

面向数据中心的虚拟机部署及优化算法

王加昌*, 曾 辉, 何腾蛟, 张 娜

(中国核动力研究设计院 核反应堆系统设计技术重点实验室, 成都 610041)

(*通信作者电子邮箱 jiachang.wang@163.com)

摘 要: 虚拟机动态配置是解决数据中心能耗低效的有效方法。针对动态配置过程中的虚拟机部署及优化问题展开研究, 提出一种新的面向系统能耗的虚拟机部署算法以及基于主动迁移的优化策略。为了降低系统能耗, 新算法采用基于服务器利用率的最佳适配降序算法求解虚拟机部署方案; 同时为了适应应用负载的动态变化, 新算法启动主动迁移策略对部署方案进行优化, 即通过启发式算法在当前部署的基础上搜索使系统能耗更低的优化方案, 并根据新部署对虚拟机执行主动迁移。考虑到迁移会导致应用服务质量降级和额外能耗, 新算法通过在优化策略中设置基于服务器利用率的启动门限, 对虚拟机主动迁移频率进行控制。仿真实验表明, 所提算法在系统能耗、虚拟机迁移频率、服务器状态切换频率以及服务质量等多项性能指标上均有显著提高。

关键词: 数据中心; 动态部署; 能耗优化; 主动迁移

中图分类号: TP393 **文献标志码:** A

Virtual machine placement and optimization for data center

WANG Jiachang*, ZENG Hui, HE Tengjiao, ZHANG Na

(Laboratory of Science and Technology on Reactor System Design Technology,
Nuclear Power Institute of China, Chengdu Sichuan 610041, China)

Abstract: Dynamic consolidation of Virtual Machine (VM) is a promising solution to address the energy inefficiency of data centers. This paper focused on VM placement and its optimization. First, in order to improve the energy efficiency, a CPU utilization-based best fit decreasing algorithm was presented to complete the VM placement. However, due to the variability of workloads experienced by applications, the VM placement should be optimized continuously in an online manner. Therefore, a threshold-based active VM migration mechanism was proposed to solve the dynamic optimization. Extensive simulation results show the proposed algorithms can significantly reduce the energy consumption and the number of VM migrations, while keeping the metrics of Performance Degradation due to Migration (PDM) and Overload Time per Active Server (OTAS) in low level.

Key words: data center; dynamic placement; energy optimization; active migration

0 引言

近年来, 数据中心因其良好的扩展性、灵活的管理机制受到业界和学术界的广泛关注^[1]。然而随着数据中心的大规模部署, 物理设备运行导致的高能耗问题日益突出。美国能源部报告^[2]指出, 2006 年数据中心的能耗占全美所有能耗的 1.5%, 并且对电能的需求仍以每年 12% 的速度增长。高能耗不仅导致电能的浪费, 而且其排放的大量二氧化碳会加剧温室效应, 对环境造成不良影响。文献^[3]指出数据中心的高能耗主要由两方面导致: 一是不断增加的硬件设施数量; 二是大量服务器的低效利用。文献^[4]对数据中心的 5000 台活跃服务器进行监测, 发现服务器通常只运转在其服务能力的 10%~50%; 同时服务器本身的能耗范围较窄, 即使完全空闲的服务器, 其消耗的电能仍达到峰值时电能消耗的 70% 左右。图 1 展示了服务器能耗与其利用率之间的关系^[3], 观察发现只要服务器处于活跃状态就存在基本能耗, 且能耗效率与服务器利用率呈近似正比的关系。因此如何提高服务器利用率是改善能耗低效的关键问题之一。

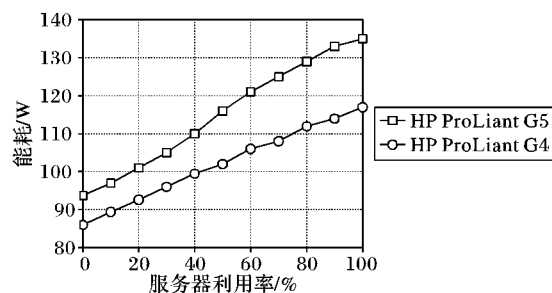


图 1 能耗与服务器利用率的关系

虚拟化技术是解决能耗低效的有效途径, 其基本思路是: 数据中心的物理资源以虚拟机的方式提供给应用进行部署, 因此允许在一台物理服务器上配置多个虚拟机, 从而达到提高服务器利用率、改善能耗效率的目的^[5]。然而对于负载可变的应用(例如 Web 应用), 如果按照应用的峰值负载去部署, 虚拟机有可能大部分时间都处于低负载状态, 即对应的物理服务器仍然存在能耗低效问题。为此, 研究者提出虚拟机动态配置方案^[6], 即虚拟机可根据当前部署应用的实际负载进行动态迁移, 集中到部分活跃服务器上, 从而提高服务

收稿日期: 2013-04-08; 修回日期: 2013-06-23。 基金项目: 国防军工技术基础“十二五”科研项目(科工技[2010]1425 号-41)。

作者简介: 王加昌(1978-), 男, 四川广元人, 工程师, 硕士, 主要研究方向: 高性能计算、资源管理; 曾辉(1972-), 男, 四川成都人, 研究员级高级工程师, 硕士, 主要研究方向: 高性能计算、云计算; 何腾蛟(1983-), 男, 四川成都人, 工程师, 硕士, 主要研究方向: 高性能计算; 张娜(1981-), 女, 山东青岛人, 工程师, 硕士, 主要研究方向: 计算机仿真。

器的利用率和能耗效率;同时将处于空闲状态的服务器切换到低能耗(或休眠)状态,以节省电能,如图2所示。

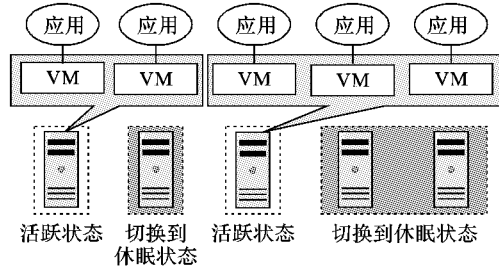


图2 虚拟机动态部署示意图

目前虚拟机动态配置方案的研究已经取得积极成果^[7-12]。例如文献[7]将能耗管理定义为连续优化问题,提出求解该问题的 LLC (Limited Lookahead Control) 算法;其目标是 minimized 系统能耗及服务等级协议 (Service-Level Agreement, SLA) 违约开销,但是算法需要预先知道应用对资源的需求,通过仿真学习达到对算法参数的优化配置,因此 LLC 算法不适用于负载动态变化的应用部署。文献[8]针对能耗优化,提出基于启发式 Bin-Packing 算法的虚拟机动态部署方案;该方案在保证应用性能的前提下,最优化能耗及虚拟机迁移开销,该方案与本文的研究思路类似,但它只针对应用负载固定不变的情况,不考虑负载动态变化引起的应用性能降级。文献[9]考虑虚拟机部署请求连续到达的情况,提出 BSP (Backward Speculative Placement) 策略,通过虚拟机对资源需求的历史数据为即将部署的虚拟机预先分配一个备选的目标服务器;BSP 主要考虑预分配服务器是否满足虚拟机的资源需求,没有考虑能耗因素。文献[10]对虚拟机迁移过程中的迁移虚拟机选择算法展开研究,通过详细分析 CPU 温度、资源利用率、能耗对虚拟机迁移的影响,提出面向多目标优化的迁移虚拟机选择算法;迁移虚拟机选择是动态配置的关键环节之一,与本文研究的部署算法配合使用。在已有方案中,受到广泛关注的是文献[3,11]提出的考虑能耗优化和服务性能的虚拟机动态配置方案,但是该方案仍然在系统能耗、虚拟机迁移频率、服务器状态切换频率以及因迁移引起服务性能降级等方面存在较大改进空间,详细分析参考第1章。

本文在文献[3]的虚拟机动态配置方案的基础上,针对其中的虚拟机部署算法及轻负载迁移机制展开研究,提出面向系统能耗优化的虚拟机部署 (VM Placement based on System Energy Optimization, VP-SEO) 算法以及基于门限的主动迁移 (Threshold-based Active VM Migration, T-AVM) 策略。大量仿真实验表明,在不同应用负载场景下,本文算法在系统能耗、虚拟机迁移频率、服务器状态切换频率以及服务质量等多项性能指标上均有显著提高。

1 虚拟机动态配置方案

文献[3]提出的考虑能耗优化和服务性能的虚拟机动态配置方案主要包括四个关键算法:虚拟机部署、轻负载情况下的虚拟机迁移、服务器过载预测及迁移虚拟机选择。其中虚拟机部署算法——能量感知的最佳适配降序 (Power Aware Best Fit Decrease, PABFD) 算法的作用是为待部署的虚拟机计算其需要部署的目标服务器。PABFD 主要应用在三个环节:1) 虚拟机初始部署;2) 当服务器处于轻负载状态,需要将其上运行的所有虚拟机迁移到其他活跃服务器(主动迁移),以优化能耗;3) 当服务器检测到过载风险,需要将其上运行的

部分虚拟机迁移到其他活跃服务器(被动迁移),以避免过载导致服务性能降级。PABFD 算法描述如下:

```

Input: serverList, vmList;
Output: allocation of VMs;
vmList.sortDecreasingUtilization()
foreach vm in vmList do
    minPower ← MAX
    allocatedServer ← NULL
    foreach server in serverList do
        if server has enough resources for vm then
            power ← estimatedPower( Server, vm)
            if power < minPower then
                allocatedServer ← server
                minPower ← power
    if allocatedServer ≠ NULL then
        allocation.add( vm, allocatedServer)
return allocation

```

PABFD 的优点是考虑了每个虚拟机部署的能耗优化且算法复杂度低,但是仍然存在以下问题:1) 算法会导致大量的虚拟机迁移及频繁的服务器状态切换;2) 算法只考虑单个虚拟机部署的能耗最低,但是对于整个部署方案的系统能耗非最优。究其原因,是 PABFD 在部署时将已部署虚拟机的服务器(活跃状态)和未部署虚拟机的服务器(休眠状态)均作为活跃状态对待,能耗计算如式(1):

$$estimatedPower = allocatedPower - unallocatedPower \quad (1)$$

其中: $unallocatedPower$ 、 $allocatedPower$ 分别表示部署前和部署后服务器的能耗,如果服务器部署前未部署其他虚拟机,则 $unallocatedPower =$ 服务器的基础能耗。因此在 PABFD 中,有可能出现式(2)所述的情况,其中 $estimatedPower'$ 表示虚拟机部署到活跃服务器产生的能耗增量, $estimatedPower''$ 表示部署到休眠服务器产生的能耗增量,从而导致系统启动更多服务器,然而活跃服务器数量越多,产生的系统能耗也就越大。

$$estimatedPower' > estimatedPower'' \quad (2)$$

考虑到 PABFD 部署算法对于系统能耗是非最优的以及应用负载的动态变化,文献[3]提出轻负载状态下的主动迁移机制,来优化系统能耗。该机制通过周期性将系统中利用率最低的服务器上运行的所有虚拟机迁移到其他活跃服务器,从而尽量减少活跃服务器数量,降低能耗。但是将分散部署的虚拟机主动迁移到少数活跃服务器,必然导致服务器状态频繁切换以及迁移次数增加。此外,虚拟机根据应用的实际负载而非峰值负载来动态部署,也会导致服务器存在过载风险,即部署在该服务器上的应用对资源需求的总和超过了该服务器能够提供的最高容量限制,从而导致服务质量降级。因此,为了避免服务器发生过载,文献[3]提出绝对离差中位数 (Median Absolute Deviation, MAD)、四分位距 (InterQuartile Range, IQR)、局部回归 (Local Regression, LR) 等多种过载预测算法。当预测到服务器存在过载风险时,需要将该服务器上运行的部分虚拟机迁移到其他活跃服务器上(被动迁移),以使服务器的实际负载控制在安全范围内。在文献[3]中提出随机选择 (Random Choice, RC)、最小迁移时间 (Minimum Migration Time, MMT)、最大相关 (Maximum Correlation, MC)^[12] 等多种迁移虚拟机选择算法来确定需要迁移的虚拟机集合。

综上分析,虚拟机部署算法及轻负载迁移机制,是导致虚拟机频繁迁移及服务器状态频繁切换的主要原因。为此,本

文的研究主要集中在这两个方面。

2 虚拟机部署算法及优化策略

2.1 问题描述

假设系统中需要部署的虚拟机用 $VMList = (vm_1, vm_2, \dots, vm_N)$ 表示, vm_i 对 CPU、内存、带宽的资源需求用 $vm_i = (CPU_i, RAM_i, BW_i)$ 表示, 服务器用 $ServerList = (Server_1, Server_2, \dots, Server_M)$ 表示, $Server_c = (CPU_c, RAM_c, BW_c)$ 表示服务器的容量限制, 服务器能耗用 $Power(Server_j)$ 表示, 虚拟机部署方案用矩阵 X 表示, 如式(3), 则系统能耗最优化的虚拟机部署问题可用式(4)表示:

$$X = \begin{bmatrix} x_{11} & x_{12} & \dots & x_{1M} \\ x_{21} & x_{22} & \dots & x_{2M} \\ \vdots & \vdots & & \vdots \\ x_{N1} & x_{N2} & \dots & x_{NM} \end{bmatrix}_{N \times M} \quad (3)$$

其中 $x_{ij} = \begin{cases} 1, & vm_i \text{ 部署在 } Server_j \text{ 上} \\ 0, & \text{其他} \end{cases}$

$$\begin{aligned} \min \text{SystemPower} &= \sum_{j=1}^M Power(Server_j) \\ \text{s. t.} \quad &\sum_{j=1}^M x_{ij} = 1; i \in (1, 2, \dots, N) \\ &\sum_{i=1}^N CPU_i \cdot x_{ij} \leq CPU_c; j \in (1, 2, \dots, M) \\ &\sum_{i=1}^N RAM_i \cdot x_{ij} \leq RAM_c; j \in (1, 2, \dots, M) \\ &\sum_{i=1}^N BW_i \cdot x_{ij} \leq BW_c; j \in (1, 2, \dots, M) \end{aligned} \quad (4)$$

其中 $Server_j = [x_{1j} \ x_{2j} \ \dots \ x_{Nj}]^T$ 。

因此, 求解满足式(4)的矩阵 X_{OPT} , 即是本文关注的系统能耗最优化的虚拟机部署方案。式(4)中的 $Power()$ 是服务器的能耗函数, 与其本身的物理资源配置、部署虚拟机的负载有关。文献[3]指出, 当无虚拟机部署在服务器上时, 可以将服务器切换到休眠状态, $Power() = 0$; 当有虚拟机部署在服务器上时, 服务器处于活跃状态, $Power()$ 是一个有基本能耗的线性分段函数, 如图1所示。

2.2 面向系统能耗优化的虚拟机部署算法

求解 X_{OPT} 是 NP-Complete 问题, 无法在多项式时间内精确求解。为此本文提出一个面向系统能耗优化的虚拟机部署算法 VP-SEO。该算法的核心思想是将虚拟机部署在满足资源需求且部署后能耗效率最高(即服务器利用率最高)的服务器上。伪代码描述如下:

```
VP_SEO (ServerList, VMList)
1) vmList.sortDecreasingUtilization()
2) foreach vm in VMList do
3)   maxUtilization ← 0
4)   allocatedServer ← NULL
5)   foreach server in ServerList do
6)     if server has enough resources for vm then
7)       if server is powered off then
8)         poweredoffServerList.add(server)
9)       else
10)        utilizationAfterAllocation ←
            getUtilizationAfterAllocation(server, vm)
11)    if utilizationAfterAllocation > maxUtilization then
12)      allocatedServer ← server
13)    maxUtilization ← utilizationAfterAllocation
```

```
14)    if allocatedServer ≠ NULL then
15)      allocationMap.add(vm, allocatedServer)
16)    else if poweredoffServerList is not empty then
17)      allocatedServer ← poweredoffServerList[0]
18)      allocationMap.add(vm, allocatedServer)
19)    return allocationMap
```

2.3 基于门限的虚拟机主动迁移策略

由于虚拟机部署受到 CPU、内存、带宽等多种资源的容量限制, 因此 VP_SEO() 得到的部署方案只是充分考虑能耗效率的优化解, 而非系统能耗的最优解。另一方面, 由于服务器运行期间应用负载动态变化, 因此需要根据实际的虚拟机负载, 对虚拟机部署进行动态调整。为此, 本文提出一个基于门限的虚拟机主动迁移策略 T-AVM, 以实现虚拟机部署的动态优化。其基本思想是通过启发式算法, 在当前部署方案的邻域解中搜索使系统能耗更优的部署方案, 根据新方案对虚拟机进行迁移, 把迁移后空闲的服务器切换到休眠状态。然而虚拟机在迁移过程中, 会引起应用的服务质量降级, 且迁移和服务器状态切换也会产生一定的能耗。因此在设计迁移策略时, 应考虑到能耗优化和虚拟机迁移频率的折中。在 T-AVM 中, 通过设置服务器最低利用率门限来控制迁移频率, 即只有当服务器利用率低于门限值时才启动主动迁移策略。伪代码描述如下:

```
T_AVM (allocatedServerList, threshold, allocationMap)
1) if allocatedServerList is empty then
2)   return allocationMap
3) minUtilization ← 1
4) underUtilizedServer ← NULL
5) foreach server in allocatedServerList do
6)   utilization ← server.getUtilizationOfCPU()
7)   if utilization < threshold and utilization < minUtilization then
8)     minUtilization ← utilization
9)     underUtilizedServer ← server
10)  if underUtilizedServer == NULL then
11)    return allocationMap
12)  else
13)    allocatedServerList.remove(server)
14)    currAllocationMap ← VP_SEO(allocatedServerList,
        underUtilizedServer.getVmList())
15)  if currAllocationMap is empty or not all vms can be migrated
        then
16)    return allocationMap
17)  else
18)    allocationMap.add(newAllocationmap)
19)  T_AVM (allocatedServerList, threshold)
20) return allocationMap
```

3 仿真实验及分析

3.1 仿真方案设计

本文利用 CloudSim^[13] 仿真平台对虚拟机动态部署方案进行原型仿真, 并从系统能耗、虚拟机迁移频率、服务器状态切换频率、服务器平均过载时间、因迁移引起的服务质量降级等方面对文献[3]提出的虚拟机配置方案和本文提出的基于 VP-SEO 和 T-AVM 的配置方案进行性能比较。需要说明的是由于本文只关注虚拟机部署算法和轻负载迁移机制, 因此服务器过载预测及迁移虚拟机选择沿用了文献[3]的基于局部回归的预测算法 LR 和基于最小迁移时间的选择算法 MMT。为便于描述, 文献[3]的方案称为 PABFD, 本文的算法称为 VP-SEO。仿真假定在一个数据中心, 均采用 HP ProLiant G5

服务器,其能耗模型是分段线性函数。设置 4 种虚拟机类型,仿真时间为 86 400 s,主要参数设置如表 1 所示,本文采用的评价指标如表 2 所示。

表 1 仿真参数设置

参数	数值
服务器的容量限制	[5 320 MIPS, 8 192 MB, 1 Gb/s]
虚拟机的 CPU 配置	[2 500, 2 000, 1 000, 500] (MIPS)
虚拟机的 RAM 配置	[870, 1 740, 1 740, 613] (MB)
虚拟机的 BW 配置	[100, 100, 100, 100] (Mb/s)

表 2 评价指标说明

评价指标	含义
$Energy$	服务器总能耗
M_{VM}	虚拟机迁移次数
M_{SH}	服务器从活跃切换到休眠状态的次数
T_{SH}	服务器处于连续活跃状态的平均时间
$OTAS$	服务器处于过载状态的时间比例
PDM	虚拟机迁移引起的性能降级比例

$$OTAS = \frac{1}{M} \sum_{j=1}^M \frac{TO_j}{TA_j} \quad (5)$$

$$PDM = \frac{1}{N} \sum_{i=1}^N \frac{CD_i}{CR_i} \quad (6)$$

其中: M 表示服务器数量, TO_j 表示服务器 j 处于过载状态的时间, TA_j 表示服务器 j 处于连续活跃状态的时间; N 表示虚拟机数量, CR_i 是虚拟机 i 请求的 CPU 容量总和, CD_i 记为虚拟机 i 在迁移期间请求 CPU 容量的 10%。

3.2 仿真结果分析

3.2.1 负载随机变化时的性能对比

实验 1 考察负载随机变化情况下两种动态配置方案的性能,仿真结果如表 3 所示。仿真中设置的可用服务器数为 50,启动的虚拟机数为 50,每个更新周期虚拟机负载按其容量限制的 $[0,100]$ (%) 随机设置,VP-SEO 中主动迁移机制的最低利用率门限设置为 1.0。

表 3 实验 1 中两种方案的性能对比

算法	$Energy$ /kWh	M_{VM}	PDM /%	$OTAS$ /%	M_{SH}	T_{SH} /s
PABFD	36.82	3865	0.20	15.54	1 102	1 038.33
VP-SEO	30.07	2109	0.13	18.31	176	4 562.47

从表 3 看出,与 PABFD 相比,VP-SEO 的能耗有一定程度的降低,减少 18.33%;同时虚拟机迁移次数和服务器从活跃到休眠状态的切换次数分别减少 45.54% 和 84.03%,改善非常显著;迁移导致的性能降级减少 0.07%。这是因为 VP-SEO 在虚拟机部署时考虑了系统能耗的优化,使运行期间服务器利用率较高,需要启动主动迁移的情况较少,由迁移引起的性能降级也随之减少。同时主动迁移次数减少使得服务器从活跃到休眠状态的切换次数也大幅下降,而服务器处于活跃状态的平均时间大幅提高,达到 76 min,而 PABFD 只有约 17 min,说明 PABFD 中服务器的状态切换非常频繁。两种方案的过载时间比例都较高,分别达到活跃状态时间的 15.54% 和 18.31%,且 VP-SEO 比 PABFD 高约 2.77%,说明 VP-SEO 的过载风险更高。这是因为两种方案均采用基于局部回归的服务器过载预测算法,该算法对于负载随机变化的情况准确度较差。此外,VP-SEO 是按照服务器利用率最大化来设计部署算法,而过高的服务器利用率,必然存在更高的过载风险。

3.2.2 特定负载场景的性能对比

实验 2 考察应用处于轻负荷、中等负荷、近饱和负荷三种场景下算法的性能对比。仿真设置轻负荷场景中,虚拟机负载在其容量限制的 $[0,30]$ (%) 范围中随机选择;中等负荷场景中,负载在其容量限制的 $(30,75]$ (%) 范围内随机选择;近饱和场景中,负载在其容量限制的 $(75,100]$ (%) 范围内随机选择。VP-SEO 启动主动迁移机制的最低利用率门限设置为 0.5。仿真结果如表 4~表 6 所示。

表 4 轻负荷场景下两种算法的性能对比

算法	$Energy$ /kWh	M_{VM}	PDM /%	$OTAS$ /%
PABFD	10.89	1 531	0.05	3.79
VP-SEO	9.14	558	0.01	2.31

表 5 中等负荷场景下两种算法的性能对比

算法	$Energy$ /kWh	M_{VM}	PDM /%	$OTAS$ /%
PABFD	35.37	1 830	0.07	5.00
VP-SEO	31.72	971	0.03	3.55

表 6 近饱和负荷场景下两种算法的性能对比

算法	$Energy$ /kWh	M_{VM}	PDM /%	$OTAS$ /%
PABFD	54.34	371	0.02	0.46
VP-SEO	53.06	346	0.01	0.30

观察发现,随着系统负载的增加,两种算法启动的服务器数量均随之增加,导致耗电量均大幅提升。在三种负载场景下,VP-SEO 算法的性能均优于 PABFD,特别是虚拟机迁移次数指标有较大幅度的减少,轻负载时减少 63.55%,中等负载时减少 46.94%,重负载时减少 6.74%。这是因为 VP-SEO 在计算部署方案时就考虑到充分利用服务器资源,使虚拟机部署尽可能集中,因此在其优化过程中需要启动的主动迁移次数就相对减少,从而使得因迁移导致的服务性能降级比例也随之减少。PABFD 在初始部署时比较分散,它需要通过主动迁移来优化部署,因此 PABFD 在低负载和中等负载的情况下会导致频繁的迁移。但是当系统处于重负载状态时,虚拟机部署时就接近最佳配置,因此两种算法在运行期间需要启动的主动迁移次数均大幅减少,从而使虚拟机迁移引起的性能降级也减少。

需要注意的是,两种算法在不同负载场景下,都会不可避免地使服务器在某些时间处于过载状态,这是因为两种算法都是按照应用负载的实际需求,而不是峰值需求来部署虚拟机。但是即使在重负载情况下,服务器处于过载的时间比例仍然控制在很低的范围内。

通过实验 2 的仿真分析,本文提出的 VP-SEO 算法在不同负载场景下,性能均比 PABFD 有一定程度的改善。

3.2.3 负载来自实际跟踪数据时的性能对比

实验 3 考察大规模服务器及虚拟机部署,且虚拟机负载来自实际跟踪数据时的算法性能。实验 3 中的负载数据来自文献[14]对虚拟机负载的实际跟踪数据,可用服务器数量为 800 台,仿真结果如图 3~6 所示,其中 VP-SEO-1.0 表示不设置主动迁移最低门限的 VP-SEO 方案,VP-SEO-0.3 表示设置门限为 0.3 的 VP-SEO 方案。

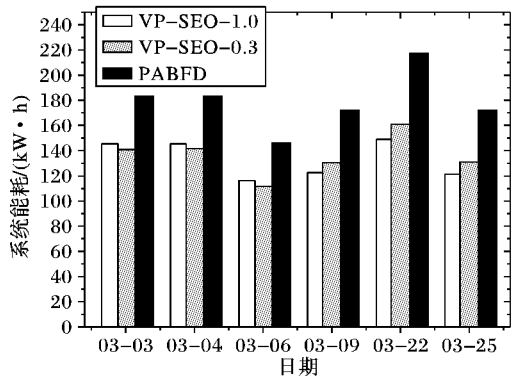


图3 系统能耗

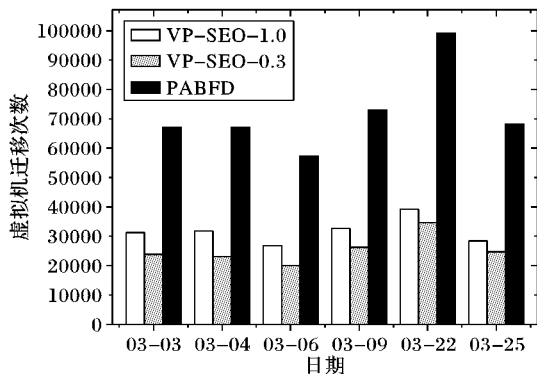


图4 虚拟机迁移次数

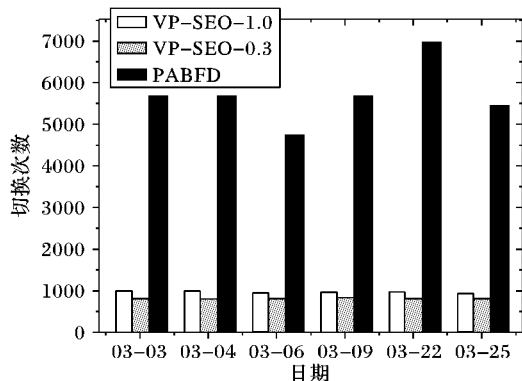


图5 服务器从活跃到休眠状态的切换次数

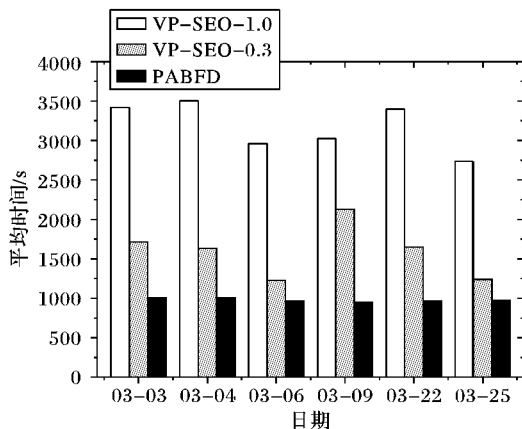


图6 服务器处于连续活跃状态的平均时间

从图3 观察可知,PABFD 方案的能耗明显高于 VP-SEO-1.0 和 VP-SEO-0.3,后两种方案的能耗无明显差别。这是因为 PABFD 算法在部署时只考虑单个虚拟机部署的能耗增量,使得系统需要启动更多的活跃服务器来完成部署,而活跃状态服务器的数量越多,产生的系统能耗越大;而 VP-SEO 方案是否设置门限对能耗无明显影响。图 4 是虚拟机迁移次数的

对比,PABFD 启动的迁移次数最多,VP-SEO-1.0 次之,VP-SEO-0.3 最少。这是因为 PABFD 可能导致虚拟机部署比较分散,在后续优化过程中 PABFD 会频繁启动虚拟机迁移,以实现将分散部署的虚拟机集中到少数活跃服务器上;而 VP-SEO 在部署时就以服务器利用率作为优化指标,其初始部署方案是面向系统能耗优化的,因此在后续优化过程中需要启动虚拟机迁移的情况明显减少,特别是增加启动门限值的 VP-SEO-0.3 算法,只有当服务器利用率小于 0.3 时,才允许启动主动迁移策略,从而有效控制了虚拟机迁移频率和服务器状态切换频率,如图 5 所示。图 6 显示两种方案中服务器处于活跃状态的平均时间,VP-SEO-1.0 的平均活跃时间最长,VP-SEO-0.3 次之,PABFD 最短,说明 PABFD 的状态切换最频繁,因此每台服务器的活跃状态时间最短。

表 7、8 显示三种方案在服务质量方面的对比。观察发现,在 6 天以实际数据作为负载的仿真中,每种方案 *PDM* 和 *OTAS* 两个指标均无明显变化,这说明负载方案不同对这两个指标的影响较小。对于 *PDM* 指标,由于 PABFD 启动的迁移次数最多,因此它的迁移引起的性能降级比例是最大的,VP-SEO-1.0 次之,VP-SEO-0.3 最小;而服务器过载比例显示也是 PABFD 最差,VP-SEO-1.0 次之,VP-SEO-0.3 最小。与 3.2.1 节负载随机变化场景下的结果不同,三种方案的过载时间比例都明显大幅降低,这是因为对于实际负载的情况,三种方案中采用的过载预测算法能够较准确地预测过载风险,从而通过一定的虚拟机迁移机制,避免了服务器过载的情况发生,而且 VP-SEO 表现出比 PABFD 更好的性能。

表 7 各种算法的 <i>PDM</i> 指标对比			%
日期	VP-SEO-1.0	VP-SEO-0.3	PABFD
03-03	0.06	0.04	0.13
03-04	0.06	0.04	0.13
03-06	0.06	0.04	0.13
03-09	0.07	0.05	0.14
03-22	0.05	0.04	0.13
03-25	0.05	0.04	0.12

表 8 各种算法的 <i>OTAS</i> 指标对比			%
日期	VP-SEO-1.0	VP-SEO-0.3	PABFD
03-03	4.19	2.89	6.35
03-04	4.13	2.63	6.35
03-06	4.30	2.76	6.50
03-09	4.85	3.12	7.23
03-22	4.26	2.94	6.68
03-25	4.21	2.72	6.29

综上所述,在不同的负载背景下,VP-SEO 在系统能耗、虚拟机迁移频率、服务性能降级、服务器状态切换频率等指标上均优于 PABFD 方案;虽然因为服务器高利用率导致存在一定的过载风险,但是通过服务器过载预测和设置最低利用率门限,可在很大程度上避免过载发生。

4 结语

高能耗已成为制约数据中心发展的关键因素,虚拟机动态配置方案被认为是解决数据中心能耗低效问题的有效方法。本文针对现有方案中存在的系统能耗非最优、虚拟机迁移频繁、服务器状态切换频繁等问题展开研究,通过建立系统能耗最优化的虚拟机部署模型,提出面向系统能耗优化的虚

拟机部署算法以及基于门限的虚拟机主动迁移策略。算法在部署虚拟机时以最优服务器能耗效率(服务器资源利用率)为指标,使系统整体部署方案在系统能耗方面得到优化。通过启发式搜索求解使服务器能耗效率最优的虚拟机部署方案;同时通过增加服务器利用率最低门限,控制虚拟机主动迁移频率。仿真实验表明,新算法在系统能耗、虚拟机迁移频率、服务器状态切换频率以及服务质量等多项性能指标上均有显著提高。

在将来的研究中,将更多地关注动态配置方案中的服务器过载预测算法的研究。例如,在文献[3]的方案中,只对服务器的负载进行监测,并依据历史数据预测下个周期服务器的负载情况,从而判断是否存在过载风险;但是服务器上部署的虚拟机是动态调整的,因此根据服务器的历史负载数据来预测过载风险是不准确的。如何改进预测算法,提高预测准确率是将来研究的重点。

参考文献:

- [1] 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.
- [2] KURP P. Green computing[J]. Communications of the ACM, 2008, 51(10): 11-13.
- [3] 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.
- [4] BARROSO L A, HOLZLE U. The case for energy-proportional computing[J]. IEEE Computer, 2007, 40(12): 33-37.
- [5] CLARK C, FRASER K, HAND S. Live migration of virtual machines[C]// Proceedings of the 2nd Symposium on Networked Systems Design and Implementation. Berkeley: USENIX Association, 2005: 273-286.
- [6] NORMAN B, ANDRZEJ K, BEATY K. Dynamic placement of virtual machines for managing SLA violations[C]// Proceedings of the 10th IFIP/IEEE International Symposium on Integrated Network Management. Piscataway: IEEE, 2007: 119-128.
- [7] KUSIC D, KEPHART J O, HANSON J E, *et al.* Power and performance management of virtualized computing environments via look-ahead control[J]. Cluster Computing, 2009, 12(1): 1-15.
- [8] VERMA A, AHUJA P, NEOGI A. pMapper: Power and migration cost aware application placement in virtualized systems[C]// Proceedings of the 9th ACM/IFIP/USENIX International Conference on Middleware. New York: Springer-Verlag, 2008: 243-264.
- [9] CALCAVECCIA N M, BIRAN O, HADAD E. VM placement strategies for cloud scenarios[C]// Proceedings of the 5th International Conference on Cloud Computing. Piscataway: IEEE, 2012: 852-859.
- [10] SONG A B, FAN W, WANG W, *et al.* Multi-objective virtual machine selection for migrating in virtualized data centers[C]// Proceedings of Pervasive Computing and the Networked World, LNCS 7719. Berlin: Springer, 2013: 426-438.
- [11] 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, 2011, 28(5): 755-768.
- [12] VERMA A, DASGUPTA G, NAYAK T K, *et al.* Server workload analysis for power minimization using consolidation[C]// Proceedings of the 2009 USENIX Annual Technical Conference. Berkeley: USENIX Association, 2009: 28-28.
- [13] 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.
- [14] PARK K S, PAI V S. CoMon: a mostly-scalable monitoring system for PlanetLab[J]. ACM SIGOPS Operating Systems Review, 2006, 40(1): 65-74.
- [15] SHANMUGAM S K, NIELSEN J, LACHAPPELLE G, *et al.* Differential signal processing schemes for enhanced GPS acquisition[C]// Proceedings of 18th International Technical Meeting of the Satellite Division of the Institute of Navigation. Bonn, German: Institute of Navigation, 2005: 212-221.
- [16] 莫建文, 欧阳缙, 肖海林, 等. 基于DBZP差分相干的GPS信号捕获算法[J]. 系统工程与电子技术, 2012, 34(6): 1085-1089.
- [17] 赵琳, 高帅和, 丁继成. 基于FFT的高动态GPS信号捕获方法优化[J]. 系统工程与电子技术, 2011, 33(1): 151-156.
- [18] 刘杨, 金天. 小波降噪在微弱GPS信号捕获中的应用[J]. 系统工程与电子技术, 2011, 33(4): 842-846.
- [19] 郝燕玲, 祖秉法, 黄平. “北斗二号”B2频段信号直接多普勒频移估计[J]. 华中科技大学学报, 2010, 38(6): 15-18.
- [20] LU Y. Software radio implementation of multi-frequency global navigation satellite system receiver[D]. Riverside: University of California, 2006.
- [21] 鲁豫. GPS全球定位接收机——原理与软件实现[M]. 北京: 电子工业出版社, 2009.
- [22] TSUI J B-Y. Fundamentals of global positioning system receivers: a software approach[M]. 2nd ed. New York: Wiley, 2004: 113-117, 192-227.
- [23] VETTERLI M, HERLEY C. Wavelets and filter banks: theory and design[J]. IEEE Transactions on Signal Processing, 1992, 40(9): 2207-2232.

(上接第2771页)

省约一半的时间,这对于研究复杂环境下要求拥有快速处理能力的全球导航卫星系统(Global Navigation Satellite System, GNSS)接收机具有良好的借鉴意义。

参考文献:

- [1] KOVAR P, KACMARIK P, VEJRAZKA F. Interoperable GPS, GLONASS and Galileo software receiver[J]. IEEE Aerospace and Electronic Systems Magazine, 2011, 26(4): 24-30.
- [2] 马永奎, 张一, 张中兆, 等. 改进的高动态高灵敏度GPS信号捕获算法[J]. 系统工程与电子技术, 2009, 31(2): 265-269.
- [3] 武玲娟, 崔莹莹, 路卫军, 等. 高灵敏度GPS接收机载波跟踪环路的设计优化与实现[J]. 北京大学学报, 2011, 47(5): 783-788.
- [4] GERNOT C. Development of combined GPS L1/L2 C acquisition and tracking methods for weak signals environments[D]. Calgary, Canada: University of Calgary, 2009.
- [5] 陈景霞, 李建文. 新的全球定位系统弱信号高灵敏捕获算法[J]. 计算机应用, 2012, 32(11): 3262-3267.
- [6] BORIO D, O'DRISCOLL C, LACHAPPELLE G. Coherent, noncoherent and differentially coherent combining techniques for acquisition of new composite GNSS signals[J]. IEEE Transactions on Aerospace and Electronic Systems, 2009, 45(3): 1227-1240.
- [7] 黎山, 易清明, 陈庆, 等. 适用于GPS软件接收机的弱信号捕获方法[J]. 计算机应用, 2012, 32(3): 816-818.
- [8] 祝雪芬, 陈熙源, 涂刚毅, 等. GNSS软件接收机新型高灵敏度捕获算法及实现[J]. 系统工程与电子技术, 2009, 31(9): 2055-