

基于 OpenFlow 网络的数据中心服务器负载均衡策略

秦 华, 阎 钢

(北京工业大学计算机学院, 北京 100124)

摘 要: 目前基于虚拟机迁移的服务器负载均衡研究多数只考虑服务器端的资源使用情况, 忽略了网络状况以及服务器之间的通信关联等因素, 导致虚拟机迁移后网络拥塞而影响系统性能。针对该问题, 结合 OpenFlow 网络控制器可通过网络状态信息集中管理整个网络的特点, 设计基于 OpenFlow 网络的数据中心服务器负载均衡策略。综合考虑系统资源、网络拓扑以及网络链路可用带宽, 根据服务器负载状况和虚拟机之间通信对网络资源的需求评估虚拟机迁移后的运行和通信开销, 以及迁移过程带来的网络资源开销, 选择开销最小的主机作为迁移目标的主机。实验结果表明, 该策略能实现服务器的负载均衡, 保证虚拟机上运行业务的服务质量。

关键词: OpenFlow 网络; 数据中心; 虚拟机迁移; 服务器负载均衡; 网络感知; 时延优化

中文引用格式: 秦 华, 阎 钢. 基于 OpenFlow 网络的数据中心服务器负载均衡策略[J]. 计算机工程, 2016, 42(3): 130-137.

英文引用格式: Qin Hua, Yan Gang. Server Load Balancing Strategy in Data Center Based on OpenFlow Network[J]. Computer Engineering, 2016, 42(3): 130-137.

Server Load Balancing Strategy in Data Center Based on OpenFlow Network

QIN Hua, YAN Gang

(School of Computer, Beijing University of Technology, Beijing 100124, China)

【Abstract】 Previous researches mainly focus on balancing physical server utilization based on Virtual Machine (VM) migration and ignore network status and the communication dependencies among VMs, which may result in network congestion and affect the system performance. To solve the above problems, this paper designs a server load balancing strategy in data center based on OpenFlow network. OpenFlow controller has the ability of managing the whole network according to switch and flow level aggregate statistics collected from the data plane. The load balancing strategy proposed in this paper takes account of both server-side constraints and network resources required by communications among VMs based on topology and the available link bandwidth, as well as the migration cost. It selects a destination server with least amount of system and network cost for a migrated VM. Experimental result shows that this strategy can reduce the possibility of network congestion and ensure the quality of service for the application on the VM.

【Key words】 OpenFlow network; data center; Virtual Machine (VM) migration; server load balancing; network awareness; latency optimization

DOI: 10.3969/j.issn.1000-3428.2016.03.024

1 概述

随着数据中心提供的应用服务越来越丰富, 数据中心内部网络流量模式已由传统的以“南北”流量为主逐渐变为以“东西”流量为主导。网络流量传输模式的变化和业务量不断增加使数据中心的服务器处理能力和网络传输能力受到了极大的考验, 如何有效合理利用服务器资源和网络资源, 降低运行成本并保证运行业务的服务质量成为了亟待解决的问题。

此外, 数据中心需要提供 7×24 h 不间断服务, 任何服务器的运行崩溃或者服务中断都会导致数据丢失而造成很大的损失。若单纯采用高性能服务器、增加网络带宽等提升硬件性能的方法而没有合理科学的管理策略, 很容易造成系统资源和网络资源的浪费, 而且硬件的价格昂贵, 给数据中心的维护管理带来很高的成本。为解决上述问题, 更好地利用服务器和网络资源, 可以采用传统的负载均衡技术结合流行的虚拟化技术以充分利用系统资源。负

作者简介: 秦 华 (1965 -), 女, 教授级高级工程师、博士, 主研方向为互联网体系结构、网络安全; 阎 钢, 硕士。

收稿日期: 2015-02-12 **修回日期:** 2015-03-31 **E-mail:** qinhua@bjut.edu.cn

负载均衡是实现资源有效利用和共享的重要手段,其意义就是将计算机资源重新分配,对资源合理利用以获得最佳资源利用率。OpenFlow 技术的兴起给管理人员对网络管理和控制带来了新的方法。OpenFlow 协议支持对网络设备和网络流的管理和配置,以及对它们运行状态的监控,可以按照管理人员的需求和网络的状态在控制器上制定相应的策略实现网络资源的灵活使用,提高网络资源利用率。OpenFlow 技术在数据中心有广泛的应用和研究。现代数据中心特点与以往有很多变化,如何根据现代数据中心的特点结合 OpenFlow 技术制定有效的负载均衡策略,实现系统资源和网络资源的充分利用成为新的研究热点。

本文设计基于 OpenFlow 网络的数据中心服务器负载均衡框架,集中监控物理服务器和虚拟机的资源使用情况,结合 OpenFlow 控制器对网络进行集中管理和网络状态监控,通过 OpenFlow 控制器获取网络拓扑、链路可用带宽、网络传输时延等信息,作为负载均衡算法的依据;设计服务器网络联合负载均衡算法,在迁移目的主机选择上综合考虑服务器和网络资源开销代价以及虚拟机的迁移开销,避免虚拟机在迁移后由于服务器资源或网络拥塞影响应用程序的运行,优化虚拟机之间通信时延;最后利用 SDN 网络模拟实验平台 Mininet 创建数据中心拓扑,通过实验对本文的 OpenFlow 负载均衡框架和服务器网络联合负载均衡算法加以验证,并与仅考虑服务器资源的贪心算法进行比较。

2 相关研究

目前,虚拟化技术被广泛用于数据中心的,基于虚拟机的资源分配调度模式已成为数据中心的网络资源有效利用的主要方式^[1]。它允许一台物理服务器运行多个虚拟机,并以虚拟机为单元按需为上层应用提供支持。当服务器过载时,可以通过热迁移技术将虚拟机迁移到轻载物理主机上,减小源服务器的负载,极大提高服务器资源的利用率。但由于虚拟机这种动态迁移的灵活性,也给数据中心网络管理带来了极大的挑战。当前关于虚拟机的研究主要集中于资源调度,能量优化,虚拟机部署问题,负载均衡问题等几方面。在基于虚拟机迁移的负载均衡研究方面,文献[2]设计了 Sandpiper 系统,提出了基于黑盒方法和灰盒方法的服务器资源利用率监控方法,它能自动监测和探测热点,并通过虚拟机迁移完成新的虚拟机到物理服务器的映射,消除系统热点。文献[3]提出了虚拟机集群模式下的动态资源分配负载均衡方案,通过监控物

理服务器内部多个虚拟机的资源使用情况对虚拟机进行资源重新分配,达到本地虚拟机之间的负载均衡,在多个物理服务器之间通过虚拟机的热迁移达到全局负载均衡。文献[4]针对虚拟机集群资源负载不平衡的问题,基于虚拟机迁移技术提出一种虚拟机集群资源调度策略,使虚拟机集群达到负载均衡和节能的双重目标。上述文献只考虑服务器的系统资源使用状况,然而在现代数据中心中由于服务器之间的虚拟机会进行频繁的数据通信而使网络资源成为数据中心新的瓶颈资源^[5-6]。

文献[7]提出了基于网络感知的虚拟机放置研究,根据虚拟机之间流量矩阵关系部署虚拟机,以减小虚拟机之间通信开销占用的网络资源,避免网络拥塞。上述文献提出的方法只分析了虚拟机的初始部署方法,没有考虑虚拟机运行过程中因发生迁移使相关流量分布发生变化而可能导致的网络拥塞,影响运行业务的服务质量。

在 OpenFlow 网络中服务器负载均衡研究方面,文献[8]在面向无结构网络中,通过监测网络链路延迟和服务器负载,将用户的访问请求分配到合适的服务器处理并通过 OpenFlow 控制器控制请求转发的网络通路,以最小化响应时间。该方案的不足是随着网络规模的扩大和服务器数量的增多,若控制器同时对网络和服务器做监测会增加控制器的负担,影响控制器对网络的管控功能。文献[9]使用连接到 NOX 控制器的 OpenFlow 交换机实现了一个替代传统基于硬件的负载均衡架构。该架构在 NOX 控制器上维护通过交换机连接的所有服务器信息,并在 handle 模块中实现了 3 种负载均衡策略:Random, Round Robin 和 Load-Based,当用户请求到达时修改数据包的目的 MAC 和 IP 地址将请求分配到不同的服务器上。该方案提出的网络结构简单,负载均衡策略也只是单一考虑了计算资源而没有考虑网络状况。文献[10]在 OpenFlow 网络中将 FlowVisor 和多控制器结合,让不同的控制器分别管理运行不同业务的服务器,并根据它们的状态将用户请求分配到不同服务器上实现负载均衡。由于虚拟化在数据中心的应用,虚拟机迁移技术使得其所在物理主机的位置不固定,而该架构无法很好地监控网络中运行的虚拟机,且 FlowVisor 的使用使其容易成为多控制器区分业务和处理数据的瓶颈。文献[11]提出了基于通配符匹配的服务器负载均衡算法,以网络中流请求的 IP 地址前缀为通配符并按 0 或 1 构建一棵二叉树,根据 IP 地址中相应位是 0 还是 1 匹配到相应服务器,并形成转发规则安装到交换机中,降低控制器对数据包的处理次数^[12]。该方法需要对 OpenFlow

交换机的流表定义形式做修改,且需要特殊的交换机硬件支持。

现代数据中心呈现出集群通信的特点^[13],服务器和服务器之间通信更加频繁,包含多种复杂的通信模式,数据中心会承载很大的流量。例如:很多搜索服务往往会执行分布式计算,涉及一对多的任务分发和汇总;MapReduce 数据处理模型应用于海量数据处理,涉及大量的文件数据传输和多对多节点之间计算结果交互,这些对网络带宽和时延都有较高的要求。然而,网络传输能力的增长往往滞后于计算能力的增长,相比于高性能服务器的发展,网络资源管理和控制显得更为薄弱。最新研究表明,网络性能往往成为数据中心的性能瓶颈,网络拥塞、负载不均衡等都会导致分组丢失、超时等影响业务的服务质量^[13]。为此,现在的数据中心不能仅是计算资源负载优化,更要考虑到网络实时状态对应用服务的影响,避免网络拥塞保证服务质量。本文提出了基于 OpenFlow 网络的数据中心服务器负载均衡框架,并设计相应的负载均衡策略,通过迁移虚拟机减轻过载服务器上的负载。迁移决策的制定不仅考虑了服务器系统资源限制的因素,还考虑虚拟机之间通信关联和底层网络的拓扑结构,网络传输时延

以及当前网络的可用带宽,根据虚拟机与其他虚拟机之间通信关联对网络资源的需求,预估虚拟机迁移后的通信延迟开销以及虚拟机迁移对目的主机系统资源开销,选择开销最小的物理主机作为目的主机,实现服务器端负载均衡的同时避免网络拥塞而影响虚拟机上业务的运行。

3 数据中心负载均衡设计

3.1 负载均衡系统框架

本文采用基于 OpenFlow 架构来管理数据中心网络,OpenFlow 控制器可以对网络中设备进行集中管理和控制,集中控制器可以从全局角度调配网络资源来满足业务需求,实现网络资源利用率优化。服务器管理节点负责获取物理主机和虚拟机的资源使用情况以及根据负载均衡决策执行虚拟机迁移。OpenFlow 控制器会与服务器管理节点交互得到服务器的负载信息,为控制器做出负载均衡决策提供依据。控制器根据获得的参数和负载均衡策略做出迁移决策,选出迁移的虚拟机和放置虚拟机的目的主机,将决策发送给服务器管理节点由它做执行迁移命令实现迁移。本文提出基于 OpenFlow 网络的服务器负载均衡框架,如图 1 所示。

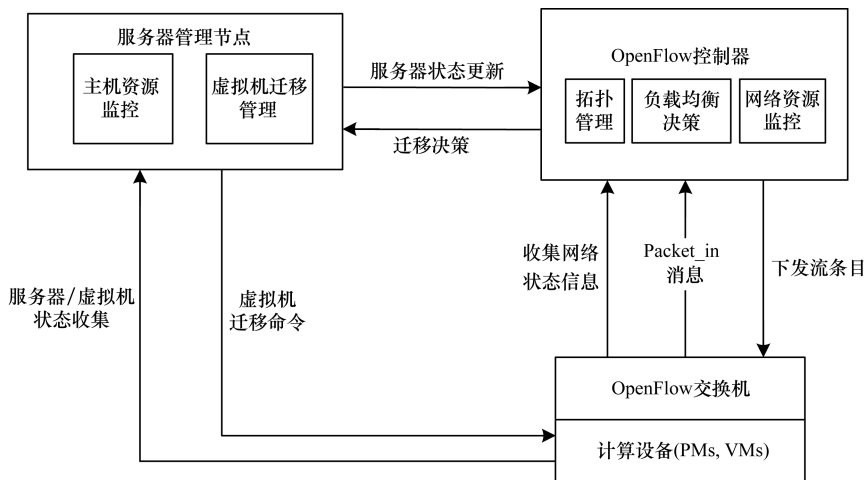


图 1 基于 OpenFlow 网络的服务器负载均衡框架

3.2 服务器管理节点

服务器管理节点主要负责监控和管理网络中的服务器和虚拟机,收集它们的资源使用情况,这些信息将作为负载均衡决策的依据,并根据策略对虚拟机实现迁移。服务器管理节点上的服务器监控模块负责收集服务器负载信息(CPU 使用率、内存使用率、网络带宽)以及运行在服务器上的虚拟机的资源使用情况。为有效监控各服务器的运行状况,管理服务器每隔时间 Δt 收集一次负载信息,并计算各系统资源利用率,判断是否有服务器过载。为了避免

由于服务器突发性的负载变大而造成不必要的虚拟机迁移,将 T 时间内监测到的各系统资源利用率的平均值作为服务器负载的度量指标。在 OpenFlow 控制器做出虚拟机迁移决策后,会将决策发送给管理节点,由迁移管理模块向虚拟机发送迁移命令,完成迁移。

3.3 基于 OpenFlow 网络的监控与管理

OpenFlow 网络有对整个网络进行配置和管理的能力,通过 OpenFlow 协议可以根据策略向交换机下发数据转发规则,查询交换机状态信息,获取网络

中每个网络流的统计信息等。所以, 通过 OpenFlow 网络可以掌握全局的网络拓扑信息及其变化情况, 并通过周期性的查询交换机端口信息, 得到每条链路的带宽利用率和可用带宽, 从而得到此时整个数据中心的网络资源使用情况, 构建全网资源视图, 作为服务器负载均衡决策的依据。

OpenFlow 控制器周期性地发送端口查询消息 (OFPST_PORT)^[14], 收到统计信息计算交换机上每个端口在查询间隔内收发字节数, 计算出对应链路上的可用带宽。服务器管理节点确定要迁移的虚拟机 VM_i 后, 需要知道虚拟机与其他主机的通信关联。OpenFlow 控制器上的拓扑发现模块和 host_trakcer 模块探测整个网络拓扑和主机位置, 为网络管理提供了全局拓扑视图。

4 基于 OpenFlow 网络感知的负载均衡策略

现代数据中心内部主要表现为多节点之间的集群通信, 通信模式主要有一对多和多对多。一对多通信时, 服务器之间往往有较大通信量, 因此, 通信的服务器往往都被放在同一个接入交换机或者汇聚交换机下, 使它们之间可以有较大的通信带宽, 但这容易产生 incast 现象, 导致数据包丢失^[1]。对于多对多通信, 很多应用都有软时延限制, 响应时延超过一定限制就会影响系统数据处理进而影响用户体验, 因此, 系统的性能取决于网络中传输延迟, 而网络中大量的服务器通信流量很容易导致网络拥塞, 应通过流的优化调度及服务器的合理布局实现网络流量平衡及 QoS 保证^[1]。

为了实现网络资源的有效利用, 合理布局服务器避免网络拥塞、传输延迟等影响服务器上应用的性能, 服务器管理节点上的负载均衡模块以收集到的服务器负载状况和从 OpenFlow 控制器上得到的拓扑信息、网络带宽及网络延迟等网络状态作为负载均衡策略制定的依据, 执行负载均衡策略并做出迁移决策。负载均衡方法过程如下:

(1) 虚拟机迁移触发策略。设网络中物理主机集 $PM = \{PM_1, PM_2, \dots, PM_n\}$, 用向量描述任一物理服务器 PM_i 上的各系统资源利用率: $pmLoad_i = \{PM_i: <pLoad(cpu), pLoad(mem), pLoad(net)>\}$ 。网络中虚拟机集 $VM = \{VM_1, VM_2, \dots, VM_k\}$, 虚拟机的资源利用率描述向量 $vmLoad_i = \{VM_i: <vLoad(cpu), vLoad(mem), vLoad(net)>\}$ 。对于物理主机 PM_i 上运行的虚拟机可表示如下: $PM_i_VM = \{vm_ip_1, vm_ip_2, \dots\}$ 。虚拟机上运行的不同的业务对系统资源需求不同, 因此, 在收集到系统负载信息后分别判

断各项统计数据是否超过系统阈值, 即 $pLoad(cpu) \geq TH_{cpu} \parallel pLoad(mem) \geq TH_{mem} \parallel pLoad(net) \geq TH_{net}$, 当有任何一个系统资源超过阈值后, 就需要对服务器的负载进行调整。阈值 TH_{cpu} , TH_{mem} 和 TH_{net} 由虚拟机执行的任务对系统资源的需求而定。

(2) 迁移目标选择策略。当服务器的某项或某几项资源超过阈值, 则这几个资源即为系统目前的瓶颈资源。找到系统瓶颈资源后, 要选择虚拟机迁移以减轻服务器的负载。不同于传统的贪心算法选择占用瓶颈资源最多或最少的虚拟机作为迁移目标, 迁移目标选择还要考虑虚拟机迁移带来的开销。在基于共享存储的数据中心中, 虚拟机迁移开销与虚拟机占用的内存大小有关^[15], 内存越大则迁移过程中传输的数据量越多, 消耗的系统资源也越多。因此, 为了减小过载主机负载的同时减小虚拟机迁移开销, 以传输单位数据量可以转移的负载^[2]作为迁移目标选择的依据, 即:

$$VS = \frac{\sum util(resource)}{vMemory} \quad (1)$$

其中, $\sum util(resource)$ 为虚拟机各资源利用率之和; $vMemory$ 为虚拟机占用内存大小。VS 值越大, 则说明转移单位数据量减小的负载越大。对每个虚拟机计算 VS 值并按该值降序排列, 按序将虚拟机作为迁移目标, 直至主机负载降到阈值以下。

(3) 目的主机选择策略。得到过载物理机上的虚拟机排序列表后, 按照虚拟机的排列顺序以第一个虚拟机作为迁移对象, 然后选出目的主机接受迁移虚拟机。更新服务器资源视图, 获得网络中最新的服务器资源使用情况, 找到满足虚拟机资源需求的主机且虚拟机迁移到目的主机后物理主机的资源使用率不会超过负载阈值。

$$hostCost = \frac{VM_{cpu}}{PM_{rcpu}} \times \frac{VM_{mem}}{PM_{rmem}} \times \frac{VM_{net}}{PM_{rnet}} \quad (2)$$

其中, PM_{rcpu} , PM_{rmem} , PM_{rnet} 分别表示服务器上的剩余系统资源; VM_{cpu} , VM_{mem} , VM_{net} 表示虚拟机运行需要的资源, 服务器的每项剩余资源都要大于虚拟机的需求才能满足迁移条件。hostCost 值越小则虚拟机与物理主机越匹配。将满足虚拟机需求的主机加入候选主机列表并由小到大排列。

更新网络资源视图, 由 OpenFlow 控制器获得最新的网络拓扑信息以及网络链路可用带宽。确定了迁移目标后, 通过 OpenFlow 协议提供的 OFPST_FLOW 类型的 OFPST_STATS_REQUEST 消息查询虚拟机连接的接入交换机上所有的流信息, 得到每个流的 packet 和 byte 的数量^[14], 由这些

信息汇总所有与迁移虚拟机 u 通信的其他主机集合 COM_VM_u 以及它们之间的流传输速率。由于虚拟机迁移后,与其他虚拟机通信的网络流的传输路径也会变化,为使虚拟机迁移后使其上面运行的业务受迁移的影响最小,需要将系统资源和网络资源均满足虚拟机运行和通信的需求,避免迁移虚拟机受到网络状况的影响而降低服务质量,同时防止迁移后的流量分布变化造成网络拥塞。所以,本文根据数据中心网络拓扑和物理主机构造一个全局模型,将网络容量、可用带宽和网络延迟等信息都量化到每个边上。通过这个全局模型,预测虚拟机迁移到目的主机后的网络通信资源开销,并且还要保证分配到链路上的数据流满足链路容量,不会造成链路拥塞。

先对数据中心网络模型做如下定义:拓扑图 $G(V, E)$, V 表示图中的节点集合,可以是网络设备或者是主机, E 表示图中的边集合,可以是交换机之间或者交换机与主机之间的链路。用向量描述链路 $e < s, t, Re, de >$, s, t 分别表示链路两端交换机, Re 为链路 e 上的剩余带宽, de 表示链路传输延迟。虚拟机 u, v 之间通信关联 $f < u, v, f_{u,v}, n >$, $f_{u,v}$ 表示虚拟机 u 和 v 之间流的传输速率,这里 u 和 v 之间的网络流 $f_{u,v}$ 是基于主机粒度的,包括 2 个虚拟机之间通信的所有数据流的速率和。 n 是 u 和 v 之间存在的网络流数目, n 越大,则说明主机 u, v 间通信越频繁。设与迁移虚拟机 u 有通信关联的主机集合 $COM_VM_u = \{VM_i | \forall < VM_i, VM_j, f_{u,v}, n >, f_{u,v} > 0\}$ 。

当虚拟机 u 从源物理主机 PM_i 迁移到目的主机 PM_j 后,与该虚拟机有通信关联的任一虚拟机 v 之间通信路径变为 P ,则 u 与 v 之间通信开销 $R(p)$ 是对于沿路径 P 各链路为该路径预留资源的总和与该业务流占用该路径的时间的乘积^[16]:

$$R(p) = \sum_{t \in n} \sum_{s \in n} x_{st} f_{u,v} D(P) = h(P) f_{u,v} D(P) \quad (3)$$

其中, x_{st} 取值为 0 或 1,表示链路 s, t 是否属于路径 P ; $h(P)$ 为路径 P 的跳数; $D(P)$ 为路径 P 的各链路传输时延和:

$$D(P) = \sum_{e \in P} \frac{1}{R_e} \quad (4)$$

表示链路 e 上传输单位数据需要的时间; $f_{u,v}$ 为虚拟机 u, v 之间网络流的传输速率。由式(4)可以看出,通信开销大小与跳数和网络传输时延有关,体现了时延优化准则^[16]。则对于 $\forall v \in COM_VM_u$, 由式(3)和式(4)可得总的网络通信开销为:

$$comCost = \sum_{v \in COM_VM_u} \frac{n}{flow_all} h(P) f_{u,v} \sum_{e \in P} \frac{1}{R_e} \quad (5)$$

其中, $flow_all$ 表示与迁移虚拟机 u 相关的所有网络流数,以 $\frac{n}{flow_all}$ 作为其他主机与虚拟机 u 的通信开销计算的权重。

由于虚拟机在线迁移需要传输其内存和 CPU 等状态信息,也会给网络带来一定开销。因此,迁移数据量与虚拟机的内存大小有很大的关系。由文献[17]得出,整个迁移过程传输数据量与内存大小、脏页率有关,整个迁移过程需要迁移的数据量 N 为:

$$N = M \times \frac{1 - (R/L)n + 1}{1 - (R/L)} \quad (6)$$

其中, M 为系统占用内存大小; R 为内存脏页率; n 表示为内存迭代次数; L 为传输路径上分配给迁移网络流的可用带宽。由迁移数据量和传输路径上可用带宽,可以估算整个迁移过程需要的时间 $T = \frac{N}{L}$ 。当目的主机与源主机之间经过的跳数越多,

则说明传输流量占用的链路带宽资源越大,迁移代价与网络传输距离也相关。设迁移流量传输路径为 P' ,则由式(3)和式(6)可得,迁移过程的网络资源开销为:

$$migCost = h(P) LT = h(p) N \quad (7)$$

总的网络开销应包括迁移开销和迁移后虚拟机的通信开销:

$$netCost = migCost + comCost \quad (8)$$

约束条件:

$$\sum_{v \in COM_VM_u} f_{u,v} x_e < R_e, e \in P \quad (9)$$

$$Lx_e < R_e, e \in P' \quad (10)$$

$$x_e \in [0, 1] \quad (11)$$

式(9)表明分配到链路 e 的所有网络流或者迁移流量占用带宽之和不能超过链路可用带宽 Re 。式(10)则表示分配到链路的迁移流量带宽也不能超过路径 P' 的可用带宽限制。对于每个候选主机计算其网络通信开销,并按 $netCost$ 值升序排列。根据候选主机的 $netCost$ 和 $hostCost$ 值的排序,选择两者排序和最小的主机作为目的主机。

(4)通过上述计算模型计算得到目的主机后,继续判断过载主机过载状态是否消除,若没有,则重复上述过程,继续选择虚拟机和目的主机进行迁移。整个服务器网络联合优化算法过程的伪码描述如下:

输入 由步骤(1)和步骤(2)得到的过载主机列表 *overPMs* 以及对应的排序虚拟机集合

输出 迁移决策列表 *Migrate_list* $\{ (VMmig, Sdest) \}$

```

Function DstPM_Select
Migrate_list  $\leftarrow \Phi$ 
PMLoad  $\leftarrow$  网络中所有物理服务器内存, CPU 和网卡带宽使用状态
E  $\leftarrow$  更新各网络链路可用带宽, 时延状态
for p in overPMs
//从过载主机p上选择虚拟机迁移, 直到没有资源超过
//负载阈值
While (pLoad(CPU)  $\geq$  THcpu || pLoad(Net)  $\geq$  THnet || pLoad(Mem)  $\geq$  THmem)
    for u in p.vmlist //p.vmlist 是主机 p 上按 vs 值降序
//排列的虚拟机集合
        //获得与v有通信关联的虚拟机集合 COM_VMv 和相应
//的网络流 fu,v 集合 F
        (COM_VMu, F)  $\leftarrow$  GetFlow(u)
        //检查满足虚拟机约束的物理机, 返回满足条件的候选
//物理主机列表
        Candidate_host  $\leftarrow$  Check_constraints(PMs)
        //对候选主机按照 hostCost 值降序排列
        Sort_host(Candidate_host)
        //计算迁移开销和迁移后通信开销
        for PMi in candidate_host
            //flag 标识物理主机 PMi 是否满足虚拟机 v 的网络资源
//需求, 初始值为 true
            flag  $\leftarrow$  true
            for v in COM_VMu
                计算 v 和 PMi 之间的最短路径 P
                If 路径 P 不满足 u 和 v 的通信带宽需求
                    flag = false
                    break
                else
                    计算 u, v 之间的通信开销 comCost
                    为流 fu,v 预留带宽, 更新路径 P 上各链路的
可用带宽
                end for
                If flag = false //PMi 不满足虚拟机 u 的网络通信资源
//需求
                    将链路可用带宽恢复到原状态
                Else
                    计算虚拟机 u 迁移到 PMi 的迁移开销 migCost
                    netCost  $\leftarrow$  migCost + comCost
                End for
            //对所有满足需求的主机按 netCost 值升序排列
            Sort_net(candidate_host)
            DestPM  $\leftarrow$  PMi with the highest sum of hostCost and

```

```

netCost ranks
if DesPM  $\neq$  p
    VMmig  $\leftarrow$  u
    Sdest  $\leftarrow$  DestPM
    更新 DestPM 上个资源的状态, 为虚拟机 u
预留系统资源
End if
在目的主机 DestPM 上保留资源接受迁移虚
拟机 v
更新主机 DestPM 上的可用资源状态记录
Migrate_list.append((VMmig, Sdest))
End for
End while
End for
Return Migrate_list

```

(5) 得到迁移决策 (*VMmig*, *Sdest*) 列表后, 服务器管理节点向虚拟机发送迁移命令实现迁移。当管理服务器监控到虚拟机启动并再次收集到系统信息后说明迁移完成, 服务器管理节点更新服务器资源使用情况。

5 实验评估和分析

5.1 虚拟软件测试平台搭建

Mininet 是一款虚拟化的软件定义网络测试平台, 本文采用 Mininet 自定义数据中心网络拓扑。Mininet 提供了多种拓扑类型, SingleSwitchTopo, LinearTopo 等, 自身并不支持三层树形网络, FatTree 等网络拓扑, 因此, 需要通过 Mininet 的自定义拓扑功能构建自己需要的树形数据中心网络。之后, 通过 Mininet 的 intf 接口将安装 Mininet 主机的网卡添加到虚拟数据中心拓扑的接入层交换机上, 这样外部主机可以通过接到 Mininet 主机的网卡与内部 Mininet 自定义网络拓扑连接, 犹如将主机接到数据中心网络中。为测试本文提出的算法, 搭建了如下实验环境: 将 3 台配置相同的物理主机接到 Mininet 网络中, 并在上面创建虚拟机。在另一台服务器上安装 POX 控制器作为数据中心网络的控制器。限于实验环境在控制器上实现虚拟机管理功能, 并通过监控管理服务器通过迁移虚拟机实现服务器负载均衡。服务器之间通过 NFS 实现共享存储, NFS 服务器安装在虚拟机中连入数据中心网络, 实验拓扑如图 2 所示。整个网络中共有 4 台物理主机, 创建 7 台虚拟机, 同时在 Mininet 内部创建若干个虚拟主机 host。虚拟机分布和资源需求如表 1 所示。虚拟机与物理主机之间采用桥接的网络连接模式, 每个虚拟机有自己独立的 IP 地址。

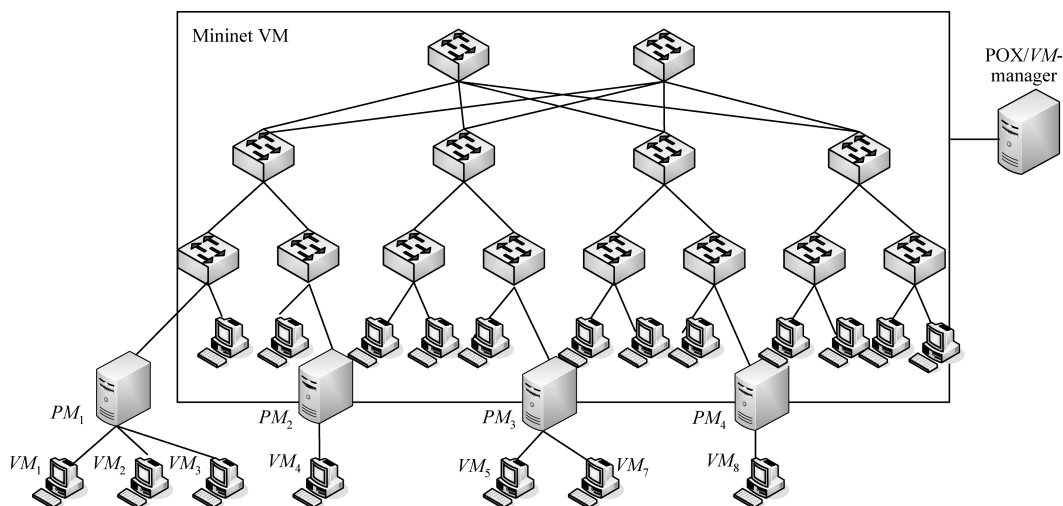


图2 实验环境

表1 虚拟机分布及资源需求表

物理主机	虚拟机	CPU 数量	内存/ MB	网络带宽/ (Mb · s ⁻¹)
PM ₁	VM ₁	2	512	100
	VM ₂	1	1 024	100
	VM ₃	1	512	100
PM ₂	VM ₄	1	512	100
PM ₃	VM ₅	2	512	100
	VM ₆	1	512	100
PM ₄	VM ₇	1	1 024	100

数据中心网络流量模拟,本文采用2种方式:

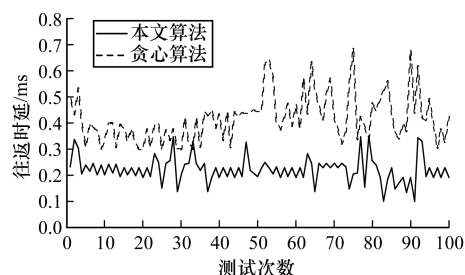
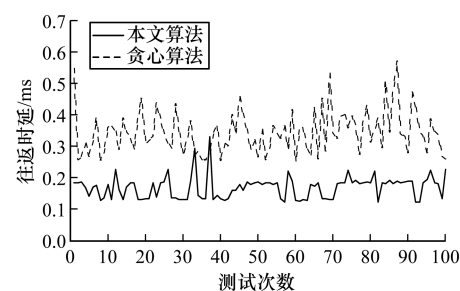
(1)类似 OpenStack 等 IaaS 中^[18],网络中虚拟机的初始放置位置采用随机算法,虚拟机被分配到第一个满足条件的服务器上,由此使得每一个终端主机以等概率的方式与网络中其他主机进行通信,虚拟机与其他主机通信流量均匀分布在网络中。

(2)为模拟数据中心中基于集群通信的特点,对于某一终端主机来说,其50%的流量发生在与其连接在同一接入交换机或者汇聚交换机下的主机,而其余流量需要经过核心层交换机,与其他分支上的主机通信^[19]。

5.2 实验结果分析

因为虚拟机的迁移也会消耗部分系统资源,所以设置物理服务器上 CPU、内存和网络带宽的资源利用率阈值均为70%。在多个虚拟机上安装配置 apache 服务器,并在 Mininet 中随机选择若干 host 通过 httpperf 向虚拟机发出 http 请求。为了测试提出的算法,通过 httpperf 和 iperf 制造主机的工作负载。加快向 VM₁ 和 VM₂ 的 HTTP 请求速率,使主机 PM₁ 的 CPU 利用率很快达到70%,此时 PM₁ 的瓶颈资源就是 CPU。为可以尽快发现服务器负载过重现象,由管理服务器每隔5s收集一次物理主机的负载状态。在此种情境下,本文比较了本文提出的策略和只考

虑服务器性能的贪心选择策略,并在上述2种流量模式下测试迁移后虚拟机到与它通信的主机的平均往返时延。结果如图3和图4所示。

图3 第1种流量模式 VM₁ 迁移后的往返时延图4 第2种流量模式 VM₁ 迁移后的往返时延

贪心算法只考虑了目的主机的系统资源是否能够最大限度地满足虚拟机的资源需求,不能保证虚拟机的网络通信需求也能够得到保证。本文算法基于全网的资源视图,目的主机的选择考虑到网络负载状况。在第1种流量模式下,本文算法选择的目的地主机为 PM₃,综合考虑了迁移开销和虚拟机迁移后的通信开销,避免了发生拥塞的可能,所以,有较小的往返时延。同样在第2种流量模式下,本文的通信开销评估(式(5))综合考虑了跳数、时延和网络带宽限制以及迁移流量的网络传输开销,因此,将 VM₁ 迁移到 PM₂ 上,使更多的流量被限制在接入层和汇聚层,从而降低网络流传输时延,这样可确保系

统提供的服务质量和用户体验,使系统不会由于网络拥塞而影响其整体性能。

6 结束语

本文提出了 OpenFlow 网络下结合虚拟机迁移技术的服务器负载均衡策略。综合考虑目的主机端的系统资源开销、迁移虚拟机开销以及迁移后的网络通信开销,避免网络拥塞的发生,与贪心算法相比降低了虚拟机迁移后其上运行的服务与其他主机之间的通信时延。实验结果表明,该策略在通过虚拟机迁移实现服务器负载均衡的同时,在 2 种流量模式下相关网络流的平均往返时延低于只考虑系统资源的贪心算法,使计算任务得到更快的响应,从而保证用户的服务质量。但本文只在传统的树形网络下进行了实验,如何使本文策略在其他数据中心拓扑(如 FatTree, BCube 等)下根据各自特点得到更好的应用,是下一步研究的方向。

参考文献

- [1] 邓 罡, 龚正虎, 王 宏. 现代数据中心网络特征研究[J]. 计算机研究与发展, 2014, 51(2): 395-407.
- [2] Wood T, Shenoy P, Venkataranmani A, et al. Black-box and Gray-box Strategies for Virtual Machine Migration[C]//Proceedings of the 4th USENIX Symposium on Networked Systems Design & Implementation. [S. l.]: USENIX Association, 2007: 229-242.
- [3] Zhou Wenyu, Yang Shoubao, Fang Jun, et al. VMCTune: A Load Balancing Scheme for Virtual Machine Cluster Based on Dynamic Resource Allocation[C]//Proceedings of the 9th International Conference on Grid and Cloud Computing. Washington D. C., USA: IEEE Press, 2010: 81-86.
- [4] Buyya R, Murshed M. GridSim: A Toolkit for the Modeling and Simulation of Distributed Resource Management and Scheduling for Grid Computing[J]. Concurrency and Computation: Practice and Experience, 2002, 14(13-15): 1175-1220.
- [5] Greenberg A, Hamilton J R, Jain N, et al. VL2: A Scalable and Flexible Data Center Network [C]//Proceedings of ACM SIGCOMM Conference on Communication Review. Chicago, USA: ACM Press, 2009: 51-62.
- [6] Benson T, Akella A, Maltz D A. Network Traffic Characteristics of Data Centers in the Wild [C]//Proceedings of ACM SIGCOMM Internet Measurement Conference. Melbourne, Australia: ACM Press, 2010: 267-280.
- [7] Meng Xiaoqiao, Vasileios P, Zhang Li. Improving the Scalability of Data Center Networks with Traffic-aware Virtual Machine Placement [C]//Proceedings of IEEE INFOCOM' 10. San Diego, USA: IEEE Press, 2010: 1154-1162.
- [8] Handigol N, Seetharaman S, Flajsik M, et al. Plug-n-Server: Load-balancing Web Traffic Using OpenFlow [C]//Proceedings of ACM SIGCOMM' 09. Chicago, USA: ACM Press, 2009.
- [9] Uppal H, Brandon D. OpenFlow Based Load Balancing[Z]. 2010.
- [10] Marc K, Odej K. Multiple Service Load-balancing with OpenFlow [C]//Proceedings of IEEE Conference on High Performance Switching and Routing. Washington D. C., USA: IEEE Press, 2012: 210-214.
- [11] Wang R, Butnariu D, Rexford J. OpenFlow-based Server Load Balancing Gone Wild [C]//Proceedings of the 11th USENIX Conference on Hot Topics in Management of Internet, Cloud, and Enterprise Networks and Services. [S. l.]: USENIX Association, 2011: 12.
- [12] 尚志浩. OpenFlow 网络中服务器负载均衡的研究[D]. 兰州: 兰州大学, 2014.
- [13] 邓 罡, 龚正虎, 王 宏, 等. 现代数据中心网络资源管理技术分析综述[J]. 通信学报, 2014, 35(2): 166-181.
- [14] Open Flow Switch Specification Version 1.0.0 [EB/OL]. [2015-01-15]. <http://archive.openflow.org/documents/openflow-spec-v1.0.0.pdf>.
- [15] Sundaram V, Wood T, Shenoy P. Efficient Data Migration in Self-managing Storage Systems [C]//Proceedings of the 3rd International Conference on Autonomic Computing. Dublin, Ireland: [s. n.], 2006: 297-300.
- [16] 刘 欣, 李 飞, 郑宝玉. 基于量子遗传算法的多约束 QoS 路由算法[J]. 南京邮电大学学报: 自然科学版, 2011, 31(2): 31-35.
- [17] Mann V, Gupta A, Dutta P, et al. Remedy: Network-aware Steady State VM Management for Data Centers [C]//Proceedings of the 11th International IFIP TC 6 Networking Conference. Berlin, Germany: Springer-Verlag, 2012: 190-204.
- [18] 胡元元, 林 浒, 李鸿彬. IaaS 云中最小迁移代价的虚拟机放置算法[J]. 小型微型计算机系统, 2014, (4): 878-882.
- [19] Kandula S, Sengupta S, Greenberg A, et al. The Nature of Data Center Traffic: Measurements & Analysis [C]//Proceedings of the 9th ACM SIGCOMM Conference on Internet Measurement Conference. New York, USA: ACM Press, 2009: 202-208.

编辑 金胡考