哈爾濱工業大學

本科毕业论文（设计）开题报告

题 目：面向公平性的云边微服务系统部署方法

专　　业　软件工程

学　　生　付书煜

学　　号　2021111824

指导教师　贺祥

日　　期　2024 年 11 月 20 日

哈尔滨工业大学教务处制

目 录

1 课题来源及研究的目的和意义 1
2 国内外在该方向的研究现状及分析 2
3 主要研究内容 3
3.1 视频流服务的云-边协同架构的系统建模 4
3.2 JAIN 公平性指数导向的公平性优化算法 4
3.2.1 优化目标及约束条件 4
3.2.2 优化算法设计概述 4
3.3 模型公平性优化效果的评估标准 4
3.3.1 评估指标 5
3.3.2 评估方法 5
4 研究方案 5
4.1 视频流服务的云-边协同架构的系统建模 5
4.2 公平性优化算法设计 7
4.2.1 优化目标的具体建模 7
4.2.2 优化算法的实现 8
4.3 模型公平性优化效果的评估方法 9
5 进度安排，预期达到的目标 10
5.1 进度安排 10
5.2 预期目标 10
6 课题已具备和所需的条件、经费 10
7 研究过程中可能遇到的困难和问题，解决的措施 10
8 主要参考文献 11

1 课题来源及研究的目的和意义

随着互联网迅速发展，微服务架构[1]作为一种高效、灵活的架构模式，正在被广泛应用于各种需要高实时性和灵活性的服务场景，如视频流媒体[2]、在线游戏等。微服务架构将传统的单体应用划分为多个独立、细粒度的服务模块，使每个模块能够根据需要进行独立的部署和扩展，并通过轻量级通信协议相互协作，从而显著提升了系统的灵活性和扩展性[3]。然而，尽管微服务在提升系统敏捷性和故障隔离方面具有明显优势，但其也面临诸多挑战。由于服务被分布在多个节点上，服务间的调用的网络延迟可能导致系统整体的响应速度降低。此外，这种分布式架构还增大了系统管理和运维的复杂性，如服务间依赖管理、状态一致性维护等[4, 5]。并且微服务的动态扩展往往对资源使用效率有更高要求，如何高效地分配和管理资源以保障服务质量，已成为微服务应用亟待解决的核心问题之一。

为了进一步优化微服务的响应速度和服务质量体验（QoE），云-边协同计算模式应运而生。云-边协同计算通过在云端和边缘节点之间合理分配任务，使得云端和边缘节点各自的资源能够得到最大化利用[6]。云端拥有丰富的计算资源，能够高效处理大规模数据和复杂的计算任务，但由于其物理位置通常距离用户较远，可能带来较高的网络传输延迟[7, 8]。边缘节点则位于用户附近[9]，可以就近处理用户请求，从而显著降低网络传输延迟[10]，并提升用户体验。然而，由于边缘节点计算资源相对有限，在高并发访问或者复杂任务处理中，其资源限制可能导致性能瓶颈并增加延迟。为了确保云-边协同计算的高效性，不仅需要合理分配任务和资源，还需要针对不同的计算需求和网络条件，灵活地进行服务实例的部署。通过在云端和边缘节点之间优化服务实例的分配，可以更好地发挥各自资源的优势，进一步提升整体服务质量。因此，如何合理分配任务和资源，尤其是服务实例的部署，成为了进一步优化 QoE 的关键问题。

现有的云-边协同部署方案通常集中于最大化整体 QoE[11]，即通过合理分配任务和资源来优化整体服务质量。这些方案在提升整体用户体验方面取得了显著进展，但仍然面临一个重要挑战——如何解决不同用户间的体验差异问题，即公平性[12]。公平性是指在服务提供过程中，通过合理分配资源和优化调度，尽可能减少不同用户在关键性能指标（如响应时间）上的差异，从而实现用户体验的一致性和均衡性。虽然一些研究已经尝试通过优化资源分配来改善不同用户的体验差异，但由于地理位置、网络条件以及节点资源负载等差异，不同用户的响应时间可能有较大波动，进而影响部分用户的体验[13]。对于那些需要高实时性和一致性的应用场景，这种体验差异尤为明显。例如，在视频流媒体服务中，远离边缘节点的用户因网络传输延迟而产生卡顿，严重影响其体验，而接近边缘节点的用户则能够享受更高质量的流媒体服务。由此可见，如何通过进一步优化服务实例的部署和资源分配方法，减少不同用户之间的响应时间差异，确保 QoE 在所有用户之间更加均衡，仍然是云-边协同部署中的重要研究课题。

实现云-边协同环境中的公平性部署不仅是提升整体 QoE 的必要条件，也是推动云-边协同计算在更多应用场景中普及的关键。在保障整体 QoE 的前提下，通过减少用户间的服务质量差异，不仅能够提升用户的整体满意度，也为云-边协同计算的实际应用提供

了更强的适应性和推广价值。因此，在云-边协同计算中实现微服务的公平性部署，具有重要的研究意义和广阔的应用前景。

2 国内外在该方向的研究现状及分析

近年来，随着云计算和边缘计算的快速发展，微服务在云-边协同环境中的部署及资源管理成为了研究热点。为了全面优化整体服务质量体验（QoE）和用户体验公平性，我们对现有研究成果进行分析和综述。

为了提升整体 QoE，一些研究者提出了基于用户位置的任务分配方法，降低用户请求的响应时间。例如，Liu 等人[14]提出了一种基于边缘计算低延迟特性的分布式任务调度策略。该策略通过在边缘节点进行就近处理，显著提升了系统对高并发请求的响应速度。Alsurdeh 等人[15]则提出了一种混合工作流调度方法，通过在云端和边缘节点之间协同分配任务，既提高了资源利用率，也显著增强了整体的 QoE。国内学者马璐等[16]提出了一种基于混合任务调度的云-边协同方法。该方法根据用户位置和边缘节点资源动态分配任务，从而减少了请求的响应时间。王朝等人[17]提出了一种基于博弈论的协作缓存策略，通过区域间协作缓存数据资源，显著降低了用户数据获取延迟，提高了系统 QoE。

此外，张雅洁等[18]通过在电力物联网下应用云-边协同计算，提出了基于位置的任务放置算法，优化了响应延迟并提升了整体 QoE。朱仪和江雪[19]提出了一种基于云-边协同的任务卸载策略，通过智能卸载减少了延迟，提高了系统的响应效率。张文康等[20]研究了低延迟故障预测算法，在云-边协同环境下有效降低了延迟并提高了 QoE。这些研究表明，云-边协同在提高资源利用率、降低延迟方面具有显著效果。然而，这些研究大多侧重于优化单一层次的性能，尚未充分考虑如何平衡不同用户间的 QoE 差异。尤其是在远离边缘节点的用户群体中，可能会出现明显的 QoE 不均衡，进而影响整体的用户体验公平性。

随着用户公平性需求的提升，近年来的研究开始更加关注如何通过优化资源分配和任务调度来减少不同用户之间的 QoE 差异。为了进一步提升用户体验的一致性，研究者们提出了多种方法。例如，Zhang 等人[21]基于图神经网络提出了一种工作负载迁移方案。该方案通过在边缘节点间动态调整资源分配，有效平衡了不同地理位置用户 QoE，从而显著降低了用户之间的 QoE 差异。肖旋等[22]在云-边协同场景下研究了服务器负载均衡与协调，提出了负载均衡方法，以进一步提升用户体验一致性。冯起等[23]提出了一种考虑云端距离的科技服务边缘计算资源均衡调度算法。该算法通过动态评估云端和边缘节点的距离与任务需求，优化资源调度，进而提升不同用户间的 QoE 一致性。Hao 等人[24]提出了一种基于深度强化学习（DRL）的计算卸载方案。该方案通过深度学习算法，在边缘节点之间动态分配计算任务，以减少用户体验差异并提升资源利用率。此外，Zhou 等人[25]提出了一种公平性导向的移动边缘缓存策略，通过缓存优化平衡不同用户的响应时间，减少资源分配不均带来的体验差异。这一方法特别适用于资源受限环境，为用户提供了更具公平性的服务。

除了这些已有的研究，近年来一些新的工作也为提升公平性提供了新的思路。例如，吴忠辉[26]研究了基于区块链的边缘分布式计算卸载技术，提出通过去中心化的方式改善资

源分配的公平性。徐恒炜[27]探讨了基于公平通信方案的穿戴设备隐私学习方法，在保障隐私的同时提高了通信的公平性。张世焱[28]研究了云-边协同网络中的多资源管理机制，提出了多资源调度策略，以平衡不同用户的服务体验。金韬等[29]提出了一种基于区块链的云-边协同系统设计，通过区块链技术提升了资源分配的透明度和公平性。这些研究为云-边协同系统中的公平性优化提供了新的技术路径，并有助于在多用户、多节点环境中实现更加均衡的 QoE。

尽管现有研究在提高整体 QoE 和优化用户体验一致性方面取得了一定进展，但多数方法仍面临着计算效率和迁移机制的挑战。例如，Zhang 等人[21]提出的基于图神经网络的工作负载迁移方案，虽然能有效平衡不同地理位置用户的 QoE，但其复杂的迁移机制在大规模分布式环境中难以实现高效计算，仍需进一步优化。肖旋等[22]提出的负载均衡方法，尽管能够提升用户体验一致性，但其在动态负载变化下的调度效率尚有待提升。冯起等[23]的云端距离评估方法虽然在一定程度上优化了资源调度，但对于大规模服务实例的调度仍存在一定的计算开销和复杂度。

此外，现有的研究多集中于单一层次的资源优化（如边缘计算或缓存策略），缺乏对云-边协同环境中微服务部署的整体性考量，未能全面解决不同用户 QoE 差异问题。例如，金韬等[29]基于区块链的云边协同系统设计，虽然提供了透明的资源分配机制，但仍然缺乏对不同资源类型的综合调度与平衡，未能完全优化用户之间的公平性。吴忠辉[26]的基于区块链的边缘分布式计算卸载技术，虽然通过去中心化的方式提高了资源分配的公平性，但在实际部署中仍面临计算效率和大规模环境下的调度问题。因此，现有研究的局限性在于，大多数方法还没有在云-边协同架构下进行全面整合，且缺乏针对微服务部署的公平性优化策略，无法充分解决远离边缘节点的用户 QoE 差异。

在此背景下，如何在云-边协同架构中更全面地结合云端和边缘资源，实现用户体验的公平性优化，仍然是一个亟待解决的关键问题。未来的研究应进一步探索在云-边协同环境中通过智能化的部署策略和公平性优化手段，以减少不同用户之间的 QoE 差异。这样不仅能够提升用户体验的一致性，还能更有效地支持边缘计算的应用，最终满足用户对服务质量公平性日益增长的需求。

3 主要研究内容

本研究围绕在云-边协同环境中优化视频流媒体服务的公平性微服务部署，旨在通过构建合理的系统模型、设计优化算法和评估指标，最终实现服务的公平性优化。

随着视频流媒体用户数量的增长，不同用户因地理位置、网络条件等差异在服务体验上可能产生显著的响应时间差异，导致用户体验不一致。传统的云-边协同部署方法多集中于整体用户体验质量（QoE）的提升，但在不同用户之间的体验公平性方面仍存在不足。因此，本研究以最小化不同用户之间的响应时间差异为核心优化目标，力求缩小用户间体验的差距，提升用户体验的公平性。具体研究内容如下：

3.1 视频流服务的云-边协同架构系统建模

系统建模主要包括以下几个要素：

（1）**用户分布与服务实例部署** 识别用户的地理位置分布，设计不同区域的边缘节点和云节点的资源配置，确保每个区域的服务实例数量和位置能够满足区域内用户的响应需求。

（2）**延迟与响应时间** 分析物理拓扑结构、传输延迟、带宽限制和计算资源配置对用户响应时间的影响。具体来说，用户连接到边缘或云节点的延迟可细分为传输延迟（由地理距离和带宽决定）和处理延迟（由节点的计算资源决定）。两者共同影响最终的响应时间。

（3）**计算资源需求** 在每个用户连接到服务器时，服务器需要根据视频流的请求分配相应的计算资源。计算资源需求的集合包括 CPU 资源、内存资源和带宽资源等，这些需求随着用户的连接数量和请求变化。

3.2 Jain 公平性指数导向的公平性优化算法

为了减少不同用户间的响应时间差异并提升公平性，本研究基于 Jain 公平性指数设计了一套公平性优化方法，包括优化目标函数的构建与算法的初步设计。

3.2.1 优化目标及约束条件

优化目标函数的核心在于通过 Jain 公平性指数衡量和优化用户响应时间的分配公平性，从而提升用户体验的一致性。通过构建以最大化公平性为目标的数学模型，综合考虑以下约束条件：

（1）**低延迟需求** 满足视频流媒体服务的特性，确保响应时间支持实时播放要求，以提供流畅的用户体验。

（2）**成本限制** 合理控制部署成本，将经济性作为约束条件之一，以确保优化方案在预算范围内可行。

（3）**资源配置限制** 设定边缘节点和云节点的资源上限，防止因资源超载而导致性能下降，保障服务质量的稳定性。

（4）**用户与服务器连接约束** 每个用户必须连接到唯一的服务器，以保证请求能够正确地路由到合适的服务实例。

3.2.2 优化算法设计概述

基于优化目标函数，初步设计了一种解决方案，采用高效的优化算法实现公平性目标。算法将从以下思路出发：

（1）**全局优化** 通过算法设计，避免优化过程陷入局部最优。

（2）**动态调整** 根据用户请求和节点资源状况，灵活调整服务实例的部署。

3.3 模型公平性优化效果的评估标准

为验证模型的公平性优化效果，本研究建立了一个全面的评估指标体系和评估方法，

以确保模型在实际应用中的公平性表现具有可验证性和可改进性。具体内容如下：

3.3.1 评估指标

通过一组关键指标对模型的公平性优化效果进行量化评估，以全面衡量模型的实际应用表现：

（1）**用户体验的公平性** 通过计算 Jain 公平性指数来量化用户体验的公平性。Jain 公平性指数能够综合衡量所有用户的响应时间分配均衡性，包括不同优先级用户之间的公平性以及相同优先级用户的公平性。

（2）**平均响应时间** 统计所有用户的平均响应时间，确保在实现公平性优化的同时，整体服务响应时间符合实时播放需求。该指标能够衡量整体用户体验的质量。

（3）**部署成本** 计算边缘节点和云节点的总成本，确保优化方案在预算范围内可行。部署成本包括节点租用费用、计算资源使用费用等。通过该指标确保模型在保证公平性和资源配置合理性的同时，也满足经济性要求。

（4）**资源消耗** 监控各节点的资源消耗情况，包括 CPU、内存和带宽使用率。通过合理分配资源，避免边缘和云节点的过度负载，从而保障服务质量的稳定性。该指标反映了系统资源使用的效率，过高的资源消耗可能导致成本增加或服务质量下降。

3.3.2 评估方法

在本研究中，评估方法主要包括以下几种方式，以确保全面和客观地量化模型的优化效果：

（1）**量化分析法** 通过计算各项评估指标（如 Jain 公平性指数、平均响应时间、部署成本、资源消耗等）的数值，定量评估模型优化前后不同用户间的响应时间公平性、整体性能以及资源利用率的改进程度。

（2）**灵敏度分析** 在模型优化过程中，通过调整不同的约束参数（如资源限制、带宽要求、成本预算等），分析模型对不同输入条件的敏感度。通过灵敏度分析，能够评估模型在面对不同实际应用场景时的适应能力和稳定性。

（3）**综合评估** 将所有评估指标结合起来，计算加权综合得分，从而全面评估模型的公平性优化效果。此方法有助于同时考虑多个因素，尤其是公平性与经济性的平衡。

4 研究方案

本研究旨在优化视频流服务在云边协同环境中的公平性部署方案。研究方案具体包括以下几个步骤：

4.1 视频流服务的云-边协同架构系统建模

对于面向公平性的视频流服务云-边微服务系统部署模型，我们可以从以下几个方面来描述整个网络拓扑的建模，包括：用户请求的分布、服务实例的部署位置、不同节点之间的延迟模型、节点资源的分配与约束等。

具体场景如下：假设我们有一个视频流服务系统，用户分布在不同的地理位置。为了保证视频流的低延迟和高质量体验，服务的部署不仅仅依赖于云服务器，还需要考虑部署

在边缘节点的服务实例。用户请求首先由最近的边缘节点处理，如果边缘节点无法满足资源需求，则请求被转发到云服务器处理。同时，系统会根据用户类型分配优先级，付费用户的请求将被优先处理，以确保其享有更短的响应时间和更稳定的服务质量。

我们假设有$U = \{u_1, u_2, \cdots, u_n\}$个用户，每个用户$u_i$有一定的数据需求$D_i$，表示该用户请求的视频流数据量。用户之间的地理位置不同，可能存在不同的网络延迟。用户的位置用二维坐标(x_i, y_i)表示。用户u_i的优先级用Q_i表示，用户优先级越高，Q_i值越大。此外，我们有$S = \{s_1, s_2, \cdots, s_m\}$个服务器，表示所有可供用户连接的服务器集合，包括边缘服务器集S_{edge}和云服务器集S_{cloud}。边缘服务器通常部署在接近用户的区域，以提供低延迟服务，其位置也用二维坐标(x_j, y_j)表示，分布在用户覆盖范围的内部或周边区域。相比之下，云服务器位置通常固定在远离用户的中心化数据中心，以提供大规模计算和存储资源。每个服务器s_j上可能部署了一个或多个服务实例；如果没有用户连接到某个服务器，该服务器可以不部署任何服务实例，处于“非活跃”状态。服务实例的数量受到节点资源限制的约束。服务器和用户之间的连接通过变量x_{ij}来表示，x_{ij}为 1 表示用户u_i连接到服务器s_j，为 0 则表示没有连接。用户u_i连接到到服务器s_j时，服务器需要分配一定的资源来处理其视频流请求。资源需求集合$R_i = \{r_i^{cpu}, r_i^{mem}, r_i^b\}$表示了这些资源的具体量。其中$r_i^{cpu}$、$r_i^{mem}$和$r_i^b$分别表示用户$u_i$所需的 CPU、内存和带宽资源量。用户和服务器之间的物理距离d_{ij}可通过用户和服务器的二维坐标计算得到，如公式（4−1）所示。

$$d_{ij} = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2} \tag{4-1}$$

用户请求从u_i到服务节点s_j的延迟由传输延迟和处理延迟组成。

对于边缘节点，传输延迟$t^e_{trans_{ij}}$包括物理传输延迟和带宽延迟，物理传输延迟$t^e_{d_{ij}}$可以通过用户u_i到边缘服务节点s_j的距离d^e_{ij}和信号传播速度v_e来计算，如公式（4-2）；而带宽延迟$t^e_{b_{ij}}$则与用户请求数据大小D_i以及边缘网络带宽$b_e(t)$有关，如公式（4-3）。边缘节点总传输延迟为公式（4-4）所示。边缘节点的处理延迟$t^e_{proc_{ij}}$由其处理能力决定，假设边缘节点s_j的处理速率为P^e_j，则其处理延迟为公式（4-5）所示。

$$t^e_{d_{ij}} = \frac{d^e_{ij}}{v_e} \tag{4-2}$$

$$t^e_{b_{ij}} = \frac{D_i}{b_e(t)} \tag{4-3}$$

$$t^e_{trans_{ij}} = t^e_{d_{ij}} + t^e_{b_{ij}} \tag{4-4}$$

$$t^e_{proc_{ij}} = \frac{D_i}{P^e_j} \tag{4-5}$$

同样的，云节点的传输延迟$t^c_{trans_{ij}}$和处理延迟$t^c_{proc_{ij}}$也可按照类似的方式计算，但其云节点通常具有更高的带宽和处理能力。云节点的延迟计算公式如（4-6）、（4-7）所示。

$$t^c_{trans_{ij}} = t^c_{d_{ij}} + t^c_{b_{ij}} \tag{4-6}$$

$$t^c_{proc_{ij}} = \frac{D_i}{P^c_j} \tag{4-7}$$

式中，$t_{d_{ij}}^{c}$为云节点的物理传输延迟，$t_{b_{ij}}^{c}$云节点的带宽延迟，P_j^c为云节点的处理速率。

综合响应时间t_{ij}由传输延迟和处理延迟组成。若用户连接到边缘节点，则响应时间为公式（4-8），若用户连接到云节点，则响应时间为公式（4-9）：

$$t_{ij} = t_{trans_{ij}}^{e} + t_{proc_{ij}}^{e} = (\frac{d_{ij}^{e}}{v_e} + \frac{D_i}{b_e(t)}) + \frac{D_i}{P_j^e},\ s_j \in S_{edge} \tag{4-8}$$

$$t_{ij} = t_{trans_{ij}}^{c} + t_{proc_{ij}}^{c} = (\frac{d_{ij}^{c}}{v_c} + \frac{D_i}{b_c(t)}) + \frac{D_i}{P_j^c},\ s_j \in S_{cloud} \tag{4-9}$$

在上述响应时间计算的基础上，引入加权响应时间的概念，用以进一步体现不同用户优先级对响应时间的影响。加权响应时间t_{ij}^{weight}定义为用户u_i的响应时间t_{ij}与其优先级Q_i的乘积，如（4-10）所示：

$$t_{ij}^{weight} = t_{ij} \cdot Q_i \tag{4-10}$$

对于服务实例的部署，边缘节点和云节点的资源不同，边缘节点通常资源有限，而云节点资源相对充足。每个节点部署成本c_j的计算方式如下：

边缘节点的部署成本c_j^e可以表达为公式（4-11），其中$c_{fixed_j}^{e}$是固定成本，$c_{usage_j}^{e}$是基于用户需求所消耗的 CPU、内存和带宽资源的成本, 可以通过用户请求量和节点资源的消耗来计算，如公式（4-12）：

$$c_j^e = c_{fixed_j}^{e} + c_{usage_j}^{e} \tag{4-11}$$

$$c_{usage_j}^{e} = \sum_{i=1}^{n} x_{ij} \cdot (r_i^{cpu} \cdot p_{cpu}^{e} + r_i^{mem} \cdot p_{mem}^{e} + r_i^{b} \cdot p_b^{e}) \tag{4-12}$$

式中，p_{cpu}^{e}、p_{mem}^{e}和p_b^e为边缘节点的资源单价。

云节点部署成本c_j^c可以表示为云资源使用成本和网络流量成本之和，如公式（4-13），其中云资源使用成本$c_{usage_j}^{c}$与边缘节点资源使用成本计算方法类似，如公式（4-14）；而网络流量成本$c_{net_j}^{c}$是根据用户从不同区域访问云节点所产生的额外网络传输费用来计算的, 如公式（4-15）：

$$c_j^c = c_{usage_j}^{c} + c_{net_j}^{c} \tag{4-13}$$

$$c_{usage_j}^{c} = \sum_{i=1}^{n} x_{ij} \cdot (r_i^{cpu} \cdot p_{cpu}^{c} + r_i^{mem} \cdot p_{mem}^{c} + r_i^{b} \cdot p_b^{c}) \tag{4-14}$$

$$c_{net_j}^{c} = \sum_{i=1}^{n} x_{ij} \cdot D_i \cdot p_{net}^{c} \tag{4-15}$$

式中，p_{net}^{c}表示云平台的流量单价。

故总的部署成本C_{total}可以表示为所有边缘节点和云节点的部署成本之和，如（4-16）：

$$C_{total} = \sum_{s_j \in S_{edge}} c_j^e + \sum_{s_j \in S_{cloud}} c_j^c \tag{4-16}$$

4.2 公平性优化算法设计

4.2.1 优化目标的具体建模

为了解决公平性优化问题，首先需要明确优化目标以及相关约束条件。优化目标是通过最大化用户响应时间的公平性，来提高系统的公平性。使用 Jain 公平性指数作为公平性评价标准，该指数能够综合衡量所有用户响应时间分布的均衡性，尤其是在用户有不同优先级的情况下，其能够充分考虑到高优先级用户的服务体验，确保其在公平性优化过程中

获得更优的资源分配。约束条件则确保在优化过程中考虑到响应时间上限、资源限制、成本限制以及用户与服务器连接关系等实际情况。

（1）**优化目标** 优化目标是通过最大化用户加权响应时间的公平性，从而提升系统的公平性。公平性目标函数如（4-17）所示：

$$f = \min(1 - F_{Jain}) \tag{4-17}$$

其中，Jain 公平性指数F_{Jain}定义为公式（4-18）：

$$F_{Jain} = \frac{（\sum_{i=1}^{n} t_{ij}^{weight}）^2}{n \cdot \sum_{i=1}^{n} (t_{ij}^{weight})^2} \tag{4-18}$$

（2）**约束条件** 为了确保优化过程中的可行性，必须考虑响应时间响应时间、部署成本、资源消耗等多方面的约束条件。具体约束如下：

约束 1 为平均响应时间约束。为了确保系统的整体响应时间不超过服务质量要求，对平均响应时间设定了上限T_{max}，具体约束如（4-19）所示：

$$\frac{1}{n}\sum_{i=1}^{n}\sum_{j=1}^{m} x_{ij} \cdot t_{ij} \leq T_{max} \tag{4-19}$$

约束 2 为部署成本约束。为了控制系统的部署成本，我们设定了边缘节点和云节点的预算限制，确保总体成本不超过服务提供商的最大预算C_{max}，如（4-20）所示：

$$C_{total} = \sum_{s_j \in S_{edge}} c_j^e + \sum_{s_j \in S_{cloud}} c_j^c \leq C_{max} \tag{4-20}$$

约束 3 为边缘节点资源限制。每个边缘节点的计算资源不能超过其最大可用资源，包括 CPU、内存和带宽，以$R_j^{cpu_max}$、$R_j^{mem\_max}$、$R_j^{b_max}$分别表示 CPU、存和带宽的最大可用资源，具体约束如（4-21）、（4-22）和（4-23）所示：

$$\sum_{i=1}^{n} x_{ij} \cdot r_i^{cpu} \leq R_j^{cpu\_max}, \forall s_j \in S_{edge} \tag{4-21}$$

$$\sum_{i=1}^{n} x_{ij} \cdot r_i^{mem} \leq R_j^{mem\_max}, \forall s_j \in S_{edge} \tag{4-22}$$

$$\sum_{i=1}^{n} x_{ij} \cdot r_i^{b} \leq R_j^{b\_max}, \forall s_j \in S_{edge} \tag{4-23}$$

约束 4 为用户与服务器连接约束。每个用户必须连接到唯一的服务器，以保证请求能够正确地路由到合适的服务实例，如（4-24）所示：

$$\sum_{j=1}^{m} x_{ij} = 1, \forall u_i \in U \tag{4-24}$$

4.2.2 优化算法的实现

为了解决公平性优化问题并最大化用户间加权响应时间的公平性，我们考虑了几种常见的优化算法。以下是几种候选算法及其原理分析：

（1）**遗传算法（GA）** 遗传算法是一种模拟自然选择和遗传学原理的优化算法。在每一代中，算法会生成多个候选解（个体），这些个体根据适应度函数评估其优劣，适应度越高的个体被选中进行“交叉”和“变异”，生成下一代候选解。遗传算法的核心思想是通过选择、交叉、变异等操作，从一个随机的种群中逐步筛选出最优解。遗传算法特别适合处理复杂、非线性、多目标和大规模的优化问题。由于本研究的问题涉及多个约束条件，如成本、响应时间和资源配置，同时优化多个目标，遗传算法能够在解空间中进行全局搜索，有效避免陷入局部最优解，并能处理复杂的约束条件。

（2）**粒子群优化算法（PSO）** 粒子群优化算法模拟了鸟群觅食行为，在优化过程中，每个候选解（粒子）通过位置和速度的调整来寻找最优解。每个粒子都会根据自己的历史经验以及整个粒子群体的经验来更新自己的位置。粒子通过在搜索空间中迭代移动，最终收敛到最优解。PSO 算法的优势在于搜索过程较为简单，计算量较小。粒子群优化适合解决连续的优化问题，并且具有较强的全局搜索能力，能够在高维空间中有效探索最优解。其简单且高效的搜索过程使得它成为一种非常合适的优化工具，尤其在优化问题没有复杂约束的情况下，能够帮助快速找到全局最优解，支持研究中的优化目标。

（3）**模拟退火算法（SA）** 模拟退火算法是一种随机优化算法，模拟了金属冷却过程中的原子排布过程，通过引入“温度”逐渐降低的机制来控制搜索步长。初始时，温度较高，允许算法跳出局部最优解，随着温度的降低，搜索过程逐渐收敛到最优解。模拟退火算法的优点是可以接受某些不太优的解，以跳出局部最优，但随着时间的推移，会逐步趋近全局最优解。模拟退火算法特别适合处理非线性、多峰问题，能够有效避免局部最优解，并在全局最优解的搜索中展现出良好的探索能力。对于本研究中涉及的复杂优化问题，模拟退火能够提供有价值的解空间探索，有助于优化服务部署和资源分配策略。

（4）**贝叶斯优化（Bayesian Optimization）** 贝叶斯优化是一种基于概率模型的全局优化方法，主要用于解决目标函数评估代价高昂的优化问题。通过构建目标函数的概率模型，并使用贝叶斯推断方法来逐步逼近最优解。贝叶斯优化采用高斯过程（Gaussian Process）或类似模型来表示目标函数的不确定性，并利用已评估的数据指导后续搜索。贝叶斯优化在处理高维度、具有不确定性的优化问题时表现出色，尤其在计算资源昂贵或评估代价较高的情况下，能够高效地寻找最优解。在微服务部署优化中，贝叶斯优化能够通过较少的评估步骤提供较优的解决方案，帮助提高优化效率，尤其适用于评估代价较高的情境，有助于在复杂的资源约束下实现较优的微服务部署。

4.3 模型公平性优化效果的评估方法

在本研究中，评估的实施方式将通过具体的实验和数据分析来进行，确保模型在真实环境中的有效性和可行性。评估实施方式包括以下几种具体操作：

（1）**仿真实验** 通过仿真平台，模拟不同用户分布、服务节点配置、带宽条件等环境下的服务部署。仿真实验能够在没有实际部署的情况下，评估不同优化方案对公平性、响应时间和资源消耗等的影响。实验过程中，将对比优化方案与传统部署方案的差异，评估各项评估指标，如 Jain 公平性指数、平均响应时间等。此外，仿真环境还将涵盖不同的负载情况、网络延迟以及节点资源配置等，以检验模型在各种复杂情况下的表现。

（2）**对比分析** 通过在相同实验环境下对不同方案的 Jain 公平性指数、平均响应时间和资源利用率等指标进行比较，全面体现本研究算法在提升系统公平性方面的优势。对比分析还将展示本研究算法在保持整体服务性能（如响应时间满足实时播放需求）的同时，显著缩小用户体验差异的效果，为方案的实际应用提供有力支持。

（3）**敏感性分析** 对关键模型参数（如平均响应时间、资源配置上限和成本预算）进行敏感性分析，观察不同参数对 Jain 公平性指数和模型优化效果的影响。例如，通过调

整资源限制，分析模型是否能够在高负载或低预算条件下依然实现高公平性。这一分析将验证模型的健壮性，确保其在多种约束条件下能够稳定提升用户体验公平性。

（4）**成本效益分析** 通过成本效益分析，将不同优化方案的成本（如部署成本、资源消耗）与公平性收益（通过 Jain 公平性指数衡量）进行对比。分析优化方案是否能够在提升用户体验公平性的同时，控制经济成本，确保优化结果具有较高的实际应用价值。

5 进度安排，预期达到的目标

5.1 进度安排

工作安排	周数	起止时间
系统建模与设计阶段	3	2024.11.20-2024.12.15
算法设计与实现	5	2024.12.16-2025.01.19
仿真实验与数据收集	6	2025.01.20-2025.03.02
敏感性分析与模型调优	4	2025.03.03-2025.03.31
成本效益分析与优化验证	4	2025.04.01-2025.04.30
结题，论文编写	2	2025.05.01-2025.05.15

5.2 预期目标

（1）**中期目标** 完成云边协同环境中视频流服务的公平性优化模型设计和算法的基础框架，进行初步实验与参数调优。

（2）**结题目标** 实现云边协同环境下的公平性优化模型，能够基于视频流服务的实际需求，在满足资源和成本限制的前提下，通过优化服务实例的部署和任务调度，最大化 Jain 公平性指数，从而提升用户体验的一致性并减少响应时间差异。

6 课题已具备和所需的条件、经费

本课题的研究工作已经具备了充足的条件和经费保障。首先，研究将充分利用现有的实验室资源和仿真平台，这些平台能够支持大规模的云-边协同架构实验，并进行多种性能评估。实验室内拥有必要的硬件设备以及高性能计算资源，为研究提供了充分的技术保障。与此同时，相关经费也已充足，能够覆盖研究过程中所需的软硬件支出、数据采集与处理、人员支持等各项费用。因此，课题的顺利开展具有良好的资源基础和保障。

7 研究过程中可能遇到的困难和问题，解决的措施

（1）**数据真实性不足** 仿真数据可能无法完全模拟实际应用场景的数据特征，影响实验结果的可靠性；解决措施：在仿真数据的基础上，可以通过公开数据集进行结果验证，提高模型的适用性。

（2）**约束条件冲突** 多个约束条件（如延迟、成本、资源分配）之间可能存在冲突，影响优化结果；解决措施：采用多目标优化方法，平衡不同目标之间的关系。

8 主要参考文献

[1] J. Thönes, "Microservices," in IEEE Software. 2015, (1): 116-116.

[2] Ashwin Rao, Arnaud Legout, Yeon-sup Lim, Don Towsley, Chadi Barakat, and Walid Dabbous. Network characteristics of video streaming traffic. In Proceedings of the Seventh COnference on emerging Networking EXperiments and Technologies (CoNEXT '11). 2011: 1–12.

[3] Dragoni N, Mazzara M, Meyer B. Microservices: Yesterday, Today, and Tomorrow. Present and Ulterior Software Engineering. Springer, Cham. 2017: 195-216.

[4] 赵然，朱小勇．微服务架构评述．网络新媒体技术．2019, 8(1): 58-61.

[5] 李春霞．微服务架构研究概述．软件导刊．2019, 18(8): 1-3, 7.

[6] Pan J, McElhannon J. Future Edge Cloud and Edge Computing for Internet of Things Applications. IEEE Internet of Things Journal. 2018, 5(1): 439-449.

[7] Sadiku M N O, Musa S M, Momoh O D. Cloud Computing: Opportunities and Challenges. IEEE Potentials. 2014, 33(1): 34-36.

[8] Zhang W, Hu Y, Zhang Y, Raychaudhuri D. SEGUE: Quality of service aware edge cloud service migration. Proceedings of IEEE International Conference on Cloud Computing Technology and Science. 2016: 344-351.

[9] He Q, Luo J, Xie X, Liang Z, Tang X, Fu X. A game-theoretical approach for mitigating edge DDoS attack. IEEE Transactions on Dependable and Secure Computing. 2022, 19(4): 2333-2348.

[10] Cao K, Liu Y, Meng G, Sun Q. An Overview on Edge Computing Research. IEEE Access. 2020, 8: 85714-85728.

[11] He X, Xu H, Xu X, Chen Y, Wang Z. An Efficient Algorithm for Microservice Placement in Cloud-Edge Collaborative Computing Environment. IEEE Transactions on Services Computing. 2024, 17(5): 1983-1997.

[12] Shi H, Prasad R V, Onur E, Niemegeers I G M M. Fairness in Wireless Networks: Issues, Measures and Challenges. IEEE Communications Surveys & Tutorials. 2014, 16(1): 5-24.

[13] Lai P, Guo F, Jiang Y, Wu X, Chen J, Lu Y. QoE-aware user allocation in edge computing systems with dynamic QoS. Future Generation Computer Systems. 2020, 112: 684-694.

[14] Liu F, Tang G, Li Y, Cai Z, Zhang X, Zhou T. A Survey on Edge Computing Systems and Tools. Proceedings of the IEEE. 2019, 107(8): 1537-1562.

[15] Alsurdeh R, Calheiros R N, Matawie K M, Javadi B. Hybrid Workflow Scheduling on Edge Cloud Computing Systems. IEEE Access. 2021, 9: 134783-134799.

[16] 马璐，刘铭，李超，路兆铭，马欢．面向 6G 边缘网络的云边协同计算任务调度算[J]．北京邮电大学学报．2020, 43(6): 66-73.

[17] 王朝，高岭，高全力，等．边缘计算中基于博弈论的数据协作缓存策略研究[J].计算机应用研究．2020, 37 (12): 3739-3743.

[18] 张雅洁, 陆旭, 李曦, 等. 电力物联网下基于云边协同的计算任务放置算法[J].电力信息与通信技术．2024,22(10): 38-47.

[19] 朱仪，江雪．基于云边协同的任务卸载策略技术研究．无线电工程．2024: 1-21.
[20] 张文康, 赵伟, 刘德成, 等. 云边协同低延迟故障预测算法研究[J].能源与环保. 2024,46(10): 238-243.
[21] Zhang C, Yin J, Deng S. Ensuring Fairness in Edge Networks: A GNN-Based Media Workload Migration Scheme With Fairness Guarantee. IEEE Transactions on Services Computing. 2024, 17(3): 934-948.
[22] 肖旋. 云边协同场景下服务器负载均衡与协调研究[D]. 重庆大学, 2021.
[23] 冯起，薛喜红，任龙，等．考虑云端距离的科技服务边缘计算资源均衡调度算法．自动化技术与应用．2024, 43(8): 95-98+104.
[24] Hao H, Xu C, Zhang W, Yang S, Muntean G M M. Computing Offloading With Fairness Guarantee: A Deep Reinforcement Learning Method. IEEE Transactions on Circuits and Systems for Video Technology. 2023, 33(10): 6117-6130.
[25] Zhou J, Chen F, He Q, Xia X, Wang R, Xiang Y. Data Caching Optimization With Fairness in Mobile Edge Computing. IEEE Transactions on Services Computing. 2023, 16(3): 1750-1762.
[26] 吴忠辉.基于区块链的边缘分布式计算卸载关键技术研究[D].北京邮电大学. 2024.
[27] 徐恒炜. 基于公平通信方案的穿戴设备隐私学习方法[D].中国科学技术大学. 2023.
[28] 张世焱. 云边协同网络中的多资源管理机制研究[D].北京邮电大学. 2024.
[29] 金韬, 庄丽婉, 张晨, 等. 基于区块链的云边协同系统研究与设计[J].信息安全研究. 2021,7(04): 310-318.