Ceph 云存储网络中一种业务优先级区分的多播流调度方法

柯文龙[1]，王勇[2]，叶苗[1,3]，陈俊奇[2]

（1. 桂林电子科技大学信息与通信学院，广西 桂林 541004；2. 桂林电子科技大学计算机与信息安全学院，广西 桂林 541004；
3. 桂林电子科技大学认知无线电与信息处理省部共建教育部重点实验室， 广西 桂林 541004）

摘 要: 针对现有流调度方法难以满足 Ceph 云存储网络中多业务流的不同多播调度需求问题，设计了一种支持业务优先级区分的多播流调度方法。首先，采用软件定义网络技术实时获取网络状态信息，为流调度方法提供数据支撑；然后，将待处理的多播流调度任务分解为多个单播路径选择的多属性决策问题，提出基于理想解法的单播路径选择方法，根据业务流对网络性能的需求为其找到一个最优单播路径集；最后，通过各路径集间的最大公共子路径确定多播分发节点以构建多播传输路径。实验结果表明，与现有方法相比，所提方法可以在降低冗余流量、提高网络负载均衡性能的同时，降低高优先级流的传输时延。

关键词：Ceph；云存储网络；多播流调度；软件定义网络

中图分类号：TP393

文献标识码：A

doi: 10.11959/j.issn.1000−436x.2020233

Priority differentiated multicast flow scheduling method in Ceph cloud storage network

KE Wenlong[1], WANG Yong[2], YE Miao[1,3], CHEN Junqi[2]

1. School of Information and Communication, Guilin University of Electronic Technology, Guilin 541004, China
2. School of Computer Science and Information Security, Guilin University of Electronic Technology, Guilin 541004, China
3. Key Lab. of Cognitive Radio and Information Processing, Guilin University of Electronic Technology, Guilin 541004, China

Abstract: In order to solve the problem that existing flow scheduling method is difficult to meet the different multicast scheduling requirements of multi-service flows in the Ceph cloud storage network, a service priority-based multicast flow scheduling method was tailored. First, the network status was obtained via software defined network (SDN) to support flow scheduling. Then, a multicast task was decomposed into multiple attribute decision problems for multiple unicast path selection, and a method of unicast path selection based on technique for order preference by similarity to ideal solution (TOPSIS) was proposed. The unicast path selection method was used to find the optimal unicast path set for the service flow based on the flow's network performance requirements. Then, the multicast distribution node was determined by the maximum common sub-path among the optimal unicast path sets for construct a multicast transmission path. The experiment results show that the proposed method can reduce the transmission delay of high priority flows while reduce the redundant traffic and better balance the traffic loads compared with the existing methods.

Key words: Ceph, cloud storage network, multicasting flow scheduling, SDN

收稿日期：2020−07−20；**修回日期**：2020−10−09

通信作者：叶苗，ym@mail.xidian.edu.cn

基金项目：国家自然科学基金资助项目（No.61861013, No.61662018）；广西创新驱动发展专项_科技重大专项基金资助项目（桂科 AA18118031）；广西自然科学基金资助项目（No.2018GXNSFAA050028）；“认知无线电与信息处理”教育部重点实验室主任基金资助项目（No.CRKL190102）；广西高校中青年教师科研基础能力提升基金资助项目（No.2019KY0822）

Foundation Items: The National Natural Science Foundation of China(No.61861013, No.61662018), Guangxi Innovation-Driven Development Project (Major Science and Technology Project No.AA18118031), Guangxi Natural Science Foundation of China (No.2018GXNSFAA050028), Director Fund Project of Key Laboratory of Cognitive Radio and Information Processing of Ministry of Education(No.CRKL190102), Guangxi Colleges and Universities Basic Ability Improvement Project of Young and Middle-Aged Teachers(No.2019KY0822)

1　引言

全球范围内数据规模的高速增长已成为信息产业界所面临的巨大挑战。根据国际数据公司（IDC, International Data Corporation）的数据统计，全球数字领域数据量将由 2018 年的 33 ZB 增加到 2025 年的 175 ZB[1]。这种数据爆炸式增长所带来的挑战可以使用云存储技术来应对，云存储系统采用数据中心网络技术将大量存储设备集成为统一整体，进而提供海量且可扩展的存储能力[2]。目前，已有多种云存储系统投入商业使用，包括 Ceph[3]、OpenStack Swift、Dropbox 以及 Google Drive 等。其中，Ceph 因其稳定的架构、开源的思想和统一存储的设计模式，得到了越来越多云存储系统使用者的青睐。

随着云存储系统规模的不断扩大，网络问题逐渐成为云存储系统的主要问题。云存储网络中的流调度问题，作为影响用户体验的关键问题之一，正受到越来越多的关注。相比一般网络环境下的流调度问题，云存储网络下的流调度更加复杂。首先，云存储网络中流量的传输机制更加多样。为了提高数据存储的安全性与可靠性，云存储系统常使用多副本存储机制。在多副本存储机制下，用户上传的数据首先被存储在主存储节点，再由主存储节点分发至多个从存储节点。这使云存储网络中不仅包含一对一的单播传输，还包含大量一对多的多播传输以降低数据分发时产生的冗余流量。其次，云存储网络中流量的类型也更加多样。除了用户执行上传下载时所产生的业务流量，作为一个大规模的分布式系统，云存储系统本身也包含丰富的背景业务流，不同类型的背景业务流对网络性能的需求也不相同。如心跳数据流用于监测系统各模块是否工作正常，需要的传输带宽较少，但对时延相对敏感；而迁移业务流用于各存储节点之间的负载均衡，对时延要求不高，但需要较大的传输带宽。

因此，针对更加复杂的云存储网络环境，如何制定高效的云存储网络流调度方法以提高用户的使用体验，已成为云存储系统管理者所面临的巨大挑战。首先，不同类型的业务流对网络性能的要求不同，如何在一个共享的云存储系统中满足不同业务流的服务质量需求是流调度所面临的一个挑战。其次，网络的整体利用率同样重要，如何在提高不同业务流服务质量的同时尽可能地实现网络的负载均衡以提高整体利用率，是设计云存储网络流调度方法所要面对的另一挑战。最后，在云存储网络存在大量多播传输任务的背景下，如何构建流调度的多播树，并最大化降低数据分发时产生的冗余流量，也是制定云存储网络流调度方法所要面临的挑战之一。

因此，迫切需要设计一种云存储网络环境下的多播流调度方法，以实现在降低冗余流量、提高系统负载均衡性能的同时提高不同业务流的服务质量性能。虽然云存储系统采用数据中心网络技术整合存储设备，但现有的数据中心网络流调度方法大多针对流量模式符合长尾分布的数据中心网络环境，即 90%的数据流为小流；剩余 10%数据流为大流，却占据了 90%的网络带宽[4]。在这种数据中心网络环境背景下，部分研究者提出针对所有数据流的通用调度方法[5-7]，即不考虑不同业务流之间的差异性，对所有数据流采用通用的调度策略。也有研究者按数据流的大小将其分为“大象流”与“老鼠流”，并分别采用不同的调度机制[8-10]。这些方法虽然在设计的应对场合取得了较好的效果，但难以应对云存储网络环境中针对不同业务的多播流调度需求。

对此，本文在 Ceph 云存储网络环境下，给出一种针对多业务场景的多播流调度方法，主要贡献如下。

1) 提出 Ceph 云存储网络中业务优先级区分的多播流调度优化模型。模型根据不同业务流对网络性能的不同需求以及当前网络的状态信息，为各业务流定制最佳的传输路径。通过最小化全网流量、最小化累积加权时延以及最小化最大链路带宽利用率来提高系统的服务质量性能。

2) 提出基于理想解法（TOPSIS, technique for order preference by similarity to ideal solution）与最大公共子路径的多播流调度方法（MFSTM, multi-casting flow scheduling method based on TOPSIS and maximum common sub-path），将多播调度任务分解为多个单播调度的多属性决策问题，并结合寻找最大公共子路径的方法，设计求解多播流调度问题。

3) Mininet 搭建模拟 Ceph 云存储网络环境，验证 MFSTM 的有效性与适用性。模拟平台验证表明，MFSTM 多播流调度方法可以在降低冗余流量、提高网络负载均衡性能的同时，减少高优先级业务流的传输时延。

2 相关工作

在大规模云存储网络环境中，网络流调度技术对云存储网络的性能具有重要影响。它是指对于云存储系统中各项服务产生的数据流，通过调度这些数据流在云存储网络中的传输路径、传输优先级等，来优化网络流量的传输，提高用户的使用体验[11]。在优化传输路径方面，ECMP（equal cost multipath）[12]为所有数据流随机选择一条可达目的节点的传输路径，其目标是充分利用数据中心网络中存在多路径的特点，利用哈希的方式将数据流均衡地分配到各个传输路径之上。Hedera[13]是一种集中控制式的流量调度方法，它将数据流按大小分类，并将大流分配到剩余带宽较多的传输路径上。在优化传输优先级方面，PDQ（preemptive distributed quick）[14]是一种以抢占方式执行最小任务优先的启发式流调度方法，以数据流的剩余时间为传输优先级的评判依据，剩余时间越小的数据流具有越高的传输优先级。pFabric[15]将数据流的剩余量大小作为传输优先级的评判依据，剩余量越小的数据流具有越高的传输优先级。

上述流调度方法大多针对点对点的单播传输环境，然而随着网络业务越来越复杂，一对多的多播发送场景也越来越多。交互式网络电视业务的视频音频分发[16]、云存储系统的多副本复制[17]、传感器监视数据的分发[18]等业务都伴随大量的多播传输需求。现有的多播流调度技术大多结合软件定义网络（SDN, software defined network）技术，通过对网络状态信息的采集以及采用集中控制的思想提高对网络的管理效率。RMMR（robust multipath multicast routing）[19]针对视频流的分发场景，通过使用基于 SDN 的多路径多播流调度机制，获得了比传统网际互连协议（IP, internet protocol）多播机制更低的分组丢失率以及更好的网络负载均衡能力。BCMS（multicast scheduling with bounded congestion）[20]是针对胖树拓扑展开讨论的数据中心网络多播流调度方法，它根据多播业务流的网络带宽需求以及当前网络各链路的剩余带宽情况，为每个多播业务流计算最佳的传输路径，以实现网络拥塞控制与负载均衡。MSaSDN[21]针对基于胖树拓扑的数据中心网络，提出了基于最小化链路拥塞开销的多播树构建方法，提高了网络的负载均衡性能。MSaMC[22]针对目前 SDN 网络状态测量的滞后性问题，通过结合马尔可夫链，利用当前时隙的拥塞概率来预测下一时隙的拥塞概率，提高了网络的吞吐量并降低了数据流的平均传输时延。DuSM[8]针对数据中心网络的多播流调度问题，将数据流按大小分为“大象流”与“老鼠流”，针对“老鼠流”，将其多播任务转化为多个单播任务，以降低交换机所需的流表数量；针对“大象流”，构建多个共享树，以实现其在多链路上的负载均衡。

上述多播流调度方法大多针对所有数据流给出通用的调度策略，或是仅按数据流的大小进行分类并给出 2 种调度机制。然而在云存储网络环境下，出于成本的考虑，一个云存储系统往往由多业务所共享。不同业务产生的数据流对网络性能的要求不尽相同。如在以视频业务为主的网络环境下，网络带宽是需要考虑的重点因素[23]。在以游戏业务为主的网络环境下，时延是用户考量的重点[24]。而在以分布式计算任务为主的网络环境下，不仅需要考虑一对多发送带来的多播传输问题[25]，还需要考虑多对一发送所带来的 TCP-incast 问题[26]。因此，在云存储网络这种较为复杂的环境下采用单一的调度策略往往难以取得理想的效果。

因此，网络流调度方法需要根据不同业务场景下各业务流对网络性能的不同需求给出定制化的解决方案。本文针对现有网络流调度方法难以处理云存储网络中多业务流的不同多播调度需求问题，以 Ceph 云存储系统为代表，结合 SDN 中的集中控制思想，给出一种支持业务优先级区分的云存储网络多播流调度方法。

3 云存储网络多播流调度问题的建模

本节首先简单介绍基于 SDN 与胖树拓扑的 Ceph 云存储系统基本工作机制并分析其流量构成，然后对云存储网络环境下的多播流调度问题进行分析与建模。建模目标是在降低系统冗余流量、提高系统负载均衡性能的同时，减少高优先级业务流的传输时延，从而提高系统的服务质量性能。

3.1 基于 SDN 与胖树拓扑的 Ceph 云存储系统

对数据流的传输路径进行合理调度需要基于当前的网络状态信息。SDN 作为一种新的网络模式，采用集中控制的思想，将控制平面与数据平面相分离，提高了网络状态测量以及网络管理的灵活性。本文采用前期工作中基于 SDN 的网络状态测量方法[27]实时更新并维护系统网络中各链路的状

态信息，为流调度工作提供数据支撑。胖树拓扑是当前数据中心网络中最常见的多根树拓扑之一，它为网络内部各源目的节点之间构建多条并行路径，提高了网络内部东西向流量的吞吐量并降低了单点故障与网络热点的出现概率。图 1 展示了基于 SDN 与 4 叉胖树的 Ceph 云存储系统，其中 pod 表示一组直接互联的汇聚及边缘交换机。

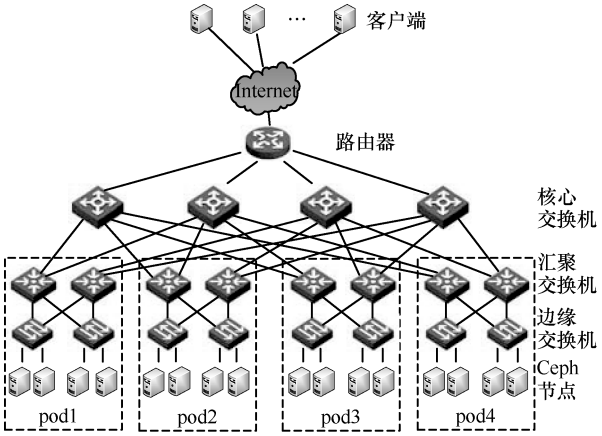

图 1　基于 SDN 与 4 叉胖树的 Ceph 云存储系统

本文主要探讨处于多副本工作模式下的 Ceph 集群，即用户上传一份数据至某个 Ceph 节点后，该 Ceph 节点会将数据复制并以多播传输的方式分发至其他 Ceph 节点进行多副本存储，以提高数据的安全性与可靠性。除了用户执行上传/下载操作产生的业务数据流之外，作为一个大规模的分布式系统，Ceph 集群也包含了心跳数据流和迁移数据流在内的背景业务流。虽然 Ceph 的数据迁移一般发生于系统空闲时，但由于数据迁移的时间常长达数小时，因此会出现迁移数据流与业务数据流同时存在的场景。在这种场景下，Ceph 云存储网络中主要包含如下 3 种类型的流量。

1) 心跳数据流，用于监测 Ceph 各节点是否正常工作。其拥有最高的传输优先级，对时延高度敏感，需要的传输带宽较少。

2) 用户业务数据流，由用户执行的上传/下载任务产生。其完成时间直接影响用户的使用体验，具有较高的传输优先级，对时延较为敏感，需要的传输带宽较多。

3) 系统迁移数据流数据流，由 Ceph 系统的负载均衡机制产生。其时延不会影响系统的正常运行以及用户的使用体验，传输优先级最低，但需要的网络带宽较多。

3.2　业务优先级区分的多播流调度问题建模

云存储网络可以表示为 $G=(V, E)$，其中，V 表示网络中的顶点，对应物理环境中的 SDN 交换机；E 表示顶点之间的边。$e=(\varepsilon_e,\delta_e)$ 表示从顶点 ε_e 到顶点 δ_e 的链路。第一个目标函数 f_1 为找到使全网流量最小化的多播传输路径。

$$\min f_1=\sum_{\varphi=1}^{M}\sum_{e\in E} b_\varphi x_{\varphi e} \tag{1}$$

$$\text{s.t.}\ \sum_{\varphi=1}^{M} b_\varphi x_{\varphi e}\leqslant b_e,\forall e\in E \tag{2}$$

$$x_{\varphi e}\in\{0,1\},\forall\varphi\in\{1,\cdots,M\},\forall e\in\{1,\cdots,Q\} \tag{3}$$

其中，b_φ 表示多播流 φ 所需要的传输带宽；b_e 表示链路 e 的最大带宽容量；式(2)所示约束条件保证了每个链路 e 上所分配的数据流的带宽之和不大于链路的带宽容量；$x_{\varphi e}$ 为一个二进制的决策参数，$x_{\varphi e}$=0 表示多播流 φ 未经过链路 e，$x_{\varphi e}$=1 表示多播流 φ 经过了链路 e。本文所用的符号及其对应的含义如表 1 所示。

表 1　　符号及其对应的含义

符号	含义	符号	含义
V	网络顶点集	E	网络边集
φ	多播流的序号	M	多播流的数量
e	连接相邻顶点的链路	Q	链路的数量
p	一源一端的单播路径	$d_{\varphi p}$	φ 经 p 的传输时延
k	一源多端的多播路径	L	多播路径的数量
b_φ	φ 所需的带宽	b_e	链路 e 的带宽容量
w_φ	φ 对时延的敏感系数	$d_{\varphi b}$	φ 的最大时延阈值
$x_{\varphi e}$	二进制决策参数	$x_{\varphi k}$	二进制决策参数

除了最小化全网流量以降低云存储网络的整体负载外，为了进一步提高用户的使用体验，需要最小化累积加权时延，对应目标函数 f_2 为

$$\min f_2=\sum_{\varphi=1}^{M}\sum_{k=1}^{L} w_\varphi \max_{p\in k}\{d_{\varphi p}\}x_{\varphi k} \tag{4}$$

$$\text{s.t.}\ \sum_{\varphi=1}^{M} b_\varphi x_{\varphi k}\leqslant \min_{e\in k}\{b_e\},\forall k\in\{1,\cdots,L\} \tag{5}$$

$$\sum_{k=1}^{L} x_{\varphi k}=1,\forall\varphi\in\{1,\cdots,M\} \tag{6}$$

$$\sum_{k=1}^{L}\max_{p\in k}\{d_{\varphi p}\}x_{\varphi k}\leqslant d_{\varphi b},\forall\varphi\in\{1,\cdots,M\} \tag{7}$$

$$x_{\varphi k}\in\{0,1\},\forall\varphi\in\{1,\cdots,M\},\forall k\in\{1,\cdots,L\} \tag{8}$$

其中，w_φ 表示多播流 φ 对时延的敏感系数，它由业务流的特性决定，对时延越敏感的数据流对应的 w_φ 值越大；k 表示一源到多端的多播传输路径；p 表示 k 中源到某一个端节点的单播传输路径；$d_{\varphi p}$ 表示数据流经路径 p 的传输时延；约束条件(5)保证了每个多播路径 k 上所分配的数据流的带宽之和不大于多播路径的瓶颈带宽；约束条件(6)确保了每条多播流 φ 必须选择且只能选择一个多播传输路径；约束条件(7)保证了被选择的多播路径需要满足数据流的最大传输时延限制；$x_{\varphi k}$ 为一个二进制的决策参数，$x_{\varphi k}$=0 表示多播流 φ 未经过多播路径 k，$x_{\varphi k}$=1 表示多播流 φ 经过了多播路径 k。

在提高各业务流服务质量的同时，整体系统的负载均衡性能同样重要。最小化最大链路带宽使用率的目标函数 f_3 为

$$\min f_3 = \max_{e\in E} \sum_{\varphi=1}^{M} \frac{b_\varphi x_{\varphi e}}{b_e} \tag{9}$$

$$\text{s.t.} \sum_{\varphi=1}^{M} b_\varphi x_{\varphi e} \leqslant b_e, \forall e \in E \tag{10}$$

$$x_{\varphi e} \in \{0,1\}, \forall \varphi \in \{1,\cdots,M\}, \forall e \in \{1,\cdots,Q\} \tag{11}$$

其中，约束条件(10)保证了每个链路 e 上所分配的数据流的带宽之和不大于链路的带宽容量，$x_{\varphi e}$ 为数据流进行链路选择的二进制决策参数。

至此得到了在云存储网络中，针对多业务多播流调度问题的 3 个目标函数，分别是最小化全网流量、最小化累积加权时延以及最小化最大链路带宽使用率。由于这 3 个目标函数不完全相互独立，无法通过单独处理同时得出各自的最优解，因此整体的优化目标函数 Z 为

$$\min Z = [f_1, f_2, f_3] \tag{12}$$

$$\text{s.t.} \sum_{\varphi=1}^{M} b_\varphi x_{\varphi e} \leqslant b_e, \forall e \in E \tag{13}$$

$$\sum_{\varphi=1}^{M} b_\varphi x_{\varphi k} \leqslant \min_{e\in k}\{b_e\}, \forall k \in \{1,\cdots,L\} \tag{14}$$

$$\sum_{k=1}^{L} x_{\varphi k} = 1, \forall \varphi \in \{1,\cdots,M\} \tag{15}$$

$$\sum_{k=1}^{L} \max_{p\in k}\{d_{\varphi p}\} x_{\varphi k} \leqslant d_{\varphi b}, \forall \varphi \in \{1,\cdots,M\} \tag{16}$$

$$x_{\varphi e} \in \{0,1\}, \forall \varphi \in \{1,\cdots,M\}, \forall e \in \{1,\cdots,Q\} \tag{17}$$

$$x_{\varphi k} \in \{0,1\}, \forall \varphi \in \{1,\cdots,M\}, \forall k \in \{1,\cdots,L\} \tag{18}$$

针对上述云存储网络中不同业务流的多播调度问题，本文提出了基于理想解与最大公共子路径的流调度方法。

4 MFSTM 多播流调度方法

Z 的 3 个子目标函数难以同时取得最小值，即难以同时找到满足 3 个子目标的最优解。由于在 Ceph 云存储网络中，提高多副本数据多播分发的效率可以有效降低用户业务数据流的规模，进而降低系统负载并提高用户使用体验。因此，MFSTM 将最小化全网流量作为主要优化目标。在找到一组满足最小化全网流量的解集后，MFSTM 再根据剩余的优化目标选择其中的最佳传输路径。具体的求解流程使用基于理想解的多属性决策方法以及最大公共子路径方法来实现。

针对一个多播流的发送请求 $\text{flow}_\varphi = (\tau, T, b_\varphi, d_\varphi)$，其中，$\tau$ 表示数据流的源节点，$T = \{t_1, t_2, \cdots, t_s\}$ 为该多播流的目的节点集合，b_φ、d_φ 分别为数据流 φ 传输所需的带宽和最大传输时延阈值。MFSTM 首先使用基于 SDN 的网络状态测量技术找出同时满足数据流传输时延以及带宽需求的链路集合；然后针对源节点与目的节点集合中的每个节点分别构建点对点传输路径，基于理想解法找出每个点对点传输的前 η 个最优传输路径；最后通过寻找各个点对点路径之间的最大公共子路径来确定多播树，多播树的分发节点为最大公共子路径中的最后一个节点。

4.1 网络状态参数的选取及其向量化表示

合适网络状态参数的选取对于理想解这种多属性决策方法的效果起决定性作用。本文针对云存储网络环境下的多播流调度场景，选择传输路径的剩余最大瓶颈带宽、传输路径的平均端到端时延以及核心交换机上的流表数量作为决策参数。这些决策参数表示为

$$P = \{p_1, p_2, \cdots, p_n\} \tag{19}$$

其中，P 表示数据流端到端传输路径的集合，每条路径 p 由多个链路 e 连接构成。这些路径的集合可以由 Dijkstra[28]算法得出，如式(20)所示。

$$B_P = \{b_1, b_2, \cdots, b_n\} \tag{20}$$

其中，B_P 表示每条传输路径的剩余最大瓶颈带宽的

集合，由于每个路径 p 由多个链路 e 连接构成，路径 p 的剩余最大瓶颈带宽即为构成它的各个链路 e 中的最小剩余带宽值。这里的链路剩余带宽可以通过基于 SDN 的网络状态测量方法[27]得出，如式(21)所示。

$$D_P=\{d_1,d_2,\cdots,d_n\} \tag{21}$$

其中，D_P 表示数据流分别经过每条传输路径的端到端时延集合。这里的端到端时延可以通过基于 SDN 的网络状态测量方法[27]得出。

$$O_P=\{o_1,o_2,\cdots,o_n\} \tag{22}$$

其中，O_P 表示每条路径所经过的核心交换机中已存在的流表数量的集合。流表数量越多则流表查找时延越高，同时选择通过流表数量较多的核心交换机也会增加数据流冲突的概率。这里的交换机流表数量可以通过 SDN 中的 OpenFlow 协议获得。

如式(19)～式(22)所示，MFSTM 在程序初始化时即采用 Dijkstra 算法计算出任意 2 个存储节点之间的传输路径，并存储于专门的数据表中，再使用基于 SDN 的网络状态测量方法实时更新这些路径的剩余带宽、平均传输时延等信息。因此，在后续的最优路径计算过程中，可以直接遍历数据表以获得当前的链路状态信息，降低重复路径发现所带来的计算开销。

4.2　基于理想解的最优单播路径计算

MFSTM 在处理一个包含 s 个目的节点的一对多的多播流请求时，首先将此多播流任务分解为 s 个点对点的单播任务，并分别使用基于理想解的路径计算方法找到每个单播任务的前 η 个最优路径。

理想解法是一种有效的多属性决策方案，该方案从归一化的原始数据矩阵中构造出决策问题的正理想解和负理想解。通过计算各方案与正、负理想解的距离作为评价方案的准则。MFSTM 中基于理想解的单播路径计算方法如下。

步骤 1　基于 SDN 的网络状态测量与候选路径选取

针对云存储网络 $G=(V, E)$，其中，V 为网络中的节点，对应物理环境中的 SDN 交换机；$E=\{e_1,e_2,\cdots,e_q\}$ 表示节点之间的边。MFSTM 首先利用 SDN 对网络链路状态进行周期性的测量，实时维护一张包含所有链路 E 及其对应链路信息的数据表 $M=(E,D,B,O)$，其中，$D=\{d_{e1},d_{e2},\cdots,d_{eq}\}$ 和 $B=\{b_{e1},b_{e2},\cdots,b_{eq}\}$ 分别对应链路集合 E 中各子链路的平均传输时延集合与剩余带宽集合，$O=\{o_1,o_2,\cdots,o_n\}$ 为各核心交换机中当前流表数量的集合。针对一个单播流传输任务 $\text{flow}_\phi=(\tau_\phi,t_\phi,b_\phi,d_\phi)$，其中，$\tau_\phi$ 和 t_ϕ 分别表示单播流 ϕ 的源节点与目的节点，b_ϕ 和 d_ϕ 分别表示单播流 ϕ 对链路带宽和传输时延的要求。MFSTM 首先根据数据流对传输路径的性能需求来对现有的网络进行过滤，再使用基于 Dijkstra 的路径发现算法找到所有的候选路径集 P^*，如式(23)所示。

$$P^*=\{p_1,p_2,\cdots,p_n\} \tag{23}$$

并满足 $P^*\subseteq P$，对于 $\forall e\in P^*$，有 $b_e\geqslant b_\phi$ 且 $d_p\leqslant d_\phi$。

步骤 2　决策矩阵的构造及其归一化处理

基于上述选取的决策参数，给出决策矩阵 $\boldsymbol{M}$。

$$\boldsymbol{M}=\begin{bmatrix} b_1 & b_2 & \cdots & b_n \\ d_1 & d_2 & \cdots & d_n \\ o_1 & o_2 & \cdots & o_n \end{bmatrix} \tag{24}$$

其中，矩阵的每一行表示一种决策参数，每一列表示一个候选的端到端传输路径。

为了消除不同决策参数间不同量纲带来的影响，采用极差标准化方法对决策矩阵进行归一化处理。处理方式如式(25)和式(26)所示。

$$u(x)=\frac{x^{\max}-x}{x^{\max}-x^{\min}} \tag{25}$$

$$v(x)=\frac{x-x^{\min}}{x^{\max}-x^{\min}} \tag{26}$$

其中，式(25)用于处理成本型决策参数，如链路的传输时延以及交换机中的流表数量，这类决策参数的特点是越小的值对应越好的效果；式(26)用于处理效益型决策参数，如链路的剩余带宽容量，这类参数的特点是越大的值对应越好的效果。

最终，得到经过标准化处理的决策矩阵 $\boldsymbol{M}^*$。

$$\boldsymbol{M}^*=\begin{bmatrix} b_1^* & b_2^* & \cdots & b_n^* \\ d_1^* & d_2^* & \cdots & d_n^* \\ o_1^* & o_2^* & \cdots & o_n^* \end{bmatrix} \tag{27}$$

其中，$b_j^*=v(b_j)$，$d_j^*=u(d_j)$，$o_j^*=u(o_j)$，$j\in\{1,2,\cdots,n\}$，j 表示矩阵列的序号，对应实际场景中候选路径的序号。

步骤 3　加权决策矩阵的构建及其正负理想解的确定

不同业务的数据流对不同网络参数的敏感程度

不同。如在云存储网络中，心跳数据流对传输时延的敏感性较高，但对带宽的敏感性较低；而迁移数据流对时延的敏感性较低，但出于网络负载均衡的考虑，会倾向选择剩余带宽较大的传输路径。为 3.1 节介绍的 3 种业务数据流分别构建权重系数向量 $\boldsymbol{W}$ 为

$$\boldsymbol{W}=\left[w_b,w_d,w_o\right]^{\mathrm{T}} \tag{28}$$

其中，w_b、w_d、w_o 分别表示业务流对链路剩余带宽、链路平均端到端时延以及链路中核心交换机上的流表数量的权重系数。各权重系数一般通过实验的方式获取[29]，本文设定的权重系数将在实验部分进行介绍。

根据权重系数向量得出不同业务流所对应的加权决策矩阵。以时延敏感流为例，其加权决策矩阵 $\boldsymbol{Z}$ 的元素 Z_{ij} 为

$$Z_{ij}=W_i M^*_{ij} \tag{29}$$

其中，$i\in\{1,2,3\}$，$j\in\{1,2,\cdots,n\}$。

根据加权决策矩阵得出其正负理想解

$$\boldsymbol{P}^{+}=\max_j\left\{Z_{ij}\mid i=1,2,3\right\},\ j\in\{1,2,\cdots,n\} \tag{30}$$

$$\boldsymbol{P}^{-}=\min_j\left\{Z_{ij}\mid i=1,2,3\right\},\ j\in\{1,2,\cdots,n\} \tag{31}$$

其中，$\boldsymbol{P}^{+}$ 表示加权决策矩阵的正理想解，由所有候选路径中每种决策参数的最大值构成；$\boldsymbol{P}^{-}$ 表示加权决策矩阵的负理想解，由所有候选路径中每种决策参数的最小值构成。

步骤 4 计算每个候选传输路径到正、负理想解的距离

$$D_j^{+}=\sqrt{\sum_{i=1}^{3}(Z_{ij}-P_i^{+})^2},\ j\in\{1,2,\cdots,n\} \tag{32}$$

$$D_j^{-}=\sqrt{\sum_{i=1}^{3}(Z_{ij}-P_i^{-})^2},\ j\in\{1,2,\cdots,n\} \tag{33}$$

其中，Z_{ij} 为候选传输路径 $[Z_{1j},Z_{2j},Z_{3j}]^{\mathrm{T}}$ 中的一个元素，D_j^{+} 和 D_j^{-} 分别为各候选路径到其到正负理想解的欧氏距离。

步骤 5 计算每个候选路径与最优候选路径的相对贴近度 C_j^{+} 为

$$C_j^{+}=\frac{D_j^{-}}{D_j^{+}+D_j^{-}},\ j\in\{1,2,\cdots,n\} \tag{34}$$

其中，相对贴近度越大表示该传输路径越适合当前的单播流任务。

4.3 通过最大公共子路径构建多播树

多播传输路径的数据分发节点应尽量靠近接收节点，以最小化系统的冗余流量。针对一个具有 s 个目的节点的多播流发送请求，MFSTM 根据前述基于理想解的单播流路径计算方法，为每个单播流计算出 η 个符合其对网络性能需求的最优单播路径，得到 $\{\eta_1\}+\{\eta_2\}+\cdots+\{\eta_s\}$ 共 ηs 个单播路径。每个路径 p 由一组有序的链路 e 连接构成，其中 $e=(\varepsilon_e,\delta_e)$ 表示从相邻节点 ε_e 到 δ_e 的链路。

定义 1 若路径 p 满足如下条件，则称 p 为一组单播流路径 $\{\eta_1\}+\{\eta_2\}+\cdots+\{\eta_s\}$ 中的最大公共子路径。

条件 1 p 同时分别为路径集合 $\{\eta_1\}$、$\{\eta_2\}$、$\{\eta_s\}$ 中某单播路径的子路径。

条件 2 没有其他符合条件 1 的路径 p' 包含比 p 更多的链路 e。

若 e_x 为从发送节点出发的最大公共子路径 p 中的最后一个有序链路，则 MFSTM 选择 $e_x=(\varepsilon_{ex},\delta_{ex})$ 中的节点 δ_{ex} 作为多播树中的数据分发节点，并分别根据原单播路径建立从 δ_{ex} 出发至各接收节点的多播分发路径，从而完成多播树的构建。若存在多个满足最小化全网流量这一条件的多播树，则从中选择具有最大累计相对贴近度 C_{total} 的多播树，具体计算如式(35)所示。

$$C_{\text{total}}=\sum_{\phi=1}^{s}\max_j(C_{j\phi}^{+}),\ j\in\{1,2,\cdots,\eta\} \tag{35}$$

其中，s 表示多播流的接收节点数量，ϕ 表示发送节点至某一个接收节点的单播任务流，j 表示每个单播流 ϕ 可以选择的候选路径，$\max_j(C_{j\phi}^{+})$ 表示单播流 ϕ 根据式(34)在各候选路径 j 中可以得到的最大相对贴近度。

最后，通过一个调度实例来进一步说明 MFSTM 构建多播路径的基本流程。假设有一个多播任务流 $\{1,(2,3),b,d\}$，表示其发送节点为交换机 1，接收节点为交换机 2 与交换机 3，所需带宽和时延约束分别为 b 和 d。MFSTM 首先将此多播任务分解为 2 个单播任务，分别为 $\{1,2,b,d\}$ 与 $\{1,3,b,d\}$；其次，利用基于理想解的最优单播路径计算方法分别为 2 个单播任务计算满足约束条件的最优单播路径集，这里取每个路径集中包含的路径数量 η=2。如图 2(a)所示，为单播任务 $\{1,2,b,d\}$ 计算得出的最

优路径集为{[1, 4, 7, 5, 2], [1, 4, 8, 5, 2]}。如图 2(b)所示，为单播任务{1, 3, *b*, *d*}计算得出的最优路径集为{[1, 4, 8, 6, 3], [1, 4, 9, 6, 3]}。最后，找到 2 个解集从发送节点出发的最大公共子路径为[1, 4, 8]。通过将交换机 8 作为多播树的分发节点，并按原单播路径建立从交换机 8 至各接收节点的多播分发路径，得到如图 2(c)中所示的多播路径[1, 4, 8, (5, 6), (2, 3)]。

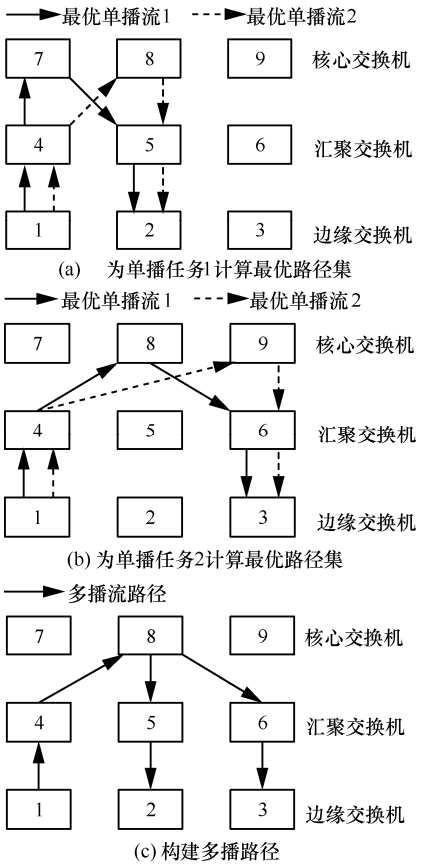

图 2　MFSTM 的调度实例

5　实验设置与性能评估

本节对 MFSTM 多播流调度方法进行实验设置与性能评估。实验使用 Mininet[30]网络模拟器构建实验网络拓扑，并在开源的 SDN 控制器 RYU[31]上部署 MFSTM 流调度方法的逻辑，最后使用 TCPreplay[32]重放真实的 Ceph 云存储系统业务数据流[33]以测试流调度方法的性能。整体实验平台搭建于一台曙光 A840r-G 服务器上，服务器拥有 64 核× 2.1 GHz 处理器，64 GB 内存以及 600 GB 硬盘。其中，每个处理器核被用于单独处理一个 TCPreplay 进程，使其能够以恒定速率发送一个长数据流。

在实验拓扑选择方面，由于 Ceph 云存储系统中用户业务数据的多个副本常被存储于不同的 pod 中以提高数据的安全性。因此，图 1 所示拓扑可以抽象为图 3。实验中利用 Mininet 2.3.0 构建图 3 中的网络拓扑作为实验系统拓扑，并设定链路带宽为 50 Mbit/s。

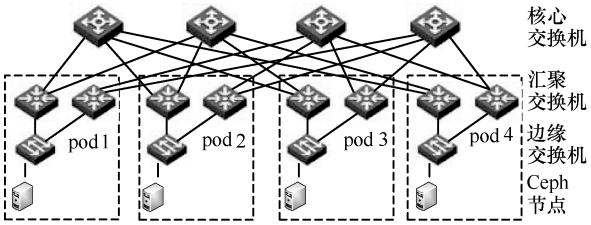

图 3　实验系统拓扑

在数据流选择方面，如前文所述，本文主要探讨多副本工作模式下，Ceph 集群执行数据迁移任务出现用户业务数据时的网络流调度问题。此时，系统中的数据流包括心跳数据流、用户业务数据流以及系统迁移数据流。实验采用本文前期工作中采集的 Ceph 云存储集群真实数据流[33]，数据流的统计特征如表 2 所示。

表 2　Ceph 云存储系统中不同业务流的统计特征

数据流类型	优先级	数据分组个数/个	平均数据分组大小/B	流速率/(Mbit·s^{-1})	流持续时间/s
心跳数据流	1	54	1 477.42	92.87	0.006 554
用户业务数据流	2	67 073	1 508.65	12.93	39.31
系统迁移数据流	3	340 354	1 505.56	4.36	654.12

表 2 中的不同业务流在采集时被分别保存为不同的 pcap 文件。在本文的实验过程中，通过使用 TCPreplay 重放这些 pcap 数据流，以模拟真实的 Ceph 云存储系统运行环境，每轮模拟实验的测试时长设定为 180 s。对于用户业务数据流以及系统迁移数据流，流量的重放速度与其自身的流速率相同，分别为 12.93 Mbit/s 以及 4.36 Mbit/s。对于心跳数据流，为了降低模拟实验环境中的分组丢失率，将对它的重放速率设定为 1 Mbit/s。同时，受限于模拟实验的测试时长，只重放系统迁移数据流的前 40 000 个数据分组，以使其流的持续时间降低至 78 s，从而可以在测试时长内完成流的全部发送。

为了进一步测试流调度方法在不同网络负载下的性能，实验设置了 3 种流量负载场景。

1) 低负载场景：60 条心跳数据流，10 条用户业务数据流，10 条系统迁移数据流。

2) 中负载场景：60 条心跳数据流，20 条用户业务数据流，20 条系统迁移数据流。

3) 高负载场景：60 条心跳数据流，30 条用户业务数据流，30 条系统迁移数据流。

因为心跳数据流是以恒定周期在不同 Ceph 节点之间进行传输的，所以其流数目在测试时长确定时保持不变。在每轮测试过程中，随机选择一个 Ceph 节点作为数据的发送节点，并在 180 s 内周期性地为每条数据流开启一个 TCPreplay 进程以执行数据流的发送。对于心跳数据流和系统迁移数据流，因其点对点的业务特性，TCPreplay 为其随机选择一个剩余 Ceph 节点作为数据的接收节点，执行单播发送；对于用户业务流，按常见三副本存储模式下主副本需要向其他 2 个从副本进行数据备份的业务场景，TCPreplay 为其随机选择 2 个剩余 Ceph 节点作为数据的接收节点，执行多播发送。具体的流量调度以及链路选择采用 MFSTM 多播流调度算法，其中单播流被认为是只有一个接收节点的特殊多播流进行处理。MFSTM 对心跳数据流、用户业务数据流、系统迁移数据流设定的权重系数向量分别为[0, 0.5, 0.5]、[0.5, 0.4, 0.1]、[1, 0, 0]。

实验将本文给出的 MFSTM 与 ECMP[11]以及 BCMS[19]进行对比。其中，ECMP 是数据中心网络中最常见的流调度方法之一，它采用哈希的方式为数据流在多条可达的路径中随机选择一条作为传输路径，以实现负载均衡。当 ECMP 处理多播流时，需要将一个多播流分解成多个点对点的单播流分别进行处理。BCMS 是一种有效的针对胖树网络拓扑的多播流调度方法，它根据多播数据流的带宽需求以及当前的网络状态为多播流选择合适的传输路径，以实现网络拥塞控制以及负载均衡。实验的对比项包括总体带宽利用率、不同链路间的带宽利用率标准差、核心交换机上的流表数量以及不同业务流的平均传输时延，每组实验测试 5 次，实验结果取平均值。

5.1 链路带宽利用率标准差

图 4 给出了在不同负载场景下，实验系统各链路之间带宽利用率的标准差。标准差越小表明各链路之间的带宽利用率越均衡，系统网络的负载均衡性能越好。如图 4 所示，3 种方法下的带宽利用率标准差在 0～42 s 都呈现快速上升的状态，并在 42 s 后保持相对稳定。这是由于占用带宽最多的用户业务数据流拥有 39.31 s 的流持续时间。在约 42 s 处，第一条用户业务数据流完成传输，此后系统网络中的用户业务数据流数目保持恒定，从而使系统的整体负载达到相对稳定的状态。

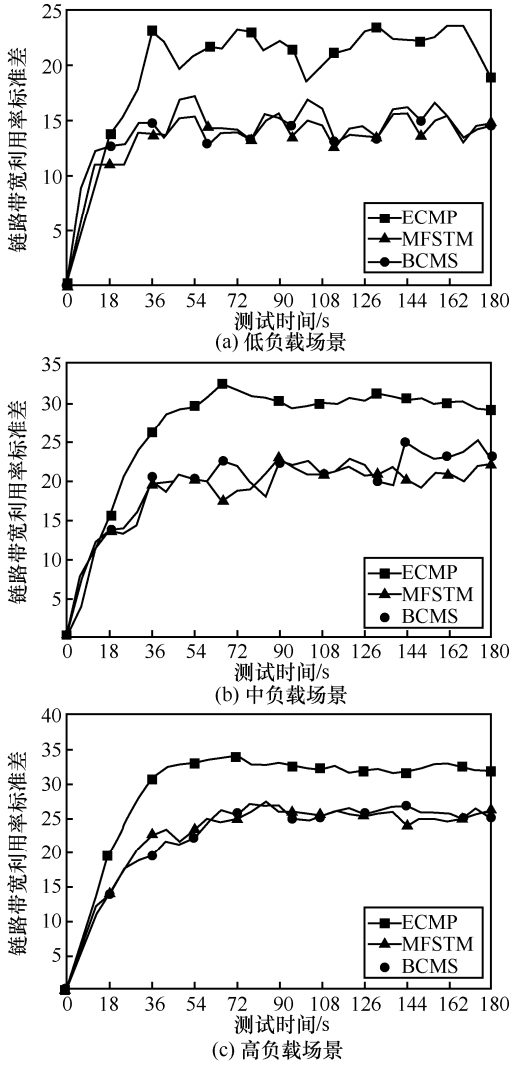

图 4 链路带宽利用率标准差

实验结果表明，在低、中、高 3 种负载场景下，使用 MFSTM 时带宽利用率标准差的平均值分别比使用 ECMP 时降低了 30.8%、29.8%以及 22.9%，取得了与 BCMS 相近的负载均衡效果。这是由于 ECMP 在进行路径选择时为所有数据流随机选择路径，若将多条大流分配到同一路径则会造成网络负载的不均衡。而 MFSTM 与 BCMS 在进行路径选择时会考虑网络当前的状态，为数据流选择剩余带宽较多的路径进行传输。

5.2　总体带宽利用率

系统的总体带宽利用率如式(36)所示。

$$p_{\text{use}}=\frac{\sum_{e=1}^{E}b_e e_{\text{use}}}{\sum_{e=1}^{E}b_e} \tag{36}$$

其中，e 表示实验系统拓扑中相邻交换机之间的网络链路，b_e 表示链路 e 的最大传输带宽，e_{use} 表示当前时刻链路 e 的带宽利用率。图 5 给出了在不同负载场景下实验系统的总体带宽利用率。当实验系统处于低、中负载场景下，使用 MFSTM 时系统总体带宽利用率的上升速度相比使用 ECMP 时更慢，且在所有测量时刻下的总体带宽利用率平均值比使用 ECMP 时的总体带宽利用率分别下降了 17.8%和 12.7%，取得了与 BCMS 相近的效果。这是由于 MFSTM 与 BCMS 为用户业务数据流建立了有效的多播传输路径，节约了网络传输带宽。

然而，在如图 5(c)所示的高负载场景下，虽然使用 MFSTM 下的系统总体带宽利用率在上升阶段低于 ECMP，但在 42 s 后的负载相对稳定阶段，使用 MFSTM 拥有比使用 ECMP 更高的总体带宽利用率。这是由于在高负载场景下，MFSTM 根据链路的剩余带宽对数据流进行了合理的分配，避免了多条大流选择相同传输路径而产生的链路拥塞，提高了系统在高负载场景下的网络吞吐量。MFSTM 与 BCMS 拥有相近的总体带宽利用率，因为在实验设计的云存储流量模式下，即多播流的接收节点有 2 个且分布在不同的 pod 中，MFSTM 与 BCMS 同时选择了各核心交换机作为多播路径中的数据分发节点，实现了最大化降低系统冗余流量。

5.3　核心交换机上的流表数量

图 6 给出了在完成全部数据流的发送后，实验系统拓扑图中 4 台核心交换机上流表数量的标准差。由于实验过程中的数据流都是在不同 pod 之间进行传输，因此其传输路径必须通过且只通过一台核心交换机。每当一条数据流被分配到一条传输路径上时，SDN 控制器会向这条路径上的所有交换机下发 2 条流表，分别对应数据流的往返传输路径。因此，各核心交换机上的流表数量越均衡，表明数据流在不同传输路径上的调度越均衡。同时，这也避免了多条流表集中于某一台核心交换机，降低了交换机所需的最大流表空间，节约了宝贵的流表资源。

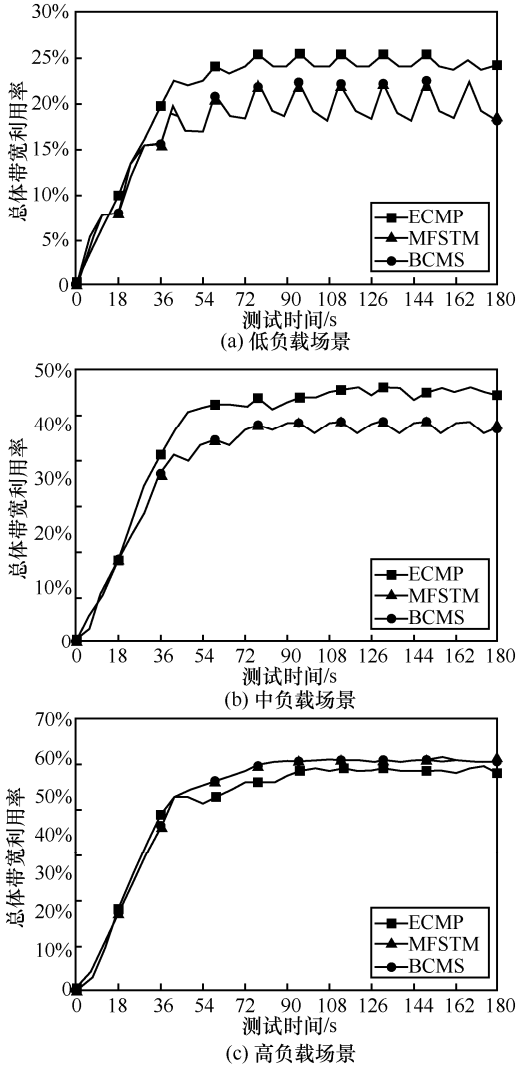

图 5　链路带宽利用率标准差

如图 6 所示，MFSTM 在所有负载场景下都拥有最小的核心交换机流表数量标准差。这是由于 MFSTM 在对高优先级流进行路径选择时，会优先选择具有更小流表数量的核心交换机所在的路径，可以降低数据流在传输过程中的流表查找时延。

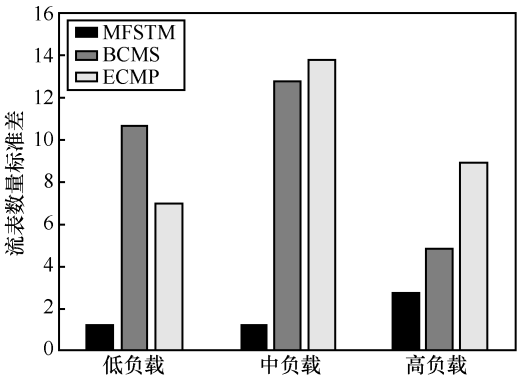

图 6　核心交换机之间流表数量的标准差

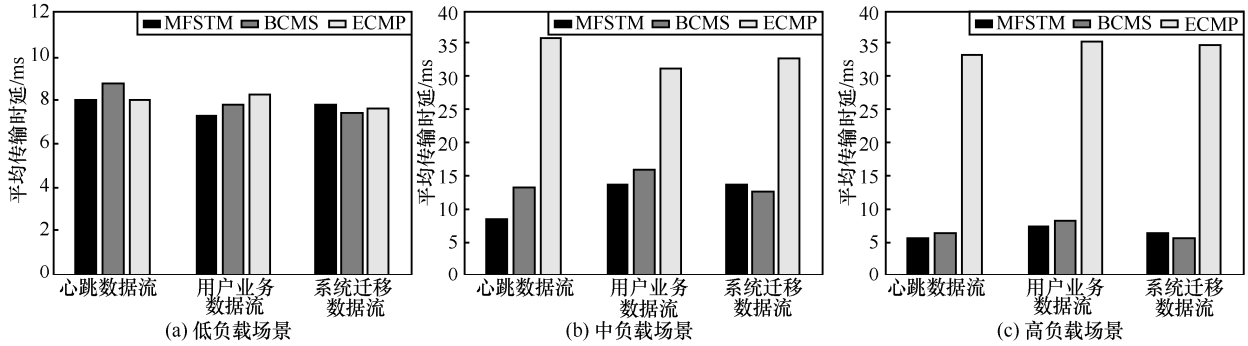

图 7 平均传输时延

5.4 平均传输时延

图 7 给出了具有不同传输优先级的业务流在不同负载场景下的平均传输时延。对于单播业务流，实验中记录的是其端到端传输时延；对于具有多个接收节点的多播业务流，实验中记录的是发送节点与最后一个收到数据的接收节点之间的端到端传输时延。在如图 7(a)所示的低负载场景下，不同方法之间各业务流的平均传输时延没有较为明显的差异。这是由于在低负载场景下，交换机可以有效处理数据流的转发，即使多条数据流被分配到同一传输路径，传输时延也不会有明显的增加。在中负载以及高负载场景下，MFSTM 与 BCMS 因较好的网络负载均衡性能，避免了 ECMP 下因链路拥塞而导致的传输时延大幅上升。

更重要的是，对于传输优先级较高的心跳数据流与用户业务数据流，MFSTM 具有更小的传输时延。如图 7(b)和图 7(c)所示，与 BCMS 相比，在中、高负载场景下使用 MFSTM，可以使具有最高传输优先级的心跳数据流的传输时延分别下降 37.9%和 9.0%。对于具有次高传输优先级的用户业务数据流，在中、高负载场景下使用 MFSTM 可以比使用 BCMS 分别降低 14.9%和 7.5%的平均传输时延。实验结果表明，本文给出的 MFSTM 可以降低高优先级流的平均传输时延，从而为系统提供更好的服务质量性能。

6 结束语

针对 Ceph 云存储网络环境中不同业务流对网络性能的差异化需求，本文给出了一种基于理想解与最大公共子路径的多播流调度方法，以实现支持业务优先级区分的多播流调度。首先将多播业务流分解为多个单播任务，给出一种基于理想解法的最优单播路径选择方法，为各单播任务寻找符合其对网络性能需求的最优单播路径集；再通过各路径集间的最大公共子路径确定多播树的数据分发节点以构建多播传输路径。实验结果表明，MFSTM 可以在降低冗余流量、提高网络负载均衡性能的同时，降低高优先级流的传输时延，提高了系统的服务质量性能。下一步计划在更大规模的云存储网络环境中测试并优化 MFSTM 的性能。在基于 SDN 的大规模云存储网络环境中，SDN 控制器需要定期向更多的交换机发送探测分组并处理回复以实时维护全网的状态信息。单个 SDN 控制器的集中控制和管理将成为瓶颈，需要使用多个 SDN 控制器进行协同管理，将涉及 SDN 多控制器的控制域划分、协同通信以及数据一致性。这些都是需要后续进行的更有意义的研究工作。

参考文献：

[1] NIELSEN L, VESTERGAARD R, YAZDANI N, et al. Alexandria: a proof-of-concept implementation and evaluation of generalised data deduplication[C]//2019 IEEE Global Communications Conference (GLOBECOM). Piscataway: IEEE Press, 2019: 1-6.

[2] ZHANG Y, WEI Q S, CHEN C, et al. Dynamic scheduling with service curve for QoS guarantee of large-scale cloud storage[J]. IEEE Transactions on Computers, 2018, 67(4): 457-468.

[3] YANG C T, LIU J C, KRISTIANI E, et al. NetFlow monitoring and cyberattack detection using deep learning with Ceph[J]. IEEE Access, 2020, 8: 7842-7850.

[4] CHOWDHURY M, ZHONG Y, STOICA I. Efficient coflow scheduling with varys[J]. ACM SIGCOMM Computer Communication Review, 2014, 44(4): 443-454.

[5] FAN F J, HU B, YEUNG K L, et al. Mini forest: distributed and dynamic multicasting in datacenter networks[J]. IEEE Transactions on Network and Service Management, 2019, 16(3): 1268-1281.

[6] HUANG K, SU X. Scalable datacenter multicast using in-packet bitmaps[J]. Distributed and Parallel Databases, 2018, 36(3): 445-460.

[7] WANG Y C, YOU S Y. An efficient route management framework for load balance and overhead reduction in SDN-based data center networks[J]. IEEE Transactions on Network and Service Management, 2018, 15(4): 1422-1434.

[8] CUI W Z, QIAN C. Scalable and load-balanced data center multicast[C]//2015 IEEE Global Telecommunications Conference (GLOBECOM). Piscataway: IEEE Press, 2015: 1-6.

[9] ZHANG Y X, CUI L, ZHANG Y. A stable matching based elephant flow scheduling algorithm in data center networks[J]. Computer Networks, 2017, 120: 186-197.
[10] SEHERY W, CLANCY C. Flow optimization in data centers with clos networks in support of cloud applications[J]. IEEE Transactions on Network and Service Management, 2017, 14(4): 847-859.
[11] 胡智尧, 李东升, 李紫阳. 数据中心网络流调度技术前沿进展[J]. 计算机研究与发展, 2018, 55(9): 1920-1930.
HU Z Y, LI D S, LI Z Y. Recent advances in datacenter flow scheduling[J]. Journal of Computer Research and Development, 2018, 55(9): 1920-1930.
[12] CHIESA M, KINDLER G, SCHAPIRA M. Traffic engineering with equal-cost-multipath: an algorithmic perspective[J]. IEEE-ACM Transactions on Networking, 2017, 25(2): 779-792.
[13] AL-FARES M, RADHAKRISHNAN S, RAGHAVAN B, et al. Hedera: dynamic flow scheduling for data center networks[C]//2010 USENIX Conference on Networked System Design and Implementation (NSDI). Berkeley: USENIX Press, 2010: 19.
[14] HONG C Y, CAESAR M, GODFREY P B. Finishing flows quickly with preemptive scheduling[J]. ACM SIGCOMM Computer Communication Review, 2012, 42(4): 127-138.
[15] ALIZADEH M, YANG S, SHARIF M, et al. pFabric: minimal near-optimal datacenter transport[J]. ACM SIGCOMM Computer Communication Review, 2013, 43(4): 435-446.
[16] 郑成渝, 焦博, 王军, 等. 基于可计算网络的 SDN 视频总线系统架构研究[J]. 通信学报, 2018, 39(Z1): 271-277.
ZHENG C Y, JIAO B, WANG J, et al. Research on SDN video bus system architecture based on computable network[J]. Journal on Communications, 2018, 39(Z1): 271-277.
[17] ISLAM S, MUSLIM N, ATWOOD J W. A survey on multicasting in software-defined networking[J]. IEEE Communications Surveys and Tutorials, 2018, 20(1): 355-387.
[18] ALSAEED Z, AHMAD I, HUSSAIN I. Multicasting in software defined networks: a comprehensive survey[J]. Journal of Network and Computer Applications, 2018, 104: 61-77.
[19] LEE M W, LI Y S, HUANG X, et al. Robust multipath multicast routing algorithms for videos in software-defined networks[C]//2014 IEEE International Symposium of Quality of Service (IWQoS). Piscataway: IEEE Press, 2014 : 218-227.
[20] GUO Z Y, DUAN J, YANG Y Y. On-line multicast scheduling with bounded congestion in fat-tree data center networks[J]. IEEE Journal on Selected Areas in Communications, 2014, 32(1): 102-115.
[21] LI G Z, GUO S T, YANG Y Y. Multicast scheduling algorithm in software defined fat-tree data center networks[C]//2017 IEEE International Symposium of Quality of Service (IWQoS). Piscataway: IEEE Press, 2017: 127-135.
[22] LI G Z, GUO S T, LIU G Y, et al. Multicast scheduling with Markov chains in fat-tree data center networks[C]//2017 International Conference on Networking, Architecture, and Storage (NAS). Piscataway: IEEE Press, 2017: 188-194.
[23] GEORGOPOULOS P, ELKHATIB Y, BROADBENT M, et al. Towards network-wide QoE fairness using openflow-assisted adaptive video streaming[C]//2013 ACM SIGCOMM Workshop on Future Human-Centric Multimedia Networking (FhMN). New York: ACM Press, 2013: 15-20.
[24] AMIRI M, ALOSMAN H, SHIRMOHAMMADI S, et al. Toward delay-efficient game-aware data centers for cloud gaming[J]. ACM Transactions on Multimedia Computing Communications and Applications, 2016, 12(5s): 1-19.
[25] ZHU T W, FENG D, WANG F, et al. A congestion-aware and robust multicast protocol in SDN-based data center networks[J]. Journal of Network and Computer Applications, 2017, 95: 105-117.
[26] ABDELMONIEM A M, BENSAOU B, ABU A J. Mitigating incast-TCP congestion in data centers with SDN[J]. Annals of Telecommunications, 2017, 73(3-4): 263-277.
[27] 王勇, 叶苗, 何倩, 等. 基于软件定义网络和多属性决策的 Ceph 存储系统节点选择方法[J]. 计算机学报, 2019, 42(2): 93-108.
WANG Y, YE M, HE Q, et al. A new node selecting approach in Ceph storage system based on software defined network and multi-attributes decision-making model[J]. Chinese Journal of Computers, 2019, 42(2): 93-108.
[28] BARBEHENN, M. A note on the complexity of Dijkstra's algorithm for graphs with weighted vertices[J]. IEEE Transactions on Computers, 1998, 47(2): 263-263.
[29] LUO Y Q, XIA J B, CHEN T P. Comparison of objective weight determination methods in network performance evaluation[J]. Journal of Computer Application, 2009, 29(10): 2624-2626.
[30] LANTZ B, O'CONNOR B. A mininet-based virtual testbed for distributed SDN development[J]. ACM SIGCOMM Computer Communication Review, 2015, 45(4): 365-366.
[31] ISLAM M T, ISLAM N, ALREFAT M. Node to node performance evaluation through RYU SDN controller[J]. Wireless Personal Communications, 2020, 112(1): 555-570
[32] HONG S S, WONG F, WU S F. TCPtransform: property-oriented TCP traffic transformation[J]. Lecture Notes in Computer Science, 2005, 3548: 222-240.
[33] KE W L, WANG Y, YE M. GRSA: service-aware flow scheduling for cloud storage datacenter networks[J]. China Communications, 2020, 17(6): 164-179.

[作者简介]

柯文龙（1989− ），男，安徽铜陵人，桂林电子科技大学博士生，主要研究方向为云存储系统网络、软件定义网络等。

王勇（1964− ），男，四川南充人，博士，桂林电子科技大学教授、博士生导师，主要研究方向为云计算、分布式存储系统、信息安全等。

叶苗（1977− ），男，广西桂林人，博士，桂林电子科技大学教授、博士生导师，主要研究方向为分布式存储、无线传感器网络、工程中的优化理论与方法、模式识别与机器学习。

陈俊奇（1997− ），男，湖南邵阳人，桂林电子科技大学硕士生，主要研究方向为云存储系统、软件定义网络等。