文章编号:1671-4598(2020)10-0241-06 DOI:10.16526/j.cnki.11—4762/tp.2020.10.049 中图分类号:TP393.093 文献标识码:A

虚拟机迁移中一种新的物理主机异常状态检测算法

徐胜超
(广东财经大学华商学院 数据科学学院,广东 广州 511300)

摘要:提出了一种新的物理主机异常状态检测算法 PHSDA (physical host status anomalous detection algorithm);PHSDA 算法包括两个阶段;在超负载检测中,采用一种迭代权重线性回归方法来预测物理资源的使用效率情况;在低负载检测中,利用多维物理资源的均方根来确定其资源使用阈值下限,避免异常状态的物理主机数量的增加;PHSDA 检测算法配合迁移过程中后续的虚拟机选择策略和虚拟机放置策略,就可以形成一个全新的虚拟机迁移模型 PHSDA—MMT—BFD;以 CloudSim 模拟器作为 PHSDA 的仿真环境;经 PHSDA 策略优化过后的新虚拟机迁移实验表明:与近几年的 BenchMark 迁移模型比较起来,可以很好地降低云数据中心的能量消耗,虚拟机迁移次数减少,云服务质量明显提高。

关键词:智能计算;物理主机;虚拟机合并;虚拟机分配;物理主机状态检测

A New Physical Host Status Anomalous Detection Algorithm in Virtual Machine Migration

Xu Shengchao
(School of Data Science, HuaShang College, Guangdong University of Finance & Ecnomics, Guangzhou 511300, China)

Abstract: A new physical host status anomalous detection algorithm called PHSDA was proposed in this paper. PHSDA includes two phases, overloading host detection and under loading host detection. In overloading host detection, it used an iterative weighted linear regression method to determine two utilization thresholds and avoid performance degradation. In under loading host detection, PHSDA used a vector magnitude squared of multiple resources to consolidate active hosts. United with Subsequent strategy in virtual machine selection and virtual machine placement, a novel virtual machine migration model called PHSDA—MMT—BFD had been formed. PHSDA had been evaluated using CloudSim tools. Experimental results show that compared with benchmark VM migration models, energy consumption and numbers of VM migration had been reduced in PHSDA. The Qos of cloud service provider had been also promoted.

Keywords: intelligent algorithm; physical host; virtual machine consolidation; virtual machine allocation; physical host status detection

0 引言

在云数据中心中如何检测出运行异常的物理节点并进行负载均衡操作是一个关键问题[1-2],目前大部分云服务提供商都采用虚拟机迁移技术[3]。

物理主机异常状态检测需要对即将迁移的源物理主机和目标物理主机都要进行负载判断,这些负载基本都封装成虚拟机的形式,周期性的检测之后,形成负载异常的物理主机列表,从而为后续的虚拟机选择过程和虚拟机放置过程提供输入参数。

目前已有的物理主机状态检测方法大多采用静态资源使用率阈值边界来确定主机是超负载或者低负载,其资源的边界考虑的维度因素也比较单一;其针对的云客户端也不是自适应的,以被动的方式检测为主。因为物理主机资源状态是一种随着时间和应用程序的访问而不断动态变化的,这对主机异常状态检测方法提出了新的要求。

为此本文提出了一种新的物理主机异常状态检测算法 PHSDA (physical host status detection algorithm)。PHSDA 是一种自适应的、动态的物理主机检测方法,它采用时间序列和线性回归来预测出物理主机在未来的一段时间内的资源使用率情况,来确定其阈值边界,是一种主动检测策略。PHSDA 检测策略配合后续虚拟机选择阶段和虚拟机放置阶段的其它优化方法,就构成了一个完整的新型虚拟机迁移模型。最后描述了 PHSDA 物理主机状态检测策略的实现和仿真,结果表明 PHSDA 可以选择适当的时刻进行虚拟机的迁移,提高物理资源的利用效率,降低能量消耗。

1 相关工作

本文重点考虑物理主机状态检测阶段,在 Cloudsim 工

收稿日期:2020-07-28; **修回日期**:2020-08-31。
基金项目:国家自然科学基金重点项目(60433040);国家自然科学基金项目(50577027);Intel 大学合作计划。
作者简介:徐胜超(1980-),男,湖北武汉人,硕士,讲师,主要从事计算机网络和云计算方向的研究。

具包中也称为物理资源阈值管理策略，这种资源阈值管理策略可以分为主动法和被动法，其中被动的方法意味着在物理主机的资源已经超过阈值边界之后再采取动作；主动的方法是指通过观察资源使用的样本数据在前一阶段的利用率情况，提前预测出可能出现状态的物理主机，接着进行虚拟机迁移的后续步骤。

PHSDA 依托于 Cloudsim 项目，它把物理主机异常划分为超负载 over－utilized 或者低负载 under－utilized 状态，常见的有 6 种策略[3]。

文献[4]提出一个自适应的三阈值主机状态检测方法，把物理主机根据资源使用阈值划分为小负载，轻负载，中负载和高负载四个状态，采用 k－means 算法来管理这些阈值，实验结果表明该物理主机状态检测算法可以减少云数据中心的能量消耗和 SLA 违规比率，但是性能提高不是很明显。

文献[5]采用一个局部代理来检测物理主机状态，把物理主机划分为超负载、预高负载、低负载和正常状态 4 个状态，采用 LiRCUP 方法来预测超负载的物理主机，避免 SLA 违规率，测试结果表明它比 Cloudsim 中已有的检测方法性能有提升。

近年来也有大量的采用智能算法进行优化物理主机状态检测的文献，例如处理器温度感知[6]、遗传算法[7]、虚拟机关联性[8]、二次指数平滑预测[9]等，这些文献在研究思路，测试指标等方面基本上都参考了 Cloudsim 项目。

2 工作背景与相关术语

2.1 物理主机状态检测的工作场景

图 1 显示了 PHSDA 物理主机状态检测的工作场景，这是一个典型的云客户端对云数据中心的服务请求场景。PHSDA 依托了 Cloudsim 各个运行模块：全局代理 Global Broker、本地代理 Local Broker、虚拟机管理器 Virtual Machine Manager。

在 Cloudsim 中每个物理主机上都运行有一个本地代理 Local Broker，PHSDA 优化策略的实现主要在此模块中完成。这个工作场景在文献都有描述[3]。

2.2 PHSDA 优化的虚拟机迁移流程

PHSDA 运用到整个虚拟机迁移过程中，工作流程具体包括下面 4 个步骤：

步骤 1：基于 PHSDA，周期性的检测云数据中心的物理主机，形成 HostsToMigrateList；

步骤 2：基于最小迁移时间选择 MMT 算法完成虚拟机选择[3]，形成 selectedVMList；

步骤 3：基于递减装箱算法 BFD 优化完成虚拟机的重新放置[3]；

步骤 4：重复上述步骤 1～3，通过设置一个是周期（通常是一周），达到该时间段就结束；

图 2 显示了经 PHSDA 物理主机状态检测优化后的虚拟机迁移工作流程。

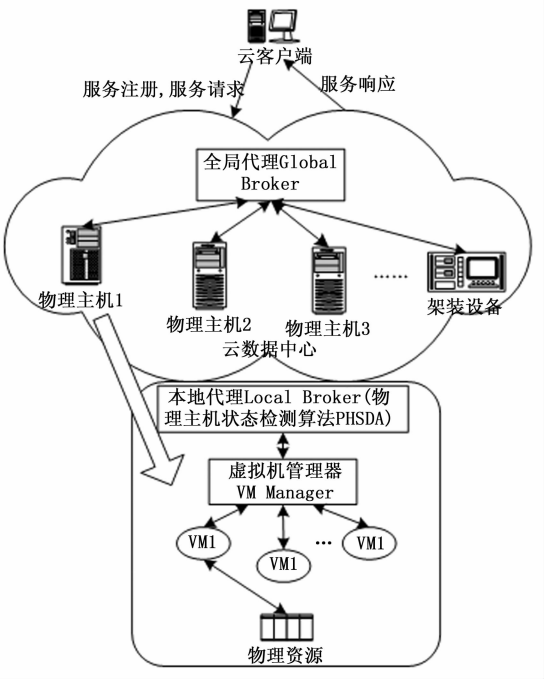

图 1 PHSDA 物理主机状态检测的工作场景

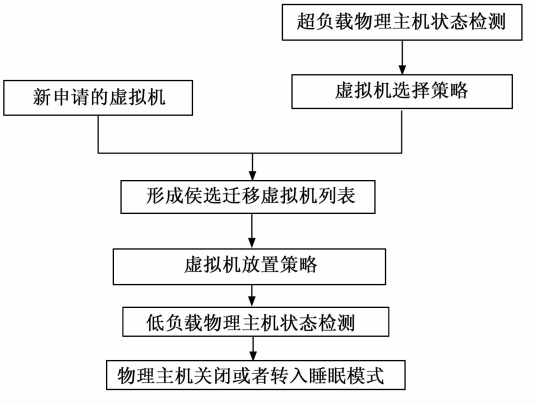

图 2 PHSDA 优化的虚拟机迁移工作流程

2.3 相关术语

2.3.1 云数据中心的能量消耗

云数据中心主要由大量的堆积在一起的物理主机组成，所以其能量消耗主要由物理主机的所有部件的能量消耗组成。已有经验表明一个物理服务器（双核 CPU、四条内存、一个磁盘、2 个 PCI 插槽，一个主板等）所消耗的能量大约为 CPU 占 41%，内存占 18%，磁盘占 7%，PCI 插槽占 23%，主板占 12%。基于这个思路，PHSDA 策略中设计的物理主机的能量消耗数学模型如公式（1）～（6）：

$$E(U_{cpu}) = E_{idle} + (E_{\max} - E_{idle})U_{cpu} \tag{1}$$

$$U_{cpu}(t) = \sum_{c=1}^{PE_j} \sum_{i \in r_j(t)} \frac{mips_{i,c}}{MIPS_{j,c}} \tag{2}$$

$$E(U_{mem}) = E_{idle} + (E_{\max} - E_{idle})U_{mem} \tag{3}$$

$$E(U_{bw}) = E_{idle} + (E_{\max} - E_{idle})U_{bw} \tag{4}$$

$$E_{host} = E(U_{cpu}) + E(U_{mem}) + E(U_{bw}) \tag{5}$$

这里 $U_{cpu}(t)$、$U_{mem}(t)$、$U_{bw}(t)$ 表示物理主机在 t 时刻的 CPU 使用率、内存使用率、磁盘使用率、网络带宽使用率：

$$0\% \leqslant U_{cpu}(t), U_{mem}(t), U_{bw}(t) \leqslant 100\% \tag{6}$$

E_{idle} 表示物理主机 CPU、内存、网络带宽在空闲时的能量消耗，即是 $U_{cpu}(t)=0$、$U_{mem}(t)=0$、$U_{bw}(t)=0$ 的时候。

$E_{\max}$ 表示物理主机在满负载时的能量消耗，即是 $U_{cpu}(t)=100\%$、$U_{mem}(t)=100\%$、$U_{bw}(t)=100\%$的时候。

$mips_{i,c}$ 是第 i 个虚拟机 VM_i 的第 c 个处理单元的 $mips$ 请求情况。

$MIPS_{j,c}$是第 j 个物理主机 M_j 的第 c 个处理单元的整体的 MIPS 计算能力。pe_i 表示虚拟机 VM_i 的处理单元的数量，PE_j 表示物理主机 M_j 的处理单元的数量，$r_j(t)$ 表示分配到物理主机 M_j 的虚拟机的索引集合。

一个虚拟机请求的 MIPS 的数量是随着应用程序变化而变化的，所以物理主机的资源使用率也应该是随着应用程序的变化而变化，因此统计物理服务器的能量消耗必须在一定的时间段内，这样根据公式（1）可以演化为公式（7）：

$$E_{host}(t)=E(U_{cpu}(t))+E(U_{mem}(t))+E(U_{bw}(t)) \tag{7}$$

这样第 j 个物理主机在 $[t_0,t_1]$ 时间段的总体能量消耗 E_{host} 可以按照公式（8）来计算：

$$E_j=\int_{t_0}^{t_1} E_{host}(t)\mathrm{d}t \tag{8}$$

整个云数据中心的能量消耗为：

$$E_{total}=\sum_{j=1}^{M} E_j \tag{9}$$

2.3.2　SLA 违规在线时间

当一个云客户端提交作业到云计算平台的时候，资源缺少就会出现（SLA，service level agreement）违规，在虚拟机分配过程中，一个重要性能指标就是每个物理主机的 SLA 在线时间 SLA violation Time per Active Host（SLA-TAH），它体现了物理主机具有高服务质量的在线时间情况。

$$SLATHA=\frac{1}{M}\sum_{i=1}^{M}\frac{T_{s_i}}{T_{a_i}} \tag{10}$$

云数据中心的主机数量和虚拟机数量分别由 M 和 N 表示，其中 T_{s_i}是物理主机 CPU 利用率达到 100%的时间，T_{a_i} 是物理主机处于在线活跃状态的时间。

2.3.3　虚拟机迁移后的性能降低（PDM，performance degradation due to migrations）

$$PDM=\frac{1}{N}\sum_{i=1}^{N}\frac{C_{d_i}}{C_{r_i}} \tag{11}$$

其中：C_{d_i} 是由于虚拟机 VM_i 迁移导致的性能下降的估计值，C_{r_i} 是请求虚拟机 VM_i 的整个时间段内总的 CPU MIPS 计算能力。

2.3.4　能量与 SLA 违规的联合指标（ESV）

SLA 的违规率的计算通过公式（12）计算：

$$SLA\ Violation = SLATAH * PDM \tag{12}$$

联合指标 ESV 是 SLA 违规比率和总体能量消耗平衡的指标，其值的大小至关重要，公式为（13），其中 $E_{total}=\sum_{j=1}^{M}E_j$ 为云数据中心整体能量消耗。公式为：

$$ESV = E_{total} * SLA\ Violation \tag{13}$$

3　物理主机状态检测算法描述

3.1　超负载主机检测过程

正如前面所提到的，PHSDA 把云数据中心的物理主机按照工作状态划分为 4 类：超负载（over－loaded）、高负载（under－pressure）、正常状态（normal）、低负载状态（under－loaded），这样其对应的物理资源的阈值边界也对应三个：Upper － Thredholds 、Pre － Thredholds、Lower － Thredholds 如图 3 所示。

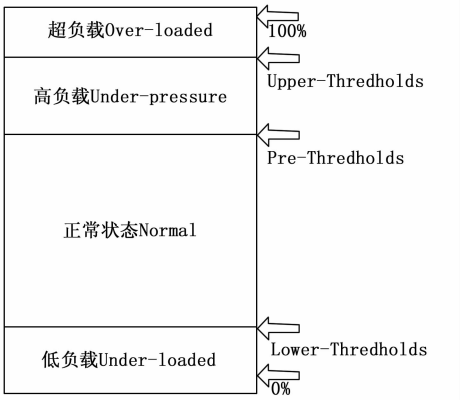

图 3　PHSDA 中物理主机的状态分类

PHSDA 的最终目的是从云数据中心的物理主机中检测出异常状态的节点，通过设置一个时间间隔，周期性的执行 PHSDA 算法。PHSDA 检测算法是基于物理主机的 CPU 资源、内存资源、网络带宽资源的使用情况。当 PHSDA 算法启动后，周期性的检测这 3 个资源的使用情况，并尽量使其在合适的阈值范围内（Upper － Thredholds、Pre － Thredholds、Lower－Thredholds）。如果有某个物理主机的资源使用情况不在这个合理的范围内，它将被列入侯选迁移物理主机列表 HostsToMigrateList。

回归是统计学中的一种量化数据分析方法，它可以预测数据的下一阶段的值，回归方法被广泛使用在数据预测领域[10]。回归技术有两种模型，单个输入的单一回归和多个输入的多回归。它采用回归函数（线性或非线性）来估计出输入变量 X 和输出变量 Y 之间的关系。PHSDA 算法采用了单一权值线性回归来预测物理主机的资源使用效率情况。该思想如公式（14）所示：

$$Y=\beta_0+\beta_1 X \tag{14}$$

这里 Y 是受依赖的变量，X 是独立的变量，β_0 和 β_1 是回归系数，它们来自于最小二乘法技术[11]，如下所示：

$$\hat{\beta}_0=\overline{Y}-\hat{\beta}_1\overline{X} \tag{15}$$

$$\hat{\beta}_1=\frac{\sum_{i=1}^{n}(X_i-\overline{X})(Y_i-\overline{Y})}{\sum_{i=1}^{n}(X_i-X)^2} \tag{16}$$

$\overline{X}$ 和 $\overline{Y}$ 是变量 X 和 Y 的均值，$\hat{\beta}_0$ 和 $\hat{\beta}_1$ 是变量 β_0 和 β_1 的估计值，对于每个观察值 x_i 和 y_i 都分配一个相邻区域权重，如下面的公式（17）：

$$T(u)=\begin{cases}(1-|u|^3)^3, & \text{if } |u|<1\\ 0, & \text{if } |u|>1\end{cases} \tag{17}$$

基于上面的这个公式，相邻区域权重的定义如公式（18）：

$$w_i(x)=T\left(\frac{x_n-x_i}{x_n-x_1}\right)=\left(1-\left(\frac{x_n-x_i}{x_n-x_1}\right)^3\right)^3 \tag{18}$$

这里 x_i 和 x_n 是变量 x 的最近的第 i 个观察值，PHSDA 算法采用 k 次迭代来检测物理主机资源使用效率变量（host utilization）的 k 个将来预测值，对于 n 个数据变量，回归的函数定义如下：

$$\hat{y}_1=\beta_0+\beta_1 x_n,\hat{y}_2=\beta_0+\beta_1\hat{y}_1,\ldots,\hat{y}_k=\beta_0+\beta_1 y_{k-1} \tag{19}$$

正如图 3 所示，PHSDA 算法定义了 2 个阈值边界，Upper－Thredholds 和 Pre－Thredholds。对于一个给定的 k 值，如果资源使用的下一次迭代（$i=1$）预测值超过了它的整体提供能力（resource capacity）的 100%，那么该物理主机将被标识为超负载物理主机（overloaded），此时物理主机的资源使用边界为 Upper－Thredholds。如果 PHSDA 检测算法检测到将来的多次迭代预测值（$i=2\sim k$）超过了它整体提供能力（resource capacity）的 100%，该物理主机将标识为高负载物理主机（under pressure），$\hat{y}_1$ 的资源使用边界为 Pre－Thredholds，在这个情况下物理主机不会容纳新的虚拟机。在该算法中，如果把 $k=2$，即是 PHSDA 检测只估计两次变量的预测值，如公式（20）：

$$\hat{y}_1=\beta_0+\beta_1 x_n,\hat{y}_2=\beta_0+\beta_1\hat{y}_1 \tag{20}$$

在这个情况下有如公式（21）：

$$\begin{cases}x_n \text{ is upper threshold}, & \text{if } c.\hat{y}_1\geqslant 1\\ x_n \text{ is pre threshold}, & \text{if } c.\hat{y}_2\geqslant 1\end{cases} \tag{21}$$

这里 c 是密度常量，下面的 Algorithm 1 和 Algorithm 2 显示了物理主机状态检测 PHSDA 算法的伪代码。

Algorithm 1：Overloading Host Detection Algorithm
Input：host
Output：overloadedList；
1. UTC ←PHSDA（CPU）. upperThreshold；
2. PUC ←PHSDA（CPU）. utilPrediction；
3. UTM ←PHSDA（Memory）. upperThreshold；
4. PUM ←PHSDA（Memory）. utilPrediction；
5. UTB ←PHSDA（BW）. upperThreshold；
6. PUB ←PHSDA（BW）. utilPrediction；
7. if（（PUC or PUM or PUB）>=1） then
8. underPressureList ← host；
9. Host will not accept new VM；
10. else
11. if（（UTC or UTM or UTB）>=1） then
12. overloadedList ← host；
13. end if
14. end if
15. return overloadedList；

Algorithm 2：PHSDA algorithm
Input：host utilization
Output：utilPrediction
1. for i=1 to n do
2. $x_i \leftarrow i$；
3. $y_i \leftarrow$ utilHistory(i)；
4. $w_i \leftarrow$ calculate using equation (18)；
5. $x_i \leftarrow x_i * w_i$；
6. $y_i \leftarrow y_i * w_i$；
7. end for
8. calculate β_0 , using equation (15)；
9. calculate β_1 , using equation (16)；
10. utilPrediction=$\beta_0+\beta_1$ * currentUtil(h)；
11. upperThreshold = utilPrediction；
12. update x, y and w；
13. update β_0 and β_1 ；
14. for $i=2$ to k do
15. KPredictionUtil(i)=$\beta_0+\beta_1$ * utilPrediction；
16. utilPrediction=KPredictionUtil(i)；
17. end for
18. return utilPrediction；

3.2 低负载主机检测过程

在完成了超负载物理主机状态检测之后，就进入到后面的虚拟机选择和虚拟机放置阶段，这样 PHSDA 算法的第一阶段结束。在虚拟机迁移完成后，必要进入低负载物理主机检测阶段，PHSDA 可以把低负载物理主机转入睡眠模式或者关闭其电源，节省能量消耗。

PHSDA 主要是通过物理资源的比较低的四分位值 quartile 值来完成判断。将该 quartile 值指定为资源使用阈值下边界 Lower－thresholds。所以 $T_{low}=u_{(+(n+1))}$，这里 u 表示物理主机资源使用的利用率，n 表示数值在数据集中的编号。只要物理资源的处理器 CPU，内存 RAM 和网络带宽 BW 的利用率低于 T_{low}，则该主机将处理低负载状态（under－loaded），根据向量的平方根公式（22），公式（23）可以用来计算不同维度对变量的影响，然后可以对 Util 完成升序排列。

$$Util=\sqrt{a^2+b^2+c^2} \tag{22}$$

$$Util=\sqrt{(Util(CPU))^2+(Util(RAM))^2+(Util(BW))^2} \tag{23}$$

在低负载物理主机检测中，PHSDA 算法维护着一个低负载物理主机列表 under－loaded hosts，首先检测这些机器上的虚拟机是否可以迁移到其他的物理节点，对于一个可以容纳虚拟机的物理主机，必须具有 3 个条件：1）处理高负载状态；2）有足够的物理资源满足虚拟机的要求；3）在容纳虚拟机后，它不能变成超负载状态；如果低负载状态的物理主机上的虚拟机都迁移到其他的节点，它将切换到睡眠模式，进一步节省云数据中心的能量消耗。具体的

算法描述见下面 Algorithm 3 的伪代码：

Algorithm 3：Under loading Host Detection Algorithm

Input：hostList，hostVMlist

Output：VMmigrationList

1. for each h in hostList do

2. if ((h. $util_{CPU}$) < T_{low}(CPU)) && ((h. $util_{RAM}$) < T_{low}(RAM)) && ((h. $util_{BW}$) < T_{low}(BW)) then

3. underloadingList ← h;

4. end for

5. for each h in underloadingList do

6. $utilCPU=(\frac{allocatedMips}{TotalMips})^2$

7. $utilRAM=(\frac{allocatedRam}{TotalRam})^2$;

8. $utilBW=(\frac{allocatedBw}{TotalBw})^2$;

9. $Util=\sqrt{utilCPU+utilRAM+utilBW}$

10. underloadingList. sortIncreasingUtil();

11. end for

12. for each h in underloadingList do

13. for each VM in hostVMlist() do

14. for each host in hostList do

15. if(host ∉ underPressureList) then

16. if ((host has enough CPU，RAM and BW) && (Not overloaded after VM migration)) then

17. VMmigrationList ← h. VM;

18. hVMlist← hVMlist− h. VM;

19. break;

20. end if

21. end if

22. end for

23. end for

24. if (hVMlist = null)

25. return VMmigrationList;

26. end if

27. end for

4　仿真实验与性能分析

4.1　仿真环境

因为 PHSDA 物理主机异常检测算法是在虚拟机迁移过程中运用的，所以进行 PHSDA 实验分析，我们构造了 Cloudsim3.0 云数据中心的虚拟机迁移场景，云数据中心的能量消耗模型及测试指标都参考了最常见的 CoMon project，它是由 planetlab 实验室开发的一个项目[12]。在 CoMon 项目中设置的云数据中心主要由两类物理服务器组成，物理服务器总数为 800 个，物理服务器配置如表 1 所示。

一周内不同虚拟机请求个数如表 2 所示。

4.2　评测标准与比较对象

本文的 PHSDA 物理主机异常检测算法结合其他虚拟机选择和虚拟机放置阶段，即形成了 PHSDA－MMT－BFD 虚拟机迁移模型。

表 1　云数据中心物理服务器硬件配置

主机个数	CPU 提供能力/MIPS	内存大小/GB	磁盘空间/GB	网络带宽/(GB/s)
200	1 600	4	1 000	1
200	1 600	4	500	1
200	1 600	8	500	1
200	2 200	8	500	1

表 2　一周内不同的虚拟机请求个数

日期	虚拟机个数	物理主机个数	运行时间/h
周一	1 000	800	24
周二	1 000	800	24
周三	1 000	800	24
周四	1 000	800	24
周五	1 000	800	24

我们还与近年来的其他物理资源阈值管理办法进行了比较，例如 LiRCUP 检测方法、ATEA 检测方法，分析这些物理主机异常状态检测对云数据中心的性能改变情况。综上所述，本实验一起涉及到的虚拟机迁移模式如表 3 所示。

表 3　PHSDA 物理主机状态异常检测略性能比较对象

虚拟机迁移模式	相关说明
PHSDA－MMT－BFD	物理主机状态检测阶段优化
LRR－MMT－BFD[3]	Cloudsim 项目原始模型
LiRCUP[13]	物理主机状态检测阶段优化
ATEA[5]	物理主机状态检测阶段优化
Stable－Matching[14]	虚拟机选择和放置阶段优化
Correlation[8]	虚拟机放置阶段优化
ACS－VMM[5]	虚拟机放置阶段优化

4.3　仿真结果与性能分析

4.3.1　云数据中心总体能量消耗

利用 PHSDA 物理主机状态检测阶段的优化之后，各个虚拟机迁移模型一周之内的总体能量消耗如表 4 所显示。PHSDA－MMT－BFD 迁移模型比其他各个迁移策略在总体能量消耗上要节约 15%～30%。分析原因是 PHSDA－MMT－BFD 模型可以很好地检测出异常物理主机的状态，确定好迁移的时刻，资源利用效率自然提高，关闭没有必要启动的空闲物理主机，所有云数据中心的总体能量消耗自然减少。

表 4　各类虚拟机迁移策略的总能量消耗比较　　kWh

能量消耗/kWh 迁移策略	周一	周二	周三	周四	周五
PHSDA－MMT－BFD	121	98	111	121	120
LRR－MMT－BFD	155	122	135	175	145
LiRCUP	162	99	128	122	116
ATEA	123	116	120	115	121
Stable－Matching	135	100	115	150	131
Correlation－Based	126	126	127	128	124
ACS－VMM	131	128	133	121	141

4.3.2 虚拟机迁移次数

表 5 显示了在一周的 5 天之内 PHSDA－MMT－BFD 的虚拟机迁移次数基本最小。原因是 LRR－MMT－BFD 很容易增加超负载或低负载的物理主机的数量，而 PHSDA－MMT－BFD 策略则与 LRR－MMT－BFD 正好相反，它利用回归预测法预测出物理资源的利用效率，动态地确定阈值边界，不会增加虚拟机迁移次数。另外 ACS－VMM 策略、Stable－Matching 策略的优化主要在虚拟机放置阶段，必须像 Correlation－Based 策略在虚拟机选择阶段、物理主机状态检测阶段完成优化才好降低迁移次数。

表 5 各类虚拟机迁移策略的虚拟机迁移次数

迁移次数/次 迁移策略	周一	周二	周三	周四	周五
PHSDA－MMT－BFD	11 000	9 600	9 300	10 000	9 800
LRR－MMT－BFD	16 000	13 000	16 000	21 000	17 500
LiRCUP	14 000	12 000	13 000	19 000	16 000
ATEA	7 519	6 808	7 000	7 200	6 900
Stable－Matching	11 000	9 000	11 000	11 000	10 000
Correlation	10 147	9 546	9 546	9 792	9 792
ACS－VMM	15 000	14 800	13 500	16 000	13 000

4.3.3 SLA 违规率分析

从表 6 可以看出，周一到周五，PHSDA－MMT－BFD 都比较少出现 SLA 违规，原因是 PHSDA－MMT－BFD 比递减装箱办法的优化能力要强，而且从本文设计的能量消耗模型来看，PHSDA－MMT－BFD 考虑的迁移因素的维度范围不仅仅限制在 CPU 方面，还考虑了内存大小，磁盘空间及网络带宽。

LiRCUP、Stable－Matching、Correlation－Based 和 ATEA 在某些时候还优于 PHSDA－MMT－BFD 策略，PHSDA－MMT－BFD 以降低虚拟机迁移次数为目标，所以也会牺牲一些 SLA 违规率的增加。

表 6 各类虚拟机迁移策略的 SLA 违规率比较

SLA 违规率/% 迁移策略	周一	周二	周三	周四	周五
PHSDA－MMT－BFD	0.001 1	0.001 2	0.001 0	0.001 3	0.000 8
LRR－MMT－BFD	0.001 8	0.001 4	0.002 3	0.001 9	0.001 8
LiRCUP	0.000 9	0.000 9	0.000 9	0.001 1	0.000 9
ATEA	0.001 5	0.001 2	0.001 3	0.001 5	0.001 4
Stable－Matching	0.001 4	0.001 1	0.002 0	0.002	0.001 2
Correlation	0.001 1	0.000 9	0.001 3	0.001 1	0.000 9
ACS－VMM	0.001 2	0.001 4	0.002 1	0.001 8	0.001 5

4.3.4 能量与 SLA 违规的联合指标 ESV

从表 7 中的结果可以看到，PHSDA－MMT－BFD 的 ESV 指标不是最好的，Stable－Matching 和 Correlation－Based 策略针对原始的 Cloudsim 各个阶段都有优化，自然比 LRR－MMT－BFD 迁移策略性能好，ACS－VMM 迁移策略的 ESV 最低，它比单一目标函数的 PHSDA－MMT－BFD 迁移策略在 ESV 方面要低。

表 7 各类虚拟机迁移的 SLA 与能量消耗联合指标 ESV

ESV(无单位) 迁移策略	周一	周二	周三	周四	周五
PHSDA－MMT－BFD	0.08	0.08	0.09	0.08	0.09
LRR－MMT－BFD	0.23	0.18	0.32	0.34	0.27
LiRCUP	0.21	0.19	0.28	0.25	0.21
ATEA	0.08	0.11	0.13	0.12	0.13
Stable－Matching	0.13	0.09	0.25	0.19	0.15
Correlation	0.07	0.09	0.14	0.13	0.14
ACS－VMM	0.04	0.05	0.05	0.04	0.04

5 结束语

本文提出了虚拟机迁移中一种新的物理主机状态异常检测算法 PHSDA，PHSDA 利用统计学中回归函数操作，经过多次循环迭代预测出云数据中心的物理资源的利用效率的 Upper－Thredholds 、Pre－Thredholds 值；在低负载检测中利用多维资源的均方根确定利用效率下界 Lower－Thredholds；PHSDA 物理主机状态检测算法可以为其他虚拟机迁移过程作为参考。

参考文献：

[1] Usman M J, Ismail A S, Chizari H, et al. Energy－efficient virtual machine allocation technique using flower pollination algorithm in cloud datacenter: a panacea to green computing [J]. Journal of Bionic Engineering, 2019, 16 (2): 354－366.

[2] Shi T, Ma H, Chen G. Energy－aware container consolidation based on PSO in cloud data centers [A]. IEEE Congress on Evolutionary Computation [C]. IEEE, 2018.

[3] Cloudsim [EB/OL]. http://github. com/Cloudslab/cloudsim/. 2019.

[4] Zhou Z, Zhigang H, Keqin L. Virtual machine placement algorithm for both energy－awareness and SLA violation reduction in cloud data centers [J]. Scientific Programming, 2016, 2016: 1－11.

[5] Farahnakian F, Ashraf A, Pahikkala T, et al. Using ant colony system to consolidate VMs for green cloud computing [J]. IEEE Transactions on Services Computing, 2015, 8 (2): 187－198.

[6] Wang J V, Cheng C T, Tse C K. A power and thermal－aware virtual machine allocation mechanism for Cloud data centers [A]. IEEE International Conference on Communication Workshop [C]. IEEE, 2015.

[7] Vasudevan M, Tian Y C, Tang M L, et al. Energy－efficient application assignment in profile－based data center management through a repairing genetic algorithm [J]. Applied Soft Computing, 2018, 67 (1): 1－10.

(下转第 255 页)