

# 云桌面系统虚拟机调度算法

柳 岸<sup>1</sup>, 彭商濂<sup>2</sup>

<sup>1</sup>四川川大智胜软件股份有限公司, 四川 成都

<sup>2</sup>成都信息工程大学计算机学院, 四川 成都

收稿日期: 2024年10月15日; 录用日期: 2024年11月13日; 发布日期: 2024年11月21日

---

## 摘要

当前云桌面系统中虚拟机调度存在效率低下、资源利用率不高、用户体验不佳等问题。针对云桌面系统中虚拟机调度的挑战, 设计了一种基于负载预测和动态调整的虚拟机调度优化算法。首先利用负载预测模型对未来一段时间内虚拟机的负载情况进行预测, 然后采取动态调整策略, 包括虚拟机的迁移和资源调整, 以实现资源的合理分配和利用。实验结果表明, 该算法能够提升云桌面系统的整体性能和用户体验。

## 关键词

云桌面系统, 虚拟机调度, 负载预测, 动态调整, 资源利用率, 数据中心

---

# Virtual Machine Scheduling Algorithm Based on Cloud Desktop System

An Liu<sup>1</sup>, Shanglian Peng<sup>2</sup>

<sup>1</sup>Wisesoft Co. Ltd., Chengdu Sichuan

<sup>2</sup>College of Computer, Chengdu University of Information Technology, Chengdu Sichuan

Received: Oct. 15<sup>th</sup>, 2024; accepted: Nov. 13<sup>th</sup>, 2024; published: Nov. 21<sup>st</sup>, 2024

---

## Abstract

Current virtual machine scheduling in cloud desktop systems suffers from issues such as low efficiency, suboptimal resource utilization, and poor user experience. In response to these challenges, a virtual machine scheduling optimization algorithm based on load prediction and dynamic adjustment is proposed. Firstly, a load prediction model is used to forecast the load of virtual machines in the near future. Then, dynamic adjustment strategies, including virtual machine migration and resource adjustment, are employed to achieve rational resource allocation and utilization. Experimental

---

results demonstrate that this algorithm can enhance the overall performance and user experience of cloud desktop systems.

## Keywords

**Cloud Desktop System, Virtual Machine Scheduling, Load Prediction, Dynamic Adjustment, Resource Utilization Rate, Data Center**

---

Copyright © 2024 by author(s) and Hans Publishers Inc.

This work is licensed under the Creative Commons Attribution International License (CC BY 4.0).

<http://creativecommons.org/licenses/by/4.0/>



Open Access

## 1. 引言

随着信息技术的不断发展，云计算作为一种计算模式，已经在各个领域得到了广泛的应用。其中，云桌面系统作为云计算技术的延伸应用之一，具有诸多优势，如可实现设备无关性、数据集中存储和管理、灵活的资源分配等特点，为用户提供了更加便捷、高效的办公和工作环境。在云桌面系统中，虚拟机调度是一项至关重要的技术，它负责管理和调度各个虚拟机实例的资源分配，直接影响着系统的整体性能和用户体验。

尽管云桌面系统带来了诸多便利，但在实际应用中仍然存在一些挑战和问题。首先，由于用户规模和任务复杂性的不断增加，虚拟机调度的效率受到了一定程度的影响，导致系统响应时间较长、资源利用率不高等问题。其次，当前的虚拟机调度算法普遍缺乏对未来负载情况的准确预测和动态调整能力，无法及时适应系统的变化需求，影响了系统的灵活性和响应性。再次，由于虚拟机调度对资源的合理分配和利用至关重要，但现有算法在资源利用率方面存在一定的局限性，无法充分发挥系统的潜力。

为解决上述问题并进一步提升云桌面系统的性能和用户体验，本文针对云桌面系统中虚拟机调度展开研究，提出一种基于负载预测和动态调整的虚拟机调度优化算法。该算法结合了负载预测模型和动态调整策略，旨在实现对系统负载的准确预测和灵活调整，从而优化资源分配、提高资源利用率，并改善用户体验。

## 2. 相关工作

针对物理主机工作负载的不确定性可能导致的过载和资源利用率低的问题，[1]提出一种基于负载不确定性的虚拟机整合(WU-VMC)方法被提出。该方法通过分析物理主机的工作负载记录和虚拟机资源请求的历史数据，以解决数据中心能源消耗和服务质量的问题。通过这种方式，该方法能够稳定数据中心各主机的工作负载，并解决负载不确定性可能带来的能耗增加和服务质量下降等问题。[2]提出虚拟机亲和度概念，综合考虑物理机负载均衡度，结合罚函数法思想，设计了一种基于多种群遗传算法的优化部署策略，以解决虚拟机间的亲和和互斥关系。[3]在研究中使用了线性回归来检测主机的重载和轻载状态，采用动态虚拟机迁移和节点睡眠切换等方法进行动态整合，以优化资源利用和能源消耗，但并未采用机器学习方法对虚拟机负载进行预测。[4]提出了一种负载均衡方法，称为预分区(Prepartition)，通过将 VM 请求划分为多个子请求并在开始时间、结束时间和容量需求上进行调度，来实现预定义的负载均衡目标。[5]提出基于卷积神经网络(CNN)的多类型负载联合预测方法(TSF)。该方法能够自动化构建关键训练样本并充分挖掘时序特征和空间特征，从而在考虑多种虚拟机负载情况下，降低训练和预测时间成本，同时提高预测精度。[6]提出了一种基于三支决策的虚拟机迁移调度策略(TWD-VMM)。该策略结合了分、治、

效模型，通过建立层次阈值树来搜索所有可能的阈值，以优化数据中心的能耗。根据不同负载情况，对高、中、低负载主机采取不同的迁移策略，包括多维资源均衡度、主机负载下降幅度等目标。[\[7\]](#)提出了一种基于逻辑分区的云虚拟机负载均衡优化算法。该算法使用逻辑分区来划分不同分区的虚拟机负载和业务形式，并以平均任务完成时长为优化目标。通过搭建资源调度数学模型，将多维资源协调问题转换为单维协调问题，并利用遗传算法对虚拟机资源进行编码与解码，实现负载均衡优化。[\[8\]](#)研究建立了云端能量消耗与在线虚拟机迁移时间消耗的数学模型，并通过全局遗传算法来优化虚拟机迁移和放置策略。该模型在虚拟机迁移时考虑了更多的问题维度，使得虚拟机的分配和迁移更加合理和公平。[\[9\]](#)提出了一种基于机器学习的虚拟机替换策略，旨在平衡主机(HM)上的负载。该策略通过学习代理在每个学习周期中选择和执行最佳动作，从而优化虚拟机的分配与迁移，以达到降低能耗、提高资源利用率等目标。[\[10\]](#)提出了一种基于负载均衡算法的优化方法，旨在提高资源利用率、改善服务交付性能，并满足服务级别协议(SLA)的要求。该算法考虑了任务参数、虚拟机优先级和资源分配等因素，通过优化任务调度来实现负载均衡。

本文针对当前云桌面系统中虚拟机调度存在的效率低下、资源利用率不高以及用户体验不佳等问题，设计了一种基于负载预测和动态调整的虚拟机调度优化算法。首先，我们通过建立负载预测模型对未来一段时间内虚拟机的负载情况进行准确预测，以便在提前进行资源分配和调整。该预测模型考虑了历史数据、工作负载记录以及虚拟机资源请求等因素，能够有效应对负载不确定性带来的挑战。其次，我们采取了动态调整策略，包括虚拟机的迁移和资源调整。具体来说，我们设计了一套灵活的虚拟机迁移方案，根据负载预测结果和实时监测数据，及时将负载过重或过轻的虚拟机进行迁移，以实现资源的合理分配和利用。同时，我们也考虑了对主机资源进行动态调整，以进一步优化资源利用效率。

### 3. 模型描述

本模板仅针对采用 A4 纸型的论文版式。请务必确保您的论文采用 A4 幅面(21 厘米 × 28.5 厘米)进行排版。假设云计算系统由多个计算设备组成，包括 CPU、内存和存储器等。这些资源通过虚拟机向客户提供服务，云系统能够实时调度多个虚拟机以支持并发服务。设有  $n$  台物理主机(host) ( $H_1, H_2, \dots, H_n$ ) 和  $m$  个虚拟机( $VM_1, VM_2, \dots, VM_m$ )需要被调度。设每个虚拟机  $VM_j$  的资源需求为  $R_j$ ，其中  $j = 1, 2, \dots, m$ 。

#### 3.1. 虚拟机调度模型

假设云计算系统由多个计算设备组成，包括 CPU、内存和存储器等。这些资源通过虚拟机向客户提供服务，云系统能够实时调度多个虚拟机以支持并发服务。设有  $n$  台物理主机(host) ( $H_1, H_2, \dots, H_n$ ) 和  $m$  个虚拟机( $VM_1, VM_2, \dots, VM_m$ )需要被调度。设每个虚拟机  $VM_j$  的资源需求为  $R_j$ ，其中  $j = 1, 2, \dots, m$ 。

假设  $x_{ij}$  是一个二元变量，表示虚拟机  $VM_j$  被分配到主机  $H_i$  上，则虚拟机调度的模型如公式(1)所示：每个虚拟机只能被分配到一个主机：

$$\text{目标函数: } \max \sum_{i=1}^n \sum_{j=1}^m x_{ij} * R_j$$

$$s.t. \begin{cases} \text{每个虚拟机只能被分配到一个主机: } \sum x_{ij} = 1, \forall j \\ \text{主机的资源容量不能超过限制: } \sum_{j=1}^m x_{ij} * R_j \leq C_i, \forall i \\ \text{虚拟机分配变量为二元变量: } x_{ij} \in \{0,1\} \end{cases} \quad (1)$$

其中, 目标函数的目标是最大化资源利用率, 约束条件包括每个虚拟机只能被分配到一个主机, 每个主机的资源容量不能超过限制, 而虚拟机的分配变量  $x_{ij}$  是一个二元变量, 表示虚拟机  $VM_j$  是否被分配到主机  $H_i$  上。

### 3.2. 虚拟机负载预测模型

本模板可直接用于论文及其文字的编排, 有的页边距、行距、字体都严格符合规定, 请勿修改! 尤其是页边距, 由于期刊在后期制作过程中需要在页眉、页脚添加各种信息, 所以所有论文务必确保现有的页边距不被修改, 页面空白不被占用。

假设我们有一个包含  $N$  个历史虚拟机调度样本的训练集, 每个样本有两个特征(特征 1: CPU 利用率, 特征 2: 内存使用情况), 表示为  $\{(x_1^{(i)}, x_2^{(i)})\}_{i=1}^N$ , 对应的目标变量为虚拟机的负载, 表示为  $\{y^{(i)}\}_{i=1}^N$ 。

假设虚拟机的负载  $y^{(i)}$  与特征  $(x_1^{(i)}, x_2^{(i)})$  之间存在线性关系, 即:

$$y^{(i)} = \theta_0 + \theta_1 * x_1^{(i)} + \theta_2 * x_2^{(i)} + \epsilon^{(i)} \quad (2)$$

其中,  $\theta_0$  是模型的截距,  $\theta_1$  和  $\theta_2$  是特征系数,  $\epsilon^{(i)}$  是误差项。我们的目标是通过训练数据来估计模型的参数  $\theta_0$ ,  $\theta_1$  和  $\theta_2$ , 使得模型能够尽可能准确地预测未来虚拟机的负载。

训练模型的过程就是找到使得模型预测值  $\widehat{y}^{(i)}$  与真实值  $y^{(i)}$  之间误差  $\epsilon^{(i)}$  最小化的参数  $\theta_0$ ,  $\theta_1$  和  $\theta_2$ 。使用最小二乘法来估计参数, 即最小化误差的平方和:

$$J(\theta_1, \theta_0, \theta_2) = \frac{1}{2N} \sum_{i=1}^N (\widehat{y}^{(i)} - y^{(i)})^2 \quad (3)$$

其中  $\widehat{y}^{(i)}$  是模型对样本  $(x_1^{(i)}, x_2^{(i)})$  的预测值, 可以表示为:

$$\widehat{y}^{(i)} = \theta_0 + \theta_1 * x_1^{(i)} + \theta_2 * x_2^{(i)} \quad (4)$$

最小二乘法的目标为找到使得  $J(\theta_1, \theta_0, \theta_2)$  最小化的参数  $\theta_0$ ,  $\theta_1$  和  $\theta_2$ 。本文使用梯度下降等优化算法来实现参数的估计。该模型描述了虚拟机负载与 CPU 利用率、内存使用情况之间的线性关系, 通过历史数据的学习, 可以预测未来虚拟机的负载情况, 从而进行资源分配和调整。

## 4. 动态虚拟机调度算法

在云计算环境中, 虚拟机负载的变化可能受多种因素影响, 包括用户请求量的变化、应用程序的运行状态变化、硬件故障等。这些因素导致虚拟机在不同时间段的负载水平不同, 从而影响了资源的合理分配和利用。传统的静态虚拟机调度策略无法有效应对这种负载变化, 可能导致资源过载或空闲, 影响系统性能和资源利用率。

因此, 虚拟机动态调整策略旨在通过实时监控和负载预测, 及时发现虚拟机负载的变化趋势, 并采取相应的措施进行调整, 以保证系统资源的合理分配和利用。其核心目标是优化系统性能、提高资源利用率, 并确保服务质量符合预期水平。结合虚拟机负载预测模型的动态虚拟机调度算法描述如图 1 所示。

首先, 在开始执行算法之前, 需要设定监控周期的长度, 并加载负载预测模型, 以便在算法执行过程中使用(算法第 2、3 行)。

在每个监控周期开始时, 算法会获取实时的虚拟机监控数据, 包括每个虚拟机的当前负载情况、资源使用情况等(算法第 6 行)。

然后使用加载的负载预测模型对未来一段时间内虚拟机的负载情况进行预测, 得到预测负载水平(算法第 7 行)。

将预测负载水平与当前实时的虚拟机监控数据结合起来, 形成完整的负载信息, 用于后续的决策(算法第 12 行)。

根据预测负载情况, 判断虚拟机是否负载过重或过轻, 然后相应地计划虚拟机的迁移或资源配置调整, 以实现资源的合理分配和利用(算法第 14~19 行)。

假设负载预测模型的计算复杂度为  $O(f(n))$ , 其中  $n$  是输入数据的规模。对虚拟机的实时监控数据和负载预测结果进行处理和结合的计算复杂度为  $O(g(m))$ , 其中  $m$  是监控数据的规模。调度算法中执行调整计划的计算复杂度为  $O(h(p))$ , 其中  $p$  是执行调整计划所涉及的数据规模。综合起来, 调度算法的整体计算复杂度为  $O(\max(f(n), g(m), h(p)))$ , 即取决于三个步骤中最高的计算复杂度。

---

**Algorithm 1: Dynamic VM Adjustment Algorithm with Load Prediction**

---

**Data:** 虚拟机监控数据  $VM\_Data$ , 负载预测模型  $Load\_Prediction\_Model$

**Result:** 调整后的虚拟机资源配置或迁移计划

```

1 初始化监控周期  $T$ ;
2 加载负载预测模型;
3 设置监控周期  $T$ ;
4 while 每个监控周期 do
    5   获取实时的虚拟机监控数据  $VM\_Data$ ;
    6   利用负载预测模型  $Load\_Prediction\_Model$  预测未来一段时间内
        虚拟机的负载情况;
    7   foreach 虚拟机  $VM$  do
        8       利用负载预测模型  $Load\_Prediction\_Model$  预测未来负载水
           平  $Load\_Prediction\_VM$ ;
        9       结合当前监控数据  $VM\_Data$  和预测结果
            $Load\_Prediction\_VM$ ;
       10      if  $Load\_Prediction\_VM$  负载过重 then
           11          计划将虚拟机  $VM$  迁移到目标主机  $Target\_Host$  或调整
              资源配置;
           12      end
       13      else if  $Load\_Prediction\_VM$  负载过轻 then
           14          考虑调整虚拟机  $VM$  资源配置;
           15      end
       16   end
       17   执行调整计划;
       18   更新负载预测模型  $Load\_Prediction\_Model$ ;
  19 end

```

---

**Figure 1.** Dynamic virtual machine scheduling algorithm

**图 1.** 动态虚拟机调度算法

## 5. 实验结果分析

### 5.1. 实验环境

为了评估动态虚拟机调整算法的有效性, 我们使用 CloudSim [11] 进行模拟实验。在实验中, 我们建立了一个包含 100 台异构主机和 100 台虚拟机的数据中心模型。每个节点都具有单核 CPU、32 GB 内存、

20 GB/s 网络带宽和 2 TB 的存储空间。

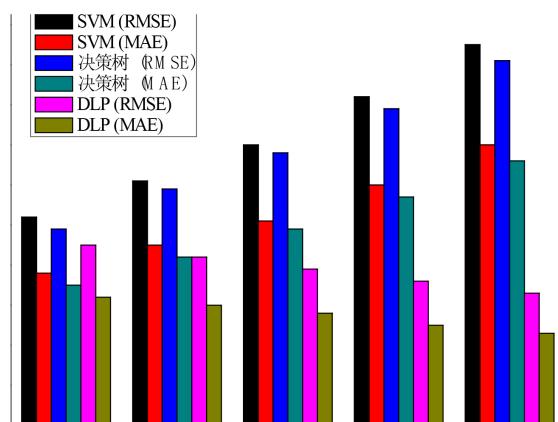
实验模拟了超高清视频融媒体平台的场景，支持 4 K 视频格式，包括  $3840 \times 2160$  4 K UHD、 $4096 \times 2160$  4 K 和  $4096 \times 2304$  分辨率。这些视频格式需要高度的资源支持和处理能力，在平台上的负载具有高度的变动性和波动性，需要及时的资源调整和负载均衡来保证视频播放和处理的顺畅性和稳定性。

实验采用了动态虚拟机调整算法来实现负载均衡。该算法根据实时监测数据和负载预测模型，及时对负载过重或过轻的虚拟机进行迁移，并对主机资源进行动态调整，以优化资源利用效率。我们设置了不同的视频负载场景，包括高峰时段、低负载时段以及视频突发事件等情况，以评估算法在不同场景下的适应性和效果。

## 5.2. 负载预测模型的性能评估

实验采用了动态虚拟机调整算法来实现负载均衡。该算法根据实时监测数据和负载预测模型，及时对负载过重或过轻的虚拟机进行迁移，并对主机资源进行动态调整，以优化资源利用效率。我们设置了不同的视频负载场景，包括高峰时段、低负载时段以及视频突发事件等情况，以评估算法在不同场景下的适应性和效果。

实验首先对动态预测算法(DLP)进行性能测试，对比算法为基于 SVM 和决策树的决策算法。三种算法对超高清视频融媒体平台的历史负载数据进行训练，然后对该平台的负载进行预测。**图 2** 显示了三种算法在均方根误差(Root Mean Square Error, RMSE)和平均绝对误差(Mean Absolute Error, MAE)的性能对比。



**Figure 2.** Comparison of RMSE and MAE of three prediction algorithms

**图 2.** 三种预测算法的 RMSE 和 MAE 比较

可以看到 DLP (动态负载预测模型) 在 RMSE (均方根误差) 和 MAE (平均绝对误差) 方面表现得更好，具有较好的性能。DLP 可能综合考虑了历史数据、工作负载记录以及虚拟机资源请求等多种因素，使得模型更加全面和准确。DLP 采用了动态调整策略，及时对负载过重或过轻的虚拟机进行迁移和资源调整，能够更好地实现资源的合理分配和利用，从而降低了预测误差。

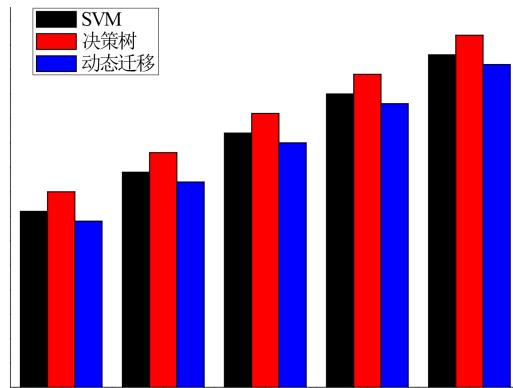
## 5.3. 虚拟机迁移性能评估

实验选取了在超高清视频融媒体平台中典型的工作负载数据集，包括不同分辨率的视频处理任务和多媒体数据传输任务。针对这些任务，实验采用了 SVM 和决策树两种算法进行虚拟机调度和迁移方案的对比。在对比过程中，主要关注任务处理延迟、系统资源利用率、虚拟机迁移次数。

### 5.3.1. 任务处理延迟对比

为测试本文提出的动态虚拟机迁移算法与 SVM [12]和决策树[13]的虚拟机迁移延迟，我们在模拟环境中生成了分别生成了 20~100 个视频编码任务。由于在超高清视频融媒体平台中，虚拟桌面的视频编码任务将耗费较大的存储和计算资源，导致虚拟机进行迁移以适应编码需求，实验对比数据如图 3 所示。

从图 3 可以看出，基于 SVM 的迁移算法的延迟随着任务数量的增加而逐渐增加，呈现出线性增长的趋势。这是因为 SVM 算法在处理大量任务时需要更多的计算资源和时间，导致延迟逐步增加。



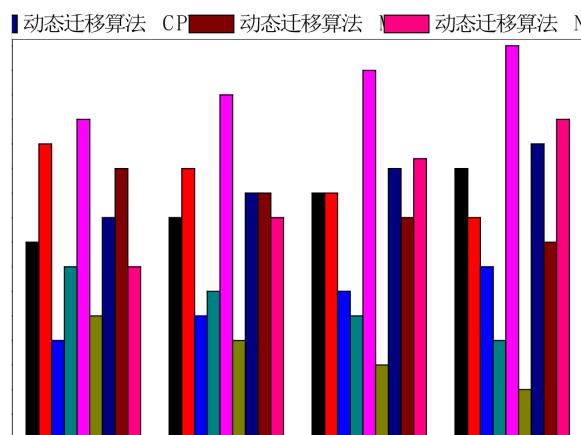
**Figure 3.** Comparison of latency of three migration algorithms  
**图 3.** 三种迁移算法的延迟比较

决策树算法的延迟也随着任务数量的增加而增加，但增长速度略慢于 SVM 算法。决策树算法在处理任务时对任务之间的关联性较好，因此在一定程度上能够减少延迟增长速度。

动态虚拟机迁移算法的延迟相对较低且随着任务数量的增加而增加的速度相对较慢。这表明该算法在处理任务时能够更高效地进行虚拟机的调整和迁移，从而减少了整体延迟。

### 5.3.2. 系统资源利用率对比

我们生成了四组融媒体平台的视频编码解码任务 T1, T2, T3 和 T4，分别测试了三种迁移算法的资源利用率，包括 CPU、内存和网络带宽，性能对比如图 4 所示。



**Figure 4.** Comparison of resource usage (CPU, memory and network bandwidth) of three migration algorithms  
**图 4.** 三种迁移算法资源利用率(CPU、内存、网络带宽)比较

从图 4 可以看出，决策树的内存利用率较好，动态迁移算法的 CPU 和网络带宽利用率整体较好，原

因在于动态迁移算法可以实时监测系统的负载情况，并根据需要动态调整虚拟机的资源分配。当系统负载较高时，可以通过迁移虚拟机来平衡负载，从而有效利用CPU和网络带宽资源。

动态迁移算法通常会考虑到各个物理主机的负载情况和资源利用率，通过智能的负载均衡策略来优化资源分配。这种方式可以避免资源过度集中于某些主机，提高整体资源利用率。

## 6. 结语

针对云桌面系统中虚拟机调度存在的效率低下、资源利用率不高、用户体验不佳等问题，设计并实现了一种基于负载预测和动态调整的虚拟机调度优化算法。该算法通过预测未来一段时间内虚拟机的负载情况，并结合动态调整策略，包括虚拟机的迁移和资源调整，成功地实现了资源的合理分配和利用。实验结果表明，该算法在整体性能和用户体验方面都取得较大提升。虚拟机的负载预测准确度较高，动态调整策略能够有效地调整资源分配，使系统运行更加稳定顺畅。用户的体验也得到了改善，响应速度更快、服务质量更可靠。

然而，论文的工作还存在一些局限性，例如实验规模较小、场景限制较多等。未来的研究可以进一步扩大实验规模，优化算法策略，提高系统的自适应性和智能化程度，以满足日益增长的云桌面系统需求。

## 基金项目

本文受四川省科技厅重点研发项目(2023YFG0144)支持。

## 参考文献

- [1] 李双俐, 李志华, 喻新荣, 等. 基于负载不确定性的虚拟机整合方法[J]. 计算机应用, 2018, 38(6): 1658.
- [2] 董浩, 李烨. 基于多种群遗传算法的虚拟机优化部署研究[J]. 控制工程, 2020, 27(2): 335.
- [3] Beloglazov, A. and Buyya, R. (2011) Optimal Online Deterministic Algorithms and Adaptive Heuristics for Energy and Performance Efficient Dynamic Consolidation of Virtual Machines in Cloud Data Centers. *Concurrency and Computation: Practice and Experience*, **24**, 1397-1420. <https://doi.org/10.1002/cpe.1867>
- [4] 田文洪, 徐敏贤, 周光耀, 等. 云数据中心的基于虚拟机预约和预分割的负载均衡方法[J]. 计算机科学技术学报, 2023, 38(4): 773-792.
- [5] 余显, 李振宇, 张广兴, 等. 基于卷积神经网络的虚拟机多类型负载联合预测方法[J]. 高技术通讯, 2020, 30(9): 884-892.
- [6] 杨翎, 姜春茂. 基于三支决策的虚拟机节能迁移策略[J]. 计算机应用, 2021, 41(4): 990-998.
- [7] 舒晓苓, 吴雪琴. 云计算网络下虚拟机负载均衡方法仿真[J]. 计算机仿真, 2022, 39(3): 358-361, 412.
- [8] 刘开南. 云数据中心基于遗传算法的虚拟机迁移模型[J]. 计算机应用研究, 2020, 37(4): 1115-1118.
- [9] Ghasemi, A. and Toroghi Haghigat, A. (2020) A Multi-Objective Load Balancing Algorithm for Virtual Machine Placement in Cloud Data Centers Based on Machine Learning. *Computing*, **102**, 2049-2072. <https://doi.org/10.1007/s00607-020-00813-w>
- [10] Mosa, A. and Paton, N.W. (2016) Optimizing Virtual Machine Placement for Energy and SLA in Clouds Using Utility Functions. *Journal of Cloud Computing*, **5**, Article No. 17. <https://doi.org/10.1186/s13677-016-0067-7>
- [11] Buyya, R., Ranjan, R. and Calheiros, R.N. (2009) Modeling and Simulation of Scalable Cloud Computing Environments and the CloudSim Toolkit: Challenges and Opportunities. 2009 International Conference on High Performance Computing & Simulation, Leipzig, 21-24 June 2009, 1-11. <https://doi.org/10.1109/hpcsim.2009.5192685>
- [12] Tseng, F., Chen, X., Chou, L., Chao, H. and Chen, S. (2014) Support Vector Machine Approach for Virtual Machine Migration in Cloud Data Center. *Multimedia Tools and Applications*, **74**, 3419-3440. <https://doi.org/10.1007/s11042-014-2086-z>
- [13] Moreno, I.S., Yang, R., Xu, J. and Wo, T. (2013). Improved Energy-Efficiency in Cloud Datacenters with Interference-Aware Virtual Machine Placement. 2013 IEEE Eleventh International Symposium on Autonomous Decentralized Systems (ISADS), Mexico City, 6-8 March 2013, 1-8. <https://doi.org/10.1109/isads.2013.6513411>