

基于改进虚拟机整合算法的虚拟资源管理工具

赵长名¹, 刘 健², 李云继³

(1. 电子科技大学通信与信息工程学院 成都 611731;

2. 北京科技大学通信与信息工程学院 北京 海淀区 100083;

3. 贵州理工学院电气工程学院 贵阳 550003)

【摘要】提出了一种基于分段迭代相关性整合(SICC)的虚拟机整合与放置策略,并将它作为云资源管理工具的核心结构。SICC算法整合了时间序列分析、线性相关性分析和传统的FFD算法,并基于虚拟机的最小资源利用率建立了一套新的虚拟机动态资源整合理论。数值仿真结果表明,在虚拟机整合过程中,新的基于SICC的架构在使用不同的初始动态条件时,以虚拟机为粒度的物理资源利用率性能提升3%~20%;在以服务器为粒度的物理资源利用率性能提升超过5%。

关键词 动态资源整合; FFD算法; 线性相关性分析; 虚拟机

中图分类号 TN915.01 文献标志码 A doi:10.3969/j.issn.1001-0548.2016.02.007

Virtualization Resource Management Tool Based on Improved Virtual Machine Consolidation Algorithm

ZHAO Chang-ming¹, LIU Jian², and LI Yun-ji³

(1. School of Communication and Information Engineering, University of Electronic Science and Technology of China Chengdu 611731;

2. School of Communication and Information Engineering, University of Science and Technology Beijing Haidian Beijing 100083;

3. Faculty of Information Engineering, GuiZhou Institute of Technology GuiYang 550003)

Abstract In the paper, we propose a virtual machine (VM) consolidation and placement strategy, named segmentation iteration correlation combination (SICC). The SICC algorithm integrates several algorithms, such as time series analysis, linear correlation analysis and traditional first fit decreasing (FFD). A new dynamic resource consolidation theory is then established based on the virtual machine minimum resource utilization parameter. The simulation results indicate that the novel SICC framework can improve the physical resource utilization by 3% to 20% in the VM granularity and by up 5% in the server granularity.

Key words dynamic resource consolidation; FFD algorithm; linear correlation analysis; virtual machine

虚拟机整合技术(VM consolidation, VMC)通常被用于释放和回收虚拟机中的碎片化虚拟资源。这项技术在概念上类似于虚拟机迁移,但相比之下VMC更强调虚拟机内的预留资源整合。当前的VMC算法主要分为静态整合算法和动态整合算法两类。其中静态算法因为静态参数模型而得名,一些静态模型常常被简化为具有低操作开销和高全局优化性能的bin-packing模型。但在资源利用率处于高动态变化的场景中,一旦资源参数被设定为VM的峰值资源利用率,静态的bin-packing算法就会表现出极差的资源利用率。因此,当任务还需同时要满足高服务等级协议(service-level agreement, SLA)时,静态算

法就会导致VM群内形成大量资源碎片^[1-4]。

与此相反,动态算法支持计算并选定两台当前资源利用率互补最好的VM进行资源整合,从而获得较好的局部资源整合效果,但其在全局资源整合效率上却有较大的缺陷^[5-7]。文献[8]提出了一种精确反映VM资源利用率互补程度的方案。该方案采用了时间序列分析和皮尔逊相关系数公式等回归统计工具,在历史统计数据的基础上对虚拟机未来资源利用率进行精确的预测计算和互补性测试,从而能对VM未来的资源整合提供依据。但这个方案仍然有一个明显的缺陷:任意一个VM在算法的整个流程中最多只有一次与其他虚拟机整合的机会,许多整合后

的VM平均资源利用率仍然较低。基于此,文献[9]提出了一种改进的算法—iterative correlation match algorithm (CMA)。该算法允许一次整合完成后的整合虚拟机(CVM)再次参与整合的相关性计算,而不是直接将CVM从待整合的VM序列中删除。这样一些具有高整合潜力的VM就拥有不止一次整合的机会。但是,该算法仍然有一个明显的缺陷:即该算法是一种贪婪算法,总是搜寻当前具有全局最小相关系数的两台VM进行整合,却忽略了按VM的资源利用率进行全局优化调度。这将会在整合流程的后半部分导致严重的线性规划难题。这个缺陷只依靠ICMA并不能得到解决。这是因为现有的动态整合算法几乎都是直接利用最小线性相关系数启发式算法。这些算法所做的改进只是单纯地尽可能增加算法流程中虚拟机最小线性相关整合的次数,而缺乏对于虚拟机相关整合原理的本质研究。

本文提出了一种基于SICC的新型虚拟资源整合工具。该工具融合了静态和动态两种完全不同的虚拟机整合算法模型,依靠SICC算法将在时域具有高动态资源特性的初始状态虚拟机(original VMs)转变为具有准静态资源特性和较高平均资源利用率的整合虚拟机。接着使用成熟的FFD算法完成这些CVM在数据中心物理服务器中的聚合过程。SICC算法的策略在于:通过一定的算法变化,大幅减小原始VM中过大的资源利用率峰值-均值差值,从而有效地避免在虚拟机中形成难以利用的资源碎片。

1 系统模型

数据中心内进行相关VM整合的先决条件是系统能够准确地预测出虚拟机在未来特定时段内的资源利用率。本文使用自回归滑动平滑模型ARMA对虚拟机的历史资源利用率数据进行处理。但是,这些预测出的VM资源利用率数据并不能直接用于服务器中的VM相关性整合运算。这是由相关整合算法的固有缺陷造成的,相关细节将在后面进行讨论。

本文提出了一种两阶段策略来完成服务器中的虚拟机整合。第一阶段利用SICC算法在初始状态VM群中完成VM粒度的相关性整合,整合后的CVM具有资源平均利用率高和资源利用率峰值-均值差小的特点。在策略的第二阶段引入适用于静态整合的FFD算法将这些CVM重新聚合在服务器中。

SICC算法本身也可以看作一个两阶段的Bin-Packing过程。本文假定所有VM仅具有CPU一种

维度的资源并且包含的资源量也是相等的,称为标准容量VM,其容量为 c 。整体策略和SICC算法的流程描述如图1所示,其中图1a和图1b描述了SICC算法的第一阶段。在SICC的第一阶段,初始状态的VM按照峰值资源利用率(peak resource utilization, PRU)的大小分为小型、中型和大型3个级别,分别对应的条件为: $PRU \leq 25\%$ 、 $25\% < PRU \leq 50\%$ 及 $PRU > 50\%$ 。随后,使用一种改进的ICMA处理这些初始状态的VM,与标准的ICMA算法不同,这种改进的ICMA只允许对相关系数值小于等于 -0.5 的VM进行整合。具体的处理的方法是:先在3个初始的VM内使用改进的ICMA进行处理,并将处理后生成的CVM集中起来使用改进的ICMA算法再处理一次。在图1a中,一个初始状态的VM在第一次整合形成CVM后可以继续进行整合直到其与任意VM和CVM的整合都不能满足相关系数不大于 -0.5 的条件。在迭代完成上述过程后,CVM整体具有足够低的峰值-均值差,如图1b所示。图1c中描述的是SICC第二阶段流程,使用FFD整合了第一阶段预处理的CVM。SICC因为同时使用了静态和动态的整合算法而具备了很高的全局整合增益。因此,这个基于SICC的资源管理方案可以保证数据中心取得比以前的整合策略更高的资源利用效率。

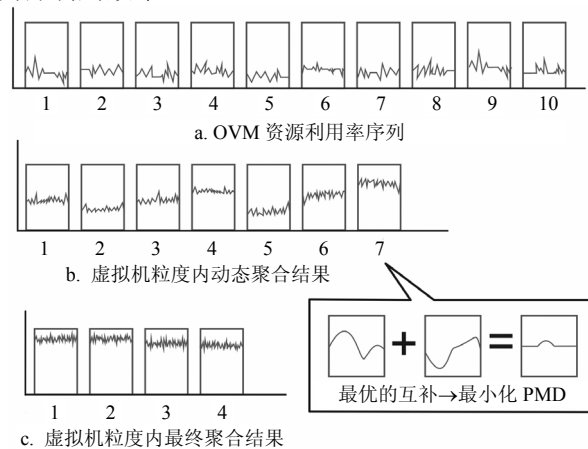


图1 资源管理工具系统模型

2 SICC算法分析

2.1 皮尔逊线性相关系数

采用1 440个数据抽样点模拟一台VM一天的时间序列,并在此基础上构造一个短相关模型对未来资源利用率进行预测。一旦能获取每个VM的未来资源利用率时间序列,就有可能找出当前全局最佳互补的VM整合配对。 $\{x_t^1\}$ 和 $\{x_t^2\}$ 分别代表ARMA模型计算出的两台VM在未来一段时间的资源利用率。皮尔逊相关系数定义为:

$$r = \frac{\sum_{i=1}^{1440} (X_i^1 - \bar{X}^1)(X_i^2 - \bar{X}^2)}{\sqrt{\sum_{i=1}^{1440} (X_i^1 - \bar{X}^1)^2} \sqrt{\sum_{i=1}^{1440} (X_i^2 - \bar{X}^2)^2}} \quad (1)$$

式中, 系数 r 是一个衡量两台VM相关系数的基本指标, 在 $-1 \sim 1$ 之间变动, 最好的互补VM整合配对是对应于具有全局最小线性相关系数的VM整合配对。

2.2 基于可测物理量的相关系数特征研究

ICMA这一类基于最小相关系数的整合算法有一个重要的缺陷: 该系数只能衡量两台或两台以上VM间的资源利用率相关互补关系。因此, 首要目标是找出一个或一组基于单个VM的可测物理量来对相关整合算法的算法效能进行描述。在文献[10]提供的实验数据和结论的基础上, 首先推导出一个结论: 对于初始状态的VM而言, 如其中一个具有更大的资源利用率方差, 则其在相关整合算法中具有更高的整合优先级。

令 $V_{a(b)}$ 代表初始状态虚拟机 $VM_{a(b)}$ 的资源利用率方差, 若 $V_a > V_b$ 且差值较大, 则可以认为:

定理 1 当系统中初始状态的VM数量足够多时, 对于类似于ICMA的基于线性相关性的整合算法而言, VM_a 相对于 VM_b 具有更高的合并优先级。

证明: 定理1等价于一台VM的资源利用率曲线包络的振幅越大, 较振幅较小的VM就有越高的概率在初始VM群中找到合适的VM组成相关整合系数足够小的配对组。建立一个简化的模型演示 VM_a 和 VM_b 的组合情况。 A, B 和 C 代表3种不同的单位步长的特征振幅信号, 分别对应于1, 0和-1。 A 和 C 定义为单位步长中明显大于和小于VM资源利用率平均值的抽样点, 而 B 代表与VM资源利用率平均值差距不大的抽样点。利用率序列中 A 的个数与 C 的个数相等, 但少于 B 的个数。

令信号 B 在 VM_a 和 VM_b 中的概率等于 ρ_0 , 信号 A 和信号 C 出现的概率相等, 则有:

$$\rho_0(A) = \rho_0(B) = \rho_1 = \frac{1}{2}(1 - \rho_0) \quad (2)$$

根据假设有:

$$\rho_0 > \frac{1}{2}(1 - \rho_0) \quad (3)$$

则可以推出:

$$\rho_0 > \frac{1}{3} \quad (4)$$

$$\begin{cases} \min(\text{Var}(VM_1), \text{Var}(VM_2)) > \min(\text{Var}(VM_3), \text{Var}(VM_4)) \\ \text{coeff}(VM_1, VM_2) < \text{coeff}(VM_3, VM_4) < 0 \end{cases} \quad (10)$$

VM_a 和 VM_b 的任意步长中, 存在9种不同配对关系:

负相关关系: $A-C, C-A, B-B$

不相关关系: $A-B, C-B, B-A, B-C$

正相关关系: $A-A, C-C$

则 VM_b 任意步长中共有45种可能配对组合, 记为 L_{45}^b , 以及14种可能的负相关组合, 记为 L_{14}^b 。则低振幅VM的负相关整合条件概率为:

$$P_b = P(L_{14}^b | L_{45}^b) = P\left(\frac{L_{14}^b L_{45}^b}{L_{45}^b}\right) = \frac{\rho_0 \rho_1 (8\rho_1^2 + 6\rho_0^2)}{(\rho_0^2 + 2\rho_0 \rho_1)(6\rho_1^2 + 8\rho_0 \rho_1 + \rho_0^2)} \quad (5)$$

令 $\frac{\rho_1}{\rho_0} = t$, 则式(5)可改写为:

$$P_b = \frac{t(8t^2 + 6)}{(1 + 2t)(6t^2 + 8t + 1)} \quad (6)$$

同理, VM_a 中的对应参数为 L_{27}^a 和 L_{11}^a , 则对应的负相关整合概率为:

$$P_a = P(L_{11}^a | L_{27}^a) = P\left(\frac{L_{11}^a L_{27}^a}{L_{27}^a}\right) = \frac{2t^4 + 4t^3 + 4t + 1}{8t^4 + 8t^3 + 6t^2 + 4t + 1} \quad (7)$$

则有:

$$\frac{P_a}{P_b} = \frac{24t^7 + 92t^6 + 108t^5 + 90t^4 + 104t^3 + 62t^2 + 14t + 1}{64t^7 + 64t^6 + 96t^5 + 80t^4 + 44t^3 + 24t^2 + 6t} \quad (8)$$

即具有连续两个特征振幅的VM有更高的合并优先级。

至此证明了资源利用率方差可以作为一个衡量初始状态VM整合潜力的重要参数, 并且整合潜力还正相关于VM的方差。与此同时还发现初始状态VM可以分为两种基本方差类型: Type I型VM和Type II型VM。其中Type I型VM的方差主要由低频率高振幅的方差构成。而Type II的方差构成形式与Type I正好相反。由此, 在定理1的基础上可推导出定理2。

定理 2 在ICMA算法中, Type I类型的VM比Type II类型的VM更可能成为当前全局最优互补配对的组成部分。

证明: 设:

$$\begin{cases} VM_1, VM_2 \in \text{Type I} \\ VM_3, VM_4 \in \text{Type II} \end{cases} \quad (9)$$

式中, 函数coeff代表相关系数; $\text{Var}(\text{VM}_a)$ 表示 VM_a 的资源利用率方差。 \bar{S}_a 代表 VM_a 的资源利用率抽样序列, 而 $\bar{S}_{(a,b)}$ 代表 $\text{VM}_{(a,b)}$ 的资源利用率抽样序列。则有:

$$\begin{cases} \bar{S}_{(1,2)} = \bar{S}_1 + \bar{S}_2 \\ \bar{S}_{(3,4)} = \bar{S}_3 + \bar{S}_4 \end{cases} \quad (11)$$

$$\begin{cases} \text{Var}(\text{VM}_{(1,2)}) < \min(\text{Var}(\text{VM}_1), \text{Var}(\text{VM}_2)) \\ \text{Var}(\text{VM}_{(3,4)}) < \min(\text{Var}(\text{VM}_3), \text{Var}(\text{VM}_4)) \\ \text{Var}(\text{VM}_{(1,2)}) > \text{Var}(\text{VM}_{(3,4)}) \end{cases} \quad (12)$$

根据定理1, 可以证明 $\text{VM}_{(3,4)}$ 在ICMA中进行整合配对的优先概率比 $\text{VM}_{(1,2)}$ 低。则可知两个Type I VM进行整合配对时有更大的概率成为当前全局最优的互补配对。但是, 由Type II生成的CVM比由Type I生成的CVM拥有更大的方差, 即Type II VM的持续整合潜力优于Type I VM。这证明ICMA算法中总是选择全局最小相关系数对应的虚拟机对进行整合配对并不是一个最优的资源整合策略。

2.3 SICC算法分析

首先定义两个概念, 即相关系数序列(CCS)和峰值利用率柱状分布(PUB)对算法效能进行衡量。CCS

包括了一个VM整合过程中所有按时间序列排列的和整合配对对应的相关系数。PUB则被用来作为衡量每个VM名义资源利用率的基准。本节的首要任务是建立CVM的峰值资源利用率(PRU)模型, 并设计出一种有效的方法测量相关系数对资源利用率的影响。假设第*i*个VM的PRU为 P_i , 并且第*i*个VM的资源利用率时间序列为 \bar{S}_i 。则CVM的PRU可以表示为:

$$\begin{aligned} P_{\text{CVM}} &= \max(\bar{S}_1 + \bar{S}_2) \\ \text{s.t. } \max(P_1, P_2) &\leq P_{\text{CVM}} \leq P_1 + P_2 \end{aligned} \quad (13)$$

时间序列 \bar{S}_i 是一个随机向量, 因此不可能通过依靠逐点计算随机序列 \bar{S}_1 和 \bar{S}_2 之和以外的方法准确预测序列 \bar{S}_i 的峰值数值大小和在序列中所处位置。研究表明, 当相关系数大于-0.5时, 两台初始VM各自的PRU的数值和所处的位置比两台VM的整体互补程度更能影响最终形成的CVM的PRU特征。图2展示的是一个相关系数为-0.38的初始VM整合配对的例子。通常, -0.38在整合产生的相关系数里被认为是一个相对较小的值。仿真结果表明CVM的总体优势被两个初始VM的PRU区域强正相关后产生的峰值区域所影响。

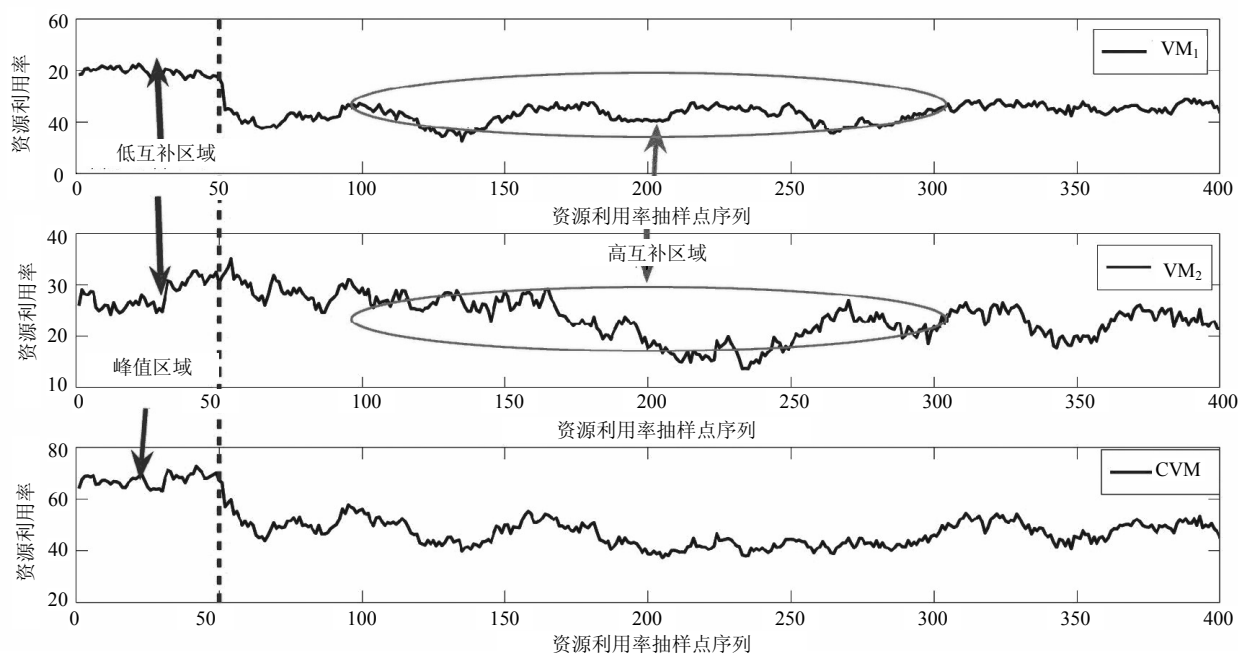


图2 相关系数为-0.38的CVM的组合峰值分析

因此并不能直接断言初始VM的低相关系数一定和CVM的低PRU相关。但另一项研究结果显示这两个参数是有关联的。这个结论建立在对10万个相关系数位为负的VM整合配对实例的数据回归分析的基础上。研究表明当相关系数低于-0.5时, 生成

的CVM的PRU和资源平均利用率的差值变化可以保持在一个相对较小的线性区域。

该研究的仿真结果如图3所示, 当一对初始状态虚拟机的线性相关系数小于-0.5时, 其峰值均值差小于8.63%标准VM。一般认为, 当峰值均值差小于

10%标准VM时就可以将对应的CVM看作为一个静态参数的VM。基于这些分析, 可以推导出定理3。

定理 3 当两台VM整合后的线性相关系数小于-0.5时, CVM的PRU满足:

$$P_{CVM} \approx \text{mean}(\vec{S}_1 + \vec{S}_2)$$

s.t. correlation coefficient ≤ -0.5 (14)

意味着当一个整合配对组的相关系数(CC)不大于-0.5时, 可以在整合时使用CVM的资源利用率均值代替资源利用率峰值(PRU), 从而有效地避免预留资源浪费。解决了上述问题, 可以在整合 ICMA_{-0.5} 和 FFD算法的基础上设计SICC算法。

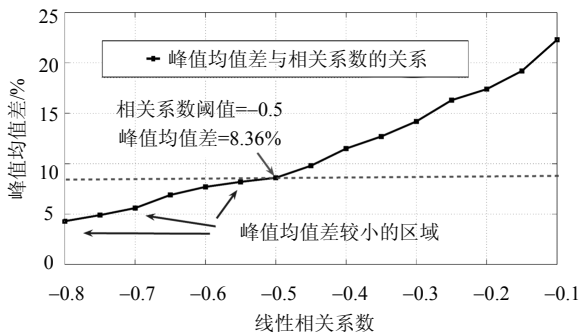


图3 CVM峰值均值差与相关系数的关系

算法1 ICMA_{-0.5} 算法流程

- 1) 将现有的N个VM/CVM放入待配对序列;
- 2) 使用ARMA算法生成 $N \times 1440$ 原始数据矩阵, 并使用式(1)生成 $N \times N$ 维CC矩阵;
- 3) 矩阵中搜索 coeff_{\min} :
 - ① 若 $\text{coeff}_{\min} \leq -0.5$ 且 $\text{PRU}_{CVM} \leq 100\%$, 令 $N=N-1$, 记录下当前 coeff_{\min} , 使用新的CVM数据更新 $(N-1) \times 1440$ 原始数据矩阵和 $(N-1) \times (N-1)$ CC矩阵;
 - ② 若 $\text{coeff}_{\min} \leq -0.5$ 且 $\text{PRU}_{CVM} > 100\%$, 令 $\text{coeff}_{\min}=1$ 并跳到步骤①;
 - ③ 若 $\text{coeff}_{\min} > -0.5$, 算法结束。

执行上述SICC算法后, 虚拟资源管理工具可以方便地再次采用FFD算法将SICC最终生成的CVM聚合到物理服务器中^[11]。

3 仿真性能分析

本文的实验在不同的方面对SICC算法进行评估。实验使用了195个初始PRU分布在15%~75%的实际VM数据。初始PRU的数据分布符合实测数据常用的假设, 即峰值分布30%上限, 标准差分布15%上限^[12]。令 μ_i 和 δ_i 分别代表初始状态虚拟机 VM_i 的均值和方差。则实验中所有初始状态虚拟机VM的平均均

值 $\bar{\mu}$ 和平均方差 $\bar{\delta}$ 大致满足关系:

$$\begin{cases} \bar{\mu} = \frac{\sum_{i=1}^N \mu_i}{N} = 30\%c \\ \bar{\delta} = \frac{\sqrt{\sum_{i=1}^N \delta_i^2}}{N} = 15\%c \end{cases} \quad (15)$$

并且, 在195个初始状态VM中, 有17个低容量初始VM、136个中等容量VM和42个高容量VM, 分别对应于 $\text{PRU} < 25\%$, $25\% \leq \text{PRU} \leq 50\%$ 和 $\text{PRU} > 50\%$ 。

算法2 SICC算法流程

- 1) 将初始状态的VM分成3个部分:
 - ① 若 $\text{RU} < 25\%$, 将符合条件的VM放入第1部分, 并在第1部分内运行 ICMA_{-0.5};
 - ② 若 $25\% \leq \text{RU} \leq 50\%$, 将符合条件的VM放入第2部分, 并在第2部分内运行 ICMA_{-0.5};
 - ③ 若 $\text{RU} > 50\%$, 将符合条件的VM放入第3部分, 并在第3部分内运行 ICMA_{-0.5}。
- 2) 将步骤1)中产生的所有的CVM合并, 并再次运行 ICMA_{-0.5}。
- 3) 依次计算步骤2)中各个CVM的 PRU_{CVM} 。并将这些 PRU_{CVM} 按顺序放入存储列表中。
- 4) 逆序排列 PRU_{CVM} 存储列表, 并运行FFD算法进行第2阶段整合。

图4比较了相同初始条件下ICMA算法产生的结果和SICC第二阶段再次整合后的结果。仿真数据表明, 第二阶段完成后, SICC算法最后剩余的CVM数量明显少于ICMA剩余的数量。并且, SICC产生的CVM在峰值均值差远小于ICMA的情况下, 平均PRU高出了ICMA相同指标13%。结果证明了SICC能有效地利用FFD带来的高效的全局整合顺序增益。

图5对SICC的资源管理控制工具在数据中心服务器级别的效能进行了研究。使用一种常用的初始假设条件, 即初始状态虚拟机的资源利用率平均值符合30%平均值标准VM资源, 标准差为15%标准VM资源。实验中设置了3 000台符合上述假设的初始状态VM, 并假设每台物理服务器的CPU资源等效于10台标准VM的CPU资源。资源管理工具在上述假设条件下进行了100次服务器级的VM整合及再聚合仿真实验。图5指出SICC整合后CVM的平均值为1 274个, 只占ICMA整合后CVM数量的85%。并且1 274个CVM平均需要125个服务器进行聚合, 是同等级初始条件下ICMA所需服务器数量的95%。该仿真

证明，这种基于SICC的资源管理工具在提高服务器资源利用率方面是相当有效的。

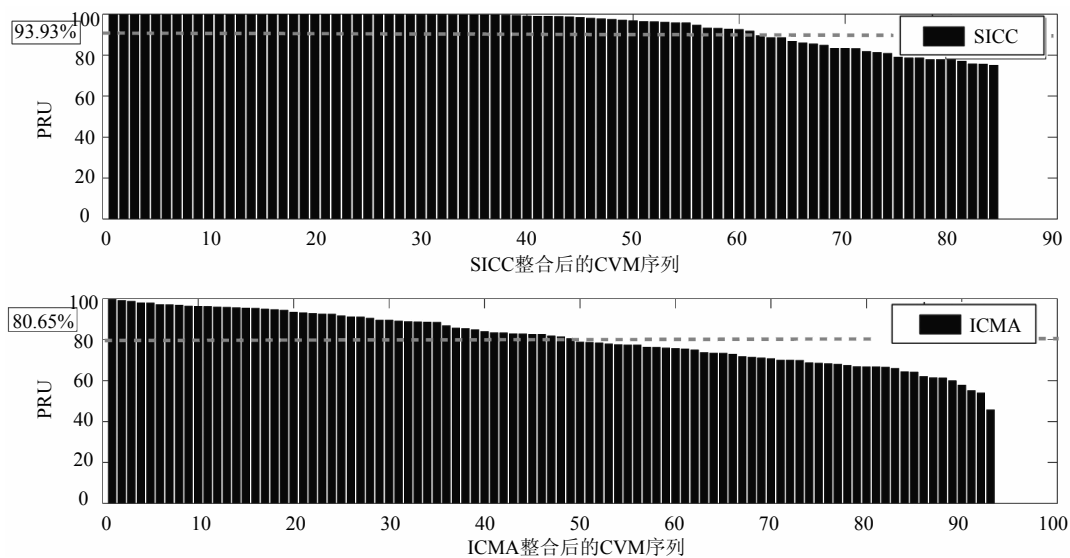


图4 初始状态VM总量与SICC第二阶段完成后的CVM总量对比

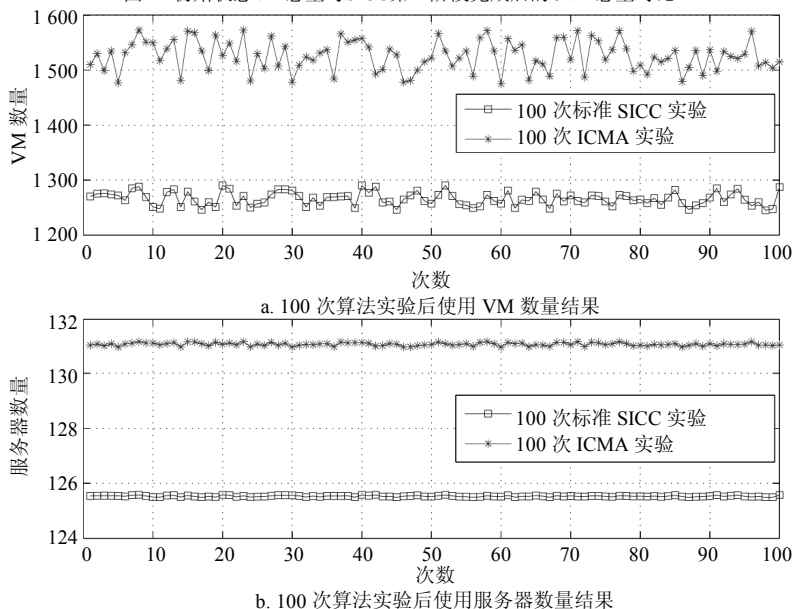


图5 SICC和ICMA的VM整合及服务器聚合对比实验

4 结论

本文介绍了一种基于SICC算法的新型虚拟资源管理工具。首先，研究了一种动态相关整合算法ICMA并指出ICMA的主要缺陷是不合理的VM整合顺序。公式化了基于最小相关系数的动态VM整合模型，并分析了这种只基于相关系数参数的策略。研究指出，这种模型不可能在相关系数和资源利用率间建立有效的联系。随后，给出了一系列关于SICC算法优点的理论推导，证明其可以通过整合中引进对峰值均值差的抑制以及FFD算法达到改变ICMA算法中不合理整合顺序的目的。最后，3个数值仿真

实验证明SICC算法在抑制CVM峰值、减少CVM数量和提高CVM资源利用率3个方面非常有效。本文证明了以SICC算法为核心的虚拟资源管理工具比基于ICMA核心的管理工具有更高的资源利用效率。

参考文献

[1] GONG W, CHEN Z, YAN J, et al. An optimal VM resource allocation for near-client-datacenter for multimedia cloud[C] //Ubiquitous and Future Network (ICUFN). Shanghai: IEEE, 2014: 249-255.

(下转第480页)

- 1999, 263(4): 341-346.
- [15] KLEINBERG J. Navigation in a small world[J]. *Nature*, 2000, 406(6798): 845.
- [16] KLEINBERG J. The small-world phenomenon: an algorithmic perspective[C]//Proc of 32nd Annual ACM Symposium on Theory of Computing. New York, USA: ACM Press, 2000.
- [17] WATTA D J, STROGATZ S H. Collective dynamics of 'small-world' networks[J]. *Nature*, 1998(393): 440-442.
- [18] SUN Ye, LIU Chuang, ZHANG Chu-xu, et al. Epidemic spreading on weighted complex networks[J]. *Phys Lett A*, 2014, 378(7): 635-640.
- [19] ZHANG Zi-ke, ZHANG Chu-xu, HAN Xiao-pu, et al. Emergence of blind areas in information spreading[J]. *PLoS ONE*, 2014, 9(4): e95785.
- [20] 王伟, 杨慧, 龚凯, 等. 复杂网络上的局域免疫研究[J]. *电子科技大学学报*, 2013, 42(6): 817-830.
WANG Wei, YANG Hui, GONG Kai, et al. Local immunization algorithm on complex networks[J]. *Journal of University of Electronic Science and Technology of China*, 2013, 42(6): 817-830.
- [21] BULDYREV S V, PARSHANI R, PAUL G, et al. Catastrophic cascade of failures in interdependent networks[J]. *Nature*, 2010, 464: 1025-1028.
- [22] FUNK S, SALATHÉ M, JANSEN V A A. Modeling the influence of human behavior on the spread of infectious diseases: a review[J]. *R Soc Interface*, 2010, 7(50): 1257-1274.

编辑 黄 莘

(上接第360页)

- [2] PAPAGIANNI C, LEIVADEAS A, PAPAVALASSIOLIS S, et al. On the optimal allocation of virtual resources in cloud computing networks[J]. *IEEE Transactions on Computers*, 2013, 62(6): 1060-1071.
- [3] LIU K, PENG J, LIU W, et al. Dynamic resource reservation via broker federation in cloud service: a fine-grained heuristic-based approach[C]//IEEE Global Communications Conference (GLOBECOM). [S.l.]: IEEE, 2014: 2338-2343.
- [4] BELOGLAZOV A, ABAWAJY J, BUYYA R. Energy-aware resource allocation heuristics for efficient management of data centers for cloud computing[J]. *Future Generation Computer Systems*, 2012, 28(5): 755-768.
- [5] BUYYA R, RANJAN R, CALHEIROS R N. Intercloud: Utility-oriented federation of cloud computing environments for scaling of application services, algorithms and architectures for parallel processing[J]. *Computer Science*, 2010, 6081: 13-31.
- [6] MENG X, ISCI C, KEPHART J, et al. Efficient resource provisioning in compute clouds via VM multiplexing[C]//The 7th International Conference on Autonomic Computing (ICAC). New York, USA: ACM, 2010: 11-20.
- [7] VERMA A, DASGUPTA G, NAYAK T K, et al. Server workload analysis for power minimization using consolidation[C]//USENIX Annual Technical Conference. [S.l.]: USENIX Association Berkeley, 2009: 28-28.
- [8] APTE R, HU L, SCHWAN K, et al. Discovering dependencies between virtual machines using cpu utilization[C]//The 2nd Conference On Hot Topics in Cloud Computing (HotCloud). California, USA: USENIX Association Berkeley, 2010: 17-23.
- [9] WAN J, PAN F, JIANG C. Placement strategy of virtual machines based on workload characteristics[C]//IEEE 26th International Parallel and Distributed Processing Symposium Workshops and PhD Forum (IPDPSW). [S.l.]: IEEE, 2012: 1827.
- [10] DINDA P A. The statistical properties of host load[C]//Fourth Workshop on Languages, Compilers, and Run-Time Systems for Scalable Computers. Berlin: Springer Heidelberg, 1998: 1-23.
- [11] MARTELLO S, TOTH P. Knapsack problems: Algorithms and computer implementations[M]. New York, USA: Wiley, 1990.
- [12] MISHRA A, HELLERSTEIN J L, CIRNE W. Towards characterizing cloud backend workloads: Insights from google compute clusters[J]. *ACM Sigmetrics Performance Evaluation Review*, 2009, 37(4): 34-41.

编辑 叶 芳