数据中心虚拟机节能管理机制

朱德剑 白光伟 蔡炎伟 任 栋 沈 航
(南京工业大学计算机科学与技术系 南京 210009)

摘 要 大规模数据中心需要消耗大量的电能,由此带来了高额的运营成本以及环境污染等问题。为了降低数据中心的能耗,在构造了数据中心管理模型的基础上,提出了虚拟机静态安置算法与动态调整算法。虚拟机的动态迁移技术能够有效地降低数据中心能耗,提升资源利用率。然而,过度地迁移虚拟机,会影响应用的运行质量,造成SLA违背。动态调整阶段,采用了动态阈值的方法来控制虚拟机的迁移,降低能耗。最后,利用CloudSim平台进行了大量的模拟实验。实验结果表明,所提出的数据中心虚拟机节能管理机制(EAMVM)能够降低能源消耗,减少虚拟机的迁移次数。

关键词 能耗,虚拟机,动态阈值,动态迁移
中图法分类号 TP393 **文献标识码** A **DOI** 10.11896/j.issn.1002-137X.2017.10.004

Energy-aware Management of Virtual Machines in Data Center

ZHU De-jian BAI Guang-wei CAI Yan-wei REN Dong SHEN Hang
(Department of Computer Science and Technology, Nanjing Tech University, Nanjing 210009, China)

Abstract Large scale data centers need to consume a large amount of power, resulting in high operating costs and other issues such as environmental pollution. In order to reduce the energy consumption of the data center, we constructed a management model of the data center and proposed the algorithm of the static placement algorithm and dynamic adjustment of the virtual machine. Dynamic migration of virtual machine can effectively reduce the energy consumption while improving resource utilization. However, excessive migration of virtual machines will affect the quality of the application and cause SLA violation. In the dynamic adjustment stage, we adopted dynamic thresholds to control the virtual machine migration and reduce energy consumption. Finally, we used CloudSim to do a lot of experiments. The results show that the energy-aware management of virtual machine (EAMVM) mechanism can reduce energy consumption and reduce the number of virtual machine migration.

Keywords Energy consumption, Virtual machines, Dynamic thresholds, Live migrations

1 引言

云计算和大数据等应用的流行,推动了数据中心的快速发展,使数据中心的数量不断增加。2011年全球有超过500000个数据中心[1],IDC预测到2017年这一数量将达到860万。大量的数据中心带来了巨大的电能消耗,2010年全球数据中心消耗的电能占世界发电功率总和的1.5%,到2020年这一比例或将达到8%。数据中心面临着高能耗问题的挑战[2]。

研究表明,全球大多数据中心的资源利用率在15%~20%之间。数据中心资源利用率不高造成了资源的浪费,大量的物理机工作在闲置状态,增加了数据中心的能耗以及运营者的负担。得益于虚拟化技术的发展,多台虚拟机(Virtual Machine,VM)可以工作在同一台物理机(Physical Machine,PM)上,提高了物理机资源的使用率,从而降低了数据中心的能耗。虚拟机在线迁移技术的发展使得物理机资源的使用更加灵活,在线迁移技术可以使虚拟机在不中断内部应用程序的状态下从一台物理机迁移到另一台物理机。管理者可以利用虚拟机的迁移,动态地调整虚拟机的放置,使得数据中心中的所有虚拟机运行在尽可能少的物理机上,关闭低负载的物理机,降低数据中心能耗。但是,过高的物理机资源使用率会带来其他问题,虚拟机中负载的动态性导致虚拟机所需要的

到稿日期:2016-09-11 返修日期:2016-12-31 本文受国家自然科学基金项目(61502230,61073197),江苏省自然科学基金项目(BK20150960),江苏省普通高校自然科学研究项目(15KJB520015),江苏省六大高峰人才基金资助项目(第八批),2015年度普通高校研究生科研创新计划(KYLX15_0804)资助。

朱德剑(1991-),男,硕士生,主要研究方向为移动云计算、云计算节能技术,E-mail:18752016068@163.com;**白光伟**(1961-),男,博士,教授,博士生导师,CCF高级会员,主要研究方向为无线传感器网络、移动互联网、网络体系结构和协议、网络系统性能分析和评价、多媒体网络服务质量等;**蔡炎伟**(1991-),男,硕士生,主要研究方向为云计算安全;**任 栋**(1987-),男,硕士生,主要研究方向为云游戏、GPU虚拟化;**沈 航**(1984-),男,博士,讲师,主要研究方向为云计算、移动互联网、无线多媒体通信协议等。

资源动态变化，负载过高的物理机中的虚拟机不能适应负载的变化，虚拟机的需求得不到满足，造成 SLA 违背；而且过高的负载使得物理机温度过高，超过安全的温度值后可能造成物理机硬件的损坏，从而造成巨大的损失。如何管理数据中心虚拟机，成为了研究的热点。

数据中心虚拟机管理主要分为初始化放置与动态放置。初始化放置主要根据虚拟机的初始需求来分配合适的物理机，在数据中心初始化时进行，该行为具有长效性；动态放置是在数据中心的实际运行中进行，利用数据中心运行状态信息来调整虚拟机放置，保证数据中心的正常运行。负载的动态变化导致虚拟机所请求的资源也在不断变化，当虚拟机请求得不到满足时，就需要将该虚拟机迁移到有足够资源的物理机上，以满足其资源需求。当物理机资源利用率较低时，就需要迁移合并物理机上的虚拟机，并关闭低负载的物理机来降低数据中心能耗。

本文针对数据中心的异构性，在虚拟机初始放置阶段利用改进的最先适应算法完成虚拟机的初始放置，减少数据中心运行过程中为了达到节能的目的所造成的虚拟机的迁移量。通过对阈值的合理设置和对迁移时机的判断，来降低数据中心 SLA 违背率，降低能耗，减少虚拟机的迁移次数。

本文第 2 节分析与总结了相关工作；第 3 节介绍了数据中心虚拟机管理的系统模型；第 4 节给出了相应的管理算法；第 5 节通过 CloudSim 进行仿真实验；最后总结全文。

2 相关工作

国内外很多研究者研究了数据中心的虚拟机调度，以达到降低能耗、平衡负载、提升资源利用率等目标。

文献[3]对移动媒体云任务的处理进行了优化，针对移动用户对视频流的不同需求，把流媒体任务分解成若干个转码与视频流传输任务。执行转码任务的虚拟机有较大计算需求，而视频流传输任务需要较大的带宽，这两种任务使用对应种类的虚拟机进行处理，通过合理安排每台物理机上两种虚拟机的数目，来提升物理机的资源利用率并最小化能量消耗。文献[4]定义了衡量物理机多种资源利用均衡程度的标准，并利用这一标准来进行虚拟机的分配，通过平衡物理机不同资源的使用率来降低资源碎片率，从而增大物理机资源的综合利用率，降低数据中心能耗。文献[5]利用改进的遗传算法，从大量的虚拟机分配方案中找出最优的一种，达到增大资源使用率、降低能耗的目的。然而，文献[3-5]注重于通过提升资源利用率来减少物理机的使用数目，从而降低能耗，但是数据中心存在异构性，不同物理机在能耗与资源拥有量方面存在差异。

在虚拟机动态整合方面，为了能够在 SLA 违背和能耗等方面取得较好的效果，许多研究中采用了设置阈值的方法，通过设置阈值，使物理机资源利用率在一定的安全范围内。文献[6]为了保证 QoS 的要求，采用了单阈值的方法，设置了资源使用的上限。但是仅仅考虑上限，使得许多物理机工作在低负载状态，浪费了资源。文献[7]采用静态阈值的方法来控制虚拟机的动态迁移，阈值的上限为 75%，下限为 25%，当物理机资源使用率在阈值之外时触发迁移。静态的阈值不能适应数据中心负载的动态变化。文献[8]基于虚拟机资源使用的历史数据，假设物理主机资源的使用近似服从学生式分布，利用历史数据得到的均值与方差，通过逆概率函数来动态地设定阈值，一旦物理机资源的使用率在阈值范围之外，就迁移相应的虚拟机；但是在虚拟迁移时机的触发方面其不能适应负载的短暂波动，造成了虚拟机不必要的迁移，带来了过多的迁移开销。

文献[9-10]利用基于时间序列的预测技术 $AR(n)$ 来预测虚拟机中负载未来的需求值。该技术通过 n 个当前时间点之前的虚拟机资源的需求值来计算未来的需求值，然后通过预测的需求值调整虚拟机资源的分配情况。在满足虚拟机需求的情况下动态迁移虚拟机，以减少物理机的使用量。每个时间间隔内最小化物理机的使用量，提高了物理机资源的利用率。但是，该技术未能考虑虚拟机迁移所带来的消耗，过度的合并带来了大量不必要的虚拟机迁移，造成了许多额外的开销。

针对以上存在的问题，本文设计了数据中心虚拟机调度机制，通过虚拟机的静态安置以及运行过程中的动态调整，合理地利用数据中心的资源。

3 系统模型

数据中心的基本模型如图 1 所示，数据中心虚拟化以虚拟机的形式体现，通过虚拟化技术使得多台虚拟机运行在一台物理机上。虚拟机通过其所需的资源来描述，虚拟机 VM_i 所需要的资源用向量 $R_{vi}=(R_{vi,1}, R_{vi,2}, \cdots, R_{vi,k}, \cdots, R_{vi,d})$ 来表示，$R_{vi,k}$ 表示虚拟机 VM_i 所需的第 k 维资源的量（如 CPU、内存、网络带宽等）。作为承载虚拟机的物理机 PM_j，其所能提供的资源可以用向量 $R_{pj}=(R_{pj,1}, R_{pj,2}, \cdots, R_{pj,k}, \cdots, R_{pj,d})$ 来表示，$R_{pj,k}$ 表示物理机 PM_j 在第 k 维资源上所拥有的资源最大量。当物理机上存在虚拟机时，该物理机处于运行状态，即使物理机处于空载状态，仍会产生较多能耗。适当关闭没有负载的物理机将有助于降低数据中心能耗。

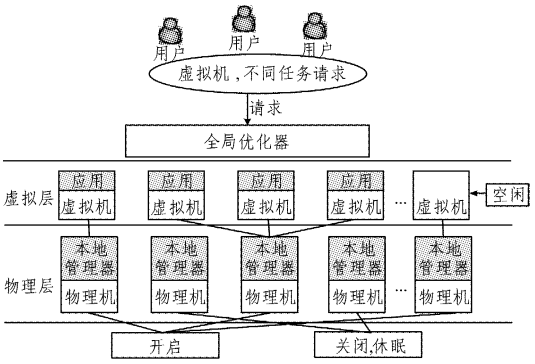

图 1 数据中心模型

3.1 模块功能

本文的数据中心虚拟机管理模块主要分为全局优化器与本地管理器，这 2 个管理模块分别执行不同的功能。

全局优化器：此模块负责监控整个数据中心的运行状况，处理用户的请求并管理虚拟机的动态迁移。在数据中心的运行过程中，此模块利用本地管理器反馈的信息，以一定的时间

间隔 T 检查当前运行的物理机的资源利用率是否在所设定的阈值范围内，若超过了阈值的上限则视为过载，此时需要迁移一定的虚拟机；若低于阈值的下限则视为低负载，此时应通过虚拟机迁移提高资源利用率。

本地管理器：此模块运行在每台运行的物理机中，主要用来监控物理机资源的使用情况，管理物理机中运行的虚拟机；及时地响应虚拟机的资源请求，当虚拟机资源得不到满足时，及时发送信息给全局优化器，并将该虚拟机迁移到有足够资源的物理机上。当物理机被判定处于超载状态时，本地管理器选择所要迁移的虚拟机。本地管理器以微小的时间间隔 t 向全局优化器发送所监控的物理机的运行状态信息。

3.2　计算策略

能耗估算：文献[11]分析了物理机各种部件的能耗情况，与其他部件相比，CPU 的功耗受使用率（负载率）变化的影响较大。依据 CPU 使用率提出功耗模型，并通过大量实验验证模型的准确性。物理机 PM_j 的功耗 P_j（Watt）可用式（1）计算：

$$P_j=\begin{cases}0, & \text{睡眠或关闭}\\ k\cdot P_j^{\max}+(1-k)\cdot a_j\cdot P_j^{\max}, & \text{空闲或活跃}\end{cases} \tag{1}$$

当物理机处于关闭状态或者深度休眠状态时，功耗为 0；当物理机处于工作状态时，功耗与 CPU 利用率近似呈线性关系。其中，a_j 为 CPU 的使用率，k 为比例系数（一般为 0.7），$P_j^{\max}$ 为 PM_j 满载时的功耗。若 PM_j 在$[t_{pj0},t_{pj1}]$处于工作状态，记 $t_{pj1}-t_{pj0}$ 为 t_{pj}，其 CPU 使用率函数为 $u_j(t)$，$k\cdot P_j^{\max}$ 记为 P^{idle}，那么 PM_j 在这段时间内的能耗为：

$$\begin{aligned}E_{pj}&=\int_{t_{pj0}}^{t_{pj1}}P_j\,\mathrm{d}t\\&=P^{idle}\cdot t_{pj}+(1-k)\int_{t_{pj0}}^{t_{pj1}}u_j(t)\cdot P_j^{\max}\mathrm{d}t\end{aligned} \tag{2}$$

在$[t_{pj0},t_{pj1}]$时间段内，所有活跃主机的总能耗为：

$$E=\sum_{i=1}^{M}E_{pj} \tag{3}$$

迁移代价：虚拟机的动态迁移虽然可以在不中断应用的情况下进行，但会造成一定的性能下降[12]。文献[8]中给出了估算性能下降的模型，针对 VM_i 的性能损耗定义为：

$$C_{vi}=l\cdot\int_{T_{vi}}u_i(t)\mathrm{d}t \tag{4}$$

$$T_{vi}=\frac{RAM_{vi}}{BW_{vi}} \tag{5}$$

式（4）中 l 是性能下降的比例系数（通常为 10%），$u_i(t)$ 是 t 时刻 VM_i 的 CPU 使用量，RAM_{vi} 是虚拟机 VM_i 所占用的内存量，BW_{vi} 是迁移虚拟机 VM_i 时的可用带宽。

SLA 违背：当虚拟机的需求得不到满足（物理机资源利用率达到 100%）时，产生 SLA 违背。SLA 违背代价 SLATAH[13]（SLA Violation Time per Active Host）的定义如下：

$$SLA=\frac{1}{M}\sum_{j=1}^{M}\frac{T_{s,j}}{T_{a,j}} \tag{6}$$

其中，M 是物理机的数目，$T_{s,j}$ 是 PM_j 的 SLA 违背的总时间，$T_{a,j}$ 是 PM_j 的总运行时间。式（6）反映了数据中心的平均 SLA 违背率。

4　算法设计

数据中心虚拟机的管理分为静态分配与动态管理。静态配置用于进行数据中心初始化的资源分配，动态管理根据数据中心的运行状态信息进行物理机配置。下面分别对这两方面的算法进行描述。

4.1　静态配置算法

数据中心初始化时的资源配置是一个背包问题，在虚拟机与物理机间进行匹配。根据第 3 节的定义，数据中心拥有不同种类的虚拟机，每个种类的虚拟机所需的资源量不同。待分配的虚拟机集合 $VM=\{VM_1,VM_2,VM_3,\cdots,VM_n\}$，物理机集合 $PM=\{PM_1,PM_2,PM_3,\cdots,PM_m\}$。结合式（1），$M$ 台物理机的总功耗为：

$$P=\sum_{j=1}^{M}P_j \tag{7}$$

初始化放置时，所有的虚拟机没有负载，此时对 CPU 使用率的计算可以使用虚拟机初始时所申请占用的 CPU 资源。若记 x_{ij} 为标示量，其取值如式（8）所示：

$$x_{ij}=\begin{cases}1, & \text{若 } VM_i \text{ 被放置在 } PM_j \text{ 上}\\ 0, & \text{其他}\end{cases} \tag{8}$$

其中，x_{ij} 用来表示每台虚拟机的放置情况（虚拟机与物理机的映射情况），若 VM_i 被放置在了 PM_j 上，相应的 x_{ij} 的值为 1，否则为 0。对于 N 台虚拟机集合 VM 与 M 台物理机集合 PM，完成分配后可得分配矩阵 X：

$$X=\begin{bmatrix}x_{11} & \cdots & x_{1m}\\ \vdots & \ddots & \vdots\\ x_{n1} & \cdots & x_{nm}\end{bmatrix} \tag{9}$$

此时，PM_j 的 CPU 资源的利用率可以用式（10）来计算：

$$a_j=\frac{\sum_{i=1}^{N}x_{ij}R_{vi,cpu}}{R_{pj,cpu}} \tag{10}$$

则静态放置问题可以描述为：

$$\text{Min: } P=\sum_{j=1}^{M}P_j \tag{11}$$

Subject to：

$$\sum_{i=1}^{N}x_{ij}R_{vi,k}\leqslant R_{vj,k}\cdot T_{high,j},\ \forall PM_j\in PM \tag{12}$$

$$\sum_{j=1}^{M}x_{ij}=1,\ \forall VM_i\in VM \tag{13}$$

问题目标是最小化总体功耗。式（12）表明物理机上虚拟机所占用的资源不能超过物理机所拥有的资源；式（13）用来约束每台虚拟机 VM_j 被唯一地分配到一台物理机上。针对虚拟机的分配问题，若虚拟机数量为 N，物理机数量为 M，则分配方案共有 M^N 种，根据约束条件去除不合适的分配方案，计算总体功耗，通过穷举法得出总体功耗最小的方案。但穷举法所无法在多项式时间内得到最优解，此问题是 NP-Hard 问题。

对于式（11），当所需的总计算资源一定时，减少总体功耗 P，等同于提高能效。借鉴文献[11]中的 PPW（Performace Per Watt）机制，PPW 的计算方法如式（14）所示：

$$PPW(PM_j)=Perf_j(a_j)/P_j \tag{14}$$

其中，$Perf_j(a_j)$ 表示 PM_j 在 CPU 使用率为 a_j 时的性能（单位为：MIPS，即每秒百万次指令数）。PPW 体现了物理机的能效，值越高表明提供相同计算力时，物理机的功耗越小。PPW 考虑了物理机的性能与能耗，适用于异构的数据中心。

根据计算的 PPW 值对物理机进行排序，将虚拟机安置

在能效高的物理机上以提高能效，从而降低总体功耗。

综上所述，物理机的静态安置算法 MDFF(Modified Decreasing First Fit)的伪代码如算法 1 所示。

算法 1 MDFF

```
Input:PM,VM
output:X
1. X 中所有元素置 0
2. 计算 PM 中 PMj 的 PPW 值(aj←Thigh,j)
3. PM 中 PMj 按 PPW 值降序排序
4. 对 VM 中 VMi 按 CPU 需求降序排序
5. For each VMi∈VM
6.    For each PMj∈PM
7.       xij←1
8.          If 满足式(12)
9.             Break;
10.         End if
11.      xij←0
12.   End for
13. End for
14. Return X
```

算法 1 描述了虚拟机静态调度的过程，第 3，4 行完成了对所有物理机 PPW 值的计算以及按照结果降序排序的过程，第 5 行对虚拟机按照 CPU 的需求量降序排序。然后采用最先适应算法，找到第一个能够安置 VM_i 的 PM_j，相应的 x_{ij} 置为 1。若虚拟机数目为 N，物理机数目为 M，则算法的复杂度为 MN。

4.2 动态调度

由于虚拟机中负载的动态性，需要对虚拟机资源进行动态调整以满足负载的资源需求。物理机面临资源利用率过高或过低等问题。迁移过载物理机上的虚拟机，以防止 SLA 违背。迁移低负载物理机上的虚拟机，然后关闭低资源利用率的物理机，从而达到减少能耗的目的。虚拟机的动态迁移主要解决以下几个问题：

1)如何确定阈值大小？

2)如何判定物理机处于过载还是低负载？

3)对于处于不正常状态的主机，如何确定被迁移的虚拟机？

4)如何放置迁出的虚拟机？

下面就这上述几个问题进行研究。

4.2.1 阈值的确定

设定阈值是虚拟机动态管理的有效方法。通过设定阈值，保证物理机运行在安全的状态下。阈值的上限决定了物理机资源的最大使用率，越高的上限越有利于提高资源使用率，但是会增加 SLA 违背的风险。阈值的下限主要用来防止物理机资源利用率偏低。

本文引用绝对中位差(Median Absolute Deviation，MAD)[13]统计方法来处理物理主机的历史信息。MAD 是统计离差的一种稳定的估计量，比样本方差或者标准差有更广泛的应用。对于数据集中的异常值，MAD 比标准差具有更好的容忍性。对于一组取样值 $X=\{X_1,X_2,X_3,\cdots,X_n\}$，其绝对中位差的计算式为：

$$MAD=median_i(|X_i-median_j(X_j)|) \tag{15}$$

定义 PM_j 阈值的上限为：

$$T_{high,j}=1-r\cdot MAD_j \tag{16}$$

PM_j 阈值的下限为：

$$T_{low,j}=0.4\cdot(1-r\cdot MAD_j) \tag{17}$$

其中，$r\in R^+$ 为自定义参数，由系统管理员进行设置。设置较小的参数 r 值，有助于提高物理机的资源利用率，降低能耗，但是会增加 SLA 违背的可能性，带来其他不利影响。

4.2.2 迁移时机的确定

负载需求的动态性导致虚拟资源使用情况不断变化。为了降低负载波动所带来的影响，本文采用滑动窗口的监测机制来确定超载与低负载。利用采集的物理机资源使用率的历史数据，判断物理机是否工作在安全的状态下。

对于设定的窗口内的 PM_j 所监测的历史数据，若是所监测的数据在安全的阈值之外的比例大于一定的数值，则触发相应的虚拟机迁移操作。若超过阈值上限的次数达到一定的比例，则认为 PM_j 处于过载的状态，选择迁移出一定的虚拟机。假如设置窗口大小为 5，PM_j 采集到了 5 条历史数据，若设定的比例值为 75%，表明当这 5 条数据中有 4 次在安全阈值之外时就触发了迁移操作。

采用滑动窗口的检测机制有利于避免负载波动所造成的虚拟机的不必要的迁移。对于历史数据收集较频繁的数据中心，可以设置较大的窗口值，反之可以采用较小的窗口值。窗口值越小，触发迁移操作的可能性越大。

4.2.3 迁移虚拟机的选择

对于低负载的物理机，采取的策略是迁出所有的虚拟机，以关闭该物理机，达到节能的目的。对于过载的物理机，应及时迁出合适的虚拟机，以达到减少 SLA 违背的目的。文献[13]中对过载物理机中所要迁移的虚拟机提出了 3 种选择策略：最大相关性(Maximum Correlation，MC)策略、最小迁移时间(Minimum Migration Time，MMT)策略以及随机选择(Random Selection，RS)策略。MC 策略分别计算每台虚拟机与其他虚拟机的相关性，然后选择相关性值最大的一个虚拟机进行迁移，这样有利于降低 SLA 违背的可能性。MMT 策略通过选取最小内存的虚拟机来减少迁移时间。RS 随机选择一个虚拟机进行迁移。MMT 未考虑到 CPU 资源的使用情况，在一次迁移后，CPU 资源的使用率仍然在阈值之外，需要迁移更多的虚拟机。本文在 MMT 的基础上，结合虚拟机 CPU 资源的使用情况提出新的选择策略。首先确定迁移哪些虚拟机能够使 CPU 资源利用率处在阈值内，然后计算这些虚拟机所占 CPU 资源与内存资源的比值，该比值代表了迁移虚拟机操作的效率，比值高说明在迁移相同内存大小的虚拟机时物理机 CPU 的使用率下降得多，迁移效果好。若迁移任何虚拟机都不能使物理机 CPU 资源利用率处于阈值内，则选择 CPU 资源利用率最高的虚拟机，以最大程度降低物理机 CPU 资源的利用率。假设超载物理机上有 n 台虚拟机，则算法的主要流程如下：

步骤 1 将超载物理集合中的所有虚拟机按照 CPU 资源利用率降序排序，假设排序后的虚拟机集合为 $\{VM_1,VM_2,VM_3,\cdots,VM_i,VM_n\}$；

步骤 2　估计迁移 VM_i 后物理机的 CPU 资源利用率是否处于阈值内，若处于则将 VM_i 加入候选集 H；

步骤 3　若 H 为空，则进入步骤 4，否则进入步骤 5；

步骤 4　选择 CPU 资源利用率最高的虚拟机加入迁移列表，更新物理机状态，转入步骤 2；

步骤 5　计算 H 集合中虚拟机所需要的 CPU 与所占用内存的比值 l；

步骤 6　选择比值 l 最大的虚拟机加入迁移列表；

步骤 7　选择结束，输出迁移列表。

4.2.4　虚拟机的迁移

迁移出的虚拟机应放置在合适的物理机上。本节主要介绍物理机的选择机制。目前主要的安置方法有 Power Aware Best Fit Decreasing(PABFD)[13] 和最小相关性系数法(Minimum Correlation Coefficient，MCC)[14]。PABFD 先对待迁移的虚拟机按照 CPU 降序排序，通过式(1)、式(10)估算虚拟机放置在物理机上的功耗增加量，然后选择功耗增加量最小的物理机。MCC 通过虚拟机与目标物理机运行的历史数据，计算虚拟机与每台物理机间的相关性系数，然后选择相关性系数最小的物理机，这样有助于降低 SLA 违背的可能性。在静态放置 PPW 的基础上，本文考虑物理机的资源使用情况，提出了物理机选择算法。

$$f(a_j)=1-\frac{1}{1+\mathrm{e}^{-\frac{\lambda}{T_{high,j}}(a_j-\beta\cdot T_{high,j})}} \tag{18}$$

式(18)是负载控制函数，在 PPW 机制的基础上综合考虑物理机的资源利用率。$T_{high,j}$ 是 PM_j 阈值的上限，λ 是控制函数下降程度的参数，β 用于调整资源的利用程度，β 值越大，越有助于增大物理机的资源利用率，但是选择资源利用率高的物理机会增加违背 SLA 的可能性。物理机的选择标准为：

$$H(a_j)=PPW(PM_j)\cdot f(a_j) \tag{19}$$

式(19)结合了 PPW 机制的特点，同时通过负载控制函数来限制选择负载较高的物理机，以在进行物理机选择时，优先选择活跃的物理机，避免开启新的物理机，增加能耗。在现有的物理机无法安置虚拟机时，选择开启新的物理机。

虚拟机动态迁移安置算法如算法 2 所示。

算法 2　虚拟机迁移安置算法

```
Input：vm，PM
output：pm
1. 在 PM 集合中选出所有工作在阈值内的物理机集合 PM′
2. 对于 PM′中的物理机，利用式(19)进行计算并将结果降序排序
3. For PMj∈PM′
4.      if vm 能被安置在 PMj 上
5.          pm←PMj
6.          Return pm
7.          break
8.    End if
9. End for
10. 在物理机集合中选取资源最多且能够安置 vm 的物理机
11. 若活跃的物理机无法安置 vm，则开启新的物理机
```

算法 2 描述了迁移物理机的选择策略，第 2，3 行对处于阈值内的物理机集合 PM' 中的物理机按照 $H(a_j)$ 值的大小进行排序，第 4－10 行首先在 PM' 中找到能够安放待迁移虚拟机的 PM_j，若没有合适的物理机，则在活跃主机中寻找资源最多的物理机，若现有的活跃物理机无法安置待迁移的虚拟机，则开启新的物理机安置虚拟机。

5　仿真实验与结果分析

本节通过仿真的方法对所提出的数据中心虚拟机节能管理机制进行性能分析。首先介绍实验环境和参数设置，然后针对实验数据进行分析。

5.1　实验设置

本文实验采用了 CloudSim toolkit[15] 平台创建了一个模拟数据中心。数据中心由一定量的物理机与虚拟机构成。为了使实验更贴近真实环境，本文使用了 CloudSim toolkit 中自带的真实负载数据模拟虚拟机的运行。本文构建了由 800 个物理主机与 1052 个虚拟机构成的数据中心，用于运行真实的负载，来测试本文调度机制的性能。物理主机的特征如表 1 所列，虚拟主机的特征如表 2 所列。

表 1　物理机特征

主机类型	MIPS	内存/GB	带宽/(GB/s)	存储/GB	最大功耗/W
PM1	2660	4	1	1000	135
PM2	1860	4	1	1000	117

表 2　虚拟机特征

虚拟机类型	MIPS	内存/GB	带宽/(GB/s)	存储/GB
VM1	2500	0.87	0.1	2.5
VM2	2000	1.74	0.1	2.5
VM3	1000	1.74	0.1	2.5
VM4	500	0.613	0.1	2.5

物理主机的相关数据来自于 SPEC[16]，该组织提供了多种物理机的测试数据。本文根据真实的测试数据，使用 Matlab 拟合出了物理机的功耗估计函数，与性能估计函数用于数据中心能耗与 PPW 值的估算。

除了使用真实的负载数据进行实验，本文还生成了不同规模的物理机与虚拟机来测试不同算法的性能。

本文实现了几种虚拟机静态分配算法：循环适应算法(Round Fit，RF)与 MDFF(本文算法)。结合文献[13]中的几种算法，进一步实现了以下几种不同的数据中心节能机制。

DVFS[8](Dynamic Voltage and Frequency Scaling)：采用了动态电压管理技术，根据物理机的实际使用情况调整物理机电压，从而达到减少数据中心能耗的目的。此管理机制未开启虚拟机的迁移，当大量物理机资源利用率较低时，不能通过虚拟机的迁移来减少活跃主机的数量和降低能耗，是作用于单物理机的节能管理机制。

MADMC[13](Median Absolute Deviation and Maximum Correlation)：采用了虚拟机动态迁移来减少数据中心能耗。为了降低能耗，采取了积极的物理机合并策略，尝试关闭资源使用率最低的物理机，迁移该物理机上所有的虚拟机。MADMC 计算同台物理机上的虚拟机之间在资源利用率方面的相关性，选择迁移相关性最大的虚拟机。但该机制未能

保证该虚拟迁移后使物理机资源利用率处于阈值内，会引起多次迁移，增加迁移开销。

本文虚拟机选择策略尽量减少虚拟机的迁移，以提升迁移效率。

5.2 实验结果与分析

图 2—图 4 分别示出了能耗、虚拟机迁移次数、SLA 在不同窗口大小下随 r 值的变化情况。在能耗方面，从图 2 中可以看出，在窗口大小 $s=1,3$ 时，能耗随着 r 值的增大呈现增加的趋势；由式(16)、式(17)可以推断，在 r 值变大时，阈值的上限会变小，阈值的下限会增大，限制了资源使用。在 r 值为 0 时阈值的上限与下限一直维持最大值，有利于增大资源使用率，但会增加 SLA 违背的风险。3 种窗口大小下，当 r 值为 0 时，几乎都取得了最小能耗，但 $s=2$ 比 $s=1$ 多了 5.36kWh 的能耗。在虚拟机迁移次数方面，总体上，虚拟机迁移次数随 r 值的增大而增加。在窗口大小为 2 时，虚拟机迁移次数较少。在窗口大小为 1 时，虚拟迁移次数较多，迁移的触发受瞬时值的影响较大。在 SLA 方面，窗口大小为 3 时，SLA 较高，说明窗口大小为 3 时，不能及时对虚拟机进行调整，从而造成了较高的 SLA 违背。通过比较不同窗口、不同 r 值下本文调度机制在能耗、虚拟机迁移次数、SLA 方面的表现，针对所使用的真实负载数据，最终设定了窗口大小 $s=2$，$r=2.5$ 来测试在不同数量的物理机与虚拟机的情况下，本文调度机制(EAMVM)，DVFS，MADMC 在能耗、虚拟机迁移次数、SLA 违背率方面的变化情况。

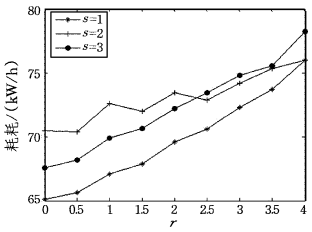

图 2 能耗

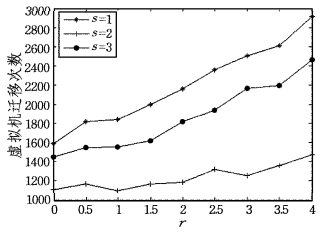

图 3 虚拟机迁移次数

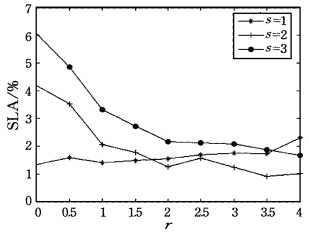

图 4 SLA

图 5 给出了不同规模的数据中心使用不同管理机制时的能耗情况(能耗根据式(3)得出)。由图 5 可知，采取了虚拟机动态迁移的管理机制比未采用虚拟机迁移的管理机制消耗的电能少。对于 DVFS 而言，MDFF 初始放置算法下的电能消耗明显小于 RF 初始放置算法，因为 RF 分配算法开启了过多的物理机，造成了巨大的能耗。在不开启虚拟机动态迁移的情况下，数据中心不能根据资源的实际使用情况对虚拟机的放置做出调整，浪费了大量的资源，增加了能耗，DVFS 机制受初始安置算法的影响较大。对于采用了虚拟机动态迁移的虚拟机管理机制而言，两种初始放置算法在能耗的表现上相近，都明显低于未开启虚拟机迁移的 DVFS。通过虚拟机的动态迁移能够弥补初始放置时的不足，根据虚拟机中负载的需求进行虚拟机放置的动态调整对能耗的降低起了主要的作用。

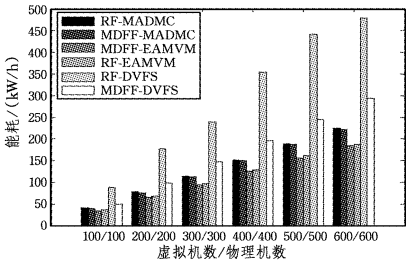

图 5 能耗比较

图 6 给出了不同规模的数据中心使用不同管理机制时的虚拟机迁移数目。虚拟机的迁移会加重数据中心的负担，因此减少虚拟机迁移是必要的。结合图 5，在相同初始放置算法下，与 MADMC 相比，EAMVM 在取得较好能耗表现的同时进行了较少的虚拟机迁移操作。MADMC 算法采取了积极的合并策略，为提高资源利用率进行了大量的虚拟机迁移操作，造成了巨大的迁移开销，频繁的虚拟机迁移给减少能耗带来了不利的影响。EAMVM 采用了动态阈值以及滑动窗口的方法，使物理机资源利用率处于阈值之内，防止过度合并物理机，同时通过设置窗口的方式减少了不必要的虚拟机迁移操作，减少了虚拟机的迁移次数。

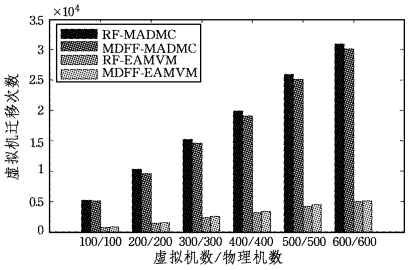

图 6 虚拟机迁移次数的比较

图 7 给出了不同规模的数据中心使用不同管理机制时的 SLA 违背率的变化情况。与 EAMVM 相比，MADMC 在 SLA 违背率方面比较稳定，但高于 EAMVM。MADMC 积极的虚拟机迁移策略造成了过多的虚拟机迁移操作，使得物理机承载了更多的虚拟机，给物理机增加了许多负担，更容易引起 SLA 违背，使 SLA 违背率偏高。EAMVM 的 SLA 违背率随着数据中心规模的增大而增加。随着虚拟机数目的增加，动态调整下物理机承载了更多的虚拟机，增加了 SLA 违背的风险。

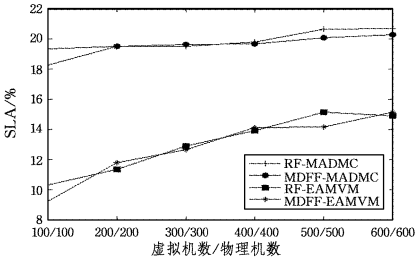

图 7 SLA 比较

结束语　本文研究了数据中心的虚拟机管理机制,在虚拟机的初始放置阶段,通过 MDFF 算法优化数据中心能耗。在虚拟机动态调整阶段,利用动态阈值的方法,根据负载的需求动态整合虚拟机,达到降低能耗的目的。对于超载的物理机节点,综合考虑迁移时间与 CPU 资源占用,减少迁移代价,降低 SLA 违背率。仿真实验表明,本文调度机制在能耗、虚拟机迁移次数方面取得了较好的效果。下一步将研究某些特定应用的特征,利用应用特征信息更好地进行虚拟机的初始配置,减少虚拟机动态迁移的消耗。

参 考 文 献

[1] State of the Data Center 2011 [EB/OL]. [2016-08-05]. http://www.emersonnetworkpower.com/en-US/Solutions/infographics/Pages/2011DataCenterState.aspx.

[2] YE K J,WU Z H,JIANG X H,et al. Power Management of Virtualized Cloud Computing Platform[J]. Chinese Journal of Computers,2012,35(6):1262-1285. (in Chinese)
叶可江,吴朝晖,姜晓红,等. 虚拟化云计算平台的能耗管理[J]. 计算机学报,2012,35(6):1262-1285.

[3] DONG Y,ZHOU L,JIN Y,et al. Improving Energy Efficiency for Mobile Media Cloud via Virtual Machine Consolidation[J]. Mobile Networks and Applications,2015,20(3):370-379.

[4] HIEU N T,DI FRANCESCO M,JÄÄSKI A Y. A virtual machine placement algorithm for balanced resource utilization in cloud data centers[C]// 2014 IEEE 7th International Conference on Cloud Computing. IEEE,2014:474-481.

[5] HUANG Z N,LI H S,ZHAO J. Virtual Machine Placement Algorithm Based on Improved Genetic Algorithm[J]. Computer Science,2015,42(S2):406-407,416. (in Chinese)
黄兆年,李海山,赵君. 基于双适应度遗传算法的虚拟机放置的研究[J]. 计算机科学,2015,42(S2):406-407,416.

[6] ZHU X,YOUNG D,WATSON B J,et al. 1000 islands:Integrated capacity and workload management for the next generation data center[C]// International Conference on Autonomic Computing,2008(ICAC'08). IEEE,2008:172-181.

[7] ADHIKARI J,PATIL S. Double threshold energy aware load balancing in cloud computing[C]// 2013 Fourth International Conference on Computing,Communications and Networking Technologies (ICCCNT). IEEE,2013:1-6.

[8] BELOGLAZOV A,BUYYA R. Adaptive threshold-based approach for energy-efficient consolidation of virtual machines in cloud data centers[C]// Proceedings of the 8th International Workshop on Middleware for Grids,Clouds and e-Science. ACM,2010:1-6.

[9] BEATY K A,BOBROFF N,KOCHUT A. Dynamic placement of virtual machines for managing violations of service level agreements(SLAs):U. S. Patent 8,291,411[P]. 2012-10-16.

[10] TANG Z,MO Y,LI K,et al. Dynamic forecast scheduling algorithm for virtual machine placement in cloud computing environment[J]. The Journal of Supercomputing,2014,70(3):1279-1296.

[11] FAN X,WEBER W D,BARROSO L A. Power provisioning for a warehouse-sized computer[C]// International Symposium on Computer Architecture(DBLP). 2007:13-23

[12] XU F,LIU F,LIU L,et al. iaware:Making live migration of virtual machines interference-aware in the cloud[J]. IEEE Transactions on Computers,2014,63(12):3012-3025.

[13] 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.

[14] FU X,ZHOU C. Virtual machine selection and placement for dynamic consolidation in Cloud computing environment[J]. Frontiers of Computer Science,2015,9(2):322-330.

[15] CALHEIROS R N,RANJAN R,BELOGLAZOV A,et al. CloudSim:a toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms[J]. Software:Practice and Experience,2011,41(1):23-50.

[16] SPECpower_ssj2008 Results [EB/OL]. [2016-08-05]. http://www.spec.org/power_ssj2008/results.

(上接第 13 页)

[45] BADER J,ZITZLER E. HypE:An Algorithm for Fast Hypervolume Based Many-objective Optimization[J]. IEEE Transactions on Evolutionary Computation,2011,19(1):45-76.

[46] FALCON C J G,COELLO C A C. IMOACO$_R$:A New Indicator Based Multiobjective Ant Colony Optimization for Continuous Search Spaces[M]. Springer,2015.

[47] MANSOUR I B,ALAYA I. Indicator Based Ant Colony Optimization for Multiobjective Knapsack Problem[J]. Procedia Computer Science,2015,60(1):448-457.

[48] LI K,ZHANG Q F,BATTITI R. MOEA/D-ACO:A Multiobjective Evolutionary Algorithm Using Decomposition and Ant Colony[J]. IEEE Transactions on Cybernetics,2013,43(6):1845-1859.

[49] STÜTZLE T,HOOS H H. Max Min Ant System[J]. Futrure Generation Computer Systems,2000,16(9):889-914.

[50] SOUZA M Z D,POZO A T R. Multiobjective Binary ACO for Unconstrained Binary Quadratic Programming[C]// Brazilian Conference on Intelligent Systems. 2015:86-91.

[51] WANG H D,JIAO L C,YAO X. Two_Arch2:An Improved Two-Archive Algorithm for Many-Objective Optimization[J]. IEEE Transactions on Evolutionary Computation,2015,19(4):524-541.

[52] BATTI R,BRUNATO M,MASCIA F. Reactive Search and Intelligent Optimization [J]. Operations Research/Computer Science Interfaces Series,2008,45(3):151-153.