

基于贪心算法的容器云资源低能耗部署方法

徐胜超, 叶朝武

(广州华商学院 数据科学学院, 广州 511300)

摘要: 将容器云平台资源整体能耗最低作为目标, 设计基于贪心算法的容器云资源低能耗部署方法; 在物理主机与虚拟机对应、虚拟机与容器对应等约束条件下, 结合静态和动态两个部分构建容器云资源能耗模型; 通过资源虚拟化与去除冗余两个步骤, 得到容器云资源的整合结果; 检测物理机负载状态, 确定虚拟机迁移源物理机和目标物理机, 利用贪心算法均衡调度容器云资源负载, 最终通过容器云资源编排重组, 实现容器云资源低能耗部署; 采用 Network Simulator 网络仿真器、Docker swarm kit 为核心, 利用 40 个物理主机、1 000 个容器、5 类粒度不同的虚拟机构建容器云仿真平台, 在贪心算法优化设计部署方法下, 容器云资源的利用率和负载均衡度得到明显提升, 能量损耗明显下降。

关键词: 云计算; 贪心算法; 容器云资源; 资源部署; 低能耗部署

Low Energy Consumption Deployment Method of Container Cloud Resources Based on a Greedy Algorithm

XU Shengchao, YE Zhaowu

(School of Date Science, Guangzhou HuaShang College, Guangzhou 511300, China)

Abstract: The lowest overall energy consumption of container cloud platform resources is taken as the goal, a low-energy deployment method of container cloud resources based on a greedy algorithm is designed. Under the constraints of the correspondences between the physical host and the virtual machine or between the virtual machine and the container, a container cloud resource energy consumption model is constructed by combining with the static and dynamic parts. Through the resource virtualization and redundancy removal, the integration results of container cloud resources are obtained. The load status of physical machines is detected, the source and target physical machines of virtual machine migration is determined, the greedy algorithm is used to balance and schedule the load of the container cloud resources, and finally the low-energy deployment of the container cloud resources is realized through the arrangement and reorganization of the container cloud resources. The network simulator and docker swarm kit are taken as the core to build the container-based cloud simulation platform of 40 physical hosts, 1 000 containers and 5 kinds of virtual machines with different granularity. Compared with the traditional deployment methods, the experiment results of the greedy optimized algorithm show the utilization and load balance of the container cloud resources are significantly improved, and the energy loss is greatly reduced.

Keywords: cloud computing; greedy algorithm; container cloud resources; resource deployment; low energy consumption deployment

0 引言

如何让云计算变得更加节能高效, 是当今政府和企业都在关注的重点问题。容器云资源的管理和部署都是以容器为基础的, 不需要涉及硬件, 因此它的重量更轻, 启动速度也更快。但是由于容器云资源较大, 数据中心能耗较高, 致使容器云资源利用率较低, 因此, 需要对容器云资源完成低能耗部署, 有效降低数据中心的能耗, 提高容器云资源利用率^[1]。

当前文献 [2] 提出基于资源预留的容器云部署方法。文献 [3] 在云平台用户个性化资源需求与虚拟机动态迁移技术有效集成的基础上, 设计了一种可根据用户实际资源

需求动态调整的节能虚拟机部署方法, 将虚拟机部署框架与基于双光标控制机制的虚拟机部署选择算法相结合, 有效地控制云数据中心的必要活动物理节点数量, 并且减少虚拟机迁移同步的频率。文献 [4] 提出了一种基于剪枝算法和深度强化学习的资源管理和任务部署方法。文献 [5] 提出面向大数据复杂应用的虚拟集群动态部署模型方法, 该方法在容器技术支持情况下建立了一个部署模型, 依据资源的动态变动, 利用该模型对虚拟群体实行调整和计算, 其中包含节点类型的计算和节点规模的调整, 并利用该模型完成对用户作业的部署, 实现资源部署的目的, 该方法建立的模型不够全面, 存在请求接收率差的问题。

收稿日期: 2023-01-16; 修回日期: 2023-02-24。

基金项目: 国家自然科学基金面上项目(61772221)。

作者简介: 徐胜超(1980-), 男, 湖北武汉人, 硕士, 讲师, 主要从事并行分布式处理软件方向的研究。

叶朝武(1999-), 男, 海南海口人, 大学本科, 主要从事信息安全技术方向的研究。

引用格式: 徐胜超, 叶朝武. 基于贪心算法的容器云资源低能耗部署方法[J]. 计算机测量与控制, 2023, 31(5): 265-270.

文献 [6] 提出基于多维资源空闲率权重的容器云多维资源利用率均衡调度方法。文献 [7] 提出基于遗传算法的容器云资源配置优化方法, 该方法将容器云平台数据中心整体能耗最低作为目标函数, 设置物理主机与虚拟机对应、虚拟机与容器对应等约束条件, 利用遗传算法通过染色体表达、初始化、交叉操作、变异操作以及设置适应度函数 5 个步骤求解目标函数, 获取最优容器云环境资源配置结果。通过对比分析认为, 上述方法虽然均完成了容器云资源部署, 但是容器云资源的利用率和负载均衡度仍然比较低。

本文将贪心算法应用到容器云资源低能耗部署方法的优化设计工作中, 在约束条件下, 构建容器云资源能耗模型, 对云资源进行虚拟化处理, 整合容器云资源后, 对物理机负载状态检测, 利用贪心算法均衡调度容器云资源负载, 然后通过容器云资源编排重组, 达到资源部署低能耗的目的。最后通过构造系统仿真平台, 验证了贪心算法优化容器云资源低能耗部署策略的正确性和优秀性能。

1 容器云及其资源组织

容器是提供程序运行的独立空间, Docker 容器平台负责容器的生命周期管理以及相关资源信息的管理。Docker 容器技术的出现, 为容器平台的诞生打下了坚实的基础^[8]。容器云平台开展资源低功耗部署时, 平台基本单位为容器, 封装软件为平台运行环境。容器云平台主要侧重于容器的编排以及资源的部署、共享, 具体的资源组织结构如图 1 所示。容器云平台包括应用、开发、资源、共享、设备、集群服务等层级。资源配置模块与项目开发模块位于集群服务层中, 本文将研究的云资源低能耗部署方法应用于该资源配置模块中, 实现云资源的低能耗部署配置。

容器运行时, 可以通过 Hypervisor 采集容器的虚拟资源运行状态参数, 但是由于采样参数过多, 会对物理机的运算带来额外能耗负担, 所以本文在建立数据中心能耗模型时, 通过灰色关联分析方法^[9], 完成容器运行状态参数的提取。设定容器虚拟能耗为 P , 容器运行参数比较序列设定 (p_1, p_2, \dots, p_n) , 通过关联系数计算方法, 完成参数比较序列的采样, 过程如下式所示:

$$\beta_i(P, a_i) = \frac{1}{n} \sum_{l=1}^n \beta(P(l), p_i(l)) \quad (1)$$

式中, 容器比较序列与 P 之间的灰色关联值标记 $\beta(P, p_i)$ 形式, 容器序列的两级最小值分别用 $P(l), p_i(l)$ 表述, 常数标记 i, n 形式。设定 $\beta_i \geq 0.6$ 为参数选取阈值, 完成容器运行状态的选取。

2 贪心算法优化容器云的低能耗部署方法

2.1 构建容器云资源能耗模型

云资源部署效果的测试指标为云资源利用率和物理机负载均衡系数, 则云资源利用率的数值结果为:

$$\eta_{used} = \frac{N_{used}}{N_{deploy}} \times 100\% \quad (2)$$

物理机负载均衡系数的数值结果为:

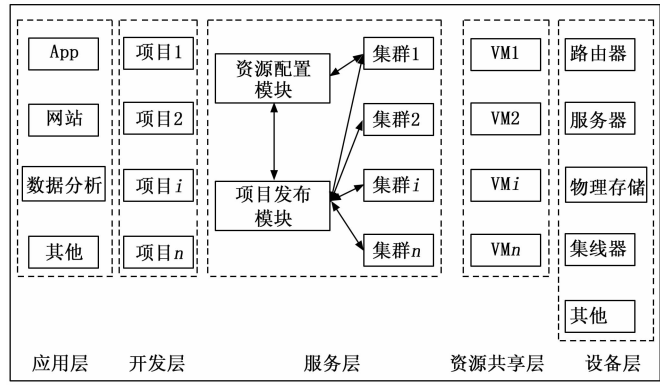


图 1 容器云平台资源组织结构图

$$\varphi = \sum_{i=1}^M \frac{G_i}{\bar{G}} \quad (3)$$

其中: N_{used} 和 N_{deploy} 分别表示被利用的容器云资源以及实验中设置的资源总量, G_i 和 \bar{G} 对应的是第 i 个物理机的实际负载以及所有物理机的平均负载。最终计算得出云资源利用率越高、物理机负载均衡系数越接近 1, 则对应方法的部署效果越好。另外能量消耗性能的测试指标为能耗量, 本文设计其能量消耗模型为:

$$P_{total} = P_{Physics} + P_{cloud} + P_{transfer} \quad (4)$$

式中, $P_{Physics}$ 、 P_{cloud} 和 $P_{transfer}$ 分别为物理机能耗、云服务器能耗和虚拟机迁移能耗, 计算得出能耗量越低, 说明对应部署方法的性能越优。值得说明的是, 在容器云资源的部署与运行过程中, 能量消耗可以分为静态能耗和动态能耗两个部分, 其中静态能耗包括物理机能耗 $P_{Physics}$ 与云服务器能耗 P_{cloud} , 而动态能耗则是容器迁移能耗 $P_{transfer}(t)$ 。

其中物理机能耗的计算公式如下:

$$P_{Physics} = P_{idle} + (P_{max} - P_{idle}) * \eta_{CPU} \quad (5)$$

式中, 参量 P_{idle} 和 P_{max} 分别为物理机在空闲和满负载状态下的能量消耗值, 该参数的具体取值由物理机的硬件条件决定, η_{CPU} 表示的是物理机的 CPU 利用率, 该值与物理机状态、虚拟机 CPU 利用率以及 CPU 资源部署量有关^[10]。云服务器能耗 P_{cloud} 计算方法为公式 (6):

$$P_{cloud} = r * C_{max} + (1 - r) * C_{min} \quad (6)$$

式中, r 表示云服务器的 CPU 利用率, C_{max} 表示云服务器忙时的最大能量消耗, C_{min} 云服务器闲置时产生能耗。将 CPU 利用率的计算结果代入到公式 (5) 中, 便可得出物理机能耗的求解结果 $P_{Physics}$ 。同理计算云服务器不同状态下的能耗总和 P_{cloud} , 即可得出服务器能耗的计算结果^[11]。

根据容器云平台中资源的实时变化, 虚拟机及其内部容器的实时位置会发生改变, 因此在能耗计算之前首先需要判定虚拟机或容器是否发生迁移。考虑虚拟机和容器在迁移过程中内存镜像传递的次数, 得出虚拟机迁移能耗, 也就是动态能耗。虚拟机迁移能耗计算方法为公式 (7):

$$P_{transfer} = k * E + (1 - k)E * u \quad (7)$$

式中, k 表示虚拟机完全闲置时产生能耗所占百分比, E 表

示虚拟机忙时的最大能量消耗, u 表示虚拟机的 CPU 利用率。

最终通过累加求和的方式, 得出得出容器云资源部署与运行过程中的总能耗值^[12]。在实际的容器云资源部署过程中, 由于静态能耗不会因为虚拟机与容器的改变而产生影响, 因此本文的算法主要保证迁移能耗最低即可。

2.2 物理机负载状态检测

Docker 或者 Cloudsim5.0 这种容器云管理平台都自带这方面的功能, 本文的贪心算法的部署过程都是以这些容器云平台为基础进行改进。物理机负载状态检测的目的是判断当前物理机是否需要将虚拟机以及容器迁移到其他物理机中, 同时也需要判断该物理机是否能够支持外部容器资源进入^[13]。将物理机的负载状态分为绝对过载 Overload、绝对低载 UnderLoad 和正常 Normal 三种状态。

在实际的检测过程中, 利用物理机的后台运行程序, 可以直接得出当前负载值, 而下一时刻负载预测值可以根据物理机历史负载数据变化规律以及容器云资源迁移任务得出^[14]。经过判断, 若当前物理机处于正常运行状态, 则可以将其作为容器云资源调度的目标主机, 但该设备的优先级较低, 且无须从中调取容器云资源^[15]。若物理机处于绝对低负载, 则在容器云资源部署过程中, 可以直接将未分配的资源调度到该物理机中, 并赋予该设备较高的选择优先级。如果当前物理机处于绝对过载状态, 则该设备无法作为资源迁移的目标主机, 且必要时需要从中调取部分资源分配给其他物理机。

2.3 容器的迁移

由于云计算环境中的任务呈现多样性, 导致任务在处理时不仅有实时的在线处理任务, 还有若干异步任务。所以云数据中心运行时, 容器、虚拟机^[16-17]以及物理机都会随着任务的进行增加负载情况。其中, 由于物理机任务完成时间较早, 所以当物理机完成任务, 而容器未完成任务时, 会出现短暂的资源浪费。

在容器云环境中, 虚拟机迁移是保障物理服务器负载均衡的关键。迁移过程中, 通常分为触发迁移时间、迁移目标服务器以及迁移位置 3 个阶段。迁移触发时间主要依据数据中心的资源部署策略来实现。迁移过程如下:

- 1) 统计物理主机资源利用率, 通过统计将物理主机中过载的利用率信息保存在相关列表中。
- 2) 基于列表信息, 对其展开降序排列, 选取对应的容器迁移目标。
- 3) 依据排列的顺序从列表中选择对应物理主机, 并将其作为迁移过程中的源物理主机, 依据顺序完成迁移后, 保证源物理主机利用率不超载, 若超载则迁移不成功, 需要依据上述流程继续寻找迁移目标。
- 4) 统计物理主机中资源利用率潜在信息, 并将其存储在相应列表中。
- 5) 选取一台不在列表当中的物理主机作为容器迁移目标, 将部署于源物理主机上的容器全部迁移至目标物理主

机中, 完成容器的迁移。

依据上述容器云虚拟资源的迁移流程可知, 容器在云资源部署以及迁移过程中, 容器云都需要面对物理主机的选取问题, 物理主机是整个数据中心的最大能源消耗部件, 所以, 找出物理主机与容器、虚拟机之间的能耗计算关系, 是完成容器云资源低能耗部署的关键。

2.4 利用贪心算法均衡调度容器云资源负载

贪心算法是一种简单、快速的设计某些问题求最优解的技术, 它的特点是不从全局的最佳状态出发, 而仅以局部最优的形式来解决问题, 从而节约了在寻求最佳方案的过程中所耗费的时间。在作出贪心的抉择时, 采用反复的方法, 每次作出一个抉择, 问题的大小就会缩小, 从而使每个步骤都能获得最优的结果。

在容器云中, 在负载状态检测完毕后, 要将部分过载主机中的虚拟机转移出来。它不仅可以保证容器云平台的高性能, 还可以降低系统的能量消耗^[18]。对所有的主机都采用过载检测法, 以判断主机是否超载, 如果超载, 则采用虚拟机选择策略, 选择虚拟机进行迁移, 直至主机不过载为止。容器云资源的负载均衡调度原理如图 2 所示。

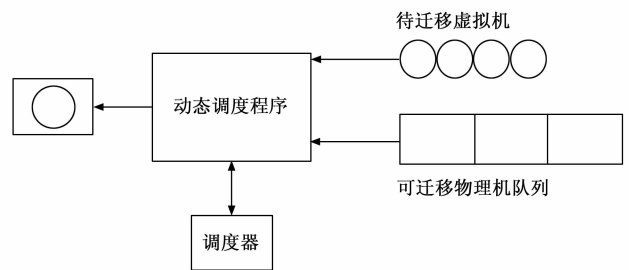


图 2 容器云资源负载均衡调度原理

根据构建容器云资源能耗模型可以看出, 容器云资源部署与运行的能耗与迁移时间以及 CPU 利用率有关, 因此利用贪心算法选择需要部署的容器云资源, 并按照图 3 表示流程, 实现容器云资源的迁移与调度。

按照上述流程分别选择被迁移的虚拟机和目的迁移节点, 在过载物理机环境中选择 CPU 资源的利用率最高的虚拟机作为待迁移虚拟机, 选择距离该物理机最近的处于绝对低载状态的物理机作为目的迁移节点。将选择的虚拟机保存到列表 OverLoadList, 并根据 CPU 利用率利用贪心算法进行降序排列, 按照顺序将被迁移物理机中的虚拟机迁移到目标物理机中, 并保证目标物理机不过载, 直到迁移源物理机不再过载^[19-21]。

若目标物理机容纳所有的待迁移虚拟机, 则需要选择其他绝对低载物理机作为新的目标物理机, 按照上述方式进行目标迁移, 直到 OverLoadList 列表中的过载虚拟机被处理完成。容器云资源迁移调度完成后, 更新其位置信息, 并在部署容器中运行相应的云任务^[22]。

根据云计算平台的初始化部署策略, 可以获得容器云资源的初始化部署方案, 综合考虑用户的需要和资源利用

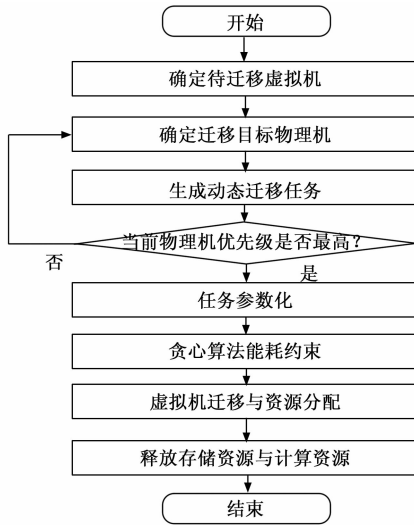


图 3 利用贪心算法的容器云资源调度流程图

的动态, 采用能耗均衡加权部署方法实现资源初始化部署^[23]。

3 仿真实验与性能测试

以测试基于贪心算法的容器云资源低能耗部署方法的应用性能为目的, 分别从容器云资源的部署效果和能量消耗两个方面进行性能测试, 并通过与传统云资源部署方法的对比, 体现出贪心算法在云资源部署工作中的应用优势。

3.1 部署容器云平台实验环境

本文在 NS2.0 (Network Simulator 网络仿真器) 仿真软件、Windows XP 操作系统、Intel (R) Celeron (R) 2.6 GHz 处理器下进行实验测试。以 Docker swarm kit 为核心, 建立了一个容器云平台。在此基础上, 采用服务器组成了节点集群。各服务器的可用度指数是相同的, 都是 0.95, 所以容器集群中的每一个节点都是相同的。这表示对于相同的服务, 在容器集群中的任何一个节点上, 对于服务的总体可用性都有相同的影响。平台环境详情如表 1 所示。

表 1 容器云平台实验环境

项目名称	说明
容器配置	CPU 内核 8 个, 存储空间 32 GB
服务器可用度指数	0.95
容器数量	1 000 个
服务器数量	40 台

为降低测试时延及网络稳定性, 采用 Docker 镜像的本地二进制档案安装方法, 以设定相关的环境^[24]。所有的服务要求都被打包成一个云任务, 一个虚拟机可以在同一时间执行多个云任务。根据需要的物理资源, 这些初始化的虚拟机依次分配到已打开的主机部件^[25]。

3.2 准备容器云资源处理样本

实验采用的样本数据为 AuverGrid 资源分配数据集, 该数据集记录了一个容器集群 15 天的运营状况, 包括 364

547 项服务, 每隔 2 分钟对容器资源使用数据进行记录与采集。实验设置容器云平台中虚拟机的数量为 1 000 个, 不能虚拟机的类型与粒度的基本运行参数如表 2 所示, 这样可以覆盖大部分任务的情况, 不失一般性。

表 2 容器云平台中虚拟机类型参数设置表

虚拟机类型编号	内存需求大小/MB	处理器能力需求/MHz
F01	600	550
F02	800	1 210
F03	1 500	1 450
F04	2 200	1 680
F05	2 800	2 200

将上述虚拟机资源存储到容器云平台中。在不考虑物理机负载均衡的前提下, 将准备的容器虚拟机资源随机分配到各个物理机中, 记录各个物理机的资源量, 以此作为容器云资源的初始部署数据。

3.3 设置性能测试指标与实验对比项

根据实验目的, 从云资源部署效果和能量损耗两个方面设置性能量化测试指标。为了体现出基于贪心算法的容器云资源低能耗部署方法的性能优势, 实验中采用了文献 [1] 方法、文献 [2] 方法作为实验的对比方法, 在性能测试过程中, 3 个部署方法处理的云资源以及部署任务数量均相同。

3.4 实验过程与结果分析

3.4.1 容器云资源部署结果

图 4 (a) 和图 4 (b) 表示的是文献 [1] 方法和基于贪心算法的容器云资源低能耗部署方法的容器云资源部署结果。

根据图 4 (a) 可知, 文献 [1] 方法在物理机 1 和物理机 2 下的容器云资源低能耗部署后的云资源量相差较大。根据图 4 (b) 可知, 无论是物理机 1 还是物理机 2, 基于贪心算法的容器云资源低能耗部署方法进行的容器云资源低能耗部署后的云资源量均一致。

3.4.2 利用率和负载均衡度测试结果

通过理论分析部分的公式 (2) 我们可以得出文献 [2] 方法和基于贪心算法的容器云资源低能耗部署方法的资源利用率的部署结果。统计相关数据得出部署效果的测试对比结果, 如图 5 所示。

通过理论分析部分的公式 (2) 也可以得到负载均衡度的实验结果。从图 6 中可以直观的看出基于贪心算法的容器云资源低能耗部署方法的负载均衡度始终高于文献 [2] 方法的部署资源, 且贪心算法的负载均衡度更接近 1, 由此证明贪心算法的资源部署方法的部署效果更佳。是因为本文采用负载均衡策略, 在两个结点间找到一对满足交换条件的容器, 从而减少结点的碎片率, 并实现负载均衡。

3.4.3 容器云资源能耗测试结果

为了进一步验证贪心算法的有效性, 在一天 24 小时内文献 [1] 方法、文献 [2] 方法和本文方法的容器云资源

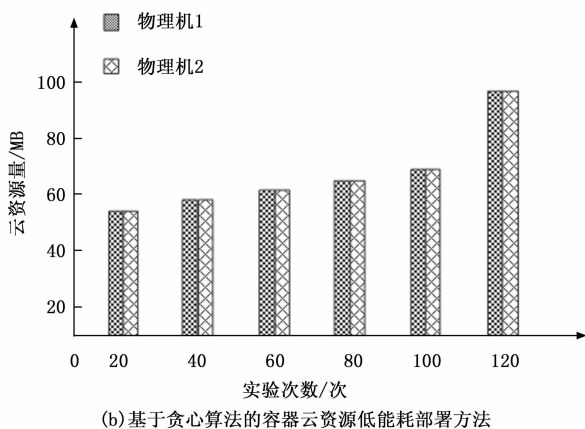
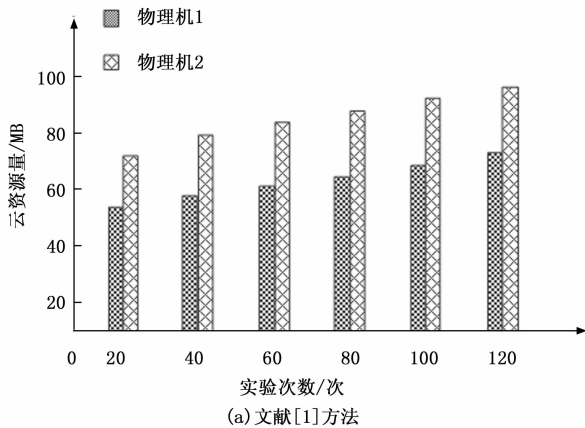


图4 容器云资源部署结果

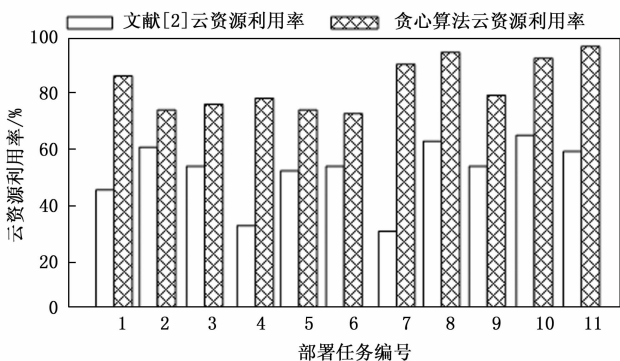


图5 两种方法的容器云资源利用率对比结果

能量损耗的测试数据如表3所示。

表3 文献[1-2]方法和贪心算法能量损耗测试数据表

次数	文献[1]方法	文献[2]方法	基于贪心算法
	P_{total}/kJ	P_{total}/kJ	P_{total}/kJ
1	37 585	43 457	24 961
2	38 064	37 451	23 196
3	41 451	34 638	25 873
4	35 244	28 804	23 014
5	39 535	29 652	20 280
6	38 453	36 570	21 041
7	41 874	31 058	20 221
8	36 455	30 452	21 831

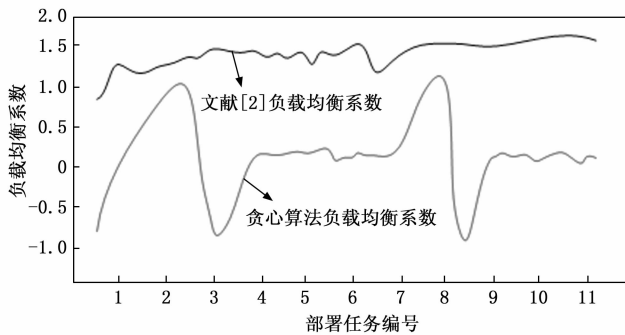


图6 两种方法的容器云资源负载均衡对比结果

由于3种资源部署方法使用的物理机与云平台型号与结构均相同,且物理机能耗 $P_{Physical}$ 和云服务器能耗 P_{cloud} 属于静态能耗,因此数据相同且不会发生变化。我们这里主要计算的是动态能耗 $P_{transfer}$, 根据公式(7)即可得出文献[1]方法、文献[2]方法和基于贪心算法的容器云资源低能耗部署方法的总能耗值 P_{total} 。

从表3可以看出,统计了8次实验结果,贪心算法法的总能耗值 P_{total} 最高值为 25 873 kJ, 最低值只有 20 221 kJ, 比文献[1]和文献[2]的方法节省一半以上的能耗,本文方法的容器云资源部署能耗更低。

4 结束语

针对容器云资源存在部署能耗高和云资源利用率低、负载不均衡的问题,提出基于贪心算法的容器云资源低能耗部署方法。综合考虑了物理机的利用率和单位时间的消耗,在保证物理机的使用效率的基础上,选取低功耗的物理机作为虚拟机、容器的目标迁移物理机,从而有效地减少了以容器为基础的数据中心的总能量消耗,对于云平台的管理运行以及云资源的有效利用均具有较好的参考价值。但是本文在研究的过程中,没有考虑容器云资源低能耗部署时间,使云资源低能耗部署效率降低,为此,在接下来的研究中,需要对算法进行改进,提升容器云资源低能耗部署效率。

参考文献:

- [1] 李启锐, 彭志平, 崔得龙, 等. 容器云环境虚拟资源配置策略的优化 [J]. 计算机应用, 2019, 39 (3): 784-789.
- [2] 尹飞, 龙玲莉, 孔峥, 等. 面向动态负载的集群容器部署方法 [J]. 计算机应用, 2021, 41 (6): 1581-1588.
- [3] LIU S, LI C, LIU Z, et al. Virtual Machine Dynamic Deployment Scheme Based on Double-Cursor Mechanism [J]. IEEE Access, 2020, (8): 214481-214493.
- [4] DONG Y, XU G, ZHANG M, et al. A High-Efficient Joint 'Cloud-Edge' Aware Strategy for Task Deployment and Load Balancing [J]. IEEE Access, 2021, (9): 12791-12802.
- [5] 王瑾, 曹云鹏, 王海峰. 面向大数据复杂应用的虚拟集群动态部署模型 [J]. 计算机应用研究, 2020, 37 (6): 1760-1764.

[6] 龚 坤, 武永卫, 陈 康. 容器云多维资源利用率均衡调度研究 [J]. 计算机应用研究, 2020, 37 (4): 1102-1106.

[7] 徐胜超, 熊茂华. 基于遗传算法的容器云资源配置优化 [J]. 计算机与现代化, 2022 (1): 108-112.

[8] POTDAR A M, NARAYAN D G, KENGOND S, et al. Performance Evaluation of Docker Container and Virtual Machine [J]. Procedia Computer Science, 2020, 171: 1419-1428.

[9] ZHANG F, ZHANG J, NI H, et al. Optimization of Al-Si10MgMn Alloy Heat Treatment Process Based on Orthogonal Test and Grey Relational Analysis [J]. Crystals, 2021, 11 (4): 385.

[10] J CHEN, WANG Y, LIU T. A proactive resource allocation method based on adaptive prediction of resource requests in cloud computing [J]. EURASIP Journal on Wireless Communications and Networking, 2021 (1): 1-20.

[11] IQBAL A, THAM M L, CHANG Y C. Double Deep Q-Network-Based Energy-Efficient Resource Allocation in Cloud Radio Access Network [J]. IEEE Access, 2021 (9): 20440-20449.

[12] HE W, GUO S, GUO S, et al. Joint DNN Partition Deployment and Resource Allocation for Delay-Sensitive Deep Learning Inference in IoT [J]. IEEE Internet of Things Journal, 2020, 7 (10): 9241-9254.

[13] NGUYEN D T, PHAM C, NGUYEN K K, et al. Placement and Chaining for Run-Time IoT Service Deployment in Edge-Cloud [J]. IEEE Transactions on Network and Service Management, 2020, 17 (1): 459-472.

[14] CAO B, ZHANG J, LIU X, et al. Edge-Cloud Resource Scheduling in Space-Air-Ground Integrated Networks for Internet of Vehicles [J]. IEEE Internet of Things Journal, 2021 (99): 1.

[15] GUO W, TIAN W, YE Y, et al. Cloud Resource Scheduling With Deep Reinforcement Learning and Imitation Learning [J]. IEEE Internet of Things Journal, 2022, 9 (8): 5765-5772.

[16] LIANG H, DU Y, GAO E, et al. Cost-driven scheduling of service processes in hybrid cloud with VM deployment and interval-based charging [J]. Future generation computer systems, 2020, 107 (6): 351-367.

[17] KRISHNA G V, BHARATI K F. Improvising Dynamic Cloud Resource Allocation to Optimise QoS and Cost Effectiveness [J]. International Journal of Engineering and Advanced Technology, 2021, 10 (3): 206-209.

[18] ZHENG J, ZHANG Z, MA Q, et al. Multi-Resource VNF Deployment in a Heterogeneous Cloud [J]. IEEE Transactions on Computers, 2022, 71 (1): 81-91.

[19] TAN D D, LONG D N, DUONG T Q, et al. Joint Optimisation of Real-time Deployment and Resource Allocation for UAV-Aided Disaster Emergency Communications [J]. IEEE Journal on Selected Areas in Communications, 2021, 39 (11): 3411-3424.

[20] XIANG Z, DENG S, TAHERI J, et al. Dynamical Service Deployment and Replacement in Resource-Constrained Edges [J]. Mobile Networks and Applications, 2020, 25 (3): 674-689.

[21] ELHALAWANY B M, HASHAD O, WU K, et al. Uplink Resource Allocation for Multi-Cluster Internet-of-Things Deployment Underlying Cellular Networks [J]. Mobile Networks and Applications, 2020, 25 (3): 300-313.

[22] LIU Y, SHANG X, YANG Y. Joint SFC Deployment and Resource Management in Heterogeneous Edge for Latency Minimization [J]. IEEE Transactions on Parallel and Distributed Systems, 2021, 32, (8): 2131-2143.

[23] MIN-ALLAH N, QURESHI M B, JAN F, et al. Deployment of real-time systems in the cloud environment [J]. The Journal of Supercomputing, 2020, 77 (4): 2069-2090.

[24] HALIMA R B, KALLEL S, NACER M A, et al. Optimal business process deployment cost in cloud resources [J]. The Journal of Supercomputing, 2020, 77 (1): 1579-1611.

[25] MAO S, HE S, WU J. Joint UAV Position Optimization and Resource Scheduling in Space-Air-Ground Integrated Networks With Mixed Cloud-Edge Computing [J]. IEEE Systems Journal, 2021, 15 (3): 3992-4002.

[11] 拓明秀, 张贵仓, 汪 凯. 结合加权的三次 Bézier 曲线的新扩展 [J]. 计算机工程与设计, 2020, 41 (3): 756-762.

[12] 朱振华, 汪 宁, 李 超. 基于 Raphael 态势标绘的研究与实现 [J]. 测绘与空间地理信息, 2019, 42 (6): 4.

[13] 赵文宁. 一种基于 OpenGL 的态势符号标绘系统设计与实现 [D]. 北京: 中国科学院大学 (中国科学院工程管理与信息技术学院), 2017.

[14] 骆小龙, 陈 玲. 红绿色盲色弱人群的地图色彩印象研究 [J]. 测绘地理信息, 2020, 45 (4): 116-120.

[15] 惠学武, 孟祥宇. 融合包围盒智能算法的虚拟场景碰撞检测研究 [J]. 计算机仿真, 2021, 38 (7): 209-213.

[16] 王小丽. 三维图像绘制过程优化处理方法研究与仿真 [J]. 计算机仿真, 2017, 34 (4): 327-330.

[17] 杨 光, 黄 婧, 徐 伟. 一种面向三维数字地球的线符号绘制方法 [J]. 信息技术与信息化, 2020 (10): 153-157.

[18] 王 旺. 基于地图的数据可视化系统的研究与实现 [D]. 北京: 北京邮电大学, 2018.

[19] 王培元, 关 欣. 混合增强视觉认知架构及其关键技术进展 [J]. 中国图象图形学报, 2021, 26 (11): 2619-2629.

[20] 韩 晓. 面向筹划决策的动态知识图谱系统设计与实现 [D]. 大连: 大连理工大学, 2022.

[21] 王 旺. 基于地图的数据可视化系统的研究与实现 [D]. 北京: 北京邮电大学, 2018.

[22] 靖 旭, 宋留勇, 王潇雨. 网络空间态势的多尺度表达研究 [J]. 中国电子科学研究院学报, 2019, 14 (11): 1202-1206.