



控制与决策

Control and Decision

ISSN 1001-0920, CN 21-1124/TP

## 《控制与决策》网络首发论文

题目: 异构云环境下 AHP 定权的多目标强化学习作业调度方法  
作者: 袁景凌, 陈旻骋, 江涛, 李超  
DOI: 10.13195/j.kzyjc.2020.0911  
收稿日期: 2020-07-06  
网络首发日期: 2021-01-05  
引用格式: 袁景凌, 陈旻骋, 江涛, 李超. 异构云环境下 AHP 定权的多目标强化学习作业调度方法[J/OL]. 控制与决策. <https://doi.org/10.13195/j.kzyjc.2020.0911>



**网络首发:** 在编辑部工作流程中, 稿件从录用到出版要经历录用定稿、排版定稿、整期汇编定稿等阶段。录用定稿指内容已经确定, 且通过同行评议、主编终审同意刊用的稿件。排版定稿指录用定稿按照期刊特定版式(包括网络呈现版式)排版后的稿件, 可暂不确定出版年、卷、期和页码。整期汇编定稿指出版年、卷、期、页码均已确定的印刷或数字出版的整期汇编稿件。录用定稿网络首发稿件内容必须符合《出版管理条例》和《期刊出版管理规定》的有关规定; 学术研究成果具有创新性、科学性和先进性, 符合编辑部对刊文的录用要求, 不存在学术不端行为及其他侵权行为; 稿件内容应基本符合国家有关书刊编辑、出版的技术标准, 正确使用和统一规范语言文字、符号、数字、外文字母、法定计量单位及地图标注等。为确保录用定稿网络首发的严肃性, 录用定稿一经发布, 不得修改论文题目、作者、机构名称和学术内容, 只可基于编辑规范进行少量文字的修改。

**出版确认:** 纸质期刊编辑部通过与《中国学术期刊(光盘版)》电子杂志社有限公司签约, 在《中国学术期刊(网络版)》出版传播平台上创办与纸质期刊内容一致的网络版, 以单篇或整期出版形式, 在印刷出版之前刊发论文的录用定稿、排版定稿、整期汇编定稿。因为《中国学术期刊(网络版)》是国家新闻出版广电总局批准的网络连续型出版物(ISSN 2096-4188, CN 11-6037/Z), 所以签约期刊的网络版上网络首发论文视为正式出版。

# 异构云环境下 AHP 定权的多目标强化学习作业调度方法

袁景凌<sup>1†</sup>, 陈旻骋<sup>1</sup>, 江涛<sup>1</sup>, 李超<sup>2</sup>

(1. 武汉理工大学 计算机科学与技术学院, 武汉 430070; 2. 上海交通大学 电子信息与电气工程学院, 上海 200240)

**摘要：**随着新型基础设施建设(新基建)的加速, 云计算将获得新的发展契机. 数据中心, 作为云计算的基础设施, 其内部服务器不断升级换代, 这造成计算资源的异构化. 如何在异构云环境下, 对作业进行高效调度是当前的研究热点之一. 本文针对异构云环境多目标优化调度问题, 设计了一种 AHP 定权的多目标强化学习作业调度方法. 首先我们定义了执行时间、平台运行能耗、成本等多个目标. 其中定义服务延迟成本用来描述用户对服务质量的满意程度. 然后设计了面向异构资源的多目标调度综合评价方法, 该方法利用层次分析法 (Analytic Hierarchy Process, AHP) 确定了各个目标的权重. 最后将该方法引入 Q-learning 的奖励值计算, 使其能反映异构云环境下作业的总体执行情况, 并对后续抵达的作业起到良好的经验学习作用. 实验结果表明本文提出的方法优于大部分对比方法, 能较好地优化作业执行效率和保障用户及服务提供商的利益.

**关键词：**强化学习; 多目标; 作业调度; 异构资源; 服务延迟成本; 层析分析法

中图分类号: TP311

文献标志码: A

DOI: 10.13195/j.kzyjc.2020.0911

## Multi-Objective Reinforcement Learning Job Scheduling Method using AHP Fixed Weight in Heterogeneous Cloud Environment

YUAN Jing-ling<sup>1†</sup>, CHEN Min-cheng<sup>1</sup>, JIANG Tao<sup>1</sup>, LI Chao<sup>2</sup>

(1. School of Computer Science and Technology, Wuhan University of Technology, Wuhan 430070, China; 2. School of Electronic Information and Electrical Engineering, Shanghai Jiao Tong University, Shanghai 200240, China)

**Abstract:** With the acceleration of new infrastructure, cloud computing will be given an entirely new opportunity to develop. Data center, as the infrastructure of cloud computing, its internal servers are continuously updated, which leads to the heterogeneity of computing resources. How to efficiently schedule jobs in heterogeneous cloud environment has become an increasingly popular research. In this paper, we design a multi-objective reinforcement learning job scheduling method using AHP fixed weight in heterogeneous cloud environment. First, we define the execution time, energy consumption, execution cost, etc. Service delay cost is used to describe the user satisfaction for service. Then, a comprehensive evaluation method for multi-objective scheduling is designed. The weight coefficient of each object is determined by the analytic hierarchy process (AHP). The method is introduced into the calculation of reward, makes it reflective of the overall situation and serves as an excellent learning tool for the following jobs. The experimental results show that the proposed method can better optimize the execution efficiency, while ensuring the interests of users and service providers.

**Keywords:** reinforcement learning; multi-objective; job scheduling; heterogeneous resources; service delay cost; analytic hierarchy process

## 0 引言

随着新基建建设的提速, 数据中心建设步伐明显加快. 普通数据中心一般由 5-10 万台服务器组成, 但国内外著名互联网公司的数据中心规模远不只如此, 谷歌全球数据中心的服务器总量已超百万, 为用户提供搜索服务; 亚马逊公司仅弹性计算云 (Elastic

Compute Cloud, EC2) 就部署了约 50 万台服务器; 国内腾讯公司的天津数据中心在 2018 年服务器总数突破 10 万台; 国内阿里巴巴将在南通建立云计算数据中心, 计划容纳 30 万台服务器. 数据中心规模越来越大, 造成了计算资源的异构化和规模化. 这给大规模数据中心的作业调度带来了一定的挑战. 云计算数

收稿日期: 2020-07-06; 修回日期: 2020-11-23.

基金项目: 国家自然科学基金项目 (61303029); 湖北省创新团队项目 (2015CFA069); 湖北省技术创新专项重大项目 (2017AAA122).

<sup>†</sup>通讯作者. E-mail: yuanjingling@126.com.

据中心作为商业计算服务平台,首先需要满足各种不同的用户需求,例如,有些用户需要快速的服务响应时间,有些用户需要服务的稳定性和可靠性,有些更倾向于低廉的计算成本,与此同时还要保障用户的服务体验<sup>[1]</sup>.再者,云计算的商业性决定了调度策略还需保障服务供应商的利益,减少能耗及相应开销.传统的调度方法已不能适应其复杂多变的调度环境.因此,研究云计算异构环境下的多目标优化调度策略具有相当重要的意义.

针对异构云环境多目标优化调度问题,本文主要工作如下:

(1) 定义了执行时间、服务延迟成本、平台运行能耗、成本等多个目标.其中服务延迟成本用来描述用户对服务质量的满意程度.

(2) 设计了 AHP 定权的多目标调度综合评价方法,并将该方法引入强化学习 Q-learning 的奖励值计算,这有利于后续作业的经验学习.

(3) 利用 MultiRECloudSim 平台进行了仿真实验,实验结果表明本文所提方法优于大部分对比方法,能较好地改善作业执行效率与保证用户及服务提供商的利益.

## 1 相关工作

云数据中心作业调度主要分为三类:传统作业调度、多目标优化作业调度及基于机器学习的作业调度.

**传统作业调度方法**一般优先考虑性能,被广泛应用的算法有先来先服务调度算法、Max-Min 调度算法、Min-Min 调度算法、公平调度算法以及单目标启发式调度算法等,这类算法一般不考虑资源需求及资源利用率,只为查找满足计算需求的节点来执行作业.但是实际的作业调度问题往往有多个目标,例如最短执行时间、最佳服务质量、最少计算成本等,因此这类方法实用性较为欠缺.

**多目标优化作业调度方法**考虑了云环境下的多个优化目标,并将作业映射至合适的云计算资源,其相较于传统的作业调度方法更具实用性<sup>[2-3]</sup>.近年来,随着云服务快速普及,多目标优化作业调度方法受到了广泛的关注,并已有了一定成果.Akbari 等人<sup>[4]</sup>设计了基于布谷鸟搜索的多目标作业调度算法,该算法不但减少了异构系统中的作业执行时间,而且最大程度的保障作业并行化.胡成玉等人<sup>[5]</sup>建立了以利润、碳排放以及 QoS 为目标的数据中心负载优化调度模型,并设计了一种非支配排序遗传算法,保证算法能快速收敛.Zhu 等人<sup>[6]</sup>以作业完成时间与成

本作为调度目标,设计了基于演化算法的多目标调度算法来优化 IaaS(Infrastructure as a Service) 平台的工作流调度.苏命峰等人<sup>[7]</sup>结合能耗、成本和系统可用性等指标构建了多维 QoS 云资源调度模型,并提出结合多重贪心算法的云资源调度算法.上述方法取得了比传统作业调度方法更好的解,但也存在一些问题.现有的调度策略的研究较少兼顾用户和服务供应商双方利益,如何使得资源供给与消费双方互利共赢,这是当前云作业调度需考虑的问题.

**基于机器学习的作业调度方法**学习数据中心资源消耗、性能变化和负载波动之间的关系,探索高效执行作业时所需资源.在实际执行调度算法时,根据学习结果与实时监控状况来调整作业分配方案<sup>[8]</sup>.Islam 等人<sup>[9]</sup>利用神经网络和线性回归设计了资源预测的和供应策略,实现了云作业资源的按需分配.Chen 等人<sup>[10]</sup>设计了基于动态加权随机森林的负载预测方法,并根据预测结果进行自适应作业调度.上述方法通过预测负载波动调节计算资源规模,为减少处于活跃状态的服务器,需频繁的进行虚拟机迁移,这对云作业调度造成在一定程度上的影响.

除此之外,基于强化学习的作业调度方法也是研究的热点.Yi 等人<sup>[11]</sup>利用 LSTM 网络构建系统状态预测模型,然后使用该模型来训练基于深度强化学习的作业分配器,有效的为数据中心节省近 10% 的计算能力.Yang 等人<sup>[12]</sup>提出了基于强化学习的资源分配方法,并通过优化长期奖励以改善工作流调度效率.

本文介绍的方法充分考虑了节能增效、用户体验、执行成本等多个目标之间的平衡,兼顾用户和服务供应商双方利益,着眼云计算资源的异构性及作业多样化,利用强化算法对云数据中心作业进行智能自主管理.

## 2 多目标与约束定义

**定义 1** 异构资源约束:云数据中心由众多的异构服务器构成,这些服务器被虚拟化成虚拟机,虚拟机  $V_i$  有  $d$  个维度资源,可表示为  $V_i = \{V_i^1, \dots, V_i^d\}$ .同理定义作业  $T_j$  所需的资源为  $T_j = \{T_j^1, \dots, T_j^k\}$ .作业和虚拟机之间的放置关系可用调度矩阵  $X = [x_{ij}]$  表示,当  $T_j$  作业分配到虚拟机  $V_i$  时,  $x_{ij} = 1$ , 否则为  $x_{ij} = 0$ .通过上述对异构资源的分配关系,我们可以获得异构资源环境下作业调度的资源约束目标为:

$$\sum_{j=1}^m T_j^k \cdot x_{ij} \leq V_i^k \quad (1)$$

公式 (1) 反映了表示作业执行所消耗的全部资源量应小于等于平台的总资源量。

**定义 2** 执行时间: 执行时间是作业调度中最为普遍的优化指标, 作业  $T_j$  的执行时间  $t_e(T_j, V_i)$  可计算为:

$$t_e(T_j, V_i) = \frac{\text{len}(T_j)}{V_i^{\text{CPU}}} \quad (2)$$

其中,  $\text{len}(T_j)$  表示作业  $T_j$  的长度;  $V_i^{\text{CPU}}$  表示虚拟机  $V_i$  中的 CPU 执行能力. 文中我们假定每个任务只能在一台虚拟机上执行, 云平台的总执行时间  $t_{all}$  为:

$$t_{all} = \max\left\{\sum_{j=1}^{m_1} t_e(T_j, V_1), \dots, \sum_{j=1}^{m_n} t_e(T_j, V_n)\right\} \quad (3)$$

其中,  $m_i$  表示全部被虚拟机  $V_i$  成功执行的作业数目,  $\sum_{i=1}^n m_i = m$ . 对于云平台来讲, 总执行时间越少, 用户的满意程度越高.

**定义 3** 服务延迟成本: 对于云计算平台管理者而言, 更关注平台的整体性能如资源负载均衡、能耗等. 对于提交作业的用户来说, 更关心的是服务执行效果. 用户访问数据中心, 不仅需要考虑到该服务的价格, 同时也要考虑服务质量. 本文用服务延迟成本来描述用户对服务质量的满意程度. 作业等待响应的时间越长, 延迟成本就越大, 服务体验就相对较差. 服务延迟成本可表示为作业等待响应时间的函数. 作业  $T_j$  的等待响应时间  $t_w(T_j)$  可计算为:

$$t_w(T_j) = t_{est}(T_j) - t_{ar}(T_j) + t_s(t_w) \quad (4)$$

其中,  $t_{est}(T_j)$  为作业开始执行的时间;  $t_{ar}(T_j)$  为作业到达时间;  $t_s(t_w)$  为调度算法  $t_w$  执行时长. 作业  $T_j$  的服务延迟成本  $cost_d(T_j)$  可计算为:

$$cost_d(T_j) = \begin{cases} 0 & t_w(T_j) < \psi \cdot t_{rt} \\ L(t_w(T_j)) & t_w(T_j) \geq \psi \cdot t_{rt} \end{cases} \quad (5)$$

$$L(t_w(T_j)) = cost_{d/u} \cdot (t_w(T_j) - \psi \cdot t_{rt}) \quad (6)$$

其中,  $t_{rt}$  为用户的可容忍等待时间;  $\psi$  为自适应调节因子, 因为用户可容忍等待时间在不同时间段是不同的;  $cost_{d/u}$  为单位时间内服务延迟成本.  $cost_d(T_j)$  越小, 则说明用户体验越好. 总服务延迟成本  $cost_d^{all}$  可计算为:

$$cost_d^{all} = \sum_{j=1}^m cost_d(T_j) \quad (7)$$

**定义 4** 平台运行能耗: 我们利用 CPU 利用率-功率函数<sup>[13]</sup>来计算数据中心运行能耗  $energy_{all}$ . 服务供应商不仅需考虑服务器能耗, 而且还要考虑制冷、照明等能耗. 因此数据中心能耗利用能源效率指标 (Power Usage Effectiveness, PUE) 来估算<sup>[14]</sup>, 具体计算公式为:

$$energy_{all} = t_{all} \cdot \sum_{i=1}^p P_i(U_i) \quad (8)$$

其中,  $p$  为数据中心内活跃的服务器数量;  $P_i()$  为第  $i$  台服务器的 CPU 利用率-功率函数;  $U_i$  为第  $i$  台服务器的 CPU 利用率, 其计算公式为:

$$U_i = \frac{1}{c} \cdot \sum_{k=1}^c VM_k^{\text{CPU}} \quad (9)$$

其中,  $c$  为部署在第  $i$  台服务器上的虚拟数量;  $VM_k^{\text{CPU}}$  为该虚拟机 CPU 利用率.

**定义 5** 成本: 总成本  $cost_{all}$  包括用户成本  $cost_u$  和服务供应商成本  $cost_p$ . 用户成本主要为作业执行成本, 服务供应商成本主要为能耗成本. 因此总成本的具体计算公式为:

$$cost_{all} = cost_u + cost_p = \sum_{j=1}^m cost_v(V_i) \cdot t_e(T_j, V_i) + energy_{all} \cdot cost_{ep} \quad (10)$$

其中,  $cost_v(V_i)$  为虚拟机  $V_i$  的单位计算成本;  $cost_{ep}$  表示单位电力成本.

至此定义了作业的调度过程中的多个目标, 其调度目标如下:

$$\min \begin{cases} t_{all} \\ energy_{all} \\ cost_{all} \\ cost_d^{all} \end{cases} \quad (11)$$

其需要满足的约束条件如下:

$$\sum_{j=1}^m T_j^k \cdot x_{ij} \leq V_i^k \quad (12)$$

### 3 AHP 定权的多目标强化学习作业调度

#### 3.1 马尔科夫决策过程

根据排队模型, 动态到达的作业调度过程可视为马尔科夫决策过程, 因此可构建马尔科夫决策过程的四元组  $E = \langle X, A, P, R \rangle$ . 作业调度的状态空间为  $X = \{\text{schedule}(T_j), \text{queue}(V_i), \text{excute}\}$ , 其有三类状态: 调度状态  $\text{schedule}(T_j)$ 、在虚拟机前队列等待  $\text{queue}(V_i)$  及执行状态;  $A$  为各个作业  $T_j$  的动作.  $P(x, x', a)$  表示作业执行动作  $a$  后到达状态  $x'$  的概率;  $R$  则为奖励函数, 具体为:

$$\begin{aligned} R1: A \times X = \text{schedule}(T_j) &\rightarrow \text{queue}(V_i) \\ R2: A \times X = \text{queue}(V_i) &\rightarrow \text{excute} \end{aligned} \quad (13)$$

当作业从调度状态转换为等待状态, 则获得奖励  $R1$ . 当作业变成执行状态时, 则获得奖励  $R2$ .

我们可发现状态空间  $X$  的规模为作业数量  $m$  和虚拟机数量  $n$  的乘积. 由于奖励  $R2$  需要很长的反馈过程, 且计算较为复杂. 本文研究重点在调度过程, 因此在后续研究中, 奖励  $R$  即因调度动作而产生的奖励  $R1$ . 奖励  $R$  是根据虚拟机前的作业等待队列长



度来确定的,即根据作业被分配至虚拟机  $V_i$  上的数量,可计算为:

$$R_{x \rightarrow x'}^a = k \cdot \text{size}(\text{queue}(V_i)) \quad (14)$$

其中,  $R_{x \rightarrow x'}^a$  表示执行动作  $a$  后,状态从  $x$  转变成  $x'$  所获得的奖励;  $k$  为常数,用于调节  $k$  的奖励  $R$  的数值大小. 但仅依据虚拟机前作业等待队列长度确定奖励值  $R$  忽略了作业调度的多个优化目标,为此将面向异构资源的多目标调度综合评价引入奖励  $R$  的计算策略.

### 3.2 AHP 定权的多目标调度综合评价

首先需构建面向异构资源的多目标调度综合评价体系,该体系是简化复杂调度问题的关键. 我们将该评价体系分为 3 层: 最上层为目标层; 中间为对象层, 包括用户与服务供应商; 最下层为指标层, 包括用户所关注的指标 (作业执行时间、成本 (作业执行成本) 和服务延迟成本) 及服务供应商所关注的指标 (成本 (平台能耗成本) 和平台运行能耗). 该评价体系的层次结构详见图 1.

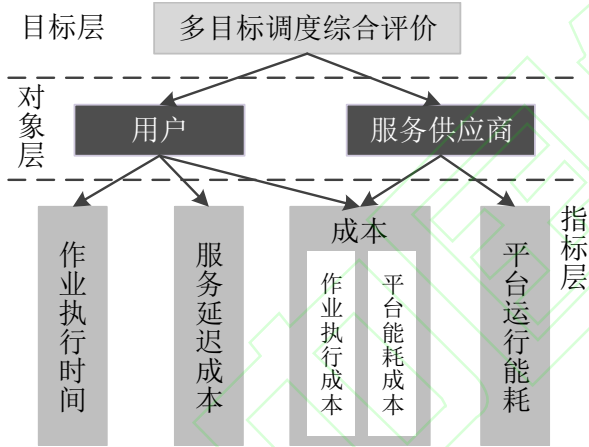


图 1 异构资源的多目标调度综合评价体系结构

在建立多目标调度综合评价体系之后, 还需确定评价体系中各指标的权重. 对此, 本文利用层次分析法设置其初始权重. 具体步骤如下:

#### 1) 建立判断矩阵

假设有  $n$  个指标  $I_1, I_2, \dots, I_n$  对上一层要素有影响, 利用 T.L.Saaty 的 9 分位标度<sup>[15]</sup>, 两两比较同层指标, 获得相对于上层要素的判断矩阵  $A = (a_{ij})_{m \times n}$ , 其中,  $a_{ij} > 0, a_{ij} = 1/a_{ji}, a_{ii} = 1$ .

利用 9 分位标度方法, 将本文所建立的多目标调度综合评价体系结构中的对象层和指标层内的各个指标之间分别进行两两比较, 比较结果如表 1 所示.

根据表 1 得到下列 3 个判断矩阵, 其中矩阵  $A_1$  为对象层的判断矩阵, 矩阵  $A_{21}$ 、矩阵  $A_{22}$  分别为指

标层中针对用户和服务供应商的判断矩阵.

$$A_1 = \begin{bmatrix} 1 & 2 \\ 1/2 & 1 \end{bmatrix}, \quad (15)$$

$$A_{21} = \begin{bmatrix} 1 & 5 & 2 \\ 1/5 & 1 & 1/3 \\ 1/2 & 3 & 1 \end{bmatrix}, \quad (16)$$

$$A_{22} = \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix} \quad (17)$$

表 1 各指标对比结果

指标对比	标度值	选取理由
用户 ( $I_1$ ) → 服务供应商 ( $I_2$ )	$a_{12} = 2$ ( $A_1$ )	当前云计算越来越关注用户的服务体验, 因此在对象层中, 用户稍微重要于服务供应商.
作业执行时间 ( $I_1$ ) → 服务延迟成本 ( $I_2$ )	$a_{12} = 5$ ( $A_{21}$ )	作业执行时间越短, 服务延迟成本一般就相对较低, 因此作业执行时间比服务延迟成本明显重要.
作业执行时间 ( $I_1$ ) → 成本 ( $I_3$ )	$a_{13} = 2$ ( $A_{21}$ )	在尽早完成作业的前提下, 用户希望开销的成本较少, 因此作业执行时间比成本 (作业执行成本) 次稍微重要.
成本 ( $I_3$ ) → 服务延迟成本 ( $I_2$ )	$a_{32} = 3$ ( $A_{21}$ )	成本 (作业执行成本) 越高, 服务供应商收取用户的费用就越多. 若超过用户的心理价位, 用户将会流失, 因此成本比服务延迟成本稍微重要.
成本 ( $I_1$ ) → 平台运行能耗 ( $I_2$ )	$a_{12} = 1$ ( $A_{22}$ )	对于服务供应商来说, 成本 (平台能耗成本) 和平台运行能耗同等重要.

#### 2) 一致性检验

判断矩阵客观地反映了不同指标间的相对重要性, 但并不可以保证每个判断矩阵都是一致的. 例如, 判断指标  $I_1$  比  $I_2$  重要,  $I_2$  比  $I_3$  重要, 那么  $I_3$  比  $I_1$  重要就明显不合理. 我们利用一致性指标  $CI$  衡量判断矩阵是否拥有完全一致性,  $CI$  具体计算公式为:

$$CI = \frac{\lambda_{max} - n}{n - 1} \quad (18)$$

其中,  $n$  为判断矩阵的阶数;  $\lambda_{max}$  为判断矩阵的最大特征值. 若  $CI = 0$ , 则表示判断矩阵拥有完全一致性; 反之, 则需计算随机一致性比率  $CR = CI/RI$ , 其中,  $RI$  为平均随机一致性指标, 其取值见表 2. 当  $CR < 0.1$  时, 判断矩阵的一致性可以接受, 否则判断矩阵需进行调整. 本文判断矩阵  $A_1$ 、 $A_{21}$ 、 $A_{22}$  一致性检测如下:

$A_1$  的  $\lambda_{max} = 2, CI = (\lambda_{max} - n)/(n - 1) = (2 - 2)/(2 - 1) = 0$ , 所以  $A_1$  是完全一致的.

$A_{21}$  的  $\lambda_{max} = 3.0037, CI = (\lambda_{max} - n)/(n - 1) = (3.0037 - 3)/(3 - 1) = 0.00185 \neq 0$ , 因此还需计算  $CR, CR = CI/RI = 0.00185/0.52 = 0.0036 < 0.1$ , 所以  $A_{21}$  的一致性可以被接受。

$A_{22}$  的  $\lambda_{max} = 2, CI = (\lambda_{max} - n)/(n - 1) = (2 - 2)/(2 - 1) = 0$ , 所以  $A_{22}$  是完全一致的。

表 2 平均随机一致性指标

$n$	1	2	3	4	5	6	7	8
$RI$	0	0	0.52	0.89	1.12	1.24	1.36	1.41

### 3) 计算相对权重

计算各指标相对权重, 即求各判断矩阵的特征向量, 也就是计算符合  $A \cdot W = \lambda_{max} \cdot W$  的特征向量, 并将  $W_{\lambda_{max}}$  归一化, 即得到其相对权重. 文中判断矩阵  $A_1$  的相对权重为 (0.67, 0.33),  $A_{21}$  的相对权重为 (0.58, 0.11, 0.31),  $A_{22}$  的相对权重为 (0.5, 0.5).

### 4) 计算综合权重

计算各判断矩阵的相对权重之后, 还需计算确定指标层各指标相对于总目标的权重, 计算方法为:

$$w_i = \sum_{j=1}^m w_{ij}^{(3)} \cdot w_j^{(2)} \quad (19)$$

其中,  $w_{ij}^{(3)}$  表示指标层第  $i$  个指标相对于对象层中第  $j$  个指标的相对权重;  $w_j^{(2)}$  表示对象层第  $j$  个指标相对于总目标的权重. 根据公式 (19) 可得指标层各指标相对于总目标的综合权重为 (0.39, 0.07, 0.375, 0.165). 图 2 为综合权重计算过程.

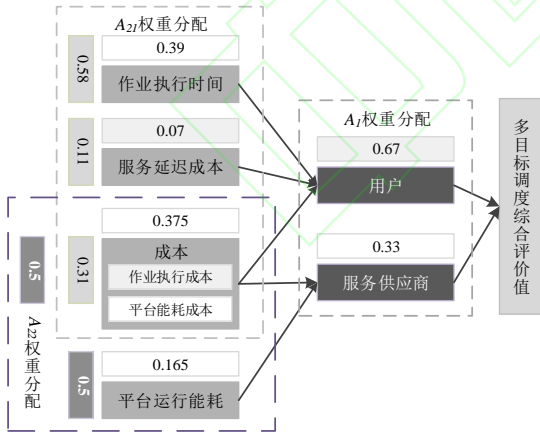


图 2 综合权重计算过程

最后结合综合权重可得异构资源的多目标调度综合评价价值计算公式, 即:

$$T_{value} = 0.39 \cdot t_{all} + 0.07 \cdot cost_d^{all} + 0.375 \cdot cost_{all} + 0.165 \cdot energy_{all} \quad (20)$$

需要注意的是在实际应用中, 各指标需进行归一化处理. 我们将异构资源的多目标调度综合评价

值  $T_{value}$  引入多目标调度奖励值  $R$  的计算策略, 得奖励值计算公式:

$$R_{x \rightarrow x'}^a = k \cdot size(queue(V_i)) \cdot T_{value} \quad (21)$$

利用公式 (21) 计算出来的多目标调度奖励值  $R$  能反映异构云环境下作业的执行时间、成本、能耗等情况, 对后续待分配作业起到良好的经验学习作用。

### 3.3 基于强化学习的调度方案

强化学习 Q-learning 算法的基本思想是根据状态和动作构造一张 Q 值表来保存 Q 值, 然后从 Q 值表找出使得作业调度收益最大的动作. 本文中的 Q-learning 算法在对下一步状态  $x'$  进行探索时, 采用了  $\epsilon$  贪心策略, 即有  $\epsilon$  的概率随机挑选一个动作, 余下  $1 - \epsilon$  的概率挑选效用值 Q 最大的动作.  $\epsilon$  一般取 0.01 ~ 0.1. 具体的 Q 值更新公式如下:

$$Q_{t+1}(x, a) = Q_t(x, a) + \alpha \cdot (R_{x \rightarrow x'}^a + \gamma \cdot Q_t(x', a') - Q_t(x, a)) \quad (22)$$

其中,  $\alpha$  为更新步长,  $\alpha$  越大则说明越靠后的累计奖励越重要.  $\gamma$  为折扣奖赏权重,  $\gamma$  值越大, 则说明之前的经验越重要. AHP 定权的 Q-learning 多目标作业调度的具体步骤详见算法 1.

算法 1 AHP 定权的 Q-learning 多目标作业调度

输入: 第  $w$  批次作业队列  $ListT$ , 虚拟机队列  $ListV$ , Q 值表

输出: 第  $w$  批次的调度方案  $assignedList$

1. **Begin**
2. 构建状态空间  $X$ ;
3. 选择第一个作业  $T_1$  作为开始
4. **For**  $T_j$  **in**  $ListT$  **do**
5.     **If** (random  $< 1 - \epsilon$ ) **then**
6.         在 Q 值表中选择最大 Q 值所对应的动作  $a$ ;
7.     **Else**
8.         为作业  $T_j$  随机选择一个动作  $a$ ;
9.     **End If**
10. 执行动作  $a$ , 并获取下阶段状态  $x'$  和奖励  $R$ ;
11. 根据下阶段状态, 选择最大 Q 值所对应的动作  $a$ ;
12. 根据公式 (22) 更新出当前状态-动作的 Q 值;
13. 把作业与虚拟机的映射关系放入  $assignedList$ ;
14. 令下一个状态  $x'$  变为  $x$ ;
15. **End For**
16. 输出  $assignedList$ ;
17. **End**

算法 1 在初始化 Q 值表过程中, 时间复杂度和状态数量、动作空间密切相关, 该步骤的时间复杂度为  $O(n \cdot m)$ ; 在利用  $\epsilon$  贪心策略进行动作选择时, 时间复杂度和动作空间大小有关, 单次复杂度为  $O(n)$ ; 最后在更新 Q 值表时, 时间复杂度为  $O(n \cdot \log n)$ .

### 3.4 AHP 定权的多目标强化学习作业调度框架

图 3 是 AHP 定权的多目标强化学习作业调度总体框架图,反映了多目标调度的总体运行状况及控制机制,其主要四个部分组成:1) 用户层;2) 计算资源层;3) 监控层;4) 决策层。

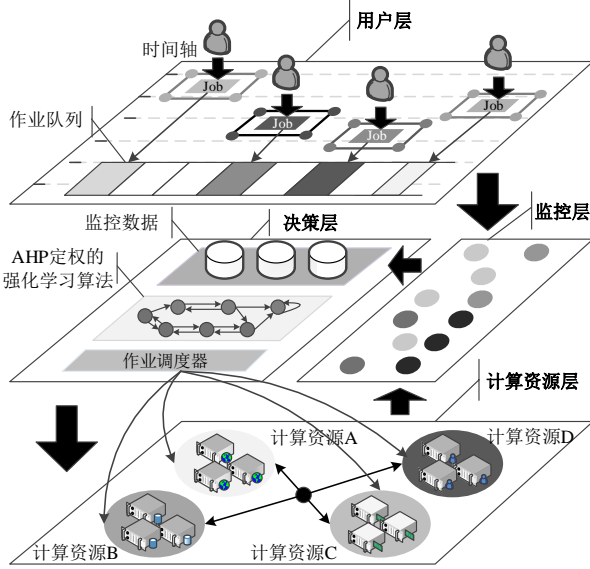


图 3 AHP 定权的多目标强化学习作业调度框架

1) 用户层: 用户在不同时刻提交所需处理的作业。这些作业种类不同,而且对各类资源的需求也各异。然后这些作业进入作业队列等待处理。

2) 计算资源层: 将异构服务器的实体资源抽象成虚拟资源,便于对计算资源进行高效管理,及最大程度的利用数据中心内的可用计算资源。

3) 监控层: 作业的执行使得虚拟机及物理服务器的资源利用率、耗电功率等发生变化。因此该部分不仅需要采集和作业执行相关的参数,而且还需要采集计算资源的变化参数。最后,这部分采集的监控数据将作为反馈信息传送至决策层,以供决策之需。

4) 决策层: 计算资源层和用户层的信息被实时的更新至决策层。而决策层利用 AHP 定权的强化学习算法对异构环境进行实时学习,并确定当前状态下的最优的作业调度方案。最后该方案交给作业调度器去执行。随着作业部署到虚拟机,计算资源层和用户层的变化信息又被监控层传递给决策层。这四个部分通力合作,周而复始,最终实现作业和异构资源的合理分配。

## 4 实验设计与结果分析

### 4.1 实验平台介绍

MultiRECloudSim<sup>[16]</sup> 是华南理工大学林伟伟老师及其团队在 2017 年基于 CloudSim3.0.3 拓展开发的云计算仿真平台。其针对原有的 CloudSim 平台增

加了作业的多种资源需求模拟、预取资源管理模块、多资源运行的能耗计算等功能。

### 4.2 异构资源平台设置

本文在 MultiRECloudSim 构建了 6 款异构服务器,其具体型号及功率映射函数如表 3。其中,功率映射函数根据 SPEC 官网提供的数据计算得到, $x$  为 CPU 利用率, $y$  为耗电功率<sup>[17]</sup>。

表 3 各类型主机的资源详情

No.	服务器型号	功率映射函数
1	Fujitsu TX150 S5	$y = 89.5 + 42.9 \cdot x$
2	Acer GR160 F1	$y = 100.7 + 103.6 \cdot x - 8.7 \cdot x^2$
3	DEPO Race X340H	$y = 81.1 + 73.2 \cdot x - 23.6 \cdot x^2$
4	Acer R520	$y = 152.9 + 115.9 \cdot x$
5	Sun Netra X4250	$y = 225.5 + 99.9 \cdot x - 31.4 \cdot x^2$
6	Hitachi HA8000	$y = 102.2 + 73.4 \cdot x + 159.1 \cdot x^2$

为了便于计算,我们在每台服务器上部署一台虚拟机 (VM),虚拟机的计算能力在 600-1000MIPS 之间,依据阿里云的收费标准,我们计算出 1 核 CPU 计费为 0.0675 元/小时,1GB 内存计费为 0.0375 元/小时,即虚拟机  $V_i$  单位时间内的计费  $cost_v(V_i)$  为:

$$cost_v(V_i) = 0.0675 \cdot n + 0.0375 \cdot m \quad (23)$$

其中, $n$  是  $V_i$  的 CPU 核数; $m$  是内存容量。由于本文模拟的作业不考虑硬盘需求,因此设定虚拟机计费标准时不考虑硬盘。

### 4.3 实验作业流

本文一共设置了 5 个作业流,作业分不同的批次达到,各批次作业的到达时间按泊松分布来设定。各作业流的到达时间及数量如图 4。具体的参数设置如表 4 所示。

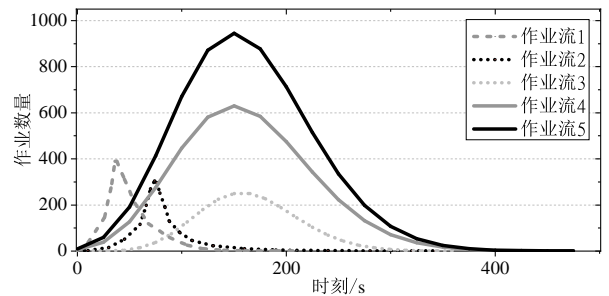


图 4 各批次作业到达时刻与数量

表 4 各作业流的参数设置

批次	间隔	最后到达时刻	作业总数	活跃 VM 数
15	15s	210s	2000	40
20	18s	342s	2000	40
25	15s	360s	2000	40
20	25s	475s	4000	80
20	25s	475s	6000	120



#### 4.4 仿真实验

在仿真调度实验中, 我们设置了如下基准作业调度算法:

先来先服务调度算法 (FCFS): 该算法根据作业到达数据中心的先后顺序, 依次分配至空闲虚拟机。

基于蚁群算法的调度算法 (ACO): 该算法以执行时间和平台能耗作为优化目标, 进行作业调度。

基于遗传算法的主资源负载均衡调度算法 (LVS): 通过对各指标加权设计适应度函数, 接着利用遗传操作挑选染色体, 当满足阈值时输出调度方案。

贪心调度算法 (GA): 该算法将每个作业分配至当前最优虚拟机。

其中, FCFS 和 GA 属于传统作业调度算法, ACO 和 LVS 属于多目标作业调度算法。这些算法和本文算法 (下文中用 AHP-MQ 表示) 在执行时间、服务延迟成本、能耗、成本这几个方面进行对比。

从图 5 可以发现 GA 和 AHP-MQ 在作业总执行时间上表现较好。AHP-MQ 分别比 FCFS、ACO、LVS 平均减少了 42.13%、19.18%、29.76%。

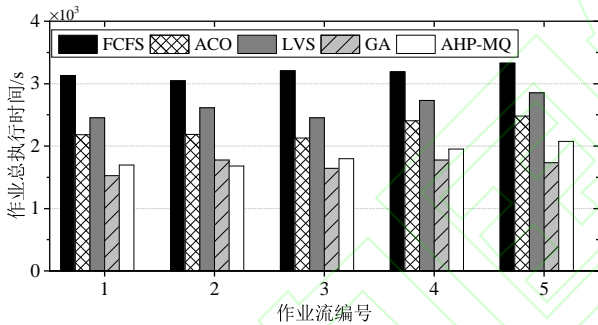


图 5 作业总执行时间

从图 6 中能够发现 FCFS 和 LVS 的服务延迟成本较大, 主要原因是 FCFS 只是按到达顺序分配作业, 并未考虑优化策略, 而 LVS 在调度时关注负载均衡, 不让过多作业集中到性能高的虚拟机上。因此不少作业处于等待状态。AHP-MQ 算法在服务延迟成本这个调度目标上仅次于 GA。

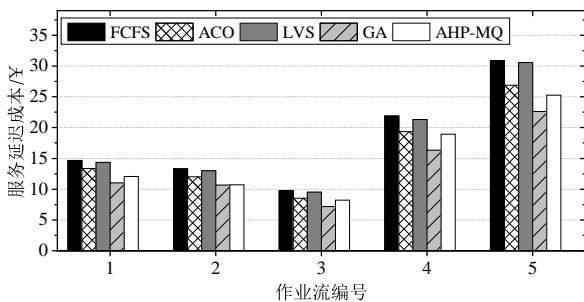


图 6 服务延迟成本

图 7 可发现在平台运行能耗方面, AHP-MQ 算法

和 GA 表现相当, 虽然 AHP-MQ 和 GA 在高性能虚拟机上部署较多作业, 导致单位时间内功率偏高, 但是 HP-MQ 和 GA 的总执行时间要远低于 FCFS、ACO 和 LVS, 所以总的来讲能耗更低。

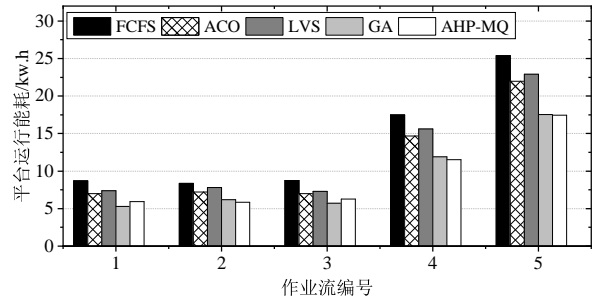


图 7 平台运行能耗

图 8 为总成本, 可以发现, GA 和 AHP-MQ 的总成本比较低。这和能耗较低的原因相似, 均是因为这些调度算法总执行时间少, 此外 AHP-MQ 在计算多目标调度综合评价时, 成本的权重占了较大比例。

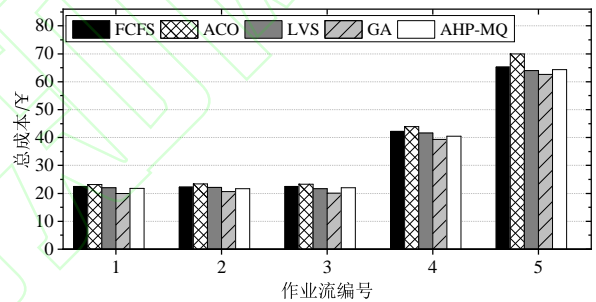


图 8 总成本

在仿真平台内的实验结果说明, 本文所提出的 AHP 定权的多目标强化学习作业调度算法, 在作业总执行时间、平台运行能耗上表现较好。但在服务延迟成本上表现比 GA 劣势, 此外 AHP-MQ 在总成本控制方面也表现的较好。本文提出的 AHP-MQ 算法能发挥强化学习的能力, 较好地优化调度方案。

## 5 结论

云计算的快速发展, 越来越多的作业提交至云数据中心。在具有海量的异构资源的云数据中心中进行作业调度是一个 NP 难问题。本文针对云环境下多目标优化调度问题, 我们定义了执行时间、服务延迟成本、平台运行能耗、成本等调度目标, 其中服务延迟成本用来描述用户对服务质量的满意程度。然后设计了 AHP 定权的多目标强化学习作业调度算法, 并将面向异构资源的多目标调度综合评价方法引入 Q-learning 奖励值的计算, 这有利于后续作业的学习。最后利用 MultiRECloudSim 仿真平台进行仿真实验, 结果表明本文所提出的方法优于大部分对比



方法,能较好地保障用户及服务提供商的利益.在今后的研究工作中,我们将考虑负载均衡,优化奖励值的计算,使该算法更加全面合理.

## 参考文献 (References)

- [1] Guo J, Chang Z, Wang S, et al. Who limits the resource efficiency of my datacenter: an analysis of Alibaba datacenter traces[C]. 2019 IEEE/ACM 27th International Symposium on Quality of Service (IWQoS). Phoenix, USA: IEEE, 2019: 1-10.
- [2] Gabaldon E, Vila S, Guirado F, et al. Energy efficient scheduling on heterogeneous federated clusters using a fuzzy multi-objective meta-heuristic[C]. 2017 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE). Naples, Italy: IEEE, 2017: 1-6.
- [3] 李智勇, 陈少淼, 杨波, 等. 异构云环境多目标 Memetic 优化任务调度方法 [J]. 计算机学报, 2016, 39(2): 377-390.  
(Li Z, Chen S, Yang B, et al. Multi-Objective Memetic Algorithm for Task Scheduling on Heterogeneous Cloud[J]. Chinese Journal of Computers, 2016, 39(2): 377-390.)
- [4] Akbari M, Rashidi H. A multi-objectives scheduling algorithm based on cuckoo optimization for task allocation problem at compile time in heterogeneous systems[J]. Expert Systems with Applications, 2016, 60: 234-248.
- [5] 胡成玉, 余果, 颜雪松, 等. 基于改进多目标优化算法的分布式数据中心负载调度 [J/OL]. 控制与决策:1-7 (2019-08-08)[2020-09-05].  
<https://doi.org/10.13195/j.kzyjc.2019.0702>.  
(Hu Y, Yu G, Yan X, et al. Modified Multi-objective Optimization Algorithm for Load Scheduling of Distributed Data Centers[J/OL]. Control and Decision:1-7 (2019-08-08)[2020-09-05].  
<https://doi.org/10.13195/j.kzyjc.2019.0702>.)
- [6] Zhu Z, Zhang G, Li M, Liu X. Evolutionary multi-objective workflow scheduling in cloud[J]. IEEE Transactions on Parallel and Distributed Systems, 2016, 27(5): 1344-1357.
- [7] 苏命峰, 王国军, 李仁发. 基于利益相关视角的多维 QoS 云资源调度方法 [J]. 通信学报, 2019, 40(6): 102-115.  
(Su M, Wang G, Li R. Multidimensional QoS cloud computing resource scheduling method based on stakeholder perspective[J]. Journal on Communications, 2019, 40(6): 102-115.)
- [8] Farahnakian F, Pahikkala T, Liljeberg P, et al. Energy-aware VM Consolidation in Cloud Data Centers Using Utilization Prediction Model[J]. IEEE Transactions on Cloud Computing, 2019, 7(2): 524-536.
- [9] Islam S, Keung J, Lee K, et al. Empirical prediction models for adaptive resource provisioning in the cloud[J]. Future Generation Computer Systems, 2012, 28(1):155-162.
- [10] Chen M, Yuan J, Liu D, et al. An adaption scheduling based on dynamic weighted random forests for load demand forecasting[J]. The Journal of Supercomputing, 2020, 76(3): 1735-1753.
- [11] Yi D, Zhou X, Wen Y, et al. Toward efficient compute-intensive job allocation for green data centers: a deep reinforcement learning approach[C]. 2019 IEEE 39th International Conference on Distributed Computing Systems (ICDCS). Dallas, USA: IEEE, 2019: 634-644.
- [12] Yang Z, Nguyen P, Jin H, et al. MIRAS: Model-based Reinforcement Learning for Microservice Resource Allocation over Scientific Workflows[C]. 2019 IEEE 39th International Conference on Distributed Computing Systems (ICDCS). Dallas, USA: IEEE, 2019: 122-132.
- [13] 罗亮, 吴文峻, 张飞. 面向云计算数据中心的能耗建模方法 [J]. 软件学报, 2014(7): 1371-1387.  
(Luo L, Wu W, Zhang F. Energy Modeling Based on Cloud Data Center[J]. Journal of Software, 2014(7): 1371-1387.)
- [14] 李翔, 姜晓红, 吴朝晖, 等. 绿色数据中心的热量管理方法研究 [J]. 计算机学报, 2015, 38(10): 1976-1996.(Li X, Jiang X, Wu Z, et al. Research on Thermal Management Methods for Green Data Centers[J]. Chinese Journal of Computers, 2015, 38(10): 1976-1996.)
- [15] 李蕊, 李跃, 徐浩, 等. 基于层次分析法和专家经验的重要电力用户典型供电模式评估 [J]. 电网技术, 2014, 38(9): 2336-2341.  
(Li R, Li Y, Xu H, et al. Assessment on Typical Power Supply Mode for Important Power Consumers Based on Analytical Hierarchy Process and Expert Experience[J]. Power System Technology, 2014, 38(9): 2336-2341.)
- [16] Lin W, Xu S, He L, et al. Multi-resource scheduling and power simulation for cloud computing[J]. Information Sciences, 2017, 397: 168-186.
- [17] All Published SPECpower\_ssj2008 Results[EB/OL]. (2020-08-12)[2020-09-13]. [http://www.spec.org/power\\_ssj2008/results/power\\_ssj2008.html](http://www.spec.org/power_ssj2008/results/power_ssj2008.html).

## 作者简介

袁景凌(1975—), 女, 教授, 博士 (博士生导师), 从事云计算、绿色计算和机器学习等研究, E-mail: yuanjingling@126.com.

陈旻骋(1990—), 男, 博士研究生, 从事云计算、分布式计算的研究, E-mail: wester589@163.com.

江涛(1993—), 男, 硕士研究生, 从事云计算、绿色计算的研究, E-mail: jany@whut.edu.cn.

李超(1986—), 男, 长聘教轨特别研究员, 博士 (博士生导师), 从事计算机体系结构的研究, E-mail: lichao@cs.sjtu.edu.cn.