面向云边端协同的数据库预聚合方法研究

崔双双，马若尧，王宏志
哈尔滨工业大学计算学部，黑龙江 哈尔滨 150001

摘要
云边端协同架构是智能制造、智慧城市等应用的基础，协同计算是其重要支撑技术。在云边端协同架构下，由于数据类型多样且规模庞大，传统数据管理技术难以支持大规模数据的实时查询。为此，提出一种面向云边端协同的数据库预聚合方法，通过物化视图自动生成策略实现数据预聚合，提升实时查询性能。实验结果表明，该预聚合方法使查询时间最多缩短68.45%，显著提升了云边端协同架构下的数据查询性能。

关键词
云边端协同；物化视图；深度强化学习

中图分类号：TP311 文献标志码：A doi：10.11959/j.issn.2096–0271.2025039

Research on database pre-aggregation method for cloud-edge-device collaboration architecture

CUI Shuangshuang, MA Ruoyao, WANG Hongzhi
Faculty of Computing, Harbin Institute of Technology, Harbin 150001, China

Abstract
The cloud-edge-device collaborative architecture serves as the foundation for applications such as smart manufacturing and smart cities, with collaborative computing as a critical supporting technology. Within this architecture, due to the diversity and large scale of data, traditional data management techniques struggle to support real-time querying of large-scale data. To address this, this paper proposes a database pre-aggregation method tailored to cloud-edge-device collaboration, utilizing an automated materialized view generation strategy to achieve data pre-aggregation and improve real-time query performance. Experimental results show that this pre-aggregation method reduces query time by up to 68.45%, significantly enhancing data query performance in cloud-edge-device collaborative architecture.

Key words
cloud-edge-device collaborative, materialized view, deep reinforcement learning

0 引言

在大数据与云计算快速发展的时代，云边端（cloud-edge-device，CED）协同的新计算模式通过整合云计算与边缘计算的优势，助力国家数字经济发展。“协同发展云服务与边缘计算服务”被列入国家“十四五”发展规划[1]。国际数据公司（IDC）也指出，预计到2025年年底，将有超过50%的数据需要在边缘侧进行存储、分析、计算等操作，49%的全球已存储数据将驻留在公共云环境中。云边端一体化旨在提供统一视角资源管理和使用，实现数据自由流通、业务应用统一的运行环境，构建立体化安全保障能力，满足多样化、实时敏捷、安全可靠的业务需求。同时，充分研究云边端协同架构的计算模式，能够为大数据和云治理提供技术基础和实施环境[2]。

云边端协同架构如**图1**所示，在云边端协同架构中，“云”是云计算中心节点；“边”是边缘计算部分，从传统的云计算技术演化发展而来，能够将强计算资源和高效服务下沉到网络边缘端，从而拥有更低的时延、更少的带宽占用、更高的能效和更好的隐私保护性[3]；“端”是终端设备、传感器等。

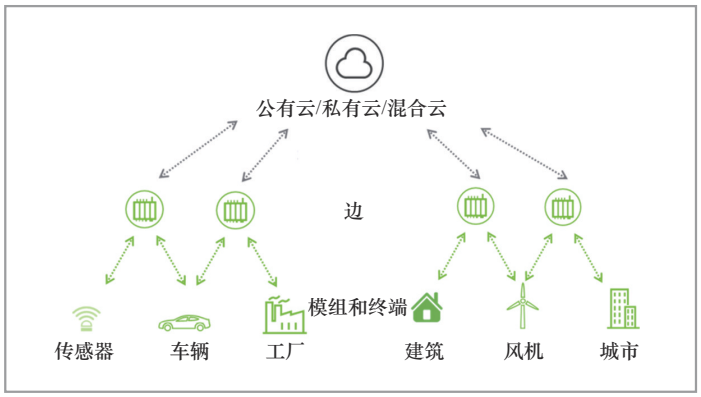

图1 云边端协同架构

云边端协同的新计算模式对数据管理提出了新的要求，需要有效融合云边端的计算能力，构筑云边端数据处理、通信、存储等能力全面协同的数据管理平台[4]。在云边端协同场景下，数据类型多样，工作负载复杂，这为数据查询处理任务带来两方面挑战。一方面，如果直接将所有原始数据上传到云侧，会造成带宽和存储资源的极大浪费；另一方面，实时性的需求又要求快速响应和高效决策。因此，如何在边缘设备或边缘节点上对数据进行初步处理和聚合，成为提升系统效率和性能的关键问题。因此，需要考虑设计预聚合算法来提高聚合查询等操作的性能。本文主要面向云边端协同架构研究数据库预聚合方法，通过物化视图自动生成策略实现数据预聚合。本文的贡献主要有以下两个方面。

- 本文提出了候选物化视图生成器。利用树状结构表示每个SQL查询计划，并且提出候选物化视图生成算法，通过子树合并等方式，得到高频率、高收益的节点，并将这些节点的对应子树作为物化视图的候选，达到从大量子树中快速找到高效子树的目标。
- 本文提出了物化视图（materialized view）自动选择器，提出了基于二分图环境与深度Q网络（deep Q-network，DQN）算法的物化视图自动选择算法，将物化视图选择状态建模为二分图，通过训练智能体，快速选择出应该创建的物化视图，实现数据预聚合，从而降低决策的时间代价。

本文使用公开数据集对提出的数据预聚合方法进行评估。实验结果表明，数据预聚合方法能够使查询时间最多缩短

68.45%，显著提升云边端协同架构下的数据查询性能。

1 数据库预聚合方法研究现状

数据库预聚合可以对一些常用的指标或者计算相对复杂的指标进行提前计算，然后将这些数据存储到新的数据指标中，查询这些计算好的数据将比查询原始的数据更快更便捷。物化视图中的聚合计算可以方便地进行预聚合操作。

物化视图由普通视图发展而来。普通视图可以理解为一张表或多张表的预计算，它可以基于视图创建时指定的查询语句返回的结果集，将需要查询的结果封装成一张虚拟表。当用户查询视图时，数据库会实时执行底层查询语句，将结果集以虚拟表的形式呈现。这种机制本质上是对复杂查询的封装和重用。

视图具有以下特点。

• 安全性：只将需要的结果呈现出来，查询者不知道具体用了哪些表或哪些字段，因此比较安全。

• 屏蔽复杂性：下层计算可能做了很多复杂的关联操作，只需要让开发者将其实现，将结果以视图形式呈现给使用者。

对于普通视图而言，其真实数据在预计算的表中，即每次查询视图都需要执行查询语句。因此，为了防止每次都查询，可以将前述真实数据的结果集存储起来，形成新的视图。一般地，人们称这种新的视图为物化视图。

显然，对合适的候选视图进行物化操作，可以很明显地使用存储数据代价来节省整个数据库的查询时间代价。因此，高效准确地找出在云边端协同环境下适合物化的视图极为关键。物化视图生成是一个已经研究了几十年的重要问题。传统方法利用数据库管理系统（database management system，DBMS）中的优化器来有效地估计物化视图的收益，基于这些收益可以迭代地选择合适的物化视图。因此，在面对新的查询工作负载时，即使物化视图从头开始构建，也能够在一定程度上实现较高的构建效率。这是因为优化器通过估计视图收益，能够快速生成初步的视图方案。然而，完全依赖优化器进行收益估计的方法存在局限性，其估计结果往往不够准确。这种不准确性可能导致生成的物化视图无法全面满足查询需求，从而降低视图的整体质量。

为了解决这个问题，研究者提出了基于学习的方法。如 Kamel 等[5]提出基于神经网络算法创建物化视图来准确估计收益。但是，推导出高质量物化视图的成本极高，因为深度神经结构会导致收益/成本估计缓慢。因此，它们无法满足云边端协同环境下数据类型多样、工作负载复杂且频繁变化的高效要求。到目前为止，很多新的方法被提出以解决这些问题。

在云计算领域，Abdelaziz 等[6]提出了一种提高数据库在云中性能的方法，其使用一种算法来识别发送到数据库的查询列表，并为每个新的复杂的频繁查询添加一个物化视图。

Yao 等[7]提出了一种改进的、有效的物化视图选择算法。该算法通过添加候选物化视图和减少视图来考虑对整体空间和成本的影响，并通过挑选出视图被选择时成本较低的方案来优化候选物化视图集。这些算法可以很好地解决云计算领域的预聚合操作，但仍然无法支撑云边端协同架构环境。

Prakash 等[8]提出了使用多目标遗传算法（multi-objective genetic algorithm，

MOGA）的物化视图选择方法。传统的物化视图选择算法将视图选择问题作为单个目标优化问题来解决，其目标是最小化评估所有视图的总成本。该成本包括两部分，即由物化视图产生的评估总成本和由非物化视图产生的评估总成本。而Prakash等将该问题解释为双目标优化问题，即两个成本同时最小化，并使用MOGA解决。该物化视图选择算法基于MOGA，能够从多维格子中选择Top-K视图，在上述两个目标之间实现最佳权衡。物化这些选定的Top-K视图将减少分析查询的响应时间，从而产生有效和高效的决策。

Han等[9]提出了使用图神经网络进行动态物化视图管理方法。传统方法侧重于静态物化视图管理，即假设不会添加或驱逐物化视图，但在实际场景中，查询工作负载通常会动态变化，由于查询分布可能发生变化，以前主要的物化视图集无法很好地适应未来的工作负载。他们的方法构建了一种新的框架GnnMV，利用图神经网络（graph neural network，GNN）来估计高效的动态物化视图管理的收益。

2 问题定义

问题定义1（物化视图自动生成）给定一组SQL查询$Q=\{q_i\}$，目标是生成一组物化视图$V=\{v_j\}$，使：

（1）物化视图集V的大小在空间预算内；

（2）使用物化视图集V来执行查询集Q中查询的性能是最优的。

物化视图自动生成问题是利用公共子查询发现的，获得候选物化视图，在候选物化视图集合中选取最优的物化视图。于是可以进一步细化问题描述，得到如下问题定义。

问题定义2（基于候选物化视图的物化视图自动生成）给定一组SQL查询$Q=\{q_i\}$和一组候选物化视图$V=\{v_j\}$，目标是从V中筛选出一个物化视图子集MV=$\{\mathrm{mv}_k\}$，其中对任意的k都有$\{\mathrm{mv}_k\}\in V$，使：

（1）物化视图子集MV的空间代价不超过预算限制；

（2）使用物化视图子集MV来执行查询集Q中查询的性能是最优的。

从上述定义可以看出，物化视图自动生成问题是背包问题的一个变种[10]。如果将mv_k看作一个物体，将其执行查询集后的性能收益看作每个物体的权重，由于每个候选物化视图能够带来的收益是未知且变化的，物化视图自动生成问题比背包问题更加困难。因此，物化视图自动生成问题至少是NP难问题。本文使用深度强化学习模型，借助神经网络来估计选取一个候选视图建立物化视图的价值，进一步求解最优的物化视图子集MV。

3 面向云边端协同的数据库预聚合

面向云边端协同的数据库预聚合方法主要包括两部分：候选视图生成器和物化视图选择器，如**图2**所示。候选视图生成器中，左侧框所示为合并公共子树的过程；右侧框所示为二分图结构，框中实线和虚线表示SQL查询可能对应的视图，可用于后续训练。

3.1 候选视图生成器

在云边端协同环境中，候选视图生成需要根据云边端的不同特点，动态调整生成策略，以适应不同层次的计算和存储能

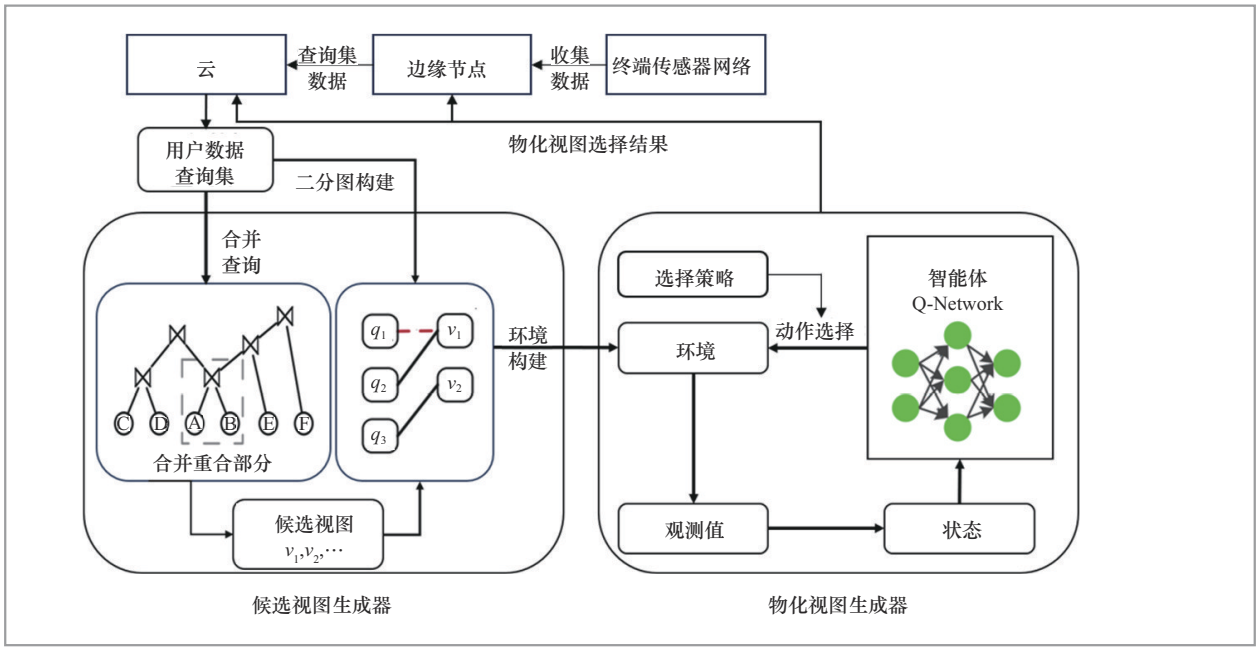

图2 面向云边端协同的数据库预聚合

力。从架构原理来看，云侧凭借其强大的计算能力和海量存储空间，需要承担全局候选视图的构建和维护任务。这种集中式的处理模式能够充分利用分布式计算框架的并行处理能力，对来自边缘侧的海量查询数据进行深度分析和模式挖掘，从而生成具有全局优化价值的候选视图。同时，云侧的大容量存储特性也为长期保存历史视图数据提供了可靠保障，能够支持跨时间维度的查询优化。而边侧和端侧靠近数据源，负责为云侧收集数据，记录并上传查询集。

物化视图自动生成的第一步是获取工作负载中的重要候选视图。为了定量描述候选物化视图转化为物化视图的可能性，候选视图生成器会计算每个子查询的权重。接下来，如何表示子查询才能快速找到候选物化视图是物化视图自动生成的关键。本文将每个SQL查询表示为树状结构查询计划，并查找高效子树。由于查询计划中的子树是有效的潜在公共子查询，因此可以选择子树并将其作为候选视图。然而，表的连接顺序不同，每个查询可能具有指数级的候选查询计划，这导致了大量不同的子树。

为了从大量子树中快速找到高效子树，本文提出了候选视图生成算法。核心思想是对于每个查询，首先，从优化器中提取树状结构的物理查询计划。然后，检测公共子树，其中查询计划树中的两个节点（子树）是等效的，如果这两个节点具有相似的连接/选择/投影条件，并且它们的子节点是等效的，那么可以合并两个节点。接下来，计算每个子树的收益，它是包含公共子树的查询数和估计收益的乘积。最后，获取最有益的子树作为物化视图候选者。下面详细介绍4个步骤。

首先，候选物化视图生成器识别出可以合并的叶子节点，并执行叶子节点的合并。这些叶子节点包含相同的关系表，并

且它们的选择和投影条件也相同。合并这些叶子节点意味着只需要执行一次对该表的访问，就可以满足多个原始查询的需求。这种合并减少了不必要的表访问操作，从而提高了查询性能。

然后，遍历查询计划树，在合并了叶子节点之后，系统进一步向上遍历查询计划树，寻找可以合并的等效子树。等效子树指的是两个具有相同选择和投影条件的子树，并且它们的子节点也是等效的。也就是说，对于第一个子树的每个子节点，都可以在第二个子树中找到一个等效的子节点。合并这些等效子树，可以进一步减少重复计算。例如，如果两个查询都要对某个中间结果进行相同的操作，并且这两个操作的条件也相同，那么这两个操作就可以合并为一个操作。这种合并减少了对CPU和内存的使用，从而提高系统的整体性能。

在合并了等效子树之后，系统会计算每个节点的频率、开销和收益。计算节点频率时，节点的频率是指合并到该节点中的查询数量。也就是说，如果一个节点被多个查询共享，那么它的频率就会很高。具有更高查询频率的子节点通常意味着它们对性能的影响更大，因此更适合作为物化视图的候选。计算节点开销时，如果一个子查询由查询工作负载中的多个查询共享，那么可以物化此子查询的视图，以避免冗余计算。物化子查询视图的开销为物化结果的空间开销，即给定一个基于子查询构建的物化视图，该物化视图的存储大小表示为 W。如果存储一个字节的费用为 α，则存储该物化视图的开销为 $W\times\alpha$。计算节点收益时，使用物化视图来回答SQL查询具有非常显著的收益，因为可以直接从物化视图中获取SQL查询的部分子查询的结果，并避免重新执行这部分子查询。具体收益可以通过使用/不使用物化视图的查询时间成本的差额来计算，通过在实际数据库中运行使用/不使用物化视图的查询，获取查询时间并计算差值，生成节点的收益。

最后，通过分析上述理论，系统会选择那些具有高频率、高收益的节点，并将这些节点的对应子树作为物化视图的候选。候选视图生成算法如算法1所示。

算法1：候选视图生成

输入：SQL查询集 Q

输出：候选视图集 S、合并节点对应的频率 f、开销 W 与收益 R

1. 初始化 S 为空表；
2. 对于查询集 Q 中的每一个查询 q_k；
3. 初始化待合并队列Queue；
4. 获取 q_k 的查询计划图 T，并将其加入 S；
5. 对于分别在 T 与 S 中具有相同结构的两个叶子节点，将该（T, S）节点组加入队列Queue中；
6. 如果Queue非空：
7. 考察节点组是否具有相同的查询结构；
8. 如果相同则将其父节点的节点组加入队列，并进行重复考察；
9. 不同则直接跳出循环，记录节点的合并频率 f，开销 W 与收益 R；
10. 排序后输出候选视图集 S 和每个合并节点对应的频率 f、开销 W 与收益 R。

该算法的第1行实现候选视图集的初始化过程。第2~3行针对查询集中的每个查询，初始化待合并队列，并提取查询计划图。第4~6行遍历查询计划图与候选视

图集中的叶子节点，识别结构相同的节点，并将其加入待合并队列。第7~9行通过迭代方式考察节点组的父节点，发现不同的父节点结构时，计算当前节点的合并频率、开销和收益，并将其记录为候选视图。第10行对所有记录进行排序后，输出候选视图集及每个节点的相关属性。

3.2 物化视图选择器

在云边端协同架构中，物化视图自动选择器通过分层协作机制实现物化视图的最优选择。数据通过端侧的传感器网络实时采集和存储，边缘节点对数据和查询进行分布式处理与上传，为物化视图选择提供基础支持。云端凭借其全局视角和强大的计算能力，基于候选视图集，通过物化视图选择器筛选出最优的物化视图集，并进行长期维护与更新，以满足全局查询需求。边缘侧则优先选择能够加速本地查询的物化视图，通过本地化优化减少查询延迟，提升区域性查询效率。这种分层协同的选择机制既保证了全局查询性能的最优化，又兼顾了边缘侧实时性和高效性需求，显著提升了整个云边端协同系统的计算效率与资源利用率。

具体来说，云端物化视图选择器通过分析全局查询模式和历史访问规律，结合成本模型（如存储成本、计算成本、更新代价等），从候选视图集中选择出最具价值的物化视图。这些视图能够通过分布式存储和索引机制来确保高效访问。边缘侧则根据本地查询负载和区域性数据特征，动态选择能够最大化本地查询性能的物化视图，并通过缓存机制和增量更新策略保持视图的时效性。

通过候选物化视图生成器得到候选物化视图集合后，需要得到每个候选物化视图的计算开销、存储开销，以及作为物化视图候选的总开销，之后才能获取每个物化视图的收益，进而确定是否应该利用该物化视图。

在计算物化视图集合的总收益时，需要评估每个物化视图候选者对整体查询集的时间收益情况。如果逐个计算所有候选视图的时间收益，计算的时间代价将会非常高。这是因为用户数据库中通常包含大量的查询，同时对应着大量的候选物化视图。因此，为了求解物化视图选择问题的计算成本非常高的问题，本文设计了物化视图选择器，提出基于二分图环境与DQN算法的物化视图自动选择算法，以降低决策的时间代价。

DQN模型主要由6个部分组成：环境、状态、智能体、奖励、行动和动作选择策略。

环境：将物化视图选择状态建模为二分图，其中左侧的节点表示查询集，右侧的节点表示候选视图集，它们之间的边表示是否进行物化操作，环境模块为智能体提供观察结果，例如二分图的边选择状态和总收益。将物化视图选择状态建模为二分图的过程如**图3**所示。左侧的1a、1b等是SQL查询的编号，右侧1、2等是候选视图的编号。

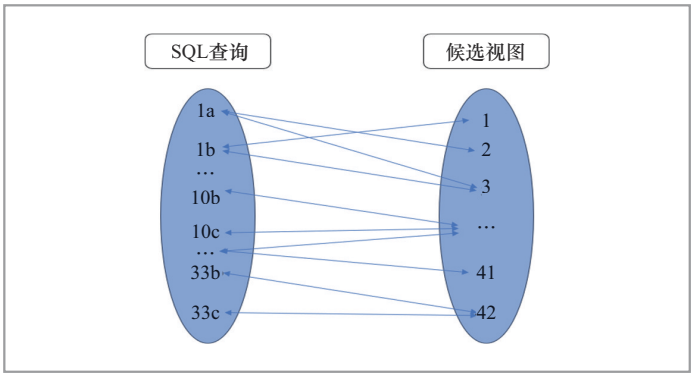

图3 二分图建模过程

状态：通过获取观测值计算当前的状态。

智能体：它由两个神经网络和经验回放机制组成。这两个神经网络可以看作近似于动作价值函数，即近似的Q表，由4层全连接层构成。该函数的值等于最大反馈G。通过正确的奖励设置，令G与最终总收益呈正相关。

奖励：要使G与最终总收益呈正相关，可以将奖励定义为每次行动后总收益的变化，该模型将尽快实现最优解。候选视图的收益为是否使用该物化视图优化对应的SQL查询的查询时间的差值，而总收益为全部差值之和。

行动：在每次迭代中，环境都会检测二分图的边，即是否用某个候选视图去优化模型，并询问智能体是否使用此边：

- 如果答案为是，那么此边将被设置为1，即将此边置于被物化的选择状态；
- 如果答案为否，那么此边将被设置为0，即从全局选择状态中删除此边。

动作选择策略：智能体根据获得更高反馈的策略行事，智能体选择具有最高反馈的操作。

基于二分图环境与DQN算法的物化视图自动选择算法如算法2所示。

算法2 基于二分图环境与DQN算法的物化视图自动选择

输入：候选视图集U，查询集SQLs，收益衰减参数γ

输出：训练好的DQN模型

1. 初始化经验池为一个容量为N的队列D，初始化Q网络和Target Q网络；
2. 根据候选视图集U，查询集SQLs，初始化环境二分图（状态空间）S；
3. 获取动作空间A；
4. 对于每个学习过程：
5. 根据状态空间S获取初始状态S_0，并预处理为状态s_0；
6. 重置总收益R为0；
7. 对于到达最终状态前的每一步决策t：
8. 以ε-贪婪选择策略从动作空间A中选择出动作a，即根据ε-贪婪选择策略选择二分图中收益最高的边；
9. 执行动作a，观测并获取下一步的状态，计算动作a带来的收益r，并计算经过衰减后的总收益$R=R+\gamma r$；
10. 将状态转移过程的经验样本存储到经验池D中；
11. 经验池满，则随机从D中提取状态转移过程样本数据，计算Q网络值，并执行梯度下降策略并更新网络参数；
12. 每经过C步，将Q网络参数复制给Target Q网络；
13. 完成全部episode，结束训练。

该算法的第1~3行实现初始化Q值网络和经验库，构建智能体运行环境；第4~6行初始化学习过程；第7~9行进行环境观测并生成动作；第10~12行实现经验数据存储和Q网络学习；第13行完成训练，并输出DQN模型。

对于环境建模的二分图的每一个边，显然边的选择状态的全部集合构成整个动作空间，而左侧节点-边-右侧节点的结构表示使用右侧节点代表的物化视图去优化左侧节点代表的SQL查询。因此，在每个学习过程中，选择某一个边的收益就是用右侧节点的物化视图去优化左侧节点的

SQL查询的查询时间优化值，即在数据库中直接查询该SQL语句的运行时间与使用物化视图后的查询时间的差值。

同时，考虑到当每一个候选视图都被物化时，全部SQL查询的运行时间显然是最短的。事实上，当选择全部的候选视图时，SQL查询时间可以被优化到极低的水平，但实际的数据库管理系统的可用存储空间不可能是无限大的，物化视图的生成需要实际的数据库存储视图查询的结果。对于云边端协同环境下的大量且复杂的数据集和查询集，选择奖励不高而空间需求极大的候选视图作为物化视图显然不可行。

因此，在获取每一个边的查询时间收益的同时，还要获取右侧候选视图成为物化视图的空间代价，以保证本文提出的DQN模型选择的物化视图集的存储空间保持在某一合理的特定值内。

在训练过程中，当模型面对连续样本的学习任务，尤其是当这些样本之间存在非常强的相关性时，直接从中学习往往会遇到效率问题。这种强相关性意味着，如果模型在某一时间点的决策或学习基于一个特定的状态转移过程，那么这个模型的学习过程很可能在接下来的一段时间内都会受到这个状态转移过程的强烈影响，因为后续的状态转移过程很可能与当前过程高度相似或相关。

这种情况在学习算法中并不理想，因为它可能导致模型对某个局部区域的数据产生过度拟合，而忽视了整体数据分布中的其他重要信息。此外，当样本之间的相关性很高时，模型的学习过程可能会变得不稳定，因为一个小的变化就可能导致模型参数的显著变化，从而增加学习过程中的方差。

解决该问题的一种常用方法是随机化样本。将样本进行随机排列可以破坏原有样本之间的相关性，使模型在每次更新时都能接触到独立的样本。这种方法可以减少学习过程中的方差，提高学习效率和稳定性。但仅仅随机化样本并不总是足够的。为了更好地利用历史数据来指导学习过程，可使用经验回放（experience replay）机制。

经验回放机制是一种在学习过程中存储和重新使用历史经验的策略。通过将这些历史经验存储在一个缓冲区中，DQN模型可以在需要时从中随机抽取样本进行训练。这种方法的好处是，它允许模型在不同的时间点之间完成学习过程，因为每次学习都可以基于来自不同时间点的随机样本进行。使用经验回放可以平滑学习过程，降低参数的振荡或发散，帮助DQN模型更稳定、更有效地从连续样本中学习。

经验记忆中存储智能体观测到的状态转移过程，DQN模型可以在后续重复使用该数据，减少连续的状态转移的几个数据之间的相关性。具体的操作为把状态转移结果transition (s_t, a_t, r_t, s_{t+1})存入一个经验回放缓冲区（buffer）中，每当经验记忆池到达buffer上限，DQN模型就会在记忆中均匀抽样出一个批（batch），进行随机梯度下降训练。

同时，考虑到大量数据导致的大量状态转移过程，仅仅选择当前状态的最优动作无法保证对所有动作价值空间都能至少一次被访问，因此采用ε-贪婪选择策略。

4 实验分析

4.1 实验设置

（1）数据集

本节采用IMDB（Internet movie database）进行实验，该数据集是一个包

含被广泛使用的电影、电视节目和演员信息的数据集，它包括电影、电视节目、演员、制作公司、编剧、导演等信息。IMDB可以为电影评论、分类、预测以及其他机器学习任务提供有用的参考信息。JOB（Join Order Benchmark）查询集是一个IMDB数据库的基准测试，旨在评估数据库优化器的能力。

（2）实验环境

本文提出的预聚合算法是面向云边端协同架构的，实际实验中用云服务器构建云端环境，用本地主机模拟边缘环境。由于实际数据已经在数据集中，故不再单独设置用于收集数据的端侧环境。因此，实验环境分为本地环境和云环境两部分。实验环境见**表1**。

（3）评价指标

为了评估物化视图给查询性能带来的效果，本文使用统一评价指标：物化视图生成前后查询集的运行时间。这一指标能够直接观测到物化视图在减少查询时间、提升查询性能方面的实际效果。这一评价指标能够准确反映物化视图技术的性能优势。

表1 实验环境

软件/硬件/模块名称	版本/参数
CPU	Intel(R) Core(TM) i7-10875H
RAM	16.0 GB
硬盘	512 GB
操作系统	Windows 10
编码工具	Pycharm 2022
GPU	CUDA 12.1/cuDNN 8.9.1
Numpy	1.26.0
Pytorch	2.0.1
数据库	PostgreSQL 16.2
华为云耀服务器	2核|2GiB
云数据库	PostgreSQL 16.2

4.2 DQN模型评估

对IMDB数据集的数据进行观测，并调整DQN模型的超参数，最终建立了DQN模型，该网络由输入层、两个全连接层和中间的隐藏层组成，采用Adam优化器。由于本实验总体数据量限制，batch_size可以设置为256，即经验库记录256次状态转移过程后开始学习。贪婪选择参数epsilon初始值设置为0.3，保证DQN模型在刚开始训练时能够探索更全面的动作空间，同时令该数值在训练过程中逐渐降低，防止模型参数在进行过多的随机选择时产生波动。

对DQN模型进行测试，使用IMDB数据集的部分SQL查询语句作为输入，确定查询树结构中的查询优化奖励为输出，模型约在350个episode后训练结果达到稳定。本实验实现的DQN模型最重要的参数为学习率，该参数影响Q值网络更新时对一个新的状态转移过程的收益的学习程度。

设置不同的学习率进行对比测试，分别令学习率的初始值为0.5、0.01、0.003、0.00001，折扣因子为0.5，batch_size为256，ε-贪婪选择参数epsilon为0.3，获取学习率曲线如**图4**所示，横坐标为训练的组数episode（由于输入数据较少，所需训练轮次较多），纵坐标为选择节点获得的总收益。

当学习率设置得过高（如实验中设置的0.5）时，模型在每次更新时会进行大幅度的权重调整，这会导致训练过程不稳定，表现为收益曲线的剧烈波动，甚至可能导致模型无法收敛到最优解，而是在解空间内波动。

相反，当学习率设置得过低（如实验中设置的0.00001）时，可以看出，低学习率下的收益曲线几乎没有明显的上升趋势，这表明模型的学习效果不佳。模型在

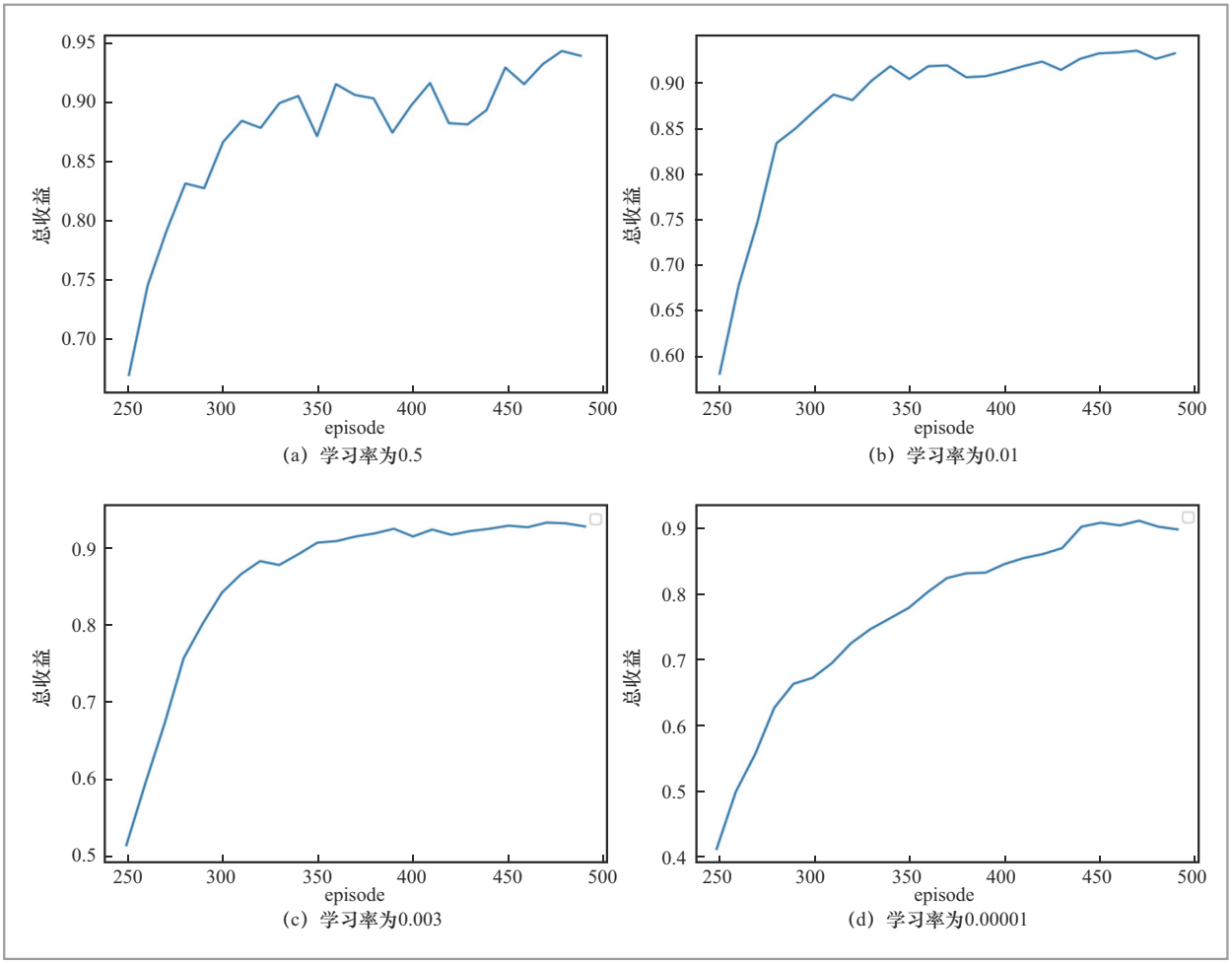

图4 不同学习率下的DQN学习曲线

每次更新时只会进行极其微小的权重调整，这虽然增加了训练的稳定性，但也可能导致训练过程变得非常缓慢，甚至无法有效地学习到有用的信息。

经过对比不同的学习率，最终发现当学习率设置为0.003时，模型能够取得最好的性能。这个学习率既能保证训练过程的稳定性，又能使模型在合理的时间内收敛到较优的解。

可以看出，随着学习过程的推进，生成的深度强化学习DQN模型的总体收益水平不断提高，直到收益达到稳定，该模型能够胜任获取最优的物化视图查询处理代价方面的工作。

4.3 预聚合效果评估

进行评估实验时，训练好的DQN模型将根据给定的SQL查询集，生成一系列的候选物化视图，并记录每个物化视图的生成时间和占用的存储空间，与原始查询集的运行时间进行比较。

对于JOB查询集的全部SQL查询语句，为了找到高收益的公共子树，该模块将工作负载中的所有查询计划合并到多视图处理计划中，合并所有查询计划树。因

此，可以将具有相同结构的等效子树合并为一个子树，以便找到高公用频率的公共子树。然后，将根位于该节点的相应子树提取出来作为物化视图的候选视图。

使用DQN模型对生成的候选视图进行处理，并与原有的全部SQL查询的运行时间进行对比。在物化视图的实际使用中，直接将全部的查询结果都进行物化操作显然可以将查询集的时间代价降到最低，但多出的空间代价是巨大且无法接受的，因此在采样时考虑设定1 MB、100 MB、200 MB、400 MB和600 MB的空间限制，对比物化视图生成前的结果。

首先，对JOB查询集进行处理，获取物化视图候选集（部分）如**图5**所示。

以其中的一个候选视图为例，其结构为：[['10a.sql', '10b.sql', '10c.sql', '15a.sql', '15b.sql', '15c.sql', '15d.sql'], [' AND cn.id = mc.company_id\n', ' AND ct.id = mc.company_type_id; \n']]。

这表示10a.sql、10b.sql、10c.sql、15a.sql、15b.sql、15c.sql、15d.sql这7个SQL查询都有的形式为：cn.id = mc.company_id AND ct.id = mc.company_type_id的cn、mc和ct 3个关系表连接对应的公共子查询。

因此，可以将全部候选视图和对应的SQL查询编码为二分图，获取其对应的状态空间和动作空间，完善DQN模型的参数，例如，第一个候选视图对应10a.sql、10b.sql、10c.sql这3个SQL查询，而第二个候选视图对应10a.sql、10b.sql、10c.sql、15a.sql、15b.sql、15c.sql、15d.sql这7个SQL查询。

获取候选视图集时，为每个候选视图提供对应的收益和空间代价，训练并使用DQN模型，分别部署在本地和云环境中，对候选视图集进行选择，获取最终推荐的物化视图集。使用被选择的候选视图进行测试，为方便进行查询优化效果，考察全部113个SQL查询中前20个查询（包括1a~1d、2a~2d、3a~3c、4a~4c、5a~5c、6a~6c查询），使用物化视图前后的具体结果对比如**图6**所示。可以看出，1a~1d与5a~5c整组的查询时间都没有明显变化，因为在DQN模型选择的物化视图集中，并没有一个物化视图能对该组SQL查询进行优化，因此查询时间不变，推测可能的原因是优化收益低或者占用的存储空间不足以接受该组的查询被优化；而2a组的查询时间远低于原有查询，其物化视图的根节点靠近原查询的根节点，几乎完全存储查询结果来对原查询进行优化，因此2a组的查询优化效果极好；3a、4a、6a组的查询时间有一定优化，其物化视图能够存储查询的一部分信息，简化查询过程。

接下来，使用IMDB/JOB数据集的全部查询，对整体查询集进行物化视图候选集的生成和物化视图选择操作。将输出的物化视图的文本数据转换为能被数据库系

```
[['10a.sql', '10b.sql', '10c.sql'], [' AND t.id = mc.movie_id\n', ' AND t.id = ci.movie_id\n', ' AND ci.movie_id = mc.movie_id\n', ' ANI
[['10a.sql', '10b.sql', '10c.sql', '15a.sql', '15b.sql', '15c.sql', '15d.sql'], [' AND cn.id = mc.company_id\n', ' AND ct.id = mc.company_ty
[['11a.sql', '11b.sql', '11c.sql', '11d.sql'], [' AND lt.id = ml.link_type_id\n', ' AND ml.movie_id = t.id\n', ' AND t.id = mk.movie_id\n', '
[['12a.sql', '12b.sql', '12c.sql'], [' AND t.id = mi.movie_id\n', ' AND t.id = mi_idx.movie_id\n', ' AND mi.info_type_id = it1.id\n', ' AN
[['13a.sql', '13d.sql'], [" AND ct.kind ='production companies'\n", " AND it.info ='rating'\n", " AND it2.info ='release dates'\n", " A
[['13a.sql', '13b.sql', '13c.sql', '13d.sql'], [' AND mi.movie_id = t.id\n', ' AND it2.id = mi.info_type_id\n', ' AND kt.id = t.kind_id\n', '
[['14a.sql', '14b.sql', '14c.sql'], [' AND kt.id = t.kind_id\n', ' AND t.id = mi.movie_id\n', ' AND t.id = mk.movie_id\n', ' AND t.id = m
[['14a.sql', '14b.sql', '14c.sql', '18a.sql', '18b.sql', '18c.sql'], [' AND it1.id = mi.info_type_id\n', ' AND it2.id = mi_idx.info_type_id;\n']]
```

图5 候选视图集(部分)

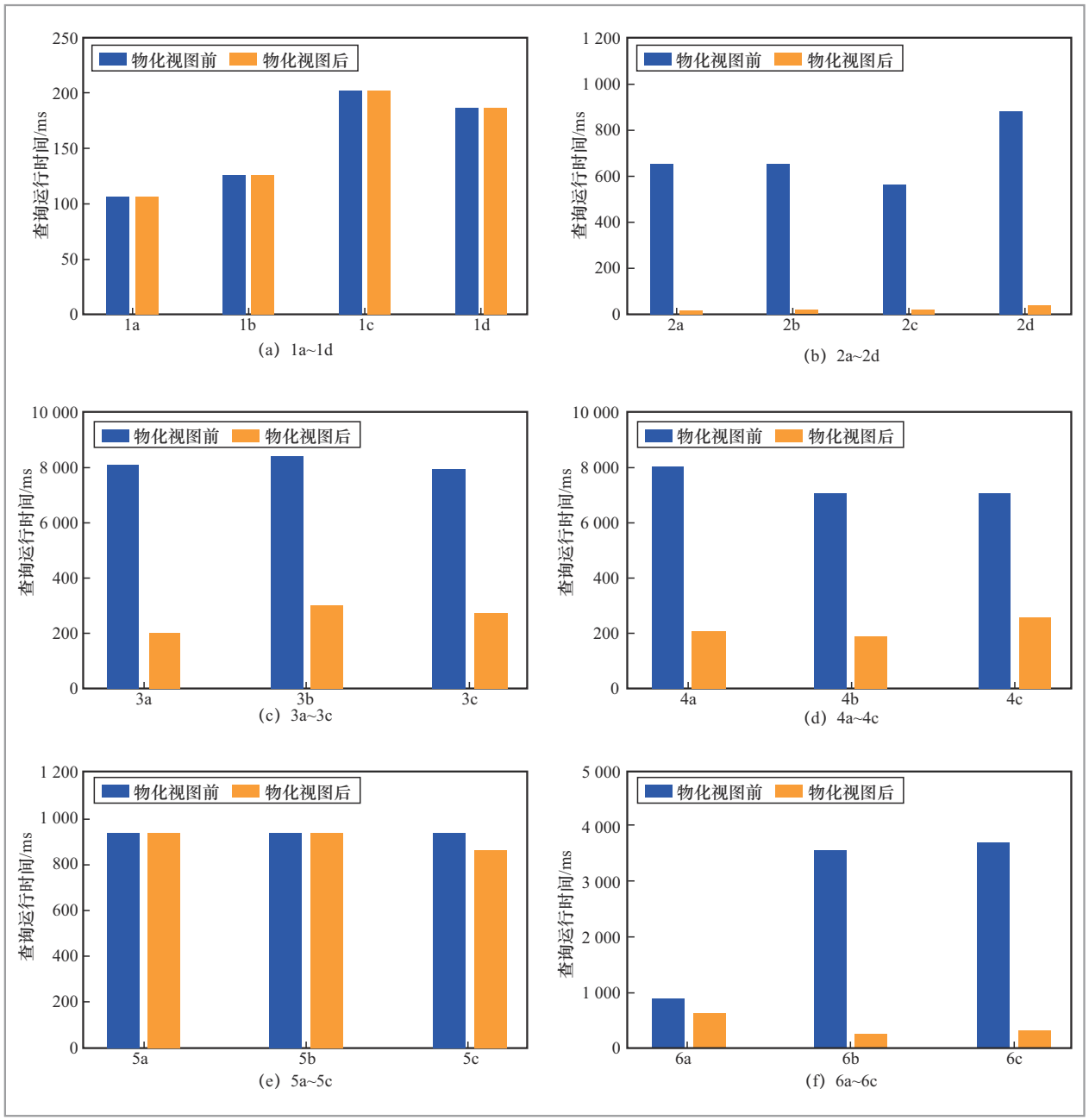

图6 前20个查询的具体运行时间对比

统识别的物化视图生成语句，使用PostgreSQL数据库系统的EXPLAIN ANALYZE命令测试全部SQL语句的查询时间并加和。具体结果见**表2**。

对于实际生成的物化视图的效果，JOB查询集的查询时间在空间限制为200 MB以下的部分变化明显，因为JOB查询集内部的复杂性，它包含了多个不同的SQL查询组（如2a~2d为一个SQL查询组），而大部分候选视图的空间需求不高，导致200 MB以下的部分中随着空间限制的增长，可选择的物化视图同步增多，总体收

表2 物化视图生成结果

查询条件	优化前时间/s	JOB查询集总时间/s	优化率
不生成物化视图	514.304	514.304	0
1 MB空间限制	514.304	427.940	16.79%
100 MB空间限制	514.304	330.862	35.67%
200 MB空间限制	514.304	202.358	60.65%
400 MB空间限制	514.304	186.459	63.75%
600 MB空间限制	514.304	162.248	68.45%

益变化明显。实验结果的对比如图7所示。

对于JOB查询集中的不同查询，DQN模型表现出高度的灵活性和适应性。对于运行时间较短和所需存储数据量较小的查询，DQN模型能够准确地为其添加物化视图，从而在增加存储空间负担较小的情况下显著提升查询性能。然而，对于运行时间较长和所需存储数据量很大的查询，DQN模型则需要更加谨慎地处理，即虽然为这些查询添加物化视图能够显著减少其执行时间，但所需的存储空间也会成倍增加。

因此，JOB查询集的不同组SQL查询的运行时间和调用的数据有极大差别。同时，JOB查询集在允许大于600 MB空间限制后的部分，总的时间成本奖励没有明显提高，而空间成本却直线上升，因此JOB查询集只需要考虑600 MB空间限制以内的部分。根据表2的数据，600 MB空间限制下进行的物化视图自动选择结果为162.248 s，与原SQL查询集的全部查询时间514.304 s相比，查询时间优化了68.45%。因此，在考虑合理的空间限制的情况下，本模型的查询优化性能可以满足大部分场景下的用户。

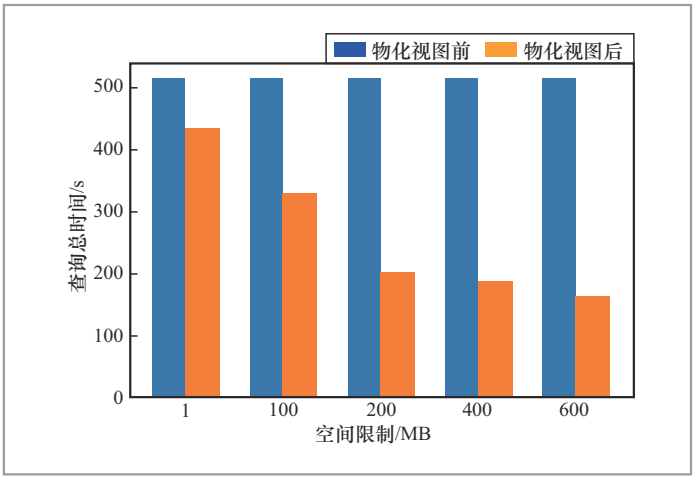

图7 查询时间对比

对于在云平台上的IMDB/JOB数据库和训练的DQN模型，其生产的物化视图集能得出很好的优化效果，因此本文提出的预聚合方法适用于简单的云边端协同环境，能够满足使用云边端架构环境的用户的基本需求。

5 结束语

本文提出了一种面向云边端协同架构的数据库预聚合方法，通过物化视图自动生成策略实现数据预聚合；使用深度强化学习模型对物化视图的候选视图进行处理，综合考察候选视图物化后对整体查询集的

收益，最终选择出对优化该查询集性能优化程度最高的物化视图集。未来，可以通过对候选视图集生成模块进行优化，减少候选视图生成时带来的极为巨大的运算量，同时提高生成的准确率，以提升效果和性能。

参考文献：

[1] 崔双双，吴限，王宏志，等. 面向云边端协同的多模态数据建模技术及其应用[J]. 软件学报，2024，35(3)：1154-1172.
CUI S S，WU X，WANG H Z，et al. Multimodal data modeling technology and its application for cloud-edge-device collaboration[J]. Journal of Software，2024，35(3)：1154-1172.

[2] 曾琛. “云治理”设想[J]. 大数据，2017，3(3)：3-14.
ZENG C. A framework of“cloud governance”system based on blockchain technology[J]. Big Data Research，2017，3(3)：3-14.

[3] 周知，于帅，陈旭. 边缘智能：边缘计算与人工智能融合的新范式[J]. 大数据，2019，5(2)：53-63.
ZHOU Z，YU S，CHEN X. Edge intelligence：a new nexus of edge computing and artificial intelligence[J]. Big Data Research，2019，5(2)：53-63.

[4] 朱锐，王宏志，崔双双，等. 面向元宇宙的云边端协同大数据管理[J]. 大数据，2023，9(1)：63-77.
ZHU R，WANG H Z，CUI S S，et al. Cloud-edge-end collaborative big data management for metaverse[J]. Big Data Research，2023，9(1)：63-77.

[5] KAMEL A，EZZEDINE T. Creation of materialized views based on neural network algorithm[C]//Proceedings of the 2018 14th International Wireless Communications & Mobile Computing Conference. Piscataway：IEEE Press，2018：1542-1547.

[6] ABDELAZIZ E，MOHAMED O. Temporary materialized views in cloud data warehouses through a web service[C]//Proceedings of the 2017 3rd International Conference of Cloud Computing Technologies and Applications. Piscataway：IEEE Press，2017：1-6.

[7] YAO D F，ABULIZI A，HOU R K. An improved algorithm of materialized view selection within the confinement of space[C]//Proceedings of the 2015 IEEE Fifth International Conference on Big Data and Cloud Computing. Piscataway：IEEE Press，2015：310-313.

[8] PRAKASH J，VIJAY KUMAR T V. Multi-objective materialized view selection using MOGA[J]. International Journal of System Assurance Engineering and Management，2020，11(2)：220-231.

[9] HAN Y，CHAI C，LIU J，et al. Dynamic materialized view management using graph neural network[C]//Proceedings of 2023 39th IEEE International Conference on Data Engineering. California：ICDE，2023.

[10] KAMEL A，EZZEDINE T. Dynamic selection of indexes and views materialize with algorithm Knapsack[C]//Proceedings of the 2019 International Conference on Internet of Things，Embedded Systems and Communications. Piscataway：IEEE Press，2019：214-219.

作者简介

崔双双（1997– ），女，哈尔滨工业大学计算学部博士生，中国计算机学会（CCF）学生会员，以第一作者发表CCF–A类国际会议论文1篇、EI检索论文5篇，主要研究方向为云边端协同数据库存储、AI4DB。

马若尧（2002– ），男，哈尔滨工业大学计算学部硕士生，主要研究方向为云边端协同数据库。

王宏志（1978– ），男，哈尔滨工业大学计算学部长聘教授、博士生导师，计算机科学与工程系主任，海量数据计算研究中心主任，数据科学与大数据技术专业负责人，美国加利福尼亚大学欧文分校博士后，微软亚洲研究院铸星计划访问学者，青年龙江学者。主要研究方向为大数据治理、大数据管理与分析、数据库系统、工业大数据等。在国内外重要会议和期刊发表学术论文300余篇，出版学术专著4本。论文被SCI收录百余次，他引4 000余次，授权发明专利32项。获微软学者、中国优秀数据库工程师、IBM博士英才等称号，获黑龙江省自然科学奖一等奖和教育部高等学校科技进步奖一等奖，获黑龙江省青年科技奖、宝钢优秀教师奖、CSC–IBM奖教金。主持国家自然科学基金重点项目、国家科技支撑计划课题、国家博士后特别资助项目等10余个项目，主讲的课程获批国家一流线上课程。CCF杰出会员，ACM SIGMOD中国秘书长、教育部高等学校计算机类专业教学指导委员会计算机系统专家委员会委员、CCF数据库专业委员会常务委员、CCF大数据专家委员会委员、哈尔滨工业大学计算学部校友会秘书长、ACM数据科学学科标准编写组专家、黑龙江省“头雁团队”成员。

收稿日期: 2025–02–24

通信作者: 王宏志，wangzh@hit.edu.cn

基金项目: 国家自然科学基金项目（No.62232005，No.62202126）；国家重点研发计划项目（No.2020YFB1006104）

Foundation Items: The National Natural Science Foundation of China (No.62232005, No.62202126), The National Key Research and Development Program of China (No.2020YFB1006104)