基于流分类的数据中心网络负载均衡机制

崔子熙[1],胡宇翔[1],兰巨龙[1],王 雨[2]
(1. 信息工程大学,河南郑州 450002; 2. 广东省新一代通信与网络创新研究院,广东广州 510670)

摘 要: 为充分利用数据中心网络的多路径带宽,现有研究多采用基于链路感知的负载均衡算法,在动态获取全局链路拥塞信息后选取最优路径对流量进行转发.然而这些研究未考虑数据中心网络流量大小分布不均匀的特性,难以在选路成本和转发效率上取得平衡.为此,设计一种基于流分类的数据中心网络负载均衡机制(ULFC,Utilization-aware Load balancing based on Flow Classification),在实现拥塞感知的基础上进行流量特征分析,采用不同的策略为大、小流分配路径,实现网络流量特征与选路方法优势的最佳匹配.实验结果表明,相比于现有方案,ULFC 的平均流处理效率提高了 1.3 倍至 1.6 倍,路由成本降低了 50% 以上.

关键词: 数据中心网络; 负载均衡; 可编程数据平面; 流分类; 可编程协议无关报文处理
中图分类号: TP393 **文献标识码:** A **文章编号:** 0372-2112 (2021)03-0559-07
电子学报 URL: http://www.ejournal.org.cn **DOI:** 10.12263/DZXB.20200199

Load Balancing Based on Flow Classification for Datacenter Network

CUI Zi-xi[1], HU Yu-xiang[1], LAN Ju-long[1], WANG Yu[2]
(1. *Information Engineering University, Zhengzhou, Henan* 450002, *China*;
2. *Guangdong Communications & Networks Institute, Guangzhou, Guangdong* 510670, *China*)

Abstract: In order to fully utilize the bandwidth of multi-paths of the datacenter network (DCN), the existing studies mostly adopt the congestion-aware load-balancing scheme, which forwards traffic along the optimal path after dynamically obtaining global congestion information. However, these works do not consider the non-uniform distribution of flow size and are difficult to strike a balance between the routing cost and the forwarding efficiency. This paper proposes ULFC, a utilization-aware load-balancing mechanism based on flow classification. By analyzing the characteristics of traffic, ULFC classifies the flows based on their sizes and assigns paths to them using different strategies, realizing the best matching between the characteristics of traffic and the advantages of the routing method. We evaluate ULFC with simulation and the results show that it outperforms the existing schemes in average flow-completion time (1.3 ~ 1.6 ×), while the routing cost has been reduced by more than 50%.

Key words: datacenter network; load balancing; programmable data plane; flow classification; programming protocol-independent packet processors (P4)

1 引言

目前,互联网中多达 80% 的数据流量位于大型企业和云提供商等部署的数据中心网络[1].为了实现服务器集群之间的高带宽、低时延通信,以交换机为核心的 Leaf-Spine 和 Fat-Tree 拓扑被广泛应用于数据中心网络设计.这类拓扑遵循 Clos 架构,能够充分利用交换机的处理性能,保证每一对通信的客户端与服务端之间存在多条等价路由[2].

典型的数据中心网络路由协议采用 ECMP(Equal-Cost Multi-Path)算法作流量调度[3].ECMP 从多条路径中随机选取一条分配给数据流,即仅从流的数量上对每条路径做平均,极易出现哈希冲撞引发的局部链路过度拥塞情况[4,5].此外,面对常见的链路失效问题,

收稿日期: 2020-02-24;修回日期: 2020-06-04;责任编辑:覃怀银
基金项目: 国家自然科学基金资助项目(No. 61521003, No. 61872382);国家重点研发计划课题(No. 2017YFB0803204);广东省重点领域研发计划项目(No. 2018B010113001)

ECMP 无法适应非对称拓扑导致其性能随网络负载水平的增加而急剧下降[6].

针对 ECMP 的上述缺陷,近年来研究人员从不同的切入点提出了多种改进的负载均衡算法. Chen 等人[7]提出了基于 OpenFlow 的集中式调度机制,但受限于南向通信瓶颈而不能对复杂的流量变化做出快速响应. Raiciu 等人[8]对端系统的传输层协议进行改造并提出了 MPTCP,打破 TCP 连接的单路径约束以期望高效利用多路径带宽,实际却增大了部署难度和网络边缘节点的拥塞. Katta 等人[6]提出了基于"巨型交换机"抽象模型的网络层协议 HULA(Hop-by-hop Utilization-aware Load balancing Architecture),通过定期向网内注入特殊的数据包(称为"探针"),以递归的方式逐跳实现拥塞感知和路由计算. 实验证明,相较于集中式调度或端到端协议栈改造的方案,在数据平面运行负载均衡算法能够达到更快的响应速率和更小的处理时延. 然而,HULA 使用分时择优的选路策略,交换机在一个时隙内仅维持一条最优路径处理所有流量会迅速拥塞该路径[9],带来严重的网络抖动问题.

基于以上分析,本文提出一种基于流分类的链路感知负载均衡机制 ULFC,并基于可编程数据平面技术开发了工程模型,主要具有以下两个方面的贡献:(1)实现了网络路由成本与网络转发效率的均衡;(2)实现了网络流量特征与选路策略优势的最佳匹配. 实验结果表明,与 ECMP 和 HULA 相比,ULFC 能够有效降低单条链路过度负载的概率,进一步减小平均流处理时间,提高网络的吞吐量.

2 ULFC 机制设计

Fat-Tree 拓扑具有高度工程化、同质化的特点[10],交换机按层次自上而下分别位于核心层、汇聚层和边缘层,其中汇聚层节点和边缘层节点组合构成 Pod,其数目 K 代表了整个拓扑的复杂度. 由于服务器直连边缘层接入网络,处于边缘层的 ToR(Top of Rack)交换机实际上充当着网关的角色. 任意一对 ToR 交换机之间存在多条等价的可达路径,为网内巨额流量转发提供保障. ECMP 算法由于采用简单、静态的路由策略已不能与该拓扑的特性相适应,而基于链路感知的自适应控制回环已经成为负载均衡算法设计的共识[6~10].

ULFC 在实现全局链路感知的基础上对数据流做细粒度处理,高效利用网络带宽资源的同时保证良好的自适应性和可扩展性. 首先,与 HULA 相似,ULFC 借助探针在网络内实现可达路径探测和拥塞信息扩散的同步进行. 在交换机内部,对探针的解析、更新和广播过程独立于用户数据流的处理过程,保证各节点之间能够互相获取所有可达路径信息及每条路径的拥塞情况. 其次,考虑到数据中心流量大小分布不均匀的特性,ULFC 根据实时网络设定动态流分类阈值,依据待处理流的大小进行分类,对标记的"大流"和"小流"分别使用不同的路由策略规划路径,满足不同尺寸的流对传输性能的需求. 此外,为尽可能降低接收端的重排序时延,ULFC 依据 RTT(Round Trip Time,网络往返时延)将流分割为更小的单位进行细粒度处理,在网络层即可避免 TCP 报文的乱序问题.

2.1 可达路径探测与评价

如图 1 所示,该网络拓扑的 $K=2$,任意一对不同 Pod 下的 ToR 交换机之间存在 4 条不同的可达路径,每个 ToR 交换机按期向网内注入探针以更新其他节点到本节点的路由信息.

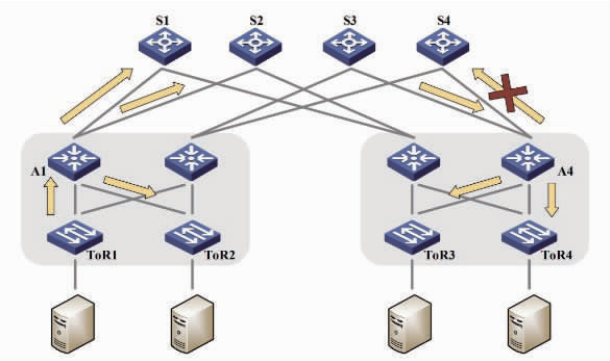


图1 可达路径探测示意图

为了搜索通信双方之间的所有路径,探针的复制和广播基于 Fat-Tree 拓扑固有的分层属性完成[6]:当探针从下层端口进入交换机时,交换机会将探针从其他所有端口进行洪泛;当探针来自上游端口时,交换机仅将其广播给下层交换机. 以 ToR1 为例,其产生的探针沿链路向上进入网络. 对于汇聚层节点,与 ToR1 属同一 Pod 的 A1 会将探针复制到除入端口外的所有端口,而另一个 Pod 中的 A4 只会将来自核心层的探针传递给下游交换机. 最终探针到达其他 ToR 交换机时即完成一个路由更新周期.

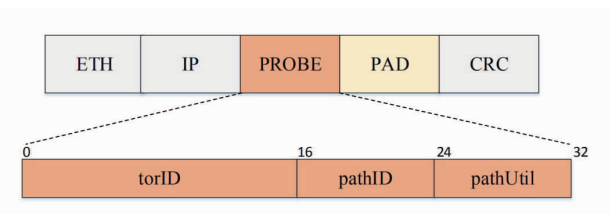


图2 探针头域信息

探针在传播时需要记录瓶颈链路的带宽利用率,作为交换机评价可用路径优劣性的唯一指标. 在此定义瓶颈链路为一条路径上负载率最高的链路:瓶颈链路带宽利用率越低,剩余可用带宽越高,整条路径越优. ULFC 改

造常规 IP 数据包设计探针数据包格式. 如图 2 所示,数据帧长度为64 字节,附加在以太网帧头域和IP 数据报文头域之后的探针头域占用4 个字节,包括以下信息域:

torID (16 bits)　记录生成该探针的 ToR 交换机 ID,其他交换机据此识别探针的来源.

pathID(8 bits)　记录探针途经节点的校验值,下游交换机据此辨别不同的可达路径.

pathUtil (8 bits)　记录探针已搜索路径上瓶颈链路的带宽利用率,交换机据此评估路径的拥塞情况.

与"全部链路成本之和"相比,采用"瓶颈链路成本"作为评价指标避免了个别过载链路的负担可能进一步加重的问题,也降低了转发节点在处理探针过程中的的计算复杂度. 当探针按照既定多播规则由源 ToR 交换机到达其他 ToR 交换机时,所有可达路径均被遍历,沿途交换机记录所有探针的入端口为可达路径集合. 因此,在之后的路由阶段,数据包的转发路径均为探针传播的反向路径.

2.2　拥塞感知与路由更新

为了实现准确的链路拥塞感知,交换机的各个端口在接收任意数据包时更新入端口直连链路的拥塞信息. 目前广泛采用 EWMA(Exponentially Weighted Moving Average,指数加权移动平均)算法[6]计算链路带宽利用率 U:

$$U = D + U \times \left(1 - \frac{\Delta t}{\tau}\right) \tag{1}$$

其中 D 表示数据包的字节数,Δt 表示到达时间间隔,τ 是与 RTT 相关的常量,决定拥塞感知的响应灵敏度[10].

由于头部携带路径评价信息,探针的到达除了会增加链路负载,还可能会触发交换机对路由表进行更新. 网络初始化时,每个交换机各自维护以 torID 为索引项的路由表,表项包括可达路径信息(availPath, nextHop)和最优路径信息(bestHop,minUtil). 当交换机从端口 i 收到探针时,路由表更新算法如下:

算法 1　最优瓶颈链路路由更新算法

Step1　解析探针头部信息,将可达路径 pathID 和端口 i 记入 torID 对应的表项(availPath,nextHop);

Step2　依式(1)计算端口 i 直连链路的带宽利用率 linkUtil;

Step3　取 pathUtil 和 linkUtil 的较大值赋给 pathUtil,同时计算新的路径校验值 pathID;

Step4　查询 torID 对应的最优路径信息;

Step5　比较 minUtil 和 pathUtil 的大小. 若 pathUtil 小于 minUtil 则继续执行 Step6,否则直接跳转至 Step7;

Step6　将最优路径信息进行更新为端口 i 和 pathUtil;

Step7　按照既定规则对探针进行广播,算法结束.

其中,Step1 ~ Step3 将本节点加入探针的已搜索路径,完成对探针头域的更新. 而在 Step5 中,当 pathUtil 大于 minUtil 时会跳过对最优路径信息的更新,保证交换机总是按照"最小瓶颈链路利用率"的标准从所有可达路径中选出最优路径.

由于探针的生成具有周期性,随之生成的路由表项也具有一定的有效期. 交换机更新路由表项时会记录探针到达时间戳,一旦发生链路失效导致在 T_{fail} 时间内没有新的探针到达对计时器进行重置,交换机将自动删除该表项. 这种链路失效消息同样按照探针传播的路径迅速向前传递,避免上游节点继续分配流量造成下游链路拥塞.

2.3　阈值计算与流判定

在一个路由更新周期内,目的 ToR 交换机会收到多个源地址相同但搜索路径不相同的探针,探针的数量即为可达路径总数,每个探针携带着一条端到端完整路径信息及其瓶颈链路的带宽利用率. 因此 ULFC 在边缘节点设置阈值计算模块和流判决模块. 其中阈值计算模块收集来自同一 ToR 的探针,解析探针头部信息并计算流分类阈值,如式(2)、(3).

$$\bar{C} = \frac{1}{N}\sum_{l \in E} C_l = \frac{1}{N}\sum_{l \in E} (1 - U_l) \times W \tag{2}$$

$$Q = k \times \bar{C} \tag{3}$$

其中,N 表示可达路径总数,E 表示所有可达路径的集合,U_l 表示可达路径 l 上瓶颈链路的负载率,W 表示链路固有带宽,$\bar{C}$ 表示所有可达路径平均剩余可用带宽,Q 表示流分类阈值,k 为阈值系数.

该算法之所以依据瓶颈链路的带宽利用率计算流分类阈值,是因为瓶颈链路剩余可用带宽的平均值可以真实反映当前网络的负载状况. 如式(3)所示,流分类阈值 Q 与平均剩余可用带宽 $\bar{C}$ 成正比——网络负载水平越高,平均剩余可用带宽越小,流分类阈值随之变小,更多的流被判定为"大流",需要谨慎为其规划确定性路径. 由于网络负载的动态变化直接影响流分类阈值,阈值系数 k 是决定整个分类处理机制性能的关键参数. 根据文献[11]的研究,一般认为数据中心网络大小流阈值 S 为 1MB. ULFC 使用该值作为 Q 的初始值,即当网络的负载为空时,有 $Q = S = 1$MB. 因此阈值系数 k 应设置为 S 与链路固有带宽 W 的比值. 举例来说,假设链路固有带宽 $W = 100$Mbps,则 $k = S/W = 1\%$.

当有新的待处理流进入数据中心网络时,流判决模块查询其目的节点对应的阈值后对该流分类,字节数大于阈值的流被标记为"大流",其余标记为"小流". 流分类阈值 Q 的设定本质上反映了不同字节数的流量对当前网络负载影响的不同. 所有交换机对"大流"采用最优路径进行传输,至于"小流"则运行哈希算法从

可达路径中任意为其分配一条路径.

2.4 流的转发

由于探针携带的链路状态信息在一条端到端路径上传递存在时延,且各层交换机分布式完成路由决策,同一条流的数据包可能会先后按照不同的路径进行转发. 相较于采用最短路径算法(如 OSPF)的骨干网,TCP 数据包乱序问题在多路径传输的条件下变得尤为突出.

基于文献[5]的研究,ULFC 将流(flow)分割为更小的单位(flowlet)规划路由,尽可能保证数据包按序到达接收端. 对于任意一台交换机,flowlet 的定义为:如果一条流中两个相邻数据包的到达时间间隔足够大,以致于后序的数据包即使采用更优路径也不会早于前序的数据包到达接收端,则在数据包之间标记一个分割点,两个分割点之间的所有数据包属于同一 flowlet. 整个 TCP 报文的处理流程如图 3 所示,交换机记录 flowlet 路由信息,强制同一 flowlet 的数据包采用相同端口转发.

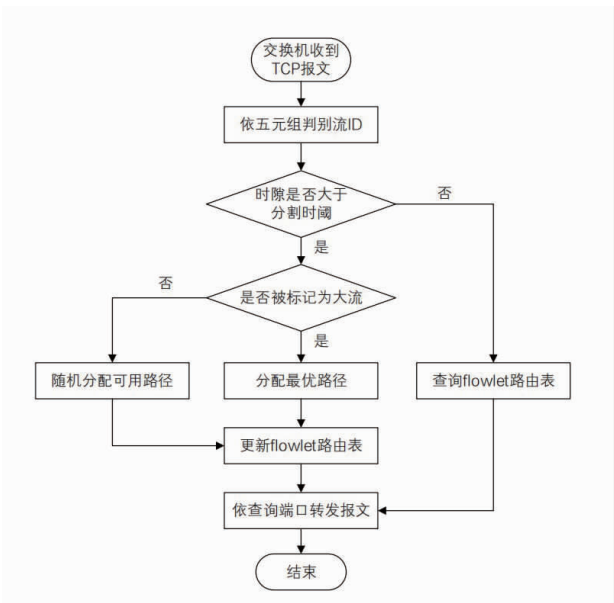


图3 TCP报文处理流程图

理想条件下,分割时阈 T_f 达到端到端最大时延可完全消除 TCP 报文因转发路径不同造成的时间差[10],减少应用层的重排序时延. 同时 flowlet 的引入为 ULFC 带来两个改变:(1)数据平面需要维持 TCP 流量状态信息(在 P4 中使用 register 完成),整个负载均衡算法由无状态变为有状态;(2)实现了细粒度的负载均衡,大流的冲击效应进一步弱化,流传输效率和系统稳定性大大提高.

2.5 参数讨论

探针是实现拥塞感知的关键一招,也是路由计算的先决条件. 合理的探针生成频率既要保证网内拥塞信息更新的时效性,也要考虑探针传播的带宽资源代价.

由于交换机仅在检测到新的 flowlet 时才会使用最新计算的路由表,探针生成频率 T_{probe} 与分割阈值 T_{flowlet} 相关. 结合 TCP 控制理论对实验逻辑进行分析,得出三个时间阈值存在以下约束关系:

$$\text{RTT}_{\max} \leqslant T_{\text{flowlet}} < T_{\text{probe}} \leqslant T_{\text{fail}} \tag{4}$$

根据文献[6]的总结,任意链路上探针传播的平均带宽代价为

$$P = \mu \frac{\text{probeSize} \times \text{numToRs}}{\text{probeFreq} \times \text{linkBandwidth}} \tag{5}$$

其中 probeSize 为探针的大小(64 字节),numToRs 为拓扑中 ToR 交换机的数量,μ 表示一个广播周期内每条链路收到的平均探针个数. 在给定的探针广播机制下,$\mu \approx 1.5$.

为了保证流传输效率,HULA 不得不保证网络状态在任意时刻以 200μs(RTT 的 1 ~ 10 倍)的频率进行更新,否则最优路径将迅速拥塞,造成严重的网络抖动. 尽管 HULA 在不影响算法逻辑的前提下对探针的传播机制做进一步优化(使得 μ 下降为 1),但在大规模拓扑中依然需要承担 3% 以上的固定链路带宽代价. 而 ULFC 能够在一个路由计算周期中获取更多路径信息熵,在 T_{probe} 的设定方面有较大的弹性. 特别是在网络负载率持续维持在较低水平时,由 3.4 节的实验结果可知,提高网络更新频率只会牺牲链路带宽换取系统对链路故障的快速响应,不会对流传输产生加速作用.

3 实验评价

本文基于 P4[12] 语言完成 ULFC 负载均衡机制模型. 为了评估 ULFC 的负载均衡性能,在相同环境下设置 ECMP 和 HULA 两种方案作为对照组.

3.1 实验设计

实验平台使用仿真工具 Mininet 构建网络环境,拓扑采用 $K=4$ 的 Fat-Tree 拓扑,其中交换机均为支持 P4 编程的 Bmv_2. 由于软件交换机的转发性能有限[13],交换机之间的链路带宽设置为 400Mbps,每个边缘层交换机与 4 个直连主机之间的链路带宽为 200Mbps.

根据第 2 节的分析,决定系统性能的关键参数默认值如表 1.

表 1 实验参数设置

参数	数值
探针生成频率 T_{probe}	50ms
流分割阈值 T_{flowlet}	10ms
拥塞感知灵敏度 τ	500μs
阈值系数 k	0.25%

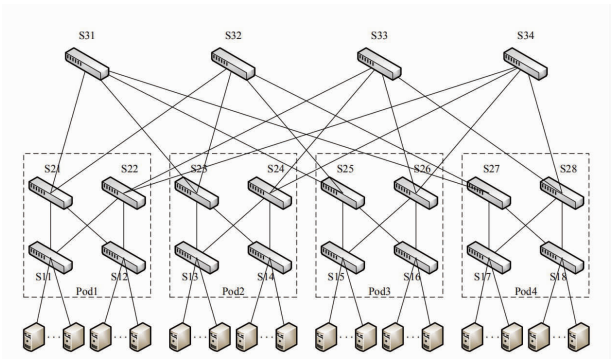


图4　实验拓扑

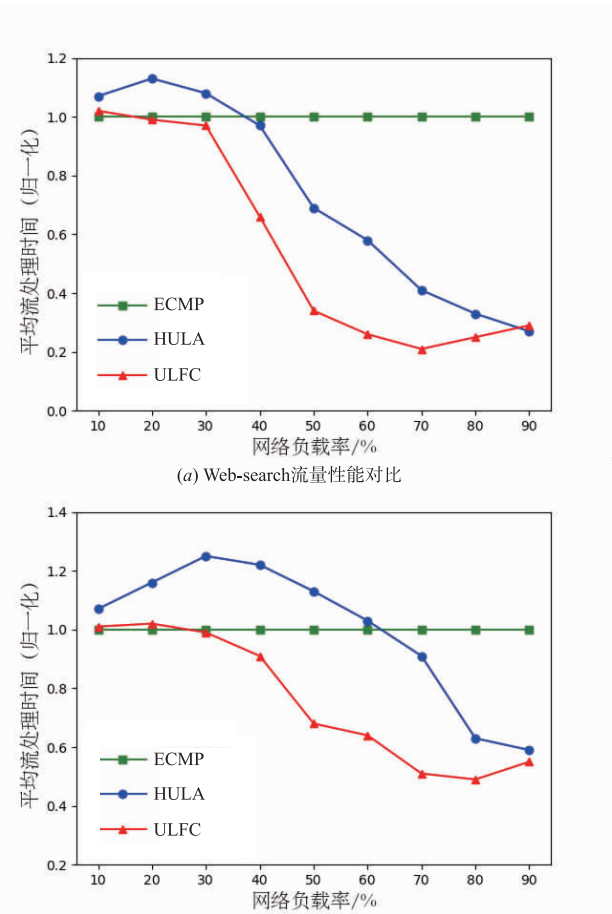
使用 Web-search 流量[14]和 Data-mining 流量[15]模拟网络中的真实流量. 两种流量的分布函数确定,均符合典型的数据中心网络流量特征. 不同点在于,80% 的 Data-mining 流量其尺寸小于 10KB,而大部分 Web-search 流量集中在 10KB ~ 10MB 之间.

采用服务器-客户端模式进行仿真实验:使用 Iperf 工具,每个主机作为客户端随机选择一个位于不同 Pod 的主机建立 TCP 连接,保证所有的流传输均经过核心层交换机. 流的生成时间间隔服从指数分布,通过调整指数分布的平均值达到期望的网络负载水平(10% ~ 90%). 以平均流处理时间为性能指标对三种机制进行比较评价,每轮实验重复 10 次,计算 10 次结果的平均值为最终结果. 除非特别说明,所有的实验数据均以 ECMP 为基准做归一化处理.

3.2　对称拓扑测试

图 5 显示了三种方案在对称拓扑下面对两种不同类型负载流量的平均流处理时间. HULA 和 ULFC 能够收集网络拥塞信息动态决策路由,整体性能优于 ECMP. 当网络负载处于低水平时,链路可用带宽资源充足,三种方案的平均流处理时间持平,其中 ECMP 与 ULFC 能够充分发挥多路径传输的优势,而 HULA 严格选择单一路径导致性能较差. 随着网络负载率的升高,设置路径优劣评估的作用逐渐显现. ULFC 的阈值也随之逐渐减小,更多的流被判定为大流而通过最优路径进行转发. 当网络负载量达到 80% 以上时,ULFC 几乎与 HULA 等价,两种方法的平均流处理时间远远小于 ECMP.

如图 5(*a*)所示,当背景流量为 Web-search 流量且负载率达到 70% 时,HULA 的平均流传输时延比 ECMP 减小了接近 80%,这是因为严格匹配最优路径的策略更适合于网络负载较高情况下对链路拥塞状态影响较大的大流进行调度处理. ULFC 集中了 ECMP 与 HULA 两种方案的优势,面对任意类型的流量负载均取得最短的平均流处理时间. 在图 5(*b*)中,对于平均字节数较小的 Data-mining 流量,在网络负载率为 40% 至 80% 的



(*a*) Web-search流量性能对比

(*b*) Data-mining流量性能对比

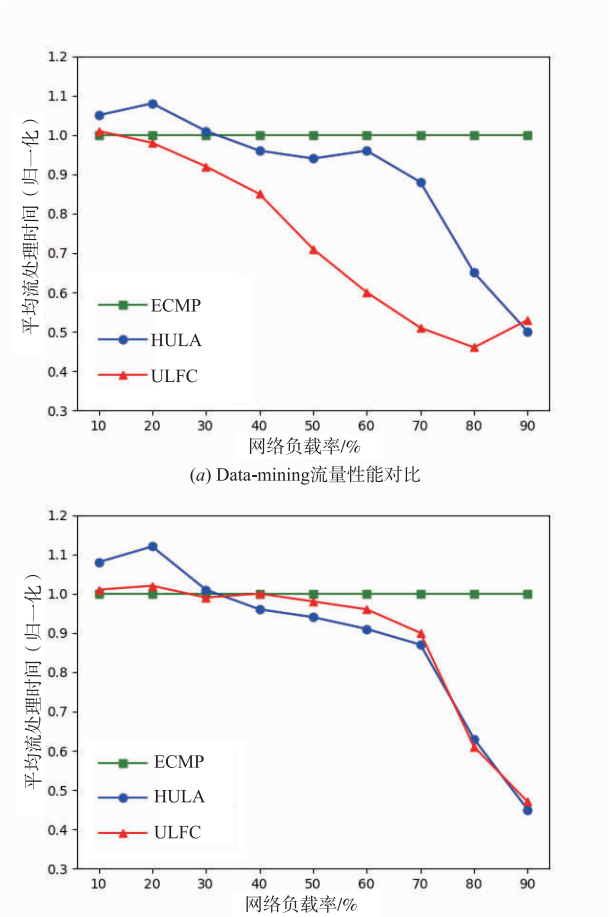
图5　三种方案在对称拓扑下的性能对比

区间内,ULFC 能够随负载水平的提高迅速取得较好的均衡效果,始终逼近问题最优解. 当网络负载率为 70% 时,ULFC 的平均流处理时间较 ECMP 减少了 49%,较 HULA 减少了 44%.

3.3　非对称拓扑测试

为了测试三种机制在非对称拓扑下的性能,断开核心交换机 S34 与汇聚交换机 S28 之间的链路,同时统计对象范围缩小为所有经过 Pod4 的 Data-mining 流量.

从图 6(*a*)可以看出,在网络带宽资源下降 25% 的情况下,ECMP 的性能衰退问题更加严重,择优选路策略的优势进一步放大,使得 HULA 能够迅速缩小与 ECMP 的差距并实现反超. 而 ULFC 依旧保持着稳定的性能曲线,在 70% 的网络负载率下流传输效率较前两种方法分别提高了 96% 和 72%. 图 6(*b*)显示了 Data-mining 流量中数据量小于 100KB 的平均流传输时间. 当网络负载率小于 40% 时,所有的流量均被判决为小流,此时 ULFC 与 ECMP 等价;随着网络负载率提高至一定程度,流分类阈值开始小于 100KB,ULFC 的性能曲线逐渐与 HULA 的性能曲线拟合,两者在网络负载率大于 70% 之后重叠. 综合两张图可知,ULFC 的动态阈值算



(*a*) Data-mining流量性能对比

(*b*) Data-mining小流（<100KB）性能对比

图6 三种机制在非对称拓扑下的性能对比

法对网络负载变化的反应比较灵敏，在剩余带宽资源紧张的情况下一些尺寸不大的流也被视作“大流”进行择优选路，有效缓解了高并发小流的尾延迟现象.

结合前一小节的结果不难看出，流分类处理机制不是简单地对现有方案增量合并，而是基于不同尺寸的数据流造成的网络波动不同的考虑，将两种选路策略的优势与流量特征结合起来，特别是解决了“大流”对网络波动影响较大这一主要矛盾，各条链路上的网络负载水平始终保持相对平稳的均衡状态.

3.4 探针生成频率测试

为了探究参数 T_q 对负载均衡性能的影响，在非对称拓扑和 Data-mining 流量的环境下，测试 ULFC 在探针生成频率分别为 20ms、50ms、100ms 和 200ms 时的平均流处理时间，同时实验结果以 HULA-1ms 作为最优解进行归一化处理.

由图 7 可知，在网络负载率小于 30% 的情况下，探针生成频率的改变对 ULFC 的流传输效率几乎没有影响. 随着网络负载率提高至 50% 以上，大流的哈希碰撞问题凸显，此时适当提高探针生成频率意味着交换机能够及时更新路径信息，为流量选择真实的最优路径

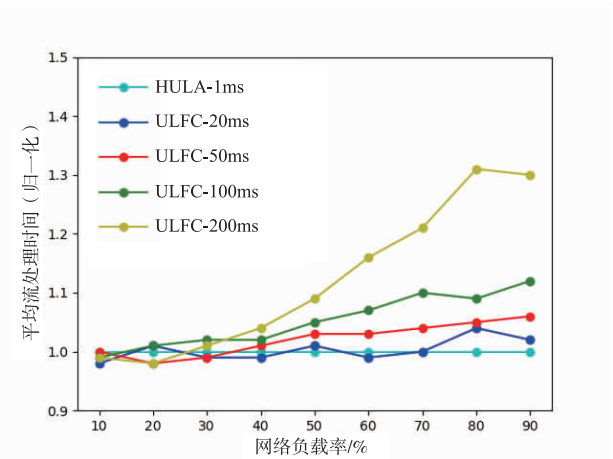


图7 ULFC在不同探针更新频率下的性能对比

进行传输. 当探针生成频率提高至 20ms 时，ULFC 能够取得与 HULA-1ms 等价的性能曲线，逼近理想的流量均衡效果. 此外，当网络负载率为 90% 时，ULFC-200ms 的平均流处理时间较 ULFC-50ms 仅提高 21%，较 ULFC-20ms 仅提高 25%，充分说明 ULFC 在探针生成频率的设定方面具有较高的弹性.

4 结束语

本文基于可编程数据平面技术提出一种面向数据中心网络的负载均衡机制 ULFC，同时考虑链路状态信息与流量特征. ULFC 结合网络发展趋势，将流量调度算法分布式部署于全网，在实现全局链路感知的基础上对大、小流进行分类处理，充分发挥了不同路由算法的优势. 实验结果表明，与现有方案相比，ULFC 能够高效利用多路径的带宽资源，在任意网络负载状态下实现流量快速转发，在系统响应性和稳定性方面具有明显优势.

参考文献

[1] William S. Foundations of Modern Networking: SDN, NFV, QoE, IoT, and Cloud [M]. New Jersey: Addison-Wesley Professional, 2015.

[2] Al-Fares M, Loukissas A, Vahdat A. A scalable, commodity data center network architecture [J]. ACM SIGCOMM Computer Communication Review, 2008, 38(4): 63 – 74.

[3] 陈果，张潍丰. ELAB：基于端系统的新型拥塞感知负载均衡机制[J]. 通信学报，2019，40(03)：196 – 205.

[4] Al-Fares M, Radhakrishnan S, Raghavan B, et al. Hedera: dynamic flow scheduling for data center networks [A]. USENIX Symposium on Networked Systems Design and Implementation[C]. San Jose, CA, USA: USENIX, 2010. 10 – 19.

[5] Kandula S, Katabi D, Sinha S, et al. Dynamic load balan-

cing without packet reordering[J]. ACM SIGCOMM Computer Communication Review,2007,37(2):51 – 62.
[6] Katta N,Hira M,Kim C,et al. Hula:scalable load balancing using programmable data planes[A]. The Symposium on SDN Research[C]. Santa Clara,CA,USA:SOSR 2016. 1 – 12.
[7] 陈鸣,胡慧,刘波,邢长友,许博. 一种基于 OpenFlow 的多路径传输机制[J]. 电子与信息学报,2016,38(05):1242 – 1248.
[8] Raiciu C ,Barre S,Pluntke C ,et al. Improving datacenter performance and robustness with multipath TCP[J]. ACM SIGCOMM Computer Communication Review, 2011, 41(4):266 – 277.
[9] Wang P,Xu H,Niu Z,et al. Expeditus:congestion-aware load balancing in clos data center networks[J]. IEEE Transactions on Networking,2017,PP(99):1 – 14.
[10] Alizadeh M,Edsall T,Dharmapurikar S,et al. CONGA:distributed congestion-aware load balancing for datacenters[J]. ACM SIGCOMM Computer Communication Review,2014,44(4):503 – 514.
[11] Kandula S,Sengupta S,Greenberg A G,et al. The nature of data center traffic:measurements & analysis[A]. ACM SIGCOMM Conference on Internet Measurement Conference[C]. Chicago,Illinois,USA:ACM 2009. 202 – 208.
[12] Bosshart P,Daly D,Gibb G,et al. P4:Programming protocol-independent packet processors[J]. ACM SIGCOMM Computer Communication Review,2014,44(3):87 – 95.
[13] P4lang. Performance of bmv2[EB/OL]. https://github.com/p4lang/behavioral-model/blob/master/docs/performance.md,2019-11-18.4
[14] Alizadeh M,Greenberg A,Maltz D A,et al. Data center TCP (DCTCP)[J]. ACM SIGCOMM Computer Communication Review,2010,40(4):63 – 74.
[15] Greenberg A,Hamilton J R,Jain N,et al. VL2:A scalable and flexible data center network[J]. Communications of the ACM,2009,54(3):95 – 104.

作者简介

崔子熙　男,1996 年生于河南焦作. 现为战略支援部队信息工程大学硕士研究生. 主要研究方向为可编程数据平面、软件定义网络.
E-mail:czxndsc@163.com

胡宇翔　男,1982 年生于河南周口. 现为战略支援部队信息工程大学副教授、博士生导师. 主要研究方向为新兴网络体系结构、路由与交换技术.
E-mail:ndschyx@163.com

兰巨龙　男,1962 年生于河北张家口. 现为战略支援部队信息工程大学教授、博士生导师. 主要研究方向为未来信息通信网络关键理论与技术.
E-mail:ndscljl@163.com

王　雨　男,1975 年生于河南许昌. 现为广东省新一代通信与网络创新研究院研究员. 主要研究方向为软件定义网络.