

基于阈值滑动窗口机制的虚拟机迁移判决算法

曲晓雅 刘 真

(北京交通大学计算机与信息技术学院 北京 100044)

摘 要 数据中心是云计算中数据运算、交换、存储的中心。近年来以虚拟机为粒度的虚拟机放置管理成为云数据中心能耗管理、实现动态可伸缩资源提供的重要支撑技术。在虚拟机放置的动态管理阶段,虚拟机迁移触发机制主要是根据物理主机中资源利用率的变化情况,决定是否需要将虚拟机迁移到其它主机。迁移时机判决准确能够有效地平衡过热点并关掉过冷点。当前的迁移时机缺乏对整个数据中心负载变化行为趋势的反映,也因为静态的阈值设定容易发生频繁的迁移,造成不必要的迁移代价和传输开销。提出了基于阈值滑动窗口机制的虚拟机迁移判决算法(iWnd),其能够根据整个数据中心任务量的多少动态调整高低阈值间窗口的大小,减少了任务量满负荷时期需要迁移虚拟机的数量,从而避免不必要的迁移开销和传输代价,有效地实现节能。在云计算平台 Cloudsim 上进行了仿真实验。结果表明,提出的 iWnd 算法在减少虚拟机迁移数量、降低迁移失败率上有良好的效果,同时并未产生过多额外的功耗。

关键词 虚拟机放置,迁移时机,滑动窗口,迁移失败率

中图分类号 TP393 **文献标识码** A **DOI** 10.11896/j.issn.1002-137X.2016.4.013

Dynamic Threshold Window Algorithm for Virtual Machines Migration Decision

QU Xiao-ya LIU Zhen

(School of Computer and Information Technology, Beijing Jiaotong University, Beijing 100044, China)

Abstract Cloud data center(DC) is the center of data operation, exchange and storage. Based on virtualization technology, virtual machine(VM) placement has become an important technology for power management and elastic resource provision. In the stage of dynamic VMs management, with the changes of resources utilization, migration trigger mechanism will determine when to migrate the VMs from one host to the other. The accurate judgment of trigger time can balance the hot spots effectively and turn off cold spots in DC. However, current migration trigger mechanism lacks the response to the changes of DC workloads, and static threshold setting is easy to cause frequent migration with unnecessary migration and transmission cost. To solve these problems, a dynamic threshold setting algorithm, iWnd was proposed, which adjusts the size of threshold windows according to the workloads on the whole DC. In addition, iWnd reduces the number of VMs which need to be migrated, avoiding unnecessary migration and transmission cost and saving power. We made the experiments on a simulated cloud environment using Cloudsim toolkit. Our efforts show that iWnd can effectively reduce the number of VMs migration and migration failure rate without producing additional power consumption.

Keywords VM placement, Trigger time, Threshold windows, Migration failure rate

1 引言

数据中心(Data Center, DC)资源管理是当前云计算技术的研究热点,它将大量的计算资源、存储资源与软件资源链接在一起,形成了巨大规模的共享资源池,是数据运算、交换、存储的中心。随着当前云计算相关技术的发展,资源管理动态化、弹性化和自动化需求更加突出。实现资源的按需动态伸缩对于云数据中心的可用性是至关重要的^[1,2]。然而有资料显示,当前全球多数的云计算数据中心的资源利用率在15%~20%之间,服务器在一天中有很大大一部分时间处于闲置^[3]。

得益于技术进步和硬件支持,虚拟化技术再一次兴起,集结了基于进程粒度和基于服务器粒度平台的优势,以虚拟机(Virtual Machine, VM)为粒度能够实现对资源的快速部署和重部署,提高了资源利用率。虚拟机的放置管理成为当前的研究热点^[4-6]。

虚拟机放置管理分为初始化放置和动态放置。虚拟机的初始化放置管理是研究如何在一个没有负载的云数据中心根据虚拟机的类型和对资源的请求选择合适的目的物理机(Physical Machine, PM)的过程,类似于 NP-hard 的装箱问题。初始化放置具有长期效应,对数据中心资源的有效利用、

到稿日期:2015-04-07 返修日期:2015-08-07 本文受国家自然科学基金(61202429),中央高校基本科研业务费专项资金(2015JBM042)资助。
曲晓雅(1990—),女,硕士,主要研究方向为云计算与虚拟化,E-mail:13120420@bjtu.edu.cn,刘真(1977—),女,博士,副教授,主要研究方向为云计算与虚拟化、数据中心虚拟机资源管理和优化,E-mail:zhliu@bjtu.edu.cn(通信作者)。

能耗节约起到重要作用^[7]。虚拟机的动态放置管理是指由于云数据中心负载的动态变化所引起的虚拟机重新放置的过程。在虚拟机的动态放置阶段,需要考虑3个方面,即3W(When、Which和Where)的问题:对资源监控决定何时进行迁移(Migration Trigger),哪些虚拟机需要被迁移(VM Selection)和决定迁移到哪些物理机上(Destination Selection)^[7]。

在虚拟机迁移时机的判决中,当前研究往往基于单台物理机的负载情况,没有考虑数据中心总体负载量的大小,导致整个数据中心超载时虚拟机由于找不到目的主机而迁移失败。同时,当前的迁移时机缺乏反映随应用负载变化的行为趋势,也因为静态的阈值设定容易发生频繁的迁移造成不必要的迁移代价和传输开销。

本文针对虚拟机动态管理阶段迁移时机的问题,提出一种新的基于阈值滑动窗口机制的虚拟机迁移判决算法 iWnd。与现有方法不同, iWnd 考虑了数据中心负载不断变化的情况,高低阈值间的窗口能够随着数据中心整体负载量的变化而调整。本文建立了数据中心任务量的预测机制,当预测的数据中心任务量接近满负荷时,将阈值窗口调大,以避免虚拟机不必要的迁移,从而降低迁移失败率,减少传输代价。只有当阈值窗口调整之后,当前负载的资源利用率仍不在阈值窗口内时,才会触发虚拟机的迁移。

本文第2节介绍相关工作;第3节介绍阈值窗口滑动下虚拟机迁移时机的问题模型,并提出了基于阈值滑动窗口机制的虚拟机迁移时机判决算法;第4节主要对提出的算法进行了实验验证,首先通过实验对比不同数量级的物理机在此迁移判决算法下的耗电情况,并分别与静态阈值迁移判决机制和动态阈值迁移判决机制比较其迁移数量和迁移代价,最后验证不同阈值变化幅度对能耗的影响;最后总结全文并提出进一步的工作。

2 相关工作

国内外针对虚拟机迁移时机的研究主要分为两个方面:基于阈值的虚拟机迁移时机和基于预测模型的虚拟机迁移时机。

基于阈值的方法又分为两种:静态阈值方式^[8-10]和动态自适应阈值方式^[11-13]。Zhu等人^[8]提出的是一个静态单阈值触发机制,即当CPU利用率大于85%时判定物理机超载,触发物理机上的虚拟机迁移。单阈值的设置没有考虑到装载不足所带来的资源浪费和能源消耗,达不到最优化节能。文献^[9]在此基础上提出基于CPU利用率的双阈值触发机制,并将高低阈值分别设为45%和81%,当物理机负载的CPU利用率不在此范围内时,则触发迁移。文献^[10]同样提出了二维单阈值触发机制,二维单阈值同时报警和触发迁移。二维单阈值分别代表CPU温度以及多维资源(CPU、输入/输出、网络带宽)的利用率。触发迁移的判决条件是:在一定时间间隔内,一定数量的物理机(例如80%)资源(CPU、输入/输出、网络带宽)利用率和CPU温度都达到或超出各自阈值,则触发迁移。二维单阈值的触发机制避免了瞬时峰值导致的不必要的虚拟机迁移。

上述研究所设定的静态域值无法适应负载的动态变化特征,针对单台物理机设定阈值也无法合理有效地适应整个数据中心的负载量变化情况。

文献^[11]对于不断变化的负载,提出了动态阈值的触发机制来决定虚拟机何时迁移,并且考虑了多个资源维度(CPU、内存、网络带宽)来定义主机的负载量。作者利用为各个资源维度设置权重系数的方式,来定义当前物理机的综合利用率。依据不同时刻的负载的动态变化,高低阈值的调整是通过乘以一个经验系数来确定。通过动态比较均衡算法(DCABA)比较当前负载与高低阈值来判断该虚拟机是否要被迁移。该方法有效地减少了总体耗能,优化了数据中心的耗能情况。然而,通过经验系数来调整高低阈值的幅度存在一定的误差。

Sinha等人^[12]对于动态变化的工作负载提出了动态高低阈值判断机制,在高低阈值的设置上考虑了当前物理机的CPU利用率、内存利用率和网络带宽利用率,阈值随着当前各资源利用率的变化而变化。这种阈值的设置方式考虑了当前物理机多种资源的使用情况,最小化了SLA违背率。然而这种方法没有考虑数据中心总体的负载量,如果针对单台物理机设置阈值,当整个数据中心负载量很大时,找不到目的主机的虚拟机数量会增多,增加了不必要的迁移代价和传输开销。

与文献^[12]类似,文献^[13]面对数据中心应用程序性能的下降,为降低SLA违背率,提出了虚拟机部署算法,采用一种自适应阈值的虚拟机迁移判决机制,针对负载不同的物理机设置不同的阈值。阈值的设置基于历史CPU利用率的统计,在一定时间内,统计分析每一个主机CPU利用率的历史值,通过求平均值和方差来设置阈值。这种阈值的设置考虑了应用程序一段时间的行为,提高了资源利用率,降低了SLA违背率,减少了不必要的虚拟机迁移数量。但是这种阈值的设置只考虑了CPU利用率,没有综合考虑其他维度的负载资源利用率情况。

除了基于动静态阈值的方式,虚拟机迁移时机的判决还有基于预测模型的方式。文献^[14]通过基于时间序列预测技术的自回归模型 $AR(n)$ 来预测将来的负载值。该模型通过对 n 个当前时间点之前且按时间序列进行排序的负载值进行计算,得出的结果就是未来的负载值。例如观测的负载值序列为: Z_1, Z_2, \dots, Z_t ,对于这个给定的时间序列,利用 P 阶自回归模型 $AR(t)$ 对第 $t+1$ 个时刻的负载值进行预测,则第 $t+1$ 个时刻的预测值 Z_{t+1} 为参数值加前 t 步所有加权输出值之和。同理, $t+2$ 步预测值即为其为参数的 $t+1$ 步预测值与前 t 步负载值加权输出值之和,以此类推。当预测值 Z_{t+1} 在闭区间 $[Z_{\min}, Z_{\max}]$ 之外时(Z_{\min} 是所设置的低阈值, Z_{\max} 是高阈值),虚拟机迁移将触发执行。基于预测模型的虚拟机迁移时机有效地避免了资源利用率瞬时峰值和谷值所产生的不必要的迁移,减小了迁移代价,然而预测模型下的预测值与真实值之间存在一定误差,会造成一定的性能下降,产生过多的能耗。

综上,针对国内外对虚拟机迁移时机研究的不足,本文提出了一种基于阈值滑动窗口机制的虚拟机动态迁移判决算法。

3 阈值滑动窗口机制

3.1 决定虚拟机迁移时机的因素

对虚拟机迁移时机的研究大多考虑的是一台物理机的资

源而没有考虑到整个数据中心的负载量。失败的虚拟机迁移将导致不必要的传输开销。

图1显示了一个数据中心中 n 台物理机 $PM_1, PM_2, PM_3, \dots, PM_n$ 中虚拟机的装载情况, CPU利用率的高低阈值窗口范围为 $0.4 \sim 0.8$ 。从图1中可以看到, PM_3 和 PM_n 都出现了虚拟机的超载, 按照超出阈值则进行迁移的方法, PM_3 和 PM_n 中都存在VM需要寻找目的主机进行迁移。而此时, 在大部分物理主机都满负载的情况下, 只有 PM_1 还有接受待迁虚拟机的空间, 因此 PM_3 或 PM_n 中的待迁虚拟机必将面临迁移失败。图2中, 调整CPU利用率的阈值窗口范围, 阈值变化幅度为 0.05 , 因此阈值窗口的大小变为 $0.35 \sim 0.85$ 。由图可知窗口大小增大后, 在图1中超载的物理机 PM_3 在图2中由于物理机CPU利用率在窗口范围内, 因此不被认为是超载主机, 不做虚拟机的迁移。图2表明在数据中心负载量很大时, 通过调整CPU利用率阈值窗口大小, 可以减少需要迁移的虚拟机数量, 降低了迁移失败率, 也减小了由于找不到目的物理机而带来的不必要的迁移代价。

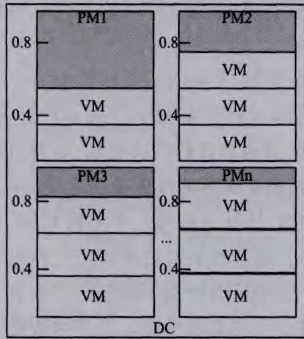


图1 数据中心中物理机的超载情况

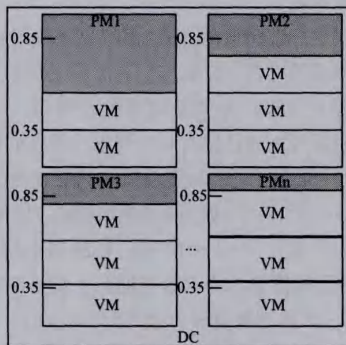


图2 阈值窗口变化后物理机的超载情况

为此, 本文提出阈值的动态滑动窗口机制, 用来调整阈值窗口大小。与虚拟机迁移时机相关的参数定义如下。

定义1(阈值窗口 $TWin$) 阈值窗口 $TWin$ 的大小是指资源利用率的最高阈值 T_{high} 与最低阈值 T_{low} 之间构成的范围 $[T_{low}, T_{high}]$ 。

本文主要考虑CPU资源利用率的使用情况来定义阈值, 若当前物理机的CPU利用率过高, 则会影响物理机的性能; 如果CPU利用率过低, 会造成资源的浪费。因此, 设置基于CPU利用率的高低阈值。其中, T_{high} 和 T_{low} 高低阈值分别指当前物理机CPU利用率所允许达到的最高和最低值。CPU利用率超过 T_{high} 称为超载, 低于 T_{low} 称为低载, 超载或低载情况都会引发告警, 落在高低阈值窗口 $TWin$ 内的CPU利用率

被认为是属于正常范围。

$TWin$ 窗口的大小随着数据中心负载的变化而变化, 如果数据中心负载过大, 则增大窗口, 如果数据中心负载过小, 则减小窗口, 这样的窗口称为滑动窗口。

定义2(数据中心任务量 R_{DC}) 用户向数据中心提交任务, 申请虚拟机资源。本文假设一个虚拟机上分配一个任务, 因此分配在物理机上的所有虚拟机的总数即为总任务数。为了更有效地刻画数据中心任务量的状况, 通过式(1)来衡量数据中心整体任务量的大小:

$$R_{DC} = \frac{NP_{over}}{NP_{sum}}, 0 < R_{DC} < 1 \quad (1)$$

其中, NP_{over} 为当前超载物理机的总数, NP_{sum} 为数据中心中所有物理机的总数。

定义3(迁移失败率 F_{mig}) 指在迁移过程中虚拟机找不到合适目的物理机的数量与总共需要迁移的虚拟机数量的比值, 由式(2)表示:

$$F_{mig} = \frac{NV_{fmg}}{NV_{mig}} \quad (2)$$

其中, NV_{fmg} 表示找不到合适目的物理机的虚拟机的数量, NV_{mig} 为总共需要迁移的虚拟机的数量。

F_{mig} 的值用来判断当前 $TWin$ 阈值窗口大小是否合适, 是否需要继续调整 $TWin$ 。在一段时间内, 通过调整 $TWin$, 使得 F_{mig} 降低, 表明当前数据中心的主机阈值窗口大小合适, 否则继续调整 $TWin$ 。

定义4(迁移代价 M_{cost}) 整个数据中心中迁移 n 台虚拟机的迁移代价定义为:

$$M_{cost} = \sum_{i=1}^n \frac{RAM_i}{BW_i} \quad (3)$$

超出阈值的虚拟机在迁移到其它主机时, 会产生迁移代价, 本文用虚拟机内存与所需带宽的比值来计算, 计算得到迁移所耗费的时间代价。式(3)中, RAM_i 表示虚拟机 i 所占的内存, BW_i 表示虚拟机 i 所占的带宽。

定义5(电源能耗 E) 参考文献[15]的定义, 对于一个有 m 台物理机的云数据中心, 第 j 台物理机的静态能耗记为 c_j ($1 \leq j \leq m$), 该物理机的CPU在 t 时刻的频率为 $f_j(t)$, CPU利用率为 $u_j(t)$, 功率(Watt)为 $e_j(t)$, 则第 j 台物理机在 t 时刻的电源能耗为:

$$e_j(t) = c_j + k * f_j^3(t) * u_j(t) \quad (4)$$

其中, k 是系数, 表示数据中心中单个物理机的动态能耗与CPU的频率的三次方和利用率成正比^[16]。

在 T 时间内, 数据中心的总能耗为:

$$E = \sum_{j=1}^m \int_0^T f_j(t) u_j(t) dt \quad (5)$$

3.2 问题描述

数据中心不断变化的任务量与不合理阈值窗口的设定可能会导致虚拟机迁移的失败, 虚拟机迁移失败率过大会造成大量的不必要的迁移消耗和传输代价。基于上述定义, 本文考虑对物理机CPU高低阈值窗口进行动态滑动, 以便根据数据中心的任务量情况, 自适应地对阈值窗口的大小进行调整, 使得迁移消耗、电源消耗最小化。在此需要解决以下问题:

- (1) 阈值窗口 $TWin$ 如何滑动?
- (2) 窗口的滑动幅度如何变化?
- (3) 如何动态预测数据中心的任务量 R_{DC} ?

因此,当前的问题描述为: u_j 表示当前物理机 j 上的 CPU 资源利用率。阈值窗口大小 $S=[T_{low}, T_{high}]$, $1 > T_{high} > T_{low} > 0$ 。当 $u_j \in S$ 时不迁移虚拟机,否则考虑当前数据中心任务量 R_{DC} 很大时,寻找合适的 CPU 阈值窗口 S ,使得 F_{mig} 下降。

R_{DC} 的预测模型是通过基于时间序列预测技术的自回归模型 $AR(n)$ 来预测将来数据中心的任务量。系统隔一定时间(如 30s)监测数据中心任务量情况,保存过去最近 n 个时刻任务量构成的序列。利用文献[14]中的 n 阶自回归模型对下一时间值进行计算,第 $n+1$ 时刻预测值 R_{n+1} 可表示为:

$$R_{n+1} = a_1 R_1 + a_2 R_2 + \dots + a_n R_n + \epsilon_{n+1} \quad (6)$$

其中, a_i 表示自回归系数, R_i 表示 i 时刻的任务量, ϵ_{n+1} 是满足正态分布的噪声。

如果 R_{n+1} 的值大于所设置的 R_{DC} 最大值,则认为数据中心任务量过大,调整 CPU 利用率高低阈值窗口,寻找合适的 CPU 阈值窗口 S ,使得 F_{mig} 下降。

3.3 算法描述

算法 基于阈值滑动窗口机制的虚拟机迁移判决算法

输入:虚拟机数量 n ,物理机数量 m ,初始化阈值 $[T_{low0}, T_{high0}]$, R_{DC} 上限常量 R_0 , F_{mig} 上限常量 F_0

输出:电源能耗 E ,迁移代价 M_{cost}

1. for each j from 0 to $m-1$ do
2. if isPMoverload(pm) high_flag=1; /* 存在主机超载 */
3. if isPMUnderloaded(pm) low_flag=1; /* 存在主机装载不足 */
4. End for
5. if high_flag then
6. {if ($R_{DC} > R_0$) then /* 当预测的数据中心任务量 R_{DC} 很大时,调整窗口大小 */
7. { $S=[T_{low}-D, T_{high}+D]$;
8. while ($F_{mig} > F_0$) && $u_j \notin S$ && ($T_{low} > 0$) && ($T_{high} \leq 1$) /* 当迁移失败率 F_{mig} 很大时,继续调整窗口大小 */
9. { $T_{low} = T_{low} - D$; $T_{high} = T_{high} + D$; $S=[T_{low}, T_{high}]$;
10. if ($T_{low} < 0 \parallel T_{high} > 1$) { $T_{low} = T_{low} + D$; $T_{high} = T_{high} - D$; break;}
11. if $u_j \in S$ then
12. setDisableMigrations(true);} /* 当物理机负载值在窗口范围内时,不迁移虚拟机 */
13. if ($u_j \notin S$) vmsToMigrate.add(getVmsToMigrateFromOverloadedPm(pm)); /* 迁移超载物理机上的虚拟机 */
14. compute E, M_{cost}
15. vmsToMigrate.clear()
16. if low_flag then
17. vmsToMigrate.add(getVmsToMigrateFromUnderloadedPm(pm)); /* 迁移装载不足物理机上的虚拟机 */
18. compute E, M_{cost}
19. return E, M_{cost}

算法流程为:

当物理机 CPU 利用率大于当前所设置的高阈值时,认为此时的物理机是超载的。同时当计算所得数据中心总的任务量 R_{DC} 时,如果虚拟机迁移失败率 F_{mig} 没有下降,则分别调整高低阈值的阈值幅度(阈值幅度为 D),直到迁移失败率下降,此时窗口大小合适,迁移当前超载物理机上的虚拟机。

当物理机 CPU 利用率低于当前所设置的低阈值时,认为此时的物理机是装载不足的。对于装载不足的物理机,直接迁移其上的虚拟机。

4 实验

本文采用 Cloudsim 平台^[16]对 iWnd 算法进行了实验。实验在 Cloudsim 平台上创建了模拟数据中心,构造了一定规模的虚拟机 VM、一定规模的物理机 PM,VM 和 PM 的内存、带宽、存储和 CPU 资源的配置如表 1 所列。

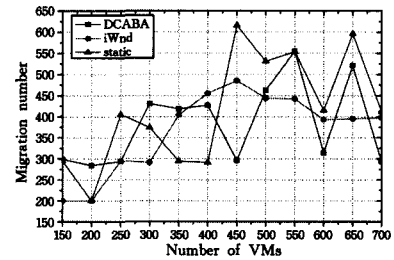
表 1 VM 和 PM 的实验配置

	物理机	虚拟机
内存	100GB	256MB
带宽	20Gbps	20Mbps
存储	1TB	2.5GB
CPU 主频	1000/2000/3000MIPs	250/500/850/1000MIPs

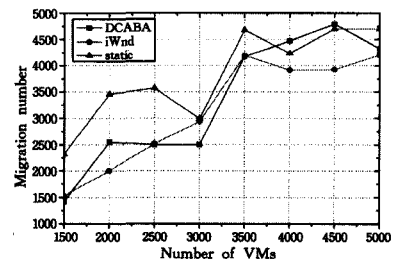
实验为每一个主机配置一个 CPU,虚拟机的分配满足使所分配的物理机所剩资源最少。因为算法关系到整个数据中心的任务量大小,为了讨论算法在不同规模数据中心下的效果,本文模拟了两种不同规模的数据中心,分别是百量级规模(由百台虚拟机和物理机组成)和千量级规模(由千台虚拟机和物理机组成)。通过 3 组实验模拟了不同规模(百量级、千量级)虚拟机与物理机之间的耗能差别和迁移代价,其中虚拟机与物理机的数量级规模相同。实验中每台虚拟机配置相同,每台物理机配置亦相同,比较了不同数量级虚拟机采用 iWnd 虚拟机迁移判决机制和静态阈值迁移判决机制(static)、文献[11]中动态阈值迁移机制(DCABA)时虚拟机的迁移数量和产生的迁移代价,同时对比了 iWnd 和 non-iWnd 方法数据中心的耗能。最后,基于不同的阈值变化幅度,比较耗能大小。

实验 1 迁移数量的对比

图 3 比较了不同数量级规模的虚拟机分别在 iWnd 和静态阈值迁移判决以及 DCABA 动态判决情况下的虚拟机迁移数量,其中 iWnd 算法和静态阈值迁移判决时都设置高阈值为 80%,低阈值为 20%,DCABA 算法高、低阈值的设置由最优优化耗能得到。



(a) 百量级规模的虚拟机迁移数量



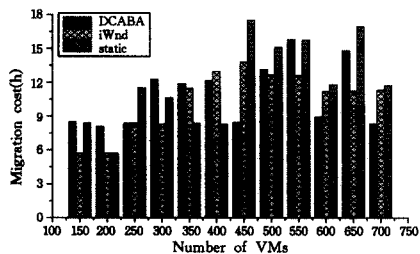
(b) 千量级规模的虚拟机迁移数量

图 3 不同数量级规模的虚拟机迁移数量

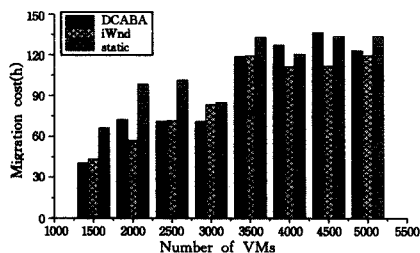
在图 3(a)、图 3(b)中,与 iWnd 迁移判决机制相比,静态阈值判决和 DCABA 迁移判决时虚拟机迁移数量的变化趋势起伏较大。不断变化的负载在达到瞬时峰值谷值时导致了虚拟机的迁移,iWnd 考虑了数据中心整体负载的变化,一定程度上避免了由于瞬时峰值谷值所造成的不必要的迁移,减少了虚拟机的迁移数量。比较图 3(a)与图 3(b)可知,随着虚拟机数量的增多,使用 iWnd 虚拟机迁移判决机制更能减少虚拟机的迁移数量,减少不必要的迁移开销和传输代价。

实验 2 迁移代价对比

图 4 比较了不同数量级规模的虚拟机在采用 iWnd 和静态阈值迁移判决机制以及 DCABA 迁移判决时产生的迁移代价。迁移代价为整个数据中心虚拟机迁移所产生的消耗,由式(3)得出。



(a) 百量级规模的虚拟机迁移代价



(b) 千量级规模的虚拟机迁移代价

图 4 不同数量级规模的虚拟机迁移代价

图 4(a)表明百量级规模的虚拟机在静态阈值迁移判决时由于负载的动态变化导致了迁移数量的起伏较大,产生了波动较大的迁移代价。与静态阈值迁移判决方法比较,iWnd 随着虚拟机数量的增长,迁移代价的变化较为平缓。当虚拟机多于 450 台时,静态阈值迁移判决算法产生的迁移代价始终大于 iWnd 所产生的代价。阈值窗口的设置有效地避免了 CPU 利用率在瞬时峰值和谷值所导致的虚拟机迁移。当数据中心中虚拟机数量较少时,DCABA 迁移机制所产生的迁移代价小于 iWnd。虚拟机少于 250 台时,DCABA 判决机制由于数据中心规模较小,产生较少的迁移代价。在虚拟数量为 250 台时,两种判决机制产生相同的迁移代价,约为 8.4h。随着虚拟机数量的增长,DCABA 产生的迁移代价远远大于 iWnd。

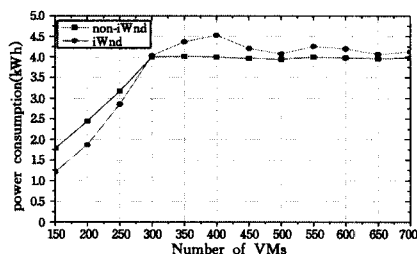
图 4(b)表明随着虚拟机数量级规模的增大,静态阈值迁移判决算法所产生的迁移代价始终大于 iWnd。在虚拟机数量为 3000 台时,二者所产生的迁移代价相近,大约为 84h。在虚拟机少于 3000 台时,静态阈值迁移判决算法产生的迁移代价是 iWnd 的 1.5 倍左右。虚拟机多于 3000 台时,迁移代价的增长趋势有所下降。迁移次数的减少一方面可以保证系统的稳定,另一方面也可以减小因为迁移而带来的迁移开销和传输代价。随着虚拟机数量级规模的增大,DCABA 与 iWnd 迁移判决机制产生近似的迁移代价。当虚拟机少于

2000 台时,DCABA 产生的迁移代价小于 iWnd 产生的代价。虚拟机数量为 2000 台时,iWnd 产生的迁移代价比 DCABA 产生的迁移代价减少了约 22.22%。当虚拟机多于 3500 台时,DCABA 与 iWnd 产生的迁移代价之间的差值增长。在虚拟机数量为 4500 台时,差值达到了最大值,DCABA 产生的迁移代价约为 iWnd 的 1.22 倍。

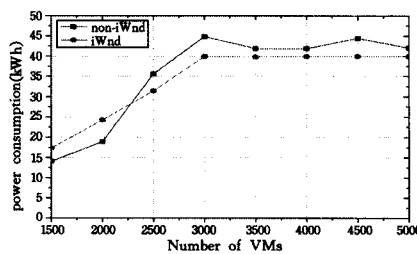
比较图 4(a)、图 4(b)可知,对于不同数量级规模的虚拟机,迁移代价随着虚拟机数量级的增大而逐渐减少,说明数据中心整体规模越大时,应用 iWnd 迁移代价越不明显。

实验 3 整体耗能对比

图 5 比较了不同数量级虚拟机在 iWnd 和 non-iWnd 情况时电源能耗的对比。其中,non-iWnd 指的是不用 iWnd 时产生的能耗。电源能耗为系统运行过程中数据中心所有物理机消耗电源的总和,由式(5)计算得出。



(a) 百量级规模的虚拟机电源能耗



(b) 千量级规模的虚拟机电源能耗

图 5 不同数量级规模的虚拟机电源消耗

其中,Cloudsim 中根据当前设置的物理机配置,测算得到物理机的平均静态能耗 $c=175\text{W}$ 。本文假设 $f_j(t)$ 是固定的,能耗的取值取决于物理机 CPU 的利用率,取多组同一频率下不同 CPU 利用率下 $e_j(t)$ 的值,根据式(4),计算系数 k 的值,求解得出当 $f_j(t)$ 为 3GHz 时, k 值为 2.66; $f_j(t)$ 为 2GHz 时, k 为 8.58; $f_j(t)$ 为 1GHz 时, k 为 75.06。

在图 5(a)中,当虚拟机少于 300 台时,虚拟机迁移的电源消耗比不迁移时小。当虚拟机数量为 150 台时,与不迁移相比较,迁移时电源的消耗节省了将近 32%。当虚拟机多于 300 台时,迁移与不迁移相比,电源损耗之间的差距很小。因此,对于百量级虚拟机,使用 iWnd 算法不仅减少了迁移的虚拟机数量,而且电源损耗增加很小,甚至在一定范围内虚拟机不迁移时比迁移更节能。

在图 5(b)中,当虚拟机少于 2000 台时,与不迁移时相比,迁移时虚拟机消耗电源近似节省了 20%。电源损耗随着虚拟机数量的增加而增加,在虚拟机多于 3500 台时,能耗趋于稳定。当虚拟机多于 2000 台时,不迁移比迁移时消耗的电源少,但它们之间的差距是很小的。因此,在千量级虚拟机情况下,使用 iWnd 在减少虚拟机迁移数量的同时,并没有带来太多的电源损耗。

实验4 不同阈值幅度下虚拟机的耗能状况

图6给出了不同数量的虚拟机在不同阈值幅度情况下的能耗对比情况。限定虚拟机数量不变,改变虚拟机迁移判决算法中阈值窗口的调整幅度,以观测阈值窗口的变化幅度对耗能的影响。

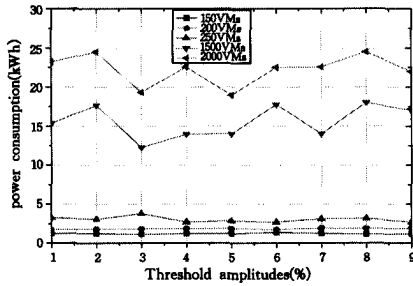


图6 不同阈值幅度下的电源能耗

图6表明,虚拟机能源的消耗既取决于虚拟机的数量级又取决于阈值窗口的变化幅度。在百量级虚拟机的情况下,阈值窗口的变化幅度对能源消耗的影响是很小的,然而阈值窗口的变化幅度对千量级虚拟机时能耗的影响是很大的。在虚拟机的数量是1500台和2000台时,阈值窗口一次变化的最佳幅度是3%和5%。因此,对于数量级很大的虚拟机需要考虑阈值窗口的变化幅度。

结束语 本文针对数据中心不断变化的负载量提出了基于阈值滑动窗口机制的虚拟机迁移判决算法,阈值窗口的大小随着负载量的变化而变化,与静态阈值的迁移判决机制相比,有效地减少了在数据中心负载量很大时虚拟机迁移的数量,避免了由于虚拟机找不到目的物理机所产生的不必要的迁移代价。实验结果表明本文提出的*iWnd*有效地降低了系统开销和传输代价。由不同数量级的虚拟机在迁移和不迁移两种情况下的比较可知,*iWnd*在减少虚拟机迁移数量的同时,并没有增加过多的能耗;与已有的动态算法比较,迁移失败率也有所降低。因此,本文的方法有效地减小了系统开销和迁移代价。同时,实验验证了不同的阈值窗口变化幅度对于千量级范围内虚拟机的耗能有一定的影响。因此,对于数量级很大的虚拟机要考虑阈值窗口的变化幅度。下一步将在考虑数据中心负载量的基础上,对负载的不同类型做分类放置和迁移决策,以便更有效地提高资源利用率。

参考文献

[1] Armbrust M, Fox A, Griffith R, et al. A view of cloud computing [J]. *Communications of the ACM*, 2010, 53(4): 50-58

[2] Chu Ya, Ma Ting-huai, Zhao Li-cheng. Cloud Computing Resource Scheduling, Policy and Algorithm [J]. *Computer Science*, 2013, 40(11): 8-13 (in Chinese)

储雅, 马廷淮, 赵立成. 云计算资源调度: 策略和算法 [J]. *计算机科学*, 2013, 40(11): 8-13

[3] Lin Chuang, Tian Yuan, Yao Min. Green Network and Green Evaluation: Mechanism, Modeling and Evaluation [J]. *Chinese Journal of Computer*, 2011, 34(4): 593-612 (in Chinese)

林闯, 田源, 姚敏. 绿色网络和绿色评价: 节能机制、模型和评价 [J]. *计算机学报*, 2011, 34(4): 593-612

[4] Verma A, Ahuja P, Neogi A. pMapper: Power and Migration

Cost Aware Application Placement in virtualized Systems [C] // Proc. Ninth ACM/IFIP/USENIX Int' Conf. Middleware. 2008: 243-264

[5] Jung G, Hiltunen M A, Joshi K R, et al. Mistral: Dynamically Managing Power, Performance, and Adaptation Cost in Cloud Infrastructures [C] // Proc. 30th Int'l Conf. Distributed Computing Systems (ICDCS). 2010: 62-73

[6] Ai Hao-jun, Gong Su-wen, Yuan Yuan-peng. Research of Cloud Computing Virtual Machine Allocated Strategy on Multi-objective Evolutionary Algorithm [J]. *Computer Science*, 2014, 41(6): 48-53 (in Chinese)

艾浩军, 龚素文, 袁远朋. 基于多目标演化算法的云计算虚拟机分配策略研究 [J]. *计算机科学*, 2014, 41(6): 48-53

[7] Zhang Wei, Song Ying, Ruan Li, et al. Resource Management in Internet-Oriented Data Centers [J]. *Journal of Software*, 2012, 23(2): 179-199 (in Chinese)

张伟, 宋莹, 阮利, 等. 面向 Internet 数据中心的资源管理 [J]. *软件学报*, 2012, 23(2): 179-199

[8] Zhu X, et al. 1000 Islands: Integrated Capacity and Workload Management for the Next Generation Data Center [C] // Proc. Fifth Int'l Conf. Autonomic Computing (ICAC). 2008: 172-181

[9] VMware. VMware Distributed Power Management Concepts and Usage [R]. 2010

[10] Jing Xu, Fortes J A B. A Multi-objective Approach to Virtual Machine Management in Datacenters [C] // Proc of the 6th International Conf. on Autonomic Computing (ICAC'11). 2011: 225-233

[11] Sahu Y, Pateriya R K, Gupta R K. Cloud Server Optimization with Load Balancing and Green Computing Techniques Using Dynamic Compare and Balance Algorithm [C] // 2013 5th International Conference on Computational Intelligence and Communication Networks. 2013: 527-531

[12] Sinha R, Purohit N, Diwanji H. Energy efficient dynamic integration of thresholds for migration at cloud data centers [C] // IJCA Special Issue on CN. 2011: 44-49

[13] Beloglazov A, Buyya 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, USA: ACM, 2010: 1-6

[14] Bobroff N, Kochut A, Beaty K. Dynamic Placement of Virtual Machines for Managing SLA Violations [C] // Proc. IFIP/IEEE 10th Int'l Symp. Integrated Network Management (IM). 2007: 119-128

[15] Song Jie, Li Tian-tian, Yan Zhen-xing, et al. Energy-Efficiency Model and Measuring Approach for Cloud Computing [J]. *Journal of Software*, 2012, 13(2): 200-213 (in Chinese)

宋杰, 李甜甜, 闫振兴, 等. 一种云计算环境下的能效模型和度量方法 [J]. *软件学报*, 2012, 23(2): 200-213

[16] Calheiros R N, Ranjan R, et al. CloudSim: A Novel Framework for Modeling and Simulation of Cloud Computing Infrastructures and Services [J]. *Software: Practice and Experience*, 2011, 41(1): 23-50