



研究与开发

云联盟三层架构：博弈型 QoS 建模

马焜^{1,2,3}, 胡创悦¹, 张玉志², 马方超⁴, 王晓冬⁵, 党中华³, 肖文红³, 陈双喜^{3,6}

- (1. 嘉兴市长三角区块链技术研究院, 浙江 嘉兴 314006;
2. 南开大学软件学院, 天津 300350;
3. 嘉兴职业技术学院嘉兴市工业互联网安全重点实验室, 浙江 嘉兴 314036;
4. 浙江省公安厅, 浙江 杭州 310009;
5. 浙江省教育技术中心, 浙江 杭州 310014;
6. 浙江大学工程师学院, 浙江 杭州 310058)

摘要: 面对国内外大型公有云供应商的激烈竞争, 中小云厂商的生存难度加大。为此, 建立一个基于相互合作的云联盟成为了这些厂商的一种可行策略。然而, 在追求个体最大利益和保障联盟整体服务质量 (quality of service, QoS) 之间存在着复杂的博弈关系。针对上述问题, 一种基于 QoS 的云联盟模型被提出, 其涵盖云计算的三层架构。在应用层至虚拟层, 引入了一种基于差分进化 (differential evolution, DE) 算法的创新任务分配策略, 专门用于处理多 QoS 任务分配问题。在虚拟层至物理层, 设计了合作与竞争并存的虚拟机迁移模型, 适用于在云联盟博弈计算环境下实现虚拟机迁移的能耗与 QoS 之间的平衡。实验结果表明, 所提出的解决方案改进了云计算环境的服务质量, 并揭示了在云联盟环境中, 合作和竞争两种模式的相对优势。

关键词: 云联盟; 多项服务质量; 动态调度算法; 虚拟机迁移; 能耗优化

中图分类号: TP393

文献标志码: A

doi: 10.11959/j.issn.1000-0801.2024200

Cloud federation three-layer architecture: game-theoretic QoS modeling

MA Kun^{1,2,3}, HU Chuangyue¹, ZHANG Yuzhi², MA Fangchao⁴, WANG Xiaodong⁵,
DANG Zhonghua³, XIAO Wenhong³, CHEN Shuangxi^{3,6}

1. Jiaxing Yangtze Delta Region Blockchain Technology Research Institute, Jiaxing 314006, China
2. College of Software, Nankai University, Tianjin 300350, China
3. Jiaxing Key Laboratory of Industrial Internet Security, Jiaxing Vocational and Technical College, Jiaxing 314036, China
4. Zhejiang Provincial Department of Public Security, Hangzhou 310009, China

收稿日期: 2024-01-06; 修回日期: 2024-08-08

通信作者: 陈双喜, abelchen@zju.edu.cn; 肖文红, zjjxxwh@163.com

基金项目: 浙江省重点研发计划项目 (No.2024C01066); 嘉兴市自然科学基金资助项目 (No.2023AY11011, No.2022R470A008); 浙江省科技创新基金资助项目 (No.2022R470A008)

Foundation Items: The Key Research and Development Program of Zhejiang Province (No.2024C01066), Jiaxing Municipal Natural Science Foundation (No.2023AY11011, No.2022R470A008), The Science and Technology Innovation Program of Zhejiang Province (No.2022R470A008)

5. Zhejiang Provincial Education Technology Center, Hangzhou 310014, China

6. School of Engineering, Zhejiang University, Hangzhou 310058, China

Abstract: Facing intense competition from large public cloud providers both domestically and internationally, the survival difficulties of small and medium cloud providers have increased. To address this, establishing a cloud federation based on cooperation has become a viable strategy for these providers. However, there is a complex interplay between pursuing individual maximum benefits and ensuring the overall quality of service (QoS) of the federation. A QoS based cloud federation model was proposed to address the above issues, covering a three-layer architecture of cloud computing. At the application to virtual layer, an innovative task allocation strategy based on the differential evolution (DE) algorithm was proposed, specifically designed to handle multi-QoS task distribution challenges. At the virtual to physical layer, a model for virtual machine migration that coexists with cooperation and competition was proposed, suitable for balancing energy consumption and QoS in a cloud federation game-theoretic computing environment. The experimental results indicate that the proposed solution improved the service quality of cloud computing environments and revealed the relative advantages of running cooperation and competition modes in cloud federation.

Key words: cloud federation, multi-QoS, dynamic scheduling algorithm, VM migration, energy consumption optimization

0 引言

工业 4.0 的发展离不开物理世界与虚拟世界的互联互通, 其中, 云计算作为网络虚拟化的核心技术发挥了关键作用。以庞大的服务器基础设施为中心, 云计算可以通过网络为用户提供虚拟服务, 这已然成为一种新型的商业模式。该模式承诺以可扩展的、按需付费的方式提供几乎无限的能力和服。云环境不仅通过网络为各种服务提供了大量的虚拟化动态计算资源, 而且还整合了来自各种机构和个人的分布式资源。用户只需要连接一个网络, 就可以获得广泛的计算环境, 无须关心所分配的资源细节或这些资源的部署位置^[1]。

虚拟化是一种技术, 它将物理基础设施抽象出来, 提供被称为虚拟机 (virtual machine, VM) 的虚拟资源以支持高级应用程序^[2]。在云计算领域, 虚拟化发挥了至关重要的作用。它可以从服务器集群中聚合计算资源, 并根据任务需求动态地为服务器分配 VM。对于每一个任务执行请求, 可以初始化一个新的 VM, 或者将任务分配给同一用户的现有 VM^[3]。云计算的发展, 结合其广泛的采纳和应用, 使得云数据中心数量

和规模的显著增长^[4]。然而, 这种增长也带来了挑战, 尤其是大量的能源消耗。云数据中心占全球电力消耗的 1.1%~1.5%, 是能源密集型的实体。这样的趋势对全球能源需求产生了重大影响, 需要进行严格的分析, 同时广泛引述的数据显示, 过去十年全球云数据中心的能源使用量已经翻倍^[5]。因此, 解决云数据中心的能源消耗已经成为一个备受关注的紧迫问题^[6]。

在云计算领域, 除能源消耗外, 直接影响用户的另一个关键因素是服务质量 (quality of service, QoS)。如今, 云系统在提供服务时必须满足每个用户和每个应用的 QoS 要求, 因此用户或应用在从云资源请求服务时必须明确其对 QoS 的需求, 同时 QoS 的满足度是评估云环境中任务调度策略效果的关键指标^[7]。因此, 云系统中的任务调度策略通常会考虑具体的 QoS 目标和可用资源的性能指标, 以确保大量的应用任务能够与可用的计算资源有效匹配^[8-9]。

服务级别协议 (service-level agreement, SLA) 是云服务提供商 (cloud service provider, CSP) 与用户之间订立的基础合同。它明确了提供给用户的预期 QoS, 从而量化用户的期望值。



协议中详细描述了如吞吐量、可靠性、阻塞概率和响应时间等核心指标,还包括有关支付流程和SLA违规的处罚细节。鉴于SLA在保障用户满意度和系统性能方面具有关键作用,动态调度虚拟机资源的研究受到了广泛关注,研究的目的是降低系统资源的过度利用,同时满足规定的SLA标准,为用户提供更为高效和稳定的服务^[10]。

尽管云计算为用户和企业带来了较大的发展机遇,但中小型供应商在面对大型、占市场主导地位的云服务提供商时仍遭遇了巨大的挑战和竞争压力^[11]。为了突破这一困境,企业采取了有效的应对策略,即通过建立合作关系,汇聚多家中小型供应商的资源,从而构建成一个强大的资源集合体——云联盟。云联盟为中小型供应商提供了一种可行的方式,使它们能够在竞争激烈的云计算市场中实现可持续发展^[12]。

云计算环境中的QoS管理问题日益凸显,该问题包含了用户端(应用层)到虚拟端(虚拟层),再到物理端(物理层)的三层架构,这涉及两个关键模型,分别是任务资源分配模型以及虚拟资源分配模型。云计算三层资源模型如图1所示。

云计算三层资源模型中,物理层由数据中心组成,这一层以主机服务器的形式容纳各种硬件

资源。虚拟层位于物理层之上,它将物理主机划分为虚拟主机以增强资源管理。应用层位于虚拟层之上,为用户提供一系列的服务,包括软件即服务(software as a service, SaaS)、平台即服务(platform as a service, PaaS)和基础设施即服务(infrastructure as a service, IaaS)。

从服务的角度看,云计算模型分为两个模块,一个是任务分配管理模块,为用户提供一种成本效益的方法来分配虚拟资源;另一个是虚拟资源调度模块,建立物理资源与虚拟机之间的对应关系。本研究假设每台主机至少提供1个VM。

本研究深入探讨了虚拟机管理任务分配的机制及其后续如何迁移到物理资源的博弈层面。该研究为云服务提供商(cloud service provider, CSP)提供了参与资源共享的灵活性,同时,云联盟中的不同CSP之间既可以是合作伙伴,也可能成为竞争对手。本文对文献[13-16]中关于云联盟的模型进一步展开研究,旨在为中小规模的CSP开辟合作的新路径,如在资源需求波动明显(如网络流量高峰期)的情境下,当CSP的资源储备充足时,他们更倾向于选取合作的方式,将富余的资源提供给合作伙伴;相反,当资源有限时,他们会优先确保满足自身用户的需求。

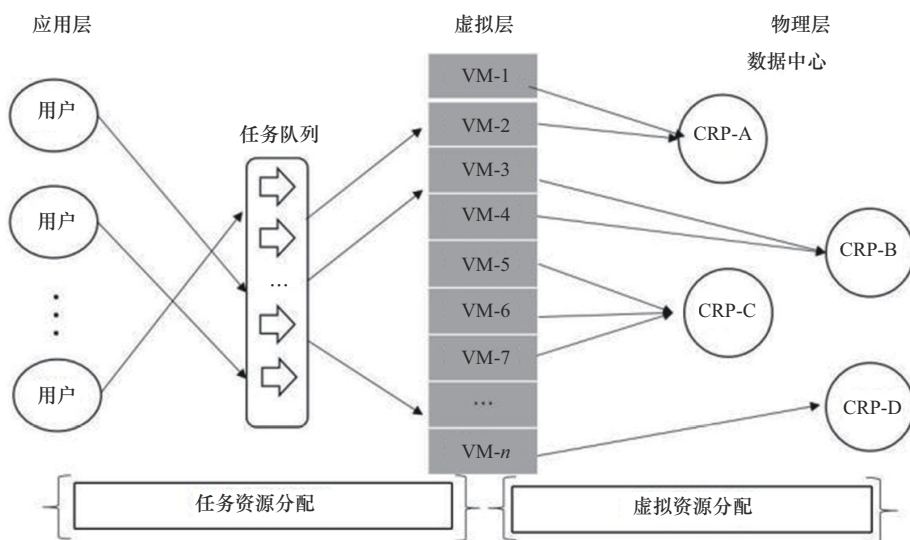


图1 云计算三层资源模型

1 相关工作

目前, 研究人员在云计算任务调度和资源管理领域开展了大量的研究工作。在任务调度方面, 不同的计算资源有不同的规范, 每个任务也有自己的请求, 这意味着在任务请求和计算资源之间找到最佳解决方案是一个非确定性多项式 (non-deterministic polynomial, NP) 难题。在此基础上, 大量的启发式算法被提出, 并用于上述最优匹配问题的近似解。在众多应用任务调度策略中, Min-Min 方法^[17]和 Max-Min 方法^[18]经常被用作评估其他调度策略性能的基础, 这两个原理很相似, 是代表性的经典启发式算法。对于特定的任务调度问题, 一些智能优化算法, 如用于实际科学工作流程的遗传算法^[19], 也被用于近似任务调度问题的全局最优解。针对计算密集型应用任务, Nooriantalouki 等^[20]提出了一种调度方法, 该方法可以同时处理异构平台中的多项应用任务, 这些应用任务可以是单个独立任务, 也可以是多个任务 (任务包)。Li 等^[21]提出了一种针对任务类型偏好和资源服务能力设计的混合博弈模型, 采用基于多业务的进化博弈调度算法, 对 QoS 进行了优化。考虑任务调度的 QoS 约束, Tang 等^[22]提出了一种预算约束的任务调度器, 该调度器能够在不同的最小完成时间约束、最大预算约束和不同的性能参数下处理多个任务包。

虚拟机迁移同样也是一种资源分配形式, Baskaran 等^[23]提出了一种基于模糊软集的虚拟机 (fuzzy soft set virtual machine, FSS-VM) 合并算法, 该算法针对 11 种不同的选择参数组合进行了实验测试, 其中每个组合都被视为模糊软集, 并与现有算法进行了各种度量的比较, 旨在提高虚拟机提供的 QoS。Gharehpasha 和 Masdari^[24]在云数据中心集成运行的虚拟机时考虑了内存利用率, 使用正弦余弦算法 (sine cosine algorithm,

SCA) 和蚁狮优化器 (ant lion optimizer, ALO) 的组合作为离散多目标和混沌函数进行最优 VM 分配, 成功整合后, 将关闭空闲的计算节点。Padmavathy 等^[25]考虑了虚拟机集成问题中的迁移数量, 并建立了一个虚拟机集成模型, 该模型最大限度地减少了计算节点的数量和虚拟机迁移时间。传统的重搜索在虚拟机迁移的相关领域已经相对成熟。在这样的环境下, Verma^[26]仍然找到了突破, 并引入了一种新的改进优化算法, 即双条件 Moth-Flame 算法 (dual-condition moth-flame algorithm, DC-MFA), 该算法考虑了所提出的多目标函数, 分别是 CPU 利用率、能耗、安全性、制造跨度、迁移成本和资源成本。

本文总结了几个典型非联盟和联盟的相关研究工作, 见表 1。

2 系统模型

本节深入探讨了任务分配模型的设计细节, 首先对任务分配模块进行了全面的 QoS 公式化, 为后续的解释奠定了基础; 在此基础上, 详细描述了任务分配过程的架构, 这一架构旨在模仿其核心的算法设计; 最后, 介绍了基于差分进化算法绑定策略 (differential evolution-based banding policy, DEBBP), 这是支撑过程中任务分配实施的关键组成部分。

2.1 任务分配模块的 QoS 公式

云服务中的 QoS 可以用不同的属性来衡量, QoS 属性见表 2。所提出的模型中所使用符号的含义见表 3。

本文提出了一个名为多 QoS 目标调度模型 (multiple QoS task scheduling model, M-QoS-TSM) 的新型任务调度模型, 作为任务分配管理难题的解决方案。M-QoS-TSM 模型的核心是 3 个关键的 QoS 目标参数: 调度期限, 调度预算, 一个综合可靠性、可用性和安全性三者性能的参数。



表1 几个典型非联盟和联盟的相关研究工作总结

研究者	类型	总结
Dahan 等 ^[27]	非联盟	提出了一种基于高效 Agent 的蚁群优化 (ant colony optimization, ACO) 算法来解决云服务组合 (cloud service composition, CSC) 问题, 该算法是由不同服务 CSP 提供的具有不同 QoS 属性的相似服务的扩散而引起的
Yu 等 ^[28]	非联盟	设计了一种新的在线调度框架, 称为最小距离树 (minimal distance tree, MDT), 它由离线调度阶段和在线调度阶段组成, 可以对动态 VM 迁移提供快速响应
Patel 等 ^[29]	非联盟	提出了一种能量仓库虚拟机迁移算法来平衡物理机 (physical machine, PM) 上的负载。由临时峰值负载引起的 VM 迁移是云中的关键问题之一, 严重影响了云的性能。为了避免这些不必要的迁移, 该方法采用了一种预测方法, 在此之前预测 PM 上的负载
Vijin 等 ^[30]	联盟	创新的经济共享模型用于在 IaaS 云联盟中共享容量, 这可以使用 CSP 之间的交互来实现。这是一种重复的虚拟机游戏, 可以识别现货市场中所有未使用的容量, 它们的工作更侧重于安全条件, 而不是联盟设计本身
Ahmend 等 ^[31]	联盟	提出了基于 QoS 的信任评估机制, 允许选择最有用的 (基于实用价值的) 提供者, 这些提供者可以用于支持多个不同应用领域的云服务联合
Khan 等 ^[32]	联盟	提出了一种混合优化算法来处理音频环境下的 VM 迁移, 将 Cuckoo 搜索优化算法和部分优化算法相结合, 得到了所提出的混合优化模型, 其主要研究目标是减少能量消耗、计算时间和迁移费用

表2 QoS 属性

属性名	含义
执行成本 (P_{ex})	CSP 执行任务所需要的成本
执行时间 (T_{ex})	CSP 执行任务所需要的时间
可靠性 (R_{el})	云服务正常运行的概率
可用性 (A_{va})	成功访问云服务的概率
安全性 (S_{ec})	云服务的安全级别

调度期限规定了在所有虚拟机上执行任务的最大允许时间。调度预算概述了在预定目标环境内完成所有任务的全面成本。可靠性、可用性和安全性不仅仅是技术层面的术语, 更代表着 CSP 的核心价值, 因为它们能够有效区分不同 CSP 之间的差异。这些差异可能会影响用户的选择和偏

好, 使得基于这些属性的考量, 某些 CSP 在市场上比其他 CSP 更受欢迎。

云计算虚拟资源由虚拟机组成, 可以恰当地表示为一个集合, $VM = \{i, i \in 1, \dots, N\}$, N 表示 VM 的总数。同样, 等待调度的应用任务被表示为任务集 $T = \{j, j \in 1, \dots, n\}$, n 表示任务总数。

从用户的角度来看, 云计算系统应该允许它们按时完成任务, 同时节省执行成本。这两个目标可以采用多目标优化模型来表达。需要注意的是, 其中执行时间以需要花费最长时间的 VM 为准 (虚拟机可以同时工作), 执行成本则只要计算总值, 两者同时受到期限和预算的限制, 表达

表3 符号的含义

符号	含义
S	表示在任务调度过程完成时任务和虚拟机之间的映射
$\alpha(i, j)$	布尔值, 用于显示任务 i 和虚拟机 j 之间是否匹配
$F(S)'$	任务和虚拟机之间执行时间和执行成本的影响向量
$1 - \frac{T_{ex}(i, j)}{D_{ead}}$	完成任务后与节省的时间相关的得分
$1 - \frac{P_{ex}(i, j)}{B_{udg}}$	完成任务时与节省的成本相关的得分
$RAS(S)$	系统 S 的可靠性、可用性和安全性三者得分总和
$R_{el}(j), A_{va}(j), S_{ec}(j), m_j$	$R_{el}(j), A_{va}(j), S_{ec}(j)$ 分别表示虚拟机 j 的可靠性、可用性和安全性; m_j 表示第 j 个虚拟机被执行的任务数量
$F(S)$	表示映射 S 得分的最终适应度函数
α, β	权重因子分别调整执行时间和价格的重要性

式如下。

$$\min_{i,j} \begin{cases} \max_{j \in N} \sum_{i=1}^n T_{\text{ex}}(i,j) \cdot \alpha(i,j), \text{其中执行时间} & \max_{j \in N} \sum_{i=1}^n T_{\text{ex}}(i,j) \leq D_{\text{ead}} \\ \sum_{i=1}^n \sum_{j=1}^N P_{\text{ex}}(i,j) \cdot \alpha(i,j), \text{其中执行成本} & \sum_{i=1}^n \sum_{j=1}^N P_{\text{ex}}(i,j) \leq B_{\text{udg}} \end{cases} \quad (1)$$

其中, $\alpha(i,j) = \begin{cases} 1, \text{任务} i \text{与虚拟机} j \text{匹配} \\ 0, \text{任务} i \text{与虚拟机} j \text{不匹配} \end{cases}$, D_{ead} 为

虚拟机上所有任务的执行期限, B_{udg} 为分配在虚拟机上执行任务的总预算, $T_{\text{ex}}(i,j)$ 为任务 T_i 的执行时间 ($i \in n$)。在 VM_j 上, $j \in n$, $P_{\text{ex}}(i,j)$ 为与虚拟机 VM_j 执行任务 T_i 相关联的价格。因此, 定义提出的模型中任务匹配结果的综合适应度函数如下。

$$F(S)' = \alpha \left(1 - \frac{T_{\text{ex}}(i,j)}{D_{\text{ead}}} \right) + \beta \left(1 - \frac{P_{\text{ex}}(i,j)}{B_{\text{udg}}} \right) \quad (2)$$

$$\text{RAS}(S) = \sum_{j=1}^{m_j} (R_{\text{el}}(j) + A_{\text{va}}(j) + S_{\text{ec}}(j)) \cdot m_j \quad (3)$$

$$F(S) = \begin{cases} F(S)' \cdot \text{RAS}(S), & F(x_i)' \neq 0 \\ 0, & F(x_i)' = 0 \end{cases} \quad (4)$$

其中, $F(S)'$ 为执行时间和执行成本的得分值, α 和 β 分别为调整执行时间和执行成本权重的参数, $\text{RAS}(S)$ 为可靠性 (reliability)、可用性 (availability)、安全性 (security) 的综合得分, m 为目标 VM 的数量。因此, $F(S) = F(S)' \cdot \text{RAS}(S)$ 为相应任

务匹配结果的得分, 即适应度函数。

M-QoS-TSM 的体系结构如图 2 所示。图 2 描述了任务分配过程的完整流程。

步骤 1: 用户提交任务及相应的 QoS 要求到任务调度器 (在数据中心), 同时, VM 将其资源信息发送到任务调度器。

步骤 2: 调度器基于 DEBBP 策略生成仿真结果。

步骤 3: 调度器评估返回的信息。

步骤 4: 判断 VM 是否能够完成用户的任务请求, 若可以, 则接受仿真结果并将任务分配给获取的 VM。否则, 调度器由于资源能力限制拒绝请求, 并通知用户拒绝的原因。

步骤 5: VM 根据接收到的仿真结果执行分配的任务。

2.2 基于 DE 的绑定策略

如前文所述, 云计算环境中的任务调度是一个 NP 难题。在参与者人数众多的情况下, 很难找到最佳解决方案。通常的方法是应用各种智能优化算法来逼近其最优解, 作为满意解。差分进

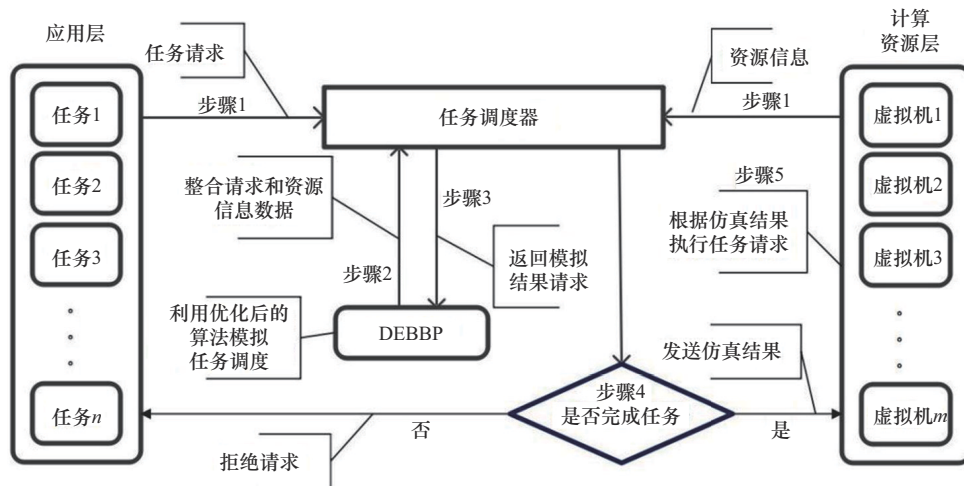


图2 M-QoS-TSM的体系结构



化 (differential evolution, DE) 算法^[33]就是得到近似最优解的算法之一。根据模型需求对DE算法进行相应的调整,从而完成用户任务到VM的匹配,主要设置参数为群体大小 (NP)、控制突变的参数 (f) 及交叉概率 (CR),具体步骤如下。

(1) 初始种群。创建种群 X 由个体 x_k 组成,在不同维度的个体用 $x_{k,w}$ 表示,其中 $k=1, \dots, NP$ 。 $x_{k,w}^L \leq x_{k,w} \leq x_{k,w}^U, k=1, 2, \dots, NP$, 其中 $x_k(0)$ 是第 k 个个体, w 表示第 w 个维度。因此,个体在每个维度上的初始值通过式 (5) 生成:

$$x_{k,w} = x_{k,w}^L + \text{rand}(0, 1) \cdot (x_{k,w}^U - x_{k,w}^L) \quad (5)$$

其中, $x_{k,w}^L$ 、 $x_{k,w}^U$ 分别表示第 w 维度的下界和上界, $\text{rand}(0, 1)$ 表示 0 和 1 之间的随机数。

在 DEBBP 中,个体 x_k 表示虚拟机和任务之间的一个匹配结果, x_1 表示 VM_1 , x_1 的值对应 VM_1 上匹配的任务, $x_1, x_2, \dots, x_{NP}$ 是一个包含 NP 个 VM 的匹配结果。每个维度都代表一种匹配结果,并对应一个 $f(-)$ 。

(2) 变异。DE 算法通过差分策略实现个体的变异。常用的差分策略是在种群中随机选择两个不同的个体: x_{r2} 和 x_{r3} , 再使用一个缩放因子 f 来缩放 x_{r2} 和 x_{r3} 之间的差异,接着添加另一个随机种群 x_{r1} 来生成突变向量 V_i^{g+1} :

$$V_k^{g+1} = x_{r1}^g + f(x_{r2}^g - x_{r3}^g) \quad (6)$$

其中, $r1$ 、 $r2$ 和 $r3$ 是 $[1, NP]$ 中的 3 个随机数, f 是一个确定的常数, g 表示第 g 代。

(3) 交叉。交叉操作的目的是随机选择个体。交叉操作的方法可以描述为:

$$\begin{aligned} u_{k,w}^{g+1} &= V_{k,w}^{g+1}, \text{rand}(0, 1) \leq CR, j=1, \dots, D \\ u_{k,w}^{g+1} &= x_{k,w}^g, \text{rand}(0, 1) > CR, j=1, \dots, D \end{aligned} \quad (7)$$

其中, CR 为交叉概率。交叉操作是指优化过程。如果 $F(u_{k,w}^{g+1}) < F(x_{k,w}^g)$, 表明交叉后的 ($g+1$) 这一代优于原来的 g 这一代, 则 $x_{k,w}^g = u_{k,w}^{g+1}$, 其中, $F(-)$ 为适应度函数式 (4)。

应用DE的伪码描述如算法1所示。

算法1 应用DE的伪码描述

输入: 种群大小 NP , 控制变异的参数 f , 交叉概率 CR , 适应度函数 $F(x)$, 一个随机种群 x_G^k , 及重复的代数。

输出: 相对最优解 (任务调度结果), 即最后一代种群中适应度最好的个体。

初始化 $t=0$

当 $t \leq$ 迭代次数时, 重复以下步骤

针对每一个从种群 X 中的向量 x_G^k

根据式 (6) 生成突变向量 $v_G^{k,k}$

根据式 (7) 进行向量交叉

如果 $F_{g+1}^k < F_g^k$, 那么 $x_{g+1}^k = U_{g+1}^k$

结束针对每一个向量的循环

$t=t+1$

结束整个循环

返回最后一代种群中适应度最好的个体

以上描述了DE算法通过一个差分策略来生成新的个体,并通过交叉操作来增加种群的多样性,最终经过迭代选择适应度最好的个体 (任务分配结果) 为输出。

3 虚拟机迁移模型

虚拟机迁移通常在物理资源层和虚拟资源层之间。其中,虚拟资源层为应用层提供操作载体,它位于物理资源层之上,具有虚拟服务器功能。这一层有能力在单个虚拟机中容纳多个应用程序,表明应用程序和虚拟机之间的多对一映射关系。物理资源层在数据中心内包含各种类型的硬件服务器,为虚拟资源层提供了必要的硬件资源。这一层同样遵循多对一的映射,其中单个物理机可以容纳多个虚拟机。

考虑提出模型中应用服务的访问连接要求的动态性,需要一个响应灵活的云资源管理平台。该平台可以根据不断变化的应用服务需求来调整虚拟机的数量,可伸缩性支持VM在主机之间的

多次迁移, 并显著影响物理服务器的整体利用率、资源分配的均衡性和数据中心的能源消耗。

3.1 云联盟模型

在云联盟领域, 合作模型通常表现出不同的CSP和谐地协作。首先, 它们共享资源, 但每个服务提供者仍然保持一个明确和独立的实体身份, 并在彼此之间维持清晰的边界。其次, 在这个云的大集合中运行的应用程序并不知晓彼此的具体位置, 这是通过虚拟局域网来实现的, 它促进了不同应用组件之间的通信。最后, 这些云服务提供者根据自身成本、信任水平等因素进行差异化定位。

云计算允许分配给用户任务的虚拟机可以有

$$\max D(j, k) = P(j_y, k_x) (R_{pn}(k) - D_{vn}(j)),$$

$$R_{pn}(j) \geq D_{vn}(k) \quad \forall j \in V, k \in P; \forall j \in V, k \in P \rightarrow P(j_y, k_x) \in \{0, 1\}$$
(8)

其中, $P(j_y, k_x)$ 是模型中用于区分合作和竞争云计算的二进制参数, 如式(9)所示。

$$P(j_y, k_x) = \begin{cases} 1, x \neq y, \text{合作云联盟} \\ 0, x = y, \text{竞争云联盟} \end{cases} \quad (9)$$

不同的迁移路径。这些机器可以找到自己在用户当前的CSP的物理资源, 或者其他CSP的物理资源。将虚拟资源分配给物理域的这种灵活性导致了两种可能的情况: 一种是合作模型, 其中用户的虚拟机可以自由迁移到任何位置; 另一种是竞争模型, 这些虚拟机被限制在原始CSP的物理基础设施范围内迁移。云联盟资源管理框架如图3所示。

在提出的模型中, CSP_x 的 PM_k 可用的资源由 $R_{pn}(k_x)$ 表示, 而需要被迁移的 VM_j 在迁移期间对资源的需求由 $D_{vn}(j_y)$ 表示。云联盟问题包括为每个在等待序列的 VM_j 找到物理资源 PM_k 的映射, 该映射最大化了式(8)的效用函数 $D(j, k)$:

3.2 VMMNDSA

为了配合上述合作云和竞争云中的虚拟机迁移, 本文提出的模型引入了基于虚拟机迁移必要性的动态调度算法(virtual machine migration neces-

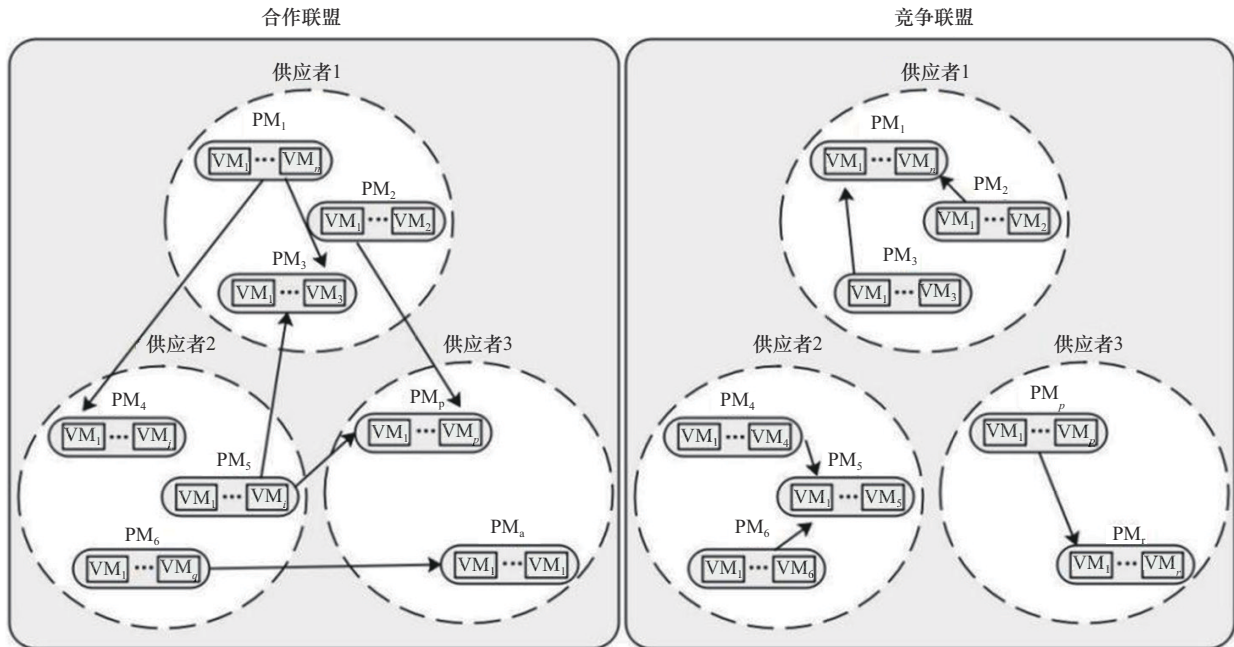


图3 云联盟资源管理框架



sity dynamic scheduling algorithm, VMMNDSA), 这是通过云资源的动态调度模型以及利用在文献[34]中讨论的虚拟机迁移技术来实现的。

此算法考虑了物理节点上的负载, 评估了虚拟机迁移而可能产生的潜在损失, 并有策略地定位目标主机, 主要目的是确保通过有效的资源调度实现高效的负载平衡。基于迁移技术的VMMNDSA体系结构如图4所示, 可分为5个步骤。

(1) 热节点负载评估: 将负载大于设定阈值上限或小于阈值下限的物理节点PM定义为热节点。

(2) 虚拟机迁移损失评估: VMMNDSA考虑迁移成本, 不会通过VM迁移消除所有热节点, 其目的是选择迁移损失比例低的VM作为目标, 因此, 综合排序CPU和内存使用率最低的VM作为被迁移目标。

(3) 脏页跟踪: 一旦开始迁移虚拟机, 虚拟机将无法工作, 直到迁移完成, 它们的内存将通过多次迭代从热节点传输到目的计算节点, 即传输的内存页大小是被修改的部分, 被修改的页称为“脏页”。最后, 热节点停止虚拟机操作, 并将最后一次迭代生成的“脏页”传输到目的计算节点。脏页管理用于管理脏页的转换过程。

(4) 目标节点PM选择: VMMNDSA根据虚拟机需求与所有候选物理节点之间不同资源的差值, 来确定匹配方法。将被迁移的VM和所有可选PM的CPU内存资源差值进行综合排序, 选取差值最大的作为目标PM。

(5) 完成虚拟机迁移和迁移日志更新。

4 实验结果

实验环境为CloudSim, 是由墨尔本大学和Gridbus项目共同开发的用于模拟云计算环境的工具包^[35], 其由Java语言编写, 包含了完整的云计算基础架构, 支持对云计算模拟实验所包含的任务、虚拟机以及物理机进行数量和参数设置, 还支持任务分配策略及虚拟机迁移策略的自定义编写。实验选取的CloudSim模拟运行环境如下: CPU为英特尔奔腾双核P6000, 时钟频率为1.87 GHz; 内存为16.0 GB; 机械硬盘(hard disk drive, HDD)为1 TB。

4.1 任务分配管理模型实验结果的DEBBP实现

实验使用了8个计算节点, 每个节点都有明确的属性参数。它们的性能配置和计算能力是根据Amazon EC2按需实例定价模型建模的。任务分配管理模型的VM见表4, 其中MIPS为CPU性能。

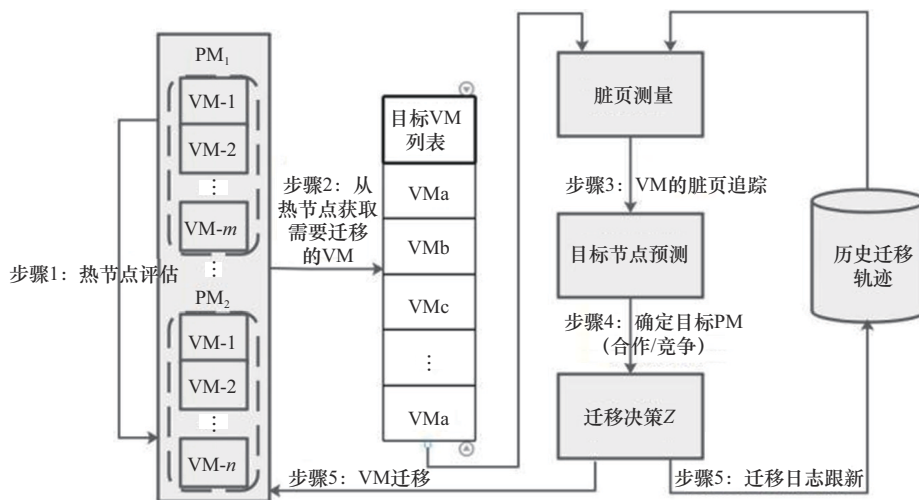


图4 基于迁移技术的VMMNDSA体系结构

表4 任务分配管理模型的VM

属性	VM-0	VM-1	VM-2	VM-3	VM-4	VM-5	VM-6	VM-7
执行成本	1	1	1	1	1	1	1	1
MIPS	101	410	213	54	55	363	70	118
价格	5	20	10	5	5	7	6	11
可用性 (Ava)	40	10	20	20	30	50	10	30
可靠性 (Rel)	10	10	10	60	20	40	10	20
安全性 (Sec)	20	40	10	40	10	60	50	30

实验模拟了两项任务：任务A (Job A) 和任务B (Job B)。这两项任务使用了3种绑定策略进行测试：DEBBP策略、CloudSim默认的绑定策略 (Conventional) 以及基于Max-Min的绑定策略，为VM分配任务Cloudlet。两项任务都包括需要VM分配的20个Cloudlet，任务分配管理模型的任务见表5。模拟的权重因子被设定为 $\alpha=\beta$ ，调度的期限和预算分别为100和1 800。关于资源分配方法的更多详细信息可以参考文献[36]。

表5 任务分配管理模型的任务

Cloudlet ID	Job A (长)	Job B (长)
0	2 020	930
1	700	110
2	170	310
3	100	1 300
4	440	840
5	620	120
6	710	310
7	660	60
8	820	2 220
9	820	290
10	1 100	600
11	300	820
12	300	1 150
13	40	670
14	1 750	930
15	460	260
16	300	300
17	110	410
18	550	500
19	900	90

该DE算法的操作参数设置如下：种群大小设置为100，最大值和最小值分别设置为8和-8，突变缩放设置为0.5、交叉设置为0.1，最大迭代设置为2 000。Job A和Job B的模拟结果见表6和表7，Job A和Job B的执行时间和执行成本实验结果如图5和图6所示，RAS实验结果如图7所示。

实验比较了DEBBP与Conventional的绑定策略和Max-Min绑定策略在每个作业中的性能，以及总执行时间、总执行成本和RAS。从图5 (a) 和图5 (b) 可以看出，对于Job A和Job B，DEBBP中Cloudlet的执行时间均比Conventional绑定策略和基于Max-Min的绑定策略的执行时间短。对于Job A，DEBBP的总执行时间比Conventional绑定策略短68.61%，比基于Max-Min的绑定策略短8.97%。对于Job B，DEBBP的总执行时间比Conventional绑定策略短73.84%，比基于Max-Min的绑定策略短13.53%。从图6 (a) 和图6 (b) 可以看出，对于Job A，DEBBP的总执行成本比Conventional绑定策略低37.16%，比基于Max-Min的绑定策略低22.79%。同样，对于Job B，DEBBP的总执行成本比Conventional绑定策略低37.64%，比基于Max-Min的绑定策略低23.17%。从图7 (a) 和图7 (b) 可以看出，对于Job A，DEBBP的总RAS分别比常规绑定和基于Max-Min的绑定策略高47.06%和43.31%。对于Job B，DEBBP的总RAS分别比常规绑定和基于Max-Min的绑定策略高44.44%和39.87%。

从模拟结果可以看出，CloudSim的Conventional绑定策略没有对QoS进行优化，因此其QoS的性能远低于另外两种绑定策略，而DEBBP和Max-Min相比，得益于其更适合资源分配的进化策略，其性能也有着显著的优势。

4.2 用于VM迁移的VMMNDSA的实施

对于虚拟机VM的迁移，实验设置了4种类型的虚拟主机，每种类型由20个不同的虚拟机组成，虚拟机具有MIPS、Pes、RAM、带宽和存储



表6 Job A模拟结果

Cloudlet ID	DEBBP				Max-Min				Conventional			
	VM ID	执行时间/s	执行成本/元	RAS	VM ID	执行时间/s	执行成本/元	RAS	VM ID	执行时间/s	执行成本/元	RAS
0	1	4.93	98.51	50	1	4.93	98.51	50	0	8.12	40.59	70
1	2	3.28	32.85	40	0	6.92	34.62	79	1	1.71	34.11	50
2	5	0.47	3.28	150	3	3.13	15.66	110	2	0.80	7.98	40
3	5	0.28	1.93	150	2	0.47	4.69	40	3	1.93	9.63	110
4	5	1.21	8.48	150	7	3.73	40.98	80	4	7.99	39.93	60
5	5	1.71	11.94	150	6	8.86	53.14	70	5	1.71	32.11	50
6	2	3.41	34.06	40	5	1.95	13.68	150	6	10.14	60.82	70
7	1	1.66	33.11	50	2	3.09	30.95	40	7	5.59	61.46	80
8	5	2.26	15.81	150	7	6.96	76.61	80	0	8.12	40.59	70
9	5	0.55	3.86	150	6	2.89	17.32	70	1	0.49	9.72	50
10	0	10.96	54.79	70	2	5.16	51.14	70	2	5.16	51.60	40
11	5	0.83	5.78	150	1	0.73	14.63	50	3	5.55	27.75	110
12	5	0.83	5.79	150	5	0.83	5.78	150	4	5.44	27.21	60
13	5	0.11	0.77	150	0	0.40	1.98	70	5	0.11	0.77	150
14	1	4.27	42.66	40	5	4.82	33.75	150	6	25.00	149.99	70
15	5	1.27	8.87	150	3	8.5	42.51	110	7	3.89	42.82	80
16	5	0.91	6.36	150	0	3.26	16.30	70	0	3.27	16.37	70
17	5	0.30	2.12	150	4	2.06	10.32	60	1	0.27	5.37	50
18	3	10.17	50.84	110	4	9.99	49.94	60	2	2.58	25.82	40
19	2	4.27	42.66	50	1	2.19	43.86	50	3	16.66	83.32	110
总计		11.06	507.12	2 250		12.15	656.88	1 570		35.24	807.13	1 530

大小的属性；实验还设置了4个CSP，每个CSP有10个主机，主机的参数为MIPS、Pes、RAM、带宽和存储大小，同时定义了处理器和内存的使用价格分别为3.0元和0.05元，以及虚拟主机的最大/最小负载阈值为80%/0，VM迁移的实验参数见表8。该实验通过虚拟机的VMMNDSA动态调度算法模拟云计算中心的负载平衡实验，直到所有任务完成，并给出了迁移次数、平均SLA违反率以及总体能耗作为输出。

文献[37]同样讨论了虚拟机迁移的能耗和SLA，并讨论了LRMMT和IQRMC两种虚拟机迁移的算法。为了验证VMMNDSA的性能，实验对比了上述两种迁移策略的3个指标作为性能

参考，即SLA、VM迁移时间和能耗，结果分别如图8、图9、图10所示。

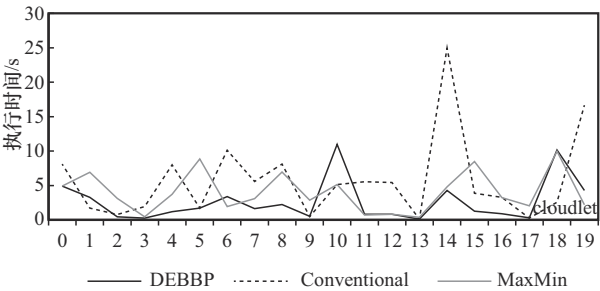
VMMNDSA旨在避免不必要的虚拟机迁移，因此在合作和竞争联盟中，它在虚拟机迁移时间方面具有突出优势，从图8可以看出，它比LRMMT低84.05%和80.89%，比IQRMC低86.20%和80.02%。

同时，从图9可以看出，VMMNDSA是最高效的一种，因为它在合作和竞争联盟中的能耗最低，与LRMMT相比，它的能耗降低了22.99%和32.18%，与IQRMC相比，降低了23.72%和28.52%。

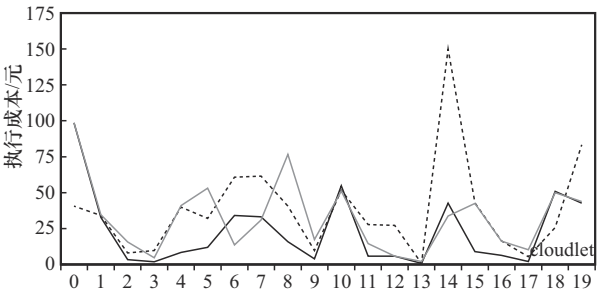
在提出的模型中，SLA与热节点的数量有关，这是因为VMMNDSA侧重于避免不必要的

表 7 Job B 模拟结果

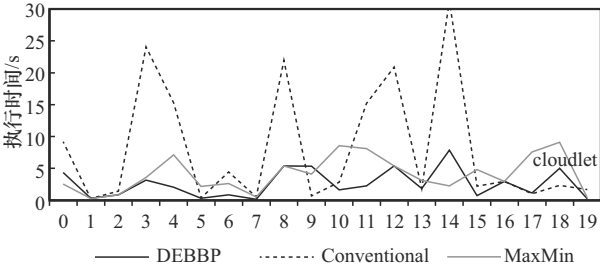
Cloudlet ID	DEBBP				Max Min				Conventional			
	VM ID	执行时间/s	执行成本/元	RAS	VM ID	执行时间/s	执行成本/元	QoS	VM ID	执行时间/s	执行成本/元	RAS
0	2	4.37	43.65	40	5	2.56	17.92	150	0	9.20	45.99	70
1	5	0.30	2.12	150	1	0.27	5.37	50	1	0.27	5.37	50
2	5	0.85	5.98	150	5	0.85	5.98	150	2	1.45	14.53	40
3	1	3.17	63.39	50	5	3.58	25.07	150	3	24.07	120.35	110
4	1	2.05	40.84	50	7	7.11	78.25	80	4	15.27	76.36	60
5	5	0.33	2.13	150	4	2.18	10.89	60	5	0.38	2.65	150
6	5	0.85	5.98	150	7	2.62	28.81	80	6	4.43	26.56	70
7	5	0.17	1.16	150	7	0.51	5.59	80	7	0.51	5.56	80
8	1	5.41	108.28	50	1	5.41	108.26	50	0	21.97	109.85	70
9	3	5.36	26.82	110	6	4.13	24.77	70	1	0.71	14.11	50
10	5	1.65	11.56	150	6	8.57	51.39	70	2	2.82	28.17	40
11	5	2.26	15.81	150	0	8.12	40.58	70	3	15.18	75.9	110
12	2	5.40	53.95	40	2	5.40	53.96	40	4	20.90	104.52	60
13	5	1.84	12.90	150	2	3.14	31.43	40	5	1.84	12.91	150
14	7	7.87	86.62	80	1	2.27	45.34	50	6	31.27	14.85	70
15	5	0.72	5.01	150	3	4.81	24.07	110	7	2.20	24.20	80
16	0	2.96	14.81	70	0	2.97	14.83	70	0	2.97	104.52	70
17	5	1.13	7.90	150	3	7.58	37.90	110	1	1.00	19.97	50
18	0	5.02	25.11	70	4	9.09	45.44	60	2	2.34	23.45	40
19	5	0.25	1.74	150	2	0.42	4.23	40	3	1.67	8.33	110
总计		10.73	536.05	2 210		12.79	660.07	1 580		41.02	813.26	1 530



(a) Job A

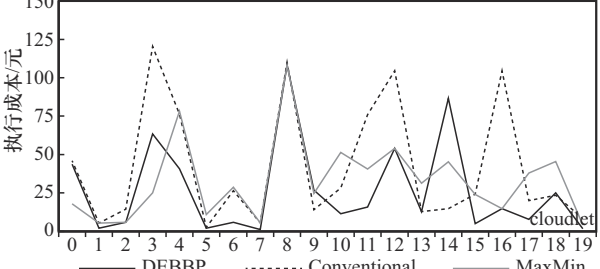


(a) Job A



(b) Job B

图5 Job A和Job B的执行时间结果



(b) Job B

图6 Job A和Job B的执行成本实验结果

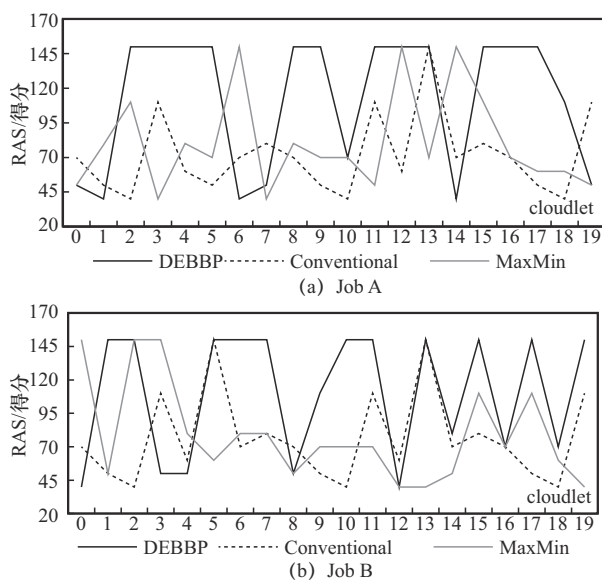


图7 Job A和Job B的RAS实验结果

表8 VM迁移的实验参数

类型设置		数量	MIPS	Pes	RAM	存储大小
VM 类型	类型 1	20	500	1	870	2 500
	类型 2	20	1 000	1	1 740	
	类型 3	20	2 000	1	1 740	
	类型 4	20	2 500	1	613	
CSP 1	10 台类型 1 主机	25	1 860	2	4 096	1 000 000
CSP 2	10 台类型 2 主机	25	2 660	2	4 096	
CSP 3	10 台类型 3 主机	25	2 980	2	4 096	
CSP 4	10 台类型 4 主机	25	3 220	2	4 096	

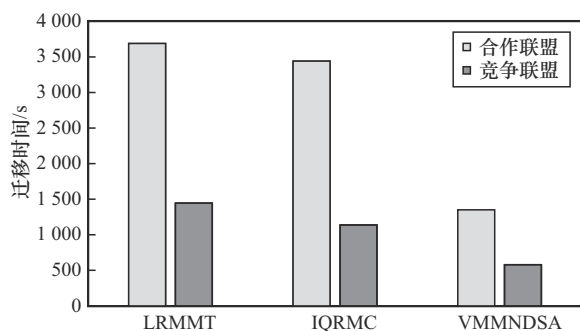


图8 虚拟机迁移时间对比

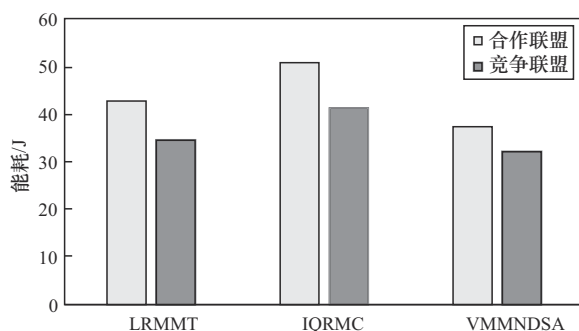


图9 虚拟机迁移能耗对比

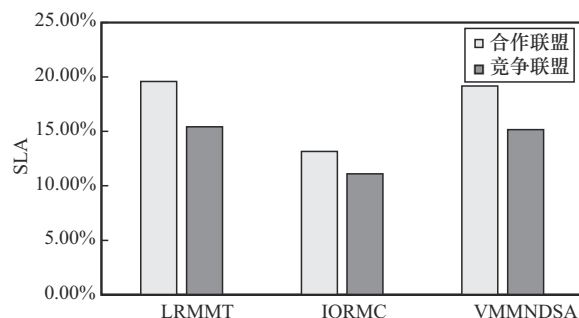


图10 虚拟机迁移SLA对比

VM迁移以节省能源；与LRMMT和IQRMC相比，它的SLA违反率相对较高，从图10可以看出，对于合作和竞争联盟，分别高出14.60%、33.41%和19.16%、29.31%。

合作联盟和竞争联盟（显示表现更优的联盟数据及其百分比）的比较结果见表9。对于迁移时间，竞争联盟更优，因为它对目标节点的选择更少。然而，在能耗方面，合作云联盟发挥得更好，虚拟机有更多的选择选项，但迁移范围更广。相比之下SLA和能量的性能是反相关的，因此竞争云联盟会产生更好的SLA。

基于上述分析，可以得出结论，本文提出的VMMNDSA侧重于减少虚拟机迁移的次数，从而

表9 合作联盟与竞争联盟的比较

比较项	合作联盟			竞争联盟		
	MadMmt	IqrMc	VMMNDSA	MadMmt	IqrMc	VMMNDSA
迁移时间（低）				46.52%	55.73%	35.91%
能量消耗（低）	13.28%	18.50%	14.11%			
SLA（低）				7.72%	13.98%	12.68%

减少迁移对云计算中心整体性能的影响, 实现更节能的目标, 同时将SLA保持在可接受的水平。

5 结束语

在硬件设施有限的前提下, 云计算的服务质量水平在很大程度上取决于所采用的资源分配策略。为了应对这一挑战, 本文基于云计算环境中的资源结构, 从3个层对云资源进行优化, 并提出了一种适合中小型CSP的云联盟建模, 该模型分别完成了任务分配和VM迁移的优化, 从而平衡了云联盟的总体利益和QoS质量。

未来, 这项工作可以进一步扩展, 特别是考虑云的外部因素, 如网络, 这是因为网络可能会严重影响对云节点的访问以及由云提供或用户感知的QoS。此外, 可以考虑在其他平台(如Open-Stack)上进行实验, 进一步对数据仿真进行研究。

参考文献:

- [1] ABIDA A, MANZOOR M F, FAROOQ M S, et al. Challenges and issues of resource allocation techniques in cloud computing[J]. KSII Transactions on Internet and Information Systems (TIIS), 2020, 14(7): 2815-2839.
- [2] GOEL G, TANWAR P, BANSAL V, et al. The challenges and issues with virtualization in cloud computing[C]//Proceedings of the 2021 5th International Conference on Trends in Electronics and Informatics (ICOEI). Piscataway: IEEE Press, 2021: 1334-1338.
- [3] RIMAL B P, CHOI E, LUMB I. A taxonomy and survey of cloud computing systems[C]//Proceedings of the 2009 Fifth International Joint Conference on INC, IMS and IDC. Piscataway: IEEE Press, 2009: 44-51.
- [4] MELL P, GRANCE T. NIST special publication 800-145: the NIST definition of cloud computing[S]. Gaithersburg: National Institute of Standards and Technology, 2011.
- [5] MASANET E, SHEHABI A, LEI N A, et al. Recalibrating global data center energy-use estimates[J]. Science, 2020, 367(6481): 984-986.
- [6] BELOGLAZOV A, BUYYA R. Optimal online deterministic algorithms and adaptive heuristics for energy and performance efficient dynamic consolidation of virtual machines in Cloud Data Centers[J]. Concurrency and Computation: Practice and Experience, 2012, 24(13): 1397-1420.
- [7] FARID M, LATIP R, HUSSIN M, et al. A survey on QoS requirements based on particle swarm optimization scheduling techniques for workflow scheduling in cloud computing[J]. Symmetry, 2020, 12(4): 551.
- [8] QI L Y, CHEN Y, YUAN Y, et al. A QoS-aware virtual machine scheduling method for energy conservation in cloud-based cyber-physical systems[J]. World Wide Web, 2020, 23(2): 1275-1297.
- [9] LIANG H G, WEN X Q, LIU Y K, et al. Logistics-involved QoS-aware service composition in cloud manufacturing with deep reinforcement learning[J]. Robotics and Computer-Integrated Manufacturing, 2021(67): 101991.
- [10] HEIDARI A, JAFARI NAVIMIPOUR N. A new SLA-aware method for discovering the cloud services using an improved nature-inspired optimization algorithm[J]. PeerJ Computer Science, 2021(7): e539.
- [11] TRICOMI G, MERLINO G, PANARELLO A, et al. Optimal selection techniques for cloud service providers[J]. IEEE Access, 2020(8): 203591-203618.
- [12] PAL R, YIN X L, GOLUBCHIK L. Graphical federated cloud sharing markets[J]. IEEE Transactions on Sustainable Computing, 2021, 6(4): 710-712.
- [13] BAGULA M F, BAGULA H, MANDAVA M, et al. Cyber-healthcare kiosks for healthcare support in developing countries[C]//MENDY G, OUYA S, DIOUM I, et al. International Conference on e-Infrastructure and e-Services for Developing Countries. Cham: Springer, 2019: 185-198.
- [14] BAGULA A, MANDAVA M, BAGULA H. A framework for healthcare support in the rural and low income areas of the developing world[J]. Journal of Network and Computer Applications, 2018(120): 17-29.
- [15] LUBAMBA C, BAGULA A. Cyber-healthcare cloud computing interoperability using the HL7-CDA standard[C]//Proceedings of the 2017 IEEE Symposium on Computers and Communications (ISCC). Piscataway: IEEE Press, 2017: 105-110.
- [16] CELESTI A, FAZIO M, GALÁN MÁRQUEZ F, et al. How to develop IoT cloud e-health systems based on FIWARE: a lesson learnt[J]. Journal of Sensor and Actuator Networks, 2019, 8(1): 7.
- [17] MURAD S S, BADEEL R, SALIH N, et al. Optimized Min-Min task scheduling algorithm for scientific workflows in a cloud environment[J]. J. Theor. Appl. Inf. Technol, 2022(100): 480-506.



- [18] SHAKIL K A, ALAM M, KHAN S. A latency-aware Max-Min algorithm for resource allocation in cloud[J]. International Journal of Electrical and Computer Engineering (IJECE), 2021, 11(1): 671.
- [19] GAO Y, RONG H, HUANG J. Adaptive grid job scheduling with genetic algorithms[J]. Future Gener Comput Syst, 2005 (21): 151-161.
- [20] NOORIENTALOUKI R, HOSSEINI SHIRVANI M, MOTAMENI H. A heuristic-based task scheduling algorithm for scientific workflows in heterogeneous cloud computing platforms[J]. Journal of King Saud University - Computer and Information Sciences, 2022, 34(8): 4902-4913.
- [21] LI T S, ZHANG X X. Differentiated service-based evolutionary gamescheduling algorithm for cloud computing[J]. Journal of Beijing University of Posts and Telecommunications, 2013, 36(1): 41-45.
- [22] TANG X Y, SHI C, DENG T, et al. Parallel random matrix particle swarm optimization scheduling algorithms with budget constraints on cloud computing systems[J]. Applied Soft Computing, 2021(113): 107914.
- [23] BASKARAN N, ESWARI R. Efficient VM selection strategies in cloud datacenter using fuzzy soft set[J]. Journal of Organizational and End User Computing, 2021, 33(5): 153-179.
- [24] GHAREHPASHA S, MASDARI M. A discrete chaotic multi-objective SCA-ALO optimization algorithm for an optimal virtual machine placement in cloud data center[J]. Journal of Ambient Intelligence and Humanized Computing, 2021, 12(10): 9323-9339.
- [25] PADMAVATHY T, ANITHA R. An efficient virtual machine allocation using single stage weapon target assignment model in cloud software-defined network environment[J]. International Journal of Communication Systems, 2022, 35(6): e5086.
- [26] VERMA G. Secure VM migration in cloud: multi-criteria perspective with improved optimization model[J]. Wireless Personal Communications, 2022, 124(1): 75-102.
- [27] DAHAN F. An effective multi-agent ant colony optimization algorithm for QoS-aware cloud service composition[J]. IEEE Access, 2021(9): 17196-17207.
- [28] YU Q H, WAN H, ZHAO X B, et al. Online scheduling for dynamic VM migration in multicast time-sensitive networks[J]. IEEE Transactions on Industrial Informatics, 2020, 16(6): 3778-3788.
- [29] PATEL D, GUPTA R K, PATERIYA R K. Energy-aware prediction-based load balancing approach with VM migration for the cloud environment[M]. Singapore: Springer, 2019: 59-74.
- [30] VIJIN R, BANUMATHI J. Federation of selfish cloud providers by innovative economic and secure sharing model[J]. Journal of Network Communications and Emerging Technologies (JNCET) 2015, 2(1).
- [31] AHMED U, AL-SAIDI A, PETRI I, et al. QoS-aware trust establishment for cloud federation[J]. Concurrency and Computation: Practice and Experience, 2022, 34(3): e6598.
- [32] KHAN M S A, SANTHOSH R. Hybrid optimization algorithm for VM migration in cloud computing[J]. Computers and Electrical Engineering, 2022(102): 108152.
- [33] SUN J Y, ZHANG Q F, TSANG E P K. DE/EDA: a new evolutionary algorithm for global optimization[J]. Information Sciences, 2005, 169(3/4): 249-262.
- [34] KAUR P, RANI A. Virtual machine migration in cloud computing[J]. International Journal of Grid and Distributed Computing, 2015, 8(5): 337-342.
- [35] CALHEIROS R N, RANJAN R, BELOGLAZOV A, et al. CloudSim: a toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms[J]. Software: Practice and Experience, 2011, 41(1): 23-50.
- [36] FOSTER I, KESSELMAN C. The Grid 2: blueprint for a new computing infrastructure[M]. Amsterdam: Elsevier Press, 2003.
- [37] NASAB S S F, MARJANEH T Z, BAHREPOUR D. Energy efficiency and establishing service level agreement using fuzzification of virtual machine selection policies for migrating in cloud computing[C]//Proceedings of the 2023 9th International Conference on Web Research (ICWR). Piscataway: IEEE Press, 2023: 201-207.

[作者简介]



马焜 (1988-), 男, 博士, 南开大学在站博士后, 主要研究方向为云计算、边缘计算、人工智能等。



胡创悦 (1990-), 女, 嘉兴市长三角区块链技术研究院工程师, 主要研究方向为人工智能、数字化建筑工程。



张玉志（1964-），男，南开大学讲席教授、博士生导师，主要研究方向为人工智能和软件工程。



党中华（1966-），男，嘉兴职业技术学院副教授，主要研究方向为软件工程。



马方超（1990-），男，浙江省公安厅警务技术二级主管，主要研究方向为网络和数据安全、网络安全等级保护。



肖文红（1971-），男，嘉兴职业技术学院副教授，主要研究方向为计算机网络技术。



王晓冬（1981-），男，浙江省教育技术中心信息技术部主任，主要研究方向为数字化应用建设、网络安全、数据安全。



陈双喜（1980-），男，博士，嘉兴职业技术学院教授，浙江省网络空间安全协会人才培养专委会秘书长，主要研究方向为网络空间安全拟态防御。