

自动调优技术在智慧校园数据库的应用与优化

田 浩

(辽宁金融职业学院, 辽宁沈阳 110122)

【摘要】文章分析了智慧校园数据平台建设过程中应用MPP架构数据库出现的主要问题,然后提出将基于机器学习的数据库自动调优技术OtterTune应用到其中,并针对MPP架构数据库负载刻画不够准确的问题,提出一种新的算法与解决思路,最后对新的算法进行了性能分析,结果显示调优效果提升。

【关键词】智慧校园;MPP架构数据库;数据库自动调优;自动参数调优;多节点负载刻画

【doi:10.3969/j.issn.2095-7661.2021.04.008】

【中图分类号】TP311.13

【文献标识码】A

【文章编号】2095-7661(2021)04-0026-04

Application and Optimization of Automatic Tuning Technology in Smart Campus Database

TIAN Hao

(Liaoning Finance Vocational College, Shenyang, Liaoning, China 110122)

Abstract: This paper analyzes the main problems in the MPP database applied in the construction of smart campus data platform, then proposes to apply OtterTune which is automatic database tuning technology based on machine learning to the tuning process, and puts forward a new algorithm and solution to the problem that the load characterization of MPP architecture database is not accurate enough. Finally, the performance of the new algorithm is analyzed. The result shows that the tuning effect is promoted.

Keywords: smart campus; MPP database; automatic database tuning; automatic parameter tuning; multi-node load characterization

1 研究背景

近年来基于大数据分析的人工智能等技术在互联网应用中发展得非常好,应用到了各个方面,从不同角度使人们的生活更便利。实际上依托校园信息系统基础建设,结合云计算、大数据、人工智能、物联网等技术的应用,打通各子系统之间的数据连接共享,也可以帮助高校实现安全高效管理,为管理者、教师及学生带来各种便利及价值的提升^[1]。在此过程中,数据作为分析与挖掘的核心,是智慧校园建设过程中需要重点关注的元素,因此一个设计良好的数据存储系统,是整个建设过程中最基础的工作。在智慧校园数据平台的建设过程中,有些高校可能面临集成异构平台数据结构复杂、业务数据量大且增速快的问题。同时针对

这些异构大量数据的查询,在传统的关系型数据库存储方式中进行一次查询扫描时数据范围和传输数据量是非常大的,导致查询性能的瓶颈。

为了解决以上这些问题,可以选择MPP数据库作为存储搭建智慧校园数据平台。MPP架构将数据和任务并行地分散到多个服务器和节点上,可以解决传统单台数据库存储容量有限的问题。同时列式存储方式可以高效地集成来自异构平台的结构不同的数据,而且列式存储和数据压缩技术可以有效减少数据扫描范围和数据传输时的大小^[2],从而满足海量数据分析的需求。因此采用MPP数据库作为底层存储数据库采集来自不同子系统的业务数据,传统OLTP事务型数据库与Hadoop HDFS作为辅助存储,如图1所示。

【收稿日期】 2021-09-09

【作者简介】 田浩(1991-),男,辽宁沈阳人,辽宁金融职业学院教师,硕士,研究方向:机器学习、大数据分析。

【基金项目】 辽宁金融职业学院2021年度校级课题“基于大数据技术的决策型智慧校园的设计研究”(课题编号:LJXJ202101)。

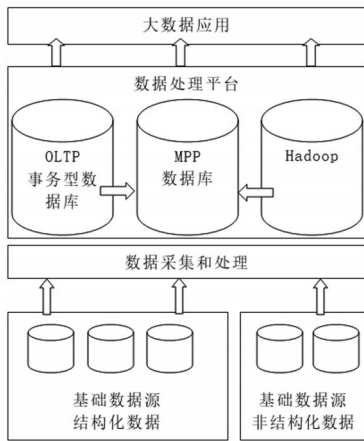


图1 智慧校园数据平台结构图

2 自动调优技术的应用与改进

通过OtterTune技术进行数据库自动调优,首先需要获取数据库的负载情况,并对负载情况进行特征化处理,从而可以匹配知识库中最接近当前情况的负载与参数配置,并根据这些数据计算出接下来的参数配置与性能提升^[3]。但传统OtterTune算法中负载特征化算法的输入为单节点数据库的负载情况向量,在智慧校园数据平台中采用的MPP数据库却是多节点的,因此如何用多节点上不同的负载情况向量进行负载情况刻画是需要解决的问题。为了方便描述,在负载向量中选取延迟和吞吐量这两个比较重要的维度投射到二维平面,图2给出了某一时刻多个节点的负载二维分布情况,从中可以看到大多数节点的负载情况比较接近,但有个别孤立点的负载情况与大多数节点负载情况相差较大,在这种情况下,简单地将所有节点的负载情况进行均值处理后的特征化向量容易受孤立点极值的影响造成偏差,不能很好地描述整个数据库集群的负载情况。最直观的解决方案是去除孤立点后进行均值计算,但数据库节点因为服务器配置与数据分片策略的影响,负载情况动态变化,不能简单地排除某个节点去刻画整体情况。

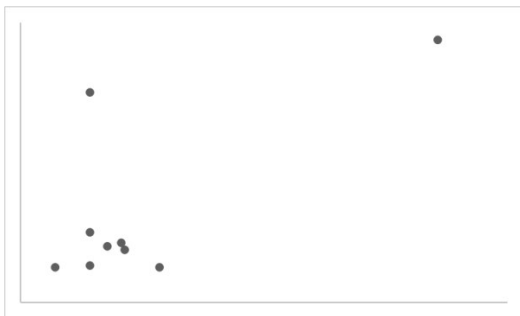


图2 多节点负载二维分布图

2.1 多节点负载刻画

为了更准确地描述MPP数据库某一时刻的负载情况,定义一个Workloads的概念来表述整体的负载情况,Workloads是一个时刻多个节点的负载的集合,可表示为式(1):

$$Workloads = \{ workload_1, workload_2, \dots, workload_n \} \quad (1)$$

其中Workload为每个节点的负载特征向量,可以采用这个向量集合作为初始计算矩阵,通过提出的K-Pick算法计算出合适的负载向量。

2.2 K-Pick算法

通过前文分析知道Workloads集合中的向量存在孤立点,不能通过均值计算得到结果。但是由于数据库节点服务器配置与数据分片策略的影响,负载情况是动态变化的,不能简单地去掉某个节点的负载数据。因此需要一个有效的方式,能够每次在Workloads集合中消除孤立点对最终计算结果的影响。

数据聚类可以将空间中的N个点聚为K个类,并最大化类内对象的相似性,同时最小化类间对象的相似性^[4]。因此可以采用聚类的思想将孤立点从集合中寻找出来。基于密度的聚类算法不需要预先确定聚类数目,能够识别任意形状类簇,并且能较好地识别异常点和噪音点,得到了广泛的应用^[5]。借鉴这种算法思想,可以在集合中选择k个密度最大的向量作为整体的代表,并通过这k个向量来刻画系统整体的负载情况。

对于具有n个向量的集合Workloads中的每个向量都有一个向量密度 ρ_i :

$$\rho_i = \sum_{p=1}^n \lambda(d_{(i,p)} - d_{cut_off}) \quad (2)$$

其中, $\lambda(x) = \begin{cases} 1, & x < 0 \\ 0, & x \geq 0 \end{cases}$, $d_{(i,p)}$ 表示向量i与向量p之间的距离;

d_{cut_off} 是一个超参数,有文献指出其可根据这个参数选择所有向量的距离比它小的比例来确定值^[6]。 ρ_i 相当于以向量i作为中心向量, $d_{(i,p)}$ 作为半径,在向量空间中画一个球体,落在这个球体内的向量的数量。

根据聚类算法的目标不同,一般向量之间的距离有欧氏距离、曼哈顿距离、余弦相似度等^[7]。因为负载向量中不同维度的度量方式不同,数值差异化较大,选择欧氏距离与曼哈顿距离容易导致某个数值过大的维度直接决定距离计算的结果。因此选择余弦相似度作为向量距离的计算方式,如公式3:

$$d_{(i,p)} = \cos \theta_{ip} = \frac{i \times p}{\|i\| \times \|p\|} = \frac{\sum_{i=1}^n (i_i \times p_i)}{\sqrt{\sum_{i=1}^n (i_i)^2} \times \sqrt{\sum_{i=1}^n (p_i)^2}} \quad (3)$$

采用类似基于密度的聚类思想将负载特征向量集合中密度最大的k个向量选择出来,可以更好地刻画系统整体的负载情况,其过程如图3所示算法。算法输入是当前系统负载向量集合,输出是挑选出的k个密度最大的向量的中心向量。算法首先建立负载特征向量空间,之后选取一个适合的超参数,分别计算每个向量的密度,最后选择k个密度最大的向量计算出中心向量,这样可以得到一个OtterTune的输入向量,匹配知识库中的(Knobs, Workload)向量对用以计算推荐的Knobs。

$$\text{Workloads} = \{\text{workload}_1, \text{workload}_2, \dots, \text{workload}_n\}$$

$$\text{Result} = \sum_{i=1}^k \text{workload}_i / k$$

- 1) for j=1 to n
- 2) i = j
- 3) key = ρ_i
- 4) for i to n
- 5) if key > ρ_i
- 6) key = ρ_i
- 7) m = workload_i
- 8) workload_i = workload_j
- 9) workload_j = m

图3 基于密度的K-Pick算法

3 实验分析

为了测试K-Pick算法的多节点负载刻画的效果,将K-Pick算法输出的负载向量作为OtterTune的输入,形成了基于K-Pick算法的K-OtterTune数据库自动调优工具。K-OtterTune架构与原生系统相同,通过K-OtterTune可以计算给出负载情况下推荐的配置参数。

分别采样数据平台几个时刻的负载情况作为测试条件,选取常用的30个配置参数交给数据库运维人员,根据给出的负载情况进行调优,将调优后的参数列表作为标注数据,采用轻量级虚拟化平台docker^[8]构建测试集群,运行OtterTune与K-OtterTune自动调优工具分别获得推荐的配置参数列表作为推荐数据。通过比较OtterTune与K-OtterTune给出的推荐配置参数与标注数据,分析两种工具推荐的参数列表在自动调优上的表现性能。

3.1 构建评价指标体系

Kappa系数是在1960年由Cohen提出的用于评估一致性的一种方法^[9],Kappa系数不仅可以用于评价两个样本的一致性,也可以用于监督学习算法的性能评估。Kappa系数的计算是基于混淆矩阵的,表1给出了实验过程中通用的混淆矩阵。其中横向代表人工标注给出的调整建议,纵向代表自动调优工具给出的推荐调整建议。对于一个具体的配置变量,分别给出调大、调小与保持不变三种调整建议。其中横向与纵向交叉的数字N_i代表样本中同时满足横向建议与纵向建议的样本数量。

表1 混淆矩阵表

推荐标注	Turn up	Turn down	Keep
Turn up	N ₁	N ₂	N ₃
Turn down	N ₄	N ₅	N ₆
Keep	N ₇	N ₈	N ₉

Kappa系数的计算公式为:

$$k = \frac{p_0 - p_e}{1 - p_e} \quad (4)$$

其中p₀表示总体推荐精度,计算方式为:

$$p_0 = \frac{N_1 + N_5 + N_9}{\sum_{i=1}^n N_i} \quad (5)$$

其中N₁、N₅、N₉为推荐与标注建议匹配的类型。

p_e表示推荐建议与标注建议相一致的概率,计算方式为:

$$p_e = \frac{\sum (p_i + \times p + i)}{(\sum_{i=1}^n N_i)^2} \quad (6)$$

其中p_{i+}为第i个推荐与标注建议匹配的类型所在列的像元数目之和,p_{+i}为第i个推荐与标注建议匹配的类型所在行的像元数目之和。

在不同精度的标准评价中,由于评价方法的不同,划分标准和含义也会存在不同。本文主要是将Kappa系数应用在推荐精度的评价中,所以使用通用评价标准,如表2所示。

表2 Kappa系数分类标准表

Kappa	<0	0~0.2	0.21~0.4	0.41~0.6	0.61~0.8	0.81~1
一致性程度	很差	极低	一般	中等	高度	几乎完全一致

3.2 实验对比分析

为了对比K-OtterTune自动调优工具与原生OtterTune给出的调优建议在智慧校园数据平台自动调优的性能表现,分别在6个不同的时段对数据平台的负载情况进行采样,得到标注数据,并将负

载情况分别交给K-OtterTune与OtterTune进行分析,获得推荐数据,一共得到了6组数据。针对每组实验数据中的K-OtterTune与OtterTune推荐数据分别建立混淆矩阵,计算Kappa系数,得到了6组对比结果。如图4所示。

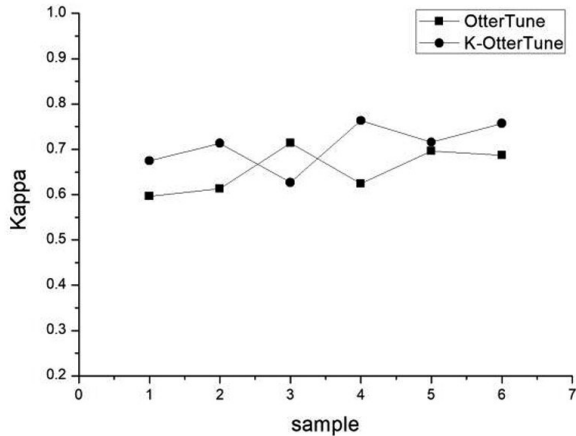


图4 Kappa系数对比图

其中横坐标为采样序列,纵坐标为两组推荐数据与标注数据的Kappa系数。可以发现两种工具给出的推荐配置与标注数据均达到了高度一致的标准,说明自动调优工具可以满足代替人工进行数据库调优的工作要求。同时经过优化后的K-OtterTune给出的推荐普遍比原有的OtterTune给出的推荐与人工调优的修改建议一致性更强,K-Pick算法可以有效地提升自动调优的性能表现。

4 结束语

本文针对智慧校园数据平台建设过程中MPP数据库的自动调优工作,分析了多节点负载难以刻画数据平台整体情况的问题,提出了一种基于密度聚类的多节点负载刻画的方式,能更准确反应系统整体的负载情况。并将其融合进OtterTune自动调优系统中,形成了K-OtterTune,最后通过实验验证了以上方法在智慧校园数据平台MPP数据

库的优化效果。

对于准确的数据库负载信息,可以有效地利用知识库系统匹配相似的负载情况进行参数调优推荐。但在MPP架构数据库下,多个节点的负载情况不能直接用来匹配知识库中的数据集,所以多节点数据库的优化关键在于如何进行负载刻画。本文未来的工作将参考参数空间分级^[10]的方式,研究一种多节点数据库调优的知识库系统,来对MPP架构数据库的优化提供学习模型,然后根据模型推荐的参数进行数据库自动调优。

【参考文献】

- [1]张健.智慧校园基础建设设计要点探析[J].安徽建筑,2021(8):53-54.
- [2]朱凯.ClickHouse原理解析与应用实践[M].北京:机械工业出版社,2020.
- [3]Dana Van Aken, Andrew Pavlo, Geoffrey J. Gordon, Bohan Zhang. Automatic Database Management System Tuning Through Large-scale Machine Learning[P]. Management of Data,2017.
- [4]包志强,赵媛媛,胡啸天,赵研.一种对孤立点不敏感的新的K-Means聚类算法[J].现代电子技术,2020(5):109-112.
- [5]刘颖莹,刘培玉,王智昊,李情情,朱振方.一种基于密度峰值发现的文本聚类算法[J].山东大学学报(理学版),2016(1):65-70.
- [6]张嘉琪,张红云.拐点估计的改进谱聚类算法[J].小型微型计算机系统,2017(5):1049-1053.
- [7]林滨.K-Means聚类的多种距离计算方法的文本实验比较[J].福建工程学院学报,2016(1):80-85.
- [8]谢睿,段盛,于芳.基于Docker的课程实验平台设计与实现[J].邵阳学院学报(自然科学版),2018(4):38-42.
- [9]唐万,胡俊,张晖,吴攀,贺华.Kappa系数:一种衡量评估者间一致性的常用方法[J].上海精神医学,2015(1):62-67.
- [10]刘智.面向多系统的数据库调优[D].深圳:中国科学院大学(中国科学院深圳先进技术研究院),2020.