

doi:10.13756/j.gtxyj.2018.03.002

光通信系统与网络技术

SDN 架构下数据流量调度算法的设计

许文庆^{1,2}, 余 庚^{1,2}

(1. 福州理工学院, 福州 350506; 2. 福建工程学院 国脉信息学院, 福州 350014)

摘要:目前云技术被广泛应用于数据中心网络(DCN),为确保 DCN 服务质量(QoS),等价多路径路由(ECMP)算法和动态负载均衡(DLB)算法被作为解决方案。然而这些算法仅能获得局部优化效果,将其应用在 DCN 重载环境下时,得到的传输时延和带宽利用率等指标不尽如人意。文章针对这些算法的局限性,从全局优化的角度提出一种软件定义网络(SDN)架构下的胖树网络数据流量载荷均衡调度算法。该算法通过 Ryu 控制器监视全局实时参量来评估 SDN 节点和链路载荷情况,选出最符合流量载荷需求的最优转发路径。实验表明,所提算法在改善 DCN 服务性能方面较等价多路径路由算法和动态负载均衡算法有显著提升。

关键词:软件定义网络;数据中心网络;评估;均衡;仿真

中图分类号:TP393 **文献标志码:**A **文章编号:**1005-8788(2018)03-0005-04

Design of Data Flow Scheduling Algorithm Under SDN Architecture

Xü Wen-qing^{1,2}, YU Geng^{1,2}

(1. Fuzhou Institute of Technology, Fuzhou 350506, China;

2. Guomai Information College, Fujian University of Technology, Fuzhou 350014, China)

Abstract: Recently, cloud technology is widely used in Data Center Network (DCN). In order to ensure Quality of Service (QoS) in DCN, Equal Cost Multipath Routing (ECMP) algorithms and Dynamic Load Balancing (DLB) algorithms are employed as solutions. However, these algorithms only obtain local optimization results. In particular, the transmission delay and bandwidth utilization are not satisfactory when these algorithms are applied in the DCN overloaded environment. Considering the limitations of these algorithms, this paper proposes a load balancing scheduling mechanism for fat tree network data traffic under Software Defined Network (SDN) architecture from the perspective of global optimization. The mechanism evaluates the SDN node and link load by monitoring the global real-time parameters through the Ryu controller, and selecting the optimal forwarding path which meets the demand of traffic load. Experiments show that the scheduling mechanism proposed in this paper has significantly improved the performance of DCN services compared with the ECMP/DLB algorithm.

Key words: SDN; DCN; evaluation; balance; simulation

0 引 言

随着云技术、物联网技术和虚拟化技术的不断融合发展,海量宽带数据流量将以指数级增长呈现在数据中心网络(Data Center Network, DCN)上。为了均衡多径数据流量,提升网络服务质量(Quality of Service, QoS)^[1], DCN 中大多选用等价多路径路由(Equal Cost Multipath Routing, ECMP)算法和动态负载均衡(Dynamic Load Balancing, DLB)算法。前者属于静态散列算法,无法根据实时链路状况调整数据流量,因此可能导致网络拥塞,且最多允许 16 条等价路径负载均衡。同时,ECMP 采用哈希(Hash)算法使得每条链路对数据流量实行等量分配。如果调整节点数量则要重新分布 Hash,这将终结当前传输控制协议(Transmission Control Protocol, TCP)连接。尤其当 DCN 处于忙时段时,网络流量载荷的剧增也势必增加路由路径的复杂度,

这样一来,DCN 上交换节点的路由表也随之增多,这将直接导致存储代价以及延时问题。这些问题究其原因,都是因为流量载荷均衡的各条链路是完全等价的。DLB 算法是通过读取 OpenFlow 控制器所收集的交换机上传流统计信息,再由控制模块中的流量载荷均衡单元实施调度。当交换机将数据上传到 OpenFlow 控制器后,控制器中的均衡单元对数据包的源/目的地址、接收数据包的交换机的输入/出端口进行分析,进而计算出该条流所要访问的最高层节点,最后为该数据流适配出一个完整的传输路径。这种算法的选路思想应用在复杂流量负载的网络中很可能导致网络阻塞。因为这种算法是在无法预知其他路由带宽闲置情况下来简单地确定向上传输方向的最佳路径,这显然缺乏全局性。鉴于上述常用算法策略的不足,本文从全局角度出发,提出一种基于软件定义网络(Software Defined Network, SDN)的胖树网络数据流量均衡调度算法。

收稿日期:2017-08-23

基金项目:福建省教育厅中青年教育科研项目资助(JAT170802, JAT160623)

作者简介:许文庆(1977—),男,福建福州人。讲师,硕士,研究方向为系统分析设计。

1 SDN 体系架构

如图 1 所示,SDN 在架构上分为转发层、控制层和应用层。从图中不难看出该架构与传统网络有所不同。SDN 交换设备不具备网络控制功能,它将控制功能移至控制层。转发层中的各类网络基础设备通过 OpenFlow 协议与控制层相连。控制层由多个 Ryu 控制器组成,通过应用编程接口 (Application Programming Interface, API) 可周期性地调用全局感知模块和网络状态测量模块,并通过 OpenFlow 交换机^[2]将实时监视到的拓扑、时延和带宽等全局参量送到机制评估模块,以实现全局流量工程的均衡控制。当 Ryu 控制器与 Mininet 仿真平台下的虚拟网络连接时,整个 DCN 环境都将呈现,这有助于流量均衡控制的实施。在应用层,网管可通过 API 来实施业务需求和策略评估,如流量工程评估、QoS 和最佳转发路径评估等。

SDN 通过实时获取全局信息既保证了网络服务链路带宽需求,又实现了流量负载的灵活调度。胖树结构的 DCN 具有多径特性,很适合 SDN 优势的发挥。因此将两者结合起来,动态选择路由由路径进行网络流量负载分流操作实现均衡的方案是可行的。

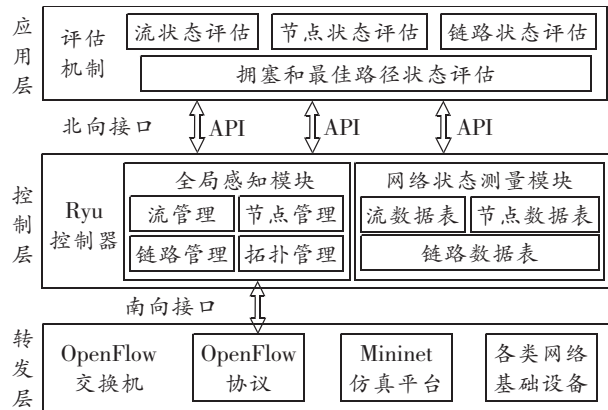


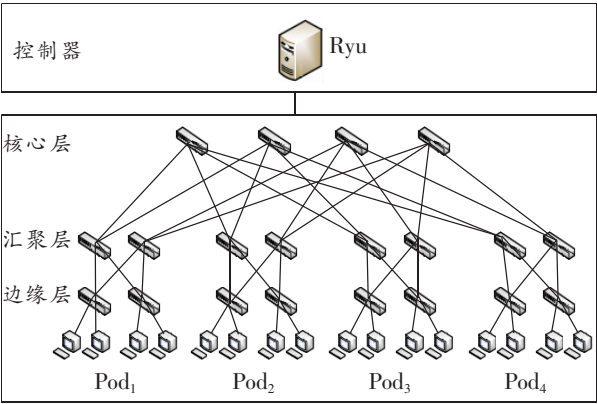
图 1 SDN 体系架构

2 流量载荷调度原理

2.1 DCN 数学模型

结合图 1 中 SDN 的 3 层结构构建一个由 4 个集群 ($k = 4$) 组成的胖树网络中心,并通过在 Mininet 仿真平台中编写 Python 脚本来搭建虚拟节点。如图 2 所示。

为了便于分析,做如下假设:所有交换机集合表示为 S_{ALL} ,单点交换机表示为 S_{Node} ,所有链路集合



注:Pod 为胖树网络中的终端群或转发站。

图 2 4 元胖树网络结构

表示为 L_{ALL} ,单条链路表示为 l ,节点和链路带宽均为 B ,所有数据流量集合表示为 D_{ALL} ,第 i 个交换节点已占用的带宽和带宽利用率分别表示为 $BU(S_{Nodei})$ 和 $BR(S_{Nodei})$ 。链路最大负荷表示为 $MAX_p(l)$,链路关键度表示为 $\rho(l)$,链路平均期望负荷表示为 $AVE(l)$,链路闲置带宽表示为 $B_{Idle}(l)$ 。据此构建矩阵 $\mathbf{X} = D_{ALL} \times L_{ALL}$,其中 $x(D_i, l_i) \in (0, 1)$ 表示第 i 个数据流是否途径第 i 个链路; $\mathbf{Y} = D_{ALL} \times S_{ALL}$,其中 $y(D_i, S_{Nodei}) \in (0, 1)$ 表示第 i 个数据流是否途径第 i 个交换节点。

2.2 评估交换节点

评估的目的在于设置交换节点转发阈的上限。路由模块参照交换节点带宽使用情况,首先淘汰掉那些数据流量负载靠近转发阈^[3]上限的交换节点,再结合节点最大转发带宽和闲置带宽计算出各个节点的可使用带宽,最后锁定最佳转发交换节点。转发阈值上限决定了节点最大转发带宽。由于途径 S_{Nodei} 的数据流量具有突发性,本次选取绝对中位差 (Absolute Median Deviation, AMD) 来近似表征该节点的数据流量稳定度。当前交换节点的 AMD 表示为

$$AMD = median(|y(D_i, S_{Node}) \cdot B(D_i) - median(y(D_i, S_{Node}) \cdot B(D_j))|)。$$

AMD 值与当前节点数据流的稳定性呈反比。若引入调度参数 α ,则可得当前交换节点转发阈上限表征: $T(S_{Node}) = 1 - \alpha \cdot AMD$ 。调度参数可以适当调整转发阈的上限,当 AMD 越大时,途径该交换节点的数据流量呈现出高度突发性,此时应将该交换节点的转发阈上限下调,以防止当前交换节点由于超负荷导致时延^[4]代价;反之则上调。此时当前节点的带宽利用率须满足

$$BR(S_{Nodei}) = BU(S_{Nodei}) / B(S_{Nodei})，$$

$$BR(S_{Node_i}) < T(S_{Node_i})。$$

据此,可得当前交换节点的可用带宽为 $T(S_{Node_i}) \cdot B(S_{Node_i}) - BU(S_{Node_i})$ 。

2.3 评估链路

网络中的许多因素都会造成链路上数据流量阻塞,要解决这个问题,在评估^[5]链路时就必须考虑链路关键度 $\rho(l)$,以便为链路设置最大负载。将网络中所有流过链路 l 的流量带宽之和与数据流数量之比定义为 $AVE(l)$,则有

$$AVE(l) = \sum_{i=1}^n x(D_i, l) \cdot B(D_i) / \sum_{i=1}^n x(D_i, l),$$

此时 $\rho(l) = AVE(l) / B(l)$ 。 $AVE(l)$ 越大, $\rho(l)$ 越高,该链路越会被锁定为转发路径,并可为该链路设置更大的 $MAX_P(l)$ 。考虑到不同节点链路对最大带宽需求不同,此处引入需求系数 β ,则有

$$MAX_P(l) = \beta \cdot \rho(l),$$

$$B_{idle}(l) = MAX_P(l) \cdot B(l) - \sum_{i=1}^n x(D_i, l) \cdot B(D_i)。$$

对链路进行评估^[6]需在两个约束的前提下开展:(1) 链路对带宽的总需求要比链路当前闲置带宽小;(2) 对任意节点而言,输入口和出口口的数据流量保持恒定。

2.4 调度过程

首先,通过 SDN 控制器监视全局所有链路和节点的实时参数,当节点发出的数据流量或链路上的数据流量想要重新选路时,根据 IP 地址为该数据流构建路由信息库。然后,再根据节点转发阈上限和链路最大载荷约束,丢弃那些闲置带宽不足以及超载节点的路径。最后,在剩下的路由中,根据交换节点评估策略和链路评估策略选择“瓶颈”带宽最大的路由通道转发该数据流量。“瓶颈”带宽大小是由路由路径上所有链路带宽的最小值和节点带宽的最小值综合来确定的,“瓶颈”带宽最大的那个路由被流量载荷调度机制选为最佳转发路由,通过这种方式实现流量载荷均衡调度的目标。

3 方案评估

3.1 测试方案

本次测试基于 Mininet 2.2.1 测试环境。测试过程选择一个 4 元^[7]的胖树拓扑网络。该网络含 20 台交换机和 16 台主机,控制器和交换机分别选用 Ryu 和 OpenSwitch。通过调用命令: `$ sudo mn - custom ~ / mininet / custom / fattree. py - topo = mytopo - switch = ovsk - mac - controller = remote,`

`ip = Ryu` 控制器远程主机 IP 地址, `port = 6 633`,使控制器 Ryu 和仿真平台 Mininet 在两个主机上分开运行以便共同承担实验机器生成的数据流量。

在 Mininet 仿真平台生成拓扑和 Ryu 控制器连接后,使用 Iperf 灌包软件作为用户数据报协议 (User Datagram Protocol, UDP) 数据流量发生源。在随机模式下,任意选择一对源/宿主机来发送 UDP 流。Iperf 参数中,流量载荷服从 $[10 \text{ Mbit/s}, 10^7 \text{ Mbit/s}]$ 分布,且在该区间内,流量载荷分布呈现出逐步增大趋势。每次生成数据流量的时间间隔为 5 s,每条数据流量持续 60 s。UDP 长度设置为 128 Byte,每隔 1 s 搜集一次交换机端口的数据流参量。

当前,在基于胖树拓扑结构的 DCN 上运用的多径数据流量路由机制有 ECMP 算法和 DLB 算法。ECMP 算法将等价链路上的流量载荷进行等额分配的做法虽然能提高局部网络的带宽和吞吐量,但未能及时地根据路由链路即时状态的变化做出载荷比例上的合理调度;DLB 算法则是在监测到路由链路中存在较大的数据流量载荷时就立即对所有有效路由路径展开搜索,一旦找到满足带宽需求的那条链路就马上将其选作转发路由,不再继续往下搜索和比较是否有更适合的链路,显然这种缺乏比较的思想是没有考虑到网络全局负载状态。针对这些算法的差异,本文通过节点超载机会、平均时延和平均带宽利用率 3 个指标来将本次提出的方案和上述两种算法做出对比。

3.2 实验结果与分析

(1) 节点超载机会

对节点而言,超载机会与 DCN 的 QoS 成反比。超载机会越多,则表明该算法无法监测当前多径路由的负载状态,无法合理分配数据流转发路径,这必然导致数据流量的拥塞。图 3 所示为 3 种策略下节点超载机会对比。

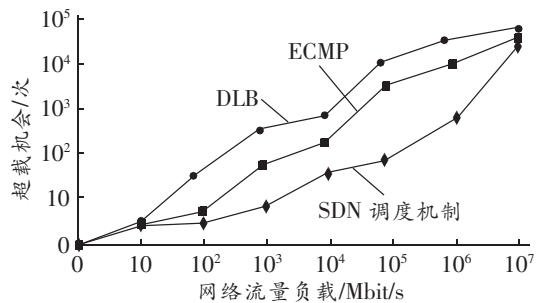


图 3 3 种策略下节点超载机会对比

从仿真^[8]结果不难看出,在本文提出的流量载

荷调度机制下节点超载的机会最少;ECMP 算法虽然能有效减少节点超载机会,但在网络重载时节点超载现象也随之严重;相比之下,DLB 算法引起的节点超载次数最多。随着网络负载逐步增加,网络流量也逐渐接近系统阈值,3 种算法机制下的节点超载机会最终趋于一致。由此可知,本文提出的机制仅在系统性能阈值范围内才表现出优越性。

(2) 平均时延

图 4 所示为 3 种策略下平均传输时延随网络流量负载变化的趋势图。当网络流量负载低于 10^3 Mbit/s,即网络轻载时,3 种算法的平均传输时延^[9]均缓慢增加,且 3 种时延相差无几。在网络轻载时,DLB 算法选出最短路由作为转发路由的过程实现起来最简单便捷,需要的时间较短;而本文提出的 SDN 架构下多路由数据流量载荷调度机制要监测数据流量并定期向控制器发送网络即时参数,同时,由于一开始交换机内部没有流表,当数据流到达时要通知 Floodlight 控制器计算该数据流的转发路径并下发流表,上述两个繁琐的过程导致了 SDN 架构下调度机制表现出的时延稍劣于 DLB。当网络载荷流量超过 10^3 Mbit/s 时,3 种策略下的平均传输时延随流量负载增加总体呈现抖升趋势。这是因为随着网络重载加大,拥塞现象开始出现并愈发严重。此时 DLB 算法却仍然秉承选择最短路由路径作为转发链路的思想,而 ECMP 算法和本文提出的调度机制都能实现多路径载荷分担,时延指标相对较好。同时,参照图 1 不难看出,相对 ECMP 而言,本文的调度机制建立在 SDN 架构下,因具有拥塞状态评估功能,故时延性能优于 ECMP 算法。

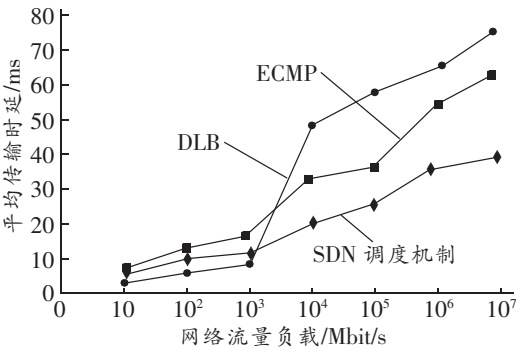


图 4 3 种策略下数据流传输时延对比

(3) 平均带宽利用率

图 5 所示为 3 种策略下平均带宽利用率随数据流量载荷变化的对比情况。当流量载荷低于 10^3 Mbit/s 时,由于网络剩余带宽较多,3 种算法机制下的平均带宽利用率接近且呈现上升趋势;当流

量负载超过 10^3 Mbits/s,即网络处于重载时,DLB 算法的平均带宽利用率最低。这是因为无论网络实时状态发生怎样的变化,选择最短路由不变始终是 DLB 一贯的策略。而 ECMP 算法和本文提出的调度机制都能实现载荷分担,尤其是本文所提的调度机制具有拥塞控制功能,可在拥塞出现时即刻展开重路由来疏导数据流,因此相对 ECMP 算法的带宽利用率有所提升。当网络负载超过 10^5 Mbits/s 时,3 种策略的平均带宽利用率都稍有降低并趋于稳定。由于 ECMP 和 DLB 两种算法的局限性造成一定程度上的网络拥塞,降低了平均带宽利用率。而本文提出的调度机制是通过监视全局参量综合选出最佳路由,对 DCN 全网而言具有广泛的适应性,因此平均带宽利用率指标最好。

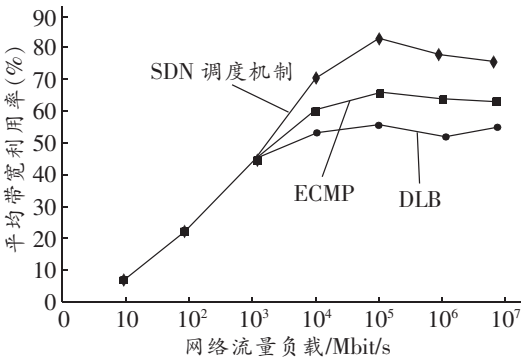


图 5 3 种策略下平均带宽利用率对比

4 结束语

DCN 在传统的多径路由机制下存在拥塞和带宽利用率低等问题。本文充分利用 SDN 具有全局视角的优势提出一种基于 SDN 的胖树网络数据流量均衡调度算法。该算法下的负载均衡分配过程是通过 Ryu 控制器获取全网状态信息,并通过求解交换节点转发阈值上限确定交换节点可调用的带宽以及求解链路最大负载确定链路闲置带宽,再确定最优转发路径进而达到链路负载均衡目的。该分配算法经由 Mininet 和 Floodlight^[10] 仿真平台验证,结果表明,相对于 ECMP 和 DLB 两种算法,该算法能有效降低节点超载机会和传播时延,提升 DCN 平均带宽利用率,并能提供良好的负载均衡保障。

参考文献:

[1] 林闯,罗军舟,吴建平,等.新一代网络架构及技术专辑前言[J].计算机学报,2012,35(6):1075—1076.