计算机科学 COMPUTER SCIENCE
http://www.jsjkx.com
DOI:10.11896/jsjkx.190900189

基于自适应虚拟机迁移的云资源调度机制

李双刚[1]　张　爽[1]　王兴伟[2]
1 东北大学软件学院　沈阳 110169
2 东北大学计算机科学与工程学院　沈阳 110169
(786051741@qq.com)

摘　要　虚拟机迁移是当前云计算资源调度的重要研究方向之一。目前,用户规模的不断增长带来了一些新的挑战,传统迁移策略很难适应动态变化的内外部环境。对此,设计了自适应虚拟机迁移的总体框架,通过对虚拟机迁移建模,提出了“迁移路径”和“服务开销”等概念,并以服务器的CPU利用率和服务器间的带宽利用率为指标,为系统中所有迁移的虚拟机规划最优的迁移路径,以使系统总的服务开销最小化。首先,设计了基于阈值的虚拟机筛选算法来挑选可迁移的虚拟机;接着,设计了基于自回归积分滑动平均模型的时间序列预测算法,用以预测服务器未来时间窗口内的服务开销;然后,利用动态规划基于服务器服务开销的预测值设计了迁移路径计算算法,为每个待迁移虚拟机规划出最优的迁移方案;最后,利用迁移路径下服务器服务开销的预测值与真实值之间的差距所反映出的预测窗口性能的优劣,设计并实现了一个预测窗口自适应调整算法。实验表明,该自适应虚拟机迁移算法在自适应性调整和最小化服务开销等方面具有良好的效果。

关键词:虚拟机迁移;自适应;ARIMA模型;动态规划;预测窗口
中图法分类号　TP302

Cloud Resource Scheduling Mechanism Based on Adaptive Virtual Machine Migration

LI Shuang-gang[1],ZHANG Shuang[1] and WANG Xing-wei[2]
1 College of Software,Northeastern University,Shenyang 110169,China
2 College of Computer Science and Engineering,Northeastern University,Shenyang 110169,China

Abstract　Virtual machine (VM) migration is an important research field of current cloud computing resource scheduling. Now the continuous growth of users has brought some new challenges,and current typical migration strategies are difficult to adapt to dynamically changing internal and external environments. Aiming at this problem,this paper proposed an overall framework of adaptive VM migration. Via modeling VM migration,the concepts of “migration path” and “service overhead” were proposed,and the server's CPU utilization and bandwidth utilization of links between servers were used as indicators to plan the optimal migration path for all to-be-migrated VMs in the system to minimize the total service overhead. Firstly,a threshold-based selection algorithm is presented for the selection of the to-be-migrated VMs. Secondly,an auto regressive integrated moving average model (ARIMA)-based time series prediction algorithm is designed to predict the service overhead within the server's future time window. Then,the migration path calculation algorithm is designed based on servers' predicted service overhead and dynamic programming,and an optimal migration plan is made for each to-be-migrated VM. Finally,based on the performance of the prediction window determined by the difference between the predicted service overhead and the real value via the migration path,a prediction window adaptive adjustment algorithm is designed and implemented. Experiments prove that the adaptive VM migration has good effects in terms of adaptive adjustment and minimizing service overhead.

Keywords　VM migration,Adaptive,ARIMA,Dynamic programming,Prediction window

云计算通过虚拟化等技术把数据中心(Data Center,DC)[1-2]中大量服务器的CPU、内存等资源整合起来,构成共享资源池,在资源池内放置虚拟机(Virtual Machine,VM),对外提供统一服务。云通常包括成千上万甚至数十万台物理服务器和虚拟机,因此有效地调度众多的云资源十分重要。

目前,针对云资源调度的研究有拍卖[3-7]、博弈[8-9]、性能与能量感知[10-13]、基于神经网络与优化算法[14]等。相比之下,云数据中心的资源调度机制聚焦在虚拟机的放置和迁移

到稿日期:2019-09-27　返修日期:2019-12-14　本文已加入开放科学计划(OSID),请扫描上方二维码获取补充信息。
基金项目:国家自然科学基金(61872073,61572123);辽宁省高校创新团队支持计划(LT2016007)
This work was supported by the National Natural Science Foundation of China (61872073, 61572123) and Program for Liaoning Innovative Research Team in University (LT2016007).
通信作者:张爽(zhangs@swc.neu.edu.cn)

等决策上，前者只须考虑在合适的物理服务器上按照用户需求创建虚拟机，而后者面对的问题更加复杂。随着用户需求和服务器负载等内外部环境的变化，放置后的虚拟机很可能面临新的挑战，如服务器负载不均衡，用户对虚拟机资源需求临时增加而当前服务器剩余资源相对不足，单个或多个服务器节点故障，系统面临虚拟机失效和用户服务中断的风险等。以上问题影响了系统服务的安全性和稳定性，而研究云资源调度下虚拟机的迁移是解决该问题很好的突破点。虚拟机迁移把一台虚拟机从一台服务器迁到另一台服务器上运行，迁移期间需要考虑服务器间链路带宽、网络拥塞程度、负载均衡以及迁移后的服务开销评估等因素，涉及迭代内存复制和网络重新配置，通常用于负载平衡、节能和灾难保护等[15]。

近年针对云资源调度中虚拟机迁移的研究，主要集中在系统节能、可靠性和安全性等方面。文献[16]提出了一种服务感知策略，用于云到云之间多个虚拟机的迁移服务，其分析多个虚拟机的依赖关系，根据网络流量强度来确定受依赖的虚拟机的迁移顺序，以减少服务停机时间。文献[17]研究了云中虚拟机的动态迁移问题，建立了动态迁移的三目标优化模型，考虑了能耗、虚拟机之间的通信和迁移成本，提出了基于二进制图匹配的桶代码学习算法，该算法使用桶编码和学习来寻找最优解，并且使用二进制图匹配评估候选解。文献[18]针对云数据中心虚拟机整合的问题，提出了一种功率感知调度算法，其与利用率阈值策略和虚拟机选择策略相结合，每当物理机被认为过载时，功率感知调度算法会尝试选择最合适的虚拟机进行迁移。文献[19]提出的用于管理虚拟机迁移和物理机分配的框架，着眼于控制源和目标物理机上的用户流量大小，以便提供虚拟机迁移所需的带宽。文献[20]阐述了虚拟机迁移以及迁移性能的定量分析工作，讨论了其优势和挑战，总结了将虚拟机迁移与用户移动性相关联的研究，并指出了虚拟机迁移未解决的问题。文献[21]重点关注大型无共享 IaaS 云中在线虚拟机迁移的实时调度问题，提出了一个可扩展的分布式代理网络，用于监督提供商环境中所有正在进行的迁移操作的进度。为了应对托管基础架构内有效迁移多个虚拟机，同时最大限度地减少服务中断的挑战，文献[22]提出了一种几何编程模型，其目标是优化多个虚拟机的在线迁移的比特率分配，并最小化总迁移时间；同时，定义了停机时间和资源利用时间之间的成本权衡函数。文献[23]提出了一种基于簇感知的虚拟机协同迁移策略，旨在应对媒体云中应用的高带宽以及服务器之间的大流量挑战。预拷贝是一种广泛使用的在线迁移机制，文献[24]针对当前网络技术难以提供预拷贝机制所需的频繁变化的带宽问题，先从理论上分析了能保证在线虚拟机迁移的总迁移时间和停机时间的带宽需求，然后提出了 Reciprocal-based Model 模型来保证计算的带宽。为了适应第五代雾计算支持的无线接入网络（简称 5G FOGRAN）即将到来的趋势，文献[25]实现了一种自适应的带宽管理器 SCBM，通过将 MPTCP 作为传输协议的无线 5G FOGRAN 连接，实现了虚拟机在线迁移所消耗的网络能量的约束最小化。文献[26]设计了 Scatter-Gather 在线迁移机制，以在迁移期间解耦源和目标。文献[27]最大限度地利用虚拟机的 CPU 容量，提出了基于动态加权在线迁移机制，通过阈值模型来设计负载不平衡管理策略。文献[28]聚焦于节能和迁移成本的降低，提出了虚拟机迁移启发式新算法，该算法采用虚拟机集群，考虑了虚拟机放置伴随的流量大小问题。文献[29]为了保证虚拟机的迁移与用户的策略一致，提出了一种用户策略限制的虚拟机迁移协议，使虚拟机迁移能够由云用户控制和验证。文献[30]提出了两种新的虚拟机选择算法，即最小化虚拟机迁移次数和最小化迁移时间下的最少虚拟机迁移次数，以避免在同一虚拟机上频繁发生 SLA 违规。文献[31]设计了 Xen 环境下智能预拷贝的在线迁移机制，该机制估计迭代预拷贝期间的停机时间以确定是否进入停止和拷贝阶段。文献[32]采用整型线性规划来计算每个时期虚拟机的最佳迁移策略。文献[33]使用处理器工作负载预测方法迁移出过载的虚拟机，避免过载。文献[34]提出了一种基于节能和预测的虚拟机放置和迁移集成框架，用于分配和使用云资源。文献[35]提出了修改优化预拷贝的方法来减少不必要的页面传输，并将它与基于特征的压缩算法组合起来，使迁移过程更有效。文献[36]在保证虚拟机以一定概率可用的情况下，研究了如何在最少的节点上放置多个虚拟机组且每组有虚拟机，构建了多路径路由问题模型以解决服务器间的连接可用性问题，并设计了启发式算法进行求解。文献[37]将预拷贝（Pre-copy）和后拷贝（Post-copy）组合，提出了一种混合虚拟机迁移技术，并在 CPU 使用、内存和网络方面进行优化。

针对云资源调度机制中虚拟机迁移的问题，国内外已有研究成果一般是在虚拟机整个运行周期只迁移一次或者在迁移时只考虑一步，没有综合考虑未来一段时间内连续多轮迁移的情况。传统迁移虽然操作简单，但是越来越无法适应用户规模和需求的不断增长以及外部环境的动态变化。鉴于此，本文提出了能够自适应于环境和用户需求变化的虚拟机迁移机制，通过对虚拟机剩余运行周期进行建模，提出了“迁移路径”和“服务开销”的概念。该机制根据服务器相关历史数据预测出服务器未来某段时间内的“服务开销”，利用动态规划为待迁移虚拟机计算出迁移路径，综合优化后续的多次迁移，并在预测结果和实际运行结果间建立对比评价体系（预测窗口调整机制），再依据评价结果动态调整相关参数，优化系统模型，实现迁移的自适应性，以更好地适应内外部环境变化。

本文第 1 节介绍自适应虚拟机迁移的系统框架；第 2 节介绍问题模型；第 3 节详细介绍虚拟机自适应迁移机制的实现步骤及相应算法；第 4 节对仿真实验结果进行阐述；最后对全文进行总结。

1 系统框架

自适应虚拟机迁移是在云资源调度下虚拟机迁移的基础上加入预测和自适应调整优化，通过对比系统监测数据和预测数据来自动调整系统参数。如图 1 所示，自适应虚拟机迁移系统框架主要由筛选模块、监测模块（资源监测和迁移监测）和迁移决策与实施模块组成。筛选模块是基于一定阈值条件，从所有虚拟机中抽取符合条件的虚拟机组成待迁移虚拟机集合；监测模块负责对迁移前虚拟机和服务器运行数据

以及迁移后相关数据进行监测，前者为筛选做准备，后者根据监测迁移虚拟机时服务器所需开销以及迁移后运行产生的开销来为自适应调整机制提供系统环境参数；迁移决策与实施模块根据历史数据先预测出未来一段时间内服务器的服务开销，然后基于此开销为待迁移虚拟机规划未来一段时间内的多个运行位置，即目标服务器（虚拟机未来迁入的服务器，而不是当前服务器）序列，并实施迁移。

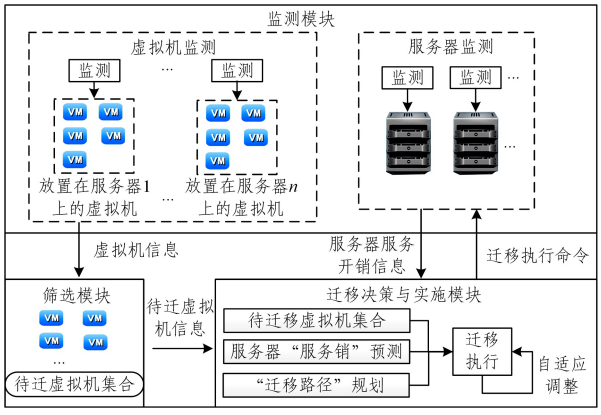

图 1 自适应虚拟机迁移的系统框架

Fig. 1 System framework of adaptive VM migration

虚拟机自适应迁移工作流程：首先，系统监测模块会监测系统各个虚拟机和服务器的运行和资源使用情况，收集相关参数；其次，系统根据阈值筛选条件和监测的数据，筛选出符合条件的虚拟机形成待迁移集合；接着，系统会调用迁移决策与实施模块，先根据服务器历史数据预测出未来时间段内服务器运行虚拟机开销的时间序列和服务器间迁移虚拟机开销的时间序列，再基于此序列和当前各个服务器的情况，通过一定算法规划出迁移的目标服务器序列，同时调用监测模块对服务器运行开销和服务器间虚拟机迁移开销进行实时监测；最后，将预测值与真实结果对比，判断预测时间窗口的优劣性，并进行相应调整，以达到自适应的目的。

2 问题建模

2.1 迁移路径

本节对虚拟机在剩余运行周期内各时间段的运行位置进行建模，研究虚拟机在剩余运行周期内的服务器“选址”问题。首先，对虚拟机剩余运行时间进行规范化处理：把虚拟机剩余运行时间离散成一个个时间片，得到一段段较小的运行周期。在每个时间片内，虚拟机运行在一台服务器上。各个时间片与各个服务器相对应，连续的时间片对应的服务器可能相同，也可能不同，两者没有必然关系。时间片对应的服务器按先后顺序构成了该虚拟机某段时间内的目标服务器序列。迁移路径是虚拟机在剩余运行周期内的目标服务器序列，这些序列决定了虚拟机在以后某时刻的运行位置。在迁移过程中，通过比较下一跳目标服务器与当前服务器来决定是否需要迁移，只有下一跳服务器与当前服务器不是同一台时，迁移才会发生。

假设一台虚拟机的运行周期长度为 $length$，被放置的时刻是 $place_time$，则可以得出这台虚拟机的运行截止时间即关机时间为 $place_time+length-1$。若当前时刻为 $cur\in[place_time, place_time+length-1]$，则该虚拟机的剩余运行周期长度为 $place_time+length-1-cur$。其迁移路径可以表示为：

$$migr_path=(serv_0, serv_1, \cdots, serv_{T-1}) \tag{1}$$

其中，T 表示虚拟机的剩余运行周期划分的时间片个数，$serv_i$ 表示在时间片 i 内运行该虚拟机的服务器的标识。

本文所用变量及其含义如表 1 所列。

表 1 符号及其意义

Table 1 Mathematical symbols and its meaning

数学符号	含义
$length$	虚拟机运行周期
$place_time, cur$	虚拟机放置时刻/当前运行时刻
$migr_path$	虚拟机迁移路径
$serv_i$	在时间片 i 内运行某虚拟机的服务器的标识
cpu_rate_i	服务器 i 的 CPU 利用率
$bd_rate_{i,i+1}$	服务器 i 和 $i+1$ 间的链路带宽利用率
α, β	虚拟机运行开销/迁移开销占总开销的比重
$overhead_{t,i}$	在时间片 t 内服务器 i 的服务开销
N_Serv	服务器数量
$start, end$	在预测的迁移路径下，虚拟机开始/完成所有可能迁移的时刻
PW_upper	预测时间窗口长度的上界
N	待迁移虚拟机数量
$cpu_avg_util_i$	服务器 i 的 CPU 平均利用率
cpu_thres	CPU 利用率阈值
N_slice	统计服务器 i 的 CPU 平均利用率时用到的 CPU 利用率的时间片个数
$thres_v$	筛选待迁移虚拟机时必须满足的剩余运行周期的阈值
d, p, q	ARIMA 模型的差分次数、自回归项数和滑动平均项数
PW	预测时间窗口长度
LC_{ij}	在时间片 i 内服务器 j 的运行开销
MC_{kij}	在时间片 k 内虚拟机从服务器 i 迁移到 j 的迁移开销
$overhead_ROMP$	真实的最优迁移路径下服务开销
$overhead_POMP$	预测的最优迁移路径下服务开销

2.2 服务开销

服务开销是虚拟机在剩余运行周期内迁移和运行所带来的链路通信开销和服务器开销的权重和，它包含两个层面：服务器上运行虚拟机的开销（称为运行开销）和服务器间虚拟机迁移所需的通信开销（称为迁移开销）。

本节对运行开销和迁移开销进行了离散化处理，前者基于服务器 CPU 利用率，即 CPU 工作时间占总时间的比重；后者基于服务器间的链路带宽利用率，即当前链路带宽使用量与链路带宽容量的比值。

一台虚拟机剩余运行周期的开销是在其迁移路径上运行开销和迁移开销的权重和，如式(2)、式(3)所示：

$$overhead=\alpha\sum_{i=0}^{T-1} cpu_rate_i+\beta\sum_{i=0}^{T-2} bd_rate_{i,i+1} \tag{2}$$

$$\alpha+\beta=1, \alpha, \beta>0 \tag{3}$$

其中，cpu_rate_i 表示服务器 i 的 CPU 利用率，$bd_rate_{i,i+1}$ 表示服务器 i 和 $i+1$ 之间的带宽利用率，α 和 β 分别表示前两者各自所占权重。为了降低虚拟机服务开销，应让虚拟机在剩余运行周期内最大可能地运行在低负载的服务器上，进行迁移时应最大可能地选择低负载的链路。

2.3 问题模型

自适应虚拟机迁移机制通过为各个待迁移虚拟机在各自

剩余运行周期下动态寻找最优的迁移路径，来使得所有虚拟机的总服务开销（亦即所有服务器的总服务开销）最小化。该问题模型如式（4）所示：

$$\min overhead_{\text{total}} = \sum_{i=0}^{N-1} overhead_i \tag{4}$$

其中，N 表示待迁移虚拟机的数量，$overhead_i$ 为虚拟机 i 在其所分配的迁移路径下的服务开销，$overhead_{\text{total}}$ 为所有迁移的虚拟机在其所分配的迁移路径下的服务开销总和。

3 自适应迁移机制

虚拟机的自适应迁移过程是动态持续进行的，根据虚拟机运行参数、服务器间链路状况和服务器负载情况，结合迁移效果评估，调整预测算法的参数，优化预测模型，进而优化虚拟机未来规划的迁移路径。迁移主要包含 4 个阶段：待迁移虚拟机的筛选，基于 ARIMA 时间序列模型的服务开销预测，规划迁移路径并实施迁移，自适应预测窗口调整。

3.1 第一阶段：待迁移虚拟机的筛选

虚拟机迁移时应考虑服务器的工作负载[38]（负载以服务器的 CPU 利用率为指标）和虚拟机剩余运行时间。易知，需要从高负载服务器中迁移虚拟机以降低其负载；对于剩余运行周期较短的虚拟机而言，迁移所能降低的系统开销相对有限，而对于运行周期较长的虚拟机，迁移所带来的系统优化空间更大。因此，迁移高负载服务器上剩余运行周期较长的虚拟机是合适的选择。

虚拟机筛选模块的流程如算法 1 所示。

算法 1 虚拟机筛选模块算法

输入：集群中服务器集合及其虚拟机集合
输出：待迁移的虚拟机集合
BEGIN
1. FOR 每一台服务器 DO
2. IF 服务器 i 最近 N_slice 个时间片内 CPU 平均利用率 $cpu_avg_util_i$ 达到阈值 cpu_thres
3. FOR 每一台虚拟机 DO
4. IF 该虚拟机的剩余运行时间大于阈值 thres_v THEN
5. 将该虚拟机唯一标识符放入待迁移的虚拟机集合
6. END IF
7. END FOR
8. END IF
9. END FOR
10. RETURN 待迁移的虚拟机集合
END

3.2 第二阶段：基于 ARIMA 时间序列模型的服务开销预测

本节引入预测窗口（Prediction Window，PW）的概念，即预测窗口是未来的某时间段长度，它规定了服务器开销预测的时间范围。就某台虚拟机而言，其预测窗口应当是其本身的剩余运行周期长度，由于要考虑多台待迁虚拟机，真实的预测窗口值很难与它们的剩余运行周期长度一致。此外，预测窗口的大小对预测的精确性具有重要的影响，其大小应当适中。

服务器运行虚拟机的历史开销数据和服务器间虚拟机的迁移开销历史数据被视为时间序列数据，具有显著的非线性和时变特性。本节利用 ARIMA 时间序列模型预测出未来一段窗口时间内（预测窗口长度的时间段）服务器的服务开销时间序列。

对于服务器服务开销时间序列（$overhead_0$，$overhead_1$，…，$overhead_{T-1}$），本文所设计的基于 ARIMA 模型的开销预测算法的流程如下。

（1）对于时间序列数据，采用单位根检验法判断序列是否平稳，若不平稳，需要采用 d 阶差分运算将其转化，直至平稳。

（2）对于生成的平稳时间序列，计算若干阶自相关系数和偏自相关系数。根据相关公式确定 p 和 q 的值；构造 $ARMA(p,q)$ 模型，然后代入时间序列的值，利用最小二乘法求出模型的参数。

（3）利用所建立的时间序列模型生成理论的时间序列，用理论的时间序列与真实时间序列作差，生成残差序列。然后对残差序列进行检验，如果残差服从高斯白噪声分布，说明所建立的模型已经能够较好地拟合所分析的时间序列，可以用于理论计算；否则，修改 p 和 q 的值，跳转到步骤（1）。

（4）根据预测窗口的大小 PW，利用构造好的模型对未来时间段内长度为 PW 的时间序列值进行预测。

3.3 第三阶段：规划迁移路径并实施迁移

寻找一条最优的迁移路径使得虚拟机的运行开销和迁移开销最小化是本节动态规划的核心问题。为了更清晰地表示迁移路径的选择过程，采用按时间片分层的表达形式，如图 2 所示，每一层与上下层对应位置的椭圆代表不同时间片内同一个服务器的带权运行开销。此外，为了使路径看起来具有统一性和连贯性，这里增加了两个特殊的虚拟节点，即 *start* 开始节点和 *end* 结束节点，分别代表虚拟机运行周期的开始和结束，连接到它们的箭头权重为 0，表示零开销。

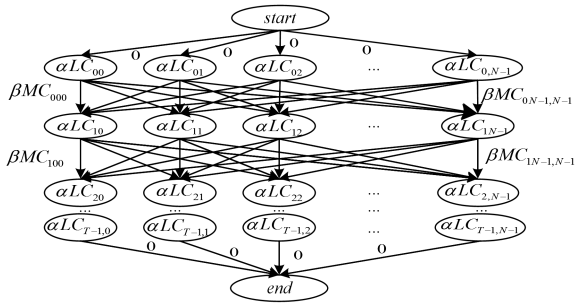

图 2 迁移路径问题

Fig. 2 Problem of path of migration

在图 2 中，LC_{ij} 代表在时间片 i 服务器 j 的运行开销，MC_{kij} 表示在时间片 k 虚拟机从服务器 i 向服务器 j 迁移时的迁移开销，α 和 β 分别为服务器运行开销和迁移开销在服务开销中所占的比重。

假设长度为 T 的最优迁移路径包含节点 $\alpha LC_{T-2,i}$，该路径是由到达节点 $\alpha LC_{T-2,i}$ 的长度为 $T-1$ 的最优路径加上从该节点到第 $T-1$ 层节点的最短路径构成，可得动态规划递推方程和反向推导方程，如式（5）、式（6）所示。

$$overhead_{t,i} = \min_k \{overhead_{t-1,k} + \beta MC_{t-1,k,i}\} + \alpha LC_{t,i},\quad k = 0,1,\cdots,N_Serv-1 \tag{5}$$

$$overhead_{t,i} = overhead_{t+1,i+1} - \alpha LC_{t+1,i+1} - \beta MC_{t,i,i+1} \tag{6}$$

其中，N_Serv 表示服务器数量，LC_{ti} 代表在时间片 t 服务器 i 的运行开销，$MC_{t,i,i+1}$ 表示在时间片 t 虚拟机从服务器 i 向服务器 $i+1$ 迁移产生的开销，$overhead_{t,i}$ 表示服务器 i 在时间片 t 的服务开销。

迁移路径计算算法的具体步骤如算法 2 所示。

算法 2 迁移路径计算算法

输入:服务器运行开销变化时间序列;服务器间虚拟机迁移开销变化时间序列

输出:迁移路径

```
BEGING
1. 创建动态规划状态矩阵 overhead
2. FOR 每一个时间片 t=1,2,…,T-1 DO
3.    FOR 每一个服务器 i DO
4.       按照式(5)计算出 overhead_{t,i}
5.    END FOR
6. END FOR
7. 找到最小的 overhead_{T-1,i}
8. FOR 每一个时间片 t=T-2,T-3,…,1
9.    FOR 每一个服务器 i DO
10.      IF 按照式(6)计算出状态值,如果与当前节点的状态值相等 THEN
11.      将该服务器放入迁移路径
12.   END IF
13. END FOR
14. END FOR
15. RETURN 迁移路径
END
```

求解得到迁移路径以后,在虚拟机剩余的运行周期内根据迁移路径迁移该虚拟机。根据文献[39]所述,虚拟机具有向运行该虚拟机的服务器发送迁移请求的能力,收到请求后服务器可以将该虚拟机迁移到另外一台服务器。如图 3 所示,随着时间片的更新,虚拟机检查迁移路径上的服务器与当前服务器是否匹配。如果不匹配,则向服务器发送迁移请求,然后由服务器将该虚拟机迁移到与迁移路径对应的服务器上;如果匹配,则虚拟机在该时间片内不发送迁移消息,仍在当前节点上运行。可以看出,虚拟机的迁移触发条件是系统运行的时间变化,这恰好与本设计所关注的系统时变特征相吻合。迁移路径概念的提出,是本机制自适应的体现之一。

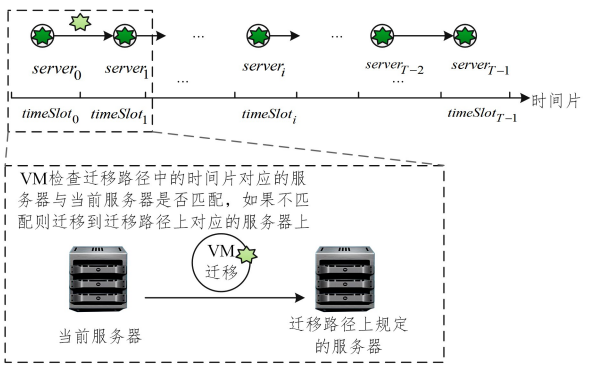

图 3 迁移路径的工作方式

Fig. 3 Action mode of migration path

3.4 第四阶段:自适应预测窗口的调整

为了避免预测异常带来较大偏差,须对预测窗口进行自适应优化调整。

基于 ARIMA 模型预测得到的迁移路径被称为预测最优迁移路径(Predicated Optimal Migration Path,POMP);通过对服务器监测,利用真实的系统运行开销时间序列计算出来的最优路径被称为真实最优迁移路径(Real Optimal Migration Path,ROMP)。

从历史数据中可以找到服务器中虚拟机运行开销的真实时间序列,如式(7)所示:

$$(LC_{0i},LC_{1i},\cdots,LC_{start,i},\cdots,LC_{k,i},\cdots,LC_{end,i},\cdots,LC_{Ti}) \tag{7}$$

其中,$0,\cdots,start,\cdots,end,\cdots,T$ 为对应的各个时间片,LC_{ki} 表示服务器 i 在时间片 k 中的运行开销值。

同理,可以得到进行迁移的真实开销时间序列:

$$(MC_{0ij},MC_{1ij},\cdots,MC_{startij},\cdots,MC_{kij},\cdots,MC_{endij},\cdots,MC_{Tij}) \tag{8}$$

其中,MC_{kij} 表示在时间片 k 内服务器 i 与服务器 j 之间的虚拟机迁移开销值。

假设虚拟机从 *start* 时刻开始按照预测出的迁移路径进行迁移,在 *end* 时刻该虚拟机所有可能迁移已经完成,那么可以得到在这段时间内截断的真实开销时间序列。

$$(LC_{start,i},LC_{start+1,i},\cdots,LC_{k,i},\cdots,LC_{end,i}) \tag{9}$$

$$(MC_{start,i,j},MC_{start+1,i,j},\cdots,MC_{kij},\cdots,MC_{end,i,j}) \tag{10}$$

结合开销的真实时间序列,基于动态路径规划可以得到一条基于真实数据的最优迁移路径 *ROMP*,其对应的服务开销记为 *overhead_ROMP*。

该虚拟机执行迁移前的真实服务器运行开销时间序列、真实迁移开销时间序列如式(11)、式(12)所示:

$$(LC_{0i},LC_{1i},LC_{2i},\cdots,LC_{start-1,i}) \tag{11}$$

$$(MC_{0ij},MC_{1ij},MC_{2ij},\cdots,MC_{start-1,ij}) \tag{12}$$

与真实开销时间序列相对的预测开销时间序列如式(13)、式(14)所示:

$$(PLC_{start,i},\cdots,PLC_{ki},\cdots,PLC_{end,i}) \tag{13}$$

$$(PMC_{startij},\cdots,PMC_{kij}\cdots,PMC_{endij}) \tag{14}$$

其中,PLC_{ki} 表示服务器 i 在时间片 k 中的运行开销的预测值,PMC_{kij} 表示服务器 i 与服务器 j 之间的虚拟机在时间片 k 中的迁移开销预测值。

同理,基于式(13)、式(14)可以得到对应的预测最优迁移路径 *POMP*,该路径对应的预测的服务开销记为 *overhead_POMP*。

预测开销与真实开销的差异程度反映了预测窗口的好坏,可以用来评价预测窗口的性能。虚拟机 i 预测开销和真实开销的差记作虚拟机 i 的开销差,如下:

$$\varphi_i(PW)=overhead_POMP_i(PW)-overhead_ROMP_i(PW) \tag{15}$$

对所有迁移的虚拟机带来的系统开销差求和,得到:

$$\psi(PW)=\sum_{i=0}^{N-1}\varphi_i(PW) \tag{16}$$

根据预测时长为 t 的总开销差与预测时间长度为 $t-1$ 的总开销差来决定是否对预测窗口评价值 PW 进行更新。

$$\psi(t-1)>\psi(t) \tag{17}$$

若满足式(17),则用新值 t 来更新 PW。

需要指出的是,将待迁移虚拟机的最长剩余运行周期的长度作为 PW 的上界 PW_upper,因为迁移路径最大长度和虚拟机剩余运行周期的长度是相等的,所以需要的未来时间代价序列值最长为 PW_upper。

自适应预测窗口调整算法的具体步骤如算法 3 所示。

算法 3 自适应预测窗口调整算法

输入：服务器运行虚拟机的开销变化时间序列；虚拟机在服务器之间迁移的开销变化时间序列；虚拟机运行周期的最大值 PW_upper

输出：调整之后的预测预测窗口大小

```
BEGIN
1. FOR t 从 1 到 PW_upper DO
2.    FOR 每一台待迁移的虚拟机 DO
3.       按照当前时刻和虚拟机的关闭时刻截取时间序列分段计算 ROMP
4.       按照虚拟机的开启和关闭时间计算预测的时间序列分段计算 POMP
5.       按照式(15)计算评价函数值 φ(t)
6.    END FOR
7.    将所有的虚拟机计算出来的函数值求和得到 ψ(t)
8.    IF ψ(t−1)>ψ(t) THEN
9.       PW=t
10.   ELSE
11.      break;
12.   END IF
13. END FOR
14. Return PW
END
```

4 实验评估

4.1 数据描述与实验设置

本文设计的基于虚拟机迁移的资源调度机制面向计算集群中的服务器运行开销不断变化的问题场景，采用的数据集来自 PlanetLab 数据集合[40]。PlanetLab 记录了 1000 多台虚拟机每天内 CPU 的占用率数据，数据集记录的时间间隙是 5 min，每天有 288 个样本。随机选择 10 天的数据作为本次实验数据集，数据集的统计信息如表 2 所列。

表 2 PlanetLab 数据集的特征

Table 2 Characteristics of PlanetLab dataset

Date	Numbers of VMs	*Mean*/%	*St. dev*/%
Day1	1052	12.31	17.09
Day2	898	11.44	16.83
Day3	1061	10.70	15.57
Day4	1516	9.26	12.78
Day5	1078	10.56	14.14
Day6	1463	12.39	16.55
Day7	1358	11.12	15.099
Day8	1233	11.56	15.07
Day9	1054	11.54	15.15
Day10	1033	10.43	15.21

本次实验使用 C++语言进行开发，使用 Visual Studio 2013 作为开发工具。实验中，将服务器 CPU 利用率筛选阈值 *cpu_thres* 设为 0.9[41]，将服务器 CPU 利用率采集时间间隔和系统时间片的时间间隔均设为 5 min，*N_slice* 取 12，虚拟机剩余运行时长的筛选阈值 *thres_v* 设为 24 h，*T* 值取该虚拟机剩余时长内划分的时间片(5 min)个数，迁移开销和运行开销在服务开销中的比重均设为 1/2，待调度虚拟机数量依次设置为 50，75，100，125，150，175 和 200。

4.2 基准机制

依据文献[42]中计算虚拟机资源与服务器资源之间的关系度来选择迁移目标服务器的思想。本实验设计了基准机制，即最小代价机制(Smallest Cost，SC)，根据运行迁移算法时的服务器运行开销，将待迁移虚拟机迁移到运行开销最小的服务器上。与本文基于自适应虚拟机迁移(Based on Adaptive Virtual Machine Migration，BAVMM)机制同时考虑多个时间片不同，SC 机制仅仅表现出对下一个时间片的优化。

4.3 实验结果与分析

设定虚拟机请求量的到达服从均匀分布，采样时间间隔为 10 min，记录迁移虚拟机的服务开销，如图 4 所示。可以看出，在一次实验中，随着虚拟机请求的不断到达，BAVMM 机制在服务开销指标上明显优于 SC 机制。

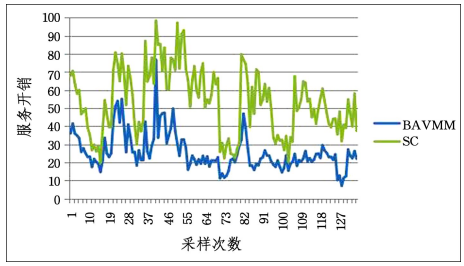

图 4 一次实验中的服务开销对比

Fig. 4 Comparison of service overhead in an experiment

从虚拟机数量的角度对 BAVMM 与 SC 的服务开销指标进行对比。服务开销指标的计算过程如下：1)按照式(2)计算服务开销；2)进行 20 次实验并取平均值；3)将不同虚拟机数量下的服务开销指标同时除以最大值，进行归一化操作。

如图 5 所示，本文设计的基于自适应虚拟机迁移的资源调度机制按照一定的迁移路径实时地将虚拟机进行多次迁移，相比将虚拟机一直固定放置于同一台服务器运行，服务开销更小，而且随着虚拟机数量的增加，这种开销的差距会越来越大。这是因为本文设计面向的问题场景中，服务器的运行开销是随着时间变化的，服务器在某一个时间片中的运行开销较低，并不能保证虚拟机在剩余运行周期内的总服务开销较低。

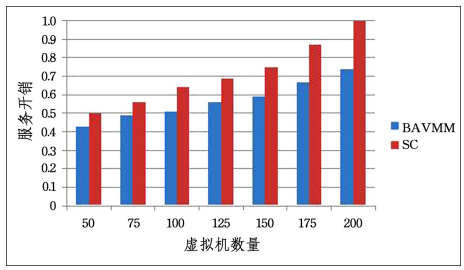

图 5 不同虚拟机数量下服务开销的对比

Fig. 5 Comparison of service overhead under different virtual machine numbers

在本次实验中，为了更清晰地进行开销值的对比，没有开启自适应预测窗口调整算法。自适应预测窗口调整算法是针对所提出的迁移机制进行优化，下面的实验将通过纵向对比来验证该机制的有效性。

首先，通过实验来证明不同的预测窗口大小确实会对 BAVMM 算法的性能产生影响。如图 6 所示，当预测窗口大小为不同的值时，自适应虚拟机迁移算法在服务开销的优化

上具有明显的优劣之分，并且服务开销的差距随着待调度虚拟机数量的增加逐渐扩大，这说明不同的预测窗口大小确实对本机制的性能产生了影响。

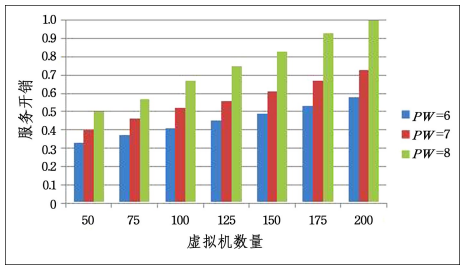

图 6 预测窗口值不同时服务开销的对比

Fig. 6 Comparison of service overhead under different predictive window values

下面对基于自适应虚拟机迁移机制中的自适应预测窗口调整算法进行测试，记不开启自适应预测窗口调整算法的机制为 NAPWA（Non-Adaptive Prediction Window Adjustment），开启自适应预测窗口调整算法的机制为 APWA（Adaptive Prediction Window Adjustment）。

图 7 为开启自适应时间预测窗口大小调整算法与不开启自适应预测窗口大小调整算法的服务开销对比图。可以看出，开启了自适应预测窗口调整算法以后，虚拟机的运行开销相比于不开启时有了显著降低。原因在于不开启自适应时间窗口调整算法时，预测窗口过大或者过小，导致系统运行开销的预测误差较大或者没有充分利用预测数据。

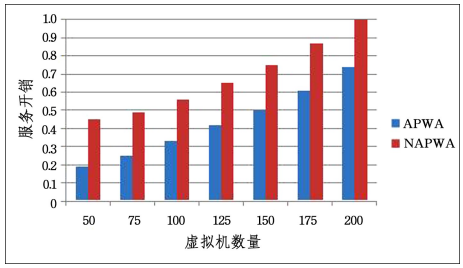

图 7 自适应预测窗口调整算法对服务开销的影响

Fig. 7 Effect of adaptive prediction window adjustment algorithm on service overhead

结束语 云资源调度下的虚拟机迁移一直都是云计算的重要研究方向。本文介绍了基于自适应虚拟机迁移的资源调度机制，首先描述了系统框架，包括虚拟机迁移的总体流程以及系统中的主要模块；然后提出了相关的领域概念，以优化目标为指导，设计了自适应的虚拟机迁移机制，并论述了该机制实施的 4 个阶段。从实验结果来看，与非自适应虚拟机迁移相比，系统性能有了较大提升。然而，本文的虚拟机迁移机制的算法步骤较为复杂，缺少相关实现代价的评估，缺乏实际应用场景的验证，以上问题有待进一步的研究。

参 考 文 献

[1] XU C J, DUAN B Y, XU H P, et al. A Data Center research based on Cloud computing[C]// 2011 International Conference on Computer Science and Network Technology. 2011: 808-811.

[2] CHEN T, GAO X, CHEN G. The features, hardware, and architectures of data center networks: A survey[J]. Journal of Parallel and Distributed Computing, 2016, 96: 45-74.

[3] WANG X W, SUN J J, LI H X, et al. A Reverse Auction Based Allocation Mechanism in the Cloud Computing Environment [J]. Applied Mathematics and Information Sciences, 2013, 7(1): 75-84.

[4] ZHANG X Z, HUANG Z Y, WU C, et al. Online Auctions in IaaS Clouds: Welfare and Profit Maximization With Server Costs [J]. IEEE/ACM Transactions on Networking, 2017, 25(2): 1034-1047.

[5] JIN A L, SONG W, ZHUANG W H. Auction-Based Resource Allocation for Sharing Cloudlets in Mobile Cloud Computing [J]. IEEE Transactions on Emerging Topics in Computing, 2018, 6(1): 45-57.

[6] JIANG C X, CHEN Y, WANG Q, et al. Data-Driven Auction Mechanism Design in IaaS Cloud Computing[J]. IEEE Transactions on Services Computing, 2018, 11(5): 743-756.

[7] WANG X W, WANG X Y, CHE H, et al. An Intelligent Economic Approach for Dynamic Resource Allocation in Cloud Services [J]. IEEE Transactions on Cloud Computing, 2015, 3(3): 275-289.

[8] PILLAI P S, RAO S. Resource Allocation in Cloud Computing Using the Uncertainty Principle of Game Theory[J]. IEEE Systems Journal, 2016, 10(2): 637-648.

[9] ZHANG H Q, XIAO Y, BU S R, et al. Computing Resource Allocation in Three-Tier IoT Fog Networks: A Joint Optimization Approach Combining Stackelberg Game and Matching[J]. IEEE Internet of Things Journal, 2017, 4(5): 1204-1215.

[10] WANG H B, KANG Z L, WANG L. Performance- Aware Cloud Resource Allocation via Fitness-Enabled Auction [J]. IEEE Transactions on Parallel and Distributed Systems, 2016, 27(4): 1160-1173.

[11] AL-TARAZI M, CHANG J M. Performance-Aware Energy Saving for Data Center Networks [J]. IEEE Transactions on Network and Service Management, 2019, 16(1): 206-219.

[12] XU X L, DOU W C, ZHANG X Y, et al. EnReal: An Energy-Aware Resource Allocation Method for Scientific Workflow Executions in Cloud Environment[J]. IEEE Transactions on Cloud Computing, 2016, 4(2): 166-179.

[13] WU D C, WU Q H, XU Y H, et al. QoE and Energy Aware Resource Allocation in Small Cell Networks With Power Selection, Load Management, and Channel Allocation[J]. IEEE Transactions on Vehicular Technology, 2017, 66(8): 7461-7473.

[14] SUN J J, WANG X W, GAO X, et al. Resource Allocation Scheme Based on Neural Network and Group Search Optimization in Cloud Environment [J]. Journal of Software, 2014, 25(8): 1858-1873.

[15] UTTAM M, PULAK C, MASSIMO T, et al. Bandwidth Provisioning for Virtual Machine Migration in Cloud: Strategy and Application[J]. IEEE Transactions on Cloud Computing, 2018, 6(4): 967-976.

[16] JARGALSAIKHAN N, ZHANG H N, HYUK L. Service-A-ware Cloud-to-Cloud Migration of Multiple Virtual Machines [J]. IEEE Access, 2018, 6: 76663-76672.

[17] TAO F, LI C, LIAO T W, et al. BGM-BLA: A New Algorithm for Dynamic Migration of Virtual Machines in Cloud Computing

[J]. IEEE Transaction on Services Computing，2016，9(6)：910-925.

[18] WU X D，ZENG Y Z，LIN G X. An Energy Efficient VM Migration Algorithm in Data Centers[C]// 2017 16th International Symposium on Distributed Computing and Applications to Business，Engineering and Science. 2017：27-30.

[19] SURATH L，SOUHEIL K，JAROSLAW F. Virtual Machine Migration Strategy in Cloud Computing[C]// 2015 14th International Symposium on Distributed Computing and Applications for Business Engineering and Science. 2015：147-150.

[20] ZHANG F，LIU G M，FU X M，et al. A Survey on Virtual Machine Migration：Challenges，Techniques，and Open Issues[J]. IEEE Communications Surveys & Tutorials，2018，20(2)：1206-1243.

[21] KONSTANTIONS T，VASILIS V，MEMA R，et al. Live VM Migration Under Time-Constraints in Share-Nothing IaaS-Clouds[J]. IEEE Transactions on Parallel and Distributed Systems，2017，28(8)：2285-2298.

[22] WALTER C，FLAVIO E. Optimizing Live Migration of Multiple Virtual Machines[J]. IEEE Transactions on Cloud Computing，2018，6(4)：1096-1109.

[23] ZHANG W Z，CHEN Y X，GAO X，et al. Cluster-aware virtual machine collaborative migration in media cloud[J]. IEEE Transactions on Parallel and Distributed Systems，2017，28(10)：2808-2822.

[24] ZHANG J，REN F Y，SHU R，et al. Guaranteeing Delay of Live Virtual Machine Migration by Determining and Provisioning Appropriate Bandwidth[J]. IEEE Transactions on Computers，2016，65(9)：2910-2917.

[25] ENZO B，MICHELE S，ALIREZA M. Fog-Supported Delay-Constrained Energy-Saving Live Migration of VMs Over MultiPath TCP/IP 5G Connections[J]. IEEE Access，2018，6：42327-42354.

[26] UMESH D，DANNY C，STEVEN C，et al. Scatter-Gather Live Migration of Virtual Machines[J]. IEEE Transactions on Cloud Computing，2018，6(1)：196-208.

[27] PRADEEP K T，SANDEEP J. Dynamic Weighted Virtual Machine Live Migration Mechanism to Manages Load Balancing in Cloud Computing[C]// 2016 IEEE International Conference on Computational Intelligence and Computing Research. 2016：1-5.

[28] VEENS R R，SWETHA K，MELODY M. Energy Efficient Traffic-Aware Virtual Machine Migration in Green Cloud Data Centers[C]// 2016 IEEE 2nd International Conference on Big Data Security on Cloud. 2016：268-273.

[29] HOU C W，HUANG C L，DAI H D，et al. Enabling User-Policy-Confined VM Migration in Trusted Cloud Computing[C]// 2016 IEEE 1st International Workshops on Foundations and Applications of Self-Systems. 2016：66-71.

[30] SUHIB M M，ANJALI A，NISHITH G，et al. Minimizing Biased VM Selection in Live VM Migration[C]// 2017 3rd International Conference of Cloud Computing Technologies and Applications (CloudTech). 2017：1-7.

[31] OSANAIYE O，CHEN S，YAN Z，et al. From cloud to fog computing：A review and a conceptual live VM migration framework [J]. IEEE Access，2017，5：8284-8300.

[32] YANG C，GUO Y，HU H，et al. An effective and scalable VM migration strategy to mitigate cross-VM side-channel attacks in cloud[J]. China Communications，2019，16(4)：151-171.

[33] JABALIN R P，S S R，JAYAN J P. A Secure Virtual Machine Migration Using Processor Workload Prediction Method for Cloud Environment[C]// 2016 International Conference on Circuit，Power and Computing Technologies. 2016：1-6.

[34] MEHIAR D，BECHIR H，MOHSEN G，et al. An Energy-Efficient VM Prediction and Migration Framework for Overcommitted Clouds[J]. IEEE Transactions on Cloud Computing，2018，6(4)：955-966.

[35] MEGHA R D，HIREN B P. Efficient Virtual Machine Migration in Cloud Computing[C]// 2015 Fifth International Conference on Communication Systems and Network Technologies. 2015：1015-1019.

[36] SONG Y，PHILIPP W，RAMIN Y，et al. Reliable virtual machine placement and routing in clouds[J]. IEEE Transactions on Parallel and Distributed Systems，2017，28(10)：1-14.

[37] PRAKASH C N，DEEPAK G，ABHISHEK K S，et al. A research paper of existing Live VM Migration and a Hybrid VM Migration approach in Cloud Computing[C]// Proceedings of the 2nd International Conference on Trends in Electronics and Informatics (ICOEI 2018). 2018：720-725.

[38] SUHIB B M，ANJALI A，NISHITH G，et al. Selection Process Approaches in Live Migration：A Comparative Study[C]// 2017 8th International Conference on Information and Communication Systems (ICICS). 2017：23-28.

[39] WANG H，LI Y，ZHANG Y，et al. Virtual machine migration planning in software-defined networks[J]. IEEE Transactions on Cloud Computing，2015(99)：1-15.

[40] FU X，ZHOU C. Predicted affinity based virtual machine placement in cloud computing environments[J]. IEEE Transactions on Cloud Computing，2017(99)：1-1.

[41] ANTON B，RAJKUMAR B. OpenStack Neat：a framework for dynamic and energy-efficient consolidation of virtual machines in OpenStack cloud[J]. Concurrency and Computation：Practice & Experience，2015，27(5)：1310-1333.

[42] ZHENG S P，YU H F，ANAND V，et al. Virtual machine migration techniques managing time-varied workloads[C]// International Conference on Optical Communications & Networks. 2014：1-5.

LI Shuang-gang，born in 1992，master candidate，is a member of China Computer Federation. His main research interests include mobile social cloud and cloud computing.

ZHANG Shuang，born in 1971，Ph.D，associate professor，is a member of China Computer Federation. Her main research interests include mobile social cloud and cloud computing.