

文章编号:1674-2869(2018)04-0468-05

# 基于模糊聚类的云计算集群资源调度算法

周丽娟

山西财经大学实验教学中心,山西 太原 030006

**摘要:**针对现有的云计算集群资源调度算法具有的负载不均衡和在线动态适应能力不强的缺点,提出了一种基于模糊聚类的云计算动态集群资源调度算法。首先,构建了云计算环境下的资源调度模型。然后采用模糊聚类对云计算集群资源进行聚类,根据节点与所有聚类中心的距离判断是否需要增减聚类数量。当新任务到来时,自动计算其到各个聚类中心的距离,将具有最小聚类距离的聚类中心分配给该任务。在Cloudsim环境下进行仿真试验,结果表明该方法能有效地实现云计算集群资源的动态调度,且较其它方法相比,具有反应实时和负载均衡的优点,是一种适合云计算环境的可行任务调度方法。

**关键词:**资源调度算法;云计算;负载均衡

中图分类号:TP393

文献标识码:A

doi:10.3969/j.issn.1674-2869.2018.04.023

## Cluster Resource Scheduling Algorithm in Cloud Computing Based on Fuzzy K-Means

ZHOU Lijuan

Experimental Teaching Center, Shanxi University of Finance and Economics, Taiyuan 030006, China

**Abstract:** To solve the problems of cluster resource scheduling algorithm in cloud computing, such as unbalanced load and weak on-line suiting, we proposed a cluster resource scheduling algorithm based on fuzzy K-means. Firstly, a resource scheduling model was established in cloud environment. Then the cluster resources in cloud computing were cluster-analyzed by a fuzzy cluster technique, and the number of clusters was decided by the distance between the cluster node and the entire cluster centers. When a new task is arrived, the algorithm will automatically calculate the distance to each cluster center and the cluster center with a minimum cluster distance will be assigned to this task. The algorithm is simulated in CloudSim environment and the results demonstrate that the proposed method can effectively realize the dynamic scheduling of cluster resources in cloud computing. Moreover, the algorithm displays real-time reaction and load balance, appearing superior to the other reported methods. The algorithm can be regarded as a feasible task scheduling method suitable for cloud computing environment.

**Keywords:** resource scheduling algorithm; cloud computing; load balance

云计算(cloud computing)作为一种将计算资源作为服务并通过网络并提供给用户的计算模式,使得用户可以通过按需使用计算资源,这些资源主要包括数据、软件、硬件和带宽。云计算通过提供软件即服务、平台即服务和基础设施即服务这三类服务来为用户作业提供解决方案。云计算

中的资源具有地理分散、资源异构和动态变化等特征。

云计算是在网络计算的基础上发展而来,其主要特征是通过引入虚拟化计算,使得云计算环境下的资源调度不同于网络计算,但其目标往往是最小化最大完工时间,并通过启发式算法来寻

收稿日期:2017-06-06

作者简介:周丽娟,硕士,实验师。E-mail: nihao0759@126.com

引文格式:周丽娟. 基于模糊聚类的云计算集群资源调度算法[J]. 武汉工程大学学报, 2018, 40(4): 468-472.

求实现资源最优分配。

美国国家标准技术院将云计算定义为:云计算是一种通过网络并使得用户可以按需来共享的一个可配置的资源池,该资源池能在仅需较少管理开销和交互的情况下,对资源进行快速配置和释放,云计算模式的5个基本特征为:按需自助服务、网络的广泛访问和共享资源池以及快速的弹性能力。

资源调度是云计算的一个核心问题,直接关系到云计算服务的稳定性、资源的使用效率和用户满意程度。云计算的资源调度问题的研究从理论技术本身上来说具有非常重要的意义<sup>[1-3]</sup>。

文献<sup>[4]</sup>针对云计算领域的任务调度问题,提出了一种基于人工免疫的云计算平台动态任务调度算法,通过排队论粗略确定保持云平台稳定的条件,为后面的计算提供基础数据,然后利用免疫理论中的免疫克隆算法来为不同节点分配资源实现最优配置,实现负载平衡处理。为了提高传统资源调度算法的资源利用率,文献<sup>[5]</sup>提出了一种云计算环境下的资源调度模型,该资源调度模型基于人工蜂群算法进行调度方案优化。文献<sup>[6]</sup>分析讨论了1组 Directed acyclic graph (DAG)共享云计算资源调度<sup>[7-8]</sup>中的多DAG数量、属性结构分布特点和资源需求量关系,提出了一种基于资源需求强度预测变异方法的进化算法(evolutional algorithm based on the forecasting of resource demand, EFRD)。文献<sup>[9]</sup>研究了云端资源的管理和调度,从服务商需求角度出发来构建云资源调度方法,能在不损害用户和生产商利益的前提下实现收益平衡。

上述工作研究了云计算任务调度,具有重要的意义,本文提出了一种基于模糊聚类的云资源调度方法,在对云计算异构资源进行模糊聚类的基础上,实现资源的动态调度,并通过实验证明了文中方法的有效性。

1 资源调度目标函数

本文的云计算资源调度模型采用 Map-Reduce 编程/资源调度模型<sup>[10-14]</sup>,如图1所示。

从图1可以看出,云计算每个单元包含2个节点:主控节点 Jobtracker 和从节点 TaskTracker,主控节点负责任务的资源分配、调度以及失败任务的跟踪,从节点主要负责任务的执行。

TaskTracker 从节点可以形式化的表示为  $G(V, E)$ , 其中  $V$  表示其所属主控节点所能管理的

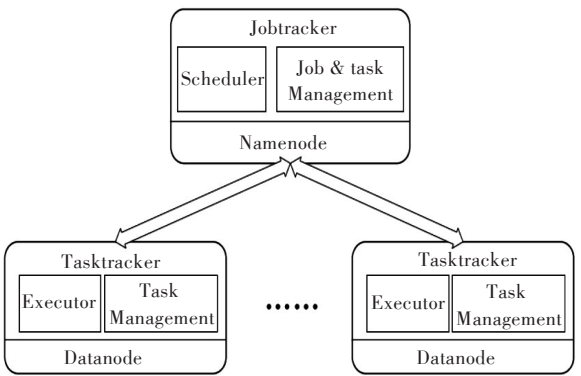


图1 Map-Reduce 编程/任务调度模型  
Fig. 1 Program/task scheduling model of map-reduce

所有节点构成的集合,而  $E$  是对应的边集。因此,这些节点可以执行一系列任务。

本文的主要工作就是采用模糊聚类算法选择最合适分配给节点的资源进行分配。

资源分配过程中考虑的性能指标主要包括:预计执行时间  $T(r)$ 、网络带宽  $B(r)$  和网络延迟  $D(r)$ 。

1)  $T(r)$ :任务  $t_i$  在处理节点  $tn_j$  执行的最早完成时间,可以表示为:

$$T(r) = E_{ij} + S(n_j) \tag{1}$$

式(1)中,  $S(n_j)$  为某任务在节点  $n_j$  上可以执行的最早时间,  $E_{ij}$  为在节点  $n_j$  上执行某任务  $t_i$  所需的执行时间。

- 2)  $B(r)$ :表示路径  $r$  对应的网络带宽最大值。
- 3)  $D(r)$ :路径  $r$  对应的网络延迟最大值。

因此,资源调度的目标函数为:

$$F = \min \frac{aT(r) + bD(r)}{cB(r)} \tag{2}$$

满足:

$$\begin{cases} T(r) < T_{LT} \\ B(r) < E_{LT} \\ D(r) < D_{LT} \end{cases} \tag{3}$$

其中,  $a$ 、 $b$  和  $c$  分别为预计执行时间  $T(r)$ 、网络带宽  $B(r)$  和网络延迟  $D(r)$  的各自权重,  $T_{LT}$ 、 $E_{LT}$  和  $D_{LT}$  为对应的界限。

2 基于模糊聚类的资源调度算法

2.1 模糊聚类

模糊  $K$ -MEANS 算法具有计算简单和聚类速度快的特点,是一种动态聚类算法,能通过不断最小化误差平方和来求取聚类,并求出每个数据样本点到聚类中心的隶属度,隶属度最高的聚类被

定位样本点所属的分类,模糊K-MEANS算法需要确定最优聚类个数和初始聚类中心:聚类个数对于聚类精度具有较大的影响,由于初始阶段无法准确确定聚类个数,因此,通过随机确定聚类个数。

不同的初始聚类中心对聚类结果影响巨大,传统的K均值算法随机选取K个点作为初始聚类种子,并不断通过迭代直到聚类中心确定和算法收敛。

## 2.2 基于模糊K-MEANS资源分配算法

采用模糊聚类算法来对节点资源进行分配的原理可以表示为:

首先设定一个初始的聚类数K,最大聚类数 $K_{\max}$ ,距离阈值为 $\theta_d$ ,当新节点被释放时,需对所有资源进行重新聚类,而当新任务到来时,首先需计算与其具有最小距离的聚类,将最小聚类对应的中心作为资源分配的节点。

选择具有最小距离的聚类中心作为任务被分配的资源节点。

当新节点被释放时,需要计算其与所有聚类中心的距离,如果其与所有聚类中心的距离均大于阈值 $\theta_d$ ,且聚类中心的数量尚未超过允许的最大数量时 $K \leq K_{\max}$ 时,增加一个新的聚类 $S_{K+1}$ ,并计算新的聚类中心。

当新任务到来时,则计算其与所有聚类中心的距离,选择具有最小距离的聚类作为其所属的聚类。

**算法1** 基于改进模糊K-MEANS算法的资源分配算法可以描述为:

输入:所有节点集合和任务集合;

初始化:距离阈值为 $\theta_d$ 、聚类数K,聚类最大数量 $K_{\max}$ ,初始化聚类集合 $S_1, S_2, \dots, S_K$ 空集,初始化数据中心 $c_1, c_2, \dots, c_K$ ,当前迭代次数 $t=1$ 。

**步骤1:**对聚类于新来的任务 $x_i$ ,计算其到每个节点聚类 $S_j$ 的隶属度 $w_{ij}$ :

$$w_{ij} = \left( \frac{\sum_{q=1}^K \left( \frac{\|x_i - c_j\|}{\|x_i - c_q\|} \right)^{2/(m-1)}}{\sum_{q=1}^K \left( \frac{\|x_i - c_j\|}{\|x_i - c_q\|} \right)^{2/(m-1)}} \right), \quad 1 \leq i \leq n, \quad 1 \leq j \leq K \quad (4)$$

其中, $\|x_{ik} - c_{jk}\|$ 表示节点资源 $x_i$ 到数据中心 $c_j$ 之间的欧几里得距离,当中心点的坐标与节点资源 $x_i$ 相同时,即 $\|x_i - c_q\| = 0$ , $w_{ij} = 1$ ,否则 $w_{ij} = 0$ ,且 $w_{ij}$ 满足

$$\sum_{j=1}^K w_{ij} = 1 \quad (1 \leq i \leq n) \quad (5)$$

**步骤2:**对各个节点资源聚类 $S_j$ 的中心 $c_1, c_2, \dots, c_K$ 进行更新:

$$c_j = \frac{\sum_{i=1}^n w_{ij}^m x_i}{\sum_{i=1}^n w_{ij}^m} \quad 1 \leq j \leq K \quad (6)$$

**步骤3:**计算各资源节点 $x_i$ 到所有聚类中心的距离的最小值 $J(W, C)$ :

$$J(W, C) = \min w_{ij}^m \sum_{k=1}^D \sum_{j=1}^K \|x_{ik} - c_{jk}\| \quad (7)$$

**步骤4:**对距离最小值 $J(W, C)$ 进行判断。

如果其大于距离阈值 $\theta_d$ ,判断 $K \leq K_{\max}$ 是否成立:如果成立,则转入步骤5;否则,转入步骤6;

**步骤5:**将节点 $x_i$ 作为聚类中心,并重建新聚类, $c_{K+1} = x_i$ ,聚类 $S_K = \{x_i\}$ ,并将聚类数 $K = K + 1$ 。

**步骤6:**查找任务应分配的节点,即计算 $J(W, C)$ 对应的聚类,使其:

$$j = \arg \min J(W, C_j) \quad (8)$$

**步骤7:**对于具有最小 $J(W, C)$ 的聚类 $S_j$ ,对其聚类中心 $c_j$ 进行调整,得到新的聚类中心为:

$$c_j = c_j + \eta[x_i - c_j] \quad (9)$$

其中, $\eta$ 为学习率。

**步骤8:**判断算法已达到最大迭代次数。

如果达到,则算法结束;否则,判断任务与所有节点聚类中心的距离是否小于距离阈值:

$$J_{\text{sum}}(W, C) = \sum_{i=1}^n w_{ij}^m \sum_{k=1}^D \sum_{j=1}^K \|x_{ik} - c_{jk}\| \quad (9)$$

其中, $n$ 为样本总数。

当大于距离阈值时,需对节点进行重新聚类,增加聚类总数。

## 3 结果分析

在Cloudsim云计算仿真环境<sup>[15-17]</sup>下对文中所提的资源调度算法进行仿真和实验,并与文献<sup>[5]</sup>方法和文献<sup>[6]</sup>方法进行比较,文中方法的参数设置为:物理节点数量为100,CPU计算能力2000(MI·s<sup>-1</sup>),内存为8GB,网络带宽为150(MI·s<sup>-1</sup>),任务总数为1000个,任务个数与任务分配均为随机的。

### 3.1 负载均衡程度

采用负载均衡因子来作为性能比较指标,负载均衡因子可以定义为:

$$\phi = \sqrt{\frac{\sum_{j=1}^m (L_k - \bar{L}_{jk})^2}{m-1}}$$

(10)

式(10)中, $m$ 为实验总次数, $L_j$ 为虚拟机 $r_j$ 在某时刻 $k$ 的负载均衡值, $\bar{L}_j$ 为云计算数据中心在某时刻 $k$ 的平均负载。

当实验次数为50时,运行文中方法与文献<sup>[5]</sup>和文献<sup>[6]</sup>方法进行比较,3种方法得到的负载均衡因子的比较如图2所示,在图2中可以发现,文中方法的负载均衡能力最强,文献<sup>[6]</sup>方法次之,而文献<sup>[5]</sup>方法的负载均衡能力较差,因为其负载均衡因子远远大于另外两种方法,且在仿真时间为400 s时,即仿真结束时仍没有收敛的趋势。而文中方法通过模糊聚类对所有节点进行动态重分配,并将任务分配给需求最匹配的节点,因此具有最好的负载均衡能力。

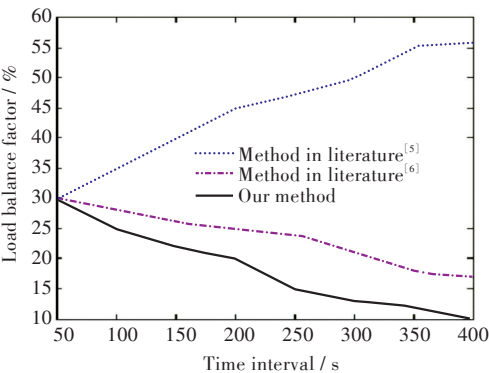


图2 负载均衡效果比较

Fig. 2 Comparisons of load balance effect

3.2 负载均衡效率

为了进一步对3种算法的负载均衡效率进行比较,在计算出所有的负载均衡因子后,可以计算得到负载均衡效率:

$$E_{T_k} = \frac{\eta_{load\_0} - \eta_{load\_k}}{T_k}$$

(11)

其中, $\eta_{load\_0}$ 表示聚类中心的负载均衡值,从公式(11)可以看出,当 $E_{T_k}$ 的值越小时,对应算法的负载均衡效率越高。

3种算法的负载均衡效率的比较如图3所示。从图3可以看出,文中方法的负载均衡效率较高,文中方法得到的负载均衡效率的平均值分别为0.012 7,而文献<sup>[5]</sup>和文献<sup>[6]</sup>方法的值则分别为0.041 6和0.027 5,文中方法的负载均衡效率最高。

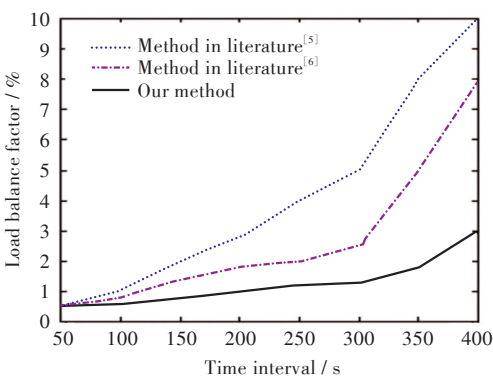


图3 负载均衡效率

Fig. 3 Comparisons of load balance effect

4 结 语

为了实现云环境下资源的高效调度,设计了一种基于模糊聚类的云计算集群资源调度方法。通过模糊聚类方法对所有资源节点进行聚类,当新任务到来时,自动计算其到各个聚类中心的距离,将具有最小聚类距离的聚类中心分配给该任务。在Cloudsim仿真工具下进行实验,结果表明了文中方法能有效实现节点负载均衡,与其他方法相比,具有较好的负载均衡程度和负载均衡效率,具有较强的应用前景。

参考文献:

[1] 崔勇, 宋健, 缪葱葱, 等. 移动云计算研究进展与趋势[J]. 计算机学报, 2017, 40(2): 273-295.

[2] 武志学. 云计算虚拟化技术的发展与趋势[J]. 计算机应用, 2017, 37(4): 915-923.

[3] BOKHARI M U, MAKKI Q, TAMANDANI Y K. A survey on cloud computing [J]. International Journal on Computer Science & Engineering, 2018, 5(6): 302-311.

[4] 杨镜, 吴磊, 武德安, 等. 云平台下动态任务调度人工免疫算法[J]. 计算机应用, 2014, 34(2): 351-356.

[5] 卓涛, 詹颖. 改进人工蜂群算法的云计算资源调度模型[J]. 微电子学与计算机, 2014, 7(31): 147-155.

[6] 田国忠, 肖创柏, 赵娟娟. 云计算环境下多DAG调度的资源分配进化算法[J]. 计算机应用研究, 2014, 31(9): 2798-2802.

[7] 张水平, 邬海艳. 基于元胞自动机遗传算法的云资源调度[J]. 计算机工程, 2012, 38(11): 11-13.

[8] 徐鹏, 张岩江, 苏森. PaaS云资源调度技术研究[J]. 华中科技大学学报(自然科学版), 2013, 41(增刊2): 52-56.

[9] 张爱科, 符保龙. 基于最大收益平衡点动态变化的云



资源调度算法[J]. 重庆邮电大学学报(自然科学版), 2014, 26(5):706-711.

[10] 王福成, 齐平, 李龙澍. 一种基于局部最优云服务选择模型的云资源调度算法[J]. 小型微型计算机系统, 2014, 35(12):2697-2701.

[11] 袁正午, 李君琪. 基于改进粒子群算法的云资源调度[J]. 计算机工程与设计, 2016, 37(2):401-404.

[12] 张恒巍, 韩继红, 卫波, 等. 基于 Map-Reduce 模型的云资源调度方法研究[J]. 计算机科学, 2015, 42(8):118-123.

[13] 张群慧, 李仁发. 膜计算改进粒子群优化算法的云资源调度[J]. 计算机工程与应用, 2013, 49(20):40-44.

[14] 边根庆, 张文敬, 邵必林, 等. 基于帕累托最优的云资源调度研究[J]. 计算机工程与应用, 2014, 50(19):70-73.

[15] 邢乃若, 惠晓滨. 改进的云资源调度自相关匹配数据融合算法[J]. 科技通报, 2014(12): 217-219.

[16] 黎明, 吴跃, 陈佳. 基于语义搜索引擎的云资源调度[J]. 计算机应用研究, 2015, 32(12):3735-3737.

[17] 金瑾, 何嘉. 基于遗传蚁群算法的云资源调度问题研究[J]. 成都信息工程学院学报, 2013(2): 109-113.

本文编辑:陈小平



(上接第 454 页)

2007(8):892-897.

[12] 李凯, 王庆锋, 郝帅, 等. 竖直管降膜蒸发器溢流孔式布膜器 CFD 模拟研究[J]. 北京化工大学学报(自然科学版), 2017, 44(1):57-62.

[13] 王为术, 王涛, 彭岩, 等. 竖管内 R113 降膜流动特性的结构影响研究[J]. 矿山机械, 2017, 45(2):65-71.

[14] 李铁, 吕昌尧, 宋济洋. 降膜随气体同时穿越液池过程数值模拟研究[J]. 东北电力大学学报, 2017, 37(1):80-86.

[15] 战洪仁, 吴众, 慧晓, 等. 基于 VOF 模型的倾角对重力热管性能影响数值研究[J]. 热力发电, 2017, 46(6):40-45.

[16] 谭思超, 赵富龙, 李少丹, 等. VOF 模型界面传质与体积传质的转换方法[J]. 哈尔滨工程大学学报, 2015, 36(3):317-321.

本文编辑:陈小平