



# 计算机科学

COMPUTER SCIENCE

## 基于SDN的通存一体化边缘在网存储节点选择方法

叶苗, 王珏, 蒋秋香, 王勇

引用本文

叶苗, 王珏, 蒋秋香, 王勇. 基于SDN的通存一体化边缘在网存储节点选择方法[J]. 计算机科学, 2025, 52(8): 343-353.

YE Miao, WANG Jue, JIANG Qiuxiang, WANG Yong. SDN-based Integrated Communication and Storage Edge In-network Storage Node Selection Method [J]. Computer Science, 2025, 52(8): 343-353.

---

## 相似文章推荐 (请使用火狐或 IE 浏览器查看文章)

Similar articles recommended (Please use Firefox or IE to view the article)

### [基于深度强化学习的在线并行SDN路由优化算法研究](#)

Online Parallel SDN Routing Optimization Algorithm Based on Deep Reinforcement Learning  
计算机科学, 2025, 52(6A): 240900018-9. <https://doi.org/10.11896/jsjcx.240900018>

### [EvoTrace:基于非线性数据包遥测和批处理的轻量级带内网络遥测方法](#)

EvoTrace:A Lightweight In-band Network Telemetry Method Based on Nonlinear Embedding and Batch Processing  
计算机科学, 2025, 52(5): 291-298. <https://doi.org/10.11896/jsjcx.240100164>

### [基于动态贝叶斯博弈的工业控制网络恶意接入检测研究](#)

Study on Malicious Access Detection in Industrial Control Networks Based on Dynamic Bayesian Games  
计算机科学, 2025, 52(1): 383-392. <https://doi.org/10.11896/jsjcx.231200083>

### [SDN中基于统计与集成自编码器的DDoS攻击检测模型](#)

DDoS Attack Detection Model Based on Statistics and Ensemble Autoencoders in SDN  
计算机科学, 2024, 51(11): 389-399. <https://doi.org/10.11896/jsjcx.230900028>

### [基于机器学习的异常流量检测模型优化研究](#)

Study on Optimization of Abnormal Traffic Detection Model Based on Machine Learning  
计算机科学, 2024, 51(6A): 230700051-5. <https://doi.org/10.11896/jsjcx.230700051>

# 基于 SDN 的通存一体化边缘在网存储节点选择方法

叶苗<sup>1</sup> 王珏<sup>1</sup> 蒋秋香<sup>2</sup> 王勇<sup>3</sup>

1 桂林电子科技大学信息与通信学院 广西 桂林 541004

2 桂林电子科技大学光电工程学院 广西 桂林 541004

3 桂林电子科技大学计算机与信息安全学院 广西 桂林 541004

(yemiao@guet.edu.cn)

**摘要** 通常的边缘分布式存储系统将数据存储在多个边缘端服务器上,不仅传输时延受限于数据到边缘服务器的距离,而且服务器节点之间的网络通信管理和配置不够灵活,数据通过网络传输完成边缘存储受到带宽、吞吐量和网络故障等因素的影响,并且在考虑数据存储位置时只考虑存储节点容量,忽略了边缘网络负载和存储节点负载的因素给数据存储效率带来的影响。为解决这些问题,设计了一种通存一体化边缘在网存储架构。该架构融合了软件定义网络(SDN)的灵活性与服务消息块协议(SMB)的高效性,将边缘网络产生的数据存储在网络转发节点,并通过开发定制边缘通存一体化网络交换机实现了所设计的原型系统。首先,利用开发定制一种耦合存储功能的 SDN 交换机作为具备在网存储功能的存储节点,将数据存储到这些网络转发节点上,以有效减少数据传输的网络延迟;然后,通过使用 SDN 技术实时获取网络状态信息和存储节点自身信息,不仅实现了网络传输的动态优化,解决了网络配置管理繁琐的问题,还能以此为基础,建立一种数据存储节点选择的多属性决策模型和设计相应的层次分析求解算法,综合考虑网络状态和节点状态来完成数据在网存储位置的选择;最后,通过系列实验表明,与现有边缘分布式存储系统的数据存储方法相比,所设计和实现的通存一体化边缘在网存储系统能够更灵活地进行网络管理和配置,显著地降低数据存储的时延。

**关键词:** 分布式存储;在网存储;软件定义网络;无线通信;节点选择

**中图分类号** TP393

## SDN-based Integrated Communication and Storage Edge In-network Storage Node Selection Method

YE Miao<sup>1</sup>, WANG Jue<sup>1</sup>, JIANG Qiuxiang<sup>2</sup> and WANG Yong<sup>3</sup>

1 School of Information and Communications, Guilin University of Electronic Technology, Guilin, Guangxi 541004, China

2 School of Optoelectronic Engineering, Guilin University of Electronic Technology, Guilin, Guangxi 541004, China

3 School of Computer Science and Information Security, Guilin University of Electronic Technology, Guilin, Guangxi 541004, China

**Abstract** In conventional edge-distributed storage systems, data is stored across multiple edge servers, where transmission latency is constrained by the distance to the edge servers, and network communication management and configuration between server nodes lack flexibility. Data transfer for edge storage is affected by factors such as bandwidth, throughput, and network failures. Moreover, traditional systems often consider only storage node capacity when selecting storage locations, overlooking the impact of edge network load and storage node load on data storage efficiency. To address these issues, this paper designs a converged edge in-network storage architecture, integrating the flexibility of Software-Defined Networking(SDN) with the efficiency of the Server Message Block(SMB) protocol. The architecture stores data generated within the edge network on certain network forwarding nodes, and the prototype system is implemented through a custom-developed edge-converged network switch. Firstly, a custom SDN switch, coupled with storage functionality, is developed to serve as an in-network storage node, allowing data to be stored on these network forwarding nodes to effectively reduce data transmission latency. Then, using SDN technology to acquire real-time network status and storage node information, dynamic optimization of network transmission is achieved, alleviating the

到稿日期:2024-09-30 返修日期:2024-12-14

基金项目:国家自然科学基金(62161006,62172095);广西研究生教育创新计划(YCSW2023310);“认知无线电与信息处理”教育部重点实验室主任基金项目(CRKL220103);广西无线宽带通信与信号处理重点实验室主任基金项目(GXKL06220110)

This work was supported by the National Natural Science Foundation of China(62161006,62172095), Subsidization of Innovation Project of Guangxi Graduate Education(YCSW2023310), Ministry of Education of Key Laboratory of Cognitive Radio and Information Processing(Guilin University of Electronic Technology)(CRKL220103) and Guangxi Key Laboratory of Wireless Wideband Communication and Signal Processing(GXKL06220110).

通信作者:蒋秋香(jiangqiuxiang@guet.edu.cn)

complexity of network configuration and management. Based on this data, a multi-attribute decision-making model for data storage node selection is established, along with a hierarchical analytical algorithm that considers both network and node states for in-network storage placement. Finally, experimental results demonstrate that, compared to conventional data storage methods in edge-distributed storage systems, the designed and implemented converged edge in-network storage system offers more flexible network management and configuration, significantly reducing data storage latency.

**Keywords** Distributed storage, In-network storage, SDN, Wireless communication, Node selection

## 1 引言

随着无线通信技术<sup>[1]</sup>的不断进步、边缘计算<sup>[2]</sup>理念的兴起与实践,以及物联网技术的日益成熟,智能网络设备的普及和应用在边缘环境中愈发显著,其被广泛应用于智慧医疗<sup>[3]</sup>、智能交通<sup>[4]</sup>、工业物联网<sup>[5]</sup>等领域。这一趋势推动了基于分布式存储系统应用的迅速扩展,并显著提升了移动终端用户对各类存储系统性能与功能的需求。因此,对低时延、低开销、高可靠性的分布式存储系统的研究,成为了当前学术界和工业界共同关注的焦点。

在传统的单机存储系统架构中,系统性能受限于单一服务器的存储容量与处理能力,随着数据量的持续增长,网络带宽的限制导致数据传输速度显著放缓;同时,单一服务器可能无法有效处理所有存储服务,从而成为性能瓶颈。在边缘分布式存储系统架构中,存储服务器通常分布在网络边缘端,长距离的数据传输会增加传输延迟,并且消耗更多带宽;同时,传统的边缘存储系统往往依赖有线部署,这增加了布线的复杂度和部署成本,且移动终端设备难以有效接入网络,使其不适用于动态环境和临时搭建的场景。在云数据中心存储系统架构中,大量数据通常通过网络节点存储至云端服务器<sup>[6]</sup>,这对数据中心的服务器提出了较高的带宽要求,使用云存储方式存在传输时延高、数据不安全和隐私容易泄露的问题;而且紧密的、时刻依赖网络连通的连接,导致网络的部署也不够灵活,极大地限制了边缘网络场景下海量数据的存储<sup>[7]</sup>。

在该背景下,提高边缘数据存储系统的存储性能以应对快速增长的边缘流量,对于缓解存储系统的带宽和存储压力以及保护用户隐私数据,具有重要的理论意义和应用价值。额外部署边缘数据中心不仅增加了成本开销,而且部署和配置繁琐,不够灵活,对此,少量早期关于 WSN 的研究提出过在网存储技术<sup>[8-10]</sup>。本文结合边缘场景设计了一种通存一体化的边缘在网存储架构,解决了其中关于数据存储位置选取的关键问题,并实现了系统原型。在网存储技术<sup>[11]</sup>能将数据存储在网络转发节点中,利用网络设备的缓存和存储能力,充分发挥现代网络硬件低延迟、高带宽的优势。当网络拥塞或者与云端等外网通信发生中断故障时,可以将数据存储在网络转发节点中,从而解决网络传输故障发生时的数据丢失问题。这样的方式可以有效缓解主干网络带宽和数据中心负载的压力,解决节点故障、网络拥塞以及存储资源利用不平衡等问题<sup>[12]</sup>,被认为是解决分布式存储和大规模数据存储环境中节点故障以及网络拥塞问题的有效方法。此外,边缘端设备通常采用无线通信技术接入网络,实现移动终端安全可靠的动态连接。5G 无线通信技术<sup>[13]</sup>解决了传统有线网络部署所面临的安装和维护成本高昂、可扩展性受限以及移动性不足

等问题,存储系统与无线技术的融合<sup>[14-15]</sup>使得通存一体化的无线边缘在网存储的实现成为了可能。

通存一体化在网存储系统中最基本的关键问题是数据存储节点位置选择问题。现有的一些分布式存储系统中,数据存储节点选择方法主要考虑节点存储容量<sup>[16]</sup>、数据相关性<sup>[17]</sup>、数据访问热度<sup>[18]</sup>等因素,这些方法通常仅侧重考虑一些节点方面的单一因素。由于数据存储节点选择涉及数据传输和在存储节点上进行数据读写这两个基本阶段,因此,合理的解决方案不仅需要考虑节点容量等特性,还需要考虑边缘网络的负载和存储节点的负载这两个因素。因此需要设计高效合理的获取网络负载和数据存储节点负载的机制,为节点选择决策提供依据。

传统网络最初是分布式架构,在这种架构下每台设备都具有自己的控制平面和数据平面。控制平面负责路由协议的计算,数据平面则执行数据转发操作。每台设备基于局部的信息(如相邻节点的状态)做出路由判断,各个设备之间缺乏快速的全局信息共享机制,因此网络状态信息测量不够灵活,很难准确及时地获取全局网络负载状态。软件定义网络(Software Defined Network, SDN)的出现简化了传统网络繁琐的网络配置,通过将控制平面和数据平面解耦(控制平面通过控制器管理和收集网络负载信息,数据平面负责数据包的处理和转发),能准确灵活地获取全局网络状态信息,方便管理和配置网络,为优化网络性能提供了基础。不仅如此,借助软件定义架构中关于数据平面上报网络状态信息的机制,可以将存储节点负载信息加载到该网络状态信息中完成同时上报,从而同时完成网络负载信息和节点负载信息的获取,为决策存储节点选择提供依据。

综合考虑以上问题,本文基于 SDN 设计了一种通存一体化边缘在网存储架构,并基于多属性决策数学方法设计了一种边缘在网存储的存储节点选择算法。首先,通过定制耦合存储功能的 SDN 交换机作为具备在网存储功能的存储节点,并通过无线 Mesh 构建网络拓扑;然后,通过 SDN 技术设计了一种状态信息上报机制,实时获取网络负载信息和存储节点负载信息;最后,当客户端进行数据存储的写操作时,基于负载信息建立了多属性决策的节点选择模型并设计了相应的求解算法来确定数据存储的节点位置。

本文的创新点如下:

1) 针对边缘分布式存储在网络拥塞或网络故障时不能满足及时存储的场景需求,设计了一种基于 SDN 的通存一体化边缘在网存储新架构。设计的架构将数据存储到边缘网络转发节点上,能有效减少数据传输的网络延迟,提高应对网络故障场景的系统可靠性。

2) 设计的通存一体化边缘在网存储新架构使用 SDN 技

术实时获取网络状态信息和存储节点自身信息,不仅实现了网络传输的动态优化,解决了网络配置管理繁琐的问题,还以此为基础建立了一种数据存储节点选择的多属性决策模型和设计了相应的层次分析求解算法,综合考虑网络状态和节点状态完成数据在网存储位置的选择,能有效提高边缘数据在网存储的读写性能。

3)基于SDN和SMB协议,实现了所设计的通存一体化边缘在网存储架构下的通存一体化边缘在网存储系统原型。

本文第2章介绍了相关工作;第3章描述了数据的在网存储节点选择问题;第4章介绍了系统架构及算法实现;第5章介绍了实验实现与测试;最后进行总结和展望。

## 2 相关工作

本章主要回顾存储架构和存储节点选择方法的相关工作,对不同架构的特点以及现有在网存储相关工作进行分析。

在网存储是分布式存储系统中的一个重要研究方向,能够根据存储需求对数据进行合理的分配,从而提高数据的存储速度和访问速度。目前的研究主要集中在边缘环境下的分布式存储系统,而对通信与存储一体化的在网存储研究较少。

边缘分布式存储:在边缘环境中,面对爆发式增长的边缘数据,数据存储的高效性通常依赖于边缘服务器存储实现。Qin等<sup>[19]</sup>针对工业场景中的业务系统多而复杂、数据量庞大的问题,提出了云边协同的数据分布式存储架构,设计了基于RDF图模型的数据编码规则和基于S-tree的数据多层次高效存储策略;Bogdanov等<sup>[20]</sup>针对地理分布存储系统中数据中心数据量不断增长的问题,提出一种名为Kurma的分布式存储系统,其允许网络内跨相邻位置部署的复制服务在必要时共享工作负载;Yang等<sup>[21]</sup>提出了一种基于SDN的软件定义边缘存储模式和边缘分布式存储架构,其允许边缘端产生的数据按存储策略分散存储到网络边缘服务器上;Darabseh等<sup>[22]</sup>利用SDN的思想,通过Minnet<sup>[23]</sup>软件模拟构建了软件定义存储(SDStorage)实验框架,将控制层与数据层隔离开来,最终将数据传输到结合了存储原始数据的各种存储设备的基础设施层(云存储和服务器等)。上述存储方案将数据存储在与数据源或用户的边缘服务器或云端上,存储数据的传输路径较长,特别是当多个边缘节点需要互相通信时,容易受到带宽和延迟的限制;而本文提出的在网存储强调的是如何在网络内部进行数据存储,将存储功能嵌入网络节点中,减少了数据传输路径,进一步降低了延迟。

在网存储:目前的在网存储研究工作主要通过网络硬件扩展和存储软件抽象,应对通信和存储一体化在网存储这一挑战。在网络硬件扩展方面,Khan等<sup>[24]</sup>提出了一种用于无线传感器网络数据持久化的自适应、自配置的网络在网存储模型,通过建立一个集群传感器领域,将密集节点部署在路由节点附近,用于在拥塞期间充当数据缓冲区,以避免数据丢失。Seemakhupt等<sup>[25]</sup>提出了一种可编程数据平面PMNet(如交换机或NIC),并在FPGA上实现,用于在网络中持久化数据。PMNet记录传入的更新请求并直接确认客户端,而不需要客户端等待服务器提交请求。在存储软件抽象方面,Wang等<sup>[26]</sup>提出了一种基于信息中心网络(ICN)的网内协同

存储机制,利用转发路径上的多个存储节点分销流量,以增加少量通信开销为代价,提高了网内存储系统整体的写入速率;Nicolaiescu等<sup>[27]</sup>提出了一种通过部署在网络边缘的数据存储库来实现网络内存储管理的框架,通过考虑数据受欢迎程度、数据与边缘处理功能的接近程度等标准,侧重于为原始数据和处理过的数据提供尽可能靠近用户的存储。Gheorghe等<sup>[28]</sup>设计了一种去中心化分散式的存储应用程序,并将该程序作为平台运行在多种类型的边缘节点上,边缘节点通过无线通信环境进行连接,最终的站点是云。Al-Badarneh等<sup>[29]</sup>通过利用软件定义系统(SDS)与网络功能虚拟化(NFV)技术,为存储应用程序提供了一种软件定义移动边缘计算(SD-MEC)支持机制。上述存储方案在网络硬件扩展和存储软件抽象方面提供了不同的创新,但是没有考虑动态网络负载的变化,并且存储软件抽象带来了较高的计算和存储负担,对边缘设备的资源提出了更高要求,现有方案响应速度不够,难以满足快速适应的需求。

存储节点选择:在边缘分布式无线存储系统中,数据被分布在多个节点上,存储节点的选择对于资源分配和确保满足最终用户的QoS要求都至关重要。Zhou等<sup>[30]</sup>提出了一种名为DECS的协作边缘数据存储服务,通过考虑数据的受欢迎程度来提前复制数据,以选择最合适的边缘节点来卸载存储或计算任务;Matri等<sup>[31]</sup>提出了一种动态数据复制算法,通过客户机与站点的位置来进行数据的分配和存储决策,使客户端能够暂时定位最近的数据副本,将存储延迟降至最低;Zhang等<sup>[32]</sup>提出了一种用于边缘计算的多策略通道分配算法(MSCA),通过考虑网络带宽、丢包率等因素进行节点选择,并构建组播树模型,将数据预先存储在组播树的节点中。上述节点选择方法在考虑数据存储位置时只考虑了某一方向的特性,忽略了边缘网络负载和存储节点负载的因素给数据存储效率带来的影响;同时,在动态网络拓扑条件下,实时监测存储节点负载状态和计算能力通常需要添加额外的监控节点来实现,增加了系统部署的成本。

本文设计的基于SDN的通存一体化边缘在网存储架构,在网络中间节点耦合存储模块,为在网存储系统提供硬件支持;创新性地将理论模型与实际场景需求相结合,利用SDN的集中控制和全局视角,获取网络状态信息和节点自身状态信息,以实现更全面的节点选择决策,使之能快速适应网络状态的动态变化,从而更具实用性和适应性;利用无线通信技术,使得移动客户端和存储节点能够使用无线网络与系统交互,大幅度增强了系统的灵活性和可扩展性,整体提高了在网存储系统的存储性能。

## 3 边缘在网存储节点选择问题

在传统边缘分布式存储系统中,通常将存储与网络分离,存储节点只是提供数据存储功能,网络用于数据传输和管理,缺少对彼此状态的实时反馈和协调,容易导致网络传输与存储操作脱节,增加了延迟,并且缺乏灵活性。同时,在存储节点选择方面,传统边缘分布式存储系统主要考虑节点的存储容量,以此计算出存储节点的优先级排序,这往往会导致一些问题。例如,当系统只基于存储容量选择节点时,容易导致某

些高容量节点频繁被选中,而其他节点被忽略,这种不均衡的负载分布进而导致部分节点过载,影响数据存取性能;此外,容易忽视网络性能带来的影响,即使存储节点有足够的容量,若该节点与用户之间的网络延迟较高或带宽不足,数据访问速度也会受到显著影响,增加了整体系统的响应时间。

基于以上描述,本文要解决的问题是:在边缘环境下的移动分布式存储系统中,如何在提出的通存一体化的在网存储架构下最优地将边缘数据存储至不同的节点,从而保证数据传输的速率及可靠性等指标,以满足客户端的需求。

假设由  $N$  个节点通过组网的形式组成分布式存储集群部署在边缘端,通过 SDN 控制器实时收集网络数据,并监听和响应客户端发来的数据存储请求。在这个过程中,存储集群的网络拓扑可以用  $G=(V, E, X)$  表示。其中,  $V=(v_1, v_2, v_3, \dots, v_N)$  表示网络中节点的集合,  $v_i$  表示每个节点的唯一标识;  $E=\{e_{ij} | v_i, v_j \in V\}$  表示点间相互通信的边集合,例如  $e_{ij}$  表示节点  $i$  与节点  $j$  之间相互通信的边;  $X$  为边属性集合。

形象的通存一体化边缘在网存储过程如图 1 所示。边缘端产生的数据需要进行存储时,首先需要对所传输的文件分析后进行文件切块操作,其操作决策取决于多属性决策节点选择算法的结果。假设需要存储大小为  $F$  的边缘数据,在存储集群  $G$  中收集网络链路信息和节点信息,包括带宽、传输时延、丢包率、节点剩余内存等,利用节点选择方法综合考虑以上因素,选择出了 3 个最优节点  $\{v_1, v_2, v_3\}$ ,其切块权重由 3 个节点得分决定。最后,为切块后的文件添加文件标识符以保证数据读取时的一致性,并将其通过网络存储到符合条件的存储节点中。在这个过程中,存储集群中的节点在充当存储角色的同时又兼具数据通信功能,所以如何通过网络硬件拓展实现网络节点的通信与存储一体化,也是在网存储要解决的问题。

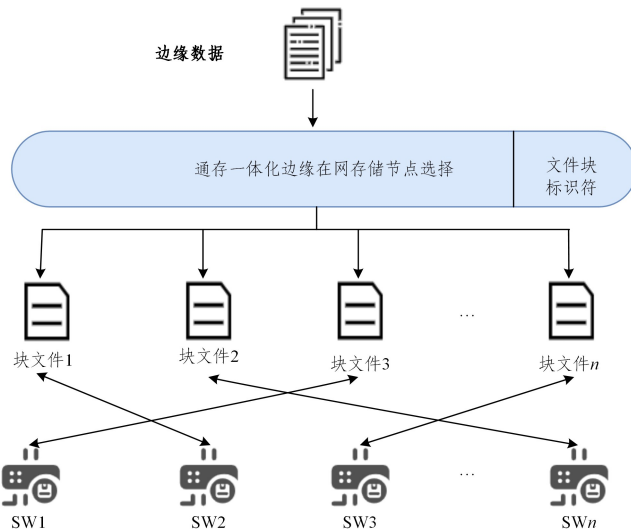


图 1 通存一体化边缘在网存储过程

Fig. 1 Process of integrated communication and storage in the edge in-network storage system

控制器在响应客户端数据存储请求时,通过特定的策略选择存储节点进行数据存储操作。因此,本文基于多属性决策模型对存储集群中的节点选择问题进行建模,其主要思想

是,综合考虑节点与邻接节点间的边属性与节点本身的状态信息,选择得分较高的节点作为存储节点,实现边缘数据在网存储,提升数据存储效率。

在网络拓扑  $G$  中,对于每一个节点  $v_i$ ,用  $B, D, L$  分别表示与邻接节点链路的剩余带宽、时延和丢包率的属性,综合考虑带宽、时延和丢包率后,通过加权求和函数,求取综合网络状态  $S$ ,如式(2)所示。

$$x' = \frac{x - \min(X)}{\max(X) - \min(X)} \quad (1)$$

$$S = \omega_B B' + \omega_D (1 - D') + \omega_L (1 - L') \quad (2)$$

首先,使用式(1)对网络负载信息进行归一化处理。其中,  $x'$  表示属性归一化值,  $x$  表示某一具体属性值,  $\min(X)$  和  $\max(X)$  表示边的属性集合中该属性的最小值与最大值,各属性归一化后的结果分别为带宽  $B'$ 、延迟  $D'$  以及丢包率  $L'$ 。在式(2)中,  $\omega_B, \omega_D, \omega_L$  分别表示链路的剩余带宽、延迟和丢包率的权重,在综合网络状态中,由于带宽与时延和丢包率呈负相关,带宽越大越好,时延和丢包率越小越好,因此需要对时延和丢包率进行处理,使得综合状态的数值越大越好,使用  $1 - D'$  和  $1 - L'$  将式(2)的结果转换为一个越大越好的指标。

综合网络状态  $S$  这一评分用于指导存储节点的选择决策,通过考虑节点之间的网络属性,确保在存储数据时选择网络性能更好的节点。

除了考虑网络状态,SDN 控制器还可以通过设计的节点自身状态信息的上报机制,将节点的 CPU 利用率  $C$ 、内存利用率  $M$  和硬盘剩余容量  $H$  作为存储节点选择决策的指导因素。通过设计合理的映射机制  $map$ ,获取节点  $v_i$  的评价得分  $score_i$ ,如式(3)所示。

$$score_i = map(S, C, M, H) \quad (3)$$

其中,  $map$  为设计的存储节点选择方法,通过考虑节点的多个因素进行比较决策,即可得到节点选择结果,具体实现在 4.3 节中进行详细描述。

最后,通过得到的节点得分列表,对大数据文件进行分割操作,并将分割后的块文件存储在对应的存储节点上,如式(4)所示。

$$F_i = F \cdot \frac{score_i}{\sum_{j=1}^z score_j} \quad (4)$$

其中,  $F$  为大数据文件的总大小,  $F_i$  为存储节点  $v_i$  分配的待存储的数据量,  $z$  为分块数量。

在通信与存储一体化的在网存储架构下,控制器响应客户端存储请求,通过节点选择模型做出决策后计算得到最优存储节点,再进行数据分布式存储操作。选定的存储节点距离客户端的网络距离、网络带宽和路由跳数及节点自身状态都将直接影响数据传输延迟。本文提出的边缘在网存储架构将存储服务迁移至边缘网络中,相对于传输至边缘服务器的存储服务,减少了传输路由跳数,从而提高了响应速度;并结合存储节点选择方法,综合考虑网络因素与节点自身因素,避免了过多的并发请求带来的带宽压力,确保了系统在高负载时仍能提供良好的存储性能。

## 4 基于 SDN 的通存一体化边缘在网存储架构设计与存储节点选择算法设计

本章首先介绍通存一体化边缘在网存储系统的总体架构,然后介绍系统的工作原理和流程,最后介绍存储节点选择算法。

### 4.1 系统架构设计

存储系统中,存储的数据需要通过通信网络进行访问和

传输。无论是从存储节点到客户端,还是在不同存储节点之间的数据迁移,通信网络都是必不可少的。存储系统的性能(如数据读写速率)在很大程度上依赖于通信网络的带宽,通信延迟也会直接影响存储请求的响应时间,因此,在设计网络和存储架构时,需要综合考虑通信和存储之间的关系,以确保系统在性能、可靠性上的平衡。

本文提出的原型系统架构主要由控制平面、决策平面、数据处理平面和数据转发与存储平面构成,其完整框架如图 2 所示。

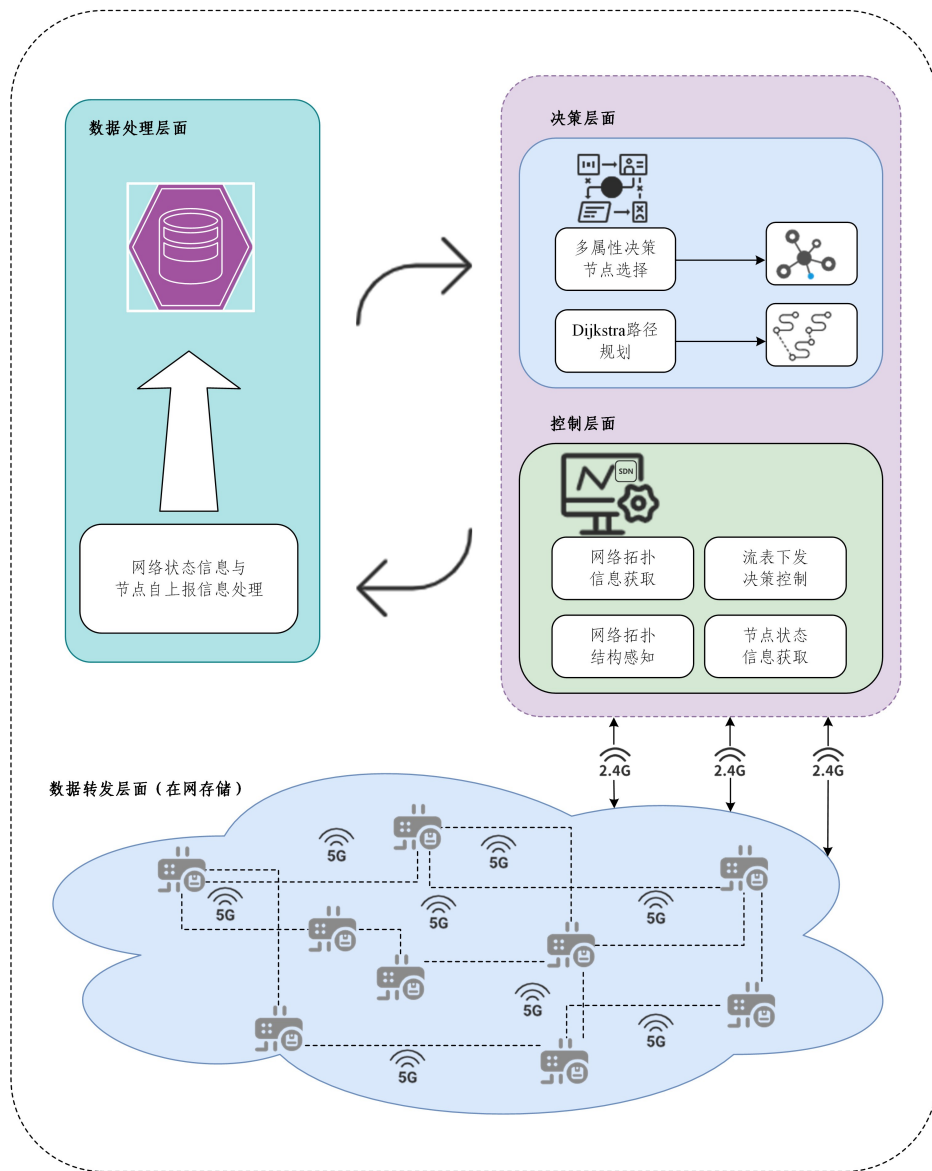


图 2 通存一体化边缘在网存储系统总架构

Fig. 2 Overall architecture of the integrated communication and storage edge in-network storage system

控制层面和决策层面是本设计的核心模块,分别包含控制模块和决策模块。控制模块具有网络拓扑感知功能、节点状态信息监测功能、网络状态信息监测功能和流表下发功能,控制平面通过南向接口对数据平面进行控制和管理,通过解析来自数据平面的数据包获取交换机状态信息来构建全局网络拓扑,从而进一步实现网络资源调度。决策模块具有最优存储节点选择功能和路径计算功能,通过获取数据处理层面存储的网络状态信息和节点状态信息,利用多属性决策算法选出最优存储节点,通过 Dijkstra 算法<sup>[33]</sup>综合考虑链路的

时延、带宽和丢包率获取最短转发路径,并将决策结果对应数据包通过控制模块发送到数据转发层和存储层面并执行决策。控制层面利用 SDN 在控制平面和数据平面解耦合和集中控制的特性,通过控制模块来实现对全局的管理,完成存储服务在通信网络中所需的必要工作。

数据处理平面具有对 SDN 网络的实时状态、流量和各个交换机设备自身状态的处理能力,以便控制器能够进行有效的网络管理和优化。该平面通过分析来自交换机端口的原始数据包信息,计算链路的剩余带宽、延迟和丢包率,为存储服

务提供必要的调度资源,这些调度资源被实时、周期性地采集和处理;然后将计算得到的网络状态信息和节点状态信息进行存储并提供给决策平面,决策平面解析后做出相应的响应决策。

数据转发与存储层面是本设计中硬件部分的创新组成模块,由集成存储功能的 SDN 交换机组成,负责数据的转发以及数据的存储,是在网存储系统的基础。其中,每个交换机作为无线接入节点(Access Point, AP)。5 GHz 频段具有更高的带宽,适合承载大数据量的传输需求,Mesh 网络的自组织和自修复特性使得网络拓扑更加灵活和可靠,其适应性强,可以在网络节点移动或者失效时自动调整连接,所以本文设计通过 5G 无线 Mesh<sup>[34]</sup>方式组成一个多跳的网络,在数据层面进行数据传输和存储;2.4 GHz 频段具有更强的稳定性,控制器通过 2.4G 信号与数据层面进行连接,确保了系统控制和管理的可靠性。2.4 GHz 和 5 GHz 频段分别用于控制信号和数据传输,这种信道分离的方法,减少了两个信道之间的干扰,从而提升了通信的稳定性和可靠性。

## 4.2 网络状态信息和节点状态信息的测量

在通存一体化系统中,通信和存储紧密相连,相互制约。通存一体化的目标是通过融合通信和存储功能,优化系统的整体性能,包括降低延迟,以及提升数据传输和存储的效率。在通存一体化的在网存储设计和实现过程中,这些牵制关系要求在性能、时延、资源分配等方面找到合适的平衡点。因此,本文算法需要考虑的关键信息包括:数据平面的链路剩余带宽、时延和丢包率,以及网络节点的 CPU 利用率、内存利用率和硬盘剩余容量。

带宽:数据读写速率依赖于通信网络的带宽,增加存储服务可能会占用更多带宽,影响通信效率;同样,提升通信性能可能会牺牲存储所需的带宽。控制器在链路带宽的计算过程中,首先从交换机 AP1 与 AP2 的端口收集流量统计信息,包括在时间  $t_i$  至  $t_j$  间隔内交换机 AP1 的端口发送字节数  $ap1_{tx}$  和接收字节数  $ap1_{rx}$ ,以及交换机 AP2 的端口发送字节数  $ap2_{tx}$  和接收字节数  $ap2_{rx}$ ,用计算这些数据量的总和除以时间间隔,可以得到链路的实际使用带宽,如式(5)所示。剩余带宽  $B_{availale}$  为从最大带宽  $B_{total}$  中减去实际使用带宽  $B_{used}$ ,如式(6)所示。

$$B_{used} = \frac{ap1_{tx} + ap1_{rx} + ap2_{tx} + ap2_{rx}}{t_j - t_i} \quad (5)$$

$$B_{availale} = B_{total} - B_{used} \quad (6)$$

时延:通信延迟会直接影响存储请求的响应时间,特别是在需要频繁读写的应用中。网络控制器通过交换机之间的 LLDP(Link Layer Discovery Protocol)包和 OpenFlow 的 Echo Request/Reply 消息来测量网络链路的时延。控制器向网络中的每个交换机发送 Echo Request 消息,并记录发送时间;交换机收到 Echo Request 后,会立即回复 Echo Reply;控制器接收到 Echo Reply 后,根据当前时间与记录的发送时间之差,计算得到交换机与控制器的往返时延。同理,每个交换机定期向其他交换机发送 LLDP 包,LLDP 包中包含发送时间戳,当一个交换机收到 LLDP 包时,控制器根据当前时间与 LLDP 包中记录的发送时间之差,计算得到该 LLDP 包

的传输时延。控制器同时获取到两个交换机之间的 Echo 时延和 LLDP 时延后,利用式(7)计算交换机之间的链路时延。

$$D_{ij} = \frac{T_{forward} + T_{reverse} - E_{src} - E_{dst}}{2} \quad (7)$$

其中, $D_{ij}$ 表示从交换机  $i$  到交换机  $j$  的链路时延; $T_{forward}$  和  $T_{reverse}$  分别为从交换机  $i$  到交换机  $j$ ,以及从交换机  $j$  到交换机  $i$  的 LLDP 传输时延; $E_{src}$  和  $E_{dst}$  分别为从交换机  $i$  和交换机  $j$  到控制器的 Echo 往返时延。

丢包率:在数据传输过程中,丢失的数据包占总发送数据包的比例。丢包率影响了网络的可靠性、传输质量和应用性能。在网络链路中,链路的丢包率可以通过从交换机  $i$  到交换机  $j$  和从交换机  $j$  到交换机  $i$  两个方向的丢包率的最大值来表示,这种计算方式可以更全面地反映链路在双向传输中的丢包情况,如式(8)所示。

$$L_{ij} = \max\left(1 - \frac{apj_{rx}}{api_{tx}}, 1 - \frac{api_{rx}}{apj_{tx}}\right) \quad (8)$$

其中, $api_{tx}$  为交换机  $i$  在特定时间内发送的总字节数, $apj_{rx}$  表示交换机  $j$  接收到的总字节数, $apj_{tx}$  为交换机  $j$  在特定时间内发送的总字节数, $api_{rx}$  为交换机  $i$  接收的总字节数。

文献[21]巧妙地利用 Packet-in 机制,将存储节点的自身状态信息上报给控制器。针对网络内存存储节点的选择,本文考虑了节点的异构性,与文献[21]类似,将设计的节点自身状态信息的上报机制部署在网络内节点。数据平面的每个 SDN 交换机获取自身的 CPU 利用率、内存使用情况和硬盘剩余容量,然后构造一个包含自身状态信息的数据包,并将该数据包的目的地址设置为空。由于该包无法与已有交换机的流表项进行匹配,会立即触发 Packet\_in 机制将其发送给 SDN 控制器,控制器接收到该 Packet-in 事件时,会解析并提取网络节点的状态信息。

## 4.3 算法设计

本节根据第 3 章对通存一体化边缘在网存储节点选择问题的描述,利用全局网络拓扑和节点负载信息,并基于多属性决策模型设计了相应的层次分析<sup>[35]</sup>求解算法。

在现代网络架构中,通信链路的带宽、时延和丢包率是影响存储服务性能的关键因素。带宽决定了存储服务的最大数据传输能力;时延是数据从源头到目标的传输时间,高时延会导致用户在使用存储服务时等待较长的时间;丢包率则表示在传输过程中丢失的数据包比例,较高的丢包率会导致数据不完整,影响数据的可靠性。综合考虑这 3 个指标,可以用综合得分  $S$  表示通信链路性能。节点  $v_i$  的综合状态  $S_i$  为所有邻接节点  $j$  的综合状态  $S$  的平均值,如式(9)所示。

$$S_i = \frac{1}{n} \sum_{j \in v_i} S \quad (9)$$

其中, $n$  表示节点  $v_i$  的度,即与该节点直接相连的边的数量。

然后,考虑节点自身状态对存储性能的影响,包括节点的 CPU 利用率、内存利用率以及硬盘剩余容量,将这些因素作为最优存储节点选择指标。使用比例标度表来量化决策元素之间的相对重要性,如表 1 所列。

表 1 比例标度

Table 1 Proportional scale

数值	定义
1	同等重要
2	介于同等重要与稍微重要之间
3	稍微重要
4	介于稍微重要与明显重要之间
5	明显重要
6	介于明显重要与非常重要之间
7	非常重要
8	介于非常重要与极端重要之间
9	极端重要

首先,根据表 1,量化分析各指标对存储节点选择的重要性,如表 2 所列,其中的每一个值代表其横向指标相比于纵向指标的标度。通过表 2 构建判断矩阵  $A$ ,如式(10)所示。

表 2 各指标对在网存储节点选择的重要性分析

Table 2 Analysis of the importance of each indicator for the selection of in-network storage nodes

	综合 网络状态	CPU 利用率	内存 利用率	硬盘 剩余容量
综合网络状态	1	4	5	2
CPU 利用率	1/4	1	3	1/4
内存利用率	1/5	1/3	1	1/5
硬盘剩余容量	1/2	4	5	1

$$A = \begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \cdots & \cdots & \ddots & \cdots \\ a_{n1} & a_{n2} & \cdots & a_{nm} \end{pmatrix} \quad (10)$$

其中, $n$ 表示指标的数量; $a_{ij}$ 表示与指标  $j$  相比,指标  $i$  的重要程度,即两两之间的重要性的比较。当  $i=j$  时,两个指标相同,因此同等重要记为 1,且需要满足条件  $a_{ij} > 0$  且  $a_{ij} \times a_{ji} = 1$ ,这种关系确保了判断矩阵的逻辑一致性。

同理,需要分析存储节点的综合网络状态、CPU 利用率、内存利用率、磁盘剩余容量 4 个指标在各个节点上的表现情况,并结合 SDN 控制器实时获取的状态值进行重要性分析,根据表 1 构建判断矩阵。然后对判断矩阵进行一致性检验,以确认判断矩阵合理,无逻辑上的错误,如式(11)所示。

$$CR = \frac{\lambda_{\max} - n}{RI(n-1)} \quad (11)$$

其中, $\lambda_{\max}$ 表示判断矩阵的最大特征值; $n$ 表示节点的数量; $RI$ 表示随机一致性指标,如表 3 所列。如果  $CR < 0.1$ ,则可认为判断矩阵的一致性可以接受;否则需要对判断矩阵进行修正。

表 3 平均随机一致性指标 RI

Table 3 Average random consistency index RI

$n$ 阶	3	4	5	6	7	8	9
RI 值	0.52	0.89	1.12	1.26	1.36	1.41	1.46

最后,通过式(12)计算判断矩阵的权重向量,用于构建权重矩阵。

$$\omega_i = \frac{1}{m} \sum_{j=1}^n \frac{a_{ij}}{\sum_{k=1}^m a_{kj}}, i=1,2,\dots,m \quad (12)$$

其中, $\omega_i$ 为第  $i$  个准则的权重, $m$ 为判断矩阵的维度, $a_{ij}$ 表示判断矩阵中第  $i$  行第  $j$  列的元素, $\sum_{k=1}^n a_{kj}$ 表示判断矩阵中第

$j$  列所有元素的和。

根据权重矩阵计算得分,式(13)所示:

$$score_i = \sum_{j=1}^n \omega_j x_{ij} \quad (13)$$

其中, $score_i$ 为节点  $i$  的得分, $\omega_j$ 是第  $j$  个准则的权重, $x_{ij}$ 是第  $i$  个节点在第  $j$  个准则下的权重。

权重矩阵的计算过程如图 3 所示。

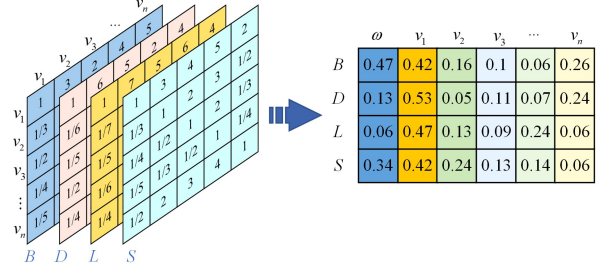


图 3 权重矩阵计算过程

Fig. 3 Process of calculating the weight matrix

首先,计算综合网络状态得分  $S$ ,如算法 1 所示。

### 算法 1 综合网络状态得分算法

输入:网络拓扑的图结构数据 graph、链路属性的加权因子  $\omega_B, \omega_D, \omega_L$

输出:网络状态得分字典 node\_score

1. 初始化:空字典 network\_status;
2.  $\max_{bandwidth}, \max_{delay}, \max_{loss}, \min_{bandwidth}, \min_{delay}, \min_{loss} = (\max(\text{边的带宽属性 for 边的属性字典 in graph'edges}), \max(\text{边的延迟属性 for 边的属性字典 in graph'edges}), \max(\text{边的丢包率属性 for 边的属性字典 in graph'edges}), \min(\text{边的带宽属性 for 边的属性字典 in graph'edges}), \min(\text{边的延迟属性 for 边的属性字典 in graph'edges}), \min(\text{边的丢包率属性 for 边的属性字典 in graph'edges}));$
3. for 节点 in graph'node 列表 do
4. 初始化节点网络状态属性得分 node\_status;
5. 获取节点的邻居节点列表;
6. for 节点 in 该节点的邻居节点列表 do
7. 获取节点与该邻居节点链路的剩余带宽 bandwidth、时延 delay 和丢包率 loss 属性;
8.  $bandwidth' = \frac{bandwidth - \min\_bandwidth}{\max\_bandwidth - \min\_bandwidth}$
9.  $delay' = \frac{delay - \min\_delay}{\max\_delay - \min\_delay}$
10.  $loss' = \frac{loss - \min\_loss}{\max\_loss - \min\_loss}$
11. 边的综合得分 =  $\omega_B * bandwidth' + \omega_D * (1 - delay') + \omega_L * (1 - loss')$ ;
12. 总分 += 边的综合得分;
13. end for
14. 在 node\_score 中节点的最终综合得分 = 总分 ÷ 节点的邻居节点数量
15. end for
16. return 网络状态得分字典 node\_score

将控制器获取到的网络拓扑的图结构数据和链路属性的加权因子作为算法的输入,输出每个节点的综合链路状态得分字典。第 1 行是初始化节点的网络状态得分字典,每个节点对应一个得分值。第 2 行是获取网络中的所有边,计算并存储剩余带宽、时延和丢包率的最大值和最小值,方便后续的

归一化计算。第 3—14 行是获取最终综合得分,其中,第 4 行是对每个节点,初始化节点的网络状态属性得分;第 5 行是获取该节点的所有邻居节点列表;第 6—10 行是对每个邻居节点,获取链路的剩余带宽、时延和丢包率,并进行归一化处理;第 11 行和第 12 行是计算链路的综合得分,并将与该节点连接的所有链路的综合得分相加求和。最后,将总得分除以邻居节点的数量,得到该节点对应的最终的综合网络状态得分,并将该得分存储在字典中。

获取到每个网络节点的网络状态得分后,将节点 CPU 利用率、节点内存利用率以及节点硬盘剩余容量作为存储节点选择的评价指标,通过 SDN 控制器获取的实时状态信息进行两两比较后构造判断矩阵,经过一致性检验后,对判断矩阵进行归一化处理,计算得到该矩阵的权重向量,并构建权重矩阵,根据权重矩阵计算出每个存储节点的得分,选取最优存储节点。存储节点选择算法的详细实现如算法 2 所示。

#### 算法 2 多属性决策存储节点选择算法

输入:网络拓扑的图结构数据 *graph*、节点综合网络状态 *S*、节点 CPU 利用率 *C*、节点内存利用率 *M*、节点硬盘剩余容量 *H*

输出:节点的评价得分字典 *node\_score*

1. 获取节点列表 *node\_list*
2. 获取节点数量 *n*
3. 初始化判断矩阵 *judgment\_matrix*
4. 初始化权重矩阵 *weight\_matrix*
5. for *i* in 节点列表 *node\_list* do
6.   for *j* in 节点 *i* 之后的每一个节点 *do*
7.     标度确定和构造判断矩阵 *A*
8.   end for
9. end for
10. 计算判断矩阵的特征向量和最大特征值  $\lambda_{\max}$
11. While True do
12.   判断矩阵的一致性检验  $CR = \frac{\lambda_{\max} - n}{RI(n-1)}$
13.   if  $CR < 0.1$  then
14.     Berak
15.   end if
16.   修正判断矩阵,更新
17. end while
18. 计算判断矩阵的权重向量  $\omega_i = \frac{1}{n} \sum_{j=1}^n \frac{a_{ij}}{\sum_{k=1}^n a_{kj}}$
19. 构建权重矩阵
20. 根据权重计算得分并排序

第 1—4 行是通过网络拓扑的图结构数据 *graph* 获取节点列表和节点数量 *n*,并初始化判断矩阵 *judgment\_matrix* 和权重矩阵 *weight\_matrix*;第 5—9 行,对同一层次各元素关于上一层次中某一准则的重要性进行两两比较,再根据比例标度表构造判断矩阵;第 10 行,求取判断矩阵的特征向量和最大特征值;第 11—17 行,是对判断矩阵进行一致性检验及修正,用于确认判断矩阵的合理性,不存在逻辑上的错误;第 18 行和第 19 行是计算判断矩阵的权重向量,并构建对应的权重矩阵。通过以上步骤,最终可以确定数据转发与存储平面中每个存储节点的优劣排序。

## 5 实验及结果分析

本实验旨在验证基于 SDN 和多属性决策的通存一体化边缘在网存储节点选择算法的有效性和性能。与现有的传统存储系统不同,本文针对边缘端场景进行了硬件设计和选型,选择的硬件设备在性能和功能上更符合边缘端应用的需求。通过这种针对性的硬件配置,本文实验系统在资源利用和数据存储效率上相较于传统存储边缘分布式存储系统有显著优势,更加贴合边缘端应用的实际需求。实验使用的网络拓扑如图 4 所示。

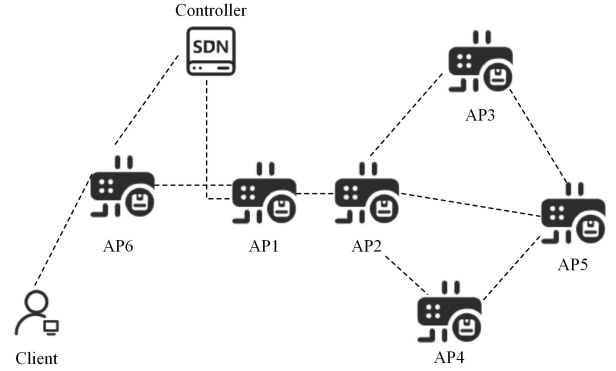


图 4 网络拓扑

Fig. 4 Network topology

### 5.1 实验环境及硬件设计

该原型系统中共有 1 台控制器和 6 台 SDN 交换机;软件环境为:OpenWrt 版本号 23.01,Open vSwitch 版本号 2.15.0,Ryu 控制器版本号 4.8,OpenFlow 版本号 1.3,Samba 版本号 4.13,控制器操作系统为 Ubuntu 20.04 LTS,固态硬盘型号为 Nvme SSD PCIE 3.0 1TB。

在 SDN 中,控制器的主要作用是集中管理和控制网络设备的配置和数据流动,通过下发流表和策略来实现网络的智能化和自动化管理。设计最终选择畅网微控 N5105 六网口软路由作为 RYU 控制器的硬件载体。该软路由由板载  $4 * \text{Intel i226-V } 2.5\text{G}$  网卡,支持 HDMI 2.0b+DP 1.4a+Type-C 3.1 显示输出,且体积仅为  $17.8\text{cm(长)} \times 12.5\text{cm(宽)} \times 5\text{cm(高)}$ 。N5105 软路由实物如图 5 所示。



图 5 N5105 软路由实物图

Fig. 5 N5105 soft router physical diagram

SDN 交换机的作用是根据控制器下发的流表规则,执行数据包的转发和处理,同时收集流量统计信息,并通过与控制器的通信,实现网络的动态配置和智能化管理。在本文设计的通存一体化边缘在网存储系统中,将数据转发平面与数据存储平面进行耦合,以此构建在网存储架构原型。在本文实验中,设计的 SDN 交换机硬件配置如下:处理器为 MediaTek

MT7621,系统型号为 Netgear WNDR3700-v5,配备 1TB NVMe SSD PCIe 3.0 硬盘以提供高速存储,支持 USB 3.0 接口和串口通信;此外,交换机还支持双频无线网络通信。实验用 SDN 交换机的实物如图 6 所示。

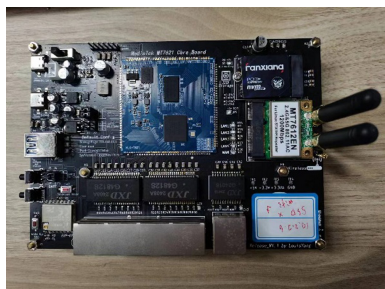


图 6 耦合存储功能的 SDN 交换机

Fig. 6 Physical diagram of an SDN switch with coupled storage function

耦合存储功能的 SDN 交换机的配置步骤如下:首先,在 Linux 环境下编译 OpenWrt 23.05 固件,集成所需的应用程序,如 Open vSwitch、Openflow 协议、Nvme 协议、Samba 协议和 SSH 协议;接着,使用 CH341 编程器将 U-Boot 引导程序烧录到硬件系统中,并通过 U-Boot 将编译好的 Squashfs-sysupgrade 固件传输并安装;最后,配置烧录好的系统,设置 SDN 交换机,通过添加虚拟网桥、挂载物理网卡、配置接口及 IP 地址,将路由器配置为一个具备存储功能的 SDN 交换机。

## 5.2 实验结果及分析

在本项实验中,为了进行存储性能的对比,选择了与在网存储(INWS)具有相同配置的传统边缘无线存储系统(TEWS)和边缘分布式无线存储系统(EDWS),并针对不同数据大小的文件(500 MB,1 000 MB,2 000 MB)进行了存储性能测试。在数据存储过程中,各参与节点持续监控并更新其性能指标,包括节点综合网络状态、剩余存储空间、CPU 利用率、内存利用率等。此外,实验采取并行处理机制,以确保数据存储的高效性和实时性。

实验设计包括两种不同的操作条件,分别在节点正常操作状态和节点高负载操作状态下进行了各 10 组的存储测试。在这两种状态下,数据存储的传输时间是通过记录时间戳来测量的,并将 10 次测试的平均传输时间作为本研究性能分析的关键评估指标。

5 GHz 频段在大多数情况下可提供更高的存储效率,尤其是在需要高速传输数据的应用场景中;2.4 GHz 频段在信号穿透力和稳定性方面具有优势,特别是在干扰较多或环境复杂的情况下。本实验首先考查了客户端在不同的无线频段环境对存储性能的影响,分别包括在 2.4 GHz 频段、5 GHz 频段条件下的数据存储操作。图 7 展示了客户端使用不同频段信号与交换机相连时的存储 10 次 1 000 MB 数据的平均传输时间。实验结果显示,INWS 的平均传输时间明显少于 TEWS 与 EDWS 的平均传输时间,在 5 GHz 频段下的数据传输明显优于在 2.4 GHz 频段下的数据传输。本实验通过使用 5G Mesh 组网的方式进行数据传输,以最大化网络的传输效率,使得在网存储系统的优势得以充分展现。

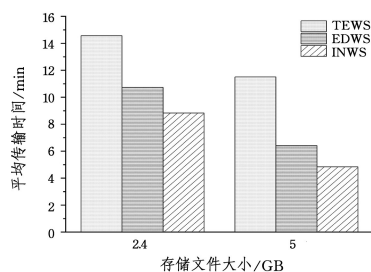


图 7 2.4 GHz 频段和 5 GHz 频段数据传输时间的对比

Fig. 7 Comparison of data transmission time between 2.4 GHz band and 5 GHz band

在网络状态正常的情况下进行数据存储实验。图 8 显示了对不同大小的数据进行写入的时间统计数据,其中的横坐标代表的是不同大小的数据,纵坐标代表写入数据的平均传输时间。从实验结果可以看出,在边缘环境下,相比 TEWS 和 EDWS,INWS 的数据写入时间最短,随着数据量的增加,INWS 的优势开始显现,得益于 SDN 技术的网络控制和数据层分离、数据层和存储功能耦合架构和高效的存储节点选择算法,INWS 能够更快地将数据就近存储在靠近数据生成或传输路径上的节点中,缩短了数据在网络中的传输距离,避免某些节点因承担过多存储或通信任务而成为瓶颈,从而降低了延迟,能够更快、更稳定地完成数据传输;TEWS 在存储节点选择方面仅考虑了容量单一因素,链路及存储节点性能影响了其传输效率;EDWS 存储系统可能会因为网络负载增加而表现出较高的传输延迟和较低的吞吐量。

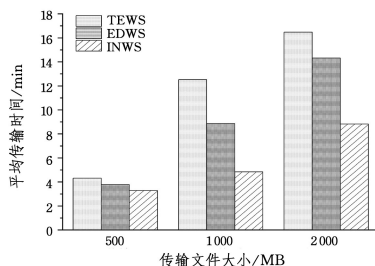


图 8 正常状态下边缘数据存储时间的对比

Fig. 8 Comparison of edge data storage time in normal state

为了观察在节点高负载情况下存储系统的表现,对比 INWS 与 TEWS 和 EDWS 的性能。本文在测试过程中,在网络内 AP2,AP3 节点中进行数据备份操作以增加节点 CPU 和内存的利用率;同时,使用 iPerf 工具在 AP1 与 AP3 之间随机发送 1 MB,3 MB 和 5 MB 的数据流以适度增加负载。图 9 为增加节点负载状态下的边缘数据存储时间对比图。

如图 9 所示,在节点负载增加的情况下,INWS 的传输时间均少于 TEWS 和 EDWS 的传输时间,可以表现出更好的稳定性。随着网络节点负载的增加,当 TEWS 和 EDWS 数据传输至特定的边缘服务器时,通信和存储都会争抢有限的带宽资源,存储任务和通信任务的延迟也会增大,从而影响数据传输速度;而网络控制与数据存储分离的架构使得 INWS 可以更高效地管理和调度网络资源,优化传输路径和传输速度,并且 INWS 在负载增加的情况下,数据可以在网络内的多个存储节点间进行动态调度,传输时间的增加

幅度通常小于传统边缘存储系统。

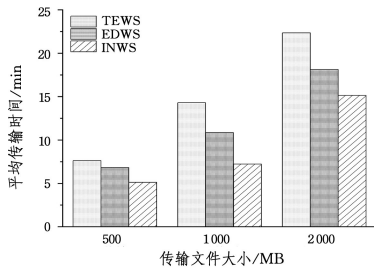


图9 负载状态下边缘数据存储时间的对比

Fig. 9 Comparison of edge data storage time under load

通过上述实验可以得出,本文提出的基于SDN的通存一体化边缘在网存储节点选择方法,其数据存储效率在不同对比系统下都是最优的;并且随着网络负载的增加,INWS能够维持稳定的存储表现。此外,存储节点选择方法对数据存储传输时间有显著影响,SDN可以高效地协调网络资源分配,优化系统性能。同时,结合节点选择、数据分片等策略,SDN能够显著提升通存一体化在网存储系统的效率和稳定性,从而在负载增加时实现更好的通信与存储协作。

**结束语** 本文设计了一种基于SDN的通存一体化边缘在网存储架构,并搭建了原型实物系统。通过引入SDN的集中控制和全局视图能力,系统能够动态地获取网络负载信息和节点负载信息,通过多属性决策算法智能地选择最优存储节点,实现了更高效的数据传输和存储。对原型实物系统的存储性能测试表明,与传统的边缘分布式存储系统相比,本文设计的存储系统能够动态调度和分配系统资源,将数据存储在网络内的最优存储节点上,减少了数据传输时延和网络拥塞,具有更优的存储性能。此外,在实际边缘场景中,在网存储系统能够更灵活地进行部署和调整,易于维护和管理。未来的研究可以探索更加高效的跨域数据的在网存储策略,确保系统在大规模分布式环境中的可扩展性。

## 参考文献

[1] WU Y, GAO X, ZHOU S, et al. Massive access for future wireless communication systems [J]. *IEEE Wireless Communications*, 2020, 27(4): 148-156.

[2] QIU T, CHI J, ZHOU X, et al. Edge computing in industrial internet of things: Architecture, advances and challenges [J]. *IEEE Communications Surveys & Tutorials*, 2020, 22(4): 2462-2488.

[3] AL-TURJMAN F, NAWAZ M H, ULUSAR U D. Intelligence in the Internet of Medical Things era: A systematic review of current and future trends [J]. *Computer Communications*, 2020, 150: 644-660.

[4] VERMA S, ZEADALLY S, KAUR S, et al. Intelligent and secure clustering in wireless sensor network (WSN)-based intelligent transportation systems [J]. *IEEE Transactions on Intelligent Transportation Systems*, 2021, 23(8): 13473-13481.

[5] URREA C, BENÍTEZ D. Software-defined networking solutions, architecture and controllers for the industrial internet of things: A review [J]. *Sensors*, 2021, 21(19): 6585.

[6] TAHERKORDI A, ZAHID F, VERGINADIS Y, et al. Future cloud systems design: challenges and research directions [J].

*IEEE Access*, 2018, 6: 74120-74150.

[7] HAZRA A, RANA P, ADHIKARI M, et al. Fog computing for next-generation internet of things: fundamental, state-of-the-art and research challenges [J]. *Computer Science Review*, 2023, 48: 100549.

[8] MEKKI K, DERIGENT W, RONDEAU E, et al. In-network data storage protocols for wireless sensor networks: A state-of-the-art survey [J]. *International Journal of Distributed Sensor Networks*, 2019, 15(4): 1-22.

[9] XU J, TANG X, LEE W C. EASE: an energy-efficient in-network storage scheme for object tracking in sensor networks [C]// *SECON*. 2005: 396-405.

[10] FASOLO E, ROSSI M, WIDMER J, et al. In-network aggregation techniques for wireless sensor networks: a survey [J]. *IEEE Wireless Communications*, 2007, 14(2): 70-87.

[11] ALIMI R, RAHMAN A, YANG Y. A survey of in-network storage systems [R]. New Haven: Yale University, 2011.

[12] WANG Q, LI J R, SHU J W. Survey on In-Network Storage Systems. [J]. *Journal of Computer Research and Development*, 2023, 60(11): 2681-2695.

[13] ZHU H, JIANG W, HONG Q, et al. When In-Network Computing Meets Distributed Machine Learning [J]. *IEEE Network*, 2024, 38(5): 238-246.

[14] TIAN C J, XIE J. Research Developments of 5G Network Slicing [J]. *Computer Science*, 2023, 50(11): 282-295.

[15] KAZMI S H A, QAMAR F, HASSAN R, et al. Survey on Joint Paradigm of 5G and SDN Emerging Mobile Technologies: Architecture, Security, Challenges and Research Directions [J]. *Wireless Personal Communications*, 2023, 130(4): 2753-2800.

[16] WEIL S A, BRANDT S A, MILLER E L, et al. CRUSH: controlled, scalable, decentralized placement of replicated data [C]// *Proceedings of the 2006 ACM/IEEE Conference on Supercomputing*. ACM, 2006: 122.

[17] YUAN D, YANG Y, LIU X, et al. A data placement strategy in scientific cloud workflows [J]. *Future Generation Computer Systems*, 2010, 26(8): 1200-1214.

[18] LIU F, JIANG D J. Heterogeneous Storage Aware Data Placement of Ceph Storage System [J]. *Computer Science*, 2018, 44(6): 17-22.

[19] QIN X C, LIU C Y, LI BAO, et al. Cloud-edge collaborative data storage and retrieval architecture for industrial scenarios [J/OL]. *Journal of Computer Applications*, 1-12. (2024-10-16). <http://kns.cnki.net/kcms/detail/51.1307.TP.20241016.1056.003.html>.

[20] BOGDANOV K L, REDA W, MAGUIRE G Q, et al. Fast and accurate load balancing for geo-distributed storage systems [C]// *Proceedings of the ACM Symposium on Cloud Computing*. ACM, 2018: 386-400.

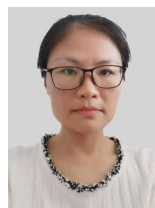
[21] YANG Y, YE M, JIANG Q, et al. A Novel Node Selection Method in Wireless Distributed Edge Storage Based on SDN and Multi-attribute Decision Model [J]. *Electronic Research Archive*, 2024, 32(2): 1160-1190.

[22] DARABSEH A, AL-AYYOUB M, JARARWEH Y, et al. SDStorage: A Software Defined Storage Experimental Frame-

- work[C]//2015 IEEE International Conference on Cloud Engineering. 2015:341-346.
- [23] DE OLIVEIRA R L S, SCHWEITZER C M, SHINODA A A, et al. Using mininet for emulation and prototyping software-defined networks[C]//2014 IEEE Colombian Conference on Communications and Computing(COLCOM). IEEE, 2014:1-6.
- [24] KHAN M I, GANSTERER W N, HARING G. In-network storage model for data persistence under congestion in wireless sensor network[C]//First International Conference on Complex, Intelligent and Software Intensive Systems(CISIS'07). IEEE, 2007:221-228.
- [25] SEEMAKHUPT K, LIU S, SENEVIRATHNE Y, et al. PM-Net: In-network data persistence[C]//2021 ACM/IEEE 48th Annual International Symposium on Computer Architecture (ISCA). IEEE, 2021:804-817.
- [26] WANG Y, HAN R, DANG S J. An ICN-based in-network cooperative storage mechanism[J]. *Electronic Design Engineering*, 2024, 32(20):1-5.
- [27] NICOLAESCU A C, MASTORAKIS S, PSARAS I. Store edge networked data (SEND): A data and performance driven edge storage framework[C]//IEEE INFOCOM 2021—IEEE Conference on Computer Communications. IEEE, 2021:1-10.
- [28] GHEORGHE A G, CRECAN A C, NEGRU C, et al. Decentralized storage system for edge computing[C]//2019 18th International Symposium on Parallel and Distributed Computing (ISPD). IEEE, 2019:41-49.
- [29] AL-BADARNEH J, JARARWEH Y, AL-AYYOUB M, et al. Software defined storage for cooperative mobile edge computing systems[C]//2017 Fourth International Conference on Software Defined Systems(SDS). IEEE, 2017:174-179.
- [30] ZHOU F, CHEN H. Decs: Collaborative edge-edge data storage service for edge computing[C]//Collaborative Computing: Networking, Applications and Worksharing; 16th EAI International Conference. Springer, 2021:373-391.
- [31] MATRI P, PEREZ M S, COSTAN A, et al. Keeping up with storage: Decentralized, write-enabled dynamic geo-replication[J]. *Future Generation Computer Systems*, 2018, 86:1093-1105.
- [32] ZHANG D, PIAO M, ZHANG T, et al. New algorithm of multi-strategy channel allocation for edge computing[J]. *AEU-International Journal of Electronics and Communications*, 2020, 126:153372.
- [33] BUZACHIS A, CELESTI A, GALLETTA A, et al. Evaluating an application aware distributed Dijkstra shortest path algorithm in hybrid cloud/edge environments[J]. *IEEE Transactions on Sustainable Computing*, 2021, 7(2):289-298.
- [34] CHAI Y, ZENG X J, LIU Z. The future of wireless mesh network in next-generation communication: a perspective overview[J]. *Evolving Systems*, 2024, 15(4):1635-1648.
- [35] BILANDI N, VERMA H K, DHIR R. AHP-neutrosophic decision model for selection of relay node in wireless body area network[J]. *CAAI Transactions on Intelligence Technology*, 2020, 5(3):222-229.



**YE Miao**, born in 1977, Ph.D, professor, doctoral supervisor. His main research interests include edge storage, cloud storage, software-defined networking, wireless sensor networks, pattern recognition, machine learning, deep reinforcement learning and graph neural network.



**JIANG Qixiang**, born in 1978, engineer. Her main research interests include wireless sensor networks, pattern recognition, machine learning and graph neural networks.

(责任编辑:柯颖)