

一种面向节能的虚拟机在线迁移解决方案

赵 丹¹ 沈苏彬² 吴振宇¹

(1.南京邮电大学 物联网学院 江苏 南京 210003;

2.南京邮电大学 计算机学院 江苏 南京 210003)

摘 要: 在云计算环境中, 针对由于负载的动态变化导致某些节点负载过低, 造成资源闲置和能源消耗的问题, 提出一种面向节能的虚拟机在线迁移解决方案。综合考虑多种性能参数, 提出一种节点负载率的计算方式, 并通过云平台实时监测各个节点的负载情况。按照以节能为目标的资源调度规则, 选择需要迁移的虚拟机, 并为所有待迁移的虚拟机匹配合适的节点。通过基于内存页混合拷贝方式的虚拟机在线迁移技术, 将所有待迁移的虚拟机迁移到其对应的目标节点上, 并进一步降低迁移过程对服务质量的影响。迁移完成之后, 关闭零负载节点, 从而达到节能的目的。实验结果表明, 该方案可以实现虚拟机迁移过程中的透明服务, 并大大降低能耗, 实现节能的优化目标。

关键词: 节能; 混合拷贝; 在线迁移; OpenStack

中图分类号: TP39

文献标识码: A

文章编号: 1673-629X(2018)02-0078-05

doi: 10.3969/j.issn.1673-629X.2018.02.018

A Live Migration Solution of Virtual Machine for Energy-saving

ZHAO Dan¹ SHEN Su-bin² WU Zhen-yu¹

(1.School of IoT, Nanjing University of Posts and Telecommunications, Nanjing 210003, China;

2.School of Computer, Nanjing University of Posts and Telecommunications, Nanjing 210003, China)

Abstract: In the cloud computing environment, so as to solve the problem of idle resources and energy consumption due to some nodes with low workload caused by dynamic change of workload, we put forward an energy-efficient live migration solution of virtual machine. Considering a variety of performance parameters, we give a method of calculating the node load rate and realize the real-time monitoring of the load to each node through a cloud platform. The virtual machines are selected and matched with the appropriate nodes under the constraint of energy-saving resource scheduling rules. And all the virtual machines needed to migrate are migrated to the corresponding nodes, and the impact of migration on the quality of service through the virtual machine hybrid-copy live migration mode is further reduced. When migration completed, the zero-load nodes are closed, achieving the purpose of energy saving. Experiment shows that the proposed solution can achieve a transparent service in the virtual machine migration, extremely reducing energy consumption, achieving the aim of energy-saving.

Key words: energy-saving; hybrid-copy; live migration; OpenStack

0 引 言

云计算以按需付费、动态扩展等优势极大地方便了各用户, 在满足客户业务需求的同时, 大大降低了他们的使用成本。然而, 出于对温度等环境因素的需求, 云计算中心每年要消耗大量的能源。据 IDC 统计, 2014 年我国云计算中心整体耗电量占社会总用电的 1.6%, 并预测 10 年后, 云计算中心整体的耗电量大约要占 15%^[1]。云计算中心的能耗主要来源于 IT 设备、

照明系统、空调系统、供配电系统, 而在所有因素中, IT 设备的耗电量居首, 约占 45%, 而空调系统次之, 约占 40%^[2]。由此可见, 云计算中心的节能措施首先是降低 IT 设备的能耗。

在云计算中心, 针对 IT 设备的节能技术分为静态节能技术和动态节能技术。静态节能是指在系统设计之初就将能耗因素考虑进来, 在电路层采用节能技术, 如停用未使用的电路组件, 对 CPU、RAM 和磁盘等主

收稿日期: 2017-03-13

修回日期: 2017-07-26

网络出版时间: 2017-10-19

基金项目: 国家自然科学基金(61502246); 江苏省科技计划项目(未来网络前瞻性研究项目)(BY20130951108)

作者简介: 赵 丹(1990-), 女, 硕士研究生, 研究方向为云计算; 沈苏彬, 博士生导师, CCF 会员(E200005482S), 研究方向为计算网络、下一代电信网及网络安全。

网络出版地址: <http://kns.cnki.net/kcms/detail/61.1450.TP.20171019.1626.080.html>

要耗电组件采用电压频率动态调整技术。动态节能是从资源管理角度出发,根据运行云任务的负载的动态变化,自适应地对云平台的能耗进行管理。目前,云计算中心的能耗优化管理技术主要有电压动态调整技术、关闭/休眠技术和虚拟化技术^[3]。在云计算架构的基础设施即服务(IaaS)层,云平台使用KVM、VMware等虚拟化技术对计算、存储等资源进行扩展,以提高资源利用率,并将资源进行封装,以虚拟机的形式向用户提供服务^[4]。因此,虚拟机可作为云计算的基本服务单元。在节能措施上,可通过虚拟机的迁移将负载重新整合,减少运行节点的数量,从而达到节能的目的。

几个主流的云计算平台,如OpenStack^[5]、Open-Nebula^[6]不支持资源的动态调配,作为业界成功应用IaaS商业模式的代表AWS,其弹性云计算EC2中集成了弹性负载均衡(ELB)功能,提供基于动态配置和动态迁移的资源调配,却没有提供针对节能目标的调度策略^[7]。而在以节能为目标的实现方式上,使用KVM、Xen等作为底层虚拟机管理程序的云计算平台,使用基于内存页预拷贝方式的虚拟机在线迁移技术,该技术存在传输速率以及页重传问题。针对这些问题,有人提出了基于内存页后拷贝方式的虚拟机在线迁移技术,极大地改进了预拷贝方式在停机时间上的不足,但存在由于内存页丢失而导致的虚拟机迁移失败问题。为了保证在虚拟机迁移过程中的鲁棒性,需要一种更为高效可靠的迁移方式。

基于以上问题,文中分析了针对云计算节能的相关工作,设计了一种节能方案,并通过实验对其进行验证分析。

1 相关工作分析

在IaaS层,目前常用的节能措施是通过针对节能的资源调度策略为虚拟机重新匹配物理主机,并通过虚拟机在线迁移技术将虚拟机迁移到对应的物理主机上,提高单个节点的资源使用率,减少节点的运行数量。

在面向节能的资源调度策略中,文献[8]通过将效率控制器EC、服务器管理器SM、附件管理器EM、群组管理器GM和虚拟机控制器VMC五种不同的能源管理策略相结合来研究云计算中心的能源管理问题,并在协调控制器上对控制理论和反馈控制循环进行了深入研究。这种方式不受工作负载类型的约束,但只针对CPU资源,并未考虑系统中的其他资源,如RAM、磁盘等。文献[9]综合考虑了多种资源(CPU、磁盘、网络)的使用情况,将负载集中于少数物理节点,关闭闲置节点,以达到节能的目的。该方案指定能源或性能的开销值,以保证负载的整合不会对应用的

性能产生影响。然而该方案并没有考虑如何将虚拟机的迁移数量降到最小,从而节省迁移总时间。

在面向节能的资源调度完成之后,需要通过虚拟机迁移实现节能目标。Clark等^[10]提出了基于内存页预拷贝方式的虚拟机在线迁移技术,将虚拟机在线迁移的过程视为两个主机的透明交互,将虚拟机的在线迁移过程细分为六个阶段:预迁移阶段、保留阶段、迭代预拷贝阶段、停机拷贝阶段、应答阶段和激活阶段。通过这六个阶段将虚拟机从源主机迁移到目的主机上。由虚拟机在线迁移算法优劣的评判标准^[11]可知,内存页预拷贝方式存在着传输速率问题和页重传问题。针对这些问题,Hines等^[12]提出一种基于内存页后拷贝方式的虚拟机在线迁移技术,该技术直接在源主机上挂起虚拟机,然后将虚拟机的CPU(vCPU)状态拷贝到目的主机,大大降低了采用预拷贝方式产生的停机时间,最后通过按需拷贝方式按照需求从源主机获取内存页。

2 云计算节能方案设计

针对云计算能耗高的问题,文中提出一种面向节能的虚拟机在线迁移技术,综合多种性能参数衡量节点的资源使用情况,而不仅仅是文献[8]中所考虑的CPU参数,并在资源调度策略中考虑虚拟机的迁移数量因素,尽可能减少虚拟机的迁移数量,降低虚拟机迁移对服务质量的影响。最后通过一种更高效可靠的基于内存页混合拷贝方式的虚拟机在线迁移技术完成虚拟机的迁移。待迁移工作完成后,关闭空闲节点,从而优化整个云计算中心的节能目标。

从上述分析可知,该方案主要完成三个功能:

(1) 资源监测:根据应用需求,灵活监测云计算中心节点的资源使用率。

(2) 资源调度:根据节点的资源使用情况,为所有需要迁移的虚拟机匹配最佳节点。

(3) 虚拟机迁移:根据资源的匹配结果,通过一种更高效可靠的虚拟机迁移技术,将虚拟机迁移到目标节点上。

2.1 节点资源的监测

节点资源的监测负责周期性监测云计算中心所有运行节点的负载率,定量分析各节点的资源使用情况。

为根据应用需求灵活反映节点的负载情况,节点负载率应为多种负载使用率的加权综合值。针对节能需求,文中的性能参数有CPU、内存、磁盘、带宽以及温度,因此节点的负载性能参数可用一个集合 $L = \{L_1, L_2, \dots, L_5\}$ 表示。其中 $L_i (1 \leq i \leq 5)$ 分别表示vCPU核数、RAM、磁盘的使用率、带宽以及温度/100。

假设云计算中心有 m 个节点,则节点 $N_j (1 \leq j \leq$

m) 的负载率为:

$$\text{load_ratio}(N_j) = \sum_{i=1}^5 (\alpha_i * L_i) \quad (1)$$

其中, α_i 为性能参数 L_i 所对应的权重系数, 且 $\sum \alpha_i = 1$ 。

可以根据应用需求对权重系数进行调整, 如对计算性能要求更高, 可以增加 CPU 和 RAM 使用率的权重系数, 如果应用偏重于数据的存储, 可以提高磁盘使用率的权重系数。

2.2 资源调度

资源调度负责根据资源调度的约束条件, 选择需要迁移的虚拟机, 并为它们匹配最佳的节点。

为实现节能目标, 优化资源调度的时间损耗, 对待迁移的虚拟机的选择和目标节点的处理做如下约束:

(1) 为达到节能目的, 设定负载率迁出阈值, 低于该阈值的节点上的所有虚拟机都需要迁移。

(2) 为减少虚拟机和节点匹配的计算量, 设定负载率迁入阈值。在虚拟机迁入之后节点负载率不能低于负载率迁入阈值。在此阈值之上, 表明该节点满负载, 不再与其他虚拟机匹配。

在上述条件约束下, 根据设定的负载率迁出阈值, 将云计算中心所有运行的节点划分为正常负载节点和欠载节点, 所有欠载节点上的虚拟机均为需要迁移的虚拟机, 并将这些虚拟机添加到迁移列表中。

为降低占用资源多的虚拟机迁移的概率, 从而减少虚拟机迁移的数量以及所有虚拟机迁移的总时间, 按照虚拟机迁移时间的大小, 对所有待迁移的虚拟机作升序排序。在虚拟机迁移时间的计算上, 因虚拟机在线迁移传输的是该虚拟机的 RAM 和 CPU 的数据, 因此虚拟机迁移时间为该虚拟机占用的 RAM 和 CPU 的资源与带宽的比值。

在确定虚拟机迁移的顺序后, 按序为每个需要迁移的虚拟机匹配目标节点。在节点的选择顺序上, 首先考虑正常负载节点集合中的资源, 若正常负载节点集合无节点可用, 且迁移列表未空, 则选择欠载节点集合中的资源。在欠载节点集合中节点的选择顺序上, 倒序选择迁移列表中的元素所在的节点, 以此减少虚拟机迁移的数量以及所有虚拟机迁移的总时间。在目标节点的匹配上, 通过计算待迁移虚拟机和所有符合条件的节点的成本值^[13], 定量分析虚拟机和各节点的匹配程度, 并以最小成本值为原则, 为虚拟机匹配最佳的目标节点。虚拟机 VM_k ($1 \leq k \leq n$, k 为待迁移虚拟机的个数) 与节点 N_j 的成本值计算公式如下所示:

$$\text{total_cost}(VM_k, N_j) = \min\{\text{solo_cost}(S_i)\} \quad (2)$$

其中

$$\text{solo_cost}(S_i) = \frac{\text{left_}S_i - \text{request_}S_i}{\text{total_}S_i} \quad (3)$$

其中, $\text{left_}S_i$ 为节点上剩余的 S_i 资源; $\text{request_}S_i$ 为虚拟机所请求的 S_i 资源; $\text{total_}S_i$ 为节点上所有的 S_i 资源。

2.3 虚拟机迁移

基于文献[11]提出的虚拟机迁移原则, 综合考虑现有迁移方式中存在的问题, 提出一种更高效可靠的虚拟机在线迁移方式——基于内存页混合拷贝方式的虚拟机在线迁移技术。该技术借鉴了内存页预拷贝方式中的迭代拷贝思想, 并将其应用于虚拟机内存页后拷贝方式的起始阶段, 从而降低虚拟机因缺页导致的在目标节点恢复失败的风险, 同时虚拟机在迁移过程的按需拷贝阶段可首先查询本地数据, 从而减少迁移总时间, 进一步提高服务性能。迁移流程如图 1 所示。

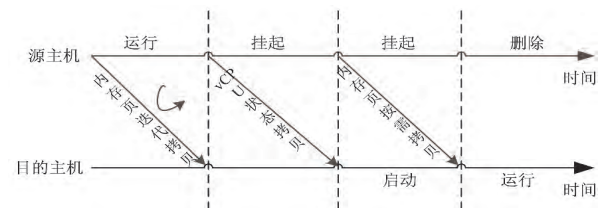


图 1 虚拟机混合拷贝在线迁移流程

如图 1 所示, 该迁移流程分为三个阶段: 内存页迭代拷贝阶段、vCPU 状态拷贝阶段和内存页按需拷贝阶段。在内存页迭代拷贝阶段, 将某一时刻虚拟机的内存数据保存下来, 并通过迭代拷贝方式传输到目的主机。vCPU 状态拷贝阶段在虚拟机所占用的内存数据拷贝完成后, 挂起虚拟机, 并将此刻该虚拟机的 vCPU 状态拷贝到目的主机中, 在目的主机中启动该虚拟机。在内存页按需拷贝阶段, 优先从本地获取数据, 如果该数据所在页面被标记为“脏页”或者查找不到该数据, 则向源主机发送缺页请求。而在源主机上, 启动计时器, 计算等待请求的时间, 每收到一次请求就更新一次计时器。如果在设定时间内没有收到目的主机发送过来的请求, 则表明虚拟机在目的主机上恢复运行, 终止源主机上的虚拟机。

3 基于 OpenStack 的节能实现

基于 OpenStack 平台, 通过扩展其计算服务 Nova^[14]和虚拟机管理软件 Qemu 实现节能目标。首先, 利用资源调度功能生成迁移列表 $\text{vm_migrating_table}$ 和匹配列表 mapping_table 。然后, 通过虚拟机迁移技术将每一个待迁移的虚拟机按序迁移到其对应的目标节点上。在迁移技术的实现上, 首先扩展 Qemu 实现基于混合拷贝技术的单个虚拟机的迁移, 然后通过编写脚本实现多个虚拟机的自动迁移。

通过脚本文件 migration.sh 实现多个虚拟机的迁移。该脚本的功能是首先配置环境变量,获取管理员权限,调用迁移功能;然后读取迁移列表和匹配列表,遍历迁移列表,根据匹配列表的匹配结果调用迁移命令 nova live-migration 完成迁移工作。脚本实现代码如下:

```
#!/bin/bash
#配置环境变量
echo "source admin-openrc.sh"
#遍历迁移列表 vm_migrating_table 中所有 #虚拟机
for key in `cat /vm_migrating_table`
do
#调用 nova live-migration 命令根据匹配#列表 mapping_table 迁移所有虚拟机
echo "nova live-migration $key `cat /mapping_table |grep key`"
done
```

nova live-migration 命令通过调用基于内存页混合拷贝方式的虚拟机在线迁移技术完成迁移工作。该技术通过 Qemu 的自定义模块 myMigration 实现。该技术的实现分为三个阶段:内存页迭代预拷贝阶段、vCPU 状态拷贝阶段和内存页按需拷贝阶段。假设通过资源的调度为 host₁ 上的虚拟机 vm₁ 匹配节点 host₂, 迁移步骤如下:

阶段 1: 迭代预拷贝内存页。首先在源主机 host₁ 上,启动 Qemu 中的 socket 接口,监听目的主机 host₂, 等待 host₂ 连接;调用迁移线程函数 migration_thread() 完成迭代预拷贝功能。该函数的具体功能是:保存虚拟机 vm₁ 的状态,确保虚拟机 vm₁ 在迁移失败之后还可以在源主机上继续运行;将 vm₁ 的状态改为“migrating”;获取 vm₁ 所占用的内存数据大小;保存当前所有内存页数据;设定传输速率,通过 socket 接口将所有数据迭代拷贝到目的主机。在目的主机 host₂ 上,启动 Qemu 中的 socket 接口,调用 qemu_loadvm_state() 函数保存接收到的所有内存页。

阶段 2: 拷贝 vCPU 数据。首先,在源主机 host₁ 上,所有内存页传输完毕,将虚拟机 vm₁ 的状态改为“suspend”;调用 qemu_savevm_state_complete() 函数同步 CPU 所有信息(包括寄存器数据以及缓存数据)。然后,在目的主机 host₂ 上,通过 socket 接收数据,并存储在 /var/lib/nova/instances 目录下。等待所有数据传输完毕,启动虚拟机。

阶段 3: 按需拷贝内存页。在目的主机 host₂ 上,如果所用数据在内存页中找到且未被更新过,就继续运行。如果找不到或已被更新,就向源主机发送请求,请求所用数据的相应内存页,接收数据,并通过逆 xbzrl 算法(xbzrl 算法的反过程)恢复数据,并继续运行。

在源主机 host₁ 上,等待接收虚拟机的请求信息,如果接收到请求信息,就将该信息所占用的内存页数据与备份数据对比,通过 xbzrl 算法压缩数据再传输到 host₂ 上;如果超过设定的时间没有接收到请求,表明迁移完成,将虚拟机终止并删除中间过程中产生的信息。

4 实验及结果分析

实验基于 Centos7 操作系统,搭建 OpenStack Kilo 版本五节点平台,并在该平台上进行验证。根据官方文档依照基本环境配置、Keystone 服务安装和配置、Glance 服务安装和配置、Nova 服务安装和配置、Cinder 服务安装和配置的流程搭建完成 OpenStack 平台^[15]。为实现迁移功能,采用 NFS 软件实现共享存储服务^[13]。在该系统中,OpenStack 平台由一个控制节点、一个网络节点和三个计算节点组成,各节点硬件环境如表 1 所示。

表 1 OpenStack 平台硬件环境

节点名称	CPU 型号	内存大小/GB	磁盘空间/GB
控制节点	Intel(R) Xeon(R) E5420@ 2.50 GHz	8	750
网络节点	Intel(R) Core (TM) 2 6400@ 2.13 GHz	2	250
计算节点 1	Intel(R) Xeon (R) E5310@ 1.60 GHz	4	300
计算节点 2	Intel(R) Xeon(R) E5310@ 1.60 GHz	4	600
计算节点 3	Intel(R) Xeon(R) E5310@ 1.60 GHz	4	150

在平台部署完成之后,首先在控制节点上,在 Nova Scheduler 的过滤模块 filters 扩展一个 vmMigrating-Filter 子模块,计算节点负载率并完成待迁移虚拟机的选择功能,然后在 Nova Scheduler 的权衡模块 weights 中扩展一个 myMetrics 子模块,为每一个待迁移的虚拟机匹配目标节点,并完成能耗计算。扩展完成后,将这两个模块在配置文件 nova.conf 中做相应配置^[13],启用这两个子模块。在根目录下添加脚本文件 migration.sh 实现多个虚拟机的迁移,然后在各个计算节点的 Qemu 中添加一个自定义模块 myMigration,实现虚拟机的混合拷贝在线迁移技术。

为了验证该方案的性能,进行了如下测试。

(1) 能效测试。

重启 Nova 服务,指定多组{负载率迁出阈值,负载率迁入阈值},发送节能请求,完成测试过程,计算能效^[16],并查看迁移后虚拟机的状态信息,从而计算虚拟机迁移失败率。根据这些权限值测得的数据如表 2 所示。

表 2 基于 OpenStack 的能效相关统计信息 %

负载率 迁出阈值	负载率 迁入阈值	虚拟机迁 移失败率	节能效率
50	93	4.6	30.5
45	93	3.4	29.9
50	88	1.8	24.3
45	88	1.1	23.9

分析表 2 中的数据可知,负载率迁出阈值越高,虚拟机迁移失败率就越高,而能效基本保持不变。这是因为负载率迁出阈值越高,待迁移的虚拟机数目越大,正常负载集合可用资源变少,致使虚拟机迁移失败率增高;由于负载率迁入阈值不变,资源总量基本不变,所以能效基本保持不变。负载率迁入阈值越高,虚拟机迁移失败率越高,且能效越高。这是由于负载率迁入阈值越高,节点的资源使用率越高,因而能效越高,但由于满负载区间(负载率为 100% 和负载率迁入阈值的差值)变小,虚拟机迁移失败率会增高。因此可以得出,在 OpenStack 环境中,为达到节能目标,必定会带来虚拟机迁移失败问题,即二者不可兼得;此外,为提高能效,可尽量提高负载率迁入阈值。

(2) 虚拟机在线迁移技术测试。

使用 Nova 服务原有的在线迁移命令 nova live-migration <server><destination host>对四种大小的虚拟机进行迁移性能测试,其中 server 为待迁移的虚拟机,destination host 为与该虚拟机对应的目的主机。首先使用 OpenStack 已有的迁移算法测试,其中 Qemu 版本为 Qemu-2.0.0,然后使用 Qemu 中自定义模块 myMigration 测得迁移信息。虚拟机两种在线迁移方式的停机时间以及迁移总时间对比如图 2 和图 3 所示。

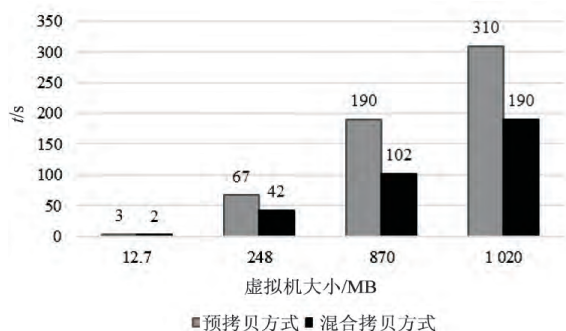


图 2 虚拟机在线迁移停机时间对比

通过图 2 和图 3 可以看出,对于停机时间参数,随着虚拟机越来越大,混合拷贝方式的停机时间低于预拷贝方式。这是由于混合拷贝方式在停机拷贝阶段只传输该虚拟机的 CPU 信息,而预拷贝方式在停机拷贝阶段传输的不仅有 CPU 信息还有内存页数据。在迁移总时间方面,内存页混合拷贝方式的迁移总时间较

预拷贝方式也明显降低。这是因为预拷贝方式的迭代传输会造成内存页冗余较大,而混合拷贝方式在按需拷贝阶段只需传输一次内存页即可。实验结果证明了这一理论,说明该方案可以降低虚拟机在线迁移过程中的停机时间,同时也大大降低了迁移总时间,从而提高了服务性能。

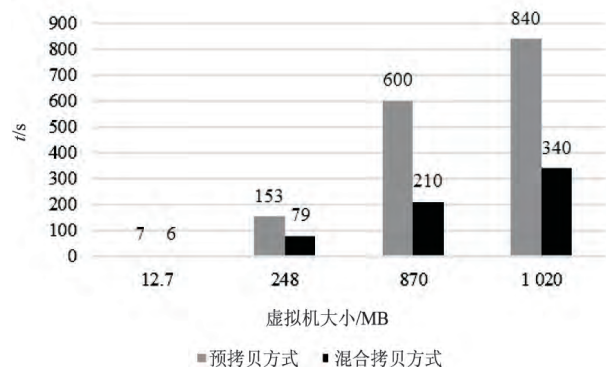


图 3 虚拟机在线迁移总时间对比

5 结束语

在云计算环境中,针对由于其基本服务单元虚拟机的动态变化造成某些节点负载过低而导致资源闲置和能源消耗的问题,提出一种面向节能的虚拟机在线迁移解决方案。通过虚拟机整合,提高单个节点的资源使用率,减少运行节点的数量,关闭零负载节点,从而达到节能的目的。该方案通过节点资源的监测、资源的调度和虚拟机的迁移技术,综合衡量节点的资源使用率,选择待迁移的虚拟机、为待迁移虚拟机匹配最佳的节点,实现高效可靠的基于内存页混合拷贝方式的虚拟机在线迁移技术,并在 OpenStack 平台上进行了实现。实验结果表明,该方案可以进一步提高虚拟机迁移过程中的服务质量,有效降低能耗,实现了节能的优化目标。

参考文献:

- [1] 中国云计算数据中心节能现状深度分析[EB/OL].2016.
<http://tech.idcquan.com/pro/102749.shtml>.
- [2] 张慧珍,李长春,周培琴,等.云计算数据中心节能技术研究[J].信息系统工程,2013(10):79-80.
- [3] 谭一鸣,曾国荪,王伟.随机任务在云计算平台中能耗的优化管理方法[J].软件学报,2012,23(2):266-278.
- [4] CORRADI A, FANELLI M, FOSCHINI L. VM consolidation: a real case based on OpenStack cloud[J].Future Generation Computer Systems, 2014, 32: 118-127.
- [5] 陈 恂.OpenStack 平台的虚拟机调度关键技术研究[D].杭州:杭州电子科技大学,2014.
- [6] OpenNebula. The cloud data center management solution[EB/OL].2016.<https://www.opennebula.org/>.

(下转第 87 页)

- [8] BOBADILLA J ,ORTEGA F ,HERNANDO A ,et al.Recommender systems survey [J]. Knowledge - Based Systems , 2013 ,46(1) : 109-132.
- [9] LEMPEL R ,KOREN Y.On bootstrapping recommender systems[C]//Proceedings of ACM international conference on information & knowledge management. New York: ACM , 2010: 1805-1808.
- [10] 景民昌,张 芹,唐第官.协同过滤系统中基于种子集评分的新用户冷启动推荐研究[J].图书情报工作,2013,57(5): 124-128.
- [11] RASHID A M ,ALBERT I ,COSLEY D ,et al.Getting to know you: learning new user preferences in recommender systems[C]//Proceedings of the 7th international conference on intelligent user interfaces. New York: ACM ,2002: 127-134.
- [12] 杨秀梅,孙 咏,王美吉,等.新闻推荐系统中用户冷启动问题的研究[J].小型微型计算机系统,2016,37(3): 479-482.
- [13] 杨 阳,向 阳,熊 磊.基于矩阵分解与用户近邻模型的协同过滤推荐算法[J].计算机应用,2012,32(2): 395-398.
- [14] 张 川.基于矩阵分解的协同过滤推荐算法研究[D].长春: 吉林大学,2013.
- [15] SAVESKI M ,MANTRACH A.Item cold-start recommendations: learning local collective embeddings [C]//Proceedings of the 8th ACM conference on recommender systems. New York: ACM ,2014: 89-96.
- [16] LIU H S ,GOYAL A ,WALKER T ,et al.Improving the discriminative power of inferred content information using segmented virtual profile [C]//Proceedings of the 8th ACM conference on recommender systems. New York: ACM , 2014: 97-104.
- [17] 陈克寒,韩盼盼,吴 健.基于用户聚类的异构社交网络推荐算法[J].计算机学报,2013,36(2): 349-359.
- [18] LING G ,LYU M R ,KING I.Ratings meet reviews ,a combined approach to recommend [C]//Proceedings of the 8th ACM conference on recommender systems. New York: ACM ,2014: 105-112.
- [19] 杨兴耀,于 炯,吐尔根·依布拉音,等.融合奇异性和扩散过程的协同过滤模型[J].软件学报,2013,24(8): 1868-1884.
- [20] SEDHAIN S ,SANNER S ,BRAZIUNAS D ,et al.Social collaborative filtering for cold-start recommendations [C]//Proceedings of the 8th ACM conference on recommender systems. New York: ACM ,2014: 345-348.
- [21] TREVISIOL M ,AIELLO L M ,SCHIFANELLA R ,et al.Cold-start news recommendation with domain-dependent browse graph [C]//Proceedings of the 8th ACM conference on recommender systems. New York: ACM ,2014: 81-88.
- [22] ADOMAVICIUS G ,TUZHILIN A.Context-aware recommender systems [C]//Proceedings of the 8th ACM conference on recommender systems. New York: ACM ,2014: 335-336.
- [23] 景民昌.从 ACM RecSys 2014 国际会议看推荐系统的热点和发展[J].现代情报,2015,35(4): 41-45.
- [24] TANG Liang ,JIANG Yexi ,LI Lei ,et al.Ensemble contextual bandits for personalized recommendation [C]//Proceedings of the 8th ACM conference on recommender systems. New York: ACM ,2014: 73-80.
- [25] HARIRI N ,MOBASHER B ,BURKE R.Context adaptation in interactive recommender systems[C]//Proceedings of the 8th ACM conference on recommender systems. New York: ACM ,2014: 40-48.

(上接第82页)

- [7] 王 旭.基于能耗约束的虚拟机调度策略研究[D].昆明: 云南大学,2011.
- [8] RAGHAVENDRA R ,RANGANATHAN P ,TALWAR V ,et al.No power struggles: coordinated multi-level power management for the data center [J].ACM SIGARCH Computer Architecture News ,2008,36(1): 48-59.
- [9] PINHEIRO E ,BIANCHINI R ,CARRERA E V ,et al.Load balancing and unbalancing for power and performance in cluster-based systems[C]//Workshop on compilers and operating systems for low power. [s.l.]: [s.n.],2003: 182-195.
- [10] CLARK C ,FRASER K ,HAND S ,et al.Live migration of virtual machines[C]//Proceedings of the 2nd conference on symposium on networked systems design & implementation. Berkeley ,CA ,USA: USENIX Association ,2005: 273-286.
- [11] SVÄRD P ,HUDZIA B ,WALSH S ,et al.Principles and performance characteristics of algorithms for live VM migration [J].ACM SIGOPS Operating Systems Review ,2015,49(1): 142-155.
- [12] HINES M R ,GOPALAN K.Post-copy based live virtual machine migration using adaptive pre-paging and dynamic self-ballooning [C]//Proceedings of the 2009 ACM SIGPLAN/ SIGOPS international conference on virtual execution environments. Washington ,DC: ACM ,2009: 51-60.
- [13] 陈伯龙,程志鹏.云计算与 OpenStack(虚拟机 Nova 篇) [M].北京: 电子工业出版社,2013.
- [14] 戴 友.OpenStack 开源云-王者归来[M].北京: 清华大学出版社,2014.
- [15] OpenStack. Install guides [EB/OL].2016. <http://docs.openstack.org/kilo/>.
- [16] BELOGLAZOV A ,BUYA 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. New York: ACM ,2010: 1-6.