

SDN 多控制器放置问题研究综述



贾吾财 吕光宏 王桂芝 宋元隆

四川大学计算机学院 成都 610065

(910091702@qq.com)

摘要 随着软件定义网络(Software Defined Network,SDN)的迅猛发展,单控制器部署的固有缺陷逐渐显露出来,多控制器部署已成为必然趋势。但由于控制器数量以及放置位置对网络性能具有决定性的影响,且在解决该问题时权衡因素多、计算复杂度高,严重阻碍了 SDN 在数据中心和广域网的应用。首先阐述了放置问题的本质和通用的求解步骤;其次基于网络模型详述了部署策略的核心构件,即优化目标、搜索算法;然后综合国内外的研究,将部署策略分为静态部署和动态部署两大类,并着重对比了典型策略的优缺点;最后展望未来的研究方向。

关键词: 软件定义网络;多控制器;放置问题;优化目标;搜索算法

中图分类号 TP393

Review on Placement of Multiple Controllers in SDN

JIA Wu-cai, LV Guang-hong, WANG Gui-zhi and SONG Yuan-long

School of Computer, Sichuan University, Chengdu 610065, China

Abstract With the rapid development of software defined network, deployment of the inherent defects of single controller gradually revealed, multiple controller deployment has become an inevitable trend. However, the number and location of controllers have a decisive influence on network performance, and the high complexity of weighing factors in solving this problem, which seriously hinders the application of SDN in data centers and wide area networks. Firstly, the essence of placement problem and its general solving steps are described. Secondly, based on the network model, the core components of the deployment strategy, namely the optimization objectives and search algorithm are described in detail. Then, based on the research at home and abroad, the deployment strategies are divided into static deployment and dynamic deployment, and the advantages and disadvantages of typical strategies are compared. Finally, the future research direction is prospected.

Keywords Software definition network, Multi-controller, Placement problem, Optimizing objective, Search algorithm

1 引言

传统网络由于自身架构僵化^[1-2]等问题,很难满足庞大用户群体日益多样化的需求。面对这种困境,一种新型的网络架构——软件定义网络应运而生。

SDN 是美国斯坦福大学 Clean Slate 研究组提出的一种基于集中控制的新型网络架构^[3]。它的核心理念是将网络设备的控制平面与数据平面分离,利用控制平面进行集中控制,而数据平面只负责转发分组。典型的 SDN 体系结构如图 1 所示。其中,数据平面主要由交换机等硬件设备组成,主要任务是根据控制器的指令进行数据转发,并将当前的链路状态、统计信息等通过南向接口向上进行传递。控制平面是 SDN 的核心部分,集中管理网络中所有的数据转发设备,它不仅能处理从应用平面下发的任务请求和从数据平面传输过来的链路信息,还能下发指令给数据平面。应用平面主要为用户提供服务,用户可以借助北向接口获取全局网络视图,实现对网络资源的优化控制。

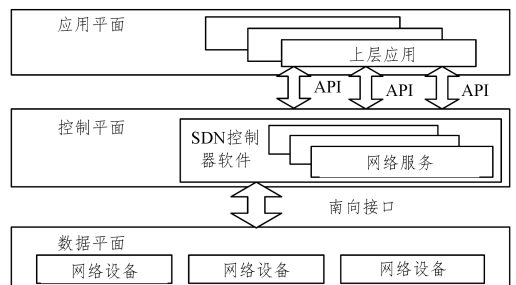


图 1 SDN 的基本架构图

Fig. 1 Basic architecture of SDN

SDN 设计之初是利用单个控制器管理网络中的所有设备^[4]。但随着网络规模的不断扩大,单控制器固有的缺陷日益凸显^[5],如单点故障、有限的控制器处理容量等,因此出现了逻辑集中、物理分布的多控制器体系架构,如 HyperFlow^[6], Kandoo^[7], Orion^[8-9]等,但也引入了多控制器部署问题。如图 2 所示,虚线代表交换机与控制器之间可能存在的映射关系,但不一定是最佳方案。

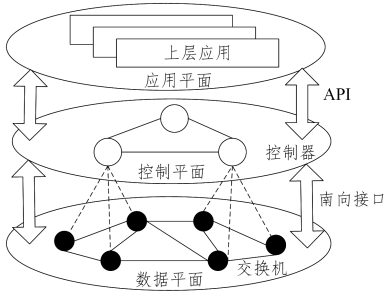


图 2 控制器部署的简化架构

Fig. 2 Simplified architecture for multi-controller deployment

目前虽然已有大量的多控制器部署策略被提出,但通过查阅国内外的文献可知,目前该方向的相关综述文献^[10-12]较少且存在着不足,如缺少对多控制器放置问题实质和通用解决思路的清晰归纳,忽略了对新优化目标的整理,缺乏对常用搜索算法的对比分析,同时欠缺对动态和静态策略具体实现差别的归纳以及各种策略优缺点的深入分析等。针对以上不足,本文将归纳出问题的实质和通用的解决思路,再详述更多的性能指标及搜索算法。最后,再对典型的放置策略进行分析对比,以为日后的研究提供有价值的参考。

2 多控制器放置问题

多控制器放置问题(Controllor Placement Problem, CPP)是一个 NP 难问题^[13-16]。其本质就是在部署多控制器的 SDN 网络中,确定控制器的最佳数目和最优放置位置,并配置控制器与交换机的映射,从而达到预设的优化目标。CPP 问题的通用求解思路是根据实际需求,先确定优化目标并构成目标函数,然后划分网络以缩小搜索范围,最后利用搜索算法寻找可行解。

3 部署策略的核心构件

3.1 建模和符号设定

为了便于对 CPP 进行研究,通常会对整个网络进行建模,符号设定如表 1 所列。

表 1 符号表

Table 1 Symbolic table

符号	定义
V	网络中节点的集合
M	控制器的个数
N	交换机的个数
C	控制器集合
S	交换机集合
d_{ij}	节点 v_i 与节点 v_j 之间的最短距离
x_{ij}	二进制数,值为 1 时表示节点 i 成功连接到节点 j
n_j	控制器 c_j 所管理交换机的个数
λ_i	交换机 s_i 的流请求速率
v_s	轮询交换机的平均流量
v_c	域间控制器信息交互的平均流量
m_p	Packet-in 数据包的大小
d'_{jk}	控制器 c_j 与控制器 c_k 的最短距离
n	网络中所有节点的个数
w_{ij}	相邻节点 v_i 与 v_j 之间的链路带宽
ρ_l	链路 l 失效的概率
m_{ij}	交换机 s_i 到控制器 C_j 之间不相交路径的数目

3.2 常见的优化目标

常见的优化目标主要有 4 个:最小化通信代价、最大化可

靠性、最小化时延以及负载均衡。

3.2.1 最小化通信代价

SDN 中多控制器的通信代价主要由以下 4 部分构成。

(1)控制器处理 Packet-in 请求的代价

控制器处理 Packet-in 请求的代价 P_{PC} 是控制器管理的所有交换机产生的流请求速率之和,如式(1)所示:

$$P_{PC} = \sum_{s_i \in S} \sum_{c_j \in C} \lambda_i * x_{ij} \quad (1)$$

(2)交换机与控制器的通信代价

P_{CS} 代表控制器与交换机的通信代价,如式(2)所示:

$$P_{CS} = \sum_{s_i \in S} \sum_{c_j \in C} d_{ij} * v_s * x_{ij} \quad (2)$$

(3)域间控制器的通信代价

P_{CC} 代表控制器之间进行跨域通信时产生的通信开销,如式(3)所示。由于控制器只会同步子域的一部分信息,因此一定 v_c 小于 v_s 。

$$P_{CC} = \sum_{c_k \in C} \sum_{c_j \in C} d'_{jk} * v_c \quad (3)$$

(4)安装流表的代价

P_t 代表当一个新流到达交换机时,控制器安装新流表项到相应交换机中所产生的代价,如式(4)所示:

$$P_t = \sum_{s_i \in S} \sum_{c_j \in C} m_p * d_{ij} * x_{ij} \quad (4)$$

3.2.2 最大化可靠性

最大化可靠性可看作由最小化控制链路故障、避免节点孤立、最小化控制链路拥塞 3 部分构成。

(1)最小化控制链路故障

1)最大化多条控制路径

多条控制路径是指交换机与控制器至少有两条不相交的路径。其中, $P(i, j)$ 为交换机 s_i 与控制器 c_j 间一条路径正常的概率, P_a 代表交换机与控制器之间有多条控制路径且都正常的概率的最大值。

$$P(i, j) = \prod_{l \in d_{ij}} (1 - p_l) \quad (5)$$

$$P_a = \max_{c_j \in C} \sum_{s_i \in S} (1 - \prod_{P(i, j) \in m_{ij}} (1 - p(i, j))) \quad (6)$$

2)最大化连接多个控制器

P_b 是指一个交换机成功连接到多个控制器上的概率的最大值,如式(7)所示:

$$P_b = \max_{s_i \in S} (\sum_{c_j \in C} (1 - \prod_{c_j \in C} (1 - P(i, j)))) \quad (7)$$

3)最小化控制路径单元数

P_c 是指最小化控制路径包含的物理单元且该路径正常工作的概率的最大值,如式(8)所示:

$$P_c = \max(\sum_{s_i \in S} \sum_{c_j \in C} P(i, j)) \quad (8)$$

(2)避免节点孤立

1)节点相近度

节点相近度 P_i 是指节点 v_i 与其他节点的平均接近程度,如式(9)所示:

$$P_i = \sum_{j=1, j \neq i}^n \frac{1}{(n-1) * d_{ij} * x_{ij}} \quad (9)$$

2)节点可靠度

节点可靠度 $P_r(i)$ 是对网络中节点 v_i 可靠性的评定,如式(10)所示。当 $P_r(i)$ 值越大时,该节点就越可靠,发生单点故障的概率就越小。

$$P_r(i) = \left(\frac{\sum_{j=1}^n x_{ij}}{\sum_{i=1, j=1, i \neq j}^n x_{ij}} \right) * P_i * \lambda_i \quad (10)$$

(3) 最小化控制链路拥塞

$P_{\text{congest}}(e_{ij})$ 代表直连链路 e_{ij} 发生拥塞的概率, 如式(11)所示。当流量 λ_i 从源节点 v_i 传输到目的节点 v_j 时, $p_{ij} = 1$, 否则 $p_{ij} = 0$ 。 $P_{\text{congest}}(e_{ij})$ 的取值范围为 $[0, 1]$ 。

$$P_{\text{congest}}(e_{ij}) = \begin{cases} \frac{p_{ij} * \lambda_i + p_{ji} * \lambda_j}{x_{ij} * \omega_{ij}}, & x_{ij} = 1 \\ 1, & x_{ij} = 0 \end{cases} \quad (11)$$

3.2.3 最小化时延

网络时延由传播时延、排队时延、处理时延和发送时延组成^[17-18]。当网络没有阻塞时, 排队延时可以忽略不计。而发送时延和处理时延通常设为常数, 因此, 在求解 CPP 时, 大多数只考虑传播时延。传播时延通常被分成以下两类。

(1) 交换机与控制器的时延

1) 交换机与控制器的平均传输时延

交换机与控制器的平均传输时延 T_{avg} 代表交换机与控制器的平均时延, 如式(12)所示:

$$T_{\text{avg}} = \frac{1}{N} * \sum_{s_i \in S} \sum_{c_j \in C} d_{ij} * x_{ij} \quad (12)$$

2) 交换机与控制器的最坏传输时延

交换机与控制器的最坏传输时延 T_{wc} 即是交换机到控制器最小时延的最大值, 如式(13)所示:

$$T_{\text{wc}} = \max_{s_i \in S} \left(\sum_{c_j \in C} d_{ij} * x_{ij} \right) \quad (13)$$

(2) 控制器间的传输时延

1) 控制器间的平均传输时延

控制器间的平均传输时延 T_{ca} 代表任意一对控制器之间的平均传播时延, 如式(14)所示:

$$T_{\text{ca}} = \frac{1}{M(M-1)} * \sum_{c_j, c_k \in C} d_{jk}^c \quad (14)$$

2) 控制器间的最坏传输时延

控制器间的平均传输时延 T_{cw} 代表任意一对控制器之间的最坏传播时延, 如式(15)所示:

$$T_{\text{cw}} = \max_{c_j, c_k \in C} d_{jk}^c \quad (15)$$

3.2.4 负载均衡

负载均衡是重要的优化目标, 通常分两种情况进行讨论。

(1) 交换机产生的负载都相同

理想状态下, 每个交换机的负载都相同, 控制器之间的负载差异只由各自管理交换机数量的差异决定。此时, 负载均衡的相关指标如下。

1) 控制器间的负载差异最大值

控制器间的负载差异最大值 B_w 为不同控制器管理交换机数目的最大差值, 如式(16)所示:

$$B_w = \max_{c_j \in C} n_j - \min_{c_k \in C} n_k \quad (16)$$

2) 控制器间负载差异的平均值

控制器间负载差异的平均值 B_{avg} 是指所有控制器之间负载差值的平均, 如式(17)所示:

$$B_{\text{avg}} = \frac{1}{M * (M-1)} * \sum_{c_j, c_k \in C, c_j \neq c_k} |n_j - n_k| \quad (17)$$

3) 控制器间的负载差异度

控制器间的负载差异度 B_p 是指所有控制器的负载差异程度, 如式(18)所示:

$$B_p = \sum_j^M \left(\frac{n_j - \frac{N}{M}}{M} \right)^2 \quad (18)$$

(2) 交换机产生的负载不同

控制器负载均衡率 P_L 需要考虑交换机个数和流请求速率两个因素, 如式(19)所示。当 P_L 值越大时, 各个控制器的负载就越接近。

$$P_L = \sum_{j=1}^M \left[\frac{n_j}{N} * \left(\frac{\sum_{m=1}^n \lambda_m}{\sum_{j=1}^n \lambda_j} \right) \right] \quad (19)$$

3.3 常用的搜索算法

常见搜索算法的对比如表 2 所列。

表 2 常见搜索算法归纳表

Table 2 Summary table of common search algorithms

搜索算法	优点	缺点	离线/在线求解	网络规模
聚类算法	高效处理大数据集	对参数设置敏感	离线	无限制
多目标规划算法	可求出近似最优解	运行时间较长	离线	中小型
贪婪算法	易理解且收敛快	获得局部最优解	离线/在线	无限制
启发式算法	收敛速度快	易陷入局部最优	离线/在线	无限制
博弈算法	求解时间相对较短	强依赖于估值算法	离线/在线	无限制
学习自动机算法	具有全局优化能力	收敛时间较长	离线	无限制
监督学习算法	缩短运算时间	难以获取有用数据	离线/在线	无限制

3.3.1 聚类算法

聚类算法是对数据分组的一种算法。在理论上, 同一组内的数据具有相同的属性或特征, 不同组之间则相差较大。在求解 CPP 时, 聚类算法常根据约束条件及优化目标, 将整个网络划分为 M 个子区域, 同一子区域内交换机与控制器的相关性较大, 子区域间交换机与控制器的相关性较小。

3.3.2 多目标规划算法

多目标规划是指对由多个优化目标构成的函数进行规划。通常情况下, 没有一种可行解能使所有的优化目标同时最小化或最大化, 但标准的 Pareto 最优解可以保证在不损害任何优化目标的前提下, 不存在其他可行解能进一步地优化一个或多个目标。因此, 将 Pareto 应用到算法时, 得到的结果被认为是近似最优解。

3.3.3 贪婪算法

贪婪算法对问题求解时, 总是做出当下最优的选择, 进而提高效率并节省时间, 但得到的往往并非最优解。将贪婪算法应用到 CPP 求解时, 它会从 CPP 的某一个初始解出发, 根据优化目标以及约束条件, 逐步逼近给定的目标, 从而求出较优解。

3.3.4 启发式算法

启发式算法是一种基于直观或者经验构造的算法^[19], 缺乏完备的理论体系, 但能在给定的约束条件下求出组合优化

问题的每一个可行解。启发式算法种类繁多,如蚁群算法^[20]、粒子群算法^[21]、布谷鸟算法^[22]等。该算法应用于 CPP 求解时,得到的结果被认为是较优解。

3.3.5 博弈算法

博弈是以一定的约束条件和规则为前提,各参与人根据所掌握的信息选择各自的策略,进而使利益最大化的过程。根据博弈各方的收益总和是否为零,可分为零和博弈与非零和博弈。CPP 在求解时常采用非零和博弈,其能在有效时间内获得可行解,但解的具体效果强依赖于具体算法。

3.3.6 学习自动机算法

学习自动机(Learning Automata, LA)是强化学习的一类算法。它运行在概率空间中,通过不断与未知环境交互来学习最优值^[23],因此非常适合解决随机环境中的各种问题。将

其运用到 CPP 求解时,会为网络拓扑中的每个节点都配置一个 LA 来随机选择是否作为控制器,反复迭代多次,求出最终结果。

3.3.7 监督学习算法

监督学习算法是机器学习算法的一类^[24]。该算法需要一个有标记的训练数据集去建立表示输入、输出之间学习关系的系统模型,使一个新的输入经过模型计算后,可以得到期望的输出。将该算法应用于 CPP 求解时,须先学习现有的解决方案,而后给出控制器放置位置的预测值。

4 多控制器部署策略

为了解决 CPP,近年来已有大量的多控制器部署策略被提出,可大致分为静态部署和动态部署,如表 3 所列。

表 3 多控制器部署策略的对比

Table 3 Multi controller placement strategy comparison

分类	算法分类	核心算法	优点	缺点
静态部署	聚类算法 ^[25-27]	K-center	首次提出多控制器放置问题	没考虑控制器容量
		容量 k-center 算法	考虑控制器容量	忽略了控制器之间的延迟
		密度峰值聚类算法	在最小化平均时延表现突出	没有考虑负载均衡
	多目标规划算法 ^[28-29]	基于 Pareto 的算法	降低故障对网络性能的影响	计算速度慢,适合小规模网络
		改进 Pareto 的算法	运算速度快	控制器的数目需预先给定
	贪婪算法 ^[30]	贪婪算法	证明平均权重的贪婪算法最有效	直接将一组聚类作为输入
	启发算法 ^[31-32]	启发算法	显著减少连接丢失	忽略了控制器的切换延迟
		粒子群算法	链路失效状态下,最小化时延	仅适用大规模网络
	博弈算法 ^[33]	合作博弈	最小化时延上接近于最优	没有考虑控制器间的延迟
	动态部署	学习自动机算法 ^[34]	学习自动机算法	任何节点到最近控制器的时延不超过期望值
监督学习算法 ^[35]		监督学习算法	缩短了启发算法的运算时间	获取有效数据困难
贪婪算法 ^[36]		贪婪算法	运算时间短	算法准确性有待进一步的提高
		启发算法	减少迁移发生次数	可能陷入局部最优解
启发算法 ^[37-38]		改进启发算法	运算速度快	控制器的数目需预先给定
		非零和博弈	节省成本并保证 QoS	忽略交换机与控制器时延
博弈算法 ^[40-41]		讨价还价博弈	最小化总通信开销较优	算法复杂度较高
		监督学习算法 ^[42]	监督学习算法	动态部署,节省运算时间

4.1 静态部署

静态部署是指在部署过程中依据当时网络拓扑的相关信息来确定控制器的最佳部署位置,即控制器一旦选定部署节点后,位置将不再改变。

4.1.1 聚类算法

Heller 等^[25]首次提出了 SDN 网络中的 CPP,并指出在广域网中控制器的最佳位置取决于交换机与控制器的传播时延。但该文献并没有对算法进行具体描述,也没有考虑控制器的容量。因此,Yao 等^[26]在文献^[25]的基础上增加了对控制器容量的考量,提出了一种容量 k-center 算法,有效减少了控制器数量。如对于实验中的拓扑,该算法最多需要 15 个控制器来确保无过载发生,但 k-center 却需要 40 个控制器。Qi 等^[27]提出一种以最小化交换机与控制器的平均时延为优化目标的放置策略,该策略首先用修正密度峰值聚类算法选择 k 个聚类中心,然后根据目标函数来确定交换机与控制器的映射。经验证,该算法与 k-means 相比,平均延迟缩短了 35%。

4.1.2 多目标规划算法

Hock 等^[28]以提升故障的恢复能力为优化目标,实现了时延和负载均衡的最小化。该策略利用 Pareto 的最优控制器布局求出最优解。经验证,若网络中超过 20% 的节点为控

制器,该算法会有效降低节点或链路故障对网络性能的影响。Ahmad 等^[29]提出了以最小化传播时延为目标的部署策略(MHNCSA-II),该策略将 pareto 底层的穷举算法替换为 NS-GA-II 算法。在实验条件下,MHNCSA-II 与 Pareto 最优解的平均偏差只有 0.8%,但所耗时间很短。

4.1.3 贪婪算法

Ishigaki 等^[30]将最小化控制器间的距离问题抽象为加权最小集覆盖问题,并设计了基于常量权重、平均权重、最坏权重的不同贪婪算法。在给定的实验条件下,对于 D2T 拓扑来说,使用平均权重的贪婪算法计算出的平均控制器的距离为 7.45,常量权重次之,为 7.97。

4.1.4 启发算法

Muller 等^[31]在考虑控制器容量的前提下,提出以最大化可靠性为优化目标的部署策略(Survivor)。该策略基于启发算法实现,并设计了故障转移机制。在实验条件下,应用该策略后可将单链路故障的连通性损失概率降低约 66%。Fan 等^[32]提出在链路失效情况下,以最小化控制器与交换机时延为目标的部署策略。该策略中,粒子群算法负责确定控制器的放置位置。在实验条件下,该算法分别可保证 81% 和 68% 的最坏网络状态延迟短于 10ms。

4.1.5 博弈算法

Killi 等^[33]提出以最小化交换机与控制器间的时延和分区均衡为优化目标的合作 k-means 策略,即将 k-means 算法与合作博弈理论相结合,把网络划分的过程建模为交换机的相互博弈,最终确定控制器的最佳数目。在实验条件下,当控制器个数为 5 时,合作 k-means 计算的最坏时延为 8 ms,而最优解的最坏时延为 7.5 ms。

4.1.6 学习自动机算法

Mostafaei 等^[34]提出了一种基于学习自动机算法的多控制器放置算法。该算法除能确定控制器的最佳数目和位置外,还可确保任何节点到最近控制器的时延不超过期望的阈值。在验证时,设置 4 种拓扑,并将该算法输出的控制器数当作 k-means 算法的输入,而后证明在最小化平均时延、最大时延上该算法均优于 k-means 算法。

4.1.7 监督学习算法

Blenk 等^[35]提出了一种以最小化时延为优化目标的部署策略,该策略在实现时采用监督学习算法(o'zapft is-MIPX, MIPX),将给定的拓扑图以及控制器比率作为输入,最优解作为训练数据,输出节点为控制器概率最高的前 k 个节点, k 为比率与节点数的乘积。在实验中,把 MIPX 的输出作为最优混合整数规划算法(Mixed Integer Programming algorithm, MIP)的输入,极大缩短了 MIP 的运算时间。

4.2 动态部署

动态部署是指可及时感知网络负载的变化,并据此动态地调整控制器数量或交换机与控制器的映射关系,以更好地适应网络的突发性,提升网络性能。其中,控制器数量的调整是通过控制器活动状态与非活动状态之间的转化来实现的,如图 3 所示。映射关系的调整通常是利用交换机的迁移来完成的,如图 4 所示。

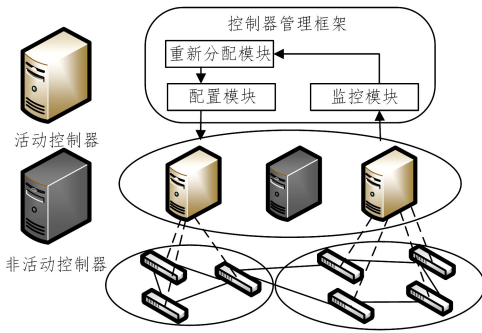


图 3 分布式 SDN 中控制器数量的优化

Fig. 3 Optimization of number of controllers in distributed SDN

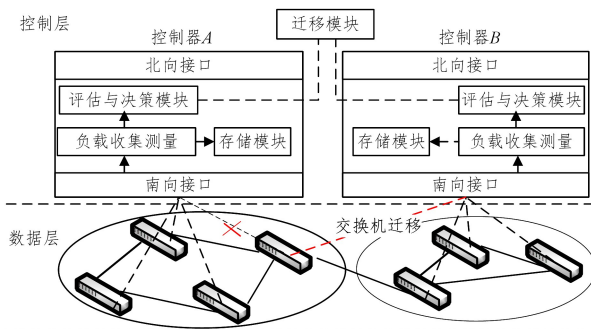


图 4 分布式 SDN 中交换机的迁移

Fig. 4 Switch migration in distributed SDN

4.2.1 贪婪算法

Barim 等^[36]提出了一种可根据网络状态的变化动态调整控制器的数量及位置,最小化通信开销和流建立时间的策略。该文献将 CPP 抽象为一个整数线性规划问题,分别使用贪婪算法(greedy approach based on the knapsack problem, DCP-GK)和模拟退火算法(simulated annealing based meta-heuristic approach, DCP-GA)进行求解。在实验中,基于 RF-II 拓扑进行计算可得,虽然 DCP-GA 在性能上表现得更好,但 DCP-GA 运算耗时约 4 min,但 DCP-GK 仅需要 0.44 s。

4.2.2 启发算法

Mouawad 等^[37]基于分层式体系架构,提出了一种以最小化控制器间延迟、切换延迟和控制器负载为优化目标的部署策略(Optimal and Dynamic Controller Placement, ODCP)。该策略由负载均衡算法和交换机迁移算法组成,其中迁移算法使用启发式算法来实现。在实验条件下,当过载阈值设置为 90% 时,使用 CFPD^[38] 迁移交换机后负载为 89%,而 ODCP 仅有 79%。Kim 等^[39]提出了一种最小化交换机与控制器时延为优化目标且保证负载均衡的放置策略(P-HS-k),该策略中和谐搜索的启发算法(Harmony Search method, HM)主要负责选择合适的交换机迁移,而 k-means 算法用来减小搜索空间。在实验中,当控制器数目为 10、交换机数目为 200 时,HM 算法的执行时间约为 160ms,而 P-HS-k 仅为 100 ms。

4.2.3 博弈算法

Rath 等^[40]以负载均衡、最大化利用率以及最小化时延为优化目标,提出了一种非零和博弈的部署策略。该策略使每个控制器都计算一个支付函数,并将值与邻居相比较,以便做出适当的添加或删除等操作。如最初有 7 个控制器服务于 363 个流请求,当请求数减少至 295 时,控制器 C1 和 C3 可以在不影响 QoS 的情况下关闭。Ksentini 等^[41]提出了以最小化总通信开销且基于讨价还价博弈理论的 FTCS 策略,经运算,最终确定控制器的位置。实验中将最小化控制器间开销的 CCA 策略与最小化交换机与控制器间开销的 SCA 策略进行对比,得出 FTCS 与单一的 CCA 和 SCA 性能相似,但在总通信开销上远优于两者。

4.2.4 监督学习算法

He 等^[42]提出了一种以最小化交换机与控制器时延为优化目标的部署策略(Neural Network Local Search, NN-LS)。在实现时先采用神经网络(Neural Network, NN)算法,输入流量矩阵,输出代表控制器位置的节点,再将该输出当作启发算法的输入,解出 CPP。在实验中,NN-LS 结果的准确度与启发算法相似,但搜索次数减少了一半。

5 未来与展望

目前为止,虽然针对 CPP 已经有了一系列的解决方案,但仍然存在着很多关键问题亟待解决。本文认为未来的研究可以从以下几个方面着手。

(1) 高效算法。CPP 是一个 NP 难问题且计算非常耗时。因此,设计出能在短时间内找到近似最优解的高效算法是未来研究的热点问题。其中,机器学习的发展^[43-46]为高效算法的实现提供了新思路。可尝试在求解 CPP 时,将机器学习与

启发算法相结合,利用重复或相似问题的历史数据来预测初始解并优化计算过程,或能提高算法的准确度并加快执行速度。

(2)多控制器动态部署策略。动态部署策略中,由于优化控制器数量的额外代价较大且算法复杂度较高,导致动态部署多采用迁移交换机来实现,但其也存在迁入域选择僵化、多交换机迁移冲突等问题。若能在选择迁入域时,除考虑迁移距离外,将控制器负载、故障等因素也纳入考虑范围,并对待迁移交换机排序,设置控制器负载门限值,将能避免选择僵化和迁移冲突。

(3)耗能指标。目前的部署策略多是对网络性能的优化,但实际上随着网络规模的不断扩大,网络耗能越发引起关注。现有的降低网络能耗的解决方案^[47-48]大都是根据具体的网络拓扑,在保证必要节点连通性的前提下,尽可能多地切断链路,但此操作所需付出的代价尚缺乏探讨,因此如何用最少网络性能的牺牲换取最大程度的能耗降低是亟需解决的重要问题之一,且网络性能与节能之间的权衡也需根据实际情况进行进一步的考量。

(4)控制器的处理时延。至今为止,绝大多数的部署策略在考虑网络处理时延时,都直接将其忽略或设置为常数。但实际上控制器的负载不同,其处理时延往往会有所差异,导致计算结果偏离预期目标。因此,如何更好地量化控制器的处理时延值得深入研究。其中,可以基于当控制器的负载接近或超过其处理能力时处理时延明显增加的现象,考虑利用负载均衡的指标来衡量处理时延;或者尝试将交换机与控制器之间的通信过程抽象为某种排队模型,再具体分析控制器的处理时延。

结束语 随着SDN在数据中心和广域网中的大规模应用^[49-51],CPP已成为亟待解决的问题。本文基于CPP的产生背景,首先总结和归纳了CPP的实质和通用的解决思路,其次详细介绍了部署策略的两大核心构件。然后对具有代表性的部署策略进行了深入的对比分析,最后展望解决CPP的未来研究方向。

参 考 文 献

- [1] ALHARTHI M,TAHA A M,HASSANEIN H S. Dynamic Controller Placement in Software Defined Drone Networks [C]// IEEE Global Communications Conference. Waikoloa USA,2019:1-6.
- [2] CHOWDHURY N M,KABIR M,BOUTABA R. A survey of network virtualization[J]. Computer Networks, 2010, 54(5): 862-876.
- [3] ZHANG S J,LAN J L,HU Y X,et al. Survey on scalability of control plane in software-defined networking[J]. Journal of Software,2018,29(1):160-175.
- [4] TALHAR P,BHAGAT A P. An Adaptive Approach for Controller Placement Problem in Software Defined Networks[C]// Conference on Research in Intelligent and Computing in Engineering. Salvador,2018:1-11.
- [5] TAO P,YING C,SUN Z,et al. The Controller Placement of Software-Defined Networks Based on Minimum Delay and Load Balancing[C]//IEEE 16th Intl Conf on Dependable, Autonomic and Secure Computing. Athens,2018:310-313.
- [6] TOOTOONCHIAN A. a distributed control plane for OpenFlow [C]//Internet Network Management on Research on Enterprise Networking. 2010.
- [7] YEGANEH S H,GANJALI Y. Kandoo: A framework for efficient and scalable offloading of control applications[C]// ACM Workshop on Hot Topics in Software Defined Networks. 2012: 19-24.
- [8] FU Y,BI J. Orion: A hybrid hierarchical control plane of software-defined networking for large-scale networks[C]// IEEE Conference on Network Protocols. Triangle Park, 2014: 569-576.
- [9] FU Y,BI J,CHEN Z,et al. A hybrid hierarchical control plane for flow-based large-scale software-defined networks[J]. IEEE Transactions on Network and Service Management,2015,12(2): 117-131.
- [10] GAO X M,WANG B S,DENG W P,et al. Survey of controller placement problem in software defined network[J]. Journal of Communications,2017,38(7):155-164.
- [11] ZHANG Y,CUI L,WANG W,et al. A survey on software defined networking with multiple controllers[J]. Journal of Network and Computer Applications,2018,103:101-118.
- [12] RAO B P R,KILLI,RAO S. Controller placement in software defined networks:A Comprehensive survey[J]. Computer Networks,2019,163:18-28.
- [13] HUANG V,CHEN G. Optimizing Controller Placement for Software-Defined Networks[C]// IEEE Symposium on Integrated Network and Service Management. Arlington,2019:224-232.
- [14] ALSHAMRANI A,GUHA S. Fault Tolerant Controller Placement in Distributed SDN Environments[C]// IEEE International Conference on Communications. Kansas,MO,2018:1-7.
- [15] ZHANG B,WANG X W,HUANG M. Multi-objective optimization controller placement problem in internet-oriented software defined network[J]. Computer Communications,2018,123:24-35.
- [16] QI Y,WANG D,YAO W,et al. Towards Multi-Controller Placement for SDN Based on Density Peaks Clustering[C]// IEEE International Conference on Communications. Shanghai, 2019:1-6.
- [17] SHI J G,XIE Y J,SUN L,et al. Multi-controller Placement Strategy Based on Latency and Load in Software Defined Network[J]. Journal of Electronics & Information Technology, 2019,41(8):1869-1876.
- [18] FAN Z,YAO J,YANG X,et al. A Multi-Controller Placement Strategy Based on Delay and Reliability Optimization in SDN [C]// Wireless and Optical Communications Conference. Beijing,2019:1-5.
- [19] PIZZUTI C. A Multi-objective Genetic Algorithm to Find Communities in Complex Networks[J]. IEEE Transactions on Evolutionary Computation,2012,16(3):418-430.
- [20] GAO Y,LIU C. Hybrid ant colony algorithm for logistics distribution problem with time windows[C]// International Symposium on Computational Intelligence and Design. Hangzhou, 2017:289-291.
- [21] RAMTEKE V T P,MANOJKUMAR V. A simplified multi-objective particle swarm optimization algorithm[J]. Swarm Intelligence,2019:18-26.
- [22] WANG K,LIAN X,PAN B. Improved Cuckoo Search Algorithm Based on Exponential Function[C]// Chinese Intelligent

- Automation Conference. Singapore, 2019:200-207.
- [23] SUTTON R S, BARTO A G. Reinforcement Learning: An Introduction[M]. MIT Press, 1998.
- [24] XIE J, YU F R, HUANG T, et al. A survey of machine learning techniques applied to software defined networking (sdn): research issues and challenges[C]// IEEE Communications Surveys & Tutorials. Toyama, 2019:393-430.
- [25] HELLER B, SHERWOOD B. The controller placement problem [C]// SIGCOMM. Computer Communication Review, 2012:473-478.
- [26] YAO G, BI J, LI Y, et al. On the capacitated controller placement problem in software defined networks[J]. IEEE Communications Letters, 2014, 18(8):1339-1342.
- [27] QI Y Z, WANG D B, YAO W B, et al. Towards Multi-Controller Placement for SDN Based on Density Peaks Clustering[C]// IEEE International Conference on Communications, 2019.
- [28] HOCK D, HARTMANN M, GEBERT S, et al. Pareto-optimal resilient controller placement in SDN-based core networks[C]// IEEE 25th International Teletraffic Congress(ITC). 2013:1-9
- [29] AHMAD J, MANIJEH K. Optimal controller placement in large scale software defined networks based on modified NSGA-II[J]. Applied Intelligence, 2018, 48(9):2809-2823.
- [30] ISHIGAKI G, GOUR R. Cluster Leader Election Problem for Distributed Controller Placement in SDN[C]// IEEE Global Communications Conference. Singapore, 2017:1-6.
- [31] MULLER L F, OLIVEIRA R R. an enhanced controller placement strategy for improving SDN survivability [C]// IEEE Global Communications Conference. Austin: TX, 2014: 1909-1915.
- [32] FAN Z F, YAO J, YANG X H. A multi-controller placement strategy based on delay and reliability optimization in SDN [C]// Wireless and Optical Communications Conference. Beijing, 2019:58-62.
- [33] KILLI B P, REDDY E A, RAO S V. Cooperative game theory based network partitioning for controller placement in SDN [C]// International Conference on Communication Systems & Networks. Bengaluru, 2018:105-112.
- [34] MOSTAFAEI H, MENTH M. A Learning Automaton-Based Controller Placement Algorithm for Software-Defined Networks [C]// IEEE Global Communications Conference. Abu Dhabi, United Arab Emirates, 2018:1-6.
- [35] BLENK A, KALMBACH P, SCHMID S, et al. o'zapft is; Tap Your Network Algorithm's Big Data! [C]// ACM Sigcomm Workshop on Big Data Analytics. ACM, 2017.
- [36] BARIM F, ROYA R. Dynamic controller provisioning in software defined networks[C]// International Conference on Network and Service Management. 2013:18-25.
- [37] MOUAWAD N, NAJA R. Optimal and Dynamic SDN Controller Placement[C]// Interational Conference on Computer and Applications. Beirut, 2018:1-9.
- [38] YAO L, HONG P, ZHANG W. Controller placement and flow based dynamic management problem towards sdn[C]// Proceedings of the International Conference on Communication Workshop (ICCW). IEEE, 2015:363-368.
- [39] KIM S, KIM S, BIRZHANDI P, et al. A Solution for Dynamic Controller Provisioning Problem of SDN using Harmony Search [C]// International Congress on Advanced Applied Informatics. Toyama, 2019:605-610.
- [40] RATH H K, REVOORI V, NADAF S, et al. Optimal controller placement in software defined networks using a non-zero-sum game[C]// Symposium on a World of Wireless, Mobile and Multimedia Networks. Sydney: NSW, 2014:1-6.
- [41] KSENTINI A, BAGAA M. On using bargaining game for optimal placement of SDN controllers[C]// IEEE International Conference on Communications (ICC). Lumpur, 2016:1-6.
- [42] HE M, KALMBACH P, BLENK A, et al. Algorithm-data driven optimization of adaptive communication networks [C]// IEEE 25th International Conference on Network Protocols (ICNP). Toronto, 2017:1-6.
- [43] BLENK A, KALMBACH P. Tap Your Network Algorithm's Big Data [C]// Proceedings of the Workshop on Big Data Analytics and Machine Learning for Data Communication Networks. New York; NY, 2017:19-24.
- [44] BLENK A, KALMBACH P. A Neural Preprocessor for Your Virtual Network Embedding Algorithm[C]// IEEE Conference on Computer Communications. Honolulu; HI, 2018:405-413.
- [45] HUANG V, CHEN G. Optimizing Controller Placement for Software-Defined Networks[C]// IEEE Symposium on Integrated Network and Service Management. Arlington, 2019:224-232.
- [46] YAO H, MAI T. An Intelligent Network Architecture for Self-Learning Control Strategies in Software Defined Networks[J]. IEEE Internet of Things Journal, 2018, 5(6):4319-4327.
- [47] HU Y, LUO T, BEAULIEU N C. The energy-aware controller placement problem in software defined networks[J]. IEEE Communications Letters, 2017, 21(4):741-744.
- [48] FERNÁNDEZ A, CERVEÓ C. Energy efficiency and network performance: A reality check in SDN-based 5G systems[J]. Energies, 2017, 10(12):2132.
- [49] KSENTINI A, BAGAA M, TALEB T. On Using SDN in 5G: The Controller Placement Problem[C]// IEEE Global Communications Conference. Washington; DC, 2016:1-6.
- [50] HIRAYAMA T, MIYAZAWA T, MUKUTAIR A H, et al. Saliency-Based Distributed Controllers Placement in Software Defined Networks[C]// IEEE Global Communications Conference. Abu Dhabi; United Arab Emirates, 2018:1-7.
- [51] HUANG V, CHEN G, FU Q. Optimizing Controller Placement for Software-Defined Networks. [C]// IEEE Symposium on Integrated Network and Service Management. Arlington, 2019: 224-232.



JIA Wu-cai, born in 1996, postgraduate, is a member of China Computer Federation. Her main research interests include SDN and deep learning.



LV Guang-hong, born in 1963, Ph. D, professor. His main research interests include SDN, optical network and machine learning.