 $-1, 0$ 等), 这不可避免地打破了自然顺序. 参数被设置为特殊值时, 其功能往往与正常值时区别较大 (例如禁用某些功能). 以 PostgreSQL 为例, 参数 `backend_flush_after` (取值范围 $[0, 256]$) 通常表示先前执行的

写操作被刷新到磁盘的页数, 而当该值为 0 时, 表示强制回写被禁用. 通常这个参数的值越大, 就意味着在发出回写请求之前, 允许内核页缓存中保留更多字节. 但是如果值为“0”, 则会完全禁用回写机制, 从而导致该工作负载的吞吐量大大提高. 特殊值使得 DBMS 性能的建模更加困难, 因为它们使得目标函数输出相对于输入不连续. 而如果调优器调整这些参数时缺乏关于特殊值含义的知识, 那么在调整过程中就不太可能选择这个值. 为此, LlamaTune 设计了一种新的基于偏置抽样的方法, 这使得调优器在随机初始化阶段以高概率意识到具有特殊情况的行为, 并最大限度地减少找到良好配置所需的样本数量.

安全性是另一个影响参数取值的重要限制, 有些不适用的参数值会显著损害数据库性能, 甚至导致数据库宕机. 为解决此问题, OnlineTune^[14]通过黑白盒方法评估候选集的安全性, 限制参数空间大小. 黑盒方法使用高斯过程生成两个指标:

$$u_n(\theta, c) = \mu_n(\theta, c) + \beta \sigma_n(\theta, c)$$

$$l_n(\theta, c) = \mu_n(\theta, c) - \beta \sigma_n(\theta, c)$$

其中 u_n 与 l_n 为置信区间的上下界, β 控制置信界限的紧密性, 若 $l_n > \tau$ 则认为候选集安全. 白盒方法采用 MysqlTuner (一种基于规则的 Mysql 调优器, 详见 5.2 节) 检查 DBMS 的指标, 并使用静态启发式来建议设置配置的范围, 根据建议从安全集中删除不满足的配置. 生成安全集后, 使用上置信界 (Upper Confidence Bound) 对安全集进行采样从而生成推荐配置.

(3) 参数空间投影

调优器需要大量迭代的主要原因是它们面对的参数搜索空间通常是非常高维的. 这些搜索空间的大小通常与用户希望调优的参数的数量成正比, 主流数据库 (如 Postgres 或 MySQL) 往往拥有数百个参数^[8]. LlamaTune 利用随机低维投影方法, 执行从原始参数空间维度 (D) 到低维子空间 (d) 的参数空间的投影, 然后使用调优器来调优这个更小的子空间. 即: 为用户提供的 D 维参数空间 X_D , 设计一个 d 维空间 X_d , 其中 $d \ll D$, 且至少存在一个点 $p \in X_d$, 其性能接近最优点 $p^* \in X_D$. 研究证明, 如果低维空间的维度大于目标函数的有效维数, 那么仅通过调优这个较小的空间就有可能实现接近最优的性能^[21-22]. 正如在前文中所讨论的, 一小部分重要的参数是 DBMS 性能改进的主要原因, 因此这种方法是

可行的。

LlamaTune 使用合成参数来生成低维空间 X_d ，这些合成参数本身没有任何物理意义，但它们的值决定了一个(或多个)DBMS 参数的值。换句话说，LlamaTune 定义了从合成参数值(即近似空间 X_d)到物理 DBMS 参数(即原始输入空间 X_D)的映射。这类使用合成的较小空间隐式优化高维输入搜索空间的思想，在 BO 领域中较为流行^[22-23]。LlamaTune 选取了其中两种主流方法，REMBO^[22]和 HeSBO^[24]，应用在参数调优器中。LlamaTune 通过消融实验证明，在参数调优任务上，HeSBO 取得了更优的效果。

2.2.2 BO 模型

在 iTuned、OtterTune 等早期调优器中，研究人员采用 Vanilla BO(即标准贝叶斯优化算法)作为调优器的 BO 模型。该调优器使用高斯过程(Gaussian Processes)作为代理模型的基于 BO 的调优器，期望改进(Expected Improvement, EI)作为采集函数。

在 Vanilla GP N ($\mu(\theta)$, $\sigma^2(\theta)$)中

$$\mu(\theta, c) = k^T (K + \sigma^2 I)^{-1} y$$

$$\sigma^2(\theta) = k(\theta, \theta) - k^T (K + \sigma^2 I)^{-1} k$$

其中 $k = [k(\theta_1, \theta), \dots, k(\theta_n, \theta)]^T$, K 为协方差矩阵, $K_{i,j} = k(\theta_i, \theta_j)$ 。内核 $k(\theta_n, \theta')$ 对目标函数的整体平滑度进行建模。对于 Vanilla BO, 核函数(如 OtterTune 中的 RBF 核)是基于两个构型之间的欧氏距离计算的(假设构型空间的自然有序性和连续性)。在此基础上, Tuneful^[25]结合马尔可夫超参数边际化算法, 设计了新的 EI 函数作为采集方程, 对 Vanilla BO 模型进行了优化。

在实际使用 OtterTune 的过程中, 研究人员发现历史数据可能是难以复用的, 因为即使负载可能是相近的, 但支撑数据库的硬件不同也可能导致不同的调优结果。考虑到这种情况, CGPTune^[18]选择不使用历史数据, 而仅仅使用在调优过程中产生的上下文(Contextual)数据来进行学习和推荐。面对冷启动的问题, 作者随机生成一组参数配置进行运行并作为初始化高斯过程的数据。在采集函数的选择上, 作者使用了多臂老虎机策略(Multi-armed Bandit Strategy), 即每次迭代采用多种采集函数, 并对采集函数的效果进行评估, 最终选出一个最好的采集函数。在核函数的选择上, 作者选用了被广泛使用的马丁核(Matérn kernel), 为了能够更好地利用上下文知识, 作者定义了一种基于参数配置-负载

对的新型核函数:

$$k((\vec{x}, \vec{w}), (\vec{x}', \vec{w}')) = k(\vec{x}, \vec{w}) \times k(\vec{x}', \vec{w}')$$

这使得当两个参数配置-负载对中参数配置和负载都相似时才能传递知识。

OnlineTune 更进一步地将马丁核与线性核结合, 设计了一种基于混合核的 BO 模型, 使调优器更加适应负载环境的动态变化。具体来说, OnlineTune 构造了混合核函数 $k((\theta, c), (\theta', c')) = k_\theta(\theta, \theta') + k_c(c, c')$, 其中 $k_c(c, c')$ 为线性核, 用于表示环境间的依赖性; $k_\theta(\theta, \theta')$ 为马丁核, 表示参数配置的非线性性能。复合内核意味着当参数配置和上下文相似时, 函数值是相关的, 实现了知识的转移。

RelM^[12]则将专家经验融合进了 BO 模型当中, 利用白盒经验来加速贝叶斯搜索过程。论文参照了 SBO(Structured Bayesian Optimization)的思路, 提出了 GBO(Guided Bayesian Optimization), 利用 RelM 白盒模型来优化基于内存的分析应用程序。论文首先通过深入理解应用内部的内存管理上的算法, 理解内存配置的影响, 据此提出了 RelM 白盒模型。RelM 接收测试应用程序的配置和一组先前执行的统计数据, 输出一组用于优化的初始的配置数据和一组派生指标 q^x 。派生指标是使用简单的分析模型导出的, 目的是从更高维的区域中分离出最合适的配置空间区域。具体来说, 其包含三个派生指标 q_1^x 、 q_2^x 和 q_3^x , 分别表示系统当前的安全性、磁盘开销和 GC(Garbage Collection, 垃圾回收)。

$$q_1^x = \frac{M_i + \min(m_c^x, m_c) + p^x * (M_u + \min(m_s^x, m_s))}{m_h^x}$$

$$q_2^x = \frac{M_i + m_c}{\min(m_0^x, m_c^x)}, q_3^x = \frac{p^x * \min(m_s^x, m_s)}{0.5 * m_e^x}$$

$$q^x = \{q_1^x, q_2^x, q_3^x\}$$

相应的, 代理模型也需要相应更改以适应模型 Q 的度量(即 $GP(x_{1:n}, q_{1:n}, y_{1:n})$), 代理模型中的下一个采样点通过 EI 分数(Expected Improvement score)确定, 即

$$X_{n+1} = \arg \max_{x \in X} EI(x, q^x; x_{1:n}, q_{1:n}, y_{1:n})$$

一般参数调优的目的都是为了提高数据库的外部指标如吞吐量、延迟等, 但 ResTune^[11]的作者认为在实际应用场景中, 过高的性能可能存在溢出的情况, 因此将注意力集中于在保证一定数据库性能的前提下, 尽可能减少数据库的资源开销。同时为了加快调优速度, 作者引入了元学习的方法来加速调优。作者在常用采集函数 EI 方法的基

础上,考虑到资源受限的情况,提出了CEI方法(Constrained Expected Improvement),即采集函数从基础的 $\alpha_{EI}(\theta) = \mathbb{E}[max(0, f_{res}(\theta_{best}) - f_{res}(\theta))]$ 变更为 $\alpha_{CEI} = Pr[\tilde{f}_{tps}(\theta) \geq \lambda_{tps}] \cdot Pr[\tilde{f}_{lat}(\theta) \leq \lambda_{lat}] \cdot \alpha_{EI}(\theta)$ 其中 $f_{res}(\theta)$ 为当参数配置为 θ 时,数据库的资源使用量, θ_{best} 为当前时刻前使用资源最少的 θ , $\tilde{f}_{tps}(\theta)$ 与 $\tilde{f}_{lat}(\theta)$ 为经高斯过程所估算出的吞吐量与延迟, λ_{tps} 与 λ_{lat} 为吞吐量性能的下界与延迟性能的上界. 使用上述采集函数可以实现选择满足限制条件且最少使用资源的参数配置 θ .

对于代理模型,作者选择了通过元学习的方式构建高斯过程. 作者给已经存放在资源仓库中的 N 个历史负载和目标负载分别建立一个高斯过程模型作为基础学习器,在这些基础学习器的基础上形成了一个元学习器 L_M , $L_M(\theta) \sim N(\mu_M(\theta), \sigma_M^2(\theta))$, 其中

$$\mu_M(\theta) = \frac{\sum_{i=1}^{N+1} g_i \mu_i(\theta)}{\sum_{i=1}^{N+1} g_i}$$

$$\sigma_M^2(\theta) = \sigma_{N+1}^2(\theta)$$

g_i 为每个基础学习器的权重. 考虑到目标负载的观察结果更可信,基于目标负载的基础学习器所占权重最大. 其余基础学习器的权重由静态和动态的方式赋予,静态方法为通过 Epanechnikov quadratic kernel 方法评估历史负载与目标负载的相似性并赋予基础学习器权重;动态方法将多组参数代入,如果一个基本学习器能够根据它们的性能指标将其正确排列,则认为该学习器对元学习器有用.

除了上述基于高斯过程的BO模型外,还有一类BO模型采用了基于随机森林的代理,如LlamaTune采用的SMAC(Sequential Model-based Algorithm Configuration). 该代理对于高维和分类输入表现良好. SMAC假设一个高斯模型 $N(y|\mu, \sigma^2)$, 其中 μ 和 σ^2 是随机森林的均值和方差. SMAC支持所有类型的变量,包括连续、离散和分类特征.

2.2.3 知识迁移

知识迁移模块通过利用历史调优经验来加速目标调优任务,能极大地提高调优效率. 在参数调优领域,知识迁移通常是基于负载的,负载刻画的对迁移效果有着重要影响. 本节首先介绍现有负载特征化方法,之后介绍两类常用的知识迁移框架,包括工作负载映射(Workload Mapping)和加权高斯过程集成(Ranking-weighted Gaussian Process

Ensemble, RGPE).

(1) 负载特征化

负载特征化是负载映射中的关键问题,其效果直接关系到调优器能否准确在历史数据中找到最相似的负载.

OtterTune 在处理数据库的负载特征化时,使用DBMS的内部指标来描述负载的表现,其内置一个控制器可以执行一些适当的操作并收集运行时所有可用的数字指标统计信息,然后生成键值对储存在数据仓库中. 首先使用因子分析(factor analysis)的方法对已获得的指标统计信息进行降维处理. 具体方法为使用一个矩阵 X 作为输入,其行为各指标,列为尝试过的参数配置, X_{ij} 表示指标 i 在参数配置 j 上的表现,通过FA方法输出一个矩阵 U ,其行为各指标,列为生成的因子,每个因子都是各指标的线性组合, U_{ij} 则表示指标 i 在因子 j 中的系数,如果两个指标在 U 中的系数相近,说明这个系数关联性很强,可以通过删除关联性很强的参数降低参数的冗余度. 其次以指标在 U 中的行值为坐标,使用k-means方法对指标进行聚类,聚簇数目 k 由启发式方法给出,并由居于每簇最中心的指标给这个簇命名,因此每一簇都反映数据库一个方面的特点.

ResTune将负载特征化主要分为三步:因为不同的查询只表现出少量的模式,所以可以过滤掉特定的变量而保留一些关键词(如Select、Update等),并采用TF-IDF(Term Frequency-Inverse Document Frequency)方法^[26]为每个查询计算特征向量;然后根据特征向量对查询使用随机森林的方法进行分类,并返回预测的概率分布;最后计算负载中所有查询的概率分布的均值,作为负载的特征值. 简言之,即通过查询的出现频率来表示负载的特征.

OnlineTune主要考虑的是面对动态的负载,因此通过查询到达速率与查询组成来描述负载. 查询到达率可以被一维编码;对于查询组成,使用标准LSTM编码-解码网络来提取查询的信息编码,并跨工作负载进行泛化,编码器网络上的最终隐藏状态为查询提供了一个密集的编码,并对查询编码进行平均,获得一个负载的查询组合特性.

(2) 工作负载映射

工作负载映射最早由OtterTune提出. 它根据数据库指标的绝对距离将目标工作负载与最相似的历史工作负载匹配,并重用来自类似工作负载的历史观察结果. 具体来说,DBA通过执行一组查询或一个特定的负载追踪,获得该数据库指标的初始观

测值,接着与历史数据中的负载进行比较,找到与当前待调优负载最相似的负载.然后使用该负载的相关数据,即参数配置和与之对应的数据库表现,来初始化代理函数.这种策略可以与任意基于BO的调优器采用,事实上,几乎所有后续的基于BO的调优器都继承了这一思想. OnlineTune 在此基础上使用DBSCAN方法进行聚类,并用SVM方法为其学习一个非线性的决策边界,最后为每一类建立一个高斯过程,使用相互信息评分(MI)来量化差异,当低于阈值时重新划分边界并更新模型.在调优一个负载时先特征化其环境因素,然后匹配最相近的负载类,利用基于该负载类的高斯过程进行调优,从而满足了动态性.

(3)加权高斯过程集成

加权高斯过程集成(Ranking-weighted Gaussian Process Ensemble, RGPE)^[27]是另一种基于BO调优器的集成模型. RGPE通过可区分的权重组合了相似历史任务的基本GP模型. 权重是使用相对排名损失分配的,以便在不同的工作负载和不同的硬件环境中进行泛化. 集成方式避免了用来自类似任务的所有观测来拟合单个高斯过程模型所带来的糟糕缩放. ResTune 采用RGPE加速目标任务的调优过程.

2.3 小结

基于贝叶斯优化的参数调优技术将参数调优视作特征空间上的黑盒优化问题,使用代理模型模拟参数-性能的映射关系,借此使用采集函数搜索最优配置(点). 基于贝叶斯优化的参数调优技术具有调优速度快、优化性能较好等优点. 然而它非常依赖训练数据,需要采集大规模、高质量的训练样本用于调优模型的训练. 此外,它往往难以精确地拟合变化曲面,而且在曲面上寻找最优解仍然是NP-Hard问题^[28].

作为参数调优的两大主流方法之一,近年来相关工作层出不穷. 研究人员的关注重点主要在缩减特征空间、BO模型和知识迁移三方面. 其中(1)缩减特征空间可以提高调优效率、保证调优过程相对平稳. 但参数空间与负载、数据库和硬件环境息息相关,现有方法大多难以适应复杂多变的调优环境. (2)对BO模型的改进主要集中在采集函数和代理模型两个方面,其目的主要出于对调优效率的考虑. 现有方法大多还是在基础BO模型上做调整,没有考虑尝试贝叶斯神经网络等新的模型,同时也缺

少对数据库调优这一特定任务的特殊处理. (3)知识迁移同样是为了加速调优,但几乎所有方法只提出了知识迁移的技术框架,对于历史数据积累、负载刻画准确性衡量等问题考虑甚少.

总体来看,调优效率、调优过程稳定性和调优方法泛用性是基于贝叶斯优化的参数调优技术主要面临的问题,未来在这些方面仍有很大的研究空间.

3 基于强化学习的参数调优技术

基于贝叶斯优化的调优方法通常以流水线式的结构将各种算法模型串联起来,机器学习过程的每个阶段都构成了整个机器学习流水线中的一个独特模块. 尽管这种方法有助于提高模型的准确性和构建效率,但存在一个严重的问题,即前一阶段的最优解不能保证后一阶段的最优解,并且模型的不同阶段可能无法很好地相互配合,因此难以实现端到端的方式优化总体性能.

因此一些研究人员尝试使用强化学习方法构建端到端的数据库调优系统. 强化学习的试错策略能够仅使用少量样本来调整参数以实现较高的性能,减轻了建模初始阶段对收集大量训练样本的负担,更符合DBA在实际场景中的判断和调整动作;使用奖励函数提供用于评估数据库性能的反馈,并基于反馈机制提出端到端的学习模型;强化学习中的深度确定性策略梯度方法在高维连续空间中寻找最优配置,一定程度上解决了现有回归方法中的量化损失问题.

图4中展示了基于强化学习的参数调优主要技术路线,表2具体列出了这些方法,同贝叶斯优化方法类似,将对强化学习的方法的改进划分为参数空间缩减、模型自身和知识迁移三部分.

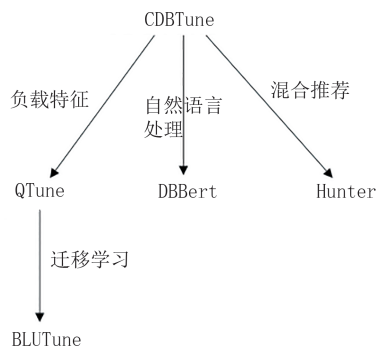


图4 基于强化学习的参数调优主要技术路线图

表2 基于强化学习的参数调优方法

调优方法	缩减特征空间	RL 模型	知识迁移	特点
CDBTune ^[29]	DBA	DDPG	Fine tune	首次应用强化学习在参数调优领域
QTune ^[30]	DBA	DS-DDPG	Fine tune	将 query 信息引入到调优流程中
BLUTune ^[31]	DBA	advantage DS-DDPG	Transfer learning	加入了两阶段迁移学习
Adaptive Multi-Model ^[32]	DBA	DDPG	Workloadmapping	与BO负载映射思路类似,训练了多个模型
DBBert ^[33]	Hints from manual	Double Deep Q-Networks	Fine tune	利用NLP模型,从用户手册等文本中提取提示用于调优
Hunter ^[34]	PCA & RF	Two-stage DDPG	Fine tune	将GA算法与强化学习结合,两阶段训练

3.1 参数调优中的强化学习

强化学习(Reinforcement Learning, RL),又称再励学习、评价学习或增强学习,是机器学习的范式和方法论之一,用于描述和解决智能体(agent)在与环境的交互过程中通过学习策略以达成回报最大化或实现特定目标的问题.反复实验(trial and error)和延迟奖励(delayed reward)是强化学习最重要的两个特征.

强化学习系统一般包括六个要素:状态(State),环境(Environment),策略(Policy),奖励(Reward),动作(Action)以及智能体(Agent).在参数调优任务中,六个关键元素与数据库配置调优之间的对应关系以及各部分的交互如图5所示.具体来说:

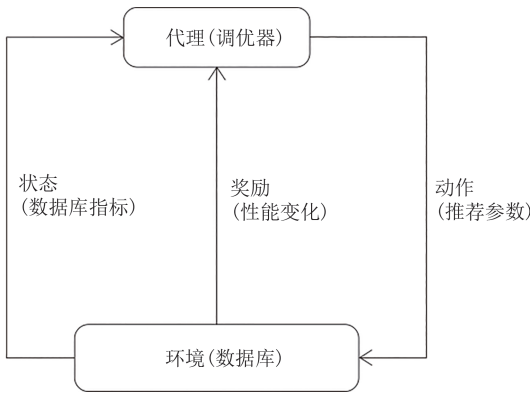


图5 强化学习组成元素与配置调优的关系

- (1)状态(State):状态即对当前时刻的环境的描述,通常以数据库的内部指标表示当前状态.
- (2)动作(Action):智能体根据当前时刻环境的状态所做出的行为,即根据数据库当前的状态所推荐的参数配置.
- (3)环境(Environment):环境是与智能体相互作用的部分,在这里为调优的目标,即数据库系统.
- (4)智能体(Agent):智能体通过与环境进行交互,接收环境的状态,根据一定的策略做出相应的动

作以改变环境的状态,并根据环境状态改变带来的奖励更新策略.在这里即为调优器,它接收当前时刻数据库的各项内部指标按照一定的策略给出推荐的配置,同时根据数据库的性能变化更新调优策略以推荐出更优的配置.

(5)奖励(Reward):奖励信号定义了强化学习问题的目标,在每个时间步骤内,环境向智能体发出的标量值即为奖励,它能定义智能体表现好坏.在这里主要是根据推荐配置所带来的性能表现提升,反映所推荐的配置对数据库影响的好坏程度.

(6)策略(Policy):策略定义了智能体对于给定状态所做出的行为,换句话说,就是一个从状态到行为的映射.策略是强化学习系统的核心,因为完全可以通过策略来确定每个状态下的行为.在这里策略指的是训练的深度学习网络.

强化学习训练时,首先从当前的状态 S_t 出发,在做出一个行为 A_t 之后,对环境产生了一些影响,它首先给agent反馈了一个奖励信号 R_t ,接下来agent可以从中发现一些信息,此处用 O_t 表示,进而进入一个新的状态,再做出新的行为,形成循环.

强化学习方法可以分为三类:基于价值的方法(Value-Based RL)、基于概率的方法(Policy Based RL)和结合二者的Actor-Critic方法.在参数调优领域,常用到的是基于价值的方法(DQN)和结合二者的Actor-Critic方法.

基于价值的方法根据环境的当前状态输出所有动作的价值,并选择价值最高的动作执行.Q-Learning是一种典型的基于价值的强化学习方法,其核心思想是以表格的形式存储Q值,表示在当前状态采取某个动作能获得的收益,并在学习过程中更新表格.然而,由于数据库的状态和参数复杂多样,特别是在云环境中,状态与配置的组合数量是庞大的指数级别,因此将Q-Learning应用于数据库参数调优是不切实际的.为此,DB-BERT^[33]采用深度Q网络(Deep Q Network,DQN)^[35]来进行参数调

优,通过神经网络根据输入状态输出每个动作(即推荐的参数配置)的Q值,避免建立庞大的Q表存储。

基于概率的强化学习是对每一种行为都会计算出一个概率,然后根据概率去决定采取哪种行为,这种方法中,概率值小的行为也有可能被选中。Actor-Critic则是将二者结合了起来,其中Actor部分就是通过基于概率的方法做出动作决策,Critic部分可以基于做出的动作给出动作的价值评判,这样就实现了在policy gradients基础上增加价值学习的过程。深度确定性策略梯度(Deep Deterministic Policy Gradient,DDPG)^[36]是其中的典型方法,现有工作大多采用其作为强化学习模型。首先在目标数据库系统中初始化配置并执行工作负载,同时检测运行时指标,以获取训练数据。具体来说,Actor根据当前数据库的状态(由数据库内部指标组成)推荐配置,Critic根据推荐的配置和当前数据库的状态输出动作价值,用以评估Actor推荐配置的好坏,Actor根据Critic的“打分”使用策略梯度算法更新网络参数以推荐更优的配置。通过执行Actor推荐的配置,Environment会根据执行前后数据库性能表现(由数据库外部指标组成)的变化计算奖励,Critic据此使用梯度下降方法更新网络参数,实现对Actor推荐的配置更精确的评价。此过程不断迭代直至模型收敛。

3.2 相关参数调优算法及改进

与基于贝叶斯优化的参数调优方法类似,同样可以将整个基于强化学习的调优技术划分为参数空间缩减、RL模型及知识迁移三部分。接下来依次介绍研究人员在这些方面所作的改进。

3.2.1 参数空间缩减

相较于贝叶斯优化等基于搜索的方法来说,强化学习的特性决定了其更适合多参数、高维度的优化问题。但DDPG等强化学习模型的训练速度同样受到数据维度的影响,如果把所有参数全部放入网络中训练,会导致模型参数规模过于庞大,进而导致训练成本高昂。考虑到存在一些对性能影响不显著的参数,因此可以考虑使用空间缩减的方法对参数空间进行降维。

CDBTune^[29]、QTune^[30]以及BLUTune^[31]通过专家经验人工选择好要调节的参数,这样做的好处在于能够快速选出对数据库性能影响重大的重要参数,提高效率。但一方面,面对不同的负载,同一参数的重要性会有所变化,固定的调优参数集会影响调优器的拓展性;另一方面,这种方法非常依赖

DBA经验,自动化水平低。

为了解决这一问题,Hunter^[34]采用和OtterTune相同的思路,在训练之前又加入搜索空间优化器,通过PCA(Principal Component Analysis,主成分分析)^[37]和RF(Random Forest,随机森林)^[38]把动作空间压缩到一个合理的维度。PCA是一种机器学习方法,通常用于降低数据的维数。它将一个大的变量集转换为一个保留大部分信息的较小的变量集。在Hunter中,PCA将数据 $X \in R_{u \times 1}$ 转换为低维数据 $Z \in R_{v \times 1}$,其中 u 为原始状态指标维度, l 为原始状态指标的个数,其目的就是把 u 维参数压缩到 v 维。定义一个压缩矩阵 $P \in R_{v \times u}$ 。此时 $Z = PX$, Z 中每个新变量都是 X 中原始变量的线性组合。 P 由 $S = XX^T$ 的 v 特征向量组成,这些特征向量对应于top- v 个特征值,因此不同的分量彼此正交。它有助于确定数据值在哪个方向上具有最大的影响,从而进行有益的调整。为了减少经验判断带来的不准确性,Hunter根据CDF(Cumulative Distribution Function of variance,方差的累积分布函数)确定 v 。

Hunter结合DBA的建议和实验结果来选择参数的数量。在实践中选择RF算法来捕获经验结果。RF是一种有监督的机器学习方法,用于处理分类和回归问题。它可以被认为是一个由多个决策树组成的投票算法,用于计算特征的重要性。Hunter选择分类和回归树(CART)来构造决策树^[39]。其RF算法由200个CART组成,采用多数投票原则。每个CART代表一个分类器,用于判断参数的重要性。CART构造的过程也是它的训练过程,特征的输入是配置的一个子集,标签是它们对应的性能。形式上,设 $K \in R_{m \times n}$ 表示 n 的原始构型。配置的子集是 $C \in R_{g \times n}$,其中 g -dimensional特征是从 K 和 $g < m$ 中的 m -dimensional特征中随机抽取的。每个CART的 C 都是独立且不同的。这意味着在不同的参数组合中探索每个参数的重要性。CART基于 C 及其性能,通过最佳特征(即参数)从节点分裂出的Gini系数从根节点构建树构造,生成新的叶节点。基于构建的CART可以统计 C 中每个参数Gini系数的值。200个CART的平均值可以表示每个参数的重要性。最后,Hunter根据这些平均值(即重要顺序),对参数进行筛选。

与OtterTune^[8]所用的LASSO算法相比,RF可以通过其层次结构自动捕捉参数之间的影响,而不是手动添加多项式组合来测试相关性。这使得RF能够为每个参数生成更准确的重要性分数,有利于

使用更多的参数面对用户预定义的各种规则。

另一种缩减特征空间的方法则通过 NLP 方法引入数据库专家的建议,以此对参数的数量和取值范围进行一定的限制,显著减小搜索空间,加快调优速度。众多数据库配置调优手册中包含了许多指导性的建议,在进行数据库调优之前,阅读这些包含调优提示的手册可以加快调优速度以及减少所需训练数据,同时还能发掘不同参数之间的依赖关系。不同于以往通过数据库管理员去理解调优手册中包含的许多有价值的信息,Immanuel Trummer^[40]在 21 年提出借助 NLP 理解调优文档,用于生成约束、先验知识以及优先级等额外的调优信息源。作者使用预训练的 Transformer 模型通过监督学习识别包含调优建议的语句,并根据简单的启发式算法提取具体的参数和值。在此基础上,作者又进一步提出了 DB-BERT 参数调优器,不仅从调优文档中提取建议,同时将建议翻译成具体参数和值的过程建模为强化学习过程,并在数据库系统上进行测试,以性能表现作为反馈,迭代优化翻译建议的过程,以推荐出更优的配置。

3.2.2 强化学习模型

CDBTune^[29]最早将强化学习与参数调优任务结合起来,使用的是 Actor-Critic 模型 DDPG。后续工作大多在其基础上做进一步改进。在 CDBTune 中作者直接使用的现有的 DRL 模型,该模型假定环境(即数据库状态)仅受到重新配置操作的影响,而没有利用到查询信息,这对于配置推荐和环境更新非常不利。

在 QTune^[30]中,作者提出了一种 Double-State Deep Deterministic Policy Gradient (DS-DDPG) 模型,该模型通过考虑 SQL 的丰富特性(包括查询类型、使用的表和查询代价),将 SQL 进行向量化表示,并设计了一个深度神经网络基于向量化的 SQL 预测执行前后数据库性能表现的变化。这样不仅充分利用了 SQL 本身所包含的数据分布和查询代价的信息,而且在训练完毕后,不必每次执行 SQL 来获取数据库的性能表现以推荐配置,大大减少了耗时。

而在 BLUTune^[31]中,作者进一步分析了调整配置具体是如何影响数据库性能表现的原因。作者通过对 IBM DB2^[41]中真实实例的分析,证明参数的调整会造成查询优化器创建查询执行计划(QEP)的结构和特性的改变,进而影响查询代价和执行时间。因此,作者构建了一个无监督的神经网络框架,

基于文档嵌入的思想,将整个 QEP 视为整个文档,并将模板化的查询子计划分解为单词进行向量化表示,相比之前 QTune 将 SQL 本身进行简单的特征化,这种方法能够利用 QEP 中更丰富的信息。除此之外,BLUTune 使用 advantage actor-critic 模型替换了 CDBTune 所用的 actor-critic 强化学习模型。传统 actor-critic 模型中的 $Q(s,a)$ 无法捕捉到一个操作比其他可能的操作“好多少”。为此,advantage actor-critic 中引入了优势函数 $A(s,a)$,通过学习优势而不是 Q 值函数,可以简化模型的输入,也可以减少策略网络内的方差,从而得到一个更稳定的模型。

BLUTune 还对奖励函数进行了优化。环境给予智能体在特定状态下进行调优操作的奖励对于引导智能体学习参数值至关重要。critic 使用奖励来评估给定状态下的行动,以了解优势价值。在这里奖励是基于给定查询的性能,主要由两个部分组成:历史表现和最佳表现。QTune 和 CDBTune 的奖励只依赖于与初始时间和前一次的性能,BLUTune 则在之前 N 迭代的窗口中保留了性能历史记录,以提高收敛性。对于历史上的每个性能 p_n ,BLUTune 计算当前性能 x 和 p_n 之间的标准化差值,然后对每个值应用指数衰减,最后将所有值相加。指数衰减被用来衡量历史上的表现,奖励逐渐减少。BLUTune 保持历史记录,而不仅仅是前一次的性能,以帮助引导代理,并确保它在一个好的方向上取得有意义的进展。同时为了避免代理处于当前操作和所有过去 N 个操作同等糟糕的情况,BLU 在奖励中引入了第二个组成部分 r_{best} ,即迄今为止遇到的最佳性能,用于从局部极小值中逃脱。

Hunter^[34]则注意到了冷启动的问题。面对个性化需求,例如针对不同工作负载的各种调优限制,预先训练的模型可能会与新工作负载不匹配而无法推荐最优配置。一个简单的方案是针对每个用户的工作负载进行在线探索,但是随之而来的便是冷启动的问题,因为需要从头开始探索每个用户的工作负载。Hunter 提出了一种使用混合模型的解决思路。在使用 DRL 对搜索空间进行探索之前,首先使用遗传算法 (Genetic Algorithms, GA)^[42]和人工规则生成一些粗粒度样本,用于 DDPG 的初始化。冷启动问题对样本的质量非常敏感,如果当前样本中没有最优的解决方案,那么无论使用何种采样方法,捕获最佳配置都是一项棘手的任务。因此,高质量样本的生成是至关重要的。遗传算法充分利用高性能参数之间的相似性,通过交叉和变异快速学习。虽然

很难在更多的步骤中找到最佳配置,但它可以在初步阶段获得更好的样本.接着,DDPG使用这些样本进行更细粒度的探索,以推荐最优配置.

不同于上述调优器所用的A2C强化学习方法(即DDPG),DB-BERT^[33]使用的是基于价值的强化学习网络.它将运行期间观察到的性能(在用户定义的基准测试上)转换为奖励值,该奖励值用于指导未来迭代中配置的选择.具体来说,DB-BERT使用了Double Deep Q-Networks强化学习算法.为了应用该算法,DB-BERT将数据库调优制定为具有离散状态和动作的马尔可夫决策过程(MDP),它将每个提示的处理表示为一系列决策,确定提示类型(例如,相对值与绝对值)以及提示权重.为了利用NLP进行这些决策,DB-BERT将每个决策选项与一个文本标签相关联.这允许DB-BERT使用BERT Transformer比较提示文本和决策标签.

3.2.3 知识迁移

从头开始训练模型需要很长时间,因为每一个训练步骤都要执行查询以产生新的状态和奖励.特别是在对数据仓库上具有大型复杂OLAP查询的工作负载进行调优时,这一点尤为显著.代理在训练期间产生的初始参数配置相当糟糕,因为它还没有进行足够的探索,这可能导致执行时间以小时为单位,进一步加剧了此问题.代理需要探索数千个选项,这意味着仅在执行时间上进行培训的成本非常昂贵.

在基于强化学习的调优器中,最常用的知识迁移方法为Fine-tune.如CDBTune^[29]和Qtune^[30]可以通过回放历史工作负载来预训练基本的DDPG模型.将调优器部署在用户的数据库实例中后,在处理用户的实际调优请求的同时,收集反馈信息更新样本池,并据此来更新(微调)模型.这样的过程可以帮助调优器只需借助少量真实样本快速适应新的工作负载,而非从头开始训练.

BLUTune^[31]则引入了基于迁移学习的多阶段学习过程,迁移学习侧重于存储解决一个任务时获得的知识,并将其应用于解决另一个不同但相关的任务^[43].具体来说,BLUTune首先训练一个模型,该模型在相对较短的时间内使用估计成本来测试大量查询的数千个配置.IBM DB2能在几毫秒内编译一个查询执行计划并估计成本.这种速度允许代理与环境交互,并非常快速地探索数千个参数配置,使训练高效.利用从该任务中得到的大量知识,第二个任务(最小化执行时间)对代理来说比较容易训

练.尽管估计的成本不直接等同于执行时间,但它们仍然是参数调优后性能改进的一个很好的指标.这使BLUTune可以花更少的时间训练执行时间来微调模型.现有的基于执行时间调优的ML调优方法(如QTune、CDBTune、BestConfig等)都可以与BLUTune的多阶段过程相结合.

与基于BO的调优方法中的负载映射(workload mapping)类似,基于多模型算法的在线调优系统^[32]根据特定的工作负载进行训练生成初始模型后,利用迁移学习对不同工作负载进行训练得到多个DDPG模型,从而用于系统的整体模型部署.之后为适用于工作负载变化的数据库环境,多模型算法对数据库系统的性能与状态进行在线监测,同时在此过程中根据需要动态建立DDPG模型.系统通过计算工作负载特征向量的相似度动态选择模型,根据当前工作负载对模型进行优化,同时更新数据库参数配置,直至工作负载发生变化后重新进行特征向量相似度计算.多模型算法能够利用相似工作负载的模型训练经验进行模型优化,提高学习曲线收敛速率,从而有效地提升数据库系统的吞吐率并且降低延迟,具有良好的稳定性与自适应性.

3.3 小结

基于强化学习的参数调优方法把强化学习的基本要素与数据库调优环境一一对应,借助DDPG等强化学习模型解决调优问题.与其他方法相比,基于强化学习的方法更适合于多参数、长调优时间的情况,在调优效果方面具有一定优势.然而,强化学习往往收敛较慢,通常需要数十个小时的训练才能达到预期效果.

与基于贝叶斯优化的方法类似,研究人员对强化学习方法所作的改进同样集中在缩减特征空间、强化学习模型和知识迁移三方面.(1)特征空间缩减同样是出于对效率的考虑,但相较于贝叶斯优化来说,强化学习方法更适应高维空间,因而可以考虑更大规模的参数.同时,强化学习端到端的训练过程可以将参数空间缩减作为迭代的一部分融入整个调优过程中,但目前仅有DB-BERT方法利用到了这一点,有待进一步探索.(2)对强化学习的模型的改进主要集中在DDPG模型上,可以考虑尝试其他强化学习模型(如A3C等).(3)强化学习所用的知识迁移方法多为Fine-tune.在大模型飞速发展的今天,能否训练出一个更为通用的调优模型是一个值得研究的问题.

4 其他参数调优技术

本节介绍了其余三种参数调优方法,包括基于规则匹配、基于启发式搜索和基于深度学习的方法. 相较二、三节所述方法,这些方法因自身局限,相关研究工作相对较少,但在参数调优领域仍发挥着重要作用. 本节将依次介绍这三种方法.

4.1 基于规则匹配的参数调优

基于规则匹配的参数调优器(如针对 PostgreSQL 的 PG Tune^[44]、PG Config^[45]等)设定了一系列判断条件,能针对用户提供的参数,匹配出一组合适的配置. 以参数 Effective IO Concurrency 为例,它表示 PostgreSQL 上可以同时被执行的并发磁盘 I/O 操作的数量,调高这个值,可以增加单个 PostgreSQL 会话能够并行发起的 I/O 操作的数目,进而提高查询效率;但过高的值(比保持磁盘繁忙所需的值更高)只会造成额外的 CPU 开销,因此需要根据硬件等条件进行权衡. SSD 以及其他基于内存的存储常常能处理很多并发请求,它们的最佳值通常是数百,因此在 PG Tune 中,如果用户的存储类型为 SSD,则输出的 Effective IO Concurrency 推荐大小为 200.

基于规则的参数调优器虽然易于使用、调参速度快,但由于只用到了数据库较少的信息,调优器模型较为简单,仅能实现十分粗粒度的调整. 为此 Wei 等人^[46]提出了一种基于模糊规则的数据库管理系统性能自调优新方法. 模糊规则(fuzzy rules)能够定义在 $X \times Y$ 上的二元模糊关系: IF x is A THEN y is B. 其中 A 和 B 分别是 X 和 Y 上的模糊集,“ x is A”称为“前提”,“ y is B”称为“结论”. 此调优器通过模式来建立一套性能指标和参数之间的关系,即当性能指标处于某一状态,则参数应如何变化.

以调优 Oracle 数据库为例,该方法首先选定数据库系统中的 DBCS、SPS、PAT 三个参数以及 BCHR、LCHR、SPU、IMSR 四个性能指标,调整参数值,提取 AWR 报告,分析参数对性能的影响,应用 IF-THEN 规则生成模糊规则. 然后将参数和性能指标的值离散化,即将其分为不同的区间,使用 Low、Medium、High 等语言变量为每个区间划分等级,然后为每个等级构建隶属函数. 由于参数的隶属函数为分段函数,当不同参数发生变化时,计算出其隶属函数值,用以判断此参数分布在什么区间,然

后计算出性能指标的隶属函数值,判断性能是否得到优化,以确定是否更新参数配置,最后通过参数分布的方差来判断参数配置是否平衡,进而做出优化以增加资源利用率以及提高系统稳定性.

然而此方法需要人工建立模糊规则库,导致其对样本,数量的依赖程度较大,本质还是依靠人工调节参数而且只能调节少量参数,有较大的局限性. 且构建出的模糊规则在不同场景下不通用.

4.2 基于启发式搜索的参数调优

启发式搜索的思想是基于当前已知对下一步搜索的每个分支进行估计,选择更优的分支. Chen 等^[47]利用启发式搜索的思想,将在之前系统中发现的有用经验复用至现有的系统,提出了“experience transfer”,将系统配置之间的依赖关系视为配置调优的有用经验,提出一种贝叶斯网络指导调优算法来发现最优配置. 针对同一类型的系统,系统本身总会随着硬件升级、现有软件版本变化、拓扑结构变化等进行变化,然而不论系统怎么更新迭代,相似模型之间参数的依赖性通常不变. 捕捉到这一特性,它发掘参数之间的依赖关系,通过从上一代建立的贝叶斯网络中抽样产生新的样本,应用重组算子将新旧样本重组,并从中得到新一代网络,循环迭代,新网络捕获配置空间采样的最优区域,整个学习过程遵循进化搜索的策略,最终收敛到最优优点.

BestConfig^[48]提出 divide-and-diverge sampling (DDS)采样方法和 recursive bound-and-search (RBS)调优方法协同工作实现采样和调优过程. DDS 首先要确定需要调节的参数数量,记为 N ,随后将每个参数的取值范围都划分成 K 个区间,便形成一个由 K^N 个点组成的离散空间,以此将参数的高维连续空间划分为多个离散子空间,采样时从每个子空间随机选择一个点,保证了对整个参数空间的完全覆盖,同时在采样时,对每个参数的每个区间只考虑一次,以达到降低测试成本且满足资源限制的目的,而参数的区间个数可以等于测试次数,即 DDS 先天满足了在资源限制方面具有可扩展性. 在初次调优时, BestConfig 通过有界递归搜索算法(Recursive Bound Search, RBS)从离散空间中随机选择 m 个样本作为参数配置来测试数据库性能,测试结束后选择使数据库性能最好的样本,记为 C_1 ,之后找到在该点每个参数值所在的最小区域,以这些区域确定边界,通过调整样本集的大小和轮数来满足资源限制要求,在有界子空间中进行更精细的搜索,当找不到性能更好的采样点时,采样器重新在整个参数空间采样,

以避免陷入次优有界子空间,如此往复,直到数据库达到理想的性能。

相比基于规则的调参技术,利用 BestConfig 调参的自动化程度有所提升,但它仍有一些局限性:首先,它基于启发式算法,可能无法在有限的时间内找到最佳的参数配置;其次,它需要搜索整个参数空间,效率很低;最后,它没有充分利用历史数据,也就是说每次调参都需要从头开始搜索,一定程度上会导致资源和时间的浪费。

4.3 基于深度学习的参数调优

高斯过程模型在高维度、大数据集的情况下表现较差^[12],因此一些工作尝试使用深度学习预测给定参数下数据库性能,以此替代贝叶斯优化中的高斯过程模型。如文献[49]使用深度神经网络(Deep Neural Network, DNN)作为调优模型。具体来说,使用的 DNN 有两层隐藏层,每层 64 个神经元,使用 ReLU 作为激活函数。为了避免局部最优,在神经网络中加入了高斯噪声以探索未知区域。

此外,无论是基于贝叶斯的调优算法还是基于强化学习的调优算法都需要通过反复运行工作负载获取数据库反馈来探索最佳的参数配置,而在某些场景下,实际执行的代价十分高昂,无法反复迭代,特别是在云数据库上(如大型分布式数据库、Spark 应用等)。一些基于深度学习的参数调优方法通过训练深度学习模型来预测参数调整后的执行结果,无需真正运行工作负载,借此避免调参过程中反复迭代运行所带来的开销。

IBTune^[50]是一个典型的基于深度学习的参数调优器,它利用缓存未命中率和缓冲池大小之间的关系来优化云数据库的内存分配。具体来说,它利用来自工作负载的信息及收集到的数据库指标(CPU 利用率、QPS 等)来找出每个实例的可容忍未命中率。然后,它利用未命中率和分配的内存大小之间的关系来单独优化目标缓冲池大小。整个调优流程的核心在于准确预测调整内存空间后的数据库响应时间。为此 Tan 等^[50]设计了一个成对的深度神经网络,该网络使用数据库当前状态、调整后内存大小等特征来预测参数调整后响应时间的上限。只有当预测的响应时间上限在安全限制内时,才能调整目标缓冲池大小。在保证可靠性的同时,与依赖手动配置的原始系统相比,在生产环境部署的数据库中安全地减少了 17% 以上的内存占用。

然而 IBTune 的深度学习模型在训练完成后就不再变化,未能有效利用数据库在实际运行过程中

产生的数据,也导致了 IBTune 面对差异较大的工作负载时调优效果较差,不能满足用户多样化的使用需求。针对这一问题,Chen 等人^[51]在设计针对 Spark 的轻量级调优系统 LITE(Lightweight knob recommender system)时,加入了基于对抗学习的自适应模型更新系统,使得训练好的深度学习模型在运行过程中可以根据收集到的反馈即时调整参数,以更好地适应实际工作负载。他们训练了名为 NECS(Neural Estimator via Code and Scheduler)的深度学习模型,用于预估 Spark 应用在一组给定的参数上的执行性能。此外, LITE 使用迁移学习的方法,先在小样本上离线训练模型,再迁移到实际场景中,进一步提高了模型训练的效率。

总体来说,基于深度学习的参数调优方法以深度神经网络的预测结果代替实际执行结果,更好地适应高维度、大数据规模情况,显著减少了调优过程带来的开销。但一方面,由于数据库性能受多种因素影响,较为复杂,预测结果的可靠性和准确性难以保证;另一方面,当迁移到新的工作负载时,基于学习的方法可能会推荐糟糕的设置,泛化性能较差。

4.4 小结

本节介绍了基于规则匹配、启发式搜索和深度学习的三种调优算法。相较前文介绍的基于贝叶斯和强化学习的方法,这些方法因自身局限性,相关研究工作相对较少。如基于规则匹配的方法受限于预设的规则,优化性能较差,改进的空间较小;基于启发式搜索的方法效率较低,一定程度上可被基于贝叶斯优化的方法替代;基于深度学习的方法很大程度上依赖于大量的训练数据及丰富的负载语义特征,多用于 Spark 等大数据平台,难以迁移到传统关系型数据库上。但另一方面,这些方法同样有其独到之处,也可用于辅助其他调优算法。如基于规则的方法可靠性强,能很好的利用专家经验,因而广泛用于工业界,在下一节中会就此展开详细介绍;基于启发式搜索的方法中用到的分块等思想同样可用于贝叶斯优化当中;基于深度学习的方法可替换调优流程中的某些组件。

综合前文的分析介绍及现有工作^[3,14],本文对现有五种数据库参数配置智能调优方法从调优质量、时间和适用性的角度做了对比,并总结了各自优缺点,结果如表 3 所示。调优质量通常通过吞吐量(Throughput)或时延(Latency)的提升比例来衡量,具体取决于设置的优化目标。吞吐量通常用每秒处

理事务量(Transactions Per Second, TPS)或每秒处理查询量(Queries Per Second, QPS)来衡量,延迟通常用执行时间来衡量. 这些指标是数据库性能最直观的体现,也是实际生产过程中用户最关心的部分. 调优时间通过达到最优效果所需的调优时间或轮次来衡量. 适应性主要取决于更换负载或数据库硬件后调优算法的表现,反映了算法对不同环境的适应情况.

表3 数据库参数配置智能调优方法对比

	基于贝叶斯优化	基于强化学习	基于深度学习	基于启发式搜索	基于规则
调优质量	较好	好	较好	较差	差
调优时间	较长	长	较长	较短	短
适应性	较好	好	较差	差	差
优点	在积累一定历史经验时,能以较快的速度和较低的成本快速给出结果	系统适应性强;在长调优时间和高参数维度时性能最佳	显著减少调优过程中的时间、资源开销	调优速度快,资源占用少	调优几乎不消耗时间;可靠性高,不会出现非法配置;易于使用
缺点	对训练样本要求高;难以应对高维参数空间	训练时间长,收敛慢	依赖于 Spark 等复杂查询语言内含的大量信息,泛用性差	调优效果差,自动化水平低	推荐参数固定,不能适应多样的负载

5 参数配置智能调优在工业界的应用

在工业界,传统参数配置调优主要由经验丰富的DBA完成. 往往依靠历史经验选择要调整的参数及取值,通过观察数据库运行指标评估调优效果. 这导致(1)调优成本高、(2)调优周期长、(3)调优效果不可靠,因而绝大多数用户数据库还是使用默认配置. 为解决上述问题,许多主流数据库厂商都推出了和数据库配套的参数调优工具. 本节选择了其中四种有代表性的工具加以介绍.

5.1 PGConfig

如4.1所提到的,PGConfig是典型的基于规则匹配的参数调优器,用于帮助用户调整PostgreSQL数据库参数. 它设定了一系列判断条件,能针对用户输入的数据库软硬件特征(操作系统、存储类型、应用领域、CPU数量、内存大小和最大连接数),匹配出一组推荐的参数配置. 它提供了在线调优网页,界面非常简洁、易用,且因为无需训练,能即时给出推荐结果. 也正因如此,它的调优是非常粗粒度的,调优效果也不那么令人满意.

5.2 Mysql Tuner

Mysql Tuner^[52]是一个分析Mysql配置和性能基准的Perl软件,它检索当前的配置变量和状态数据,并显示与性能、安全性等相关的统计信息和建议的完整列表. 它持续维护并支300+指标配置,如Galera Cluster, TokuDB, 性能模式, Linux OS Metrics, InnoDB, MyISAM, Aria等.

Mysql Tuner实际上用脚本的方式收集了数据

库所有的变量、全局变量、状态、全局状态等信息,利用官方文档中的公式做计算,和官方文档中的性能基准作比较然后进行参数推荐. Mysql Tuner脚本运行简单,只需一条命令即可启动,连接成功会输出数据库的分析报表,如数据库的性能指标、存储引擎指标、优化建议和推荐配置等.

在配置推荐时,Mysql Tuner虽然会给出调优变量的名称和参数值,但仍需要DBA手动调节配置文件. 此外,和PGConfig等基于规则的调优器类似,Mysql Tuner的优化效果比较一般.

5.3 KeenTune

KeenTune^[53]是阿里推出的一款自调优工具,它最显著的特点是它的兼容性、可移植性很高,它可以部署在不同操作系统上,使用多种调优方法(主要为贝叶斯优化方法),对多种调优对象进行调优(数据库管理系统、操作系统、应用服务等).

KeenTune一共包含四个模块:keentune-daemon、keentune-brain、keentune-target以及keentune-bench. daemon作为中央控制组件,用于调度和监控其他组件,并完成与用户的交互;brain顾名思义,它根据指定的参数和调优范围,使用一种选中的方法进行训练并生成参数配置推荐;target部署在业务环境上,用于将参数配置部署在数据库管理系统中;bench使用内置脚本调用不同的基准测试工具来评估所推荐配置的运行结果. 因为KeenTune主要采用的是贝叶斯优化方法,因此它的具体调优流程也与一般的贝叶斯优化调优流程相类似:在调优过程中,daemon接收用户发来的指令并向brain发送开启调优的消息,brain开启训练并生成推荐参数配置,发

送给 target, target 将其部署在数据库管理系统中, 最后由 bench 进行评估并将结果作为反馈发送给 brain, 完成一轮迭代.

KeenTune 在开始调优前, 首先需要考虑参数选择和取值范围的设置. 对于参数取值范围, 它主要使用专家经验进行设置, 并加入预调优环节来验证参数取值的可行性. 对于参数选择, 它除了依赖专家经验进行设置, 还提出了一种参数可解释性算法 XSen. XSen 包括三种方法: 单变量(univariate)灵敏度、非线性(non-linear)灵敏度以及聚合(aggregation)灵敏度. 单变量灵敏度通过估计参数与调优结果之间的交互信息生成; 非线性方法集成了 SHAP^[54]方法, 即首先用初始化代理模型的数据集训练一个机器学习模型如 XGBoost^[55], 然后使用 SHAP 估计参数的敏感度; 聚合灵敏度则由单变量灵敏度与非线性灵敏度进行线性组合而成. 在获得参数灵敏度之后, 通过设置参数敏感度的阈值从而实现缩小参数空间.

KeenTune 在调优方法上, 集成了许多已有的贝叶斯优化的方法, 包括基于高斯过程的贝叶斯优化、TPE(Tree-structured Parzen Estimator)^[6]以及 HyperBan^[56]等. 除此之外, 还有 KeenTune 所提出的 BGCS 方法. 该方法基于 HORD^[57]框架, 使用 CNN/MLP 模型取代 RBF 模型作为代理模型, 以提高代理模型预测准确性. 因为 HORD 算法的参数变异过程整体来说是随机的, 在这个过程中会产生大量无意义的参数配置, 因此 BGCS 针对这个过程加入敏感性梯度对参数变异的过程进行指导, 优先针对敏感性梯度较高的参数进行变异, 同时保持性能梯度低参数的取值, 降低代理模型预测误差的干扰,

避免最终盲目随机地生成大量参数配置.

KeenTune 在知识迁移方面, 为避免重复训练, 它引入了一个专门用于保存历史中间和最终调优数据的仓库, 这个仓库中保存三类数据: 敏感参数列表, 通过维护针对不同云应用或服务的敏感参数列表缩减参数空间; 推荐参数配置, 通过使用类似任务的推荐参数配置来加速调优; 历史数据样本, 这些数据样本可以用于训练和完善代理模型.

5.4 XTune

XTuner^[58]是 OpenGauss 数据库中集成的一个参数调优器组件, 通过结合深度强化学习和全局优化算法实现, 用来进行参数自调优. XTuner 最大的特点是结合强化学习和贝叶斯方法. 对于与数据库的内部状态相关联的参数配置, 例如 OpenGauss 数据库中的 shared_buffers, 该参数用来设置数据库服务器将使用的共享内存缓冲区的大小, 它与缓冲池命中率密切相关. 诸如此类内部状态指标可以被用于强化学习中进行配置推荐. 此外, 该方法还将当前参数的值也作为数据库的一个状态, 作为强化学习模型中智能体输入的一部分. 在奖励函数的设计上, 不仅考虑了基准测试下的数据库性能表现, 还以此过程中的内存使用情况作为正则项约束. 对于其余一些没有与之相对的内部状态的参数, 例如 random_page_cost、seq_page_cost 等, 由于它们彼此之间是相关的, 因此采用粒子群算法、贝叶斯优化等全局优化算法进行搜索.

5.5 小结

本节主要介绍了主流数据库厂商各自的参数调优工具, 如表4所示. 总体来看, 与学术界相比, 工业界参数调优工具普遍具有如下特点:

表4 工业界参数调优工具对比				
	PGConfig ^[45]	MySQL Tuner ^[52]	KeenTune ^[16]	XTune ^[58]
支持数据库	PostgreSQL	MySQL	MySQL	OpenGauss
所属厂商	开源社区	Oracle	Alibaba	Huawei
调优算法	基于规则	基于规则	基于贝叶斯	基于贝叶斯+强化学习
优点	调优速度快, 不占用系统资源; 调优安全性高; web 界面使用方便	可自动分析数据库配置和运行状态, 效果较好; 支持图形化界面, 使用方便	优化效率高; 兼容性、可移植性好	支持多种调优粒度, 较为灵活
缺点	调优范围固定, 调优规则简单, 优化效果差	收集数据所需时间长(至少24小时)	依赖专家经验/历史数据; 易用性较差	强化学习方法收敛慢; 易用性极差

(1)算法集成: 与学术界单一方法不同, 工业界参数调优工具往往集成了多种算法, 如阿里的 KeenTune 中集成了 TPE、HORD 等多种贝叶斯优化算

法、华为 XTune 中集成了强化学习、贝叶斯优化等多类方法. 其优势在于提供给用户的选择更加丰富, 利于适应不同的使用场景. 但遗憾的是这些方

法彼此之间并无太大联系,仅仅是简单地堆砌.

(2)策略保守:工业界参数调优工具通常更加保守,具体表现在如下两方面.其一,调优算法上,在学术界基于规则的方法寥寥无几,但其仍是工业界主流方法之一;其二,调优范围上,学术界一次调优通常会同时修改几十个参数,但在工业界参数调优工具中一次调优只会涉及几个参数.这主要是因为在实际生产过程中,对数据库可靠性的要求较高,一些激进的调优策略往往会带来数据库性能抖动乃至宕机的风险^[14].

(3)用户友好:工业界参数调优工具直接服务于数据库用户,比较注重使用体验.诸如 PGConfig、MySQLTuner 等工具都有着简洁明了的 web 工具或图形化界面,易于使用.

6 未来研究方向

通过上述分析和比较,可以看出数据库参数配置智能调优技术取得了一定的成果,但在面对复杂多变的负载和用户在线动态调优的需求时,现有的方法适用场景有限,应用参数调优技术仍然面临着许多问题和挑战,还需进一步的研究:

(1)即时在线调优:在数据库的运行过程中,工作负载往往是动态变化的,参数配置合适与否又与负载息息相关,因而产生了随负载变化动态在线调优的需求.现有方法虽然在调优效率上取得了一定进展,但离在线调优仍有一定距离.特别是在遇到新的工作负载时,无论是基于贝叶斯方法还是强化学习方法都需要反复多次迭代,往往需要数十小时的调优过程才能达到理想的效果.如何提高参数调优效率,平衡推荐参数质量和调优时间,是参数配置智能调优技术还需要继续研究的关键问题.

(2)调优安全性和可靠性:目前主流参数调优方法仅关心调优的最终效果,而调优过程中常常出现性能下降,甚至在某些情况下会因设置了非法参数导致数据库宕机^[14].如在测试比较调优算法优劣的过程中,屡次因参数 shared_buffer 设置不合适导致数据库无法启动.然而安全性和可靠性是数据库系统最关心的问题,特别是在金融、国防等领域,数据库性能的抖动乃至宕机是不可接受的.目前仅有少数工作考虑到了对调优过程中安全性的约束,且大多借助了黑盒方法,不具有可解释性,也未能完全避免安全风险.如何结合专家知识,构建白盒方法

确保调参过程中数据库的安全性是未来一个值得探索的方向.

(3)轻量化调优:无论是基于贝叶斯优化的方法还是基于强化学习的参数调优的方法,在调优过程中均需要反复运行负载,以收集统计信息来衡量数据库状态.此过程既浪费时间,又极大地占用了系统资源.如何在低资源、低配置的情况下高效调优,减少对目标系统资源的消耗,还有待进一步探索和研究.

(4)通用调优技术:当前参数调优技术往往是对某种(或一类)特定数据库设计,如 PostgreSQL、MySQL 等,甚至对数据库版本都有着严格限制,难以应用到不同种类或不同版本的数据库上,极大限制了调优技术的使用,也不利于不同技术之间进行比较.如何设计一套具有普适性的调优架构,解决兼容性问题,是一个值得研究人员考虑的问题.

(5)参数调优的评测技术:提供统一的测试平台公平地比较和分析不同参数调优方法之间的性能,能够帮助研究人员更好地研究和发展学习型索引.为了更好地挖掘参数调优技术的潜在优势,测试平台应该尽可能地包含多方面的测试内容.然而,现有的测试基准仅考虑某一类单一负载(如 TPC-H、TPC-DS^[59]等),缺乏对真实场景下混合负载发生变化时的评测.此外,不同方法调优的对象也不尽相同,难以直观看出各自性能优劣.因此,在未来还需进一步深入研究参数调优的评测技术.

7 总 结

本文从方法、应用与挑战三个方面对现有数据库参数配置智能调优技术进行系统综述.首先将现有方法按照所用调优算法不同分为五类:基于贝叶斯优化的、基于强化学习的、基于深度学习的、基于启发式搜索的和基于规则的.对于基于贝叶斯的方法和基于强化学习的方法,从基础模型入手,对其实现原理、构造和调优过程进行了剖析,并从特征空间缩减、模型自身和知识迁移三个方面对相关研究工作进行了梳理和总结;对于基于搜索的、基于规则匹配的和基于深度学习的调优方法,做了简要介绍,并对比了不同方法之间的异同.之后,本文介绍了工业界目前主流的参数调优工具,分析了在实际应用过程遇到的主要问题及产生原因.最后,探讨了参数配置智能调优未来发展趋势.参数配置智能调优作为数据库领域的研究热点,仍有许多问题亟待解决.未来参数调优将朝着安全、动态、通用、在线的

方向进一步发展,并最终广泛应用在实际生产过程中,帮助数据库发挥更优越的性能.

参 考 文 献

- [1] <https://pgtune.leopard.in.ua/>
- [2] Sun L, Zhang S, Ji T, et al. Survey of data management techniques powered by artificial intelligence. *Journal of Software*, 2020, 31(3): 600-619 (in Chinese)
(孙路明, 张少敏, 姬涛等. 人工智能赋能的数据管理技术研究. *软件学报*, 2020, 31(3): 20)
- [3] ZhangXinyi, Zhuo Chang, Yang Li, Hong Wu, Jian Tan, LiFeifei, CuiBin. Facilitating database tuning with hyperparameter optimization: a comprehensive eperimental evaluation// *Proceedings of the VLDB Endowment*. Sydney, Australia, 2022, 14(1): 1808-1821
- [4] Li G, Zhou X, Cao L. AI meets database: AI4DB and DB4AI//*Proceedings of the 2021 International Conference on Management of Data*. Virtual Event, 2021: 2859-2866
- [5] Li Y, Shen Y, Zhang W, et al. Openbox: a generalized black-box optimization service//*Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*. Virtual Event, 2021: 3209-3219
- [6] Bergstra J, Bardenet R, Bengio Y, et al. Algorithms for hyperparameter optimization//*Proceedings of the 24th International Conference on Neural Information Processing Systems*. Granada, Spain, 2011: 2546-2554
- [7] Brochu E, Cora V, Freitas N. A tutorial on bayesian optimization of expensive cost functions, with application to active user modeling and hierarchical reinforcement learning. *arXiv preprint arXiv: 1012.2599*, 2010
- [8] Van Aken D, Pavlo A, J. Gordon G, et al. Automatic database management system tuning through large-scale machine learning//*Proceedings of the 2017 ACM International Conference on Management of Data*. Chicago, USA, 2017: 1009-1024
- [9] Duan S, Thummala V, Babu S. Tuning database configuration parameters with iTuned//*Proceedings of the VLDB Endowment*. Lyon, France, 2009, 2(1): 1246-1257
- [10] Nembrini S, König IR., Wright MN.. The revival of the Gini importance? *Bioinform*, 2018, 34,(21): 3711-3718
- [11] Zhang X, Wu H, Chang Z, et al. ResTune: Resource oriented tuningboosted by Meta-Learning for cloud databases// *Proceedings of the 2021 International Conference on Management of Data*. Virtual Event, China, 2021: 2102-2114
- [12] Kunjir M, Babu S. Black or White? How to develop an AutoTuner for memory-based analytics//*Proceedings of the 2020 ACM SIGMOD International Conference on Management of Data*. Portland, USA, 2020: 1667-1683
- [13] Cereda S, Valladares S, Cremonesi P, Doni S. CGPTuner: A contextual Gaussian process bandit approach for the automatic tuning of IT configurations under varying workload conditions// *Proceedings of the VLDB Endowment*. Copenhagen, Denmark, 2021, 14(8): 1401-1413
- [14] Zhang X, Wu H, Li Y, et al. Towards dynamic and safe configuration tuning for cloud databases//*Proceedings of the 2022 International Conference on Management of Data*. Philadelphia, USA, 2022: 631-645
- [15] Kanellis K, Ding C, Kroth B, et al. LlamaTune: Sample-efficient DBMS configuration tuning//*Proceedings of the VLDB Endowment*. Sydney, Australia, 2022, 15(11): 2953- 2965
- [16] <http://keentune.io/home>
- [17] Weerts H, Mueller A, Vanschoren J. Importance of tuning hyperparameters of machine learning algorithms. *arXiv preprint arXiv:2007.07588*, 2020
- [18] Probst P, Boulesteix A, Bischl B. Tunability: Importance of hyperparameters of machine learning algorithms. *Journal of Machine Learning Research*, 2019, 20(1): 1934-1965
- [19] Tibshirani R. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society*, 1996, 58(1): 267-288
- [20] Lundberg SM., Lee S. A unified approach to interpreting model predictions//*Proceedings of the 31st International Conference on Neural Information Processing Systems*. Long Beach, USA, 2017: 4768-4777
- [21] Letham B, Calandra R, Rai A, Bakshy E. Reexamining linear embeddings for high-dimensional Bayesian optimization// *Advances in Neural Information Processing Systems 33*. Virtual Event, 2020: 1546-1558
- [22] Wang Z, Hutter F, Zoghi M, et al. Bayesian optimization in a billion dimensions via random embeddings. *Journal of Artificial Intelligence Research*, 2016, 55(1): 361-387
- [23] Eriksson D, Jankowiak M. High-dimensional Bayesian optimization with sparse axis-aligned subspaces//*Proceedings of the 37th Conference on Uncertainty in Artificial Intelligence*. Toronto, Canada, 2021: 493-503
- [24] Nayeib A, Munteanu A, Poloczek M. A framework for Bayesian optimization in embedded subspaces//*Proceedings of the 36th International Conference on Machine Learning*. Long Beach, USA, 2019: 4752-4761
- [25] Fekry A, Carata L, Thomas P, et al. To tune or not to tune? : Search of optimal configurations for data analytics//*Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. Virtual Event, 2019: 2494-2504
- [26] Hiemstra D. A probabilistic justification for using $tf \times idf$ term weighting in information retrieval. *International Journal on Digital Libraries*, 2000, 3(2): 131-139
- [27] Feurer M, Letham B, Bakshy E. Scalable meta-learning for Bayesian optimization using ranking-weighted gaussian process ensembles//*Workshop at the 35th International Conference on Machine Learning*. Stockholm, Sweden, 2018, 7: 5
- [28] Van Aken D, Yang D, Brillard S, et al. An inquiry into machine learning-based automatic configuration tuning services on real-world database management systems//*Proceedings of the VLDB Endowment*. Copenhagen, Denmark, 2021, 14(7): 1241-1253
- [29] Zhang J, Liu Y, Zhou K, et al. An end-to-end automatic cloud

- database tuning system using deep reinforcement learning//
Proceedings of the 2019 International Conference on Management
of Data. Amsterdam, Netherlands, 2019; 415-432
- [30] Li G, Zhou X, Li S, Gao B. QTune: A query-aware database
tuning system with deep reinforcement learning//Proceedings of
the VLDB Endowment. Los Angeles, California, 2019, 12(12):
2118-2130
- [31] Henderson C, Bryson S, Corvinelli V, et al. BLUTune: Query-
informed multi-stage IBM Db2 tuning via ML//Proceedings of
the 31st ACM International Conference on Information and
Knowledge Management. Atlanta, USA, 2022; 3162-3171
- [32] Gur Y, Yang D, Stalschus F, et al. Adaptive multi-model
reinforcement learning for online database tuning//24th
International Conference on Extending Database Technology.
Nicosia, Cyprus, 2021; 439-444
- [33] Trummer I. DB-BERT: A database tuning tool that “Reads the
Manual”//Proceedings of the 2022 International Conference on
Management of Data. Philadelphia, USA, 2022; 190-203
- [34] Cai B, Liu Y, Zhang C, et al. HUNTER: An online cloud
database hybrid tuning system for personalized requirements//
Proceedings of the 2022 International Conference on
Management of Data. Philadelphia, USA, 2022; 646-659
- [35] Van Hasselt H, Guez A, Silver D. Deep reinforcement learning
with double Q-Learning//Proceedings of the AAAI Conference
on Artificial Intelligence. Phoenix, USA, 2016; 2094-2100
- [36] Lillicrap T, Hunt J, Pritzel A, et al. Continuous control with
deep reinforcement learning. arXiv preprint arXiv:1509.02971,
2015
- [37] Holland S. Principal components analysis (PCA). Department
of Geology, University of Georgia, Athens, Greece, 2008;
30602-2501
- [38] Oshiro T M, Perez P S, Baranauskas J A. How many trees in a
random forest? //Proceedings of the Machine Learning and
Data Mining in Pattern Recognition; 8th International
Conference. Berlin, Germany, 2012; 154-168
- [39] Lewis R J. An introduction to classification and regression tree
(CART) analysis//Proceedings of the Annual meeting of the
society for academic emergency medicine. San Francisco,
USA, 2000
- [40] Trummer I. The case for NLP-enhanced database tuning:
towards tuning tools that “read the manual”//Proceedings of the
VLDB Endowment. Copenhagen, Denmark, 2021, 14 (7) :
1159-1165
- [41] DbIBM2 Documentation. Configuration parameters summary.
<https://www.ibm.com/docs/en/db2/11.5?topic=parameters-configuration-summary>. Accessed: 2022-05-01
- [42] Lessmann S, Stahlbock R, Crone S F. Optimizing
hyperparameters of support vector machines by genetic
algorithms//Proceedings of the International Conference on
Artificial Intelligence. Las Vegas, USA, 2005; 74-82
- [43] Tan C, Sun F, Kong T, et al. A survey on deep transfer
learning//Proceedings of the 27th International Conference on
Artificial Neural Networks. Rhodes, Greece, 2018; 270-279
- [44] <https://pgtune.leopard.in.ua/>
- [45] <https://www.pgconfig.org/>
- [46] Wei Z, Ding Z, Hu J. Self-tuning performance of database
systems based on fuzzy rules//Proceedings of the 11th
International Conference on Fuzzy Systems and Knowledge
Discovery. Xiamen, China, 2014; 194-198
- [47] Chen H, Zhang W, Jiang G. Experience transfer for the
configuration tuning in large-scale computing systems. IEEE
Transactions on Knowledge and Data Engineering, 2010,
23(3): 388-401
- [48] Zhu Y, Liu J, Guo M, et al. Bestconfig: Tapping the
performance potential of systems via automatic configuration
tuning//Proceedings of the 2017 Symposium on Cloud
Computing. Santa Clara, USA, 2017; 338-350
- [49] Van Aken D, Yang D, Brillard S, et al. An inquiry into
machine learning-based automatic configuration tuning services
on real-world database management systems//Proceedings of
the VLDB Endowment. Copenhagen, Denmark, 2021, 14(7):
1241-1253
- [50] Tan J, Zhang T, Li F, et al. ibtune: Individualized buffer tuning
for large-scale cloud databases//Proceedings of the VLDB
Endowment. Los Angeles, USA, 2019, 12(10): 1221-1234
- [51] Lin C, Zhuang J, Feng J, et al. Adaptive code learning for spark
configuration tuning//Proceedings of the IEEE 38th
International Conference on Data Engineering. Kuala Lumpur,
Malaysia, 2022; 1995-2007
- [52] <https://github.com/major/MySQLTuner-perl>
- [53] Wang R, Wang Q, Hu Y, et al. Industry practice of
configuration auto-tuning for cloud applications and services//
Proceedings of the 30th ACM Joint European Software
Engineering Conference and Symposium on the Foundations of
Software Engineering. Virtual Event, 2022; 1555-1565
- [54] Lundberg S M, Lee S I. A unified approach to interpreting
model predictions//Proceedings of the 31st International
Conference on Neural Information Processing Systems. Long
Beach, USA, 2017; 4765-4774
- [55] Chen T, Guestrin C. Xgboost: A scalable tree boosting
system//Proceedings of the 22nd Acm Sigkdd International
Conference on Knowledge Discovery and Data Mining. San
Francisco, USA, 2016; 785-794
- [56] Li L, Jamieson K G, DeSalvo G, et al. Hyperband: Bandit-
based configuration evaluation for hyperparameter optimization//
Proceedings of the 5th International Conference on Learning
Representations. Toulon, France, 2017; 53
- [57] Ilievski I, Akhtar T, Feng J, et al. Efficient hyperparameter
optimization for deep learning algorithms using deterministic rbf
surrogates//Proceedings of the AAAI Conference on Artificial
Intelligence. San Francisco, USA, 2017, 31(1): 822-829
- [58] Li G, Zhou X, Sun J, et al. opengauss: An autonomous
database system//Proceedings of the VLDB Endowment.
Copenhagen, Denmark, 2021, 14(12): 3028-3042
- [59] Poess M, Nambiar R O, Walrath D. Why you should run TPC-
DS: A workload analysis//Proceedings of the 33rd International
Conference on Very Large Data Bases. Vienna, Austria, 2007;
1138-1149



LI Yi-Yan, Ph. D. candidate. His research interests include database knob tuning, machine learning.

TIAN Ji-Kun, M. S. candidate. His research interests

include database knob tuning, reinforcement learning.

PU Zhao, M.S. candidate. His research interests include database knob tuning, bayesian optimization.

LI Cui-Ping, Ph. D., professor, Ph.D. supervisor. Her research interests include social network analysis, social recommendations, data analysis and mining.

CHEN Hong, Ph. D., professor, Ph.D. supervisor. Her research interests include database technology, the new hardware platform of high performance computing.

Background

After more than half a century of research and development, the database management system (DBMS) has achieved rapid progress and excellent performance. However, the traditional database is a static system and pays more attention to the generalization. On the other hand, the characteristics of large data scale, high query concurrency, and frequent connection operations in big data application scenarios put forward higher requirements for the performance of database systems. Hence researchers began to focus more on specific self-tuning problem. In the past 10 years, artificial intelligence technology, especially the development of machine learning and deep learning, has made many breakthroughs in natural language processing, image recognition and other fields. At the same time, represented by DB2's learning query optimizer LEO, ideas and methods of learning and statistics have begun to be valued in the database field. The new technology of data management empowered by artificial intelligence has received more extensive attention and more in-depth research.

Automatic tuning of database parameters is one of the main applications of artificial intelligence technology in databases. Configuration tuning based on artificial intelligence technology can effectively utilize historical load information and database performance, and automatically optimize according to the current workload without manual participation. Configuration tuning can not only reduce labor cost, but also further improve the

performance of database in data access, query optimization, query execution and other aspects.

This paper introduces the progress of the current intelligent knob tuning work from the aspects of industry and academia. In terms of academia, this paper divides the existing intelligent knob tuning research into five categories: intelligent tuning algorithms based on bayesian optimization, reinforcement learning, deep learning, heuristic search and rules. For each type of method, first introduce its basic principles, analyze its structure and basic tuning process, and summarize their respective advantages and existing problems. Then, this paper divides the improvement and optimization of the basic model into three categories: parameter space reduction, model itself and knowledge transfer, and introduces them in detail and sorts out the latest research progress.

In the industry, different users have different requirements for tuners in the actual production environment. Therefore, compared with academia, these tools often integrate a variety of algorithms to meet the different needs of users. Therefore, this article classifies the mainstream knob tuning tools in the industry according to the database vendors they belong to, and introduces them in turn. Finally, this paper summarizes and compares the advantages and disadvantages of various technologies, and looks forward to the future research directions.