

面向算力网络的云边端协同调度技术



Cloud-Edge-End Collaborative Scheduling Technology for Computing Power Network

周旭/ZHOU Xu, 李琢/LI Zhuo

(中国科学院计算机网络信息中心, 中国 北京 100190)
(Computer Network Information Center, Chinese Academy of Sciences,
Beijing 100190, China)

DOI: 10.12142/ZTETJ.202304007

网络出版地址: <http://kns.cnki.net/kcms/detail/34.1228.TN.20230717.1804.004.html>

网络出版日期: 2023-07-18

收稿日期: 2023-05-23

摘要: 针对网络规模庞大、资源分散和异构性等挑战, 面向算力网络的云边端协同调度技术通过算力网络将云边端多级泛在算网资源整合在一起, 从而形成一个庞大的、跨越多个地域的资源池。综合考虑网络实时状态、用户需求等要素, 该技术能够实现对算网资源的统一管理和动态调整, 提升用户体验的同时降低企业运营成本和运维复杂度。

关键词: 算力网络; 云边端协同; 协同调度技术

Abstract: Faced with challenges such as a large scale of networks, dispersed and heterogeneous resources, the cloud-edge-end collaborative scheduling technology for computing power networks integrates multi-level ubiquitous computing network resources from the cloud, edge, and end into a large, cross-regional resource pool through the computing power network. By considering factors such as real-time network status and user demands, the technology achieves unified management and dynamic adjustment of computing network resources, improving user experience while reducing enterprise operation costs and maintenance complexity.

Keywords: computing power network; cloud-edge-end collaboration; collaborative scheduling technology

引用格式: 周旭, 李琢. 面向算力网络的云边端协同调度技术 [J]. 中兴通讯技术, 2023, 29(4): 32-37. DOI: 10.12142/ZTETJ.202304007

Citation: ZHOU X, LI Z. Cloud-edge-end collaborative scheduling technology for computing power network [J]. ZTE technology journal, 2023, 29(4): 32-37. DOI: 10.12142/ZTETJ.202304007

数字化时代带来海量数据的增长。这种趋势推动了大量计算资源的需求增长, 诸如数据中心、边缘计算节点和各种终端设备等构成了一个庞大且复杂的分布式计算资源网络。在这个网络中, 有效地管理和调度各个节点的计算资源, 以提供高效、稳定的服务, 是一大挑战。传统的调度技术往往针对特定的场景和需求, 无法满足多样化、动态变化的计算需求。此外, 由于云端和边缘端的资源异构性、地理分布的广泛性以及网络环境复杂性等, 资源的管理和调度更加困难。

为了解决这一问题, 云边端协同调度技术应运而生。通过整合云、边和端的资源, 实现对资源的统一管理和动态调度, 以最大化资源的利用率, 保障用户的服务质量。在算力

网络的背景下, 云边端协同调度技术的研究尤为必要^[1]。算力网络将分布在各地的云端、边缘端和终端设备的计算资源通过网络连接起来, 形成一个分布式的、动态的计算资源池。这个资源池可以实现对所有计算资源的统一管理和调度, 从而提高资源的利用率, 降低运营成本, 提高服务质量^[2]。在算力网络中使用云边端协同调度技术, 可以充分利用网络的优势, 有效满足复杂的网络环境和多样化的计算需求。因此, 本文将重点阐述面向算力网络的云边端协同调度技术, 探讨其在实际应用中的效果和关键技术^[3-4]。

1 分布式的云边端算力

大数据和物联网等场景需要使用云计算和边缘计算技术, 这样可以在全网广泛分布的设备上提供计算服务, 将计算能力广泛分布到需要的地方, 从而提高数据处理的效率, 减少延迟, 并提供更个性化、地域化的服务。

基金项目: 国家重点研发计划 (2018YFB1800100); 国家自然科学基金 (U1909204)

1.1 云边端算力概述

为了提供更加高效、灵活和可靠的计算服务，我们提出集云计算与边缘计算于一体的计算模式——云边端算力模式。该模式综合了云计算中心的大规模存储和处理能力，以及边缘计算对于接近数据源的处理能力。在该模式中，云端负责处理大规模、复杂的计算任务，而边缘端则负责处理那些需要低延迟、快速响应的任务。通过这种方式，云边端算力能够在满足不同任务需求的同时，提高整体的计算效率。

在云边端算力模式中，数据首先会在传感器、移动设备等地方产生，再被边缘设备接收并进行初步处理。这些处理通常包括数据清理、预处理和部分分析等。随后，数据和任务会根据性质和需求，分配给云端或边端进行进一步处理。具体来讲，那些需要快速反馈的任务，通常会被留在边缘端；而那些需要大规模数据分析和深度处理的任务，则会被发送至云端。通过这种方式，云边端算力能够提供更加高效、灵活的计算服务。

1.2 分布式算力的优势和挑战

在分布式的云边端算力架构中，全网范围内的计算资源配置使得各种类型的计算任务处理更加灵活和高效。设备的资源可以根据任务需求进行动态调整，从而在维持高效运算的同时，提供稳定且高质量的服务。此外，这种配置方式还能应对各种突发的计算需求，快速进行资源调配，以满足不断变化的计算需求。

然而，分布式的云边端算力也面临着诸多挑战。在全网范围内进行高效的资源调度，需要考虑到设备状态、任务需求、网络状况等多个因素，这使得调度过程变得复杂。一方面，计算需求和设备状态可能会快速变化，因此如何在短时间内做出决策，保证服务的质量和效率，是一大挑战；另一方面，由于设备是分布式的，那么如何进行有效的分布式协调、收集设备状态信息、在设备间同步任务状态等问题也需要被解决。

1.3 分布式算力的应用和未来趋势

随着5G和6G网络技术的发展，更高的网络速度和更低的延迟推动了分布式云边端算力在更多领域中的应用，如无人驾驶、远程医疗、工业自动化等。同时，随着任务复杂性的增加，例如高精度的物理模拟、大规模的社会科学模型等，分布式计算资源的需求也将增多。此外，在机器学习和人工智能领域，分布式的云边端算力发挥着重要作用。通过利用分布式算力资源，机器学习模型能在更短的时间内进行训练和优化。

2 融合云边端的协同网络

2.1 融合云边端的网络架构

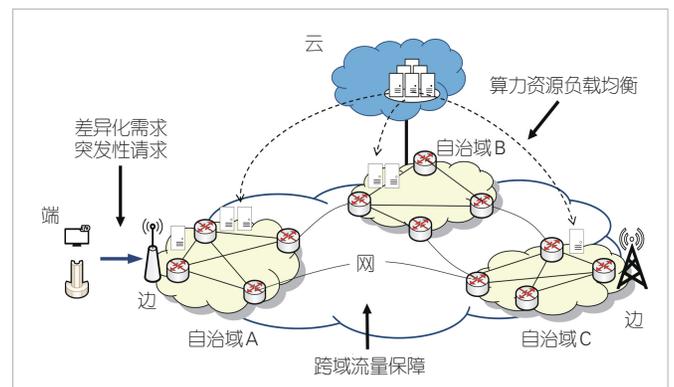
新型应用如移动互联网、物联网的普及以及新型网络技术的发展，使得传统的集中式云计算模式遇到了挑战^[5]。为了应对这些挑战，我们提出融合云边端的网络架构，具体如图1所示。该架构发挥了云计算强大的计算能力和边缘计算的低延迟优势，旨在提供更高效、更灵活的计算服务，以满足不断增长的计算需求和多样化的业务需求。融合云边端的网络架构将云端的大规模数据处理能力与边缘端的实时处理能力相结合，优化了网络资源的利用，提高了网络服务的质量。

在这种融合云边端的网络结构下，资源可以被更加高效地利用，网络的可靠性和安全性得到了提高。在该网络中，云计算作为物联网的中枢，通过将大量终端或边缘无法处理的数据进行存储、整理和分析，为云边端协同网络提供强大的计算能力支持。同时，部分任务可直接在终端设备或边缘服务器中进行处理，在减少数据传输延时的同时也缓解了云数据中心的压力。

2.2 协同网络的优化与调度策略

在融合云边端的网络架构中，协同工作非常重要。这需要云计算和边缘计算紧密协作，共同完成计算任务^[6]。为了实现这一目标，需要设计并实施有效的协同网络优化与调度策略。优化与调度策略的主要目标都是在满足任务需求的同时，最大程度地利用云边端的计算资源。协同网络的优化工作主要研究的是如何合理地分配任务和资源，如何实现高效的数据传输，以及如何维护网络的稳定性和可靠性等问题。而在调度策略方面，考虑的因素更为复杂，包括但不限于计算任务的性质、网络状况、设备能力、能耗和特定的应用需求等。

协同调度策略则需要将计算任务按照某种方式分配给云



▲图1 融合云边端的网络架构示意图

端或边缘设备，同时还需要考虑任务的执行顺序和资源的分配情况。在这个过程中，要尽可能地减少任务的执行时间，降低网络的传输延时，从而提升系统的整体性能。综上所述，优化与调度策略是实现云边端协同工作的关键，通过这些策略，可以更好地管理和调度网络中的资源，从而提升云边端协同网络的性能，满足多变和复杂的应用需求。

3 云边端协同调度关键技术

分布式的云边端算力提供了广阔的计算平台。融合云边端的协同网络构建了高效、灵活的计算资源调度模型^[7]。然而，为了挖掘这一模型的最大潜力，还需要更多的支撑技术，如云边端协同调度关键技术等（具体如图2所示）。协同网络的优化与调度策略可以最大化地利用和分配云边端的计算资源，而这一过程需要精细化的管理和调度。精细化管理和调度的实现需要云边端协同调度关键技术做支撑，具体包括跨云边端协同计算方法、端到端跨域保障机制和资源管理和任务调度策略等。在这些技术的支持下，云边端协同网络能够在满足各种复杂、变化需求的同时，解决协同网络的优化与调度策略的问题，优化网络资源配置，进一步提升网络性能^[8]。

3.1 跨云边端协同计算方法

在多终端、多任务的复杂场景下，卸载决策的制定需要综合任务计算量、数据传输量、云边端各节点计算能力和资源利用率等诸多因素。基于已有的云计算或边缘计算设计的协同计算方法不完全适用于云边端协同场景。另外，云边端设备的异构性强，这些方法也不适用于多种设备的需求。计算方法的不统一降低了处理效率，无法充分发挥协同的优势。因此，设计跨云边端的协同计算方法有很大的必要性。

为了应对云边端设备的异构性，跨云边端协同计算的方

法需要考虑不同的设备需求。一种方法是根据设备的计算能力和资源利用率来分配任务，将计算密集型任务分配给计算能力较强的云端节点，而将数据密集型任务分配给边缘节点，以充分利用各节点的资源；另一种方法是利用任务切分和协同执行，将一个任务切分成多个子任务，并将这些子任务分配给不同的设备，最后将它们的结果合并得到最终结果。

3.2 端到端跨域保障机制

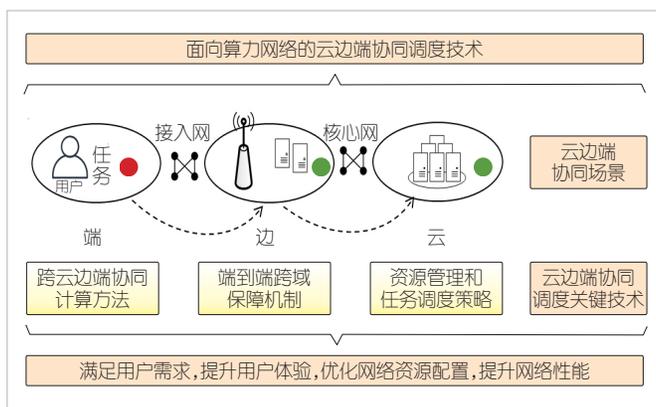
在云边端协同调度中，端到端跨域保障机制主要是指延迟优化和服务质量保证。二者紧密结合，共同支撑云边端协同调度的高效运行。有效的延迟优化，可以提升系统性能；有效的服务质量保证，可以确保系统始终处于高效稳定的状态，从而进一步降低延迟。

延迟优化是指减少任务执行和数据传输的时间延迟，尤其是在需要实时反馈或高速处理的应用中。延迟优化涉及的技术包括：1) 有效的任务调度，确保任务在最佳位置执行，减少数据传输和处理时间，优化网络路由和传输协议，减少网络传输延迟；2) 利用边缘计算的特性，将计算任务靠近数据源，减少数据传输的时间和距离。服务质量保证则是确保系统能提供用户所需的服务，这包括满足如执行速度、响应时间、数据准确性等的各种性能要求，以及满足特定的服务等级协议。服务质量保证涉及的技术包括资源和任务调度，以及各种容错和恢复机制，以保证系统有足够的资源来满足服务需求，应对可能的错误和故障。

3.3 资源管理和任务调度策略

在云边端协同网络中，资源管理和任务调度是两项关键技术。这两项技术的主要目标是优化系统性能，提升服务质量，从而实现协同网络的优化与调度策略^[9]。资源管理的核心是实现资源的高效利用，这包括了对云服务器、边缘设备和网络带宽等资源的合理分配和调度。具体来说，资源管理需要考虑系统的总体需求，以及各类资源的性能和状态，从而决定如何分配和调度这些资源。通过有效的资源管理，我们可以使系统在满足各种需求的同时，最大化地利用资源，提升系统性能。

任务调度则主要关注如何合理地分配和调度计算任务。任务调度需要考虑任务的特性，如任务的类型、大小、优先级，以及任务的执行环境等。基于这些信息，任务调度制定合适的策略，决定如何将任务分配给云服务器或边缘设备，以及如何安排任务的执行顺序^[10]。



▲图2 云边端协同调度关键技术

4 面向算力网络的典型应用场景

云边端协同调度技术已广泛应用于物联网与智慧城市、自动驾驶与无人机、远程医疗与虚拟现实等领域。本章节中，基于算力网络的典型应用场景，我们详细阐述了协同调度技术是如何在实际应用中发挥关键作用，优化性能并提升用户体验的。

4.1 物联网和智慧城市

在物联网环境中，大量的设备和传感器被用于收集各种类型的数据，如温度、湿度、位置等。然而，这些设备的计算能力和储存空间都非常有限。因此，分布式的云边端算力可以提供必要的计算资源，支持在边缘设备上数据进行预处理和实时分析。这大大减少了数据传输的延迟和网络带宽的需求。

智慧城市是物联网在更大规模下的应用。在智慧城市中，各种设备和系统都可以通过互联网进行连接，形成一个大型的、互相协作的网络。面向算力网络的云边端协同调度技术可以实现各种复杂的功能，如实时交通管理、能源优化、公共安全管理等。从隐私保护的角度来讲，通过在边缘设备上进行处理，可以确保数据的隐私和安全，同时提高服务的响应速度。

4.2 自动驾驶和无人机

对于自动驾驶来说，安全和实时性至关重要。车辆必须能够快速且准确地响应周围环境的变化，例如其他车辆的动态、行人甚至是天气状况。由于云端处理可能会引入无法接受的延迟，因此在车辆本地进行数据处理非常必要。然而，车载计算资源有限，无法处理大量的输入数据和复杂的算法。因此，云边端协同调度技术成为了一个理想的解决方案，它能够在保证实时性的同时，通过边缘计算节点的协同工作，提高数据处理能力。

无人机也有类似的需求。无人机通常需要实时的视频流处理，以进行物体检测、追踪等。这需要大量的计算资源，但无人机的载荷有限，难以满足这种需求。云边端协同调度技术可以将计算任务分配到无人机附近的边缘计算节点，从而实现实时的视频流处理。无论是自动驾驶还是无人机，云边端协同调度技术都能有效地解决有限的设备计算能力和严格的实时性要求之间的矛盾。

4.3 远程医疗和虚拟现实

远程医疗技术使医生可以在任何地方都能为患者提供服务。然而，这种类型的服务对网络的稳定性和时延有非常高

的要求。面向算力网络的云边端协同调度技术可以确保数据在云端和边缘设备之间快速、准确传输，尽可能降低延迟，满足远程医疗应用的需求。虚拟现实是一个对延迟和数据处理能力要求非常高的领域，任何微小的延迟都可能导致用户体验下降，甚至引发眩晕感。同时，虚拟现实应用通常需要处理大量的图形数据和用户交互信息，这超出了大多数个人设备的计算能力。因此，利用云边端协同调度技术，可以将部分计算任务卸载到边缘设备，如附近的边缘服务器，从而降低延迟，提高数据处理能力，实现更好的体验。

5 基于云边端协同的流量调度

在云边端协同网络中，传统的网络流量调度往往采用集中式的调度算法，由中心节点对整个网络的流量进行统一管理和调度。中心节点的调度算法效率低下，对网络性能也有很大的影响。为了能够根据实时的网络状态选择流量最佳转发路径，针对云边端分布式网络中的流量调度，我们提出基于云边端协同的流量调度，由云边端各节点共同协作完成网络流量的调度，避免单一链路出现拥塞而其他链路可能有剩余带宽未被充分利用的情况。同时，通过将不同类别的流量调度到满足传输性能的链路上，降低多流并发情况下的端到端时延，提升平均吞吐率。

5.1 云边端协同流量调度模型

流量调度中最重要的指标是流完成时间和吞吐率。当网络中的流量达到一定的程度时，网络中的拥塞会增加，从而导致流的完成时间增加，吞吐率下降。因此，流完成时间和吞吐率之间存在着一种权衡关系。如果仅仅关注流的完成时间，可能会导致网络的拥塞程度加剧，从而使得网络的吞吐率下降。相反，如果仅仅关注网络的吞吐率，可能会导致一些流的完成时间变得非常长，从而影响服务质量。因此，本文定义了包含以下内容的云边端协同流量调度模型。

1) 对于每个流，需要在规定的时间内完成传输，通过设置流量保障的重要程度以保证服务质量和用户体验。

2) 在满足流完成时间的基础上，需要保持一定的吞吐率水平，以充分利用云边端分布式网络中的带宽资源。

将端到端流量保障模型中的优化目标设置为同时最小化流平均完成时间和最大化吞吐率，以实现流量保障和网络性能的平衡，相应的优化问题可定义如下：

$$\begin{aligned} \min & \bar{\tau} + \lambda \times \frac{1}{\eta} \times \frac{1}{n} \times \sum_{i=1}^n \omega_i \times \tau_i \\ \text{s.t.} & \frac{1}{n} \sum_{i=1}^n \tau_i = \bar{\tau}, \end{aligned} \quad (1)$$

其中, $1/\eta$ 是所有流的总吞吐率的倒数, ω_i 是流*i*的权重, 表示该流对端到端流量保障的重要程度, τ_i 是流*i*的流完成时间, λ 是用来平衡完成时间和吞吐率之间的关系的系数。该优化问题的含义是: 将流的平均完成时间最小化, 同时通过对所有流的完成时间进行加权平均, 实现网络的总体吞吐率最大化。对流进行加权的目的是为了保障重要流的服务质量, 满足端到端流量保障的需求, 同时适当牺牲不重要的流, 以提高整体吞吐率。另外, 还可以通过调整 λ 的值来控制吞吐率和流完成时间之间的权衡关系。

5.2 分布式强化学习算法

为求解上述协同流量调度优化问题, 本小节在软策略演员-评论员(SAC)算法的基础上, 设计基于SAC的分布式强化学习(DSAC)算法, 具体如下:

算法1: 基于SAC的分布式强化学习算法

1. **for** iteration = 1,2,...,N do
2. **for** each worker *i* do
3. Sample a batch of traffic flows f_j from *D*
4. **for** each flow f_j
5. Compute the actions a_j, k using policy $\pi(\cdot|f_j; \theta_i)$
6. Compute the scheduling policies g_k using the policy library *G*
7. Compute the flow rate allocation by the scheduling policies
8. Compute the delay and reward of each flow
9. **end for**
10. Add the $(f_j, a_j, k, r_j, k, d_j)$ tuples to *D*
11. Compute target values
12. Update Q-function by minimizing the Bellman error
13. Update policy by maximizing the expected Q-value
14. Compute the action *a* using the current policy
15. Execute the action *a* and observe the next state *s'* and reward *r*
16. Store the transition (s, a, r, s', d) in the worker replay buffer *D*
17. Send the updated policy parameters θ_i and Q-function parameters ϕ_i to the parameter server
18. **end for**
19. Compute the new state representation *s* for each flow in the network
20. Update the network flow table with the new action *a* for each flow
21. **end for**

具体来讲, DSAC算法是SAC算法在分布式系统中的拓展。该算法可以将策略优化和Q值函数优化分配到多个智能体上来提高算法的效率。通过将强化学习算法中的演员-评

论员架构与软策略优化相结合, 实现高效的流量调度。在每个时刻, 演员通过观察当前网络状态和历史流量数据来选择一个最优的流量调度决策。评论员则根据演员的决策和真实流量数据来评估演员的决策, 并将其反馈给演员进行策略优化。算法可以在分布式环境中运行着多个演员-评论员框架同时协作以进行流量调度, 通过软策略优化来避免对策略进行硬约束, 从而使得云边端流量调度决策更加灵活。

5.3 性能仿真实验

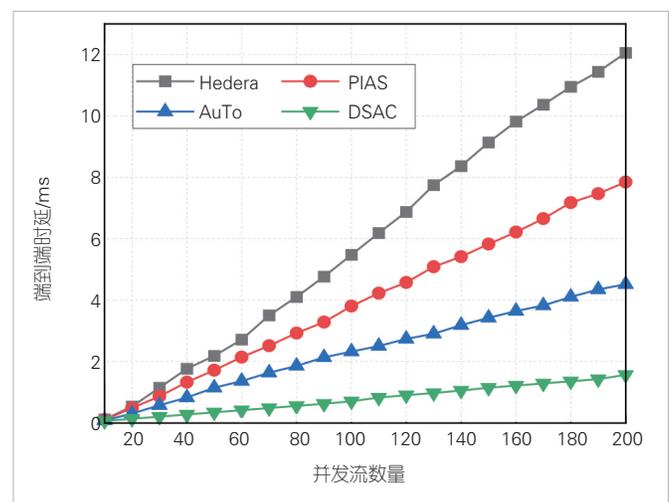
为了评估所提协同流量调度方案的有效性, 我们将所提的基于分布式深度强化学习的协同流量调度方案与以下几种典型的流量调度方案进行性能比较。

1) AuTo: 一种使用深度强化学习解决流量调度问题的方法。该方法根据网络中的流量负载自动地调整资源的使用, 以实现更好的网络流量管理, 并在保证服务质量的同时提高系统的效率^[11]。

2) PIAS: 一种信息不可知的流量调度算法。该算法能够动态地调度数据中心网络中的流量, 以确保高效的网络运行。该算法基于实用性和信息不可知性的设计原则, 通过计算流的权重和调度流来实现最佳性能和高网络利用率^[12]。

3) Hedera: 一种数据中心网络流调度方法。该方法使用了一种基于高负载优先的动态调度策略, 即优先处理那些负载更高的网络流量, 以避免网络拥塞和延迟, 实现更高的网络吞吐量和更低的延迟^[13]。

为了验证所提云边端协同调度方案的有效性, 我们分析了不同流量调度方案在多流并发下的流量调度效果。在端到端时延方面, 如图3所示, 随着并发流的增加, Hedera和PIAS的端到端时延快速增加。基于深度强化学习的AuTo和



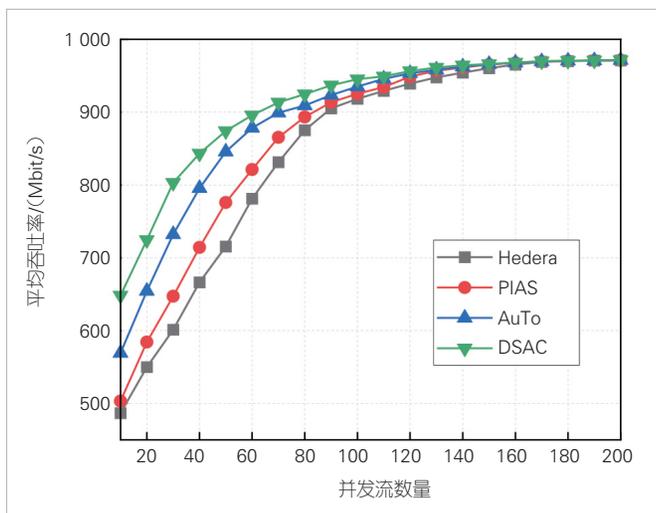
▲图3 多流并发下的端到端时延

本文所提的DSAC都能够提供较低的端到端时延。但DSAC比AuTo的时延更低。这是由于DSAC将强化学习算法中的演员-评论员架构与软策略优化算法相结合。这样一来,演员通过观察当前网络状态和历史流量数据来选择一个最优的流量调度决策,评论员则根据演员的决策和真实流量数据来评估演员的决策,再反馈给演员进行策略优化。基于DSAC算法的协同流量调度有效降低了端到端时延。

如图4所示,在平均吞吐率方面,随着并发流量数目的增多,各个算法的平均吞吐率都在增加。当并发流数量小于80时,相比于Hedera和PIAS,AuTo和DSAC获得的平均吞吐率有着明显的优势。这是由于AuTo和DRL为每个子网络分配一个深度强化学习智能体。这些智能体可以通过学习从其他子网络到自身的流量路由,实现全局流量优化,从而能够在减少端到端时延的同时提高平均吞吐率。这进一步表明了多流并发情况下,算法的端到端时延和平均吞吐率都有着更好的表现。

6 结束语

本文首先介绍了分布式云边端算力的未来发展趋势,探讨了融合云边端的协同网络架构,分析了云边端协同调度的关键技术,明确了其在解决实时性和高性能计算问题上的重要作用;其次,结合具体的应用场景如物联网、自动驾驶、远程医疗和虚拟现实等,展示了云边端协同调度技术的实际影响和潜力;最后,以云边端协同网络中的流量调度为例,从多流并发情况下的端到端时延和平均吞吐率需求出发,提出了基于分布式强化学习的协同流量调度算法,并通过实验验证了所提协同流量调度方案的有效性。本研究推动了云边端协同调度技术在分布式的云边端算力场景中的应用。



▲图4 多流并发下的平均吞吐率

参考文献

- [1] 雷波, 刘增义, 王旭亮, 等. 基于云、网、边融合的边缘计算新方案: 算力网络 [J]. 电信科学, 2019, 35(9): 44-51. DOI: 10.11959/j.issn.1000-0801.2019209
- [2] 张宏科, 权伟, 刘康. 算力网络研究与探索 [J]. 中兴通讯技术, 2023, 29(1): 1-5. DOI: 10.12142/ZTETJ.202301001
- [3] 段晓东, 姚惠娟, 付月霞, 等. 面向算网一体化演进的算力网络技术 [J]. 电信科学, 2021, 37(10): 76-85. DOI: 10.11959/j.issn.1000-0801.2021248
- [4] ASGHAR H, JUNG E S. A survey on scheduling techniques in the edge cloud: issues, challenges and future directions [EB/OL]. [2023-05-22]. <https://arxiv.org/abs/2202.07799>
- [5] KAI C H, ZHOU H, YI Y B, et al. Collaborative cloud-edge-end task offloading in mobile-edge computing networks with limited communication capability [J]. IEEE transactions on cognitive communications and networking, 2021, 7(2): 624-634. DOI: 10.1109/TCCN.2020.3018159
- [6] YANG H, LIANG Y S, YUAN J Q, et al. Distributed blockchain-based trusted multidomain collaboration for mobile edge computing in 5G and beyond [J]. IEEE transactions on industrial informatics, 2020, 16(11): 7094-7104. DOI: 10.1109/TII.2020.2964563
- [7] 金天骄, 栗蔚. 基于算力网络的大数据计算资源智能调度分配方法 [J]. 数据与计算发展前沿, 2022(6): 29-37
- [8] 袁璐洁, 王目. 区块链赋能的算力网络协同资源调度方法 [J]. 计算机研究与发展, 2023, 60(4): 750-762. DOI: 10.7544/issn1000-1239.20230002
- [9] 李铭轩, 曹畅, 唐雄燕, 等. 面向算力网络的边缘资源调度解决方案研究 [J]. 数据与计算发展前沿, 2020, 2(4): 80-91. DOI: 10.11871/jfdc.issn.2096-742X.2020.04.007
- [10] 刘泽宁, 李凯, 吴连涛, 等. 多层次算力网络中代价感知任务调度算法 [J]. 计算机研究与发展, 2020, 57(9): 1810-1822. DOI: 10.7544/issn1000-1239.2020.20200198
- [11] CHEN L, LINGYS J, CHEN K, et al. AuTO: scaling deep reinforcement learning for datacenter-scale automatic traffic optimization [C]// Proceedings of the 2018 Conference of the ACM Special Interest Group on Data Communication. ACM, 2018: 191-205. DOI: 10.1145/3230543.3230551
- [12] BAI W, CHEN L, CHEN K, et al. PIAS: practical information-agnostic flow scheduling for commodity data centers [C]// Proceedings of IEEE/ACM Transactions on Networking. IEEE, 2017: 1954-1967. DOI: 10.1109/TNET.2017.2669216
- [13] AL-FARES M, RADHAKRISHNAN S, RAGHAVAN B, et al. Hedera: dynamic flow scheduling for data center networks [J]. NSDI, 2010, 10(8): 89-92

作者简介



周旭, 中国科学院计算机网络信息中心先进网络技术与应用发展部主任、研究员; 主要研究领域为未来网络、5G 移动网络、网络人工智能等; 主持科技部、工业和信息化部、国家发展和改革委员会等重大项目 10 余项, 获得部级科学技术奖一等奖 1 项、二等奖 1 项; 发表论文 70 余篇, 申请专利 60 余项 (其中已授权 14 项), 主持/参与制定国际标准/行业标准 11 项。



李琢, 中国科学院大学在读博士研究生; 主要研究方向为未来网络、分布式协同网络、多智能体系统和强化学习。