

基于微服务架构的智慧云平台任务分配研究

徐瑛楠¹, 孙景睿²

(1. 吉林建筑科技学院管理工程学院, 吉林长春 130114;

2. 长春理工大学人工智能学院, 吉林长春 130022)

摘要: 传统智慧云平台任务分配方法无法实时调用任务信息, 导致任务分配效率以及任务分配负载均衡性均较差。提出基于微服务架构的智慧云平台任务分配方法。方法通过构建由服务网关层、任务分配层、服务层和配置集群层等组成的基于 Spring Cloud 微服务的智慧云平台, 利用 PC 端与服务网关层调用服务层和用户层任务信息, 任务分配层通过 GIT 仓库和配置集群层拉取配置, 根据用户需求调用云数据库内任务数据, 建立云任务分配模型, 采用动态规划算法实现智慧云平台任务分配。实验结果表明: 上述方法在任务数量为 100 个时, 其分配完成时间仅为 45s; 分配任务的负载数值与任务数量成正比, 同等任务数量的负载值最低, 具有较高负载平衡性; 在无负载情况下, 资源使用率较平稳, 具有较高可靠性。

关键词: 微服务架构; 智慧云平台; 任务分配; 动态规划; 负载均衡性; 云数据库

中图分类号: TP316 **文献标识码:** B

Research on Task Assignment of Intelligent Cloud Platform Based on Microservice Architecture

XU Ying-nan¹, SUN Jing-rui²

(1. School of Management Engineering, Jilin University of Architecture and Technology,

Jilin Changchun 130114, China;

2. School of Artificial Intelligence, Changchun University of Science and Technology, Jilin Changchun 130022, China)

ABSTRACT: Traditional smart cloud platform task allocation methods cannot call task information in real time, resulting in poor task allocation efficiency and load balance. Therefore, this paper reports a novel task allocation method of smart cloud platform based on micro service architecture. Firstly, the smart cloud platform based on spring cloud microservices was composed of service gateway layer, task allocation layer, service layer and configuration cluster layer. Secondly, the PC side and the service gateway layer were applied to call the task information of the service layer and the user layer, and according to the GIT warehouse and the configuration cluster layer, the task allocation layer was adopted to pull the configuration. Then, based on user requirements, the task data in the cloud database was called to establish the cloud task allocation model. Finally, the dynamic programming algorithm was utilized to achieve the task allocation of smart cloud platform. The experimental results show that when the number of tasks is 100, the allocation completion time is only 45 s; The load value of assigned tasks is directly proportional to the number of tasks, and the load value of the same number of tasks is the lowest, which has high load balance; Under no-load condition, the resource utilization rate is stable and has high reliability.

KEYWORDS: Microservice architecture; Smart cloud platform; Task allocation; Dynamic planning; Load balancing; Cloud database

基金项目: 吉林省教育厅“十三五”社会科学项目(JJKH20201242SK)

收稿日期: 2021-03-26 修回日期: 2021-04-06

$$V_{k+1}(O_{k+1}, d_{k+1}, \dots, d_n) \quad (1)$$

当动态规划过程指标为式(1)时,则该指标具有可加性^[10-11]。此时动态规划最优分配方案的递归模型:假设 $P_{k,n}$ 表示所有的策略,以状态 O_k 为基础,过程指标的最优值为最优指标函数,其由 $f_k(o_k)$ 表示,则最优指标函数表达式如下

$$f_k(o_k) = \min_{d_k \in D_k(o_k)} \{V_{k,n}(o_k, P_{k,n})\} \quad (2)$$

上述公式中 $k=1, 2, \dots, n$ 。

当动态规划指标具有可加性时,最优指标函数的加权函数表达式如下:

$$f_k(o_k) = \min_{d_k \in D_k(o_k)} \{v_k(o_k, d_k) + f_{k+1}(o_{k+1})\} \quad (3)$$

依据动态规划基本概念所建立的数学模型可称为任务图到智慧云平台的映射过程,其过程如下:

第一:令 R 表示用户需求和等待分配任务集合,则该集合可由 $R(T, Q)$ 表示成任务图,其中 $T = \{t_1, t_2, \dots, t_n\}$,任务集、节点的权重、子任务运行时间分别由 T, T_i, i 表示; $Q = \{q_1, q_2, \dots, q_m\}$,其中通信集 $q_i = (t_i, t_k)$ 由 Q 表示,是两个节点之间边的权,表示两个子任务之间的通信量。

第二:智慧云平台属于并行计算平台,平台关系图 $H = (P, E)$ 表示该平台,其中,虚拟机集合由 P 表示。

第三:令矩阵 EET 表示 n 个任务分配到 m 个虚拟机完成时间,第 i 行表示第 i 个任务分配到各个虚拟机上的分配完成时间, EET_{ij} 则表示第 i 个任务在第 j 个虚拟机上的分配完成时间。矩阵表达式如下

$$EET = \begin{Bmatrix} eet_{11} & eet_{12} & \dots & eet_{1m} \\ \vdots & \vdots & \vdots & \vdots \\ eet_{n1} & eet_{n2} & \dots & eet_{nm} \end{Bmatrix} \quad (4)$$

第四:在终端机上,以任务分配层分配任务时间最长的虚拟机分配完成时间作为任务分配完成的目标^[12],其终端条件计算公式如下

$$\min \left\{ \max_{j=1}^m \sum_{i=1}^n o_i eet_{ij} \right\} \quad (5)$$

依据上述递归模型和动态规划模式计算多任务多处理机制的分配算法,假设 R 个任务集合内存在 p 个任务,且 p 为整数,建立最短分配时间函数 $G_i(o)$,且 $0 \leq i \leq n, 0 \leq x \leq p$,分配第 i 个任务给所有节点的全局最短时间表达式如下

$$G(p) = \sum_{i=1}^n G_i(o_i) \quad (6)$$

其中

$$p = \sum_{i=1}^n o_i \quad (7)$$

在式(7)中, $G(p)$ 最小 o_i 分配方案为:将每个节点依据其顺序进行编号后,第 o 个任务分配给第 i 节点的最短时间 $f_i(o)$ 和分配给第 i 节点最短时间的任务份额数 $d_i(o)$,其计算步骤如下:

第一步:在一个节点上,分配 o 个任务,其计算公式如下

$$f_1(o) = G_1(o) \quad (8)$$

其中 $x = d_1(o)$,且满足 $0 \leq o \leq p$ 。

第二步:在第一个节点和第二个节点上分配 o 个任务,其计算公式如下:

$$\left. \begin{aligned} f_2(o) &= \min \{G_2(z) + f_1(x-z)\} \\ d_2(o) &= \text{使 } f_2(x) \text{ 最小的 } z \end{aligned} \right\} \quad (9)$$

上述公式中 $0 \leq z \leq x$ 。

依据式(9)可知,前 i 个节点分配到 o 个任务计算公式如下:

假设 b_i 表示第 i 个阶段最短分配完成时间,其计算公式如下

$$b_i = \min \{f_i(1), f_i(2), \dots, f_i(p)\} \quad (10)$$

假设 l_i 是使第 i 个阶段最短分配完成时间的前个节点的任务份额

$$l_i = f_i(o) \quad (11)$$

在每个任务分配步骤中,将计算得到的第 i 个阶段最短分配完成时间、第 i 个阶段最短分配完成时间的前个节点的任务份额、第 o 个任务分配给第 i 节点的最短时间和分配给第 i 节点最短时间的任务份额数保存。当最后一个任务分配完成后,假设 mint 表示全部任务分配完成最短时间,其计算公式如下

$$\text{mint} = \min(b_1, b_2, \dots, b_n) \quad (12)$$

当全部任务分配完成后,此时分配任务节点的最大编号为 k ,此时分配给前面 k 个节点的最优份额计算公式如下

$$l_k = f_k(o) \quad (13)$$

通过上述计算步骤,可实现任务分配求解,先计算将 x 个任务分配给前 i 个节点的最短分配完成时间和该节点分配到的任务份额数,再计算每个节点的最短完成时间、全部任务分配完成时间和总份额以及最大编号 k 值,完成所有任务分配。

3 实验分析

为验证本文方法在智慧云平台任务分配过程中的各项性能,并考虑云计算环境的复杂性,使用 Cloudsim4.0 仿真软件,处理器为 AMD Athlon(tm) IIX250 Processor,内存为 4.0GB 64 位操作系统。通过 Cloudsim 模拟云计算局部域,虚拟机节点数量为 8,设置待分配任务数为 100 个,分别运行本文方法、文献[3]方法和文献[4]方法,测试三种方法实际使用性能,其中文献[3]方法表示工业设计云服务平台协同任务模块化重组与分配方法,文献[4]方法表示基于信任的云服务系统多目标任务分配方法。

3.1 不同方法的任务耗时对比

统计三种方法分配 100 个任务耗时情况,结果如图 3 所示。

分析图 3 可知,三种方法分配任务完成时间均随着任务数量的增加而增加,本文算法在任务数为 10~30 个之间时,其任务分配完成时间为 8s 左右,任务分配完成时间无明显

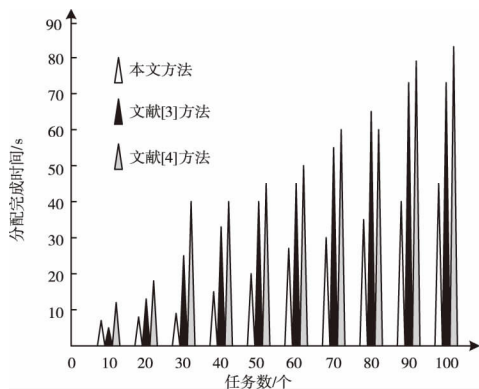


图3 三种方法任务分配耗时

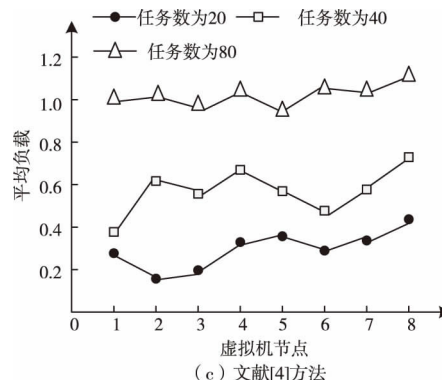


图4 三种方法分配任务负载情况

增加,当任务数为40~100个之间时,任务分配完成时间逐渐增加,且增加幅度较小,而文献[3]方法与文献[4]方法在100个任务分配过程中,二者分配完成时间均较高,且增长幅度较大,当分配任务为100个时,本文算法任务分配完成时间为45s,较文献[3]方法与文献[4]方法任务分配完成时间分别低26s和38s,由此可知,本文方法在任务数量相同时,其在任务分配时间上具有较高优势。

3.2 不同方法的平均负载对比

以平均负载为指标,统计三种方法在分配任务数为20个、40个、80个时负载情况。平均负载为单位时间内,系统处于可运行状态和不可中断状态的进程数。结果如图4所示。

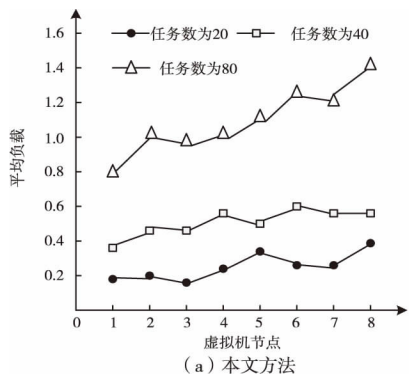
由图4三组图形数据中可知,当任务数量增加时,三种

方法分配任务时的负载值也随之增加,通过对比三种方法在任务数量相同时的负载值,可知本文方法较文献[3]方法和文献[4]方法而言具备较高的负载平衡性。

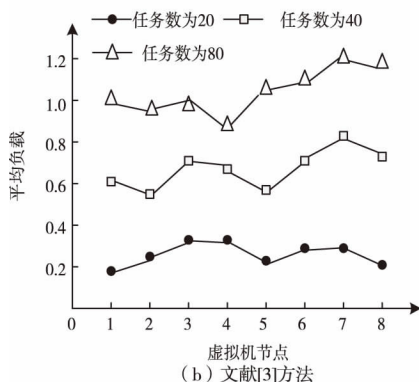
3.3 不同方法的任务利用率对比

上述实验是在有负载情况下,验证本文方法的负载平衡性,处于无负载情况时,以任务分配利用率作为衡量三种分配方法可靠性指标,统计8个虚拟机节点任务分配率情况,结果如图5所示。

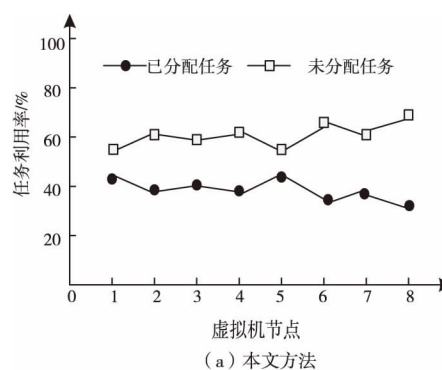
综合分析图5可知,当在无负载情况下,本文方法的资源使用率较平稳,其分配合理性较高,而文献[3]方法和文献[4]方法的任务利用率波动较大,对虚拟机节点压力相对较大,可见文献[3]方法和文献[4]方法运行处于运行不稳定状态,由此可知,本文方法任务分配合理且具有较高可靠性。



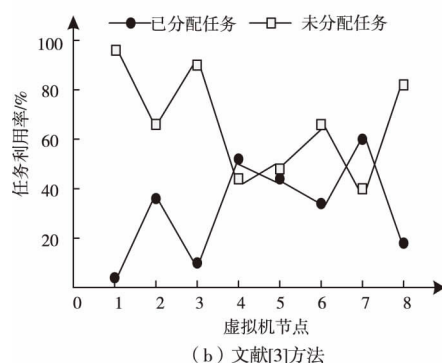
(a) 本文方法



(b) 文献[3]方法



(a) 本文方法



(b) 文献[3]方法

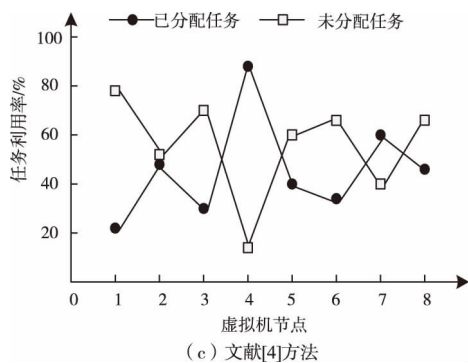


图5 三种方法任务分配利用率

3.3 不同方法的负影响因子数值对比

统计三种方法在任务分配过程中的负影响因子数值,研究三种方法任务分配效率情况,结果如表1所示。

表1 三种方法的负影响因子数值情况

任务数量	本文方法	文献[3]方法	文献[4]方法
10	0.00	0.02	0.01
20	0.01	0.04	0.04
30	0.02	0.05	0.05
40	0.04	0.08	0.07
50	0.06	0.11	0.09
60	0.06	0.13	0.13
70	0.09	0.15	0.15
80	0.11	0.19	0.17
90	0.14	0.19	0.20
100	0.16	0.24	0.28

分析表1可知,三种方法在进行任务分配时,其负影响因子数值与任务数量成正比,当任务数量相同时,本文方法的负影响因子数值均低于文献[3]方法与文献[4]方法,其中当任务数量为100时,三种方法的负影响因子数值之间差值最大,本文算法负影响因子数值仅为0.16,由此可知,本文算法任务分配效率最高。

4 结论

在云计算里,任务分配的有效度相当重要,对智慧云平台的整体服务性能来说影响巨大。为此,研究基于微服务架构的智慧云平台任务分配方法,将Spring Cloud微服务架构与动态规划算法相结合,实现任务分配。实验结果表明:该算法在任务量相同时,在任务分配时间上具有较高优势且分

配负载数值较低,负载平衡性能强;在无负载情况时,分配合理性较高,可靠性强等。

本文方法虽然能有效实现任务分配功能,且具有分配时间短、负载平衡性强等特点,但在网络技术日新月异的情况下,本文方法仍存在一定不足之处,如在分配过程中存在的冗余性,不仅消耗分配时间,也提升了分配复杂度,对于任务存储类型不够宽泛,因此还需在基于用户需求的基础上,更加完善任务分配方法。

参考文献:

- [1] 孙宇,周纲.基于微服务架构的资源发现系统平台构建研究[J].中国图书馆学报,2020,46(1):114-124.
- [2] 于曼,黄凯,张翔.基于微服务架构的ETC系统设计[J].计算机科学,2020,47(S1):643-647.
- [3] 陈健,莫蓉,初建杰,等.工业设计云服务平台协同任务模块化重组与分配方法[J].计算机集成制造系统,2018,24(3):720-730.
- [4] 束束,梁昌勇,徐健.基于信任的云服务系统多目标任务分配模型[J].计算机研究与发展,2018,55(6):1167-1179.
- [5] 杨巍,林伟明.基于微服务架构的图书馆业务系统重构研究——以深圳大学为例[J].图书馆杂志,2020,39(11):36-43.
- [6] 徐竟伟,赵泽宇,沈敏虎,等.基于微服务架构的下一代IP网络测试体系框架[J].电信科学,2019,35(9):29-37.
- [7] 万书鹏,易强,张凯,等.基于微服务架构的新一代调控系统服务编排技术[J].电力系统自动化,2019,43(22):116-121.
- [8] 翟岩龙,孙文心,包天虹,等.基于微服务的边缘侧仿真方法及框架研究[J].系统仿真学报,2018,30(12):44-53.
- [9] 成方敏,余隋怀,初建杰,等.基于用户知识存量的3D打印云平台知识服务方法[J].计算机集成制造系统,2020,26(9):2541-2551.
- [10] 陆佳伟,李杰,张元鸣,等.基于改进禁忌搜索的云任务负载均衡调度策略研究[J].小型微型计算机系统,2018,39(10):2254-2259.
- [11] 李昆仑,关立伟,郭昌隆.基于聚类和改进共生演算法的云任务调度策略[J].计算机应用,2018,38(3):707-714.
- [12] 裴树军,宋冬梅,孔德凯.Map/Reduce下快速剪枝算法在复杂任务调度中的应用[J].计算机科学与探索,2018,12(1):72-81.

[作者简介]



徐瑛楠(1981-),女(汉族),吉林长春人,硕士,副教授,研究方向:经济学。

孙景睿(1989-),男(满族),吉林舒兰人,硕士研究生,助理实验师,研究方向:仪器总体设计。