

SDN 中一种多控制器负载均衡方案^①

马 勤¹, 李志芳²

1. 郑州澍青医学高等专科学校 卫生管理系, 郑州 450064;

2. 河南理工大学 工商管理学院, 河南 焦作 454000

摘要: 针对软件定义网络(software defined network, SDN)中控制器子域分区不合理导致的控制器负载分配不平衡、网络通信性能下降的问题, 本文提出一种基于负载均衡的多控制器部署(multi-controller load balancing, MLB)方案. 在初始静态网络中, 提出一种基于域内和域间通信成本的负载平衡控制器部署模型, 并将流量请求转换为排队模型. MLB将近邻传播(affinity propagation, AP)算法中的偏置参数和阻尼系数这两个参数作为粒子, 通过粒子群优化(particle swarm optimization, PSO)算法对其进行智能调整, 解决集群性能受偏置参数和阻尼系数初始值影响的问题, 得到合理的网络规划. 实验结果表明: 与亲缘传播算法和遗传算法(genetic algorithm, GA)相比, 该方案可以提供更稳定、准确和负载均衡的多控制器部署.

关 键 词: 软件定义网络; 亲缘算法; 排队模型; 负载均衡

中图分类号: TP393

文献标志码: A

文章编号: 1000-5471(2021)07-0114-06

软件定义网络(SDN)是一种新颖的网络体系结构, 它将控制平面和数据转发平面分开^[1], 通过应用程序接口和开放程序接口控制网络流量, 为研究新的网络应用程序和未来的 Internet 技术提供一种新的解决方案^[2].

随着 Internet 流量和规模的爆炸性增长, 尤其是当网络设备广泛分布在各个区域时, 单个控制器可以支持的网络规模受到限制^[3]. 为了避免由于单个控制器负担过重而导致网络性能低下和单点故障问题, 通常在网络中部署多个控制器来实现分布式控制管理^[4]. 多控制器将控制平面分为几个子域, 每个控制器只需要管理自己的域中交换机, 因此可以减轻控制平面在可靠性、可扩展性和多功能方面的不足. Zhao 等^[5]提出一种基于亲缘传播的改进聚类算法, 与传统的聚类算法相比, 该算法可以获得更好的网络性能. Liu 等^[6]提出控制器负载均衡算法, 考虑了交换机与控制器之间的平均延时和最差延时, 并通过优化模型解决了控制器和交换机之间的部署问题. Zhang 等^[7]提出一种基于 K-Critical 算法的最小控制器数量和部署计划方法. 张栋等^[8]引入一种 k-means 算法来部署软件定义网络(SDN)控制器, 该算法开始时仅在一个区域上运行, 然后通过增加分区数进行迭代. Wang 等^[9]引入粒子群优化(PSO)来解决 SDN 的控制器部署问题, 优化目标是使控制器和交换机之间的传播延迟最小, 还考虑了控制器的容量限制. Singh 等^[10]提出一种网络聚类 PSO 算法, 该算法考虑了控制器的负载和传播延迟. Liao 等^[11]研究了 SDN 的弹性, 并将控制器和交换机之间连接的弹性作为性能评估指标. 在此基础上, 定义了网络弹性优化的控制器部署问题, 并

① 收稿日期: 2020-06-15

基金项目: 河南教育厅教育科学“十三五”规划课题(2018-JkGHYB0069).

作者简介: 马 勤, 讲师, 主要从事计算机应用研究.

提出了相应的启发式算法来寻找控制器的最优位置. 为了提高 SDN 的生存能力, killi 等^[12]提出一种控制器放置策略, 该策略考虑了网络设计时的路径多样性, 控制器容量以及故障转移机制. Li 等^[13]介绍了 SDN 可靠性的度量, 并给出了基于紧密中心的控制器部署.

当前在控制器部署中考虑的主要因素是延迟和稳定性, 控制器容量和控制器之间的负载均衡也是非常重要的因素. 本文提出一种多控制器负载均衡算法 (MLB). 该算法基于域内和域间通信成本建立负载均衡控制器部署模型, 并将流量请求转换为排队模型. 初始的静态网络中可以预先设置控制器部署问题, 将近邻传播 (AP) 算法中的偏置参数和阻尼系数作为粒子, 通过 PSO 算法对其进行智能调整. 该算法提高了聚类效果和收敛精度, 实现了平衡的控制器部署.

1 研究背景

1.1 网络模型

基于图论建立了多控制器 SDN 模型. 带内通信通常被用作 SDN 中的主要通信模式. 在本文中, 控制器网络拓扑由无向图 $G = (V, E)$ 表示, 其中 V 和 E 分别是节点和链接的集合. M 代表网络中控制器的数量, C 是控制器集, $C = \{C_1, \dots, C_M\}$. N 代表交换机的数量, S 是交换机组, $S = \{S_1, \dots, S_N\}$, 故 $|V| = M + N$. λ_i^t 是时隙间隔 t 中第 i^{th} 个交换机的请求速率, d_{ij} 表示第 i^{th} 个交换机和第 j^{th} 个控制器之间的最短距离. x_{ij}^t 是一个二进制变量, 并且 $x_{ij}^t = 1$ 表示交换机 i^{th} 在时隙 t 中已成功连接到 j^{th} 控制器. 交换机和控制器的部署可以定义为 $N \times M$ 的二进制矩阵 Ω . 为了满足动态约束, 交换机只能选择一个控制器作为主控制器. 表示为 $x_{ij}^t = \begin{cases} 1, & \text{第 } i \text{ 个开关连接到第 } j \text{ 个控制器} \\ 0, & \text{其他} \end{cases}$. 控制器的处理能力为 A , 取决于 CPU、带宽和内存等因素, $A = \{A_1, \dots, A_M\}$. L_M 是每个控制器的冗余系数, 范围从 0 到 1, 这表示控制器具有足够的能力执行状态同步. 流要求 j^{th} 控制器需要在时间 t 内处理与之相连的所有交换机的流量请求总和, 表示为 $\Phi_j^t = \sum_{i=1}^N \lambda_i^t x_{ij}^t$, 其中每个交换机的 λ_i^t 不同, 表示交换机 i^{th} 在时隙 t 的流请求率. 在 OpenFlow 网络中, 每个接入交换机都会生成流建立请求. 因此, 在控制器中流请求的服务时间内, 将有多多个源自交换机到控制器的“数据包输入”请求. 交换机和控制器的流量请求在控制器入口形成队列. 根据 OpenFlow 交换机向控制器发送请求的特征, 可以合理地假设新流建立请求的到达瞬间构成一个速率为 Φ_j^t 的泊松过程, 并且流请求的处理时间独立, 具有负指数分布的均值为 $\frac{1}{A_j}$ 的相同随机变量. 根据排队论, 流量的传输和处理描述为一个 $M/M/1$ 队列, 流请求以排队形式聚集在所连接的交换机上. 借助利特尔定律, 可以获得 j^{th} 控制器的平均逗留时间 $\omega_j^t = \frac{1}{A_j - \Phi_j^t}$. 假设计算单个源路由的时间取决于网络规模, 则 j^{th} 控制器的平均响应时间可以计算为 $\Delta t_j = |V|^2 \cdot \omega_j^t$, 其中 $|V|$ 代表 SDN 的节点数, 表明处理时间受网络大小的影响. 因此,

$$\text{在时隙 } t \text{ 中交换机和控制器之间, 控制器的平均响应时间为 } D^t = \frac{\sum_{j=1}^M \Phi_j^t \cdot \Delta t_j}{\sum_{j=1}^M \Phi_j^t}.$$

(1) 域内通信成本

当需要按需安装流表时, 交换机需要将数据包发送到控制器, 控制器计算转发路径并将相应的流标签安装到转发路径的交换机. 然后, 交换机根据流表转发数据包. 在此过程中, 对于控制器 C_j , 总计流请求路径在时域中的延迟主要包括以下内容: 交换机将数据包发送到从属控制器, 控制器计算转发路径并安装交换机流表. OpenFlow 网络中交换机和控制器的通信成本可表示为

$$D_{req} = 2v_r \sum_{j \in M} \sum_{i \in N} \left(\left\lceil \frac{\lambda_i^t}{v} \right\rceil d_{ij} x_{ij}^t \right) \quad (1)$$

式(1)中 v_r 是轮询交换机的平均速率, 它与连接到交换机链路的平均数量有关, v 是单位速率, $v = 1 \text{ kb/s}$.

(2) 域间通信成本

在多控制器环境中, 需要控制器之间信息同步, 以便每个控制器可以维护全局网络状态信息. 状态同步成本主要指控制器之间交互状态信息所引起的通信成本, 如式(2)所示.

$$D_{syn} = v_s \sum_{j \in M} \sum_{k \in M} d_{jk} \quad (2)$$

式(2)中 v_s 是控制器状态信息的平均传输速率. 但是 v_s 小于 v_r , 因为 SDN 控制器不会同步所有的信息子域.

基于以上定义和分析, 网络总的通信成本表示交换机与控制器之间以及控制器在特定时间的通信成本. 对于给定的网络拓扑, 域内通信成本随着控制器数量的增加而降低, 而域间通信成本则增加. 总体域内和域间通信成本为

$$\min Total = \gamma D_{req} + (1 - \gamma) D_{syn} \quad (3)$$

$$s. t. \sum_{j \in M} x_{ij}^t = 1, \forall i \in N \quad (4)$$

$$\Phi(t)_j \leq L_j A_j, \forall j \in M \quad (5)$$

$$D(t) \leq \delta \quad (6)$$

$$x_{ij}^t \in \{0, 1\}, \forall i \in N, \forall j \in M \quad (7)$$

式(3)中 $\gamma \in [0, 1]$ 是用于控制交易的加权因子, D_{req} 表示域内通信成本, D_{syn} 表示域间通信成本. 式(4)确保交换机只能选择一个控制器作为主控制器. 式(5)确保控制器的负载在正常范围内, 并且不超过控制器的处理能力. 式(6)表示从交换机到控制器的平均延迟小于 δ (常数). 式(7)确保每个交换机都可以连接到主控制器. 式(4) — 式(7)中 N 表示交换机数量, M 表示控制器数量.

1.2 问题提出

整个网络包括 14 个交换机(S1—S14)和 4 个控制器(C1—C4). 控制器数量和部署位置的差异会导致控制器之间的状态不平衡, 并增加分布式网络中的部署成本. 因此, 对给定的 SDN, 本文要解决的问题是获取合理数量的控制器以及交换机与控制器的映射关系.

多控制器部署的通信成本包括域内通信和域间通信成本. 控制器的数量越少, 域间通信成本就越小, 域内通信成本就越大. 为了确保网络性能, 交换机和控制器的传播延迟不应超过可容忍的阈值. 同时, 由于控制器的处理能力和带宽的限制, 单个控制器可以处理的切换请求是受限的. 在分配交换机时, 应考虑控制器处理请求的能力. 因此, SDN 中受流量传播延迟和控制器容量限制的多控制器部署可以描述如下: 给定 SDN, 其交换机和链路的拓扑已知, 最终目标是将网络划分为合理的控制域, 并以一种最小化整个域内和域间通信成本的方式表示每个域的控制域, 这时交换机到控制器的平均延迟不超过可容忍的阈值, 并且由单个控制器来管理交换机请求不超过控制器的处理能力. 此外, 应尽可能平衡每个控制器的部署.

2 多控制器负载均衡方案

传统的聚类算法, 如 k-means 算法严重取决于人工干预、影响目标函数的不同初始分区或 k 值. 因此, 本节中建议使用 MLB 来解决控制器放置问题, 无需初始化网络中控制器的数量.

AP 算法是一种非常有效的基于分区的聚类算法, 该算法将所有数据点视为潜在的聚类中心, 采用成

对的数据点之间的相似性作为输入数据, 并通过在迭代过程中数据点之间传递信息来获得最佳解. 与 k-means、模糊 C 均值和其他聚类算法相比, AP 算法具有多个优点, 包括效率、对初始化的不敏感性以及用更少的错误来确定范例的能力. 本文提出一种改进的 AP 算法, 即多控制器负载均衡 MLB 算法来解决 SDN 中的控制器部署问题. 具体而言, 在两个交换机的相似度测量中, 采用通信成本 $\lambda(t)_i d(i, j) x_{ij}$ 代替欧几里得距离. 通过将算法中的偏置参数和阻尼系数视为粒子, 使用 PSO 算法对其进行智能调整, 提高了聚类效果. MLB 可以自适应学习最佳控制器的数量, 也可以优化交换机和控制器之间的部署关系, 从而确保控制器在网络中的合理分配.

为了最小化式(3)中表示的目标函数, 将两个节点之间的相似度 $S(i, j)$ 定义为它们之间的通信, 采用通信成本 $\lambda(t)_i d(i, j) x_{ij}$, 而不是标准 AP 中通常使用的欧几里得距离

$$S(i, j) = \begin{cases} \lambda(t)_i d(i, j) x_{ij}, & i \neq j \\ P, & i = j \end{cases} \quad (8)$$

当 $i \neq j$ 时, $S(i, j)$ 表示将采样点 j 作为采样点 i 的代表点的可能性. 对角元素 $S(i, j)$ 是偏置参数 $P(j)$. $P(j)$ 的初始值通常取相同的值, 这是相似性矩阵中所有非对角元素的最小值或平均值. 它的初始大小对最终聚类数有很大的影响. P 值越大, 生成的簇越多, 反之亦然.

与 k-means 不同, AP 不需要事先知道聚类中心的数量. 责任值 $R(i, j)$ 和可用性值 $A(i, j)$ 是 AP 算法中的两个重要信息. 标准值 $C(i, j) = R(i, j) + A(i, j)$, 表示候选节点 j 作为聚类中心的可能性. 因此, AP 聚类在 4 个矩阵上进行操作: 相似性矩阵 S , 责任矩阵 R , 可用性矩阵 A 和标准矩阵 C . 考虑到偏置因素和阻尼因子对聚类结果的影响, 本文提出一种基于 PSO 的改进型 AP 算法 MLB. MLB 在 PSO 算法中将偏置因素和阻尼因子作为粒子的位置坐标. 初始化每个粒子的坐标和速度, 以选择不同的首选项因素和阻尼因子. 然后根据式(9)、式(10), 粒子的位置和方向会不断更新, 在更新过程中将粒子的位置用作 AP 算法的偏置因素和阻尼因子的值进行聚类.

$$V_{id} = \omega V_{id} + \eta_1 rand() (P_{id} - X_{id}) + \eta_2 rand() (P_{gd} - X_{id}) \quad (9)$$

$$X_{id} = X_{nd} + V_{id} \quad (10)$$

式(9)中 V_{id} 表示第 i 个粒子在 d 维上的速度, P_{id} 是粒子经历的最佳位置, P_{gd} 是该组的最佳位置, ω 是惯性权重, η_1 和 η_2 是调节 P_{id} 和 P_{gd} 的重要参数. 式(9)中, 将当前粒子位置与单个最优解和群体最优解进行比较, 获得群体最优和个体最优的发展趋势, 然后根据该趋势和原始初始值确定新的速度方向. 式(10)中粒子的新位置在距向上运动一定距离处产生. X_{nd} 奇数维的值是一个新的偏置元素, 偶数维的值是阻尼因子.

$$P = X_{nd}, d \text{ 为奇函数} \quad (11)$$

$$\lambda = X_{nd}, d \text{ 为偶函数} \quad (12)$$

当迭代次数超过最大值或集群中心连续数次未更改时, 迭代终止.

3 实验结果与分析

3.1 模拟环境设置

为了使实验更具代表性, 选择 OS3E 拓扑模型来模拟实验. OS3E 拓扑模型由 34 个节点和 42 个链接组成. 网络中所有节点都具有部署控制器和交换器的能力, 并且它们彼此独立. 算法用 JAVA 实现, 并使用 MATLAB 分析结果. 交换机的请求速率服从泊松分布, 范围为 150~550 kb/s. 处理能力 A_M 设置为 15 M. 为了在不同控制器之间产生差异, 可以在 0.9~1 之间任意选择控制器 L_M 的冗余因子. 对交换机进行轮询的平均速率为 $v_r = 10$ kb/s, 并且控制器的状态同步信息的传输速率为 $v_s = 1$ kb/s.

3.2 实验结果与分析

为了验证 MLB 的性能, 建立了一系列仿真实验, 并将 MLB 与 AP 和遗传算法 GA 进行了比较. 仿真

结果表明,本文算法可以提高更加稳定、准确的控制器部署.为了消除实验的随机误差,3种算法在相同的实验条件下均运行了20次,取平均值作为实验结果.

在 OS3E 网络中,交换机的数量是一个常数.将3种算法的 T_M 参数设置为300次迭代.目标函数的值随着控制器数量增加而减少,这仅仅是因为随着控制器数量增加,由控制器管理的交换机数量会减少,因此切换到控制器的等待时间也减少.此外,通过 OS3E 拓扑更好地处理流量请求所需的控制器数量均为5个,该值确保了这3种算法目标函数的最佳值,而 MLB 获得了目标函数的最低值.

在相同的控制器部署条件下,GA 中每个控制器管理的交换机数量差异最大,其中控制器 C1 管理的交换机数量是控制器 C2 和 C4 的两倍,这是因为遗传算法采用交叉和变异策略使整个集群不断演化并寻找最优解,但是该算法容易陷入最小值.当节点之间的距离较大时,集群操作很容易陷入局部最优状态,从而导致交换机的区域聚集和较差的子域规划效果.尽管 AP 算法不需要设置控制器的数量,但是该算法还会根据交换机之间的距离聚合节点,子域规划的效果不佳.对实验数据进行归一化后,MLB 的节点均衡率为0.87,这明显优于 GA(0.66)和 AP(0.74).

控制器部署的结果不仅影响网络区域的划分,而且还严重影响控制器的处理延迟和控制流量.

AP 算法从控制器的角度优化了子域中的控制流量,但是它不能解决处理延迟差异很大的问题.而 MLB 算法将流量请求转化为排队模型,实现了控制流量的均衡分配.

4 结 语

本文讨论了分布式 SDN 中多控制器的放置问题.以域内和域间通信成本为优化目标,提出了 MLB 来解决控制器在初始静态下的放置问题.实验结果表明,该方案可以根据网络拓扑自适应获得最优的控制器数量,为最合适的控制器分配交换机,并在分布式 SDN 中实现多控制器架构的均衡部署.未来针对 SDN 动态流量网络,如何确保控制器的负载均衡将是研究的重点问题.

参考文献:

- [1] ZHANG S J, LAN J L, SUN P H, et al. Online Load Balancing for Distributed Control Plane in Software-Defined Data Center Network [J]. IEEE Access, 2018, 6: 18184-18191.
- [2] ABDOU A, VAN OORSCHOT P C, WAN T. Comparative Analysis of Control Plane Security of SDN and Conventional Networks [J]. IEEE Communications Surveys & Tutorials, 2018, 20(4): 3542-3559.
- [3] SULTANA N, CHILAMKURTI N, PENG W, et al. Survey on SDN Based Network Intrusion Detection System Using Machine Learning Approaches [J]. Peer-to-Peer Networking and Applications, 2019, 12(2): 493-501.
- [4] 樊自甫,张丹,李书.基于软件定义网络的数据中心网络负载均衡算法研究[J].计算机工程与科学,2018,40(6): 1017-1022.
- [5] ZHAO J L, QU H, ZHAO J H, et al. Towards Controller Placement Problem for Software-Defined Network Using Affinity Propagation [J]. Electronics Letters, 2017, 53(14): 928-929.
- [6] LIU J J, SHI Y P, ZHAO L, et al. Joint Placement of Controllers and Gateways in SDN-Enabled 5G-Satellite Integrated Network [J]. IEEE Journal on Selected Areas in Communications, 2018, 36(2): 221-232.
- [7] ZHANG T Z, GIACCONE P, BIANCO A, et al. The Role of the Inter-Controller Consensus in the Placement of Distributed SDN Controllers [J]. Computer Communications, 2017, 113(15): 1-13.
- [8] 张 栋,郭俊杰,吴春明.层次型多中心的 SDN 控制器部署[J].电子学报,2017,45(3): 680-686.
- [9] WANG D S, TAN D P, LIU L. Particle Swarm Optimization Algorithm: an Overview [J]. Soft Computing, 2018, 22(2): 387-408.
- [10] SINGH S P, SHARMA S C. A Particle Swarm Optimization Approach for Energy Efficient Clustering in Wireless Sensor Networks [J]. International Journal of Intelligent Systems and Applications, 2017, 11(6): 66-74.

[11] LIAO Y Z, TSAI S C. Fast Failover with Hierarchical Disjoint Paths in SDN [C]//2018 IEEE Global Communications Conference (GLOBECOM). Abu Dhabi: IEEE, 2018.

[12] KILLI B P R, RAO S V. On Placement of Hypervisors and Controllers in Virtualized Software Defined Network [J]. IEEE Transactions on Network and Service Management, 2018, 15(2): 840-853.

[13] LI F X, XU X B, YAO H P, et al. Multi-Controller Resource Management for Software-Defined Wireless Networks [J]. IEEE Communications Letters, 2019, 23(3): 506-509.

A Multi-Controller Load Balancing Scheme in SDN

MA Qin, LI Zhi-fang

1. Department of Health Administration, Zhengzhou Shuqing Medical College, Zhengzhou 450064, China;
2. School of Business Administration, Henan Polytechnic University, Jiaozuo Henan 454000, China

Abstract: Aiming the problem of unbalanced controller loads distribution and network communication performance degradation caused by unreasonable subdomain partitioning of controllers in software defined network (SDN), a multi-controller load balancing scheme is proposed. In the initial static network, a load balancing controller deployment model based on the intra-domain and inter-domain communication cost has been proposed, and the traffic requests are transformed into a queuing model. MLB takes the two parameters that bias parameters and convergence coefficients in affinity propagation algorithm as particles, then adjusts it intelligently by particle swarm optimization(PSO) algorithm, to solve the problem of clustering performance being affected by the initial values of the bias parameters and convergence coefficients, getting the reasonable network planning. Experimental results show that compared with affinity propagation algorithm and genetic algorithm, this scheme can provide more stable, accurate and load balancing multi-controller deployment.

Key words: software defined network(SDN); affinity propagation algorithm; queuing model; load balancing

责任编辑 夏 娟