文章编号:1674-2974(2016)04-0120-13

基于网络感知的两阶段虚拟机分配算法*

陈 磊[1],章 兢[1],蔡立军[2†],孟 涛[2],何庭钦[2]
(1.湖南大学 电气与信息工程学院,湖南 长沙 410082;
2.湖南大学 信息科学与工程学院,湖南 长沙 410082)

摘 要:提出了一种基于网络感知的两阶段虚拟机分配算法(NWTP).首先,针对现代数据中心网络拓扑的随机性(树形、服务器和光纤混合),根据交互对象的不同,将虚拟机的带宽请求分为网内带宽和网间带宽两种.其次,将虚拟机的分配过程分解成带宽区域划分和物理主机分配两个彼此连续的阶段,建立网络感知模型.然后,利用流水线技术将带宽区域划分和物理主机分配看作两个连续的工序,并发进行分配处理.在带宽区域划分环节,利用节点介数和聚集系数动态感知物理主机的稳定性,通过差异化的分配策略为虚拟机子集选择合适的物理主机区域.在物理主机分配环节,将更多的虚拟机分配到负载方差最大的物理主机上,提高虚拟机网内带宽的节约度,均衡物理主机的资源负载.最后,对NWTP,遗传GA,模拟退火SA,贪婪GR四种算法进行大量的仿真实验,从分配时间、延迟、吞吐率、CPU利用率、带宽利用率和物理主机使用情况六个方面验证了NWTP算法的性能.

关键词:流水线技术;数据中心;云计算;网络感知;虚拟机分配
中图分类号:TP391 **文献标识码**:A

A Network-aware Two-phase Virtual Machine Allocation Algorithm

CHEN Lei[1], ZHANG Jing[1], CAI Li-jun[2†], MENG Tao[2], HE Ting-qin[2]
(1. College of Electrical and Information Engineering, Hunan Univ, Changsha, Hunan 410082, China;
2. College of Information Science and Engineering, Hunan Univ, Changsha, Hunan 410082, China)

Abstract: This paper proposed a two-phase network-aware virtual machine allocation algorithm (NWTP). Firstly, based on the randomness (the mixing of tree topology, servers and fiber) of modern data center network topology, the bandwidth requests of virtual machines were divided into two types of inner-bandwidth and outer-bandwidth according to the difference of interactive objects. Secondly, the virtual machine allocation requests were decomposed into two successive phases, and the corresponding network-aware model was established. Thirdly, pipelining technology was developed to regard the bandwidth region partitioning and physical host allocation as two consecutive steps, and to conduct parallel process-

* **收稿日期**:2015-09-24
基金项目:国家自然科学基金资助项目(61174140, 61472127, 61272395),National Natural Science Foundation of China(61174140, 61472127, 61272395);中国博士后科学基金(2013M540628,2014T70767);湖南省自然科学基金资助项目(14JJ3107);湖南省教育厅科研优秀青年项目(15B087);国家科技支撑计划课题项目(2012BAH09B02);长沙市重点科技计划项目(K1306004-11-1,K1204006-11-1,K1112001-11)
作者简介:陈 磊(1986—),男,四川眉山人,湖南大学博士研究生
† 通讯联系人,E-mail:ljcai@hnu.edu.cn

ing. For bandwidth region partitioning phase, the stability of physical host was dynamically perceived by using node betweenness and clustering coefficient, addressing differential allocation strategy to select the appropriate physical host set for the virtual machines subset. For the physical host allocation phase, more virtual machines were allocated to physical host resources with the largest load variance, which saved inner-bandwidth of data center and balanced physical hosts load. Finally, by comparing with the genetic algorithm GA, the annealing algorithm SA and greedy algorithm GR were simulated. Numerical results have shown the performance of NWTP in allocation time, network delay, network throughput rate, CPU utilization, bandwidth utilization and physical host usage.

Key words: pipelining; data center; cloud computing; network-aware; virtual machine allocation

数据中心[1]是成百上千台物理主机、存储通过网络设备彼此互连，利用互联网向社会提供计算和服务的场所.传统的数据中心存在租用成本高、地域限制等缺陷,仅能为大型企业和政府机构提供计算或存储服务.随着云计算[2]的不断发展和演变,按需付费的商业模式和虚拟机技术给传统数据中心带来了前所未有的机遇.传统数据中心与云计算结合的“云数据中心”应运而生[3-5].用户能够通过按需付费的模式,利用 Internet 轻松租用各种不同类型的虚拟机,进行复杂计算,并支付相应的费用.“云数据中心”已经由奢侈服务转向了大众服务,给社会发展提供了强大的动力.

云数据中心内,虚拟机的分配和调度[6]是一个长期的挑战,是影响数据中心性能的关键.好的虚拟机分配策略能够提高底层物理资源的利用率,降低成本,增加收益,给用户带来良好的体验和帮助.反之,则可能增加数据中心运营成本,甚至导致整个数据中心的崩溃.在云数据中心中,所有的虚拟机共享底层物理主机资源.因此,虚拟机分配的核心目标就是提高底层物理资源的利用率,增加收益,利用有限的资源向更多用户提供高效稳定的服务.在各种服务中,无论是计算服务、存储服务、容灾备份服务,都需要充足的网络带宽的保障.网络是云数据中心的核心资源,是连接用户和其他资源的桥梁.在云数据中心内,所有的物理资源(物理主机、存储等)通过网络设备彼此相连,共同组成了网络拓扑.在网络拓扑中,一台物理主机通过网络链路与多台设备进行连接.网络带宽是衡量物理设备间链路能力强弱的标准,是物理设备通信能力的表示.数据中心内,网络带宽大小非常有限.因此,充分利用网络资源,提高网络带宽利用率,是虚拟机分配的重中之重.

近年来数据中心的虚拟机分配问题已经被广泛研究,并取得了较好的成果.成果主要分为物理资源优化、能耗优化和带宽优化三个方面.①大多数学者将研究重点集中于物理资源(CPU、内存和存储)的分配,通过提高物理资源的利用率,增加数据中心的收益.Sindelar 等人[7]根据多个虚拟机共享底层物理主机内存资源的特性,提出了一种内存共享感知的虚拟机分配算法,通过建立层次树和聚类树结构存储虚拟机间相同的内存内容,避免相同内容的反复存储,提升内存资源的利用率.文献[8]中提出了 Best-fit 和 Worst-fit 两种分配策略,最大化底层物理资源的利用率.文献[9]利用多目标思想同时考虑资源利用率和多维资源负载,进一步优化了物理资源的分配.以上工作都将目标着眼于物理主机资源,对网络带宽的研究较少.②部分学者试图在虚拟机分配过程中进行能源优化.文献[10－11]通过遗传算法和启发式算法,在考虑能源优化的同时进行虚拟机的分配,降低了底层物理主机的使用数量,节约了能耗.然而,能耗优化工作大多尝试着减少物理主机的使用数量,并未考虑网络带宽的性能.③此外,也有部分学者开始考虑网络带宽资源的分配.R Wang 等人[12]通过观察网络带宽的峰值变化,提出了一种基于带宽估计的网络感知虚拟机分配算法,利用经验估计优化带宽分配.Y Zhao 等人[13]将网络拓扑和虚拟机分配结合,建立混合整数规划模型,利用拉格朗日松弛分解进行虚拟机和带宽分配,提升了带宽的利用率.然而,这些工作仅仅考虑了网络带宽的分配,并未对物理资源的分配做过多的研究.

针对目前带宽分配研究不够深入的现状,结合网络带宽在虚拟机分配中的关键作用,同时优化带宽分配和物理资源分配成为了本文的动机.首先,本文根据数据中心带宽作用对象,将网络带宽分为网内带宽和网间带宽两种.其次,介绍了网络感知的相关知识,分析了云数据中心内网络拓扑结构的随机特性,利用节点介数和聚集系数来感知网络拓扑中

网络元素的重要性.然后,通过分配资源的不同将整个分配过程分解为带宽区域划分和物理主机分配两个彼此连续的阶段,建立相应的网络感知模型.再次,提出了一种基于网络感知的两阶段虚拟机分配算法 NWTP.利用流水线技术模拟整个资源分配过程,将带宽区域划分和物理主机分配看作流水线上的两个连续环节,配合错误处理,实现并发资源的分配.最后,在 cloudsim 平台进行大量实验,将 NWTP 算法与贪婪(GR)、模拟退火(SA)、遗传(GA)等算法进行对比,体现了算法性能.本文的主要贡献如下:

1)根据作用对象,将用户带宽需求分为网内带宽和网间带宽两种.网内带宽是指多个虚拟机间相互进行数据通信所造成的带宽.网间带宽是指用户通过 internet 与虚拟机进行数据通信所需要的带宽.

2)根据分配资源的不同,将虚拟机的分配过程分解为带宽区划划分和物理主机分配两个彼此连续的阶段,将复杂的虚拟机分配问题简单化.

3)提出了一种基于网络感知的两阶段分配算法 NWTP.利用流水线技术模拟虚拟机的分配过程,将带宽区域划分和物理主机分配看作流水线中两个彼此连续的环节.在整个流水线中,不同环节采用不同的分配策略,配合错误处理,实现虚拟机请求的并行分配.

4)在带宽区域划分环节,利用节点介数和聚集系统感知网络元素在网络拓扑的中邻居亲密性和网络枢纽性.将不同的虚拟机请求按照带宽类型进行分组,不同分组采用不同的分配策略.高网间带宽虚拟机组将分配到网络拓扑中重要的枢纽节点上,保证用户交互快速性和可靠性;高网内带宽的虚拟机组将分配到边界网络区域中,保证彼此交互的虚拟机间拥有区域集中性,减少通信距离,降低网络突发流造成的拥塞和影响;均衡的虚拟机组将结合前两种的优点,区域化的分配到网络拓扑中较重要的节点上,保证对外通信和对内通信的快速性.

5)在物理主机分配环节,针对固定的虚拟机组和物理主机区域,以节约网内带宽通信、均衡区域内物理主机带宽负载为目标,进行虚拟机到物理主机的映射,尽可能多地将虚拟机分配到同一物理主机.

本文后续工作的组织如下:第 1 节进行网络感知的介绍,对整个分配过程进行建模;基于网络感知的两阶段虚拟机分配算法(NWTP)在第 2 节被设计;第 3 节进行实验对比,验证 NWTP 算法的性能;第 4 节总结全文.

1 基于网络感知的分配模型

首先,本节介绍了网络感知的定义和核心知识;然后,对网络感知的虚拟机分配过程进行了公式化描述;最后,将整个分配过程分解成带宽区划划分和物理主机分配两个彼此连续的阶段,并进行相应的建模.

1.1 网络感知

网络感知是对整个网络所有元素(网络拓扑、路由协议、网络设备等)使用情况的实时监控,对网络流量动态变化的预防和处理.简单而言,网络感知就是恰当的分配网络元素的剩余能力,合理的应对和处理网络突发情况.本文的网络感知主要监控两个重点和应对一种突发情况.

1)网络拓扑.网络拓扑是网络感知监控的第一个重点,所有的虚拟机分配请求都将映射到网络拓扑中的一个物理主机上,共享物理主机的底层资源,向用户提供服务.因此,网络拓扑是合理分配虚拟机,充分利用网络带宽资源和底层物理资源的重要参考.传统数据中心都采用树形的网络拓扑结构,如图 1 所示.树形网络拓扑一般分为三层:核心层(core)、聚合层(aggregation)和接入层(access).大量的服务器则为整个网络拓扑的叶子节点,通过接入层交换机彼此相连.在三层网络拓扑中,核心层的能力最强,聚合层次之,接入层最弱.常用的树形网络拓扑有 Fat-Tree,portland,VL2[5].图 2 为典型的以服务器中心的网络拓扑,其他常见的服务器为中心的网络拓扑有 Bcube,Dcell,Ficonn[5].

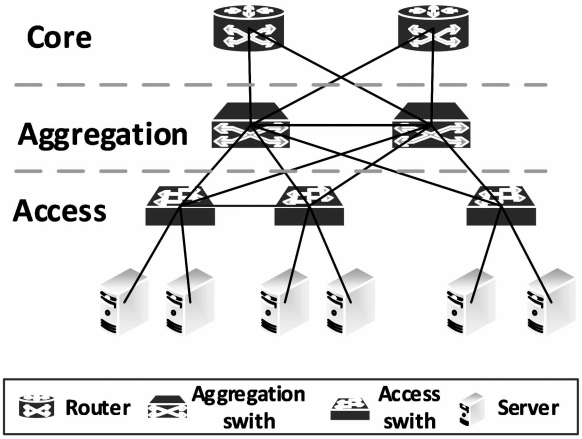

图 1 数据中心典型树形网络拓扑

Fig. 1 A typical tree network topology of data center

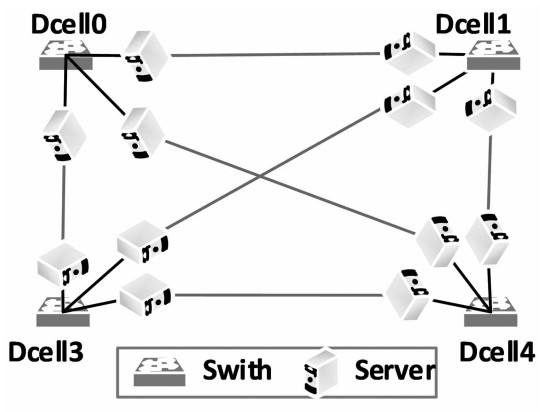

图 2　数据中心典型服务器网络拓扑
Fig. 2　A typical server-centric topology of data center

2)网络元素的重要性. 数据中心网络拓扑由许多的网络设备(交换机、路由器等)将大量物理服务器连接而成. 每个网络元素(交换机、路由器和服务器)在拓扑中起着不同的作用,拥有不同的重要性. 根据网络元素的重要性和能力,动态地进行资源分配,保证核心元素的强大服务能力,提升普通元素的平均利用率,对虚拟机的分配过程至关重要. 例如,在三层树形结构的网络拓扑中,接入层交换机重要性最差,聚合层次之,核心层最大. 在进行资源分配时,必须保证核心层交换机的强大服务能力和扩展性,尽量少地占用其带宽. 因为,核心层交换机出现故障或拥塞,必然导致整个数据中心的瘫痪和崩溃. 相反,接入层交换机仅仅影响部分物理主机,所以应该更加充分地利用接入层的网络带宽,提升网络带宽的利用率.

3)网络突发流. 网络突发流是指网络中突然出现超过正常大小的带宽请求,造成网络拥塞和延迟的现象. 在数据中心中,经常出现网络突发流情况. 如何预防网络突发流,减少网络突发流对整个数据中心的影响,缩小拥塞范围,是虚拟机分配的关键问题.

随着数据中心规模的不断扩大,地理位置的不断分散,各种数据中心拓扑结构不断涌现. 传统数据中心的树形拓扑,如图 1,已经不能满足需求. 以服务器为中心、以光纤为中心[5]的网络拓扑越来越多,如图 2. 因此,目前数据中心的网络拓扑为不完全规则的半随机拓扑结构,集合了树形、服务器、光纤等多种拓扑结构,是一种混合的拓扑结构,给虚拟机的分配带来了新的困难.

1.1.1　节点介数和聚集系数

在现代的拓扑结构中,网络元素在拓扑中的重要性更加明显,本文引入了聚集系数和节点介数两个参数来衡量网络元素的重要性.

网络拓扑可看作一个无向图 $G=\langle V,E\rangle$,所有的网络元素(交换机、服务器)可看作节点. 假设节点数量为 N,边数量为 M. e_{ij} 表示节点 i 与节点 j 之间的连接边. $V=\{i\,|\,i\in N\}$ 是所有节点的集合,$E=\{e_{ij}\,|\,i,j\in N$ 且 $i\neq j\}$ 是所有边的集合.

定义 1　节点邻居和节点度:节点邻居是指与网络节点 i 直接相连的节点集合. 定义如下:

$$N_i=\{j\,|\,j\in N\text{ 且 }e_{ij}\in E\}\tag{1}$$

节点 i 的度 k_i 就是节点的邻居个数.

定义 2　聚集系数:聚集系数是网络聚集特性的度量指标[5],表示节点与邻居间的亲疏程度. 在网络拓扑中,第 i 个网络节点的聚集系数 C_i 定义如下:

$$\begin{cases}C_i=\dfrac{E_i}{T_i},i\in N\text{ 且 }C_i\in[0,1];\\ T_i=\dfrac{k_i(k_i-1)}{2}\end{cases}\tag{2}$$

在公式(2)中,节点的度为 K_i,表示有 K_i 个邻居,E_i 表示节点 i 与 K_i 个邻居间的实际连接边的数量,T_i 表示节点 i 与 K_i 个邻居间可能形成的最大连接边的数量. 当 $C_i=1$ 时,表示节点 i 与所有邻居节点互相连接,形成全局耦合网络,连接非常紧密.

定义 3　节点介数(node betweenness):节点介数是网络拓扑的一个重要度量[5]. 节点介数是衡量网络拓扑中通过该网络节点的最短路径数量,反应了节点在网络流通中的枢纽性. 节点介数 B_i 的定义如下:

$$B_i=\frac{\sum\limits_{(s,t\in N)}\dfrac{d_{sit}}{d_{st}}}{N(N-1)/2},i\in N\text{ 且 }B_i\in[0,1]\tag{3}$$

在公式(3)中,d_{st} 表示节点 s 与节点 t 间的最短路径数量,d_{sit} 表示过节点 i 的节点 s 与节点 t 间的最短路径数量,$N(N-1)/2$ 表示节点 s 与节点 t 的所有可能组合数. 节点介数越大说明节点在网络中的枢纽性越强,删除节点(节点故障)会造成大量节点对间的最短路径变长,从而增加整个网络的压力,造成网络拥塞,甚至是网络崩溃.

节点聚集系数和节点介数能够很好地反应网络元素在整个网络拓扑中的重要程度,聚集系数体现了节点与邻居间的亲密程度,介数反应了节点对整个网络流的枢纽性. 通过聚集系数和节点介数,能够轻松感知整个网络的资源情况和动态变化,为虚拟机的分配提供了强有力的支撑.

1.1.2　网内带宽和网间带宽

通过仔细分析数据中心的网络带宽使用情况,

根据使用对象的不同,可以将网络带宽分成两种:网间带宽和网内带宽.

1)网间带宽.网间带宽是指数据中心的外部用户通过互联网获取虚拟机中计算数据所造成的带宽,也就是用户与虚拟机间的通信所产生的网络流量.网间带宽必须通过核心交换机与外界用户进行交互,需要占用整个网络拓扑的骨干网络资源.网间带宽是用户服务的基础,是保证服务 SLA 等级的关键.因此,网间带宽应该尽量接近核心交换机,更快速地与用户进行交互.

2)网内带宽.网内带宽是指用户的多台虚拟机间进行数据通信所造成的带宽.当用户的多台虚拟机为流量密集型虚拟机时,则虚拟机间需要频繁快速地进行数据交互,保证虚拟机间的及时通信和运行.网内带宽可能占用过多网络资源,造成网络突发流的现象,导致整个数据中心网络的崩溃.根据网内带宽的内部通信特性,多个相互通信的虚拟机应该尽快靠近,保证较小的通信距离.此外,为了避免网内带宽造成的网络突发流问题,应该将流量密集型虚拟机分配到边界节点上,避免核心节点和骨干网络受到影响,约束网络突发流造成的破坏.

网间带宽与网内带宽的分解使得网络感知能够更加精确地进行虚拟机分配,保证整个数据中心网络的可靠和稳定,减少网络突发流的产生和严重的破坏.

1.2 分配模型

同时考虑带宽资源和物理资源的虚拟机分配过程可公式化为:多个用户提交 P 个虚拟机请求 $RS=\{r_i,i\in P\}$ 到数据中心.每个虚拟机请求 $r_i=\langle CPU_i,MEM_i,BWout_i,BWin_i,User_i\rangle$ 包含 CPU、MEM、网间带宽 $BWout$ 和网内带宽 $BWin$ 四种需求,其中 $User$ 是虚拟机的用户属性.数据中心由 M 个物理主机构成主机集 $SS=\{s_i,i\in M\}$.每个物理主机 $s_i=\langle\begin{bmatrix}C\_cpu_j\\Z\_cpu_j\end{bmatrix},\begin{bmatrix}C\_mem_j\\Z\_mem_j\end{bmatrix},\begin{bmatrix}C\_bw_j\\Z\_bw_j\end{bmatrix}\rangle$ 包含三组属性信息,分别是 CPU 能力和剩余 CPU 能力、MEM 大小和剩余 MEM 大小、带宽大小和剩余带宽大小.数据中心的所有物理主机和网络设备构成一个无向图 $G=\langle V,E\rangle$,所有的网络元素(交换机、服务器)都是图 G 中的节点.e_{ij} 表示节点 i 与节点 j 之间相互连接.$V=\{i|i\in N\}$ 是 G 中所有节点的集合,$E=\{e_{ij}\,|i,j\in N$ 且 $i\neq j\}$ 是 G 中所有边的集合.

在虚拟机的分配过程中,每个虚拟机 r_i 需要映射到数据中心的一台物理主机 s_i 上.物理主机 s_i 的 CPU,MEM 和 BW 剩余资源必须大于虚拟机 r_i 的请求大小.一个虚拟机只能映射到一台物理主机上.同一物理主机上的多个虚拟机共享物理主机资源(CPU,MEM 和 BW).每个用户可能有多个彼此通信的虚拟机.数据中心的物理主机是异构的,存在不同 CPU、MEM 和 BW 大小的物理主机.网络拓扑图 G 中的所有交换机、路由器等网络设备不具有计算能力,只有通信能力,不能将虚拟机映射到网络设备上.

1.3 两阶段分配目标

基于网络感知的虚拟机分配过程同时考虑带宽资源和物理资源.当多个用户提交一组虚拟机请求时,其分配过程就是在网络拓扑中选择物理主机进行映射的过程.整个过程可以简单地分为两个连续的阶段.第一个阶段,根据一组虚拟机的总体请求带宽大小,在网络拓扑中选择满足带宽要求一块合适的区域,保证选择区域的稳定性和可靠性,降低网络突发流的产生概率.第二个阶段,在选择的区域中,根据用户需求寻找最合适的物理主机进行分配,保证物理主机的资源利用率和负载.因此,本文将整个网络感知的虚拟机分配过程分解成带宽区域划分和物理主机分配两个阶段.

1.3.1 带宽区域划分

带宽区域划分阶段主要目标是为一组虚拟机集合 $RS=\{r_i,i\in P\}$ 中的 P 台虚拟机,根据虚拟机类型的不同,选择多个物理主机区域,保证每个区域内的虚拟机拥有较好的网络性能,避免和减少网络突发流的产生和带来的影响.

在带宽区域划分阶段,首先需要按类型对虚拟机进行分组.虚拟机类型 $Type\_BW$ 是指虚拟机 r_i 的带宽请求对邻居依赖性和网络枢纽性的侧重程度,其具体定义如下:

$$\begin{cases}Type\_BW_i=\dfrac{BWin_i}{BWout_i},i\in P\\ Type\_BW_i<\alpha \quad (1)\\ \alpha\leqslant Type\_BW_i\leqslant\beta \quad (2)\\ Type\_BW_i>\beta \quad (3)\end{cases}\tag{4}$$

在公式(4)中,虚拟机类型由虚拟机的网间带宽和网内带宽需求大小决定.本文将虚拟机分成了网间虚拟机(1)、网内虚拟机(3)和均衡虚拟机(2)三种.其中,α 和 β 是两个常数,表示网间带宽与网内带宽大小的比值.根据 3.2 节实验测试,本文默认为(1/2 和 2).

在虚拟机类型确定以后,不同类型虚拟机分配到不同物理主机,将对整个网络拓扑的稳定性造成不同程度的影响.例如,当网内流量高的多个虚拟机分配到核心网络节点上,必然占用核心节点带宽资源,导致其他节点的通信延迟,甚至发生网络突发流.当网间流量高的虚拟机分配到边缘网络节点时,则增加了通信距离,降低了响应时间(QOS),导致整个网络的不稳定.因此,本文使用稳定系数 R_C (reliable coefficient)来描述网络拓扑的稳定,定义如下:

$$R\_C_{ij}=\begin{cases}|B_j-C_j|, & Type\_BW_i=(1)\\ 1-|B_j-C_j|, & Type\_BW_i=(2)\\ |B_j-C_j|, & Type\_BW_i=(3)\end{cases}\begin{array}{l}|\ j\in M\\ |\ i\in P\end{array} \tag{5}$$

从公式(5)中可以看出,不同类型虚拟机映射到不同的物理主机上,其稳定系数的产生方式不同.对于网间虚拟机而言,带宽和 QOS 需求较高,应映射到靠近核心的枢纽节点,且周围邻居应越稀疏,从而保证经过的路径越短,交互速度越快,对邻居的影响越小;对于网内虚拟机而言,主要是虚拟机间的彼此通信,应该映射到边界节点上,使得多个通信虚拟机彼此靠近.同时,远离核心节点,减少对整个网络的过多影响;对于均衡虚拟机而言,应该选择网络拓扑的中间区域,平衡节点枢纽性和邻居依赖性,约束网络突发流的影响,保证整个网络的稳定性.

因此,带宽区域划分阶段的目标就是将 P 台虚拟机映射到 M 台物理主机构成的多个区域内,实现网络稳定性的最大化,其目标模型如下:

$$\max \sum_{i=1}^{P}\sum_{j=1}^{M} x_{ij} * R\_C_{ij} \tag{6}$$

$$\text{st.}\begin{cases} x_{ij}=\begin{cases}1, & r_i \xrightarrow{\text{allocation}} s_j\\ 0, & r_i \xrightarrow{\text{not allocation}} s_j\end{cases}\\ \sum_{j=1}^{M} x_{ij}=1\\ \text{BWin}_i+\text{BWout}_i \leqslant Z\_bw_j, x_{ij}=1\\ CPU_i \leqslant Z\_cpu_j, MEM_i \leqslant Z\_mem_j, x_{ij}=1\end{cases}$$

在目标模型(6)中,x_{ij} 是一个二进制变量,表示虚拟机 i 是否分配到物理主机 j 上.每个虚拟机只能映射到一台物理主机.

1.3.2　物理主机分配

通过带宽区域划分阶段后,P 个虚拟机集合 $RS=\{r_i, i\in P\}$ 将被映射到 M 台物理主机组成的区域 KS 中.物理主机分配的核心目标就是降低虚拟机的网内带宽的消耗,均衡 KS 区域内所有物理主机的带宽负载.

在 P 个虚拟机集合 RS 中,同一用户的多个虚拟机属于相同的类型,且多个虚拟机间通常彼此通信,需要较大的网内带宽.若将多个虚拟机映射到同一物理主机,可大大节约数据中心带宽的使用.因为,同一物理主机内,多个虚拟机的通信不需要占用带宽.为了描述这种网内带宽的节约度,本文定义了同用户同物理主机的虚拟机分配集,表示分配到同一物理主机上的相同用户的虚拟机请求集合,定义如下:

$$\begin{cases} US_{uj}=\{r_i \mid x_{ij}=1, User_i=u, i\in P, u\in U\}\\ x_{ij}=\begin{cases}1, & r_i \xrightarrow{\text{allocation}} s_j\\ 0, & r_i \xrightarrow{\text{not allocation}} s_j\end{cases}\end{cases} \tag{7}$$

根据公式(7),u 表示虚拟机 i 的用户属性,U 表示 RS 中的用户属性数量.x_{ij} 表示虚拟机 i 是否分配到物理主机 j.根据公式(7)可得,物理主机 j 的网内带宽节约度为:

$$S\_BW_j=\sum_{u=1}^{U}\frac{\sum_{r_i\in US_{uj}} BWin_i}{\|US_{uj}\|} * \frac{1}{\sum_{i=1}^{P} x_{ij}(BWin_i+BWout_i)} \tag{8}$$

在公式(8)中,网内带宽节约度就是相同用户的网内带宽节约均值之和与总物理主机消耗带宽的比值,其值域为[0,1).

此外,在节约网内带宽的同时,需要尽量均衡各个物理主机的带宽负载,保证网络突发流情况下所有物理主机拥有相同的抵御程度和扩展性,从而提升数据中心网络的可靠性.因此,本文定义了带宽负载方差,衡量主机区域的各物理主机的带宽负载情况,物理主机 j 的带宽负载方差如下:

$$\begin{cases} LV_j=\left(\frac{1}{LV_{\max}} * \left(\frac{Z\_bw_j-\sum_{i=1}^{P} x_{ij} * (BWin_i+BWout_i)}{C\_bw_j}-\overline{LV}\right)\right)^2\\ \overline{LV}=\frac{1}{M} * \sum_{j=1}^{M}\frac{Z\_bw_j-\sum_{i=1}^{P} x_{ij} * (BWin_i+BWout_i)}{C\_bw_j}\\ LV_{\max}=\max\left(\left.\frac{Z\_bw_j-\sum_{i=1}^{P} x_{ij} * (BWin_i+BWout_i)}{C\_bw_j}\right| j\in M\right)\end{cases} \tag{9}$$

在公式(9)中,物理主机负载方差的值域同样为[0,1).

因此,物理主机分配阶段的目标模型是最大化网内带宽节约度和最小化负载方差,定义如下:

$$\min \sum_{j=1}^{M} LV_j - S\_BW_j \tag{10}$$

$$\text{st.}\begin{cases} \sum_{j=1}^{M} x_{ij} = 1 \\ BWin_i + BWout_i \leqslant Z\_bw_j, x_{ij} = 1 \\ MEM_i \leqslant Z\_mem_j, x_{ij} = 1 \\ CPU_i \leqslant Z\_cpu_j, x_{ij} = 1 \end{cases}$$

2 基于网络感知的两阶段分配算法

根据上节的分配模型,本节提出了一种基于网络感知的两阶段虚拟机分配算法,简称 NWTP 算法. NWTP 算法将网络感知的虚拟机分配过程看作一个流水线作业过程. 虚拟机请求集合为流水线的加工原料,带宽区域划分和物理主机分配为流水线的两个连续工序(环节). 2.1 节将详细介绍带宽区域划分的流程和策略,为虚拟机子集选择物理主机区域. 2.2 节将介绍物理主机分配的流程和相应策略,更多地节约网内带宽和均衡区域主机带宽负载. 2.3 节将整体介绍 NWTP 算法的详细流程.

2.1 带宽区域划分

在多个用户提交的虚拟机请求集合 RS 中,根据虚拟机类型和用户属性,可将 RS 分解成多个虚拟机子集. 带宽区域划分的目标就是根据每个虚拟机子集的虚拟机类型,在网络拓扑中选择合适的物理主机区域. 在虚拟机子集中,所有虚拟机都拥有相同的类型和相同的用户属性. 一次带宽区域划分只为一个虚拟机子集进行主机区域选择. 此外,为了保证物理主机分配的成功率,带宽区域划分中的物理主机数量将会多于虚拟机子集的虚拟机数量. 本文默认物理主机数量为虚拟机数量的 1.2 倍,可根据需要进行调整.

根据虚拟机子集内虚拟机的类型,可将虚拟机子集分成网内子集、网间子集和均衡子集三类. 三类不同的子集,其区域选择的目标是不同的、动态变化的.

1)网间子集. 网间子集内的虚拟机都是网间带宽远大于网内带宽的虚拟机集合. 这类虚拟机的目标是对外与用户交互,保证交互质量,提升交互速度是虚拟机分配的要点. 在 NWTP 算法中,网间子集虚拟机将映射到具有核心枢纽性的物理主机上. 根据节点介数的定义,核心物理主机必然在通往外网的核心路径上. 这种分配方式,能够大大减少用户与物理主机交互的通信距离,降低网络成本,增加网络带宽的资源利用率. 当网间流量虚拟机出现突发流时,会造成核心节点的延迟和拥塞. 但是,由于网内流量的边界化分配策略,核心节点的带宽能力将会最大化,扩展性较强,受到突发流的影响也会越小. 同样,网内流量虚拟机被分配到边界区域的节点和链路,且节点间交互彼此靠近,核心节点的突发流不会过多影响网内流量的虚拟机.

2)网内子集. 网内子集的所有虚拟机网内交互带宽远大于网间通信. 这类虚拟机分配的目标就是彼此尽量靠近,减少通信距离. 在 NWTP 算法中,网内子集的分配策略为:将虚拟机集合分配到网络拓扑的边界节点区域,增加边界节点的资源利用率,减少虚拟机对网络拓扑中枢纽节点的消耗. 此外,通过区域性的分配,能充分保证虚拟机间的通信距离,减少通信跳数,提高通信速率.

当网内子集出现突发流时,由于分配策略的边界化,网间虚拟机受到的影响较少. 因为,突发流发生在网络拓扑的边界,其蔓延的范围有限,不会过多蔓延到网络拓扑中的核心枢纽节点. 边界区域化分配方式增加了数据中心网络的扩展性. 此外,对于网内虚拟机而言,由于其分配策略的区域化,网络突发流的发生将被限制在固定区域内,造成区域内的延迟和拥塞,影响区域较小,不会受到其他虚拟机的干扰,突发流的消亡周期会大大缩减. 此外,通过边界区域化的虚拟机分配,能够很好地平衡整个网络的负载,增加网络的稳定性.

3)均衡子集. 均衡子集内的虚拟机其网间带宽和网内带宽相对均衡. 这类虚拟机应该同时兼顾网内子集和网间子集的分配策略. 在 NWTP 算法中,均衡子集的虚拟机将被区域化地分配到网络拓扑的中间区域. 在保证用户对外交互的同时,区域化地捆绑多个虚拟机节点,减少虚拟机间的通信距离.

图 3 展示了不同虚拟机子集的不同带宽区域的选择策略. 整个网络拓扑可看作一个圆,圆的中心点为网络中的核心枢纽点. 整个拓扑圆以核心点为中心,可分为 3 个圈:核心圈、中间圈和边界圈,图中用蓝色虚线划分. 不同的虚拟机子集将划分到不同的拓扑圈内,从而均衡整个网络拓扑的负载,保证网络的稳定性和可靠性. 在图 3 中,网内子集将被分配到最外层边界圈内. 网内子集的虚拟机将分配到彼此邻居的物理主机上,缩小虚拟机间的通信距离;网间子集的虚拟机将被分配到最内层的核心圈内,以最

快的速度与外界用户进行交互;均衡子集的虚拟机则映射到中间圈内,结合前两种策略的优点,保证网络的扩展性.当某类型虚拟机出现突发流时,区域化的分配策略必然降低突发流的破坏性,减少虚拟机间的干扰性,保障网络的稳定性.

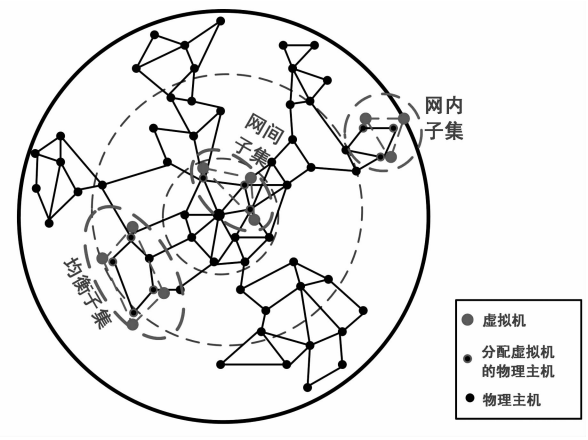

图 3　虚拟机子集选择策略示意图

Fig. 3　Virtual machine subset selection strategy

根据图 3 的示意,NWTP 中带宽区域划分的基本思想如下:根据虚拟机子集的类型(网间子集、网内子集和均衡子集),选择相应的划分策略.网间子集划分到更加接近核心枢纽节点的物理主机上,减少通信距离,增加交互速率;网内子集区域性地分配到边界网络节点上,充分利用边界节点能力,降低核心节点和带宽压力;网间子集划分到中间区域,均衡前两种策略的优点.算法 1 展示了带宽区域划分的主要流程.

算法 1　带宽区域划分过程

输入:虚拟机子集 RS_l,网络拓扑 G,稳定系数矩阵 RC_l.

输出:虚拟机子集和物理主机区域对〈RS_l, SS_l〉.

Step 1:初始化.

①初始化区域物理主机数量上限 SN,SN 是子集中虚拟机数量的 1.2 倍.

②初始化错误核心集 ES 为空集,ES 表示邻居区域内找不到足够物理主机数量(SN)的节点集合.

③初始化 SS_l,邻居集 N,合理邻居集 NR 为空集.

④初始化最小资源需求 MIN_CPU, MIN_MEM,MIN_BW,最小资源需求是所有虚拟机的资源请求最小值.

//在初始化过程中,稳定系数矩阵 RC_l 与虚拟机子集 RS_l 相对应,稳定系数矩阵是根据虚拟机子集类型按照公式(5)计算而来,并已经排好序.

Step 2:判断合理性.遍历稳定系数矩阵 RC_l,选择区域核心点 s_i,判断核心点的合理性.核心点不合理的判断标准如下:

$$\begin{cases} Z\_cpu_i < MIN\_CPU \\ Z\_mem_i < MIN\_MEM \\ Z\_bw_i < MIN\_BW \\ s_i \in ES \end{cases} \tag{11}$$

如果 s_i 不满足标准,将核心点 s_i 加入物理主机区域 SS_l 和 NR 集中,进入 step3.否则,s_i 属于错误核心集 ES,或 s_i 的剩余资源小于最小资源需求.将 s_i 加入错误核心集 ES.开始下一个核心点的判断.

Step 3:寻找 NR 的邻居点 $N(NR)$.根据公式(11)计算 $N(NR)$ 中合理的主机节点集合,并替换 NR.

①如果 NR 不为空,将所有主机加入到 SS_l 中,进入 step4.

②如果 NR 为空,则将 $N(NR)$ 加入错误集 ES.返回 Step2,判断下一核心点.

Step 4:判断 RC_l 是否遍历完成,且 SS_l 中物理主机的数量是否大于 SN.

①遍历未完成,且主机数量小于 SN.返回 step3,寻找更多的物理主机节点.

②遍历未完成,主机数量大于或等于 SN.终止程序,返回虚拟机子集和物理区域对〈RS_l, SS_l〉.

③RC_l 遍历完成.终止程序,返回虚拟机子集和物理区域对〈RS_l, $\varnothing$〉.

2.2　物理主机分配

物理主机分配环节是整个流水线的第二个环节,负责对带宽区域划分形成的虚拟机子集和物理区域对〈RS_l, SS_l〉进行主机映射.带宽区域划分和物理主机分配两个环节彼此连续.

相对带宽区域划分而言,物理主机的分配过程较简单.其核心目标就是最大化的虚拟机网内流量的节约度,均衡物理主机区域内的带宽负载,增加主机区域对网络突发流的免疫力,保证数据中心的稳定性和扩展性.物理主机分配的详细流程如算法 2 所示.

算法 2　物理主机分配过程输入:虚拟机子集和物理主机区域对〈RS_l, SS_l〉.

输出:未分配的虚拟机集合 $NoRS_l$.

Step 1:初始化.将 RS_l, SS_l 按网内带宽和剩余内存的大小进行降序排列.初始化未分配的虚拟机

集合 $NoRS_l$ 为空集.

Step 2:遍历 RS_l,分配虚拟机 r_i 到物理主机 s_i. 具体过程如下:

for r_i *in* RS_l *then*

//遍历虚拟机子集

for s_j *in* SS_l

then //遍历物理主机

//判断虚拟机的需求是否满足

$$if\begin{cases} Z\_bw_i \geqslant BWin_i + BWout_i \\ Z\_cpu_i \geqslant CPUin_i \quad is\ false \quad then \\ Z\_mem_i \geqslant MEMin_i \end{cases}$$

$j++$; *next* s_j; //不满足开始下一个主机判断

endif

$r_i \xrightarrow{allocation} s_j$ //分配虚拟机 i 到物理主机 j

$i++$; *next* r_i; //开始下一个虚拟机的分配

endfor

$r_i \rightarrow NoRS_l$;

$i++$; *next* r_i; //开始下一个虚拟机的分配

endfor

Step 3:返回未分配的虚拟机集合 $NoRS_l$.

在物理主机的分配过程中,虚拟机子集将首先分配到剩余资源多的物理主机,从而保证更多的虚拟机分配到同一物理主机上,增加网内流量的节约度公式(8). 此外,剩余资源最多的物理主机通常就是负载方差最大的主机,将更多的虚拟机分配到物理主机,可以均衡带宽负载公式(9).

2.3 NWTP 算法

带宽区域划分和物理主机分配是虚拟机分配过程的两个连续阶段. NWTP 算法将整个分配过程看作流水线处理过程,两个分配阶段就是流水线中两个连续的工序(环节). NWTP 算法的基本思想就是:根据公式(4)定义的虚拟机类型和用户属性,将虚拟机集合 RS 分解成 L 个虚拟机子集. 将每个虚拟机子集看作原材料,在两个连续的工序中进行加工,最终完成虚拟机的分配. NWTP 算法是一个并行处理算法,带宽区域的划分与物理主机的分配两个工序并行工作,加快了虚拟机分配的速度. 图 4 为 NWTP 算法的分配过程示意图.

从图 4 可以看出,NWTP 算法的流水线共有三个工序(环节),带宽区域划分、物理主机分配和错误处理. 三个环节间并行进行虚拟机的分配过程. 在 t_1 时刻,虚拟机子集 RS_1 开始进行带宽区域划分. t_2 时刻,虚拟机子集 RS_1 进行主机分配的同时,RS_2 开始进行带宽区域划分. 依次类推,直到所有虚拟机子集分配完成. 因此,NWTP 算法能够并行进行多个虚拟机子集的分配处理,加快分配速度.

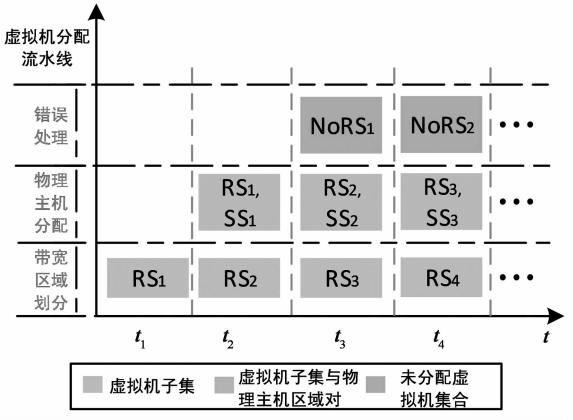

图 4 NWTP 算法分配示意图

Fig. 4 NWTP allocation process

在 NWTP 算法的整个流水线中,错误处理是非常重要的环节. 当带宽划分中虚拟机子集无法找到合适的物理主机时,当物理主机分配中某些虚拟机无法进行分配时,都要进行错误处理. 错误处理的流程如下:将未分配虚拟机集合,分割成两个虚拟机子集,重新投入流水线进程处理. 若虚拟机集合中只有一个虚拟机,则等待固定周期 T,再次进行虚拟机的分配. 因此,NWTP 的详细过程如算法 3 所示.

算法 3 NWTP 并发分配过程

输入:虚拟机集合 RS,网络拓扑 G.

输出:$NULL$.

Step 1:初始化. 根据网络拓扑 G,为每个物理主机节点生成稳定系数矩阵 RC. 在 RC 中,每个物理主机将有 3 种类型的稳定系数,因为不同类型的虚拟机分配物理主机上,其稳定系数是不同的.

Step 2:根据公式(4)中虚拟机的类型和用户属性,将 RS 分解成 L 个虚拟机子集集合 RSL.

Step 3:遍历 RSL,根据虚拟机的类型为每个虚拟机子集 RS_l 求得相应的 RC_l. 根据 RS_l,G,RC_l,按照算法 1 进行带宽区域的划分,获得虚拟子集与物理主机区域对$\langle RS_l, SS_l\rangle$. 将虚拟子集与物理主机区域对$\langle RS_l, SS_l\rangle$输入到 step4,并开始下一个虚拟机子集的带宽区域划分.

Step 4:根据输入的虚拟子集与物理主机区域对$\langle RS_l, SS_l\rangle$,按照算法 2,进行虚拟机到物理主机的分配. 获取未分配的虚拟机集合 $NoRS_l$. 将 $NoRS_l$ 输入到 step5,等待下一个输入,继续执行.

Step 5:判断 $NoRS_l$ 的虚拟机数量. 若大于 1,分

裂成两个虚拟机子集,重新加入 *RSL* 中等待分配.否则,等待周期 T,重新开始分配.

Step 6:当 *RSL* 中所有虚拟机分配完成,算法结束.

在 NWTP 算法中,通过流水线技术,并行多个阶段的处理,加快了处理效率,减少了算法的时间复杂度.NWTP 算法的时间复杂度为 $o(R*S)$,其中 R 是虚拟机集合中虚拟机数量,S 为网络拓扑中物理主机数量.

3　实验结果分析

本文利用云计算仿真平台 CloudSim3.5[14] 作为仿真工具,将 NWTP 算法与遗传算法 GA[17],贪婪算法 GR[15],模拟退火算法 SA[16] 进行对比,从分配时间、网络延迟、吞吐率、CPU 利用率、带宽利用率、主机使用情况等六个方面对 NWTP 算法进行验证.

3.1　实验环境

在 CloudSim 平台中,利用随机方式产生物理机资源和虚拟机分配请求,通过 DataCenter,Host,Vm,DataCenteBoker 等多个类,实现底层物理机和虚拟机的仿真.利用多线程思想优化了 CloudSim 仿真平台,实现了流水线式的虚拟机分配过程.在仿真环境中随机产生了 500 个物理主机,包含[〈2core－4g〉,〈4core－6g〉,〈4core－12g〉,〈8core－24g〉]四种类型,详细的物理主机参数见表 1.为了模拟网络拓扑的随机性,使用 brite 拓扑生成器随机生成 500 个节点的网络拓扑,包括边的带宽大小和延迟几率等信息.虚拟机同样采用随机策略生成,数量范围为[3～400].每个虚拟机的 CPU 需求从[1～4]随机生成,内存需求从[1～10]＊512M 随机生成,网间和网内带宽需求从[1～10]M 随机生成.

表 1　物理主机参数表
Tab.1　The physical parameter of hosts

类型	CPU 核数	内存数/G	数量
G1	2	4	220
G2	4	6	160
G3	4	12	90
G4	8	24	30

3.2　虚拟机类型门限参数测试

虚拟机类型是整个网络感知分配算法的基础,是虚拟机带宽请求对邻居依赖性和网络枢纽性的偏好.在公式(4)中,α 和 β 是两个虚拟机类型的门限参数,表示虚拟机网间带宽和网内带宽需求的比值大小.在不同的门限参数下,同一虚拟机将被视作不同的类型,从而采用不同的分配策略,占用网络拓扑中不同角色的网络节点,对整个网络产生不同程度的影响.

本文对决定虚拟机类型的两个门限参数 〈α,β〉进行仿真测试.在不同虚拟机数量(10～300)下,采用不同的 〈α,β〉值(α 从 0.1 到 1,β 从 1 到 10)进行实验,对比网络感知分配算法在分配时间和网络稳定性上的差异.表 2 描述了在 100 个虚拟机请求下 7 种不同 〈α,β〉值对网络延迟率、网络吞吐率和分配时间的影响.从表中可以看出,随着 α 的不断增加(0.1～0.5),β 的不断减少(10～2),网络感知算法的延迟率、吞吐率和分配时间逐渐减少;当 α 和 β 在 〈0.5,2〉时,算法的性能较好;随着 α 的继续增加(0.5～1),β 的继续减少(2～1),算法的延迟率、吞吐率和分配时间又逐渐增加.同样,表 3 展示了 7 种不同 〈α,β〉值在 200 个虚拟机请求下对算法的延迟率、吞吐率和分配时间的影响.从图中可以得到表 2 相似的结论,随着 α 的不断增加,β 的不断减少,算法的性能先提升后降低.

表 2　100 个虚拟机下门限参数性能对比表
Tab.2　Threshold parameter performance comparison of 100 virtual machines

〈α,β〉值	延迟率	吞吐率/(Mb/s)	分配时间/s
〈0.1,10〉	0.175	3.621	244
〈0.2,7〉	0.168	3.715	226
〈0.3,5〉	0.143	3.941	217
〈0.4,3〉	0.131	4.084	203
〈0.5,2〉	0.127	4.148	195
〈0.7,1.5〉	0.137	4.176	193
〈0.8,1〉	0.145	4.017	209

表 3　200 个虚拟机下门限参数性能对比表
Tab.3　Threshold parameter performance comparison of 200 virtual machines

〈α,β〉值	延迟率	吞吐率/(Mb/s)	分配时间/ms
〈0.1,10〉	0.191	4.146	507
〈0.2,7〉	0.176	4.045	481
〈0.3,5〉	0.161	4.201	461
〈0.4,3〉	0.149	4.345	442
〈0.5,2〉	0.138	4.471	419
〈0.7,1.5〉	0.146	4.441	433
〈0.8,1〉	0.149	4.438	445

通过表 2 和表 3 的分析,虚拟机类型的门限参

数值〈α,β〉在〈0.5,2〉时,整个算法的延迟率、吞吐率和分配时间的性能最佳,算法拥有较好的网络稳定性和分配效率.因此,在后续的实验对比中,基于网络感知的两阶段虚拟机分配算法内虚拟机类型的门限参数值,公式(4),默认设置为〈0.5,2〉.

3.3 实验结果

本次实验从分配时间、延迟率、吞吐率、CPU 利用率、带宽利用率、主机使用情况等六个方面对 NWTP 算法进行验证,体现了 NWTP 算法的性能.

图 5 为 4 种算法的分配时间对比.从图中可以看出,随着虚拟机请求数量的不断增加,分配时间不断增大.在虚拟机数量较小时(10～30),4 种算法的分配时间相近.但是,随着虚拟机数据的不断增加(30～300),NWTP 算法的分配时间明显优于 SA 和 GA 算法,仅次于 GR 算法.因为,GR 算法采用随机分配策略,不要考虑其他的分配指标,寻求局部解,分配时间很快.NWTP,SA 和 GA 算法都在寻找较优解,同时考虑负载均衡等其他目标,需要不断迭代,分配时间较多.然而,在不断迭代中,NWTP 算法采用流水线技术,并发进行多个阶段的分配,提升了分配效率.

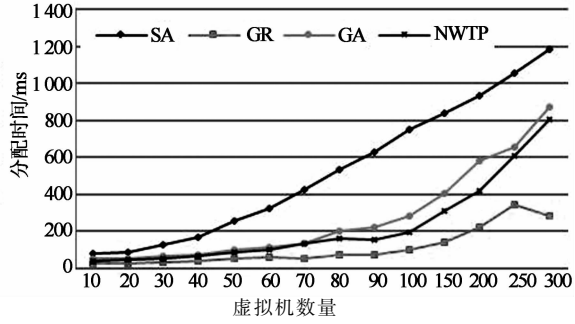

图 5 虚拟机分配时间对比图

Fig. 5 Comparison of VM allocation time

图 6 展示了 4 种算法在不同虚拟机请求下平均延迟率的对比.延迟率是指通信中延迟时间与正常通信时间的比值.从图中可以看出,随着虚拟机数量的不断增加,4 种算法的延迟率呈上升趋势,GR 算法的延迟率最大,GA 和 SA 的延迟率相近,NWTP 算法延迟率最优.当虚拟机请求数量较小时(0～100),4 种算法的延迟率都较小且频繁波动.其中,GR 延迟率最大,波动最剧烈.SA 的延迟率其次,波动比较频繁.GA 延迟率与 SA 近似,但是波动最小.NWTP 的延迟率最小,波动与 SA 近似.因为,当虚拟机请求较小时,整个网络的吞吐量较小,虚拟机间的通信彼此不会过多干扰,延迟率较小.但是,GR 算法的贪婪分配原则,会将虚拟机快速分配到高性能物理主机上,造成虚拟机的地理位置较分散,延迟率的波动最大.SA 和 GA 算法都通过反复迭代寻求较优的分配位置,其延迟率优于 GR,波动程度也较小.NWTP 则将虚拟机按照带宽请求进行分类,然后区域性地进行不同类型的虚拟机分配,保证虚拟机分配到底层彼此靠近的某几台物理主机上,减少了通信距离.但是,由于底层占用的物理主机位置比较集中,虚拟机间的冲突将增大,导致延迟率有所波动.随着虚拟机请求数量的增加(100～300),4 种算法的延迟率逐渐增加,波动程度逐渐趋于平缓.其中,NWTP 和 GA 的延迟率最小,NWTP 和 SA 波动最均衡.因为,随着虚拟机数量的增加,4 种算法占用的物理主机数量快速增加,虚拟机间的通信可能跨越多个节点,增加了网络的平均延迟率,降低了四种算法对应的延迟波动.但是,GR,SA 和 GA 算法的物理主机分散性,必然导致通信的延迟率加大,其性能差于 NWTP 算法.

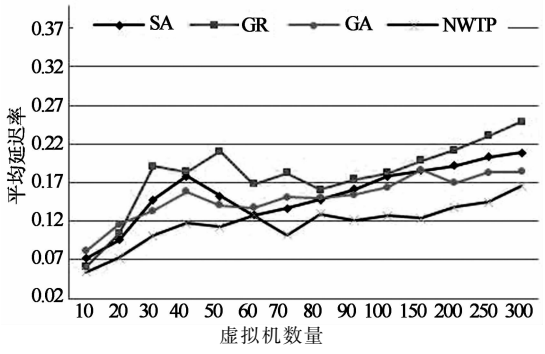

图 6 延迟率对比图

Fig. 6 Comparison of network delay rate

图 7 为 4 种算法在不同虚拟机请求下平均吞吐率的对比图.从图中可以看出,随着虚拟机请求数量的不断增加,4 种算法的平均吞吐率逐渐上升.其中,NWTP 算法的吞吐率最大,其次是 GA 和 SA,GR 的吞吐率最小.当虚拟机请求数量较小时(0～40),NWTP 算法的吞吐率明显高于其他 3 种算法.因为,NWTP 将虚拟机分配到相对集中的物理主机区域,减少了虚拟机间的通信距离,降低了虚拟机的通信延迟,增加了吞吐量.GR 算法随机选择高性能主机进行分配,虚拟机位置比较分散.SA 和 GA 通过迭代选择较优物理主机,虚拟机的位置同样较分散,通信距离较长,延迟较大,吞吐率较差.随着虚拟机数量的增加(50～300),其他 3 种算法的吞吐率逐渐逼近 NWTP 算法.其中,SA 和 GA 比较接近,GR 吞吐率最差.因为,随着虚拟机数量增加,占用底层物理主机资源数量增大,网络的通信负载加大,虚拟

机间的相互干扰增加,平均的通信延迟加大,4 种算法间的吞吐量彼此靠近. 较其他 3 种算法,NWTP 算法的区域性集中分配策略,能够减少通信距离,降低延迟,提升吞吐率.

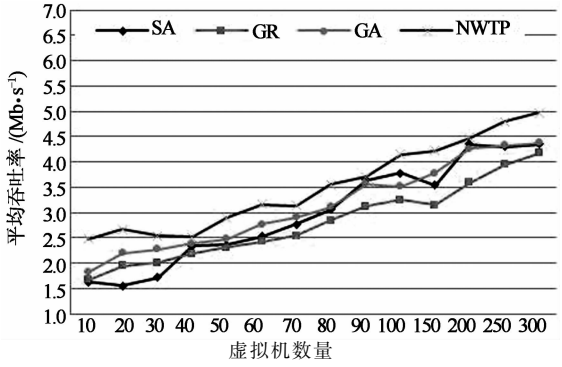

图 7　吞吐率对比图

Fig. 7　Comparison of network throughput rate

图 8 展示了 4 种算法占用底层物理主机的平均 CPU 利用率. 从图中可以看出,不同数量的虚拟机请求下(10～300),4 种算法占用底层物理主机的平均 CPU 利用率波动较大. NWTP 算法和 SA 算法的波动情况相对均衡,GA 算法次之,GR 算法最差. NWTP 算法通过区域化的分配物理主机,使得区域内物理主机的 CPU 使用相对均衡,从而提升了整体网络中物理主机的平均 CPU 利用率. GR 算法的随机分配策略,必然造成高性能的物理主机优先分配,随着虚拟数据的不断增加,浪费的虚拟机资源不断增加,平均 CPU 利用率的波动变化明显.

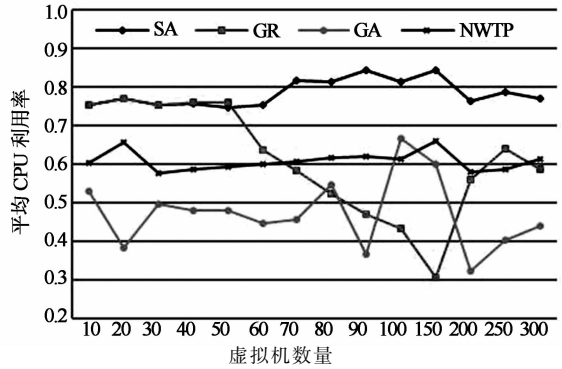

图 8　CPU 利用率对比图

Fig. 8　Comparison of CPU utilization

图 9 是 4 种算法平均带宽利用率的对比图,展示了承载虚拟机的物理主机在多次实验过程中平均带宽利用率. 从图中可以看出,随着虚拟机数量的不断变化(从 10～300),4 种算法占用底层物理主机的带宽资源快速变化. GR 算法变化最快,GA 算法次之,NWTP 算法稳定性最好. 此外,从图中还可知,较 GR 和 GA 两种算法,NWTP 算法在整个分配过程中,具有较高的带宽利用率,符合提高底层资源利用率的目标. 因为,GR 算法的随机分配策略,未考虑任务资源的负载情况,很容易造成资源负载的不均衡. GA 算法的不断迭代,选择不同类型的物理主机组合,造成主机间带宽通信较频繁,从而造成带宽负载的巨大波动. NWTP 算法将带宽分解成网间带宽、网内带宽,并根据虚拟机的带宽类型,分配到网络拓扑中的不同区域,减少网内带宽的同时,均衡了区域物理主机的带宽负载,增加了区域带宽的利用率,提升了数据中心的稳定性,对网络突发流有较强的扩展性和适应性.

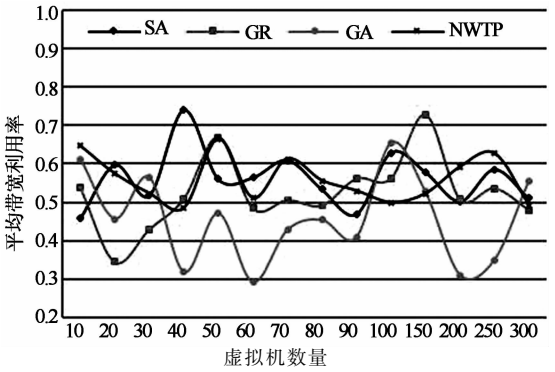

图 9　带宽利用率对比图

Fig. 9　Comparisonof bandwidth utilization

图 10 展示了 4 种算法使用底层物理主机的情况对比. 相同的虚拟机请求使用不同的分配算法,其占用的底层物理主机不同. 从图 10 中可以看出,GA 算法占用了较多的高性能物理主机,SA 和 GA 算法对高性能物理主机的占用情况都明显高于 NWTP 算法. 在数据中心中,高性能的物理主机往往起着非常核心的作用,是网络拓扑的枢纽,成本较高. 较差性能的物理主机则属于边界节点,枢纽性

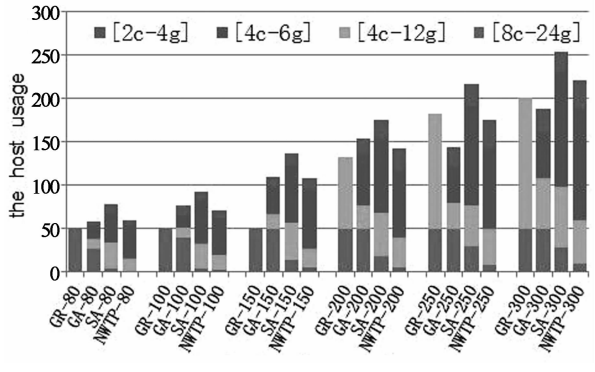

图 10　主机使用情况对比图

Fig. 10　Comparison of physical hosts usage

差,成本低. NWTP 算法的边界区域分配策略很好地迎合了这种特性,能够更加适应数据中心的现实

状况.

4 结 论

针对云数据中心网络拓扑的随机性,结合虚拟机分配很少同时考虑带宽资源和物理主机资源的现状,本文提出了一种基于网络感知的两阶段虚拟机分配算法,NWTP. 首先,根据交互对象的不同,将虚拟机的带宽请求分解为网间带宽和网内带宽两种. 利用节点介数和聚集系数感知节点在网络拓扑中的邻居依赖性和网络枢纽性;其次,将整个虚拟机分配过程分解为带宽区域划分和物理主机分配两个彼此连续的阶段,分别建立网络感知模型;然后,利用流水线技术,将带宽区域划分、物理主机分配和错误处理看作彼此连续的三个工序(环节),并为每个环节设计了相应的处理算法,并发进行虚拟机分配. 最后,将 NWTP 算法与遗传 GA,模拟退火 SA,贪婪 GR 算法进行对比,验证了 NWTP 算法在分配时间、网络延迟、吞吐率、带宽利用率、CPU 利用率和占用主机均衡度等方面的优势.

参考文献

[1] GREENBERG A, HAMILTON J, MALTZ D A, *et al*. The cost of a cloud: research problems in data center networks[J]. ACM SIGCOMM Computer Communication Review, 2008, 39(1): 68－73.

[2] PATIDAR S, RANE D, JAIN P. A survey paper on cloud computing [C]//Advanced Computing & Communication Technologies (ACCT). Rohtak: 2012 Second International Conference. IEEE, 2012: 394－398.

[3] BARI M F, BOUTABA R, ESTEVES R, *et al*. Data center network virtualization: A survey[J]. Communications Surveys & Tutorials, IEEE, 2013, 15(2): 909－928.

[4] ALIZADEH M, GREENBERG A, MALTZ D A, *et al*. Data center tcp (dctcp)[J]. ACM SIGCOMM Computer Communication Review, 2011, 41(4): 63－74.

[5] QI H, SHIRAZ M, LIU J, *et al*. Data center network architecture in cloud computing: review, taxonomy, and open research issues[J]. Journal of Zhejiang University Science C, 2014, 15(9): 776－793.

[6] GAHLAWAT M, SHARMA P. Survey of virtual machine placement in federated clouds[C]//Advance Computing Conference (IACC). Gurgaon: 2014 IEEE International. IEEE, 2014: 735－738.

[7] SINDELAR M, SITARAMAN R K, SHENOY P. Sharing-aware algorithms for virtual machine collocation[C]//Proceedings of the 23rd ACM Symposium on Parallelism in Algorithms and Architectures (SPAA'11). New York: SPAA'11. ACM, 2011: 367－378.

[8] REMESHBABU K R, SAMUEL P. Virtual machine placement for improved quality in IaaS cloud [C]//Advances in Computing and Communications (ICACC). Cochin: 2014 Fourth International Conference. IEEE, 2014: 190－194.

[9] LIU C, SHEN C, LI S, *et al*. A new evolutionary multi－objective algorithm to virtual machine placement in virtualized data center[C]//Software Engineering and Service Science (ICSESS). Beijing: 2014 5th IEEE International Conference. IEEE, 2014: 272－275.

[10] TANG M, PAN S. A hybrid genetic algorithm for the energy－efficient virtual machine placement problem in data centers [J]. Neural Processing Letters, 2014, 41(2): 211－221.

[11] TENG F, DENG D, YU L, *et al*. An energy-efficient VM placement in cloud datacenter[C]//High Performance Computing and Communications 2014 IEEE 6th Intl Symp on Cyberspace Safety and Security. Paris: 2014 IEEE 11th Intl Conf on Embedded Software and System (HPCC, CSS, ICESS). IEEE, 2014: 173－180.

[12] WANG R, ESTEVES R, SHI L, *et al*. Network-aware placement of virtual machine ensembles using effective bandwidth estimation[C]//Network and Service Management (CNSM). Rio de Janeiro: 2014 10th International Conference. IEEE, 2014: 100－108.

[13] ZHAO Y, HUANGY, CHEN K, *et al*. Joint VM placement and topology optimization for traffic scalability in dynamic datacenter networks[J]. Computer Networks, 2015, 80: 109－123.

[14] BUYYA R, RANJAN R, CALHEIROS R N. Modeling and simulation of scalable cloud computing environments and the CloudSim toolkit: Challenges and opportunities [C]//High Performance Computing & Simulation. Leipzig: International Conference on HPCS'09. IEEE, 2009: 1－11.

[15] MILLS K, FILLIBEN J J, DABROWSKI C. Comparing vm－placement algorithms for on－demand clouds [C]// Cloud Computing Technology and Science (CloudCom). Athens: 2011 IEEE Third International Conference. IEEE, 2011: 91－98.

[16] PANDIT D, CHATTOPADHYAY S, CHATTOPADHYAY M, *et al*. Resource allocation in cloud using simulated annealing[C]//Applications and Innovations in Mobile Computing (AIMoC). Kolkata: AIMoC 2014. IEEE, 2014: 21－27.

[17] TANG M, PAN S. A hybrid genetic algorithm for the energy－efficient virtual machine placement problem in data centers [J]. Neural Processing Letters, 2014: 1－11.