

一种新的物理主机资源利用阈值边界管理策略

宋宇翔¹, 徐胜超²

(1. 广东培正学院数据科学与计算机学院, 广州, 510830; 2. 北部湾大学电子与信息工程学院, 钦州, 535011)

摘要: 提出了一种新的物理主机资源利用阈值边界管理策略 (Physical host resource utilization thresholds management strategy, RUT-MS)。RUT-MS把云数据中心的虚拟机迁移过程进一步划分为超负载主机检测、虚拟机选择、虚拟机放置第1阶段、低负载主机检测和虚拟机放置第2阶段。使用一种迭代权重线性回归方法来预测物理资源的阈值上限,避免超负载的物理主机数量的增加;采用最小能量消耗策略完成虚拟机选择过程;使用多维物理资源的均方根来确定其资源使用阈值下限,减少低负载物理主机数量。实验结果表明: RUT-MS物理资源利用阈值边界管理策略使云数据中心的能量消耗和虚拟机迁移次数明显减少,SLA(Service level agreement)违规率和SLA及能量消耗联合指标只有少量的增加。

关键词: 虚拟机选择;虚拟机放置;线性回归;云数据中心;低能量消耗

中图分类号: TP393.4 **文献标志码:** A

A New Physical Host Resource Utilization Thresholds Management Strategy

SONG Yuxiang¹, XU Shengchao²

(1. School of Data and Computer Science, Guangdong Peizheng College, Guangzhou, 510830, China; 2. School of Electronic and Information Engineering, Beibu Gulf University, Qinzhou, 535011, China)

Abstract: A new physical resource utilization thresholds management strategy called RUT-MS for cloud data centers is proposed in this paper. In RUT-MS, the virtual machine migration process is divided into five steps: Overloaded host status detection, virtual machine selection, the first virtual machine placement, under-loaded host status detection and the second virtual machine placement phases. In overloading host detection, RUT-MS uses an iterative weighted linear regression method to determine two utilization thresholds so that the performance degradation is avoided. Maximum power reduction policy (MPR) is adopted in virtual machine selection phase. A vector magnitude squared of multiple dimension resources is used to decide the low resource utilization thresholds and switch them to a power saving mode. Finally, RUT-MS is evaluated using CloudSim with real-world workload data. Simulation results show that RUT-MS can reduce the energy cost incurred on the system due to migration. The reduced number of virtual machine migrations and improved energy-efficiency are also obtained in our test. Moreover, the service level agreement (SLA) violation and SLA for energy united metrics are also good.

Key words: virtual machine selection; virtual machine placement; linear regression; cloud data centers; low energy consumption

引言

虚拟机迁移是近年来大规模节能绿色云数据中心构造的关键技术^[1-3],虚拟机迁移过程中不可避免地涉及到物理主机的资源阈值边界^[4],目前 Cloudsim 工具包中提到的几种物理资源阈值管理办法往往局限在物理主机状态检测阶段;已经有很多算法对物理资源的利用率进行优化^[5],比如温度感知算法^[6-7]、稳定匹配算法^[8]、虚拟机关联性算法^[9]以及贪心算法^[10]等,但它们局限在虚拟机选择或放置阶段,缺乏主动性、自适应性。物理主机资源阈值边界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异常的物理主机,接着进行虚拟机迁移的后续步骤。已有的物理主机资源阈值管理策略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随着时间和应用程序的访问而不断动态变化的,这对物理主机资源阈值管理提出了新的要求。本文提出了一种新的物理主机资源利用阈值边界管理策略(Physical resource host utilization thresholds management strategy, RUT-MS)。该策略是一种自适应的、动态的物理资源阈值边界管理方法,它采用时间序列和迭代权重线性回归方法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,同时考虑的物理资源包括处理器、内存、网络带宽等多个维度。RUT-MS 在整个虚拟机迁移过程中都有运用到,通过对超负载物理主机和低负载物理主机的物理资源利用效率阈值进行反复判断,使整个云数据中心的物理资源利用效率进一步提高。

RUT-MS 物理资源利用阈值管理策略通过 Cloudsim 来实现和仿真,结果表明在 RUT-MS 技术之上形成的虚拟机迁移策略比其他常见的虚拟机迁移策略可以更好地节省云数据中心的能量消耗,节约了云服务提供商的企业成本,可以为其他企业构造节能大数据中心作为参考。

1 物理资源阈值管理策略

云数据中心的物理资源阈值管理最主要目标是提高资源利用效率,在空闲的情况下应该关闭物理主机,在超负载的情况下应该迁移其上运行的虚拟机。

文献[6]采用一种物理主机处理器温度阈值边界迁移方法,建立了物理主机的处理器温度阈值管理模型,实验结果验证了该阈值管理策略各个性能指标上的优点。它的不足是把物理资源阈值管理维度限制在处理器利用率的单一因素^[5],对其他的磁盘空间与内存等阈值管理没有考虑。

文献[11]提出了云数据中心考虑资源使用阈值边界的物理资源阈值管理办法,在工作负载高于或者低于设置的阈值上界与阈值下界时,完成虚拟机迁移,文中指出它比 Cloudsim 中已有的策略等性能都要优秀,它是一种固定静态的阈值管理办法,但是随着资源的不断变化,需要提出动态的阈值管理办法来动态提高资源利用效率。

文献[12]提出一个自适应的物理主机资源三阈值管理办法,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4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资源阈值管理办法可以减少云数据中心的能量消耗,但它没有考虑虚拟机放置阶段的阈值管理,对虚拟机迁移的整个过程而言,物理资源利用效率的提高不明显。

文献[13]采用一个局部代理来完成物理资源阈值管理,把物理主机划分为超负载、低负载、正常状态和预高负载4个状态,采用 LiRCUP 方法^[14]来预测超负载的物理主机,避免 SLA(Service level agreement)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,它考虑的资源维度也只限制在处理器和内存等因素,需要继续扩展到网络带宽阈值管理等多个维度的因素。

近年来也有大量的采用新型算法进行优化的物理资源阈值管理的文献,例如基于最邻近算法资源阈值预测管理^[5]、贪心算法^[10]以及虚拟机关联性^[9]等。这些文献在研究思路、测试指标等方面大部分都参考了Cloudsim项目,它们发展到有内存阈值管理、网络带宽阈值管理和网络接口阈值管理,在节省能量消耗方面都有很大的提高,对企业节能云数据中心研究都有很好的参考价值。

2 RUT-MS工作背景与相关术语

2.1 RUT-MS软件支撑环境

RUT-MS物理资源阈值管理策略依托于Cloudsim工具包,Cloudsim中涉及到虚拟机迁移的部分主要由3个模块组成:全局代理(Global broker)、本地代理(Local broker)、虚拟机管理器(Virtual machine manager),如图1所示。

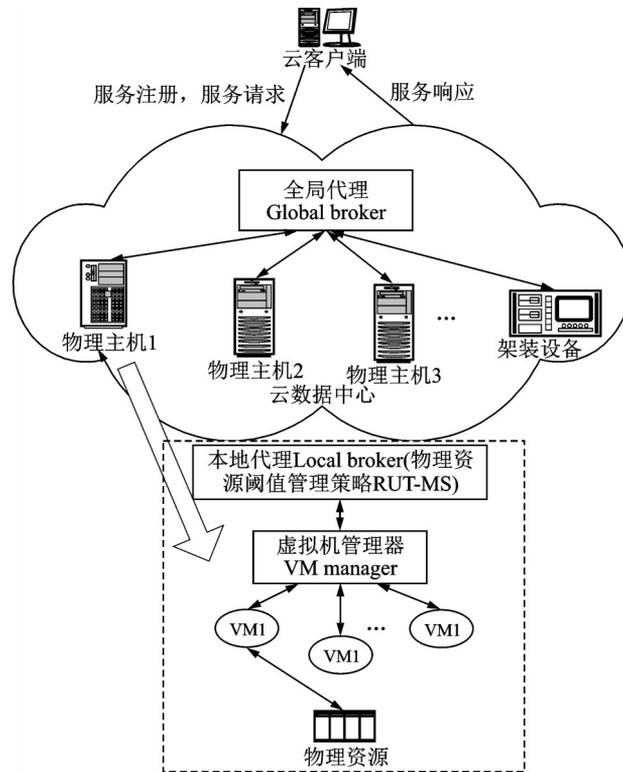


图1 RUT-MS物理资源阈值管理的工作场景

Fig.1 Working mechanism of RUT-MS

由图1可见,每个物理主机上都运行有一个本地代理,RUT-MS物理资源阈值管理策略的实现主要在此模块中完成。这个工作场景在文献[4]中有描述。

2.2 RUT-MS资源阈值管理的工作流程

考虑到RUT-MS既依赖于Cloudsim工具包的软件环境,也有自己的特点,为了管理好整个虚拟机迁移过程,RUT-MS把Cloudsim中虚拟机迁移的步骤由3个扩展到5个^[4]。RUT-MS把虚拟机迁移过程进一步划分为超负载主机检测、虚拟机选择、虚拟机放置第1阶段、低负载主机检测和虚拟机放置第2阶段,工作流程具体包括下面5个步骤,这样处理的好处是把物理资源的阈值边界可能出现异常的情况

都考虑到,同时为虚拟机选择和放置过程做参考,从而间接地进一步提高资源使用效率。图2给出了RUT-MS物理资源阈值管理策略的工作流程。

步骤1 基于RUT-MS,周期性的检测云数据中心的超负载物理主机,形成超负载物理主机列表。

步骤2 针对超负载物理主机列表,基于已有的最小能量消耗选择算法完成虚拟机选择,形成候选迁移虚拟机列表 selectedVMList。

步骤3 针对超负载物理主机中的候选迁移的大量虚拟机,基于RUT-MS物理资源阈值管理策略完成虚拟机的放置第1阶段。

步骤4 基于RUT-MS,周期性的检测云数据中心的低负载物理主机,形成低负载物理主机列表。

步骤5 低负载物理主机中的候选迁移的大量虚拟机,基于RUT-MS物理资源阈值管理策略完成虚拟机的放置第2阶段。

重复上述步骤1~5,设置1个周期(通常是1周),达到该时间段就结束。

本文的RUT-MS在第1阶段物理主机状态检测过程中,根据下述的Algorithm 1和Algorithm 2进行阈值管理,在第2阶段基于Algorithm 3和Algorithm 4完成虚拟机选择,而在第3阶段虚拟机放置中,RUT-MS继续提出新型的物理阈值管理办法,通过Algorithm 5~7来优化虚拟机放置。

3 RUT-MS算法

3.1 RUT-MS管理策略

RUT-MS物理资源阈值管理策略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)和低负载状态(Under-loaded)。这样其对应的物理资源的阈值边界为:阈值上边界Upper-thresholds、资源当前阈值Util-prediction和阈值下边界Lower-thresholds。RUT-MS在每个阶段都要明确Upper-thresholds,Util-prediction,Lower-thresholds三个变量的情况,这些物理资源阈值都是动态调整的,由物理主机的具体情况所决定,而不是设置为固定值。

当RUT-MS策略启动后,周期性地检测物理主机的处理器资源、内存资源、网络带宽资源的使用情况,并尽量使其在合适的阈值范围内。如果有某个物理主机的资源使用情况不在合理的范围内,它将被列入候选迁移物理主机列表或者关闭进入睡眠状态。

3.2 超负载物理主机检测阶段

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^[15]。RUT-MS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。

$$Y = \beta_0 + \beta_1 X \tag{1}$$

式中:Y为受依赖的变量;X为独立的变量; β_0 和 β_1 为回归系数,它们来自于最小二乘法技术^[16],即

$$\hat{\beta}_0 = \bar{Y} - \hat{\beta}_1 \bar{X} \tag{2}$$

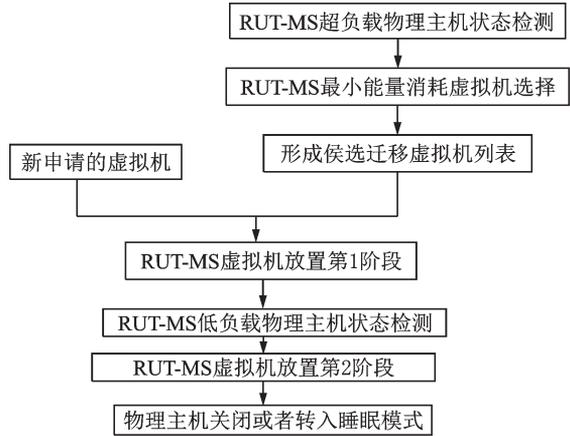


图2 RUT-MS物理资源阈值管理策略工作流程

Fig.2 The working flow of RUT-MS

$$\hat{\beta}_1 = \frac{\sum_{i=1}^n (X_i - \bar{X})(Y_i - \bar{Y})}{\sum_{i=1}^n (X_i - \bar{X})^2} \quad (3)$$

式中: \bar{X} 和 \bar{Y} 为变量 X 和 Y 的均值; $\hat{\beta}_0$ 和 $\hat{\beta}_1$ 为变量 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, 即

$$T(u) = \begin{cases} (1 - |u|^3) & |u| < 1 \\ 0 & |u| > 1 \end{cases} \quad (4)$$

基于式(4), 相邻区域权重的定义为

$$w_i(x) = T\left(\frac{x_n - x_i}{x_n - x_1}\right) = \left(1 - \left(\frac{x_n - x_i}{x_n - x_1}\right)^3\right)^3 \quad (5)$$

式中 x_i 和 x_n 为变量 x 最近的第 i 个观察值。RUT-MS 在这个阶段采用 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, 回归的函数定义为

$$\hat{y}_1 = \beta_0 + \beta_1 x_n, \hat{y}_2 = \beta_0 + \beta_1 \hat{y}_1, \dots, \hat{y}_k = \beta_0 + \beta_1 \hat{y}_{k-1} \quad (6)$$

RUT-MS 策略在超负载检测阶段定义了 2 个阈值边界, Upper-thresholds 和 Util-prediction。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的 100%, 那么该物理主机将被标识为超负载物理主机(Over-loaded), 此时物理主机的资源使用边界为 Upper-thresholds。如果 RUT-MS 策略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 的资源使用边界即为 Util-prediction, 在这个情况下物理主机不会容纳新的虚拟机。Algorithm 1 和 Algorithm 2 给出了 RUT-MS 算法在第 1 阶段的伪代码。

Algorithm 1 RUT-MS algorithm first phase

Input: host utilization

Output: utilPrediction

- (1) for $i=1$ to n do
- (2) $x_i \leftarrow i$; $y_i \leftarrow \text{utilHistory}(i)$;
- (3) $w_i \leftarrow$ calculate using Eq.(5);
- (4) $x_i \leftarrow x_i \times w_i$; $y_i \leftarrow y_i \times w_i$;
- (5) end for
- (6) calculate β_0 , using Eq.(2);
- (7) calculate β_1 , using Eq.(3);
- (8) $\text{utilPrediction} = \beta_0 + \beta_1 \times \text{currentUtil}(h)$; //线性回归预测资源的使用阈值情况
- (9) $\text{upperThreshold} = \text{utilPrediction}$;
- (10) update x , y and w ;
- (11) update β_0 and β_1 ;
- (12) for $i=2$ to k do
- (13) $\text{KPredictionUtil}(i) = \beta_0 + \beta_1 \times \text{utilPrediction}$; //通过 k 次迭代预测可能的阈值边界
- (14) $\text{utilPrediction} = \text{KPredictionUtil}(i)$;
- (15) end for

(16) return utilPrediction;

将算法 Algorithm 1 的输出作为算法 Algorithm 2 的输入,就可以完成超负载物理主机的检测。

Algorithm 2 Overloaded host detection algorithm

Input:host

Output: overloadedList

```
(1) UTC ← CPU.upperThreshold;
(2) PUC ← CPU.utilprediction;
(3) UTM ← Memory.upperThreshold;
(4) PUM ← Memory.utilprediction;
(5) UTB ← BW.upperThreshold;
(6) PUB ← BW.utilprediction;
(7) if ((PUC or PUM or PUB) ≥ 1) then //低负载物理主机判断
(8) underPressureList ← host;
(9) Host will not accept new VM;
(10) else
(11) if ((UTC or UTM or UTB) ≥ 1) then //高负载物理主机判断
(12) overloadedList ← host;
(13) end if
(14) end if
(15) return overloadedList
```

3.3 虚拟机选择阶段

RUT-MS 物理资源阈值管理运用在虚拟机选择阶段,采用的是最小能量消耗策略(Maximum power reduction policy, MPR)。最小能量消耗策略 MPR 目的是选择一个虚拟机之后,尽可能最大程度地降低物理主机的能量消耗,相反地,如果选择其上运行的其他虚拟机来迁移肯定达不到这个效果。假设物理主机 i 上有 j 个虚拟机的集合 VM_j , MPR 策略试图发现式(7)中的虚拟机 V 并选择之。

$$V = \begin{cases} L | L \in VM_j, u_i - \sum_{v \in L} u(v) < T_{up}, |L| \rightarrow \min, P | u(v) | \rightarrow \max & u_i > T_{up} \\ \Phi & \text{其他} \end{cases} \quad (7)$$

式中: u_i 表示物理主机 i 的资源利用率; T_{up} 物理资源的上限阈值, $u(v)$ 表示物理主机分配给虚拟机 v 的 CPU 可用 CPU 利用率; $P | u(v) |$ 表示物理主机 i 上面运行的虚拟机 v 所产生的能量消耗。MPR 策略的算法伪代码如下。

Algorithm 3 Maximum power reduction

```
(1) for each VM in hostVMlist do
(2) selectedVM=NULL;
(3) maxPower = MIN; //定义最大能量消耗变量
(4) power=power(host, VM);
(5) if power > maxpower then
(6) selectedVM = VM;
(7) selectedVMlist = selectedVM;
```

- (8) maxpower=power;
- (9) end if
- (10) end for

虚拟机选择策略整个过程的伪代码见 Algorithm 4。从代码中可以看出,RUT-MS在这个阶段主要是对 Upper-Thresholds 资源阈值的判断。

Algorithm 4 VM selection algorithm

Input: overloadedList, hostVMlist

Output: selectedVMList

- (1) for each host in overloadedList do
- (2) for each VM in hostVMlist do
- (3) selectedVM=NULL;
- (4) Maximum power reduction function;//计算虚拟机的能量消耗
- (5) selectedVMlist= selectedVM;
- (6) end for
- (7) currentCPU_{util} = currentCPU_{util} - selectedVMCPU_{util}
- (8) currentRAM_{util} = currentRAM_{util} - selectedVMRAM_{util}
- (9) currentBW_{util} = currentBW_{util} - selectedVMBW_{util}
- (10) if((currentCPU_{util} < upperThreshold) &&
(currentRAM_{util} < upperThreshold) &&
(currentBW_{util} < upperThreshold)) then
- (11) break; else
- (12) hostVMlist = hostVMlist -selectedVM;
- (13) go to line 2;
- (14) end if
- (15) end for
- (16) return selectedVMList

3.4 虚拟机放置第 1 阶段

在虚拟机放置阶段,RUT-MS有两个步骤要完成:(1)放置从超负载主机(Overloadedlist)列表中候选迁移的虚拟机;(2)放置从低负载主机(Underloadedlist)列表中候选迁移的虚拟机。

RUT-MS遍历正常状态的物理主机列表(Normallist),找到足够CPU、内存及网络带宽的主机安排给新虚拟机,而且在虚拟机放置之后不会使其负载过大。每个主机设计一个Mark值,即有

$$\text{Mark} = \frac{\text{RAM}(\text{VM})}{\text{availableBW}(\text{host})} \quad (8)$$

最后通过判断,具有最小Mark值的正常主机就容纳新的虚拟机。如果正常状态的物理主机不具备容纳新虚拟机的条件,那么RUT-MS将遍历低负载状态的物理主机(Underloadedlist),从新计算物理主机的Mark值,如果这些主机列表里也没有满足容纳新虚拟机的条件,则需要重新开启一个新的物理主机,过程如算法 Algorithm 5 所示。

Algorithm 5 VM Placement (First phase)

Input: hostList, selectedVMlist

Output: allocation of VMs

```

(1) selectedVMlist.sortDecreasing( );
(2) for each h in hostList do
(3) if (lowrthreshold < currentUtil < util-Prediction) then normalHostList ← host;
(4) else if (currentUtil < lowrthreshold)
(5) underloadedhostList ← host;
(6) end if
(7) end for
(8) for each VM in selectedVMlist do
(9) minMark ← MAX; selectedHost ← null;
(10) for each h in normalHost do
(11) estimate utilAfterPlacement;
(12) if (utilAfterPlacement < upperthreshold) then
(13) estimate Mark by Eq.(8);
(14) if (Mark < minMark) then //最小 Mark 值的正常主机就容纳新的虚拟机
(15) selectedHost ← host; selectedHostlist ← selectedHost; minMark ← Mark;
(16) end if
(17) end for

```

3.5 低负载物理主机检测阶段

低负载物理主机检测主要是要确定自适应低阈值 Lower-thredholds。RUT-MS 通过物理资源比较低的四分位值来完成判断。将该四分位值指定为资源使用阈值下边界 Lower-thresholds, 所以 $T_{low} = u_{((n+1)/4)}$, 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, $(n+1)/4$ 表示 u 的下标。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), 根据向量的平方根式(9), 式(10)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。

$$Util = \sqrt{a^2 + b^2 + c^2} \quad (9)$$

$$Util = \sqrt{(Util(CPU))^2 + (Util(RAM))^2 + (Util(BW))^2} \quad (10)$$

在低负载物理主机检测中, RUT-MS 算法维护着一个低负载物理主机列表, 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。具体的算法描述如 Algorithm 6 所示。

Algorithm 6 Under loading host detection

Input: hostList, hostVMlist

Output: VMmigrationList

```

(1) for each h in hostList do
(2) if ((h.utilCPU) < Tlow(CPU)) && ((h.utilRAM) < Tlow(RAM)) && ((h.utilBW) < Tlow(BW)) then
(3) underloadingList ← h; //处于低负载状态
(4) end for
(5) for each h in underloadingList do

```

$$(6) \text{utilCPU} = \left(\frac{\text{allocatedMips}}{\text{TotalMips}} \right)^2$$

$$\text{utilRAM} = \left(\frac{\text{allocatedRam}}{\text{TotalRam}} \right)^2$$

$$\text{utilBW} = \left(\frac{\text{allocatedBw}}{\text{TotalBw}} \right)^2$$

(7) Util = $\sqrt{\text{utilCPU} + \text{utilRAM} + \text{utilBW}}$ //计算均方根

(8) underloadingList.sortIncreasingUtil();

(9) end for

(10) for each h in underloadingList do

(11) for each VM in hostVMlist() do

(12) for each host in hostList do

(13) if(host \notin underPressureList) then

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

(15) VMmigrationList \leftarrow h.VM;

(16) hVMlist \leftarrow hVMlist- h.VM;

(17) end if end if

(18) end for end for end for

3.6 虚拟机放置第2阶段

检测低负载物理主机列表的时候,低负载物理主机被进一步划分为接受虚拟机列表和非接受虚拟机列表物理主机。RUT-MS 试图把那些非接受虚拟机列表的物理主机切换到睡眠或者关闭状态,减少了虚拟机迁移次数。

首先遍历接受虚拟机列表,如果没有满足条件的物理主机,继续遍历非接受虚拟机列表,如果还是没有满足容纳新虚拟机的物理主机,系统必须重新启动一台物理主机,过程如算法 Algorithm 7 所示。

Algorithm 7 VM Placement (Second phase)

Input: hostList, selectedVMlist

Output: allocation of VMs

(1) if (host admitted any VM in the first phase) then

(2) receivedVMlist \leftarrow host; else otherHostslist \leftarrow host;

(3) end if end if end for

(4) for each VM in selectedVMlist do

(5) minMark \leftarrow MAX; selectedHost \leftarrow null;

(6) for each host in normalHost do

(7) estimate Utilafterplacement;

(8) if (Utilafterplacement $<$ upperthreshold) then

(9) estimate Mark by Eq.(8);

(10) if (Mark $<$ minMark) then

(11) selectedHost \leftarrow host;

(12) selectedHostlist \leftarrow selectedHost;

(13) minMark \leftarrow Mark;

- (14) end if end if
- (15) end for end for
- (16) return selectedhostlist ;

4 RUT-MS 仿真实验与性能分析

4.1 仿真环境配置

因为 RUT-MS 物理资源阈值的管理策略运用在虚拟机迁移过程中,所以进行 RUT-MS 实验分析必须构造 Cloudsim3.0 云数据中心的虚拟机迁移场景,同时依据图 1 中的运行场景,在 Cloudsim 中实现基于 Java 语言的局部代理、全局代理和虚拟机管理器。

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^[17]。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。

表 1 云数据中心物理服务器配置
Table 1 Physical host configuration of cloud data center

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 ⁻¹)
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

在线性回归中设置观察近 1 h 的物理主机资源使用率数据,这种设置意味着循环迭代的次数为 12 次及虚拟机的迁移每 5 min 运行 1 次,一共运行 24 h,每次统计 1 天内的能量消耗,在 1 周内重复运行 5 次,1 周内每天虚拟机请求的个数见表 2,不同虚拟机的类型与粒度配置见表 3。

表 2 RUT-MS 物理资源阈值的管理策略实验环境
Table 2 Experimental environments of RUT-MS

日期	虚拟机个数	物理主机个数	运行时间/h
周一	1 052	800	24
周二	898	800	24
周三	1 061	800	24
周四	1 516	800	24
周五	1 078	800	24

表 3 虚拟机类型配置
Table 3 Experimental environments on virtual machine type of RUT-MS

虚拟机类型	虚拟机需求/MIPS	虚拟机所需内存大小/MB	带宽需求/(MB·s ⁻¹)
类型 1	500	613	100
类型 2	1 000	1 740	100
类型 3	2 500	3 750	100
类型 4	2 500	850	100

4.2 评测标准与比较对象

由于RUT-MS物理资源阈值的管理策略评价的主要指标有4个:(1)云数据中心的总体能量消耗;(2)虚拟机迁移次数;(3)平均SLA违规分析;(4)能量与SLA违规的联合指标。

根据虚拟机迁移3个阶段的步骤,Cloudsim中比较好的办法是LRR(Local regression robust)策略为物理主机状态检测,结合MMT策略完成虚拟机选择,再结合递减装箱方法(Best-fit-decreasing CPP, BFD)完成虚拟机放置,称之为LRR-MMT-BFD策略,它应该作为首要比较对象。LRR方法是一种自适应的主机利用效率阈值检测办法,它通过检测一个物理主机的最近 j 个处理器的使用效率值作为评价方法,在本实验中设 $j=10$ 。MMT选择一个在最短时间内能够完成迁移的虚拟机作为候选迁移对象,形成候选迁移虚拟机列表。BFD是递减装箱方法,它按照处理器的使用效率以递减的方式排序,然后依次完成虚拟机放置,它是一种无优化的普通放置方法。

本文还将RUT-MS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阈值检测方法^[14]、ATEA阈值检测方法^[12]以及在虚拟机放置阶段优化的智能算法,例如基于稳定匹配^[8]、基于虚拟机关联性^[9]和基于萤火虫群优化的算法^[18],并且分析这些物理资源阈值管理办法对云数据中心的性能改变情况。综上所述,本实验涉及到的虚拟机迁移模式如表4所示,包括RUT-MS,LRR-MMT-BFD,LiRCUP,ATEA,Stable-Matching,Correlation-Based和GSO-VMM共7个虚拟机迁移模式。

表4 RUT-MS物理资源利用阈值边界管理性能比较对象

Table 4 Different kinds of compared objects of RUT-MS

虚拟机迁移模式	相关说明
RUT-MS	虚拟机迁移的各个阶段都优化
LRR-MMT-BFD ^[4]	Cloudsim项目原始模型
LiRCUP ^[14]	物理主机状态检测阶段优化
ATEA ^[12]	物理主机状态检测阶段优化
Stable-Matching ^[8]	虚拟机选择和放置阶段优化
Correlation-Based ^[9]	虚拟机放置阶段优化
GSO-VMM ^[18]	萤火虫群算法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

Cloudsim模拟器仿真的云数据中心1周之内得到的总体能量消耗如图3所示。从图3可以看出,RUT-MS迁移模型比Cloudsim中的LRR-MMT-BFD迁移策略在总体能量消耗上要节约20%~25%,比Stable-Matching策略、Correlation-Based策略和GSO-VMM策略的能量消耗也要低,虽然周一到周五之间的数据有一定的波动,但是整体趋势是RUT-MS物理资源阈值管理策略性能最优。分析原因是RUT-MS能够每次回归检测到状态异常的超负载物理主机,而且在最后虚拟机放置阶段后,还可以检测到低负载的物理主机,将它们转入睡眠状态。

4.3.2 虚拟机迁移次数

图4显示了云数据中心的各个迁移模型总体的虚拟机迁移次数。大部分研究都认为虚拟机次数越

少,表示云数据中心性能比较好^[5]。从图中可以看出在1周的5天之内 RUT-MS 的虚拟机迁移次数都低于 LRR-MMT-BFD 策略;而 Stable-Matching 策略、Correlation-Based 迁移策略、GSO-VMM 策略、LiRCUP 策略和 ATEA 策略则处理性能均为中等水平。分析原因是 LRR-MMT-BFD 增加了超负载或低负载的物理主机的数量,这样很容易出现虚拟机迁移的现象,结论是增加虚拟机迁移也间接地增加了物理主机的能量消耗,因为虚拟机迁移过程也要消耗物理主机的 CPU 资源,而 RUT-MS 管理策略则与 LRR-MMT-BFD 正好相反,RUT-MS 都是基于对物理主机资源使用效率的估计策略,它不会增加低负载主机数量。另外 Stable-Matching 策略、Correlation-Based 策略以及 GSO-VMM 策略的优化主要在虚拟机放置阶段,基本没有减少虚拟机迁移次数,要想减少虚拟机迁移次数,必须在物理主机状态检测、虚拟机选择阶段完成优化。

4.3.3 SLA 违规率分析

从图 5 可以看出,周一到周五 RUT-MS 的 SLA 违规率比 LRR-MMT-BFD 迁移策略要低,原因是 RUT-MS 的虚拟机迁移次数比较少,它可以避免物理主机出现 100% 的 CPU 利用率的机会,SLA 违规比率自然会降低。Stable-Matching 策略、Correlation-Based 策略和 GSO-VMM 策略在虚拟机放置阶段都采用另外的优化算法,这样整个云数据中心的 SLA 违规率自然会比 LRR-MMT-BFD 迁移策略要低。LiRCUP 策略和 ATEA 策略则处于性能中等水平,比 Cloudsim 的原始迁移策略要 SLA 违规率低。

4.3.4 能量与 SLA 违规的联合指标 ESV^[4]

ESV 指标是体现云服务提供商的高服务质量、低 SLA 违规比率和总体能量消耗平衡的指标,即

$$ESV = E \times SLAV$$

式中 E 为能量消耗指标。从图 6 中的结果可以看到,RUT-MS 迁移策略的 ESV 也要低于 LRR-MMT-BFD 迁移策略。结果表明在各个阶段过程中智能算法优化对云数据中心的物理资源利用效率提高的重要性。LiRCUP 策略、ATEA 策略、Stable-matching 策略和 Correlation-Based 策略是 2016 年和 2017 年提出的虚拟机放置改进算法,自然比 2012 年的 LRR-MMT-BFD 迁移策略性能优秀。RUT-MS 最重要的目的是降低能量消耗,而 GSO-VMM 策略主要面向降低 SLA 违规率,所以它们的 ESV 联合指标在某些条件下也优于 RUT-MS 管理策略。

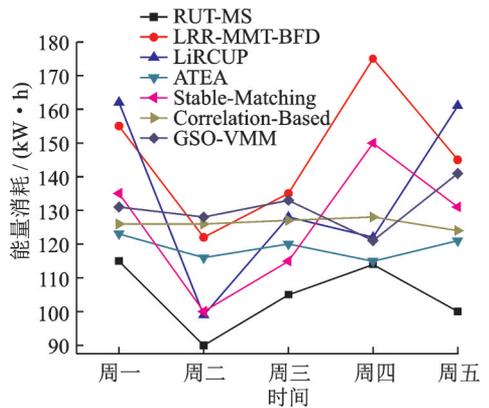


图 3 各类虚拟机迁移策略的总能量消耗比较
Fig.3 Experimental results on total energy consumption in different kinds of virtual machine migration model

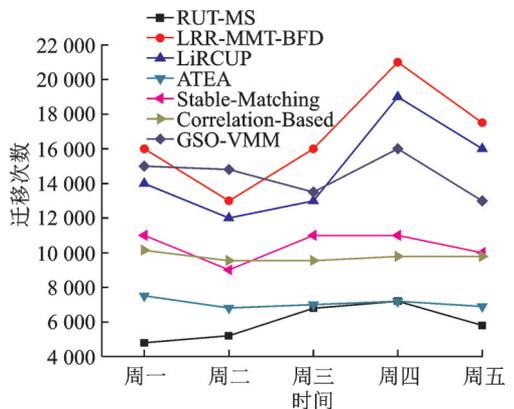


图 4 各类虚拟机迁移策略的虚拟机迁移次数
Fig.4 Experimental results on number of migration in different kinds of virtual machine migration model

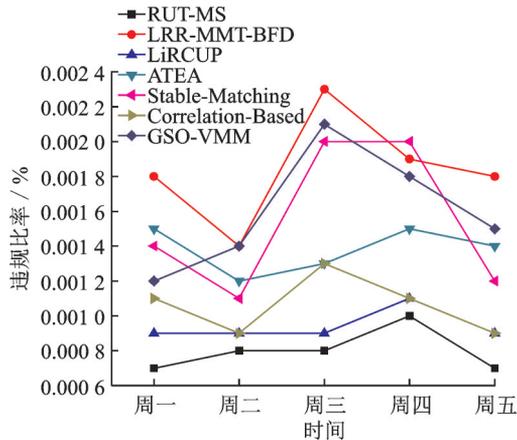


图 5 各类虚拟机迁移策略的SLA违规率比较
Fig. 5 Experimental results on SLA violation in different kinds of virtual machine migration mode

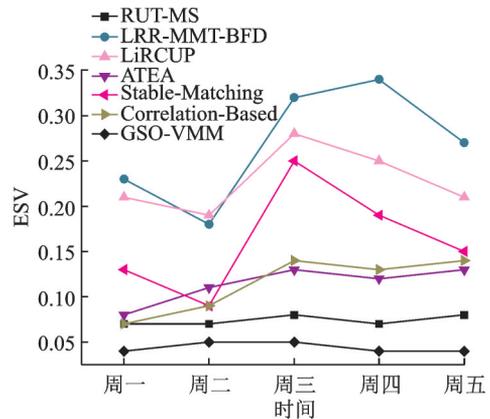


图 6 各类虚拟机迁移的SLA与能量消耗联合指标ESV
Fig.6 Experimental results on ESV in different kinds of virtual machine migration model

5 结束语

本文提出了一种新的物理主机资源利用阈值边界管理策略RUT-MS,动态设置物理资源的边界Upper-thresholds值、Util-prediction值和Lower-thresholds值。RUT-MS的阈值边界具有自适应性和动态性特点,并且在虚拟机迁移过程5个阶段都有算法进行阈值边界的优化。实验表明RUT-MS比常见的虚拟机迁移策略在总体能量消耗、虚拟机迁移次数上有明显优势,SLA违规比率、ESV等指标只有少量的增加,RUT-MS物理资源利用阈值边界管理策略的每个阶段还可以进一步细分与优化,可以继续提高系统性能。

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] LIU Xiaofang, ZHAN Zhihui. An energy efficient ant colony system for virtual machine placement in cloud computing[J]. *IEEE Transactions on Evolutionary Computation*, 2018, 22(1): 113-128.
- [3] JIN X, ZHANG F, WANG L, et al. Joint optimization of operational cost and performance interference in cloud data centers [J]. *IEEE Transactions on Cloud Computing*, 2017, 99(1): 697-711.
- [4] 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 & Computation Practice & Experience*, 2012, 24(13): 1397-1420.
- [5] FARAHNAKIAN F, PAHIKKALA T, LILJEBERG P, et al. Energy aware consolidation algorithm based on k-nearest neighbor regression for cloud data centers[C]//*Proceedings of 6th IEEE/ACM International Conference on Utility and Cloud Computing(UCC)*. [S.l.]: ACM, 2013.
- [6] MHEDHEB Y, JRAD F, TAO J, et al. Load and thermal-aware VM scheduling on the cloud[C]//*Proceedings of ICA3PP 2013-13th International Conference on Algorithms and Architectures for Parallel Processing*. New York: Springer-Verlag, 2013.
- [7] WANG J V, CHENG C T, TSE C K. A power and thermal-aware virtual machine allocation mechanism for Cloud data

- centers[C]//Proceedings of IEEE International Conference on Communication Workshop. [S.l.]: IEEE, 2015.
- [8] WANG J V, FOK K Y, CHENG C T, et al. A stable matching-based virtual machine allocation mechanism for cloud data centers[C]//Proceedings of 2016 IEEE World Congress on Services (Services). San Francisco, CA: IEEE, 2016: 103-106.
- [9] WANG J V, CHENG C T, TSE C K. Effects of correlation-based VM allocation criteria to cloud data centers[C]//Proceedings of International Conference on Cyber-enabled Distributed Computing & Knowledge Discovery. [S.l.]: IEEE, 2017.
- [10] RASOUL B. Green cloud computing: Greedy algorithms for virtual machines migration and consolidation to optimize energy consumption in a data center[J]. International Journal of Digital Application & Contemporary research, 2014, 2(9): 1-9.
- [11] KHUSHBU M, RICHA S. Energy conscious dynamic provisioning of virtual machines using adaptive migration thresholds in cloud data center[J]. International Journal of Computer Science and Mobile Computing, 2013, 2(3): 74-82.
- [12] 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(1): 1-11.
- [13] KHOSHKHOLGHI M A, DERAHMAN M N, ABDULLAH A, et al. Energy-efficient algorithms for dynamic virtual machine consolidation in cloud data centers[J]. IEEE Access, 2017: 10709-10722.
- [14] FARAHNAKIAN F, LILJEBERG P, PLOSILA J. LiRCUP: Linear regression based CPU usage prediction algorithm for live migration of virtual machines in data centers[C]//Proceedings of the 2013 39th Euromicro Conference on Software Engineering and Advanced Applications. [S.l.]: IEEE Computer Society, 2013.
- [15] GUERARD JR J B. Introduction to financial forecasting in investment analysis[M]. New York: Springer Science and Business Media, 2013.
- [16] WEISBERG S. Applied linear regression[M],[S.l.]: John Wiley & Sons, 2005.
- [17] KYOUNGSOO P, VIVEK S P. CoMon: A mostly-scalable monitoring system for PlanetLab[J]. ACM Sigops Operating Systems Review, 2006, 40(1): 65-74
- [18] ALBOANEEN D A, TIANFIELD H, ZHANG Y. Glowworm swarm optimisation algorithm for virtual machine placement in cloud computing[C]//Proceedings of 2016 Intl IEEE Conferences on Ubiquitous Intelligence & Computing, Advanced and Trusted Computing, Scalable Computing and Communications, Cloud and Big Data Computing, Internet of People, and Smart World Congress. Toulouse, France: IEEE, 2016.

作者简介:

宋宇翔(1971-),男,副教授,研究方向:计算机网络和云计算,E-mail: zhangyanmei_2015@126.com。



徐胜超(1980-),通信作者,男,讲师,研究方向:计算机网络和云计算。

(编辑:刘彦东)