

DOI:10.20033/j.1003-7241.(2026)10-0149-04

基于容器技术的虚拟化桌面程序集中调控方法

陈又咏, 庄 莉, 蔡清远, 张富林, 林梅芳

(福建亿榕信息技术有限公司, 福州 350003)

摘 要:为提高虚拟化桌面程序调控效率、精准同步执行任务,提出基于容器技术的虚拟化桌面程序集中调控方法。建立并发多媒体访问调控模型,对初始信息进行预处理,构建泊松分布序列;计算平均请求速度及响应时间参数,根据用户数量与二者之前的参数关系得出访问调控参数;设立有关虚拟化桌面程序的数据集群,计算前驱任务到达后驱调控机制的调度时间,推导不同群组的任务执行时间周期,调节周期参数,实现集中调控。仿真实验证明,在集中调控平均等待时间对比中,所提方法曲线斜率最小,调控速度最快;在平均排队队长对比中,当作业数量达到 250 时,所提方法的队列值分别比其他两种方法少 22 和 12。所提方法均衡把控效果好,调控反馈及口令收敛速度快,符合现代虚拟化资源的特征,能够满足正常工作交流需求。

关键词:集中调控;虚拟化桌面;泊松分布序列;流域通道;周期序列;容器技术

中图分类号: TP391.9;TP273

文献标志码: A

文章编号: 1003-7241(2026)10-0149-04

Centralized regulation method of virtualization desktop program based on container technology

Chen Youyong, Zhuang Li, Cai Qingyuan, Zhang Fulin, Lin Meifang

(FuJian Yirong Information Technology Co., Ltd., Fuzhou 350003, China)

Abstract: To enhance the efficiency of virtualized desktop program regulation and achieve precise synchronous task execution, a container-based centralized regulation method for virtualized desktop programs is proposed. A concurrent multimedia access regulation model is established, preprocessing initial information to construct a Poisson distribution sequence. The average request speed and response time parameters are calculated, and access regulation parameters are derived based on the relationship between user count and these parameters. A data cluster related to virtualized desktop programs is set up, computing the scheduling time of predecessor task arrival and successor regulation mechanisms, deducing the task execution time cycles of different groups, and adjusting cycle parameters to achieve centralized regulation. Simulation experiments demonstrate that in the comparison of average waiting times under centralized regulation, the proposed method exhibits the smallest curve slope and the fastest regulation speed. In the comparison of average queue lengths, when the number of tasks reaches 250, the proposed method's queue values are 22 and 12 less than those of the other two methods, respectively. The proposed method achieves balanced control with effective feedback and rapid password convergence, aligning with the characteristics of modern virtualized resources and meeting normal work communication requirements.

Keywords: centralized regulation; virtualization desktop; Poisson distribution sequence; watershed channel; periodic sequence; container technology

为适应社会,需要对计算机频繁更新换代,减少网络资源浪费,调控技术应运而生且取得了极佳的均衡把控效果。调控技术的主要思路是:搜索全局虚拟化程序进行集中管理,优化改良虚拟化桌面控制模式。国内外的研究方向主要着重于资源作业处理方面,通过高码率的程序搜索实现规划管控。Citrix 公司曾提出一个桌面虚拟化解决方案,可提供端到端的桌面交付,用户登录后可获得一个全新的桌面,且其性能不会发生改变。国内着重客户端与操控端调控的高度统一,主要改善内部资源浪费、虚拟化桌面视觉体验感差以及程序分布混乱的问题。文献[1]提

出的基于博弈论的虚拟化空间数据云资源调控分配方法。通过分析个人和集体的虚拟化空间程序资源之间的不同,提取出对应集中调控特点,综合考虑得到全局最大满意度,构建基于博弈论的调控模型,结合最优空间中心,将目标程序逐一分配到空间中心。但方法全局搜数据索耗用过大,没有考虑到资源分配不均问题,适用性及可靠性较差;文献[2]则采用云计算技术来实现虚拟化信息资源的权限调控,根据信息属性的分级分类特性,建立多个调控子集,利用分类树的拓扑关系对子集进行范畴归类,以密钥作为相应范畴内程序调控手段,但方法划分的子集存在

收稿日期:2025-09-10;录用日期:2025-10-30

作者简介:陈又咏(1978—),男,工程师,研究方向为信息技术应用创新、电力信息化等。

引用本文:陈又咏,庄莉,蔡清远,等.基于容器技术的虚拟化桌面程序集中调控方法[J].自动化技术与应用,2026,45(10):149-152.(Chen Youyong, Zhuang Li, Cai Qingyuan, et al. Centralized regulation method of virtualization desktop program based on container technology[J]. Techniques of Automation and Applications, 2026, 45(10): 149-152.)

不确定性较强、调控精准度较低,性能表现差。

综合上述问题,本文提出一种基于容器技术的虚拟化桌面程序集中调控方法,有效改善程序分配资源不均及调控性能较差的问题。容器技术可以将多个用户的数据资源划分到单个虚拟化容器中,方便归类管理,以便于更好处理孤立组程序之间的冲突性,使得虚拟化程序更具有平衡性。利用终端数据流访问调控模式,将初始虚拟化桌面内程序进行过滤、分类处理,减少因原始数据自身异常的影响,建立异构调度执行模型,寻找虚拟程序作业的最优状态,在该状态下实行调控后命令精准度最高、反馈效率最好。

1 虚拟化桌面程序访问调控

为防止多用户在同时使用虚拟化桌面时,发生程序碰撞及访问错误问题,本文利用容器技术,建立并发多媒体访问调控模型,让用户信息形成一种循环的调控 Stream 流。结合容器技术对多个程序组的划分方式,将待处理的用户访问信息集合看作 Stream 流中的分支流,在网络信道中传输,以各节点作为筛选、分类、过滤、排序、聚合、回击等处理中心,访问信息流经整理后,由 Terminal Operation(终端操作)获取最终处理结果^[3-5],大大提高了虚拟化桌面程序的调控效率。

以用户访问请求到达接收点的时间为依据,逐步过滤调控,使用户的请求访问时间符合泊松分布^[6-7],当 Stream 流在同一时刻接受到大量用户请求时,令单位请求的时间间隔遵循指数分布,使得请求接收速率和发送速率保持一致,表达式为

$$\xi = \frac{N}{T_{\text{serv}}} \quad (1)$$

式中, ξ 表示用户访问请求到达 Stream 流中的平均速率; N 表示访问用户数量; T_{serv} 表示虚拟云桌面访问时间参数,该时间参数服从指数 μ (访问请求平均速率) 分布,此时,则有

$$\mu = \frac{1}{P_n(T_{\text{serv}})} \quad (2)$$

式中, P_n 表示虚拟化桌面访问请求队列模型 $M/M/1$ 的稳态概率, n 为当前整个系统内的总任务数。请求队列中用户的请求数量为 χ , 根据稳态流量关系式,进行迭代递推计算可得

$$\mu = \frac{1}{P_n(T_{\text{serv}})}, P_n = (1 - \rho)\rho^\chi, \chi = 0, 1, \dots, n \quad (3)$$

式中, ρ 表示符合稳态流^[6] 关系的请求速率, $(1 - \rho = \frac{\chi}{\chi\mu})$, $\chi\mu$ 表示访问请求的平均处理速度,以速率值 ρ 为判定介点,当用户对桌面程序访问请求高于该值时,就可立即执行回击处理,提高安全性、降低威胁系数^[8]。

2 虚拟化桌面集中调控算法

通过容器技术对虚拟化桌面程序的归类与访问控制,

假设虚拟化云桌面是由 m 个异构程序资源集合构成, $R = \bigcup_{j=1}^m \{R_j\}$, 其中, R_j 表示给定的第 j 个异构程序资源数据。本文将虚拟化程序的调控问题转换为资源任务流分配问题,减少调控的所需的经济代价^[9]。设定桌面程序第 i 个调控任务为 A_i , $\text{sched}(A_i)$ 表示程序调控任务 A_i 的执行资源,对该资源进行以下定义。

1) 统一虚拟程序调控的周期时间。

用 $t(A_i, R_j)$ 表示任务在资源项目 $R_j = \text{sched}(A_i)$ 上的计算时间与接收机制接收到前驱任务 $A_p \in \text{pred}(A_i)$ 的时间之和,其表达式为

$$t(A_i, R_j) = \max_{A_p \in \text{pred}(A_i)} \frac{\text{data}_{pi}}{b_{pj}} + \frac{\text{workload}(A_i)}{s_j} \quad (4)$$

式中, p 为前驱任务的索引编号, data_{pi} 表示调控任务 A_i 与 A_p 之间的信息传输数量; b_{pj} 表示调控任务 A_p 的资源与执行资源 R_j 之间的数据带宽^[10-11]; $\text{workload}(A_i)$ 表示调控任务 A_i 的请求指令长度; s_j 表示执行资源 R_j 的传输速率。分析调控任务 A_i 完成时间 T_{A_i} 与前驱执行时间之间的周期关系,表达式为

$$T_{A_i} = \begin{cases} t(A_i, \text{sched}(A_i)), \text{pred}(A_i) = \emptyset \\ \max_{A_p \in \text{pred}(A_i)} \{T_{A_p} + t(A_i, \text{sched}(A_i))\} \\ \text{pred}(A_i) = \emptyset \end{cases} \quad (5)$$

将调控任务的执行时间定义为全部任务中的最大时间 T_w 可得

$$T_w = \max_{\varphi \in [1, n]} T(A_i, \text{sched}(A_i)) \quad (6)$$

式中, W 表示全体任务集合, φ 表示调控任务执行时间的临界点参数,根据该参数就可统一调控任务指令的执行和到接收机制的抵达时间,提高了整体的稳定性^[12]。

2) 统一虚拟程序调控的经济代价。

在虚拟化桌面执行集中调控任务时,必然会付出经济代价,主要包括信息数据的储存代价 $C^{(\text{data})}$ 及传输代价 $C^{(\text{comp})}$ ^[13-14]。

用 $C_{(A_i, R_j)}^{(\text{data})}$ 表示调控任务 A_i 在执行资源 R_j 上进行数据输入 $\text{in}(A_i)$ 、数据输出 $\text{out}(A_i)$ 以及数据储存 $\text{data}(A_i)$ 时需要的代价,公式为

$$C_{(A_i, R_j)}^{(\text{data})} = \text{data}(A_i) \times t(A_i, R_j) \times PS_{R_j} + \text{in}(A_i) \times PI_{R_j} + \text{out}(A_i) \times PO_{R_j} \quad (7)$$

式中, P 表示单位价格; S_{R_j} 为执行资源 R_j 的单位数据存储价格; I_{R_j} 为执行资源 R_j 的单位数据输入价格; O_{R_j} 为执行资源 R_j 的单位数据输出价格。用 $C_{R_j}^{(\text{comp})}$ 表示执行资源 R_j 的调控经济代价,用 $t_{A_i}^{(\text{start})}$ 表示包含经济代价的任务开始时间;用 $t_{A_i}^{(\text{end})}$ 来表示包含经济代价的任务结束时间,计算公式为

$$C_{R_j}^{(\text{comp})} = t_{A_i}^{(\text{start})} + t(A_i, R_j) + \max_{A_i \in \text{pred}(A_p)} \frac{\text{data}_{iq}}{b_{jq}} \quad (8)$$

式中, q 表示任务索引。

假设,数据输入 $\text{in}(A_i)$ 与数据输出 $\text{out}(A_i)$ 的时间处

于 $t_{A_i}^{(start)}$ 与 $t_{A_i}^{(end)}$ 之间。建立执行资源 R_j 的调控任务集合为 $\{J_1, J_2, \dots, J_p\}$, P 表示任务数量 $P < n$ 且 $\text{sched}(J_i) = R_j$, J_i 表示被调度到执行资源 R_j 上的第 i 个调控任务, $i \in [1, P]$ 。将调控任务运动时间关系排列: $t_{J_1}^{(start)} < t_{J_2}^{(start)} < \dots < t_{J_p}^{(start)}$ 。基于上述排列,可将所有的任务资源划分到不同的群组内,对同一组内的全部任务数据连续执行调控,此时,群组中最先开始的任务完成后,所有资源代价都可释放^[15],在一定程度上减轻了调控负担,使得调控结果具有实时性和有效性。

3 仿真实验

3.1 实验背景

为验证本文调控算法的有效性及其可靠性,以 Hadoop (分布式系统基础架构) 为基础,选择其中分布式文件系统(hadoop distributed file system, HDFS)作为本次实验的测试平台,在该系统上开展一系列的虚拟化桌面程序调控实验。HDFS 具有的容错性高、数据吞吐量大特点,可满足仿真实验数据基数大、异构数据较多的条件。

与基于博弈论的空间数据资源分配方法、虚拟化资源的访问权限调控方法进行对比,开展对比实验,本文选择了 Vmware(虚拟机)、Citrix Workspace(虚拟化应用程序)、Type-1(虚拟机管理程序)的虚拟化程序作业作为实验用例,关键参数配置如下表 1 所示。

表 1 仿真实验核心参数配置

Tab. 1 Core parameter configuration of simulation experiment

仿真实验核心参数名称	配置值
dfs. replication	3
mapred. compress. map. output	true
io. sort. factor	30
dfs. datanode. max. xcivevers	4 096
mapred. reduce. tasks	6
dfs. block. size	134 217 728
mapred. map. tasks	10
cker. tasktracker. reduce. tasks. maximum	10
mapred. child. java. opts	-Xmx512m

3.2 基于 CPU 均衡度的调控效果

评价一个调控算法的好坏,主要看算法是否符合程序负载量保持均衡的预期要求,基于此,本文采用 CPU 均衡度为指标,测试本文虚拟化桌面程序调控算法的优异性。

CPU 均衡度计算公式为

$$G_n = \frac{1}{n} \sum_{k=1}^n (\Gamma_k - \Gamma_{\text{nature}})^2 \quad (9)$$

式中, G_n 表示均衡尺度值; k 表示作业数量; Γ_k 表示在数量 k 下的属性值; Γ_{nature} 表示程序属性。

以初始程序数量为 60 个、CPU 均衡度为 5 的虚拟桌面作为调控对象,在 15、25、35 s 处随机添加 5、10、20 个程序;在 10、20、30、40 s 处随机减少 5、10、15、30 个程序。运用本文方法、基于博弈论的空间数据资源分配方法、虚拟化资源的访问权限调控方法,进行仿真对比试验,结果如图 1 所示。

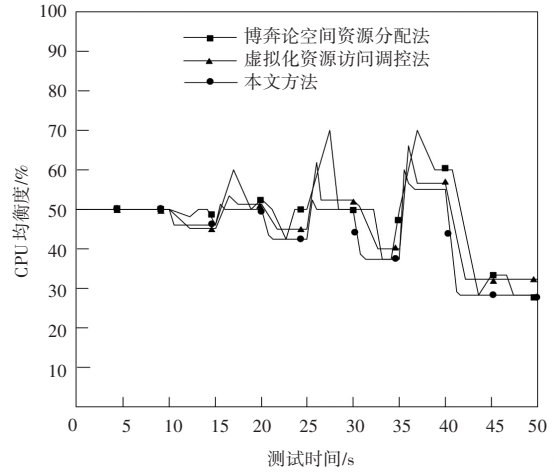


图 1 基于 CPU 均衡度的桌面程序集中调控效果

Fig. 1 Centralized regulation effect of desktop programs based on CPU balance

从图 1 中可以看出,未经过任何操控的 CPU 均衡度数值为 5,能够看出所提方法在不同的程序变动时间点上,对应的均衡度值都发生了变化,且程序删除行为和安全行为所导致的均衡度下降和上升幅度满足实际设定任务。反观其他两种方法,均衡度变化幅度偏大,这说明在执行程序删除或安装任务时,二者调控下网络存在较多冗余信息,占据一部分内部容量空间,调控性能差。

3.3 集中调控平均等待时间对比

本文将集中调控信息的逗留时间及平均排列队长作为算法性能的评价指标。其中,平均逗留时间定义为:调控作业的等待与执行时间的和与作业总数目的比值 T_{wait} ,可精准地描述算法的调控速度及效率,计算公式为

$$T_{\text{wait}} = \sum (T_{\text{queue}} + T_{\text{execution}}) / N_p \quad (10)$$

式中, T_{queue} 表示调控作业的平均等待时间; $T_{\text{execution}}$ 表示调控作业的执行时间; N_p 表示作业总数目。逗留时间 T_{wait} 越长,代表算法的效率越低、调控质量越差。

以系统 HDFS 中的桌面程序作为调控对象,3 种方法的平均等待时间对比实验结果,如下图 2 所示。

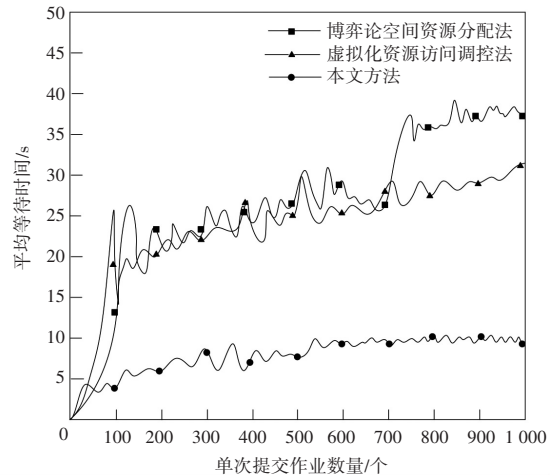


图 2 三种方法集中调控平均等待时间对比

Fig. 2 Comparison of average waiting time of three centralized regulation methods

由图2可知,本文方法的平均等待时间曲线斜度是最小的,对比方法的对应时间值及曲线增长度角都相对较大。这表明在同等条件下,本文能够以最快速度完成调控,另外两种方法等待时间过长的主要原因就是:没有优化调控系统中的反馈机制,增加调控指令发出及接受的反馈周期,随着调控作业数量的增加,逗留时间逐步叠加,降低调控算法效率。

3.4 集中调控平均排队队长对比

平均排队队长 L_{wait} 是待调控作业总长与每个作业完成调控的时间比值,用 L_{wait} 调控算法对数据的容纳性及管控性,计算公式为

$$L_{wait} = \sum L_{time} / T_{time} \quad (11)$$

式中, T_{time} 表示单位作业的完成时间(以 10 min 作为一个单位长度); L_{time} 表示调控作业总长。 L_{wait} 指标越小代表算法可消化的队长越长,效率越高、性能表现越好。

在实验环境及核心参数配置的基础上,计算 3 种方法的集中调控平均排队队长,实验结果如下图 3 所示。

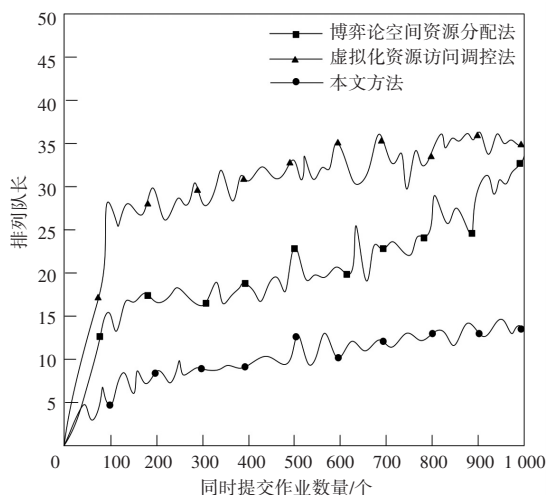


图3 三种方法集中调控平均排队队长对比

Fig. 3 Comparison of average queue length of three centralized regulation methods

从图3中可以看出,当系统中同时提交程序作业数目达到 250 时,本文方法比基于博弈论调控法、虚拟化资源访问权限调控法的队长数值,分别相差 22 和 12 个队列值。说明当同时进行同等数量的虚拟化桌面程序调控时,本文请求指令到达调控终端的时间是最短的,能提前完成调控。这是因为,本文的调控机制建立了多个调度窗口,当数据量超出负载时,就会通过服务窗口进行调控,解决信息队列较长的问题。

4 结论

本文针对虚拟化桌面程序分布混乱、调控请求及反馈机制不灵敏的问题,将异构资源程序分配法与反馈机制的动态作业调度算法相结合,提高程序调控的准确性和灵敏性。利用容器技术建立基于虚拟云桌面的安全访问序列,将所有用户的访问请求数据通过序列进行输出,有效过滤掉非法及存在风险用户的访问请求,在很大程度上提高了程序使用的安全性,同时也降低了后续程序调控的误差性。最终通过资源群组的分配,使单位程序子集中的数据实现了时间的高度统一,算法数据包容性较强,整体调控结构简单直观、易实施。仿真实验证明,本文方法具有较强的负载程序平衡能力,调控精度较强、反馈速率较快,平均等待时间及排队队长很小,可实现高效的虚拟化桌面程序集中调控,为计算机技术发展提供有效帮助。

参考文献

- [1] 翟永,刘津,刘磊,陈杰. 基于博弈论的空间数据中心私有云资源分配管理分析[J]. 计算机科学,2020,47(增刊2):373-379.
- [2] 孙维佳. 云计算环境下虚拟化技术的资源分配与负载均衡算法研究[J]. 数码设计(电子版),2024(2):234-236.
- [3] 王雷. 桌面虚拟化技术的安全性分析及对策[J]. 网络安全和信息化,2023(1):21-24.
- [4] 申帅,董晓龙,钱建生,等. 无线充电体域网基于边际效用的网络资源分配方法[J]. 计算机应用研究,2020,37(6):1844-1847.
- [5] 田甜. 桌面虚拟化提高信息化管理和运维效率的方法研究[J]. 信息与电脑,2025,37(18):92-94.
- [6] 梁子君,张卫华,杨伟,等. 基于时空资源优化的交叉口交通流动态调控方法研究[J]. 合肥工业大学学报(自然科学版),2020,43(9):1240-1249.
- [7] 农英雄,陈智斌,梁冬,等. 基于虚拟化容器和 AI 算法引擎的办公云系统研究[J]. 自动化与仪器仪表,2024(11):194-197.
- [8] 李玉荣,梁桂才. 基于 GPU 虚拟化的人工智能教学平台的实现[J]. 信息技术,2024(8):71-78.
- [9] 许琦. 基于 KVM 架构的云桌面 GPU 虚拟化技术研究[J]. 自动化与仪器仪表,2023(3):138-142.
- [10] 李鼎基,糜泽羽,吴保东,等. 基于跨虚拟机零下陷通信的加速器虚拟化框架[J]. 软件学报,2020,31(10):3019-3037.
- [11] 李院霞,梁栋,杨晓茹,等. 面向新能源接入的电网设备资源优化调度策略[J]. 自动化技术与应用,2025,44(9):49-53,78.
- [12] 赵树林,孙晓婷. 桌面虚拟化在高校多场景中的模式选择与应用研究[J]. 现代电子技术,2023,46(20):91-94.
- [13] 朱国,刘桂芹,陈果. 煤矿井下钻探装备设计虚拟化云桌面平台研究[J]. 煤矿机械,2024,45(10):196-198.
- [14] 李焕,肖宇亮,戴玉敏. 基于云计算的设计软件资源共享平台构建与实现[J]. 自动化与仪器仪表,2022(8):142-146.
- [15] 潘远,张华兵,刘梓健,等. 基于图注意力网络的算力资源调配优化研究[J]. 自动化技术与应用,2026,45(4):64-68.