

贪心算法优化云数据中心的虚拟机分配策略^①



徐胜超

(广州华商学院 数据科学学院, 广州 511300)
通讯作者: 徐胜超, E-mail: isdooropen@126.com

摘要: 如何将云客户端的大量虚拟机均匀的分配到云数据中心的物理主机上执行是一个关键问题. 提出了贪心算法优化云数据中心的虚拟机分配策略, 首先设计一个用于企业的云数据中心的工作场景, 该场景包括三层云计算系统结构, 包括用户层、云服务提供者层和云数据中心集合层. 用户层用来生成虚拟机的请求集, 云服务提供者层通过经典的装箱问题算法完成用户层的大量的虚拟机请求集到底层的云数据中心的分配. 然后建立虚拟机分配过程中各种约束因素的数学模型; 最后利用贪心算法优化云数据中心之间的虚拟机分配. 利用某个企业的大数据中心作为云端测试环境, 测试结果表明, 经典的最好适应算法 Best-Fit-Algorithm 具有比较的虚拟机分配效果, 云平台的能量消耗比较少, 该实验结果对于其他企业构造云数据中心有比较好参考价值.

关键词: 低能量消耗; 装箱问题; 虚拟机放置; 云数据中心; 贪心算法

引用格式: 徐胜超. 贪心算法优化云数据中心的虚拟机分配策略. 计算机系统应用, 2021, 30(3):134-141. <http://www.c-s-a.org.cn/1003-3254/7814.html>

Greedy Algorithms Optimized Virtual Machine Allocation for Cloud Data Centers

XU Sheng-Chao

(School of Data Science, Guangzhou Huashang College, Guangzhou 511300, China)

Abstract: It is a critical problem to uniformly allocate a large number of virtual machines at the cloud clients to the physical hosts at the cloud data centers. To this end, a greedy algorithm optimized virtual machine allocation approach for cloud data centers is proposed in this paper. First, a working scenario is designed for the enterprise-oriented cloud data centers, including three layers, a user layer, the layer of cloud service provider, and the star layer of cloud data centers. Specifically, the user layer is used to generate the request sets of the virtual machines, and the layer of cloud service provider allocates a large number of request sets of the virtual machines at the user layer to the bottom cloud data center through the classical bin packing algorithm. Then, the mathematical models considering different constraints are established during the allocation of the virtual machines. Finally, the virtual machine allocation among the cloud data centers is optimized using the greedy algorithm. In addition, the big data center of an enterprise is taken as the cloud testing environment, and the test results show that the classical Best-Fit-Algorithm (BFA) performs well in virtual machine allocation and consumes little energy of cloud platforms, providing a reference for the construction of cloud data centers in other enterprises.

Key words: low energy consumption; bin packing problem; virtual machine placement; cloud data centers; greedy algorithm

① 基金项目: 广州华商学院校内导师制科研项目 (2020HSDS04)

Foundation item: Technology & Research Project for the Supervisor of Guangzhou Huashang College (2020HSDS04)

收稿时间: 2020-07-16; 修改时间: 2020-08-13; 采用时间: 2020-08-17; csa 在线出版时间: 2021-03-03

近年来云数据中心的构造与使用成了政府和各大IT企业越来越重视的问题,为了倡导绿色云计算,云数据中心的构造目标是低能量消耗、高服务质量(Quality of Service, QoS)、节省物理空间和高可靠性等^[1-3]。一个数据中心(Data Center, DC)通常配置有大量的紧密堆积在一起物理节点(Physical Machines, PMs),目的是提高建筑物空间的利用率;虚拟化技术是云数据中心中最重要技术,虚拟化允许用户对云资源的访问是透明与简单方便的,它通过虚拟机(Virtual Machines, VM)的形式将应用程序封装在虚拟机之中,通过虚拟机分配策略将其分配到具体的数据中心的物理节点之中执行。所以虚拟机分配(Virtual Machine Distribution, VMD)等策略很大程度上影响了云平台的整体性能,目前大量的文献把性能指标关注在SLA违规比率、服务质量QoS、系统能量消耗、虚拟机迁移次数等方面^[4]。

虚拟机分配的理想目标是尽量利用个数比较少的云数据中心,同时提高每个数据中心的云资源利用效率,在此基础上提高云数据中心的灵活性和可用性,降低硬件成本和操作成本(能量消耗,物理空间占用等)^[5]。

本文设计了一种用于企业的云数据中心的工作场景,它可以接受大量客户端的虚拟机的请求,并将这些虚拟机分配到云数据中心的具体物理节点执行。参考了经典的装箱问题(classical in packing problem)模型,包括经典的最好适应算法Best-Fit-Algorithm(BFA)、首次适应算法First-Fit-Algorithm(FFA)、下次适应算法Next-Fit-Algorithm(NFA)、最坏适应算法Worst-Fit-Algorithm(WFA)和随机算法Random Allocation-Algorithm(RAA)等来将大量的虚拟机放置到各个数据中心的物理节点完成云计算。建立了虚拟机放置过程中各种约束因素的数学模型,利用贪心算法(greedy algorithm)优化云数据中心之间的虚拟机放置策略。测试结果表明,经过优化的最好适应算法BFA具有比较好的性能,这对于其他企业建立与构造云数据中心有比较好参考价值。

1 相关工作

目前学术界针对云端的虚拟机分配与迁移策略,进行了大量的研究。主要分为两类,第一类是针对从虚拟机到单数据中心中的物理节点的分配与迁移。第二类是针对从虚拟机到多个云数据中心的分配与调度。

前一类关注的虚拟机在物理节点之间的选择、分配和迁移;后者关注的是虚拟机在多个云数据中心的分配与放置。

针对第一类问题,例如文献[6]提出并评价了云端多虚拟机迁移的负载与性能变化;文献[7]提出了一种利用动态放置应用程序的方法EnaCloud,实现云平台的能效管理。文献[8]同样从CPU维度对虚拟机的动态配置问题进行建模,并利用改进的蚁群算法进行求解。文献[9]提出了PS-ABC算法,该方法能够在长期服务项目的局部时间段内实现较好节能效果,但是在服务项目全局的能耗优化问题上,效果并不理想。文献[10]在此基础上提出了PS-ES启发式算法,不仅实现了当前场景的能源优化,而且也有效降低了长期服务项目总体的能源消耗,但其考虑的维度较单一。文献[11]在Beloglazov研究的基础上提出(Service Level Agreement, SLA)违规算法,引入最小能源最大利用率策略,进一步优化虚拟机配置方法。

针对第二类研究,主要涉及到经典装箱问题,此类问题是一个基于多约束的整数规划问题,同时也是一个NP-hard问题。即将云数据中心抽象为箱子,箱子的容量是云数据中心资源的大小,包括物理服务器个数、CPU主频高低、CPU内核个数,内存大小、空余磁盘空间大小和应用程序类型等;虚拟机抽象为装入的物品。例如文献[12]利用装箱问题提出了启发式算法,但是它们考虑的维度比较单一。文献[13]提出了向量装箱和多维装箱问题,文献[14]采用了首次适应减少算法(First-Fit Decreasing, FFD)来提高装箱性能与效果。文献[15]采用遗传算法来优化资源的消耗,能量的消耗和虚拟机的放置。文献[16]将多虚拟机放置设计为一个多目标约束满意问题,目标是为了最小化物理节点个数和虚拟机迁移次数。文献[17]利用了约束编程模式,采用一个灵活和能量相关的框架来分配虚拟机到各个云数据中心。

在贪心算法方面文献[18,19]提出了云计算中的基于贪心算法的任务调度及改进,把改进任务竞争时间和改进任务执行代价作为主要因素。实验结果表明贪心算法在串行调度算法的基础上可以改进任务调度的性能,但是它们并没有将贪心算法使用到物理资源使用和低能量消耗相关的算法之中,也没有用来优化云端的虚拟机到数据中心的分配。

最近这三年又有文献提出采用软件的方法来提高

虚拟机分配和虚拟机放置策略的性能, 还有文献提出需要在虚拟机迁移过程中考虑网络攻击危险, 保证云数据中心的可靠性. 例如文献 [20] 把虚拟机迁移过程划分为物理主机状态检测、虚拟机映射、虚拟机选择、虚拟机放置 4 个复杂的步骤, 虚拟机映射主要通过任务粒度、软件代价等来进行调整; 虚拟机放置属于一类经典装箱问题, 即把大量的虚拟机 VM 放置到大量的物理节点之中, 它提出要考虑虚拟机映射和虚拟机放置之间的相互联系的算法来改善性能. 文献 [21] 提出了云数据中心基于安全检测的虚拟机迁移策略. 利用隔室技术及 SIR (Susceptible, Infected, Recovered) 模型在虚拟机迁移过程将有安全威胁的虚拟机隔离出

来, 保证云数据中心的能量消耗与安全威胁的平衡.

本文提出的虚拟机分配策略属于第二类方法, 主要针对从虚拟机到云数据中心的分配, 考虑的问题装箱问题的维度不仅是硬件条件, 还有软件条件, 采用贪心算法进行优化, 即在虚拟机分配的时候只要局部的各个维度的资源最高即可得到资源, 完成分配.

2 贪心算法的虚拟机分配策略描述

2.1 虚拟机分配的层次结构

本文提出的云数据中心的虚拟机分配的工作场景如图 1, 包括 3 层的逻辑结构: 用户层、云服务提供者层、云数据中心集层.

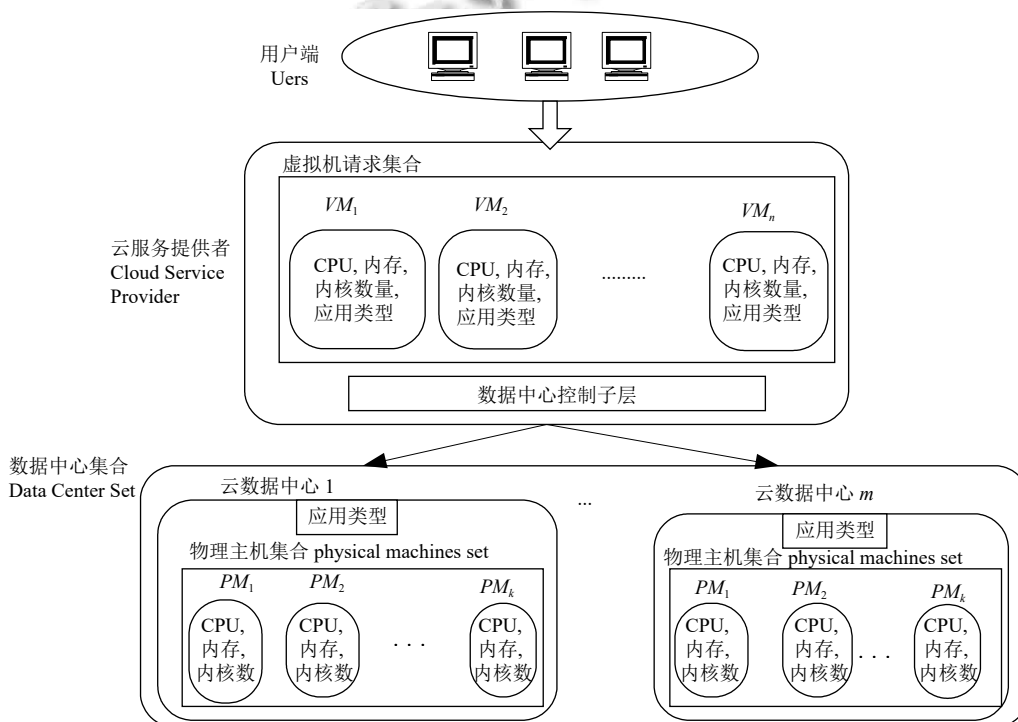


图 1 贪心算法的虚拟机分配工作场景

用户层 Users 处于顶层. 所有的应用程序的请求在这一层产生, 并形成虚拟机请求集 (VM Request Set, VMRS) 它们将被发送到下一层: 云服务提供者层 (Cloud Service Provider Layer, CSPL).

中间层 CSPL 包括两个子层, 上面的子层全部是来自用户层的虚拟机请求集. 虚拟机请求由 CPU 使用率、内存大小、内核数目、应用程序类型 (基于 CPU 的请求或基于 I/O 的请求) 组成. 下一子层是 (Data

Center Control Layer, DCCL) 数据中心控制层, 它拥有虚拟机请求和各个数据中心的信息, 虚拟机分配过程中要考虑各种约束条件, 包括资源是否满足约束条件、用户请求约束条件、用户端和 CSP 之间的 SLA 约束条件等.

所有的虚拟机请求都被放置到云数据中心被执行, 所以最下一层是 (Data Center Set Layer, DCSL) 云数据中心集层. 数据中心集层的功能是完成被分配到每个

具体的数据中心的虚拟机到其下物理节点的分配。

本文是假设每个云数据中心的节点是同构条件下的物理节点,同构条件下的计算机在统计其计算能力和存储能力的时候可以按照线性关系累加.同时该工作场景下的所有云数据中心在同一个地理位置,这样各个物理主机的网络带宽基本类似.

2.2 虚拟机分配的多维约束描述

我们把各个云数据中心的最大能量消耗和最小能量消耗分别定义为 $energymax$ 和 $energyidle$. 被分配到各个云数据中心的虚拟机也定义了一个与其相关的代价变量 $dcprice$, $dcprice$ 的数值相当于是一种资源需求,就是混合多个维度的资源需求. 虚拟机分配主要用来处理 n 个虚拟机的请求,每个虚拟机 VM_i 由 CPU 利用率,内存大小, CPU 内核数量和应用程序操作类型 (CPU based 或者 I/O based) 组成. 所有的这些虚拟机的请求组成了虚拟机请求集合: $\{VM_1, VM_2, \dots, VM_n\}$. 每个数据中心 (DC) 由于 k 个同构的物理节点组成,每个物理节点都有自己的 CPU, 内存大小和内核个数; 所有每个数据中心的所有物理节点组成了物理主机集合: $\{PM_1, PM_2, \dots, PM_k\}$. 云数据中心组成了数据中心的集合: $\{DC_1, DC_2, \dots, DC_m\}$. 这里每个 DC_j 都具有一个应用程序类型 (CPU 或者 I/O), 与具体的虚拟机请求对应, 这样方便 VM_i 可以被分配到具体的 DC_j .

DC 集合中的云数据中心必须记录一个具体的价值参数 $dcprice$, 该 $dcprice$ 为放置到该 DC 中的虚拟机的多维的资源请求数值, 在云计算的服务等级协议 SLA 协议中, 在用户 User 和云服务提供者 CSP 之间, 该价值 $dcprice$ 为固定不变的. 利用一个向量来表示 DC 集合中的所有数据中心中的价值: $dcpricevector = \{dcprice_1, dcprice_2, \dots, dcprice_m\}$. DC 集合中的每个云数据中心必须记录 $energymax$ 值和 $energyidle$ 值. 这两个值用来计算每个数据中心中的虚拟机放置后的能量消耗. 可以把能量向量定义如下:

$$\begin{cases} energymaxvector = \{dcemax_1, dcemax_2, \dots, dcemax_m\} \\ energyidlevector = \{dceidle_1, dceidle_2, \dots, dceidle_m\} \end{cases}$$

根据前面的思路, 认为把虚拟机请求集放置到具体的云数据中心为装箱问题, 我们需要考虑的问题的约束条件和维度具体包括:

1) 应用程序请求调度约束条件: 这个约束将考虑虚拟机请求的所有维度参数, 包括虚拟机请求的 CPU

使用率、内存大小、内核数目、请求类型 (基于 CPU 的请求或基于 I/O 的请求).

2) 能力约束条件 (capacity). 这个约束主要是判断云平台的整体的资源使用状态, 虚拟机请求集的整体资源需求之和必须小于或者等于云数据中心的整体资源, 当然这个资源数据必须考虑多个维度之总和.

3) 放置约束条件. 这里约束了如果整体资源已经足够, 可以只从云数据中心集合中选择一个云数据中心, 它们要求尽量少的使用云数据中心.

2.3 贪心算法优化的虚拟机分配方法

根据前面提到的, 在虚拟机分配的时候假设每个 DC 中有 k 个物理主机 PM , 设计 $CPM_r(i)$ 定义为 PM_i 中的第 r 维度的资源提供能力 (capacity), 那么一个数据中心中的所有物理主机在第 r 维度的资源提供能力可以根据式 (1) 进行计算.

$$CDC_r(i) = \sum_{j=1}^k CPM_r(i) \quad (1)$$

这里 PM_i 的维度表示的是本文前面提到的 CPU 速度、内存大小和 CPU 内核数目. 这样 m 个同构条件下的云数据中心组成的云平台在第 r 维度的总体资源提供能力可以定义为式 (2):

$$CDCSET_r = \sum_{j=1}^m CDC_r(j) \quad (2)$$

类似于式 (2), 我们定义 n 个虚拟机请求组成的虚拟机请求集的整体资源需求能力 (capacity) 为式 (3):

$$CVMRSET_r = \sum_{j=1}^n CVM_r(j) \quad (3)$$

这里 $CVM_r(j)$ 表示了第 j 个虚拟机申请在第 r 维度的资源需求. 我们把 DC_i 在第 r 维度的利用效率可以定义为放置到第 i 个数据中心的整体虚拟机请求资源和 DC_i 在第 r 维度的总体资源提供能力的比例, 见式 (4).

$$UDC_r(j) = \frac{\sum_{i=1}^n CVM_r(j) * V_{ij}}{CDC_r(j)}; \forall j \in \{1, 2, \dots, m\} \quad (4)$$

如果 VM_i 被分配到了 DC_j 中, 则这里有 $V_{ij} = 1$. 前面我们提到, 云数据中心与 3 个价值向量相关, 分别是 $dcpricevector$, $energymaxvector$ 和 $energyidlevector$. 随着 DC_s 中的 DC 个数从 1 到 m 的变化, $dcpricevector$ 会持续的增加, 这 3 个向量表达如下:

$$\begin{cases} dcpricevector = \{dcprice_1, dcprice_2, \dots, dcprice_m\} \\ energymaxvector = \{dcemax_1, dcemax_2, \dots, dcemax_m\} \\ energyidlevector = \{dceidle_1, dceidle_2, \dots, dceidle_m\} \end{cases}$$

如果把能量最大向量和能量最小向量考虑进去,那么云数据中心 DC_j 的整体能量消耗可以通过公式(5)来计算。

$$energyDC(j) = ((dcemax_j - dceidle_j) \times \max\{UDC_r(j)\} + dceidle_j, \text{ 其中, } \forall j \in \{1, 2, \dots, m\} \quad (5)$$

式中, DC_j 的能量消耗包括新增虚拟机 VM_i 的能量消耗和其在空闲时间内的能量消耗。新分配虚拟机的能量消耗是通过考虑 DC_j 在所有3个维度的价值的最大值减去 DC_j 在所有3个维度的价值的最小值,然后乘以利用效率的最大值。这样 DC_j 的整体能量消耗(Total Energy Consumption, TEC)可通过式(6)进行计算。

$$TEC = \sum_{j=1}^m energyDC(j) \quad (6)$$

与每个数据中心关系密切的这两个参数就是在跨数据中心进行虚拟机分配的总价值参数 $dcprice$ 和其对应的能量消耗参数 TEC 。我们定义 DC_j 的目标函数为 $fnDC$,它可以通过式(7)来表示。

$$fnDC(j) = \{\omega_{en} \times energyDC_r(j)\} + \left\{ \omega_{ex} \times \left(\sum_{i=1}^n dcprice_j \times V_{ij} \right) \right\} \quad (7)$$

这里 ω_{en} 和 ω_{ex} 是总能量消耗参数和总价值参数分别对应的权重, $\omega_{en} + \omega_{ex} = 1$ 。如果 VM_i 被分配到了 DC_j 中,这里有 $V_{ij} = 1$ 。整体云计算平台的目标函数为式(8):

$$overallfnDC = \sum_{j=1}^m fnDC(j) \quad (8)$$

$overallfnDC$ 是所有在 DC 集中的每个 $fnDC(j)$ 的算术总和。 $overallfnDC$ 用来分析虚拟机的请求和目标函数约束。目标函数的约束条件如式(9)~式(12)。

$$\sum_{i=1}^n CVM_r(i) * V_{ij} \leq CDC_r(j); \forall j \in \{1, 2, \dots, m\} \quad (9)$$

$$\sum_{i=1}^n CVM_r(i) \leq CDCSET_r \quad (10)$$

$$\sum_{j=1}^m V_{ij} = 1; \forall i \in \{1, 2, \dots, n\} \quad (11)$$

$$V_{ij} \in \{0, 1\}; \forall i \in \{1, 2, \dots, n\}, \forall j \in \{1, 2, \dots, m\} \quad (12)$$

式(9)满足了前面提到的虚拟机调度约束条件;

式(10)满足了资源使用能力约束条件;式(11)和式(12)满足了虚拟机放置约束条件,一个虚拟机只能放置到惟一的云数据中心,因为通过 $V_{ij} = 0$ 或 $V_{ij} = 1$ 保证了这个约束条件。

虚拟机分配是一个 NP-Hard 问题,本文应用贪心算法的启发式算法,参考前面公式中计算的 $overallfnDC$ 总体价值参数和总能量消耗 TEC 参数,将虚拟机请求集放置到目标函数最合适的云数据中心之中。

云数据中心控制层使用和收集虚拟机请求集的所有信息,数据中心集 DC ,在面向虚拟机在跨云数据中心分派的过程中,应用了贪心算法的具体规则,例如,随机选择 random allocation,下次适应 next fit,首次适应 first fit,最好适应 best fit 和最坏适应 worst fit,最终完成虚拟机的放置。

3 算法的测试与性能分析

3.1 测试环境

在针对应用程序虚拟机分配时,本文把贪心启发式算法应用到经典的装箱问题解决办法之 Random Allocation Algorithm (RAA) 随机选择算法、下次适应算法 Next Fit Allocation (NFA)、首次适应算法 First Fit Allocation (FFA)、最好适应算法 Best Fit Allocation (BFA) 和最坏适应算法 Worst Fit Allocation (WFA) 之中。

利用某个企业的4个大规模云数据中心作为测试环境,访问云服务器的客户端的物理节点硬件组成情况如下表1,云数据中心的物理节点情况配置如表2,虚拟机请求通过一个统一的Web应用程序访问,主要包括3类虚拟机,每个虚拟机对资源的请求需求如表3。

为了仿真需要,实验中改变虚拟机请求个数从25到250个,其中步长为25。因为本文利用的云数据中心中的物理主机是同构的,且每个云数据中心中的主机个数是常量,所以针对启发式算法而言,首次适应算法 FFA 和最好适应算法 BFA 的实验结果应该是一样的,所以我们的实验结果统计主要针对于 RRA、WFA、BFA、NFA 四类算法。

表1 云客户端配置

配置参数	配置值
CPU类型	Intel (R) Core (TM) i7-3770
CPU的主频	3.4 GHz
内核的个数	4
内存	4 GB
云平台环境	Java

表2 云数据中心中的物理节点配置

配置参数	配置值
CPU类型	Intel (R) Core (TM) i7-3770
CPU主频	3.0 GHz
内核数	4
内存	4 GB
每个中心物理节点数	40
云数据中心个数	4

表3 应用程序端的虚拟机需求情况

类型	所需内核个数	所需CPU主频(GHz)	所需内存大小(GB)
1	1	1	1
2	2	2	2
3	4	3	4

3.2 云数据中心的数量测试

表4显示了随着虚拟机数目的增加,被分配了云数据中心的个数的变化情况,从表可以看出,NFA,BFA及RAA三类启发式算法 DC_j 随着虚拟机的个数的增加而增加,但是在WFA算法中,云数据中心的个数一直保持在4个,这说明每个云服务器都被分配了虚拟机.得到这个实验结果的原因是WFA算法一直会分配虚拟机到那些拥有最大数量物理资源的数据中心.

表4 云数据中心个数随的虚拟机请求的变化(单位:个)

启发式算法虚拟机请求个数	RAA	WFA	BFA	NFA
25	1	2	3	3
50	2	4	3	4
75	3	4	3	4
100	3	4	1	1
125	3	4	1	1
150	3	4	1	1
175	3	4	1	2
200	4	4	1	2
225	4	4	2	2
250	4	4	2	2

因此,每个虚拟机被放置到数据中心的时候,它都被访问到,进一步观察结果可以得到,BFA比NFA使用更加少的云数据中心个数,这是因为在分配虚拟机的时候,BFA算法首先考虑的是云数据中心,而NFA算法往往首先考虑的是数据中心中的之前的虚拟机分配情况.

3.3 云数据中心的整体代价测试

表5显示了随着虚拟机请求数量的变化,云平台整体资源参数 $dcprice$ 的变化.由于表4已经表明被分配虚拟机的云数据中心数量是随着虚拟机请求个数的增加,基于这个事实,价值向量 $dcpricevector$ 也就是系统所包含的软硬件资源数量也是随着云数据中心的

增加而增加,整个系统的总体资源可以通过式(13)进行计算.

$$\begin{cases} Totalprice = \sum_{j=1}^m \sum_{i=1}^n dcprice_j \times V_{ij} \\ V_{ij} \in \{0, 1\}, \forall i \in \{1, 2, \dots, n\}, \forall j \in \{1, 2, \dots, m\} \end{cases} \quad (13)$$

如果 VM_i 被分配到了 DC_j 中,这里有 $V_{ij} = 1$.

表5 云数据中心总价值向量随着的虚拟机请求个数的变化

启发式算法虚拟机请求个数	RAA	WFA	BFA	NFA
25	100	80	40	40
50	190	170	90	90
75	240	250	150	150
100	340	350	200	200
125	400	450	250	255
150	500	510	300	310
175	610	600	390	395
200	700	710	450	455
225	760	740	540	545
250	840	860	600	610

表5的结果表明BFA启发式算法比NFA算法随着虚拟机请求数量的变化,它增加得比较缓慢,它们之间的差异几乎可以忽略;RAA算法和WFA算法的总体价值是不一致的,因为总体价值主要依赖于虚拟机请求被分配到的一个具体云数据中心.

3.4 系统整体能量消耗测试

表6显示了随着虚拟机个数请求变化,云平台的总体能量消耗情况,能量消耗的通过式(6)进行计算,和表4和表5比较起来,BFA和WFA两个被优化的贪心算法之间的差异不是很大,得到这种结果的原因是能量消耗的大小主要依赖于一个具体的云数据中心的利用效率,而不是云数据中心的被分配个数;WFA和RA两个算法中云数据中心的利用效率要低于NFA和BFA算法.NFA和BFA算法中的云数据中心的资源利用效率比较高;整体来所能量消耗都是随着虚拟机请求个数的增加而增加.

3.5 目标函数代价测试

表7显示了随着虚拟机请求个数的增加,系统的总体的代价函数值 $overallfnDC$ 变化情况. $overallfnDC$ 主要通过式(8)来进行计算,同时满足式(9)~式(12)的约束条件, $overallfnDC$ 主要考虑了2类约束条件,能量 TEC 和总体价值 $dcprice$.如果两个权重值都设置为0.5,那么能量 TEC 和总价值 $fnDC$ 都平衡考虑.

从表7中可以看出,如果既考虑云数据中心的数量也考虑云数据中心的效率,那么BFA和WFA

两个算法的差异比较小. NFA 和 FFA 算法的差异要小于 WFA 和 RAA 算法, 因为 DC_3 中被使用的个数也具有差异.

表6 云数据中心能量消耗随着的虚拟机请求个数的变化
(单位: kWh)

启发式算法虚拟机请求个数	RAA	WFA	BFA	NFA
25	35	37	33	33
50	50	52	47	47
75	70	71	65	65
100	86	90	82	83
125	104	100	94	95
150	107	119	108	110
175	133	134	126	128
200	149	148	143	144
225	170	168	163	164
250	179	178	165	172

表7 云数据中心总体价值随着的虚拟机请求个数的变化

启发式算法虚拟机请求个数	RAA	WFA	BFA	NFA
25	220	220	170	180
50	360	350	270	280
75	480	470	380	390
100	630	620	490	500
125	710	700	560	580
150	830	840	690	700
175	970	960	670	690
200	1080	1090	920	930
225	1180	1200	1020	1050
250	1300	1260	1110	1130

所有上面的实验结果图都显示 BFA 最好适应算法在跨云数据中心的虚拟机分配的启发式算法中具有比较好的性能.

4 结论与展望

本文运用贪心算法优化了传统的虚拟机分配算法, 云数据中心的物理节点规模将持续不断扩大, 这样将导致能量消耗的增加和资源规模总价值的增加, 本文主要考虑了这两个重要参数, 讨论了虚拟机分配的三类主要约束条件. 但是随着云平台上的应用程序的种类增加和可用性的需求, 为了完成高服务质量 QoS 的需求, 云数据中心的虚拟机分配需要考虑更加多的维度, 这个是本文的后续工作.

参考文献

- Mann ZÁ. Resource optimization across the cloud stack. *IEEE Transactions on Parallel and Distributed Systems*, 2018, 29(1): 169–182. [doi: 10.1109/TPDS.2017.2744627]
- Lei Z, Sun EX, Chen SB, *et al.* A novel hybrid-copy algorithm for live migration of virtual machine. *Future Internet*, 2017, 9(3): 37. [doi: 10.3390/fi9030037]
- Jin XB, Zhang F, Wang L, *et al.* Joint optimization of operational cost and performance interference in cloud data centers. *IEEE Transactions on Cloud Computing*, 2017, 5(4): 697–711. [doi: 10.1109/TCC.2015.2449839]
- Thiruvendakam T, Karthikeyani V. An approach to virtual machine placement problem in a datacenter environment based on overloaded resource. *International Journal of Computer Science and Mobile Computing*, 2014, 3(6): 837–842.
- Gao YQ, Guan HB, Qi ZW, *et al.* A multi-objective ant colony system algorithm for virtual machine placement in cloud computing. *Journal of Computer and System Sciences*, 2013, 79(8): 1230–1242. [doi: 10.1016/j.jcss.2013.02.004]
- Dad D, Yagoubi D E, Belalem G. Energy efficient VM live migration and allocation at cloud data centers. *International Journal of Cloud Applications and Computing*, 2014, 4(4): 55–63. [doi: 10.4018/ijcac.2014100105]
- Li B, Li JX, Huai JP, *et al.* EnaCloud: An energy-saving application live placement approach for cloud computing environments. *Proceedings of 2009 IEEE International Conference on Cloud Computing*. Bangalore, India. 2009. 17–24.
- Liu XF, Zhan ZH, Deng JD, *et al.* An energy efficient ant colony system for virtual machine placement in cloud computing. *IEEE Transactions on Evolutionary Computation*, 2018, 22(1): 113–128. [doi: 10.1109/TEVC.2016.2623803]
- Xu GC, Ding Y, Zhao J, *et al.* A novel artificial bee colony approach of live virtual machine migration policy using Bayes theorem. *The Scientific World Journal*, 2013, 2013: 369209.
- Zhao J, Hu L, Ding Y, *et al.* A heuristic placement selection of live virtual machine migration for energy-saving in cloud computing environment. *PLoS One*, 2014, 9(9): e108275. [doi: 10.1371/journal.pone.0108275]
- Cao ZB, Dong SB. An energy-aware heuristic framework for virtual machine consolidation in cloud computing. *The Journal of Supercomputing*, 2014, 69(1): 429–451. [doi: 10.1007/s11227-014-1172-3]
- Feller E, Rilling L, Morin C. Energy-aware ant colony based workload placement in clouds. *Proceedings of the 2011 IEEE/ACM 12th International Conference on Grid Computing*. Lyon, France. 2011. 26–33.
- Kaouache MA, Bouamama S. Solving bin packing problem

- with a hybrid genetic algorithm for VM placement in cloud. *Procedia Computer Science*, 2015, 60: 1061–1069. [doi: [10.1016/j.procs.2015.08.151](https://doi.org/10.1016/j.procs.2015.08.151)]
- 14 Mishra M, Sahoo A. On theory of VM placement: Anomalies in existing methodologies and their mitigation using a novel vector based approach. *Proceedings of the 2011 IEEE 4th International Conference on Cloud Computing*. Washington, DC, USA. 2011. 275–282.
- 15 Joseph CT, Chandrasekaran K, Cyriac R. A novel family genetic approach for virtual machine allocation. *Procedia Computer Science*, 2015, 46: 558–565. [doi: [10.1016/j.procs.2015.02.090](https://doi.org/10.1016/j.procs.2015.02.090)]
- 16 Hermenier F, Lorca X, Menaud JM, *et al.* Entropy: A consolidation manager for clusters. *Proceedings of 2009 ACM SIGPLAN/SIGOPS International Conference on Virtual Execution Environments*. Washington, DC, USA. 2009. 41–50.
- 17 Dupont C, Schulze T, Giuliani G, *et al.* An energy aware framework for virtual machine placement in cloud federated data centres. *Proceedings of the 3rd International Conference on Future Systems: Where Energy, Computing and Communication Meet*. Madrid, Spain. 2012. 1–10.
- 18 Choudhary M, Peddoju SK. A dynamic optimization algorithm for task scheduling in cloud environment. *International Journal of Engineering Research and Applications*, 2012, 2(3): 2564–2568.
- 19 Sharma RK, Sharma N. A dynamic optimization algorithm for task scheduling in cloud computing with resource utilization. *International Journal of Scientific Engineering and Technology*, 2013, 2(10): 1062–1068.
- 20 Mann ZÁ. Interplay of virtual machine selection and virtual machine placement. *Proceedings of the 5th European Conference on Service-Oriented and Cloud Computing*. Vienna, Austria. 2016. 137–151.
- 21 Ahamed F, Shahrestani S, Javadi B. Security aware and energy-efficient virtual machine consolidation in cloud computing systems. *Proceedings of 2016 IEEE Trustcom/BigDataSE/ISPA*. Tianjin, China. 2016. 1516–1523.