第 45 卷 第 5 期 计 算 机 工 程 2019 年 5 月
Vol. 45 No. 5 Computer Engineering May 2019

·人工智能及识别技术· 文章编号：1000-3428(2019)05-0135-08 文献标志码：A 中图分类号：TP399

一种基于机器学习的虚拟机放置方法

郭良敏[a,b]，高俊杰[a,b]，胡桂银[a,b]

（安徽师范大学 a. 计算机与信息学院；
b. 网络与信息安全安徽省重点实验室，安徽 芜湖 241003）

摘 要：为改善云数据中心的能耗、负载均衡性和服务等级协议（SLA）违背率，对虚拟机放置策略进行优化。基于 IaaS 环境，提出一种基于机器学习的虚拟机迁移调整方法。根据资源消耗的互补性和不均衡性对虚拟机进行预放置，使用深度神经网络预测物理机负载等级，并利用深度 Q 网络调整物理机数量。实验结果表明，该方法能够有效均衡负载分布，降低能源开销和 SLA 违背率。

关键词：虚拟机放置；机器学习；能耗；负载均衡；服务等级协议

中文引用格式：郭良敏，高俊杰，胡桂银. 一种基于机器学习的虚拟机放置方法[J]. 计算机工程，2019，45(5)：135-142.
英文引用格式：GUO Liangmin，GAO Junjie，HU Guiyin. A method for virtual machine placement based on machine learning[J]. Computer Engineering，2019，45(5)：135-142.

A Method for Virtual Machine Placement Based on Machine Learning

GUO Liangmin[a,b]，GAO Junjie[a,b]，HU Guiyin[a,b]

（a. School of Computer and Information；b. Anhui Provincial Key Laboratory of Network and Information Security，Anhui Normal University，Wuhu，Anhui 241003，China）

【Abstract】 In order to improve energy consumption，load balance，and Service Level Agreement（SLA） violation rate of cloud data centers，it is necessary to optimize virtual machine placement strategy. Therefore，based on the IaaS environment，a virtual machine migration adjustment method based on machine learning is proposed. The virtual machine is pre-placed according to the complementarity and imbalance of resource consumption，the deep neural network is used to predict the physical machine load level，and the Deep Q Network（DQN） is used to adjust the number of physical machines. Experimental results show that this method can effectively balance load distribution，reduce energy cost and SLA violation rate.

【Key words】 virtual machine placement；machine learning；energy consumption；load balance；Service Level Agreement（SLA）

DOI：10.19678/j.issn.1000-3428.0052354

0 概述

云计算是一种按需计费模式，租户可从云服务提供商购买所需的计算资源。云服务提供商将过剩的资源作为产品销售供应给租户，以获取收益。随着云计算行业的发展、租户的增多、数据中心规模的扩大，云服务提供商的运营成本逐渐增加。因此，能耗、负载均衡、服务等级协议（Service Level Agreement，SLA）违背率成为数据中心重要的 3 项指标。

数据中心能耗涉及多个方面，主要有降温系统、硬件设备等因素[1]。为了节能，对虚拟机放置策略进行优化是有效的途径之一。其中，物理机的能耗主要有 CPU、内存等，且能耗与 CPU 利用率之间存在一定的关系[2]。

负载均衡技术能够使各物理机不出现负载过高或过低的情况，以提高云系统的可延展性和可靠性[3]。当虚拟机运行时，负载动态变化需进行如下权衡：如何权衡尽可能少的物理机数量和尽可能优的负载均衡性能；如何达到单个物理机各资源的负载均衡和整个数据中心所有物理机的负载均衡；在虚拟机负载动态变化的过程中如何维持负载的相对均衡。

SLA 是云服务提供商与用户签订的协议，以保证租户权益。除提高硬件可靠性外，优化虚拟机的放置策略以减少物理机过载是降低 SLA 违背率的有效途径之一。

基金项目：安徽省自然科学基金（1908085MF190，1508085QF133，1808085MF172）。
作者简介：郭良敏（1980—），女，副教授、博士，主研方向为云计算、服务推荐、信息安全；高俊杰，学士；胡桂银，讲师、硕士。
收稿日期：2018-08-10 **修回日期：**2018-09-11 **E-mail：**lmguo@ustc.edu.cn

为权衡上述 3 项指标,本文提出基于机器学习的虚拟机放置方法。该方法利用互补性和预放置方法来均衡负载,基于深度 Q 网络(Deep Q Network,DQN)的迁移整合方法,优化初次放置结果,并调整物理机数量,以节约能耗和降低 SLA 违背率。

1 相关工作

目前,国内外提出多种以优化能耗、负载均衡等为目标的虚拟机放置方法,具体描述如下:

1)以负载均衡为目标。文献[4]使用蚁群算法解决了工作负载整合问题,并将虚拟机放置看作多维度装箱问题,在每个蚂蚁收到所有包(虚拟机)和所有箱(物理机)后,根据用于描述蚂蚁优化工作参数的可能决策规则,选择物理机,并更新信息素的数量。文献[5]以物理机内资源的均衡和物理机间的负载均衡两者为目标,通过混合在线迁移解决了多维度资源的负载均衡问题。

2)以节能为目标。文献[6]结合基于预测的 A * 算法和模糊集合技术,提出对数据中心代价和碳排放同时优化的算法,但是该算法基于阈值法的虚拟机放置,存在资源浪费问题。文献[7]将虚拟机放置问题转化为多维度装箱问题,以节能为目标,并提高物理机资源利用率,运用蚁群算法最优化放置策略。文献[8]提出半静态虚拟机整合策略,以减少迁移次数,并降低能耗。文献[9]提出基于负载高峰特征的虚拟机放置算法,对互补高峰的虚拟机进行配对,提高资源利用率,减少物理机数量,但是仅考虑负载已知的静态情况。

3)以降低 SLA 违背率为目标。文献[10]提出对虚拟机负载分类的方法,使用提出的预测算法管理虚拟机的迁移,相比于静态方法减少了物理机数量并降低了 SLA 违背率。文献[11]提出面向 SLA 的基于市场供应策略的应用程序资源分配方法,证明了云计算中基于 SLA 的资源配置的可行性和有效性。文献[12]基于 SaaS 建立了租户需求模型,提出了与租户相关的特征信息,并基于历史数据挑选出高级特征,对其做出短期预测,并根据预测上升或下降进行分类,根据动态资源放置算法为租户选择虚拟机,并使用最佳适应下降算法,为虚拟机挑选物理机,可以减少给虚拟机分配资源的停机时间,减少 SLA 违背的可能性。

4)以能耗、负载均衡和 SLA 违背率中的 2 项指标为目标。文献[13]提出多维空间分割模型,将物理机评级为多个不同的优先级,虚拟机将被放置在最高优先级的物理机上,并提出能耗模型和物理机内资源负载均衡度量模型,但此方法本质是贪婪算法,放置策略仅根据当前数据中心状态决定,没有考虑虚拟机负载变化。文献[14]提出基于强化学习的动态合并算法,通过关闭低负载的物理机和迁出虚拟机以减少能耗和 SLA 违背率,将能耗和 SLA 违背率作为惩罚函数,使得算法不需要先前知识就可以自行学习物理机的管理策略。文献[15]提出基于最佳适应递减的虚拟机放置算法,以节约能耗、降低 SLA 违背率。文献[16]提出一种基于学习自动机的算法,使用用户需求的变化来预测物理机是否可能发生过载,并阻止过载的发生,该算法提高了资源利用率,降低了能耗和 SLA 违背率。文献[17]提出一种自适应的虚拟机整合方法,以节约能耗和降低 SLA 违背率。

上述方法多数没有综合考虑 3 项指标的权衡,因此,本文提出基于机器学习的虚拟机放置方法,该方法将放置过程分成 2 个阶段,以节约能耗、均衡负载和降低 SLA 违背率,具体框架如图 1 所示。

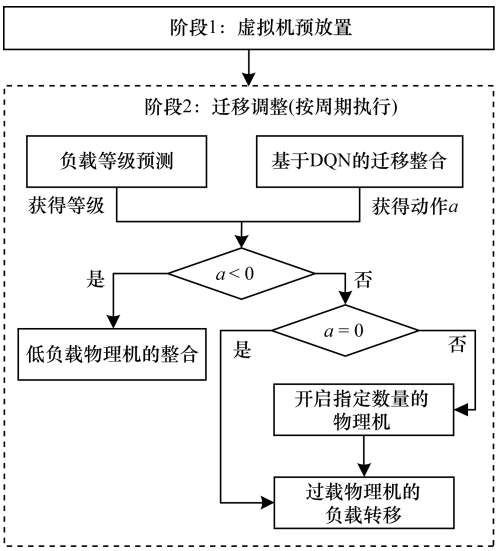

图 1 基于机器学习的虚拟机放置框架

2 基于互补性和不均衡性感知的虚拟机预放置

当租户购买虚拟机资源后,数据中心需为其选择物理机。由于在初始时虚拟机没有历史负载数据,无法直接预测负载,且初始时数据中心有一定的弹性空间为虚拟机提供资源量,因此本文按租户申请的最大资源量,并依据资源消耗的互补性和负载的不均衡性对虚拟机进行预放置,当虚拟机运行一段时间后,再进行预测和调整,一定程度上减少 SLA 违背率。

2.1 预放置模型

预放置时主要考虑 CPU 和内存资源。设数据中心有 n 个物理机,分别为 $pm_1,pm_2,\cdots,pm_n$,待放置虚拟机有 m 个,分别为 $vm_1,vm_2,\cdots,vm_m$,因此,可将虚拟机预放置问题描述如下:

$$x_{i,j}\in\{0,1\},\ i=1,2,\cdots,n,\ j=1,2,\cdots,m \tag{1}$$

$$\sum_{i=1}^{n}x_{i,j}=1 \tag{2}$$

$$L_{pm_i}^{\mathrm{CPU}} = \sum_{j=1}^{m} L_{vm_j}^{\mathrm{CPU}} x_{i,j} \leqslant R_{pm_i}^{\mathrm{CPU}} \tag{3}$$

$$L_{pm_i}^{\mathrm{RAM}} = \sum_{j=1}^{m} L_{vm_j}^{\mathrm{RAM}} x_{i,j} \leqslant R_{pm_i}^{\mathrm{RAM}} \tag{4}$$

$$\min\ corr(pm_i, vm_j) + \sum_r imbal(r) + imb(pm_i) \tag{5}$$

式(1)约束了 $x_{i,j}$的范围,$x_{i,j}$等于 0 表示 vm_j 不在 pm_i 上,等于 1 表示在 pm_i 上;式(2)约束一个虚拟机只被分配在一个物理机上运行;式(3)~式(4)表明各物理机中的资源负载量小于等于资源量,$L_{vm_j}^{\mathrm{CPU}}$和 $L_{vm_j}^{\mathrm{RAM}}$分别表示 vm_j 中 CPU 和内存资源的负载量,$R_{pm_i}^{\mathrm{CPU}}$和 $R_{pm_i}^{\mathrm{RAM}}$分别表示 pm_i 中 CPU 和内存资源的资源量。式(5)描述了每个虚拟机 vm_j 预放置时应达到的目标,即获得各虚拟机的最佳预放位置,$corr(pm_i, vm_j)$是物理机 pm_i 与虚拟机 vm_j 之间的资源消耗互补性,$imbal(r)$是物理机间的资源 $r(r \in \{CPU, RAM\})$的负载不均衡性,$imb(pm_i)$是 pm_i 上资源负载的不均衡性,相应度量方法见下文。

2.2　互补性度量

为使物理机内各资源消耗相对均衡,在放置虚拟机时,选择与虚拟机资源消耗具有互补性的物理机。本文使用 Pearson 相关系数计算物理机和虚拟机负载的相似性 $\rho_{pm,vm}$(如式(6)所示),并基于该相似性计算出虚拟机和物理机资源消耗的互补性[18-19],如式(7)所示。

$$\rho_{pm_i,vm_j} = \frac{\sum_r (\alpha_{pm_i}^r - \overline{\alpha}_{pm_i})(\overline{\alpha'}_{vm_j}^r - \alpha'_{vm_j})}{\sqrt{\sum_r (\alpha_{pm_i}^r - \overline{\alpha}_{pm_i})^2}\sqrt{\sum_r (\alpha'^r_{vm_j} - \overline{\alpha}'_{vm_j})^2}} \tag{6}$$

$$corr(pm_i, vm_j) = \frac{1 + \rho_{pm_i,vm_j}}{2} \tag{7}$$

其中,$\alpha_{pm_i}^r$是 pm_i 上资源 $r(r \in \{CPU, RAM\})$的利用率,其值等于 $L_{pm_i}^r / R_{pm_i}^r$,由于虚拟机初始放置时,负载量未知,并且虚拟机存在多种异构,此处设 vm_j 资源 r 的相对资源量为 $\alpha'^r_{vm_j} = R_{vm_j}^r / C^r$,$R_{vm_j}^r$为 vm_j 的资源 r 的最大资源量,即租户申请的资源量,$C^r = \max(R_{vm_j}^r)$是所有虚拟机中资源 r 的最大资源量,$\overline{\alpha}_{pm_i}$是 pm_i 上 CPU 和内存 2 种资源的利用率均值,$\overline{\alpha}'_{vm_j}$是 vm_j 上 CPU 和内存 2 种资源的相对资源量均值,$\rho_{pm,vm} \in [-1, 1]$,有 $corr(pm_i, vm_j) \in [0, 1]$。当其值为 0 时,物理机与虚拟机有最强的互补性;当值为 1 时,物理机与虚拟机的互补性最弱。

2.3　不均衡性度量

2.3.1　物理机间的负载不均衡性

在不同物理机上不同资源的负载可能差距很大,且各类资源的量纲不同,因此,本文使用变异系数度量物理机间的负载不均衡性,式(8)所示为物理机间资源 r 的负载不均衡性。

$$imbal(r) = \frac{\sigma^r}{\overline{L}^r} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}\left(\frac{L_{pm_i}^r}{\overline{L}^r} - 1\right)^2} \tag{8}$$

其中,σ^r 是所有物理机的资源 r 负载量标准差,$\overline{L}^r = \frac{1}{n}\sum_{i=1}^{n} L_{pm_i}^r$是所有物理机的资源 r 负载量平均值。

2.3.2　物理机内的负载不均衡性

将负载量转化为利用率,可以消除不同资源的量纲差异。因此,本文使用资源利用率的标准差衡量物理机内各资源的负载不均衡性。pm_i 上资源负载的不均衡性可表示为:

$$imb(pm_i) = \sqrt{\frac{1}{2}\sum_r (\alpha_{pm_i}^r - \overline{\alpha}_{pm_i})^2} \tag{9}$$

2.4　预放置过程

根据待放置的虚拟机集合 $VM = \{vm_1, vm_1, \cdots, vm_m\}$的内存资源总量计算出物理机数量的理论最大值 $n = \lceil \sum_{j=1}^{m} L_{vm_j}^{\mathrm{RAM}} / \overline{R}^{\mathrm{RAM}} \rceil$($\overline{R}^{\mathrm{RAM}}$表示物理机的平均资源量),用于限定物理机开启的最大数量。根据式(5)求出每个 $vm_j \in VM$ 的最佳预放位置 pm_k,并进行预放置。

3　基于机器学习的虚拟机迁移调整

3.1　CPU 负载等级预测

预放置后,随着时间的推移虚拟机的负载会周期性动态波动。本文使用深度神经网络来学习物理机负载的周期性波动,并给出对负载等级的预测,最后根据预测值进行虚拟机迁移调整。本文对 CPU 的负载等级进行预测。

式(10)给出某物理机 pm_i 的 CPU 资源利用率 $\alpha_{pm_i}^{\mathrm{CPU}}$与负载等级 $level$ 之间的关系。当 $0 \leqslant \alpha_{pm_i}^{\mathrm{CPU}} < 1$ 时,负载等级为物理机 CPU 资源利用率乘以 10 后向上取整,即 1~10 共 10 个评级,1 级包含 $\alpha_{pm_i}^{\mathrm{CPU}} \in (0, 0.1]$以及处于关闭状态的物理机;当 $\alpha_{pm_i}^{\mathrm{CPU}} = 1$ 时,负载等级为 10。

$$level(\alpha_{pm_i}^{\mathrm{CPU}}) = \begin{cases} \lceil 10 \times \alpha_{pm_i}^{\mathrm{CPU}} \rceil, 0 \leqslant \alpha_{pm_i}^{\mathrm{CPU}} < 1 \\ 10, \alpha_{pm_i}^{\mathrm{CPU}} = 1 \end{cases} \tag{10}$$

某物理机 pm_i 的 CPU 负载等级(记为 $Prediction(pm_i)$

返回值为元组[*level*,*probability*]),其预测过程描述如下:

1)收集 pm_i 过去一段时间 T(如 $T=1$ h)内的 CPU 负载数据。

2)使用深度神经网络对收集的 CPU 负载数据进行特征提取并作出预测,并用式(10)获得预测等级 *level* 和该等级的概率 *probability*。

3.2 基于 DQN 的迁移整合

传统的虚拟机迁移决策通常基于当前宿主物理机的负载,很少考虑数据中心的整体负载,因此,可能会产生不必要的重复迁移。本文基于 DQN 的迁移整合,提取数据中心负载数据的特征信息,并调整物理机数量。数据中心的负载具有不确定性,马尔科夫决策过程可以提供一种对规划和行动进行推理的形式[20],因此,本文用部分可观测的马尔科夫决策过程来描述虚拟机迁移整合问题。

3.2.1 马尔科夫决策过程

定义五元组 MDP:$(S,A,\{P_{sa}(\cdot)\},\gamma,R)$,各符号表示如下:

1)S 是状态空间。每个状态 $s(\in S)$ 由数据中心上资源的历史利用率和物理机数量构成,被定义为:[***DCLRH***,$|PM|$]。

$$\boldsymbol{DCLRH}=(\alpha_{t_1}^{\mathrm{CPU}},\alpha_{t_2}^{\mathrm{CPU}},\cdots,\alpha_{t_x}^{\mathrm{CPU}}) \tag{11}$$

其中,***DCLRH***是最近 T 时间内数据中心 x 次采样结果,$\alpha_{t_1}^{\mathrm{CPU}}$ 是 t_1 时刻数据中心的 CPU 资源利用率(即 $\sum_{i=1}^{n}L_{pm_i,t_1}^{\mathrm{CPU}}/\sum_{i=1}^{n}R_{pm_i}^{\mathrm{CPU}}$),$L_{pm_i,t_1}^{\mathrm{CPU}}$ 是 t_1 时刻 pm_i 的 CPU 负载量,$|PM|$为当前正在运行的物理机数量。

2)A 是动作空间。每个状态下都要选择一个动作,所有的动作构成动作空间 A,且 $A=\{-2,-1,0,1,2\}$。当动作为 -2 或 -1 时,表示选择 2 个或 1 个低负载物理机,迁移出其中的所有虚拟机并关闭物理机;当动作为 0 时,表示尽量维持物理机数量不变,继续执行过载物理机的选择和虚拟机迁移,但当待迁虚拟机没有合适的目的物理机时,也会开启新的物理机;当动作为 1 或 2 时,表示先开启 1 个或 2 个物理机,并选择过载物理机上的虚拟机迁移到新开启的物理机中,同样地,当待迁虚拟机没有合适的目的物理机时,还会开启新的物理机。

3)$P_{sa}(\cdot)$ 是状态转换的概率分布。若在某状态 $s\in S$ 采取了动作 $a\in A$,则转换到一个新状态 s',$P_{sa}(\cdot)$给出了转换到下一个状态的概率分布。

4)γ 是折扣因子,其取值范围为[0,1]之间的实数。

5)R 是回馈函数,从环境中获得的信息,用于反映在某状态 $s\in S$ 采取动作 $a\in A$ 后获得的反馈,被定义为 SLA 违背率回馈与能耗损失回馈之和,如式(12)所示。

$$R(s,a)=R_s^a(SLA)+R_s^a(Energy) \tag{12}$$

其中,$R_s^a(SLA)$是 SLA 违背率回馈函数,用于计算在状态 s 采取动作 a 后,SLA 产生的惩罚回馈,如式(13)所示。

$$R_s^a(SLA)=SLA_{s'}/SLA_s \tag{13}$$

$$SLA_s=\sum_{j=1}^{m}(U_s^{rj}-U_s^{aj}) \tag{14}$$

$$R_s^a(Energy)=\sum_{i=1}^{n}(Energy_{s'}^i/Energy_s^i) \tag{15}$$

$$Energy_s^i=\left(Power_s^i+\frac{Power_{s+\Delta t}^i-Power_s^i}{2}\right)\times\Delta t \tag{16}$$

其中,SLA_s 是在状态 s 时所有虚拟机违背率之和,U_s^{rj} 是在状态 s 时 vm_j 需要的资源,U_s^{aj} 为在状态 s 时 vm_j 被分配的资源,$R_s(Energy)$是能耗损失回馈函数,用运行的物理机在下一状态 s'的能耗 $Energy_{s'}^i$ 与当前状态 s 的能耗 $Energy_s^i$ 之比的和表示,$Power_s^i$ 是在状态 s 时 pm_i 的功率。

3.2.2 迁移整合算法

由上述马尔科夫决策过程得到 Q 学习中的最优动作价值函数 $Q^*(s,a)$,如式(17)所示。式(18)定义了动作价值函数 $Q(s,a)$的更新过程,通过迭代获取最优动作价值函数 $Q^*(s,a)$。

$$Q^*(s,a)=R(s,a)+\gamma\sum_{s'\in S}P_{sa}(s')\max_{a'\in A}Q^*(s',a') \tag{17}$$

$$Q(s,a)\leftarrow Q(s,a)+\lambda(R(s,a)+\gamma\max_{a'\in A}Q(s',a')-Q(s,a)) \tag{18}$$

其中,s',a'为下一状态和动作,$Q^*(s,a)$表示在状态 s 采取动作 a 后最终所能获得的期望最大回馈,λ 为学习率。

由于状态随时间动态变化,且为连续值,因此使用深度神经网络对 Q 学习中的动作价值函数 $Q(s,a)$进行拟合,形成 DQN。通过迭代最小化损失的平方来计算损失函数 $L(w)$[21](如式(19)所示),并以此为基础,使用梯度下降方法更新网络中的权值 w。

$$L(w)=E[(R(s,a,w)+\gamma\max_{a'\in A}Q(s',a',w)-Q(s,a,w))^2] \tag{19}$$

其中,$\max_{a'\in A}Q(s',a',w)$为在迭代过程中下一状态获

取的最大回馈,$Q(s,a,w)$为当前状态能获取的最大回馈,w 为神经网络中的权值。

算法 1 描述了基于 DQN 的虚拟机迁移整合方法,该算法在每一个调度时间 t 被执行。s_t 和 a_t 分别表示调度 t 时刻的状态和采取的动作,s'_t和 a'_t分别为采用动作 a_t 后到达的下一个状态和为了在 s'_t获取到最大回馈采取的动作。其中,第 6 行*done* 表示运行环境是否结束,第 7 行是将状态转变元组$(s_t,s'_t,a_t,R(s_t,a_t),done)$存入样本集 *EXP* 中,再从中随机选择 *batchsize* 个元组组成集合 *minibatch*,用于训练神经网络[22],第 9 行 ~ 第 15 行是对 *minibatch* 中的每个元组生成 Y_j,即神经网络的输出值。当 *done* 值为真时,即仿真结束,Y_j 为当前获得的 $R(s_t,a_t,w)$;否则 Y_j 为动作价值函数在 s'_t采取动作 a'_t的最大值。第 16 行是使用梯度下降更新神经网络。

算法 1　基于 DQN 的迁移整合

```
procedure Consolidation(PM)
EXP = ϕ;
Initialize the weight w;
Get state s_t = [DCLRH, |PM|];
Select a_t with probability ϵ or a_t = max_a Q*(s_t, a, w);
Perform a_t and get R(s_t, a_t), s'_t, done;
Store the tuple(s_t, s'_t, a_t, R(s_t, a_t), done) in EXP;
minibatch = {batchsize tuples selected from EXP};
for each tuple in minibatch do
if done = = true then
Y_j = R(s_t, a_t, w);
else
Y_j = R(s_t, a_t, w) + γ max_{a'_t} Q(s'_t, a'_t, w);
end if
end for
Perform gradient descent with L(w) = (Y_j − Q(s,a,w))^2;
end procedure
```

算法 2 所示为低载物理机的整合。首先使用第 3.1 节的 *Prediction* 过程获取各物理机 CPU 负载等级的预测值 *level* 和该等级的概率 *probability*(第 6 行),在所有物理机中选择出预测等级最低且属于该等级概率最大的物理机 pm_{low}(第 7 行 ~ 第 11 行),对 pm_{low}进行可迁移性测试,即测试 pm_{low}上所有虚拟机是否都可以找到目的物理机(第 13 行 ~ 第 17 行),在第 14 行中,为 vm_j 选择目的物理机(设为 pm_{des_j}),如果未选择出,那么 *close* 被置为 1,表明无合适的目的物理机接收该虚拟机,即该虚拟机不可以被迁移;如果 pm_{low}上的所有虚拟机可以被迁移,那么 *close* 为 0,则完成所有迁移后将 pm_{low} 关闭(算法 2 第 18 行 ~ 第 21 行);如果对所选 pm_{low}上的虚拟机可迁移性测试失败,那么将该 pm_{low}从集合 *PM* 中移除,重新整合。

算法 2　低载物理机的整合

```
procedure Consol_low(PM)
level_min = 10;
pr[11] = {0,0,…,0};
close = 0;
for pm_i in PM do
[level, probability] = Prediction(pm_i);
if level ≤ level_min and probability > pr[level] then
level_min = level;
pr[level] = probability;
pm_low = pm_i;
end if
end for
for each vm_j on pm_low do
if !(DestiPM(PM', vm_j, 2, 6)) then //调用算法 3
close = 1; break;
end if
end for
if !(close) then
Migrate all virtual machines on pm_low;
Shut down pm_low;
end if
end procedure
```

算法 3 描述了目的物理机的选择方法。在为低载物理机中的待迁虚拟机选择目的物理机时,将目的物理机在未来 T 时间内的预测等级限定在 2 级 ~ 6 级(算法 2 的第 14 行),排除等级为 1 级的物理机,在一定程度上避免重复迁移,也使低负载物理机中的虚拟机不会迁移至新开启的物理机。限定最高等级为 6 级,为过载物理机中的待迁出虚拟机的目的物理机选择预留缓冲资源(7 级 ~ 8 级物理机)。从符合负载等级要求的物理机中,选出预测等级较低且属于该等级概率最大的物理机作为最终目的物理机。算法 3 第 7 行中 $Contain(pm_i,vm)$函数用于判断 vm 能否放置在 pm_i 上,如果 vm 放置到 pm_i 后 $L^r_{pm_i} \geqslant R^r_{pm_i}$,则不能放置;否则可放置,即迁移可行。如果算法 3 的返回值为空,则表明没有符合条件的目的物理机;否则 pm_{des}即为 vm 的目的物理机。

算法 3　目的物理机的选择

```
procedure DestiPM(PM', vm, l, h) //PM'是 PM 的子
//集,即不含过载物理机和正在执行迁移的物理机,l、h 规定
//了目的物理机的负载预测等级范围
```

pm_{des} = NULL;
$level_{min}$ = 10;
pr[11] = {0,0,…,0};
for pm_i in PM′ do
[level,probability] = Prediction(pm_i);
if level≥1 and level≤h and Contain(pm_i,vm) then
if level≤$level_{min}$ and probability > pr[level] then
$level_{min}$ = level;
pr[level] = probability;
pm_{des} = pm_i;
end if
end if
end for
return pm_{des}; //返回为 vm 选出的目的物理机
end procedure

算法 4 描述了过载物理机的负载转移。当某物理机的负载预测等级为 10 级,即被定义为过载物理机,记为 pm_{overld},它表明该物理机在未来 T 时间内可能会发生过载情况,即未来的 $\alpha^{CPU}_{pm_{overld}}>0.9$。为在一定程度上降低 SLA 违背率,需要从 pm_{overld} 中选择出一定量的虚拟机迁移到其他物理机上。每次选择资源利用率最低的虚拟机迁出,直到 pm_{overld} 的负载预测等级低于 10。其中,$getUtiAvg(vm)$ 用于获取 vm 过去 T 时间内资源利用率的平均值。在为过载物理机中的待迁虚拟机选择目的物理机时,将目的物理机的预测等级限定在 2 级 ~8 级(第 11 行),尽量不开启新的物理机和避免可能的重复迁移。同样地,从符合负载等级要求的物理机中选出预测等级较低且属于该等级概率最大的物理机作为最终的目的物理机。如果没有符合条件的目的物理机,则会开启一个新物理机接受待迁虚拟机,以避免违背 SLA(算法 4 第 12 行 ~第 14 行)。

算法 4 过载物理机的负载转移

procedure Loadtr_overld(pm_{overld})
while Prediction(pm_{overld}).level = =10
min = 1;
for each vm on pm_{overld} do
utiAvg = getUtiAvg(vm);
if min > utiAvg then
min = utiAvg;
vm_{min} = vm;//获得资源利用率最低的 vm_{min}
end if
end for
$pm_{des_{min}}$ = DestiPM(PM′,vm_{min},2,8); //调用算法 3
if $pm_{des_{min}}$ = = NULL then
Trun on a new pm as $pm_{des_{min}}$;
end if
vm_{min} is migrated to $pm_{des_{min}}$;
end while
end procedure

4 实验结果与分析

4.1 实验设置

IaaS 实验环境基于 Cloudsim 3.0.3 仿真平台和 PlaneLab 实验室在 2011 年 3 月—2011 年 4 月随机获取的10 天中所有虚拟机的负载数据。表 1 所示为虚拟机和物理机配置。虚拟机配置使用 Amazon EC2 提供的 4 种配置,如编号 1 ~ 编号 4,虚拟机创建时随机选择其中一种配置。物理机使用 HP ProLiantML110 G4 和 HP ProLiantML110 G5 的 2 种服务器配置,如编号 5 ~ 编号 6,实验中均匀创建了 800 台上述 2 种配置的物理机。

表 1 虚拟机和物理机配置

编号	CPU/MIPS	内存/MB	核心数	带宽/(Mb·s^{-1})
1	2 500	870	1	100
2	2 000	1 740	1	100
3	1 000	1 740	1	100
4	500	613	1	100
5	1 860	4 096	2	1 000
6	2 660	4 096	2	1 000

在基于深度神经网络的 CPU 负载等级预测过程中,相关参数设置如下:输入层节点数为 12,用于输入过去时间段 T(T = 1 h)内物理机的 CPU 负载数据,隐层数为 3,各隐层神经元数量分别为 100、100、20,输出层节点数为 10,分别对应 10 种负载等级,选取 ReLU 作为神经元激活函数,选取 Softmax 函数作为分类器获取各评级的概率,更新过程使用 AdaGrad 方法[23]。

在基于 DQN 的迁移整合中,相关参数设置如下:输入层节点数为 13,隐层数为 2,各隐层神经元数量为 20,输出层节点数为 5,选取 ReLU 作为神经元激活函数,选取 Softmax 函数作为分类器,更新过程使用梯度下降方法,ϵ 的初值为 0.9,且逐渐趋于 0,γ = 0.9,λ = 1。本文负载采样周期 τ($\tau = T/12$)为 300 s,调度周期为 300 s。

4.2 性能指标

为评价虚拟机放置策略对数据中心的性能影响,对 SLA 违背率、负载均衡性、能耗等进行评价。本文使用文献[17]提出的 *SLAV* 参数作为 SLA 违背率的评价标准、*ESV* 参数作为 SLA 违背率和能耗的综合性能评价标准,负载均衡性使用式(8)和式(9)来进行评价,能耗性能 E 使用式(20)计算得到。

$$E=\sum_{S}\sum_{i-1}^{n}Energy_s^i \tag{20}$$

4.3　结果分析

4.3.1　负载不均衡性

本节将预放置方法与首次适应递减(简称 FFD)方法、下次首次适应递减(简称 NFD)方法、最佳适应递减(简称 BFD)方法在负载不均衡性上进行比较。

图 2 给出不同方法物理机间负载不均衡性的对比结果。本文方法在虚拟机数量大于 40 后,不均衡性相比另外 3 种方法有更小的数值,即有更好的物理机间负载均衡性。当虚拟机数量较少时,由于潜在互补机型数量较少,因此本文方法的负载均衡性相对弱一些。随着虚拟机数量的增加,本文方法能够搜寻并配对潜在的互补机型,降低了物理机间负载不均衡性。在虚拟机数量从 20 个增加到 240 个的阶段,本文的预放置方法逐渐将不均衡性数值稳定在 0.8(±0.07),表现出较好的负载均衡性能。

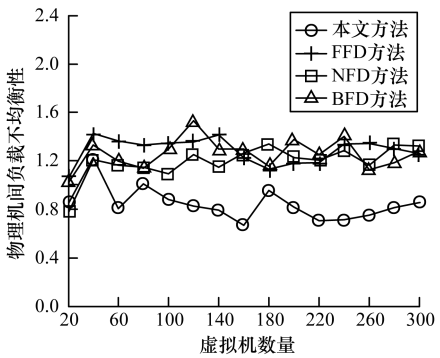

图 2　不同方法物理机间的负载不均衡性对比

图 3 给出不同方法物理机内负载不均衡性的对比结果。可以看出,4 种方法下的物理机内的负载不均衡性随虚拟机数量呈线性增长,对 4 种方法做一元线性回归,即 $y = kx + b$ 的形式,得到本文方法 k = 0.140 4,而其余 3 种方法的 k 值取值范围为[0.154 7, 0.155 3]。说明本文方法的物理机内负载均衡性要略优于其余 3 种方法。

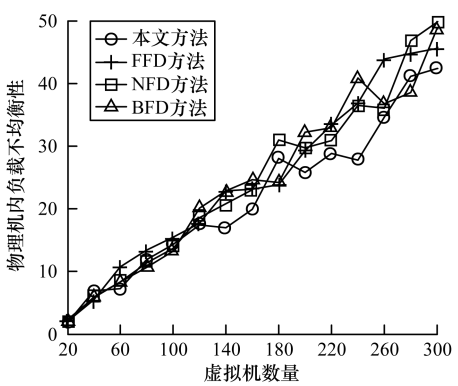

图 3　不同方法物理机内的负载不均衡性对比

4.3.2　整体性能分析

本节将与文献[17]中较为优秀的 6 种方法:MAD-MMT-2.5,LR-MMT-1.2,IQR-MMT-1.5,MAD-RS-2.5,LR-RS-1.2,IQR-RS-1.5 以及 GMM-VMC 方法进行对比,所有对比方法中的相关参数值均按文献[17]设定,并保证实验环境与本文方法统一。表 2描述了 8 种方法的 *E*、*SLAV* 以及 *ESV* 等指标的对比结果。从表 2 可以看出,本文方法的 *SLAV* 和 *ESV* 值最低,表明 SLA 违背率低,且有最好的 SLA 和能耗的综合性能 *ESV*。但能耗与 SLA 存在矛盾,相比于 GMM-VMC 方法,本文方法损失了 18% 的能耗,且分别降低 83% 和 77%,迁移次数降低 73%。相比于文献[17]中的具有最佳 *ESV* 性能的 MAD-MMT 方法,本文方法的 *ESV* 性能更好,降低了 93%。

表 2　不同方法的各指标对比结果

方法	*E*/kWh	$SLAV/10^{-5}$	$ESV/10^{-3}$	迁移次数/10^4
本文方法	138.50	0.331	0.426	0.4
GMM-VMC 方法	117.23	1.955	1.860	1.5
MAD-MMT-2.5 方法	183.50	3.348	6.143	2.6
LR-MMT-1.2 方法	161.87	4.974	8.051	2.8
IQR-MMT-1.5 方法	187.53	3.288	6.166	2.7
MAD-RS-2.5 方法	181.17	7.230	13.310	2.5
LR-RS-1.2 方法	154.37	7.060	10.900	2.3
IQR-RS-1.5 方法	185.76	6.700	12.446	2.4

图 4 描述了本文方法与 MAD-MMT-2.5 方法在物理机间的负载不均衡性(由于 GMM-VMC 方法未对负载均衡优化,选择与综合性能较好的 MAD-MMT-2.5方法进行负载均衡性能对比)。可以看出,本文方法具有更低的不均衡性。结果表明,本文基于 DQN 的物理机数量调整和深度神经网络的预测方法能够获得较稳定的负载均衡性。

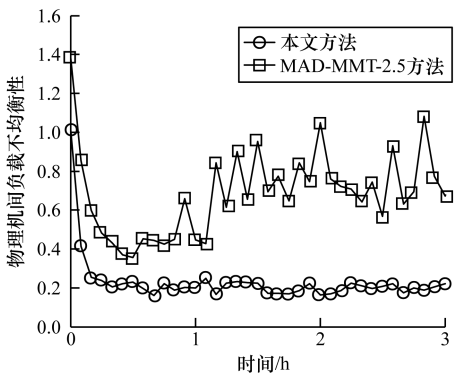

图 4　物理机间负载不均衡性

5　结束语

本文提出基于机器学习的虚拟机放置方法。该方法利用互补性和不均衡性对虚拟机进行预放置,采用深度神经网络方法对 CPU 负载进行预测评级,并基于预测结果使用强化学习调整物理机数量。实

验结果表明,本文方法相对传统启发式算法具有较好的负载均衡性和稳定性。下一步将通过加入网络通信、磁盘 I/O 等,优化预测模型、迁移策略,以减少虚拟机迁移次数和提高数据中心综合性能。

参考文献

[1] ZHANG Xiaojing, LINDBERG T, XIONG N, et al. Cooling energy consumption investigation of data center it room with vertical placed server[J]. Energy Procedia, 2017,105:2047-2052.

[2] KUSIC D,KEPHART J O,HANSON J E,et al. Power and performance management of virtualized computing environments via lookahead control[J]. Cluster Computing, 2009,12(1):1-15.

[3] ZHENG Qinghua, LI Rui, LI Xiuqi, et al. A multi-objective biogeography-based optimization for virtual machine placement[C]//Proceedings of the 15th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing. Washington D. C., USA: IEEE Press, 2015:687-696.

[4] JIANG Dailin, HUANG Peijie, LIN Piyuan, et al. Energy efficient VM placement heuristic algorithms comparison for cloud with multidimensional resources[C]//Proceedings of International Conference on Information Computing and Applications. Berlin, Germany: Springer, 2012:413-420.

[5] LI Rui, ZHENG Qinghua, LI Xiuqi, et al. A novel multi-objective optimization scheme for rebalancing virtual machine placement[C]//Proceedings of the 9th IEEE International Conference on Cloud Computing. Washington D. C., USA: IEEE Press, 2016:710-717.

[6] AHVAR E, AHVAR S, MANN Z A, et al. CACEV: a cost and carbon emission-efficient virtual machine placement method for green distributed clouds[C]//Proceedings of IEEE International Conference on Services Computing. Washington D. C., USA: IEEE Press, 2016:275-282.

[7] FELLER E, RILLING L, MORIN C. Energy-aware ant colony based workload placement in clouds[C]//Proceedings of the 12th International Conference on Grid Computing. Washington D. C., USA: IEEE Computer Society, 2011:26-33.

[8] TSAIM H, CHOU J, CHEN J. Prevent VM migration in virtualized clusters via deadline driven placement policy[C]//Proceedings of IEEE International Conference on Cloud Computing Technology and Science. Washington D. C., USA: IEEE Computer Society, 2013:599-606.

[9] 徐思尧,林伟伟,王子骏. 基于负载高峰特征的虚拟机放置算法[J]. 软件学报,2016,27(7):1876-1887.

[10] BOBROFF N, KOCHUT A, BEATY K. Dynamic placement of virtual machines for managing SLA violations[C]//Proceedings of the 10th IFIP/IEEE International Symposium on Integrated Network Management. Washington D. C., USA: IEEE Press, 2007:119-128.

[11] BUYYA R, GARG S K, CALHEIROS R N. SLA-oriented resource provisioning for cloud computing: challenges, architecture, and solutions[C]//Proceedings of International Conference on Cloud and Service Computing. Washington D. C., USA: IEEE Computer Society, 2011:1-10.

[12] VERMA M, GANGADHARAN G R, NARENDRA N C, et al. Dynamic resource demand prediction and allocation in multi-tenant service clouds[J]. Concurrency and Computation: Practice and Experience, 2016, 28(17): 4429-4442.

[13] HUANG Wei, LI Xin, QIAN Zhuzhong. An energy efficient virtual machine placement algorithm with balanced resource utilization[C]//Proceedings of the 7th International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing. Washington D. C., USA: IEEE Press, 2013:313-319.

[14] FARAHNAKIAN F, LILJEBERG P, PLOSILA J. Energy-efficient virtual machines consolidation in cloud data centers using reinforcement learning[C]//Proceedings of the 22nd Euromicro International Conference on Parallel, Distributed, and Network-Based Processing. Washington D. C., USA: IEEE Computer Society, 2014:500-507.

[15] GHOBAEI-ARANI M, RAHMANIAN A A, SHAMSI M, et al. A learning-based approach for virtual machine placement in cloud data centers[EB/OL]. [2018-07-02]. https://onlinelibrary.wiley.com/doi/full/10.1002/dac.3537.

[16] RANJBARI M, AKBARI TORKESTANI J. A learning automata-based algorithm for energy and SLA efficient consolidation of virtual machines in cloud data centers[J]. Journal of Parallel and Distributed Computing, 2017,113:55-62.

[17] BELOGLAZOV A, BUYYA R. Optimal online deterministic algorithms and adaptive heuristics for energy and performance efficient dynamic consolidation of virtual machines in cloud data centers[J]. Concurrency and Computation: Practice and Experience, 2012, 24(13): 1397-1420.

[18] ZHU Qian, ZHU Jiedan, AGRAWAL G. Power-aware consolidation of scientific workflows in virtualized environments[C]//Proceedings of ACM/IEEE International Conference for High Performance Computing, Networking, Storage and Analysis. Washington D. C., USA: IEEE Computer Society, 2010:1-12.

[19] MENG Xiaoqiao, ISCI C, KEPHART J, et al. Efficient resource provisioning in compute clouds via VM multiplexing[C]//Proceeding of the 7th International Conference on Autonomic Computing. New York, USA: ACM Press, 2010:11-20.

[20] NG A Y. Shaping and policy search in reinforcement learning[D]. Berkeley, USA: University of California, 2003.

[21] MNIH V, KAVUKCUOGLU K, SILVER D, et al. Playing atari with deep reinforcement learning[EB/OL]. [2018-07-02]. https://arxiv.org/pdf/1312.5602v1.pdf.

[22] SCHAUL T, QUAN J, ANTONOGLOU I, et al. Prioritized experience replay[EB/OL]. [2018-07-02]. https://arxiv.org/pdf/1511.05952.pdf.

[23] DUCHI J, HAZAN E, SINGER Y. Adaptive subgradient methods for online learning and stochastic optimization[J]. The Journal of Machine Learning Research, 2011,12:2121-2159.

编辑 赵 辉