软件学报 ISSN 1000-9825, CODEN RUXUEW
Journal of Software,2014,25(7):1416−1431 [doi: 10.13328/j.cnki.jos.004603]
©中国科学院软件研究所版权所有.
E-mail: jos@iscas.ac.cn
http://www.jos.org.cn
Tel/Fax: +86-10-62562563

主动休眠节点链路的高效节能虚拟网络映射*

陈晓华[1,3], 李春芝[2,3], 陈良育[1], 曾振柄[4]

[1](华东师范大学 软件学院,上海 200062)

[2](华东师范大学 计算机科学技术系,上海 200241)

[3](湖州师范学院 信息与工程学院,浙江 湖州 313000)

[4](上海大学 数学系,上海 200444)

通讯作者: 李春芝, E-mail: lichunzhi82@126.com, http://www.ecnu.edu.cn

摘 要: 网络虚拟化,使得智能能量感知网络部署成为可能.由于当前网络为高峰负荷而设计,导致资源利用率不足及能量浪费.而网络设备能量消耗对于流量负载不敏感,资源整合成为有效节能技术.根据虚拟网络映射特点及底层网络能耗,提出虚拟网络映射节能多目标决策模型;由于该模型是混合整数规划模型,求解时间复杂度高,通过分析虚拟网络映射动态特征,构造虚拟网络映射字典库,提出底层网络资源利用率的训练方法以及主动休眠底层节点和链路算法,把虚拟网络映射在一个较小的节点和链路集合中,提高休眠节点和链路数量,实现高效节能虚拟网络映射.系统仿真结果验证了主动休眠方法能够提高底层节点和链路休眠数量,显著减少系统能耗.

关键词: 虚拟网络映射;主动休眠;高效节能;资源整合

中图法分类号: TP393

中文引用格式: 陈晓华,李春芝,陈良育,曾振柄.主动休眠节点链路的高效节能虚拟网络映射.软件学报,2014,25(7): 1416−1431. http://www.jos.org.cn/1000-9825/4603.htm

英文引用格式: Chen XH, Li CZ, Chen LY, Zeng ZB. Energy efficient virtual network embedding based on actively hibernating substrate nodes and links. Ruan Jian Xue Bao/Journal of Software, 2014,25(7):1416−1431 (in Chinese). http://www.jos.org.cn/1000-9825/4603.htm

Energy Efficient Virtual Network Embedding Based on Actively Hibernating Substrate Nodes and Links

CHEN Xiao-Hua[1,3], LI Chun-Zhi[2,3], CHEN Liang-Yu[1], ZENG Zhen-Bing[4]

[1](Software Engineering Institute, East China Normal University, Shanghai 200062, China)

[2](Department of Computer Science and Technology, East China Normal University, Shanghai 200241, China)

[3](School of Information and Engineering, Huzhou Teachers College, Huzhou 313000, China)

[4](Department of Mathematics, Shanghai University, Shanghai 200444, China)

Corresponding author: LI Chun-Zhi, E-mail: lichunzhi82@126.com, http://www.ecnu.edu.cn

Abstract: Network virtualization will be an enabler for intelligent energy-aware network deployment. Current networks are designed for peak loads, resulting in inadequate resource utilization and energy consumption waste. Due to current power consumption insensitiveness of network equipment to traffic load, resource consolidation becomes an effective energy-saving technology. Based on the virtual network mapping characteristics and the substrate network energy consumption, this paper presents a multi-objective decision-making model that is also a mixed integer programming model for energy efficient virtual network embedding. To address high time complexity in solving the mixed integer programming model, the paper analyzes the dynamic characteristics of the virtual network mapping, constructs virtual network mapping dictionary database and proposes a method for training substrate network resource utilization, as well as an algorithm which actively hibernates the substrate nodes and links. By this method, the virtual network can be embedded in a smaller set of substrate nodes and links, which helps to increase the number of hibernating substrate nodes and links, and achieve energy-effective virtual network

* 收稿时间: 2013-12-22; 修改时间: 2014-03-17; 定稿时间: 2014-05-06

© 中国科学院软件研究所 http://www.jos.org.cn

mapping. Simulation results demonstrate the proposed method can effectively improve the number of hibernating nodes and links of substrate network, and significantly reduce energy consumption of substrate network.

Key words: virtual network embeding; actively hibernate; energy efficient; resource consolidation

随着电力成本不断上涨和人们生态意识的提高,网络运营商已经意识到能耗管理的重要性,减少能耗已成为亟待解决的问题.当前网络为高峰负荷而设计,网络资源超量供给确保了网络的正常运行,然而也导致资源利用率低下.据统计,大型 ISP 骨干网的平均链路利用率大约为 30%~40%[1],数据中心服务器的平均利用率为 11%~50%[2,3].过低的利用率造成了巨大的电能浪费,促使绿色网络研究的兴起,网络能耗问题成为研究热点[4,5].

网络虚拟化[6]是未来因特网、云计算和软件定义网络的重要技术[7−11],其通过整合网络基础设施资源,能够合理有效地使用能量,使得智能能量感知网络部署成为可能.虚拟网络映射是网络资源虚拟化的关键问题.当前,大部分映射算法是基于代价的虚拟网络映射[12],即,以最小化底层资源代价映射虚拟网络请求,以此获得更多的底层物理资源,进而提高虚拟网络接收率与系统收益.然而,由于虚拟网络请求是一个动态变化的过程,而底层物理网络根据流量峰值设计,基于代价的虚拟网络映射必然带来不必要的能耗.以节能为目标的虚拟网络化映射,应在满足当前虚拟网络请求的前提下最小化能耗.由于当前网络设备对流量负荷的功耗不敏感[13],因此在不影响虚拟网络映射性能的情况下,尽可能多地关闭或休眠网络节点和链路是节能的有效方法.

当前,基于能量感知的虚拟网络映射通过修改虚拟网络映射算法,使得虚拟网络尽可能地映射到活动的节点和链路,以达到系统节能的目的.如:文献[14]通过减少物理网络设备数量分配虚拟网络请求集合,提出混合整数规划的能量感知最优化模型,但是时间复杂度呈指数增长,难以适应大规模网络基础设施的虚拟网络映射;文献[15]考虑到机箱能耗比路由能耗低的特点,提出扩展流量到网络资源的节能方法,但只适合对负载敏感的设备;文献[16]提出虚拟网络重配置的最小化能耗的启发式方法;北京邮电大学的苏森等人提出虚拟网络映射能耗模型以及能量感知两阶段映射算法[17];北京交通大学的常晓林、王冰等人提出混合整数规划能耗模型及能量感知两阶段映射算法[18];文献[19]在云数据中心中应用蚁群优化算法求解虚拟网络节能映射.可见,目前相关节能感知映射算法是通过修改虚拟网络映射算法,被动地休眠网络节点和链路.

本文考虑虚拟网络映射成本收益比以及系统能耗,提出多目标决策的虚拟网络映射能耗模型,在保证虚拟网络映射收益成本比的前提下降低系统能耗.由于该模型是混合整数规划模型,求解时间复杂度高,我们通过分析虚拟网络映射动态特征,定义系统饱和与非饱和状态,并提出主动休眠底层节点和链路的方法,把虚拟网络映射在一个较小的节点和链路集合中,提高休眠节点和链路数量,有效节约系统能耗.

本文第 1 节分析虚拟网络映射模型和底层物理网络能耗模型,建立虚拟网络映射多目标决策模型.第 2 节分析虚拟网络映射动态特征,构建虚拟网络映射字典库,设计底层网络资源利用率的训练方法,建立休眠底层节点和链路公式,设计主动休眠链路算法.第 3 节仿真验证了在系统非饱和状态下主动休眠链路算法节能的有效性.最后是总结和展望.

1 虚拟网络映射及其能耗模型

网络虚拟化节能映射,应在保证虚拟网络映射收益成本比的前提下降低系统能耗.本节首先描述了虚拟网络映射模型和底层物理网络能耗模型,然后提出多目标决策的虚拟网络映射能耗模型.

1.1 虚拟网络映射模型

虚拟网络映射是网络虚拟化的主要问题之一,下面介绍底层网络、虚拟网络以及虚拟网络映射模型.

- 底层网络

本文通过无向图 $G^s=(N^s,L^s,C_N^s,C_L^s)$ 对底层网络建模,其中,N^s 为节点集合,L^s 为链路集合,C_N^s 为节点属性集合,C_L^s 为链路属性集合.本文设置节点属性为 CPU 处理器资源,链路属性为带宽资源.

- 虚拟网络

与底层网络相同,本文通过无向图 $G^v=(N^v,L^v,C_N^v,C_L^v)$ 对虚拟网络建模,其中,N^v 为节点集合,L^v 为链路集合,

© 中国科学院软件研究所 http://www.jos.org.cn

C_N^v 为节点属性集合(CPU 处理器资源), C_L^v 为链路属性集合(带宽资源).

- 虚拟网络映射

把虚拟节点和链路映射到满足虚拟资源需求的底层节点和路径,可进一步分为节点映射和链路映射.节点映射分为:一个虚拟网络的不同节点不允许映射到同一节点、一个虚拟网络的不同节点可映射到一个节点.链路映射分为单路径映射和多路径映射[20].本文假设在一个虚拟网络的不同节点不允许映射到同一节点的情况下,研究虚拟网络节能映射问题.

1.2 底层网络能耗

底层网络的能耗主要包括节点能耗和链路能耗两个部分.

网络节点主要是服务器,其能耗主要包括处理器、内存、磁盘 I/O 以及用于冷却的风扇,其中,处理器和内存占有节点能耗的大部分.当前,处理器能耗与负载具有较好的相关性,例如 Intel 公司的 Speedstep 以及 AMD 公司的 PowerNow 技术,处理器能够根据负载动态地调节性能[4].本文抽象节点的属性为处理器的属性,网络节点的能耗与该节点承载的虚拟网络节点总和成比例关系.定义第 i 个节点能耗为

$$PN^i=\begin{cases}P_b+P_l\cdot u, & \text{如果}i\text{处于激活状态}\\ 0, & \text{否则}\end{cases} \tag{1}$$

其中,P_b 为节点的基本能耗,P_m 为节点的最大能耗,$P_l=P_m-P_b$ 为与处理器利用率 u 的能耗因子.

由于当前网络设备对流量负荷的功耗不敏感[13],通常认为专有的减负引擎将被广泛地部署在网络虚拟化中[21−23],因为这种引擎既能够保持高的数据包处理率,又能够产生低的处理延时.无论接口是否空闲或者满负荷运行,网络链路的能耗 P_n 为常量[22,24].定义第 j 条链路能耗为

$$PL^j=\begin{cases}P_n, & \text{如果}j\text{处于激活状态}\\ 0, & \text{否则}\end{cases} \tag{2}$$

底层网络的节点和链路能耗模型为虚拟网络映射多目标决策模型提供了现实基础.

1.3 虚拟网络映射多目标决策模型

基于代价的虚拟网络映射,是以收益成本比最大化为目标,基于能耗的虚拟网络映射应在满足系统收益成本比最大化的前提下,实现节能的目标.本节把基于能耗的虚拟网络映射转化为多目标决策模型,采用分层法建模,如下所示:

设定收益成本比 r/c 和系统能耗 $Paver$ 这两个目标,并给出重要性序列:r/c,$PAver$.首先针对 r/c 目标,找到满足最大 r/c 的虚拟网络映射解集合 R_0;然后在 R_0 内,求满足最小系统能耗 $PAver$ 的虚拟网络映射解集合 R_1.这样就把虚拟网络映射问题转化为多目标决策模型,具体虚拟网络映射的多目标决策模型如下所示.

- 目标函数:

$$\min \sum_{i\in N^s} PN^i+\sum_{j\in L^s} PL^j \tag{3}$$

- 映射成本比最大化约束:

$$\max \frac{\sum_{l^v\in L^v} BWL(l^v)+\sum_{n^v\in N^v} CPU(n^v)}{\sum_{l^v\in L^v}\sum_{l^s\in L^s} bw(l^v,l^s)+\sum_{n^v\in N^v} CPU(n^v)} \tag{4}$$

- 容量约束:

$$(\forall u\in N^v)(\forall j\in N^s): x_j^u\cdot CPU(u)\leqslant CPU(j) \tag{5}$$

$$(\forall l_{jk}\in L^s)(\forall l_{uw}\in L^v): f_{jk}^{uw}\cdot BWL(l_{uw})\leqslant BWL(l_{jk}) \tag{6}$$

- 传输约束:

$$(\forall i,j\in N^s): LD_{BW}^{i,j}=\sum_{l_{uw}\in L^v} LD_{BW}(u,w)x_i^u x_j^w \tag{7}$$

© 中国科学院软件研究所 http://www.jos.org.cn

- 一个虚拟节点映射到一个底层节点的约束:

$$(\forall u \in N^v): \sum_{j \in N^s} x_j^u = 1 \tag{8}$$

- 相同虚拟节点不能映射到同一底层节点的约束:

$$(\forall u \in N^v): \sum_{j \in N^s} x_j^u = NNo \tag{9}$$

其中,

- $CPU(j)$表示,如果 j 是底层节点,则 $CPU(j)$是 j 剩余 CPU 资源量;如果 j 是虚拟节点,则 $CPU(j)$是 j 请求的 CPU 资源量;
- $BWL(l_{jk})$表示如果 l_{jk} 为底层链路,则 $BWL(l_{jk})$是 l_{jk} 剩余带宽;如果 l_{jk} 为虚拟链路,则 $BWL(l_{jk})$是 l_{jk} 请求带宽;
- $Bw(l^v,l^s)$为虚拟链路 l^v 分配的底层链路 l^s 带宽;
- x_j^u 为二进制变量,若虚拟节点 u 映射到底层节点 j 上,则 $x_j^u = 1$;否则 $x_j^u = 0$;
- NNo 为虚拟网络节点数量;
- f_{jk}^{uw} 为二进制变量,若虚拟链路 l_{uv} 映射到底层链路 l_{jk} 上,则 $f_{jk}^{uw} = 1$,否则 $f_{jk}^{uw} = 0$;
- $LD_{BW}^{i,j}$ 表示底层节点 i 到 j 为虚拟网络分配的带宽总量,由一个或者多个虚拟链路 l_{uv} 带宽组成,$LD_{BW}^{i,j} = \sum_{l_{uw} \in L^v} LD_{BW}(u,w) x_i^u x_j^w$, $LD_{BW}(u,w)$为虚拟链路 l_{uw} 带宽.

$LD_{BW}^{i,j}$ 根据不同的链路映射条件具有不同的涵义,即,单路径链路映射与多路径链路映射. $LD_{BW}^{i,j}$ 的约束是不同的,下面介绍单路径流以及多路径流的概念.

- 单路径流

在底层网络中,设 $path$ 是底层节点 s 到 t 的一条无环路径,满足下述条件的函数 $flow$ 称为 s 到 t 在 $path$ 上的可行流.

(1) 容量限制条件:在 $path$ 中的每条边(i,j),满足 $0 \leqslant flow(l_{ij}) \leqslant BWL(l_{ij})$;

(2) 方向条件:$\forall i,j \in N^s, flow(l_{ij}) = -flow(l_{ji})$,单路径流有方向性;

(3) 平衡条件:对于中间点,流出量等于流入量,即在 $path$ 中对每个 $u(u \neq s,t)$,有 $\sum_{(u,m) \in N^s} flow(l_{um}) = 0$;对于 s 和 t, $\sum_{(s,p) \in N^s} flow(l_{sp}) = \sum_{(q,t) \in N^s} flow(l_{qt})$.

- 多路径流

在底层网络中,s 和 t 是底层的两个节点,则 s 到 t 的多路径流 $mflow$ 由多条单路径流 $flow$ 组成,同时满足以下条件:

(1) 容量限制条件:每条边(i,j),满足 $0 \leqslant mflow(l_{ij}) \leqslant BWL(l_{ij})$;

(2) 方向条件:$\forall i,j \in N^s, mflow(l_{ij}) = -mflow(l_{ji})$,多路径流有方向性;

(3) 平衡条件:对于中间点,流出量等于流入量,即:对每个 $u(u \neq s,t)$,有 $\sum_{(u,m) \in N^s} mflow(l_{um}) = 0$;对于 s 和 t, $\sum_{(s,p) \in N^s} mflow(l_{sp}) = \sum_{(q,t) \in N^s} mflow(l_{qt})$.

如果 $LD_{BW}^{i,j}$ 为单路径流 $flow$,则映射为单路径链路映射;如果 $LD_{BW}^{i,j}$ 为多路径流 $mflow$,则映射为多路径链路映射.

设计的多目标决策模型实现了在保证虚拟网络映射成本比 r/c 最大化下的系统能耗最小化,同时,该模型是基于混合整数规划模型的多目标决策模型;由于混合整数规划求解时间复杂度随着问题规模呈指数增长,本文分析网络虚拟化动态特征,设计基于主动休眠节点和链路的启发式算法,找到虚拟网络映射解.

© 中国科学院软件研究所 http://www.jos.org.cn

2 底层网络资源主动休眠

本文根据虚拟网络映射动态特征,定义系统饱和状态与非饱和状态,构建虚拟网络映射字典库,提出底层网络资源利用率训练方法;定义底层网络节点和链路休眠公式;然后,采取主动休眠底层节点和链路的方法,缩小虚拟网络映射的底层网络资源范围,以提高休眠节点和链路数量,达到底层网络节能目的.

2.1 一个节能例子

在负荷较低的情况下,底层网络具有足够的资源使得虚拟网络接受率和系统收益达到最大值,系统节能成为重要目标.如图 1 所示,底层网络具有足够的资源,能够全部接收两个虚拟网络请求,其中,方案(a)的映射使得所有节点和链路资源处于激活状态;而方案(b)把虚拟网络映射到较小的底层资源集合中,使得节点 D 以及与之连接的两条链路(灰色表示)未被映射,可以进入节能休眠状态.本文在底层网络资源中找到一个集合,该集合中的所有节点和链路可以休眠,而对应的补集能够接受所有的虚拟网络请求.具体方法是:首先,根据虚拟网络映射的动态特征,构建虚拟网络映射字典库;然后,由机器学习方法训练出底层网络最大资源利用率;最后,求解最大资源量与当前虚拟网络请求资源量之间的差额,得到可以休眠的底层网络资源数量,从而主动休眠底层节点和链路,实现节能的目的.

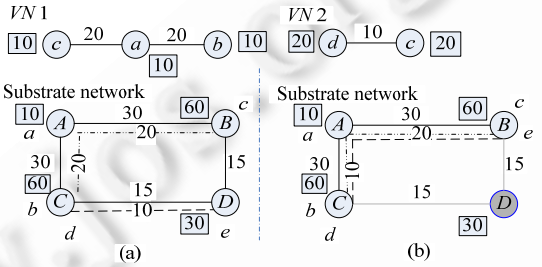


Fig.1 An example of energy-saving virtual network mapping

图 1 虚拟网络映射节能例子

2.2 网络虚拟化动态特征及系统状态

网络虚拟化动态特征包括虚拟网络、底层网络和虚拟网络映射算法动态特征,具体如下:

- 虚拟网络动态特征:虚拟网络请求的到来时间、存在时间、虚拟网络节点个数、虚拟链路条数,节点 CPU、链路带宽等;
- 底层网络动态特征:随着虚拟网络请求的到来和离开,底层网络剩余 CPU、链路剩余带宽资源量和分布将会动态地发生变化;
- 映射算法动态特征:随着虚拟网络请求资源量的变化,在不同映射算法下,底层网络资源利用率、虚拟网络接收率和系统收益是不同的.

系统饱和状态以及非饱和状态:虚拟网络全部接收的状态为系统非饱和状态;反之,为饱和状态.

当系统处于非饱和状态时,底层网络具有足够的资源,能够映射所有的虚拟网络请求,虚拟网络接收率达到 100%,系统收益达到最高值;当系统处于饱和状态时,底层网络没有足够的资源,不能映射所有的虚拟网络请求,虚拟网络接收率小于 100%.

2.3 构建虚拟网络映射字典库

由于虚拟网络映射具有动态特征,虚拟网络映射字典库的数据项包括虚拟网络特征值、底层网络特征值以及虚拟网络算法映射结果.

- 虚拟网络特征值

设 R_v 表示一个时间窗内到达的虚拟网络请求数量,其数学期望为 $E(R_v)$; N_v^l 表示一个虚拟网络的链路数量,其数学期望为 $E(N_v^l)$;B_v^l 表示一条虚拟链路带宽,其数学期望为 $E(B_v^l)$;S_v 表示一个虚拟网络的生存时间,其数学

© 中国科学院软件研究所 http://www.jos.org.cn

期望为 $E(S_v)$;设 N_v^n 表示一个虚拟网络的节点数量,其数学期望为 $E(N_v^n)$;B_v^n 表示一个虚拟节点 CPU 处理器资源量,其数学期望为 $E(B_v^n)$.

- 底层网络特征值

设 $E(B_s^l)$ 表示一条底层链路带宽的数学期望, $E(B_s^n)$ 表示一个底层节点 CPU 资源的数学期望,底层节点数量为$|N^s|$;底层链路数量为$|L^s|$.

- 虚拟网络映射结果

映射算法名称 *AName*、虚拟网络接收率 *mr*、收益成本比 *r*/*c*、底层链路资源利用率 *crl*、底层节点资源利用率 *crn*.*crl* 和 *crn* 是长期平均的利用率:*crl* 使用的是已利用的底层带宽总和除以底层带宽总和的计算方法,*crn* 使用的是已利用的底层节点 CPU 总和除以底层节点 CPU 总和的计算方法.

构造虚拟网络映射字典库的方法是:创建多组具有相同动态特征的虚拟网络请求和多个底层网络;然后,分别运行虚拟网络映射算法,并记录结果.为了保证性能评估的稳定性,记录长期虚拟网络映射结果,其中,每个相同特征的虚拟网络请求构造多组,取特定特征的底层网络.即:在特定特征的底层网络的基础上,每次映射多组具有相同动态特征的不同虚拟网络请求.

2.4 底层节点与链路资源利用率训练算法

基于虚拟网络映射字典库,根据输入底层网络、虚拟网络和算法名称,训练底层网络资源利用率,如算法 1 所示.设 X 为输入的特征向量,包括底层网络特征项、虚拟网络特征项;设 M 为字典库的历史数据数量,字典库有 M 个特征向量,表示为 $Y_1,Y_2,\ldots,Y_M$;特征向量 X 与字典库第 i 个特征向量 Y_i 相同,由底层网络特征项、虚拟网络特征项组成.字典库的字段包括 $E(B_s^l),E(B_s^n),|N^s|,|L^s|,E(R_v),E(N_v^l),E(B_v^l),E(S_v),E(N_v^n),E(B_v^n),AName,mr,r/c,crl,crn$;而 Y_i 由其中的一部分组成,包括 $E(B_s^l),E(B_s^n),|N^s|,|L^s|,E(R_v),E(N_v^l),E(S_v),E(N_v^n)$.

X 与 Y_i 的欧几里德距离为

$$D_i(X)=d(X,Y_i)=\|X-Y_i\|=\sqrt{(X-Y_i)^T(X-Y_i)} \tag{10}$$

第 1 步,通过底层网络、映射算法名称和设定 *mr* 的范围,过滤虚拟网络映射字典库;采用公式(10)计算特征向量 X 与字典库的特征向量 Y_i 的距离 $D_i(X)$;根据 $D_i(X)$降序排列,取 *Nov* 个最小项组成历史数据集合 A,其字段包括了 Y_i 特征项以及 *mr*,*r*/*c*,*crl*,*crn* 等信息.字典库中存在着不同特征的虚拟网络映射结果(即,$E(R_v)$, $E(N_v^l)$, $E(S_v)$, $E(N_v^n)$ 是不同的),公式(10)能够识别出相同特征的虚拟网络映射结果.*Nov* 的取值与字典库中虚拟网络映射训练的数量有关,本文设置为 100,表示每个相同拓扑结构的虚拟网络请求记录了 100 组映射数据.

第 2 步,按照优先 *mr* 降序、其次 $E(B_v^l)$ 和 $E(B_v^n)$ 升序的规则,排序 A 中数据项,取 *Nov* 个最小的数据项,然后计算底层节点与链路的利用率平均值,设置 *crl* 和 *crn*.

基于网络特征期望值而训练 *crl* 和 *crn*,是一种机器学习的估算方法.通过 X 与 Y_i 的欧几里德距离的匹配,能够找到有效历史数据项.取 *Nov* 个最小有效数据项的平均值,可减少虚拟网络映射动态性对系统性能的影响.算法 1 从字典库中训练有效的 *crl* 和 *crn*,下文的算法 2 通过算法 1 计算主动休眠底层节点和链路数量.算法 1 的时间复杂度取决于选择的排序算法时间复杂度,故时间复杂度较低.

算法 1. 训练底层节点与链路的利用率算法.

输入:虚拟网络映射字典库,底层网络,虚拟网络,映射算法名称;

输出:底层节点与链路的利用率.

1. 设定底层网络、映射算法名称和 *mr* 范围,筛选历史数据信息集合;然后,根据公式(10)的判别函数计算距离 $D_i(X)$,并根据 $D_i(X)$降序排列,取 *Nov* 个最小项组成历史数据集合 A;
2. 按照优先 *mr* 降序、其次 $E(B_v^l)$ 和 $E(B_v^n)$ 升序的规则,排序 A 中的数据项;取 *Nov* 个最小数据项,计算底层节点与链路利用率平均值,设置 *crl* 和 *crn*,并输出.

© 中国科学院软件研究所 http://www.jos.org.cn

2.5 节点和链路主动休眠

由于当前底层网络为高峰负荷而设计,则当系统处于非饱和状态时,必然存在一个底层网络资源集合,能够保证虚拟网络映射性能.这个底层网络资源集合包含于底层网络资源.通过计算底层网络资源与映射虚拟网络请求的所需资源之间的差额,求解主动休眠底层网络节点和链路数量,以休眠底层网络节点和链路,从而缩小了底层网络资源集合,提高了休眠节点和链路数量,达到了减少系统能耗的目的.

2.5.1 链路休眠数量

根据一个时间窗虚拟请求个数、虚拟链路数量、虚拟链路带宽、生存时间和虚拟网络接收率,计算映射的虚拟网络链路资源 $Res(VS^l)$,即:

$$Res(VS^l)=E(R_v)\cdot E(B_v^l)\cdot E(N_v^l)\cdot E(S_v)\cdot mr \tag{11}$$

在非饱和状态下,公式(11)中的 mr 为 1,是一个时间窗的虚拟网络请求链路资源的总和.

底层网络链路总会存在一部分链路资源碎片不能利用,因此设定链路资源整体利用率 crl.通过 crl 以及底层物理链路的总和,计算可使用的底层链路网络链路资源 $Res(S^l)$,即:

$$Res(S^l)=crl\cdot\sum bw^s \tag{12}$$

其中,bw^s 为底层网络带宽,$\sum bw^s$ 为底层网络带宽总和.

在非饱和状态下,可以休眠底层链路,以达到节能的目的,计算休眠的底层链路数量 $sleep^l$ 公式如下:

$$sleep^l=INT_(Res(S^l)-Res(VS^l))/E(B_s^l)) \tag{13}$$

其中,$Res(S^l)$为非饱和状态下的可以使用的底层网络链路资源;crl 通过训练得到,针对不同的映射算法,crl 是不同的; $Res(VS^l)$通过公式(11)计算,表示一个时间窗的所有虚拟网络链路资源总量;$INT_$表示取下整函数.

根据 $sleep^l$ 的计算结果,判断是否可以休眠链路:如果 $sleep^l$ 大于 0,说明可以通过休眠底层链路的方式节能;如果小于等于 0,则说明不能通过休眠链路的方式节能,否则将会影响系统收益以及虚拟网络接收率.

2.5.2 节点休眠数量

根据一个时间窗虚拟请求个数、虚拟节点数量、虚拟节点 CPU 处理器资源量、生存时间和虚拟网络接收率,计算映射的虚拟网络节点资源 $Res(VS^n)$,即:

$$Res(VS^n)=E(R_v)\cdot E(B_v^n)\cdot E(N_v^n)\cdot E(S_v)\cdot mr \tag{14}$$

在非饱和状态下,公式(14)中的 mr 为 1,是一个时间窗的虚拟网络请求节点资源的总和.

底层网络节点总存在一部分节点资源碎片不能利用,设定节点资源整体利用率 crn.通过 crn 以及底层节点 CPU 资源量的总和,计算可使用的底层节点资源 $Res(S^n)$,即:

$$Res(S^n)=crn\cdot\sum CPU^s \tag{15}$$

其中 CPU^s 为底层网络 CPU 资源量,$\sum CPU^s$ 为底层网络 CPU 资源量总和.

在非饱和状态下,可以休眠底层节点,以达到节能的目的,计算休眠的底层节点数量 $sleep^n$ 的公式如下所示:

$$sleep^n=INT_(Res(S^n)-Res(VS^n))/E(B_s^n)) \tag{16}$$

其中,$Res(S^n)$为非饱和状态下可以使用的底层网络节点资源;crn 通过训练得到,针对不同的映射算法,crn 是不同的;$Res(VS^n)$通过公式(14)计算,表示一个时间窗的所有虚拟网络链路资源总量.

根据 $sleep^n$ 的计算结果,判断是否可以休眠节点:如果 $sleep^n$ 大于 0,说明可以通过休眠底层节点的方式节能;如果小于等于 0,则说明不能通过休眠节点的方式节能,否则将会影响系统收益以及虚拟网络接收率.

2.5.3 主动休眠节点和链路算法

主动休眠底层节点和链路需要注意两点:① 在非饱和状态下,休眠节点和链路应该考虑到底层网络的连通性,即,休眠节点的同时应该考虑休眠链路,在休眠链路的同时也应该考虑休眠节点,两者之间相互关联;② 在休眠底层节点和链路时,应满足休眠节点总数小于等于 $sleep^n$ 且休眠链路总数小于等于 $sleep^l$.设计主动休眠底层节点和链路算法如算法 2 所示.

© 中国科学院软件研究所 http://www.jos.org.cn

算法 2. 主动休眠底层节点和链路算法.

输出:休眠的节点和链路集合.

1. 根据公式(13)和公式(16),求出底层链路休眠数量 $sleep^l$ 和节点休眠数量 $sleep^n$;
   1.1. 根据第 1 个时间窗的虚拟网络请求、当前底层网络以及当前算法,调用算法 1,得到系统非饱和状态下的节点资源整体利用率 *crn* 和链路资源整体利用率 *crl*;
   1.2. 根据公式(11),设定 *mr* 等于 1,计算虚拟网络链路资源 $Res(VS^l)$;
   1.3. 根据公式(12),求出底层网络可以利用的链路资源 $Res(S^l)$;
   1.4. 根据公式(13),求出底层链路休眠数量 $sleep^l$;
   1.5. if ($sleep^l$ 小于等于 0),则不休眠任何链路,返回;
   1.6. 根据公式(14),设定 *mr* 等于 1,计算虚拟网络节点资源 $Res(VS^n)$;
   1.7. 根据公式(15),求出底层网络可以利用的节点资源 $Res(S^n)$;
   1.8. 根据公式(16),求出底层节点休眠数量 $sleep^n$;
   1.9. if ($sleep^n$ 小于等于 0),则不休眠任何节点,返回;
2. 初始化所有的节点和链路为激活状态;sln=0;snn=0;求出底层网络节点的度;
3. while ($sln<sleep^l$ & & $snn<sleep^n$) do
4. 找到激活的最小度的节点 *node*;if (没有找到有效的 *node*) break;
5. foreach (与 *node* 节点连接的所有激活底层链路 l_w) {
6. 休眠链路 l_w,并更新与该链路连接的两节点 *u* 和 *v* 的度,sln++;
7. if (节点 *u* 的度==0) {休眠 *u* 节点;snn++;}
8. if (节点 *v* 的度==0) {休眠 *v* 节点;snn++;}
9. if ($sln \geqslant sleep^l || snn \geqslant sleep^n$) break;
10. }
11. end while
12. return 休眠的底层节点和链路.

底层节点和链路休眠算法的关键是第 1 步,求出可休眠的底层节点和链路数量 $sleep^n$ 和 $sleep^l$.由训练算法计算系统非饱和状态下的 *crn* 和 *crl*;根据公式(11)~公式(13),求出底层链路休眠数量 $sleep^l$;根据公式(14)~公式(16),求出底层节点休眠数量 $sleep^n$.如果 $sleep^l$ 小于等于 0 或者 $sleep^n$ 小于等于 0,则说明难以通过休眠的方式在不影响映射收益比的前提下节约系统能耗.第 2 步,sln 为底层链路已经休眠的数量,snn 为底层节点已经休眠的数量,计算底层网络节点的度.第 3 步~第 11 步,找到最小度的激活节点,休眠该节点以及与该节点相连的链路,并记录休眠底层节点和链路数量,更新底层节点的度,这样可以保证底层网络的连通性.特别是第 3 步,应满足 $sln<sleep^l$ & & $snn<sleep^n$ 条件,表示休眠底层节点或链路必须同时不能超过通过公式(13)和公式(16)计算的数量,否则将会影响收益成本比、系统收益及虚拟网络接收率.

算法 2 保证了底层网络的连通性,同时能够尽可能主动地休眠底层节点和链路数量,达到节能的目的.算法 2 的算法时间复杂度为 $o(l\cdot n)$,l 为底层网络链路数量,n 为底层网络节点数量.

2.6 性能评价指标

主动休眠底层节点和链路的方法可提高底层网络资源休眠数量,节约系统能耗,设定能量消耗、休眠节点数量、休眠链路数量和收益成本比为性能评价指标.

2.6.1 能量消耗

底层网络的能量消耗由节点和链路能耗两部分组成.虚拟网络映射将会导致底层网络的节点和链路处于激活状态或者休眠状态,且底层网络节点负载随着虚拟节点的映射而发生变化,其处理器的利用率将会产生变化,从而影响底层网络的能耗.

定义在第 *t* 个时间窗底层网络能耗为

© 中国科学院软件研究所 http://www.jos.org.cn

$$P(t)=\sum_{i\in N^s}PN^i+\sum_{j\in L^s}PL^j.$$

定义底层网络在总共 T 个时间窗内能耗总量为

$$PSum(T)=\sum_{t=1}^{T}P(t).$$

Tn 为一个时间窗占用的时间单元,定义平均每个时间单元能耗为

$$PAver=\lim_{T\to\infty}\frac{PSum(T)}{T\cdot Tn}.$$

$PAver$ 为比较不同映射算法能耗的重要性能指标,在系统非饱和状态下,应尽量提高休眠底层节点和链路数量,降低 $PAver$.

2.6.2 休眠节点数量

底层节点分为休眠节点和激活节点.休眠节点是指未承载任何虚拟节点,与之相连的底层链路未承载任何虚拟链路.激活节点分为两类:第 1 类是指既承载了虚拟节点,与之相连的链路最少也承载一条虚拟链路,称为活动节点;第 2 类是指未承载虚拟节点,但与之相连的链路最少承载一条虚拟链路,称为传输节点.活动节点能耗为 $P_b+P_l\cdot u$,传输节点能耗为 P_b,休眠节点能耗为 0.

定义:

$$SLN^i=\begin{cases}1, & \text{如果}i\text{为活动节点}\\ 0, & \text{否则}\end{cases}.$$

定义在第 t 个时间窗底层休眠节点数量为

$$SleepN(t)=\sum_{i\in N^s}SLN^i.$$

定义底层网络在总共 T 个时间窗内,底层休眠节点总量为

$$SleepNSum(T)=\sum_{t=1}^{T}SleepN(t).$$

定义每个时间窗的平均休眠节点数量为

$$SleepN=\lim_{T\to\infty}\frac{SleepNSum(T)}{T}.$$

2.6.3 休眠链路数量

底层链路分为休眠链路和激活链路.休眠链路是指未承载任何虚拟链路的底层链路.激活链路是指最少承载一条虚拟链路的底层链路.活动链路能耗为 P_n,休眠链路能耗为 0.

定义:

$$SLL^j=\begin{cases}1, & \text{如果}j\text{为激活链路}\\ 0, & \text{否则}\end{cases}.$$

定义在第 t 个时间窗底层休眠链路数量为

$$SleepL(t)=\sum_{j\in L^s}SLL^j.$$

定义底层网络在总共 T 个时间窗内,底层休眠链路总量为

$$SleepLSum(T)=\sum_{t=1}^{T}SleepL(t).$$

定义每个时间窗的平均休眠链路数量为

$$SleepL=\lim_{T\to\infty}\frac{SleepLSum(T)}{T}.$$

2.6.4 收益成本比

定义系统收益:

$$R(G^v(t))=\sum_{l^v\in L^v}bw(l^v)+\sum_{n^v\in N^v}CPU(n^v),$$

© 中国科学院软件研究所 http://www.jos.org.cn

其中, $\sum_{l^v \in L^v} bw(l^v)$ 为接收的虚拟网络链路带宽总和, $\sum_{n^v \in N^v} CPU(n^v)$ 为接收的虚拟网络节点 CPU 总和.

定义映射代价:

$$C(G^v(t)) = \sum_{l^v \in L^v} \sum_{l^s \in L^s} bw(l^v, l^s) + \sum_{n^v \in N^v} CPU(n^v),$$

其中, $\sum_{l^v \in L^v} \sum_{l^s \in L^s} bw(l^v, l^s)$ 为底层网络所消耗的带宽总和.

定义长期平均收益成本比:

$$r/c = \lim_{T \to \infty} \frac{\sum_{t=1}^{T} R(G^v(t))}{\sum_{t=1}^{T} C(G^v(t))}.$$

3 虚拟网络映射主动休眠节点节能仿真

本节首先介绍仿真环境、比较算法和构建的字典库,然后介绍参数训练、节能和其他方面的比较.其中,系统节能仿真在非饱和状态下进行,从休眠节点数量、休眠链路数量和能量消耗这 3 个方面讨论本文所提算法的节能性能.

3.1 仿真环境

由于网络虚拟化是一个新兴领域,底层网络与虚拟网络请求的实际特征还没有得到充分掌握,因此,本文采用对称网络拓扑结构.采用 GT-ITM 工具[25]随机生成一个由 100 个节点、570 条链路组成的底层网络拓扑,相当于一个中等规模的物理网络.底层网络节点 CPU 资源和链路带宽资源服从 50-100 的均匀分布.每个时间窗为 100 个时间单元.虚拟网络请求过程模拟泊松过程,在每个时间窗内虚拟网络请求到达个数服从均值为 20 的泊松分布,每个虚拟网络的生存时间服从均值为 5 个时间窗的指数分布,每个虚拟网络请求节点个数服从 2-20 的均匀分布,每对虚拟网络节点以 0.5 的概率相连.虚拟网络节点 CPU 资源与链路带宽资源需求服从 0-6 的均匀分布.每个虚拟网络请求等待映射的时间为 1 个时间窗.每次模拟实验运行约 50 000 个时间单元,包含 10 000 个虚拟网络请求.根据文献[22,26],本文设置第 1.2 节中定义的 P_b,P_m 和 P_n 分别为 150W,300W 和 15W;由于不同环境下的链路功耗差异较大[27,28],本文比较了链路功耗下在 1W,15W,30W 的系统能耗,如后文图 6 所示.为了提高实验结果的置信水平,所有的实验运行 10 组不同虚拟网络请求,然后取平均值作为结果.

3.2 比较算法

虚拟网络映射问题是 NP-hard 问题,随着网络规模(包括虚拟网络和底层网络)的扩大,基于整数规划的虚拟网络映射求解难以满足虚拟网络映射的实时性要求,因此,本文不与整数规划的方法进行比较,而选择基于能耗感知的启发式算法[17]与元启发式算法[19]以及经典的虚拟网络映射启发式算法[29],即,EA-VNE[17],ACO-PE-VNE[19]及 PR-VNE[29]进行比较,并在 EA-VNE,ACO-PE-VNE 和 PR-VNE 上应用了本文提出的主动休眠节点和链路方法,即,AS-EA-VNE,AS-ACO-PE-VNE 和 AS-PR-VNE.具体实现如下:首先统计第 1 个时间窗内所有的虚拟网络请求动态特征参数,然后根据算法 2 主动休眠底层节点和链路;再分别采用 EA-VNE,ACO-PE-VNE 或者 PR-VNE 算法,映射限制在未主动关闭的底层节点和链路集合中.特别说明的是:每次映射虚拟网络请求不需要重新主动休眠底层节点和链路,主动休眠只发生在第 1 个时间窗,且不同算法采用不同的单路径链路映射.

3.3 虚拟网络映射字典库

根据比较算法,虚拟网络映射字典库记录 EA-VNE,ACO-PE-VNE 和 PR-VNE 算法的映射结果.底层网络采用第 3.1 节中介绍的由 100 个节点、570 条链路组成的对称网络拓扑结构.虚拟网络与第 3.1 节的仿真环境基本相同,所不同的是,虚拟网络节点 CPU 资源与链路带宽资源需求分别服从 0-4,0-5,0-6,0-7,0-8,0-9,0-10,0-11,0-12,0-13 的均匀分布,意味着节点和链路资源量的期望值为 2,2.5,3,3.5,4,4.5,5,5.5,6,6.5,即,有 10 个不同动态特征的虚拟网络请求.每个相同特征的虚拟网络创建 10 组不同实例,总共有 100 组不同实例;每组实例有 10 000

© 中国科学院软件研究所 http://www.jos.org.cn

个虚拟网络请求,总共映射了 1×10^6 个虚拟网络请求;3 种算法在 100 组虚拟网络实例上映射,数据字典总共记录了 300 组虚拟网络映射结果.

3.4 仿真结果

3.4.1 参数训练

1) 在保证 *mr*=1 的条件下,底层网络资源利用率存在最大值.

图 2 显示随着节点 CPU 和链路带宽的期望值的增加,数据字典库中 3 种算法的接收率、链路资源利用率以及节点资源利用率的变化趋势,即:随着节点和链路资源量期望值的增长,链路和节点的资源利用率随之增长,虚拟网络接收率保持 100%,直到节点和链路资源量在 4.5 时出现下降趋势.这给了我们一个启示:在非饱和状态,底层网络能够接收虚拟网络资源存在最大值,表现在节点和链路资源利用率上,即 EA-VNE,PR-VNE 和 ACO-PE-VNE 的链路资源利用率最大值分别在 0.298 539,0.311 195 和 0.325 887 附近,节点资源利用率最大值分别在 0.592 600 4,0.669 519 8 和 0.669 492 2 附近.

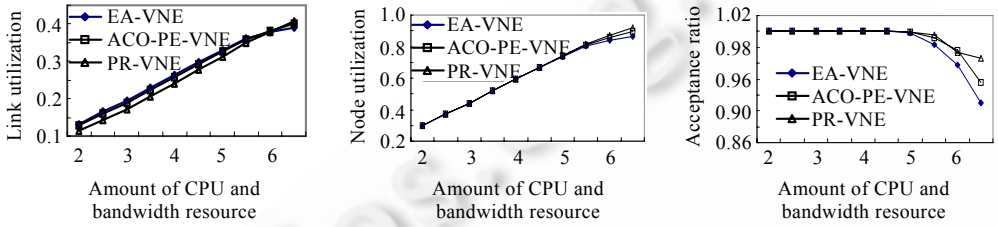


Fig.2 Average node utilization, link utilization and acceptance ratio in dictionary database

图 2 字典库的节点和链路资源利用率以及接收率

2) *mr* 与 *Nov* 对资源利用率训练值的影响.

图 3 中图例对应着[*mr*,*Nov*],例如,[0.999,20]表示 *mr* 的范围为[0.999,1]以及 *Nov* 为 20,意味着算法 1 统计虚拟网络接收率在 0.999 以上的字典库数据项,取 20 个数据项的平均值作为训练的 *crn* 和 *crl*.从图 3 可以看出:随着 *mr* 范围的扩大以及 *Nov* 的减小,所训练的 *crn* 和 *crl* 逐渐增大.当 *mr* 范围为[0.9999,1]、*Nov* 为 20 时,资源利用率最小,EA-VNE,PR-VNE 和 ACO-PE-VNE 训练的 *crl* 分别为 0.2871 194,0.291 995 25 和 0.307 2768 5,*crn* 分别为 0.644 253 35,0.700 267 35 和 0.623 920 05;当 *mr* 范围为[0.999,1]、*Nov* 为 10 时,资源利用率最大,EA-VNE,PR-VNE 和 ACO-PE-VNE 训练的 *crl* 分别为 0.330 187 7,0.348 522 9 和 0.352 377 2,*crn* 分别为 0.738 213 2,0.813 139 9 和 0.797 208 9.

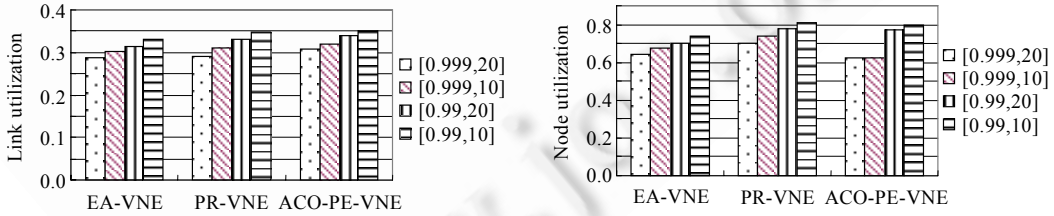


Fig.3 Average node and link utilization of different *mr* and *Nov* value

图 3 不同 *mr* 和 *Nov* 下的链路与节点资源利用率

3) 算法 2 能够训练出有效的主动休眠节点和链路数量.

在 *mr*,*Nov* 不同取值下,主动休眠节点和链路数量的期望值与标准差见表 1,可以看出:随着 *mr* 范围的扩大、*Nov* 取值的减小,主动休眠的底层链路和节点数量逐渐增多;不同算法主动休眠底层节点和链路数量是不同的;3 种算法的休眠链路和节点数量具有相近的标准差,这表明训练方法能够适应多种算法.

© 中国科学院软件研究所 http://www.jos.org.cn

Table 1 Expected value and standard deviation of the training number of active-sleeping nodes and links

表 1 训练的主动休眠节点和链路数量期望值与标准差

mr	*Nov*	算法	休眠链路数量期望值	休眠节点数量期望值	休眠链路数量标准差	休眠节点数量标准差
[0.999,1]	20	EA-VNE	121.54	29.64	14.92	12.19
		PR-VNE	124.47	35.1	14.16	11.57
		ACO-PE-VNE	133.18	27.46	14.16	11.57
[0.999,1]	10	EA-VNE	129.8	32.77	14.93	12.18
		PR-VNE	135.41	38.99	14.16	11.57
		ACO-PE-VNE	139.85	27.36	14.15	11.55
[0.99,1]	20	EA-VNE	136.27	35.29	14.9	12.2
		PR-VNE	146.05	42.69	14.16	11.57
		ACO-PE-VNE	151.26	41.9	14.14	11.59
[0.99,1]	10	EA-VNE	146.09	39.03	14.9	12.19
		PR-VNE	156.69	46.38	14.14	11.53
		ACO-PE-VNE	158.89	44.79	14.16	11.57

3.4.2 节能比较

1) 本文提出的主动休眠方法显著提高了底层网络节点和链路休眠数量,很大程度地减少了系统能耗.

图 4 表示在 *mr* 范围为[0.999,1]、*Nov* 为 20、P_n 为 15W 的环境下,平均运行 51~504 个时间窗底层休眠节点和链路数量以及系统能耗的变化情况.图 4(a)、图 4(b)表明,AS-EA-VNE,AS-PR-VNE 和 AS-ACO-PE-VNE 算法较 EA-VNE,PR-VNE 和 ACO-PE-VNE 至少提高了 25%的休眠节点数量和 10.9%的链路数量.图 4(c)显示,AS-EA-VNE, AS-PR-VNE 和 AS-ACO-PE-VNE 的系统能耗较 EA-VNE,PR-VNE 和 ACO-PE-VNE 分别降低了 11.7%,6.7%和 14.5%.这是因为在非饱和状态下,基于主动休眠方法把虚拟网络映射在一个较小的节点和链路集合中,使得其他底层节点和链路能够休眠,从而提高了休眠节点和链路数量,降低了系统能耗.

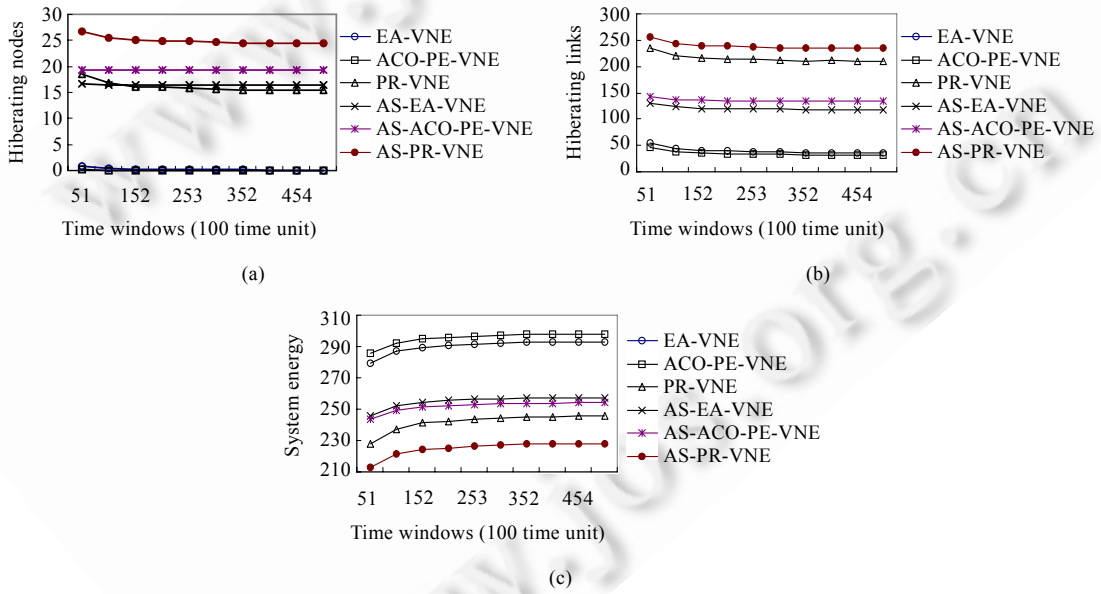


Fig.4 Average number of hiberating nodes and links, and energy consumption

图 4 休眠节点、链路数量及系统能耗

2) 主动休眠节点和链路数量对系统能耗的影响.

图 5 给出 P_n 为 15W 环境下运行 500 个时间窗的系统能耗情况,图例中的 0/0,100/20 和 200/30 表示主动休眠底层链路数量/主动休眠底层节点数量.图 5 表明:随着主动休眠底层链路和节点数量的增大,底层网络能耗逐

© 中国科学院软件研究所 http://www.jos.org.cn

渐降低.即:底层网络主动休眠链路和节点数量越多,底层网络越节能.0/0 表示未主动休眠底层链路和节点,AS-EA-VNE,AS-ACO-PE-VNE,AS-PR-VNE 算法即是 EA-VNE,ACO-PE-VNE,PR-VNE 算法.在主动休眠 200 条链路和 30 个节点时,系统能耗最少,AS-EA-VNE,AS-ACO-PE-VNE,AS-PR-VNE 能耗比 EA-VNE,ACO-PE-VNE,PR-VNE 分别降低了 23.8%,24.3%和 13.3%.

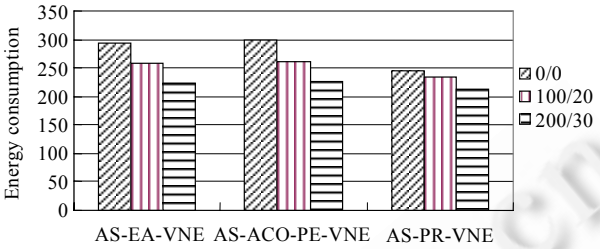


Fig.5 Energy consumption in different hiberating number

图 5 主动休眠数量对底层网络能耗的影响

3) 底层链路功耗参数对能耗的影响.

图 6 给出在 *mr* 范围为[0.999,1]、*Nov* 为 20 环境下底层链路功耗分别在 1W,15W,30W 下的系统能耗情况.图 6 显示出:链路能耗越大,系统能耗越多;主动休眠算法在不同的链路能耗下都能够实现节能目标,这是由于主动休眠方法使得更多的底层资源可以进入休眠节能状态.当链路能耗在 1W 时,AS-EA-VNE,AS-ACO-PE-VNE,AS-PR-VNE 的能耗比 EA-VNE,ACO-PE-VNE,PR-VNE 分别降低了 11%,14.8%和 5.8%;当链路能耗在 30W 时,AS-EA-VNE,AS-ACO-PE-VNE,AS-PR-VNE 的能耗比 EA-VNE,ACO-PE-VNE,PR-VNE 分别降低了 12.8%,17.4%和 5.7%.

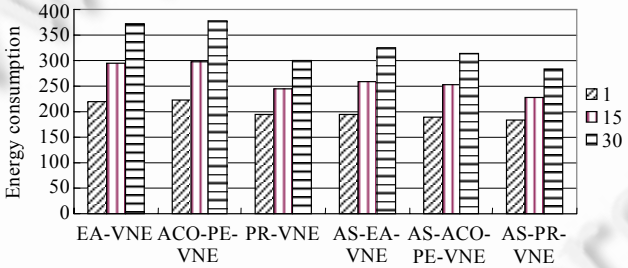


Fig.6 Energy consumption in different link power

图 6 不同链路功耗对系统能耗的影响

3.4.3 其他性能比较

1) 主动休眠算法对收益成本比的影响.

图 7 给出在 *mr* 范围为[0.999,1]、*Nov* 为 20 环境下不同映射算法的收益成本比情况,可以看出:AS-EA-VNE,AS-ACO-PE-VNE 比 EA-VNE,ACO-PE-VNE 提高了 2%.这是因为在非饱和状态下,映射成本比受到链路映射影响,主动休眠方法缩小了可利用的底层资源范围,从而缩短了链路映射路径长度,提高了收益成本比.AS-PR-VNE 相对于 PR-VNE 收益成本比降低了 0.3%,这是由于,PR-VNE 算法的虚拟节点映射集中在资源丰富的底层节点上,主动休眠算法使得底层资源量减少,导致了收益成本比有所下降,但几乎可以忽略.

2) 主动休眠算法减少了虚拟网络映射算法运行时间.

在 *mr* 范围为[0.999,1]、*Nov* 为 20 的条件下,运行 500 个时间窗所用的时间,EA-VNE,PR-VNE,ACO-PE-VNE,AS-EA-VNE,AS-PR-VNE 和 AS-ACO-PE-VNE 分别为 59s,49s,1084s,50s,42s 和 987s.可以看出,主动休眠算法缩短了运行时间.这是因为其缩小了虚拟网络映射的底层节点和链路集合,缩短了节点和链路的映射时间;且仅利

© 中国科学院软件研究所 http://www.jos.org.cn

用了第 1 个时间窗的虚拟网络请求训练资源利用率,算法时间复杂度低,对运行时间影响很小.

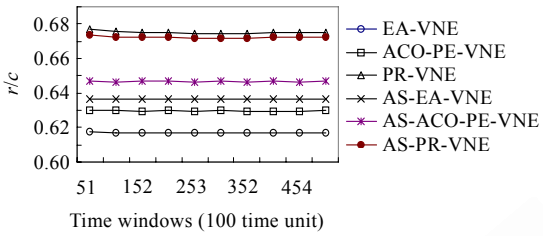


Fig.7 Long-Term *r*/*c* ratio (revenue/cost) comparison

图 7 收益成本比

4 结束语

本文研究了网络虚拟化环境下的系统能耗问题.提出多目标决策的虚拟网络映射能耗模型;利用虚拟网络映射动态特征,构造虚拟网络映射字典库,提出底层网络资源利用率的训练算法以及主动休眠底层节点和链路算法,把虚拟网络映射在一个较小的节点和链路集合中,提高休眠节点和链路数量,以实现高效节能.仿真实验结果验证了在系统非饱和状态下,基于主动休眠节点和链路的方法能够有效节约系统消耗能量,与其他算法相比,显著降低了系统能耗.然而,基于资源整合的网络能耗管理与虚拟网络请求动态行为紧密相关,构建有效模型描述系统动态行为特征,使得系统在饱和与非饱和状态之间有效切换有待进一步加以研究.同时,鉴于系统能耗与节点的利用率有关,基于节点分配的虚拟网络映射节能方法也有待深入研究.

致谢 本文仿真实验在 C3S2 服务器上完成,感谢湖州师范学院 C3S2 计算中心的大力支持.

References:

[1] Fisher W, Suchara M, Rexford J. Greening backbone networks: Reducing energy consumption by shutting off cables in bundled links. In: Proc. of the ACM SIGCOMM Workshop on Green Networking. 2010. 29−34. [doi: 10.1145/1851290.1851297]

[2] Barroso LA, Holzle U. The case for energy-proportional computing. IEEE Computer, 2007,40(12):33−37. [doi: 10.1109/MC.2007.443]

[3] Bohrer P, Elnozahy EN, Keller T, Kistler M, Lefurgy C, McDowell C, Rajamony R. The case for power management in Web servers. In: Graybill R, ed. Power Aware Computing. Norwell: Kluwer Academic Publishers, 2002. 261−289. http://dl.acm.org/citation.cfm?id=783075

[4] Lin C, Tian Y, Yao M. Green network and green evaluation: Mechanism, modeling and evaluation. Chinese Journal of Computers, 2011,34(4):593−612 (in Chinese with English abstract). [doi: 10.3724/SP.J.1016.2011.00593]

[5] Ye KJ, Wu ZH, Jiang XH, He QM. Power management of virtualized cloud computing platform. Chinese Journal of Computers, 2012,35(6):1262−1285 (in Chinese with English abstract).

[6] Chowdhury NMMK, Boutaba R. Network virtualization: State of the art and research challenges. IEEE Communications Magazine, 2009,47(7):20−26. [doi: 10.1109/MCOM.2009.5183468]

[7] Anderson T, Peterson L, Shenker S, Turner J. Overcoming the Internet impass through virtualization. IEEE Computer Magazine, 2005,38(4):34−41. [doi: 10.1109/MC.2005.136]

[8] Sun G, Anand V, Yu HF, Liao D, Li L. Optimal provisioning for elastic service oriented virtual network request in cloud computing. In: Proc. of the Global Communications Conf. (GLOBECOM). Anaheim: IEEE, 2012. 2517−2522. [doi: 10.1109/GLOCOM.2012.6503495]

[9] Drutskoy D, Keller E, Rexford J. Scalable network virtualization in software-defined networks. IEEE Internet Computing, 2013, 17(2):20−27. [doi: 10.1109/MIC.2012.144]

© 中国科学院软件研究所 http://www.jos.org.cn

[10] Sharkh MA, Jammal M, Shami A, Ouda A. Resource allocation in a network-based cloud computing environment: Design challenges. IEEE Communications Magazine, 2013,51(11):46−52. [doi: 10.1109/MCOM.2013.6658651]

[11] Wei XL, Chen M, Fan JH, Zhang GM, Lu ZY. Architecture of the data center network. Ruan Jian Xue Bao/Journal of Software, 2013,24(2):295−316 (in Chinese with English abstract). http://www.jos.org.cn/1000-9825/4336.htm [doi: 10.3724/SP.J.1001.2013.04336]

[12] Fischer A, Botero JF, Beck MT, Meer HD, Hesselbach X. Virtual network embedding: A survey. IEEE Communications Surveys & Tutorials, 2013,15(4):1888−1906. [doi: 10.1109/SURV.2013.013013.00155]

[13] Chabarek J, Sommers J, Barford P, Estan C, Tsiang D, Wright S. Power awareness in network design and routing. In: Proc. of the INFOCOM. Phoenix: IEEE, 2008. 457−465. [doi: 10.1109/INFOCOM.2008.93]

[14] Botero JF, Hesselbach X, Duelli M, Schlosser D, Fischer A, Meer HD. Energy efficient virtual network embedding. IEEE Communications Letters, 2012,16(5):756−759. [doi: 10.1109/LCOMM.2012.030912.120082]

[15] Garroppo R, Nencioni G, Tavanti L, Scutella MG. Does traffic consolidation always lead to network energy savings. IEEE Communication Letters, 2013,17(9):1852−1855. [doi: 10.1109/LCOMM.2013.070913.131244]

[16] Botero JF, Hesselbach X. Greener networking in a network virtualization environment. Computer Networks, 2013,57(9):2021−2039. [doi: 10.1016/j.comnet.2013.04.004]

[17] Su S, Zhang ZB, Cheng X, Wang YW, Luo Y, Wang J. Energy-Aware virtual network embedding through consolidation. In: Proc. of the Computer Communications Workshops (INFOCOM WKSHPS). Orlando: IEEE, 2012. 127−132. [doi: 10.1109/INFCOMW.2012.6193473]

[18] Wang B, Chang XL, Liu JQ, Muppala JK. Reducing power consumption in embedding virtual infrastructures. In: Proc. of the Globecom Workshops (GC Wkshps). Anaheim: IEEE, 2012. 714−718. [doi: 10.1109/GLOCOMW.2012.6477662]

[19] Chang XL, Wang B, Liu JQ, Muppala JK. Green cloud virtual network provisioning based ant colony optimization. In: Proc. of the 15th Annual Conf. Companion on Genetic and Evolutionary Computation. New York: ACM Press, 2013. 1553−1560. [doi: 10.1145/2464576.2482735]

[20] Yu ML, Yi Y, Rexford J, Chiang M. Rethinking virtual network embedding: Substrate support for path splitting and migration. ACM SIGCOMM Computer Communication Review, 2008,38(2):17−29. [doi: 10.1145/1355734.1355737]

[21] Turner JS, Crowley P, DeHart J, Freestone A, Heller B, Kuhns F, Kumar S, Lockwood J, Lu J, Wilson M, Wiseman C, Zar D. Supercharging planetlab: A high performance, multi-application, overlay network platform. ACM SIGCOMM Computer Communication Review, 2007,37(4):85−96. [doi: 10.1145/1282427.1282391]

[22] Lu GH, Guo CX, Li YL, Zhou ZQ, Yuan T, Wu HT, Xiong YQ, Gao R, Zhang YG. Serverswitch: A programmable and high performance platform for data center networks. In: Andersen D, ed. Proc. of the 8th USENIX Conf. on Networked Systems Design and Implementation. Berkeley: USENIX Association, 2011. 1−14.

[23] Unnikrishnan D, Vadlamani R, Liao Y, Dwaraki A, Crenne J, Gao LX, Tessier R. Scalable network virtualization using FPGAs. In: Proc. of the 18th Annual ACM/SIGDA Int'l Symp. on Field Programmable Gate Arrays. New York: ACM Press, 2010. 219−228. [doi: 10.1145/1723112.1723150]

[24] Sivaraman V, Vishwanath A, Zhao Z, Russell C. Profiling per-packet and per-byte energy consumption in the NetFPGA gigabit router. In: Proc. of the Computer Communications Workshops (INFOCOM WKSHPS). Shanghai: IEEE, 2011. 331−336. [doi: 10.1109/INFCOMW.2011.5928833]

[25] Zegura EW, Calvert KL, Bhattacharjee S. How to model an internetwork. In: Proc. of the 15th Annual Joint Conf. of the IEEE Computer and Communications Societies Conf. on Computer Communications. San Francisco: IEEE, 1996. 594−602. [doi: 10.1109/INFCOM.1996.493353]

[26] Barroso LA, Hölzle U. The Datacenter as a Computer: An Introduction to the Design of Warehouse-Scale Machines. Morgan and Claypool Publishers, 2009. 1−108. [doi: 10.2200/S00193ED1V01Y200905CAC006]

[27] Ananthanarayanan G, Katz RH. Greening the switch. In: Proc. of the 2008 Conf. on Power Aware Computing and Systems. Berkeley: USENIX Association, 2008. 1−14. https://www.usenix.org/legacy/event/hotpower08/tech/

[28] Chiaraviglio L, Mellia M, Neri F. Energy-Aware backbone networks: A case study. In: Proc. of the Int'l Conf. on Communications Workshop. Dresden: IEEE, 2009. 1−5. [doi: 10.1109/ICCW.2009.5208038]

© 中国科学院软件研究所 http://www.jos.org.cn

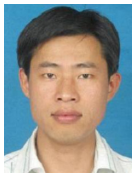
[29] Cheng X, Su S, Zhang ZB, Wang HC, Yang FC, Luo Y, Wang J. Virtual network embedding through topology-aware node ranking. ACM SIGCOMM Computer Communication Review, 2011,41(2):38−47. [doi:10.1145/1971162.1971168]

附中文参考文献:

[4] 林闯,田源,姚敏.绿色网络和绿色评价:节能机制、模型和评价.计算机学报,2011,34(4):593−612. [doi: 10.3724/SP.J.1016.2011.00593]

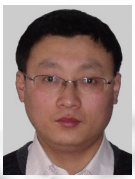
[5] 叶可江,吴朝晖,姜晓红,何钦铭.虚拟化云计算平台的能耗管理.计算机学报,2012,35(6):1262−1285.

[11] 魏祥麟,陈鸣,范建华,张国敏,卢紫毅.数据中心网络的体系结构.软件学报,2013,24(2):295−316. http://www.jos.org.cn/1000-9825/4336.htm [doi: 10.3724/SP.J.1001.2013.04336]



陈晓华(1977－),男,江西九江人,博士生,讲师,主要研究领域为下一代互联网,云计算.
E-mail: 97122046@sina.com

李春芝(1982－),女,博士生,讲师,主要研究领域为最优化方法.
E-mail: lichunzhi82@126.com



陈良育(1980－),男,博士,副教授,主要研究领域为并行计算,符号计算,自动推理.
E-mail: lychen@sei.ecnu.edu.cn

曾振柄(1963－),男,博士,教授,博士生导师,主要研究领域为数学机械化,组合优化,人工智能.
E-mail: zbzeng@shu.edu.cn

© 中国科学院软件研究所 http://www.jos.org.cn