Journal of Computer Applications
计算机应用,2013,33(8):2154 -2157
ISSN 1001-9081
CODEN JYIIDU
2013-08-01
http://www.joca.cn

文章编号:1001-9081(2013)08-2154-04
doi:10.11772/j.issn.1001-9081.2013.08.2154

云计算环境下对资源聚类的工作流任务调度算法

郭凤羽[1],禹　龙[2*],田生伟[3],于　炯[3],孙　华[3]
(1.新疆大学 信息科学与工程学院,乌鲁木齐 830046;　2.新疆大学 网络中心,乌鲁木齐 830046;　3.新疆大学 软件学院,乌鲁木齐 830008)
(*通信作者电子邮箱 yulong_xju@126.com)

摘　要:针对云计算环境中资源具有规模庞大、异构性、多样性等特点,提出了一种对资源进行模糊聚类的工作流任务调度算法。经过对网络资源属性进行量化、规范化,以预先构建的任务模型和资源模型为基础,结合模糊数学理论划分资源,使得在任务调度时能够较准确地优先选择综合性能较好的资源类簇,缩短了任务资源相匹配的时间,提高了调度性能。通过仿真实验将此算法与 HEFT、DLS 进行比较,实验结果表明,当任务在[0,100]范围增加时,该算法平均 SLR 比 HEFT 小 3.4%,比 DLS 小 9.9%,其平均 Speedup 比 HEFT 大 5.9%,比 DLS 大 10.2%;当资源在[0,100]范围增加时,该算法平均 SLR 比 HEFT 小 3.6%,比 DLS 小 9.7%,其平均 Speedup 比 HEFT 大 4.5%,比 DLS 大 10.8%。所提算法实现了对资源的合理划分,且在执行跨度方面具有优越性。

关键词:云计算;工作流任务调度;资源属性;模糊聚类;资源划分
中图分类号:TP393.027　　**文献标志码**:A

Workflow task scheduling algorithm based on resource clustering in cloud computing environment

GUO Fengyu[1], YU Long[2*], TIAN Shengwei[3], YU Jiong[3], SUN Hua[3]
(1. *School of Information Science and Engineering, Xinjiang University, Urumqi Xinjiang* 830046, *China*;
2. *Network Center, Xinjiang University, Urumqi Xinjiang* 830046, *China*;
3. *School of Software, Xinjiang University, Urumqi Xinjiang* 830008, *China*)

Abstract: Focusing on the characteristics of resource under large-scale, heterogeneous and dynamic environment in cloud computing, a workflow task scheduling algorithm based on resource fuzzy clustering was proposed. After quantizing and normalizing the resource characteristics, this algorithm integrated the theory of clustering to divide the resources based on the workflow task model and the resource model constructed in advance. The cluster with better synthetic performance was chosen firstly in scheduling stage. Therefore, it shortened the matching time between the task and the resource, and improved the scheduling performance. By comparing this algorithm with HEFT (Heterogeneous Earliest Finish Time) and DLS (Dynamic Level Scheduling), the experimental results show that the average SLR (Schedule Length Ratio) of this algorithm was smaller than that of HEFT by 3.4%, the DLS by 9.9%, and the average speedup of this algorithm was faster than that of HEFT by 5.9%, the DLS by 10.2% with the increase of tasks in a certain range of [0,100]; when the resources were increased in a certain range of [0,100], the average SLR of this algorithm was smaller than that of HEFT by 3.6%, the DLS by 9.7%, and the average speedup of this algorithm was faster than that of HEFT by 4.5%, the DLS by 10.8%. The results indicate that the proposed algorithm realizes the reasonable division of resources, and it surpasses HEFT and DLS algorithms in makespan.

Key words: cloud computing; workflow task scheduling; resource characteristic; fuzzy clustering; resource division

0　引言

云计算是随着计算、存储以及通信技术的快速发展而出现的一种崭新的共享基础资源的商业计算模型。云计算通过全面共享多种资源,结合资源的异构性、动态性等特点,处理海量数据,完成用户各种计算需求[1-3]。显然,这样在给用户带来便利的同时,对任务调度和资源分配技术提出了更高的要求。而工作流[4-5]通过描述实际应用中的具体业务流程,完成实际的业务目标可以为云计算环境下的调度提供有效的解决方案。

目前,云计算环境下工作流任务调度算法作为云计算工作流技术的重要组成部分,已经成为该领域内的研究热点。调度问题是 NP 完全问题,国内外研究者也提出了很多相关调度算法。列表调度以其简单的思想和较高的调度效率得到广泛的研究,例如:HEFT(Heterogeneous Earliest Finish Time),CPOP(Critical Path On a Processor)[6]等,此类算法处理异构环境下任务之间有依赖关系的调度问题,可以将任务插入相应资源的空闲时间,却没有考虑任务在选择资源时的处理时间。文献[7]通过定义关键任务和空闲时间片,提出了一种异构环境下关键任务调度算法 HCT(Heterogeneous Critical Task),该算法将关键任务复制到具有空闲时间片的资源上,将任务开始时间提前。此外,还有遗传算法[8-9]、蚁群算法[10]等基于人工智能的算法在调度工作方面的研究。

这些算法在选择资源时,其针对的对象是全体资源,没有

收稿日期:2013-02-04;**修回日期**:2013-03-19。　**基金项目**:新疆维吾尔自治区自然科学基金资助项目(2013211A010)。
作者简介:郭凤羽(1990 -),女,河南邓州人,硕士研究生,主要研究方向:云计算、分布式计算;　禹龙(1974 -),女,新疆乌鲁木齐人,副教授,硕士,主要研究方向:计算机网络、计算机智能;　田生伟(1973 -),男,新疆乌鲁木齐人,教授,博士,主要研究方向:云计算、分布式计算、计算机智能;　于炯(1964 -),男,北京人,教授,博士,主要研究方向:云计算、网格与分布式计算、网络安全;　孙华(1977 -),女,新疆喀什人,讲师,博士,主要研究方向:信息安全、信誉管理。

考虑资源本身的特点,导致任务选择资源的时间开销大。因此找到合适的方法对资源进行划分能够缩小资源选择的范围。聚类是一种无监督的分类方法,是解决资源分类问题的一种有效方式[11]。一般的聚类划分是将每个待处理的对象严格处理,而云计算环境下的资源没有明确的属性划分。文献[12]结合小世界理论对服务资源进行多维性能聚类,为资源分类和调度提供了有益的参考。本文结合 Zadeh 所提出的模糊集理论[13],提出一种对资源聚类[14]的工作流调度算法(Fuzzy Clustering Based Workflow Task Scheduling,FCBWTS),通过对云环境下的资源进行模糊聚类的预处理,实现资源归类,改进传统的列表调度排序将工作流中任务进行优先排序,根据优先权值完成其任务调度。

1　调度模型

1.1　工作流任务模型

云计算建立在高性能计算基础之上,且是并行系统的体现。本文采用有向无环图(Directed Acyclic Graph,DAG)来描述云计算工作流任务结构,具体采用二元组 $G=(T,E)$ 来表示,其中:T 表示 DAG 中节点集合,描述了任务集 $T=\{t_1,t_2,\cdots,t_n\}$,$n=|T|$ 表示任务总数,用 $W(t_i)$ 表示任务 $t_i\in T$ 所需要的计算量;$E=\{e_{ij}\mid e_{ij}=(t_i,t_j),e_{ij}\in T\times T\}$ 是有向边集合,表示任务 t_i 和 t_j 之间有依赖关系,t_j 必须在 t_i 完成后才能开始处理,$C(t_i,t_j)$ 描述任务 t_i 和 t_j 在不同资源上处理所需的通信量。当 t_i,t_j 分配到同一个资源上执行时,$C(t_i,t_j)$ 则为 0。$G=(T,E)$ 中没有前驱的任务称为入口任务(t_{entry}),没有后继的任务称为出口任务(t_{exit})。$PRED(t_i)=\{t_j\mid t_j\in T,e_{ji}\in E\}$ 表示 t_i 前驱任务集合,$SUCC(t_i)=\{t_j\mid t_j\in T,e_{ij}\in E\}$ 表示 t_i 后继任务。

整个工作流任务执行跨度为 *makespan*,它是通过最后一个任务的完成时间来确定的,而 DAG 关键路径上的任务决定了整个工作流任务的执行速度。在同构环境下,将任务通过 *b-level*、*t-level* 进行降序排列,优先调度关键路径上的任务,继而完成任务调度。而本文考虑到异构环境下资源计算能力和网络链路通信能力不同的特点,对任务排序方法进行改进,加入资源计算能力中值 M_{comp} 和链路传输能力中值 M_{com} 来衡量任务的优先程度,增大了调度算法准确性,这里用 $priority(t_i)$ 表示节点优先程度:

$$priority(t_i)=\frac{W(t_i)}{M_{\text{comp}}}+\max_{t_j\in SUCC(t_i)}\left(\frac{C(t_i,t_j)}{M_{\text{com}}}+priority(t_j)\right) \tag{1}$$

设定 $AST^*(t_i,p_j)$ 是工作流任务 t_i 的实际开始时间,$AFT^*(t_i,p_j)$ 是任务 t_i 在资源 p_j 上的实际完成时间,如式(2)所示。其中 $W(p_j)$ 见第 1.2 节详细说明。

$$AFT^*(t_i,p_j)=AST^*(t_i,p_j)+\frac{W(t_i)}{W(p_j)} \tag{2}$$

当 $t_j\in PRED(t_i)$,任务 t_i、t_j 分别在资源 p_x、p_y 上执行时,$AT(t_i,t_j,p_y)$ 表示任务 t_i 的数据到达 t_j 的时间,$C(t_i,t_j)$ 是任务 t_i 和 t_j 之间的通信量,$C(p_x,p_y)$ 详见下一小节。

$$AT(t_i,t_j,p_y)=AFT^*(t_i,p_x)+\frac{C(t_i,t_j)}{C(p_x,p_y)} \tag{3}$$

这样得到任务 t_j 在资源 p_y 上最早开始时间 $EST^*(t_j,p_y)$ 为

$$EST^*(t_j,p_y)=\max\left\{AFT^*(t_k,p_y),\max_{t_i\in PERD(t_j)}(AT(t_i,t_j,p_y))\right\} \tag{4}$$

1.2　资源模型

在传统的调度算法中,对于网络环境定义不明确或者忽略其异构性,而本文从资源间计算能力不同和不同链路的传输速度不同两方面来设定云计算异构环境。云环境网络用带权无向图 $G_P=(P,E_P)$ 表示,其中:$P=\{p_1,p_2,\cdots,p_n\}$ 是云计算资源集合,$n=|P|$ 是图中节点的个数,也就是资源个数,$W(p_i)$ 是节点 $p_i\in P$ 单位时间内资源的计算量;$E_P=\{e_{ij}\mid e_{ij}=(p_i,p_j),e_{ij}\in P\times P\}$ 是边(链路)的集合,$C(p_i,p_j)$ 表示链路 $(p_i,p_j)\in E_P$ 单位时间内传输的通信量。

$P=\{p_1,p_2,\cdots,p_n\}$ 是资源集合,第 i 个资源 p_i 可以进一步刻画其特征 $\boldsymbol{Q}_i=(q_{i1},q_{i2},\cdots,q_{im})$,$m$ 是特征属性个数,这里选取资源网络中重要的 5 个特点,取 $m=5$,各个属性含义如下:

1) q_{i1} 表示网络中资源 p_i 的单位计算能力,也就是 DAG 中节点权值;

2) q_{i2} 表示与资源 p_i 相连的链路通信能力的均值,即与其相连的边的平均权值;

3) q_{i3} 表示与资源 p_i 相连的传输速度最快的链路,即 DAG 中与该节点相连的边的最大值;

4) q_{i4} 表示 p_i 所处的网络位置,本文用该资源到网络中其他资源节点的距离方差来描述;

5) q_{i5} 表示与资源 p_i 相连的边数,表示与其相连接的链路数目。

任务所分配的资源的计算能力、通信能力、存储能力等因素都会影响其后继任务乃至整个工作流任务的完成,因此,对资源特征进行矢量化处理为后续划分工作做好准备。

1.3　资源聚类

云环境下工作流任务对资源需求不能准确描述,并且资源自身特征属性也具有“模糊性”。本文使用模糊聚类的方法将资源按照性能进行分类,聚类主要步骤如下:

步骤 1　以云环境中的每个资源性能特征 $\boldsymbol{Q}_i$ 建立初始化样本矩阵。

步骤 2　对原始数据进行标准化处理,消除矩阵中的量纲,将数据压缩到[0,1],得到 q_{ik}'。

$$q_{ik}'=\frac{q_{ik}-\bar{q}_k}{s_k};\ i=1,2,\cdots,N,\ k=1,2\cdots,m \tag{5}$$

其中:$\bar{q}_k=\frac{1}{N}\sum_{i=1}^{N}q_{ik}$ 是第 k 个特征向量原始数据的均值,$s_k=\sqrt{\frac{1}{N}\sum_{i=1}^{N}(q_{ik}-\bar{q}_k)}$ 为原始数据的标准差。

步骤 3　计算不同资源之间的相似度 r_{ij},构建模糊相似矩阵 $\tilde{\boldsymbol{R}}$。

$$r_{ij}=\frac{1}{m}\sum_{k=1}^{m}\exp\left(-\frac{3(q_{ik}''-q_{jk}'')}{4s_k^2}\right) \tag{6}$$

步骤 4　经过模糊相似矩阵的迭代运算而保持 r_{ij}' 不变,得到模糊等价关系矩阵 $\boldsymbol{R}^*$。

$$r_{ij}'=r_{ij}=\bigvee_{k=1}^{N}(r_{ik}\wedge r_{jk})=\max_{1\leqslant k\leqslant N}(r_{ik}\wedge r_{jk}) \tag{7}$$

步骤 5　设定阈值 $\alpha(0\leqslant\alpha\leqslant1)$,得到表示资源分类的数据聚类。

2　调度算法

2.1　算法描述

在云计算环境中资源按照特征性能模糊聚类的基础上,设计了此工作流任务调度算法,主要包括三个阶段:

1)构建调度顺序表。

算法结合资源特点,通过计算出每个任务的优先权值建立调度的优先级队列(readylist),同时,记录关键路径上的任务节点(如果有多条关键路径,选择通信量大的)。选择优先级别高的任务优先调度,将所记录关键路径的任务尽量放置于同一资源,减小其通信开销。

算法按照 1.1 节所示计算每个任务的优先权值 $priority(t_i)$,以图 1 为例,图中每个任务的优先值如表 1 所示。

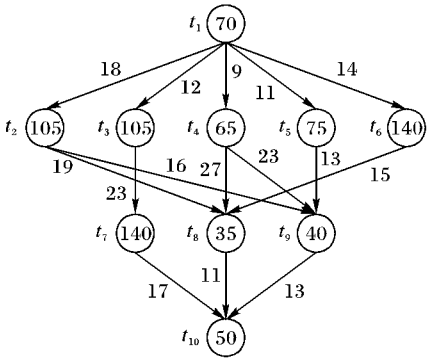


图 1 工作流任务 DAG 结构示意图

表 1 图 1 工作流任务优先权值

节点	优先权值	节点	优先权值	节点	优先权值
t_1	15.63	t_5	7.31	t_8	3.53
t_2	8.47	t_6	9.03	t_9	3.87
t_3	12.43	t_7	7.13	t_{10}	1.43
t_4	8.09				

2)资源聚类。

资源的聚类是在工作流任务进行资源选择之前的预处理,根据对每个资源的信息提取,得到原始数据,根据模糊聚类算法对资源进行划分,以包含 13 个资源节点的网络图为例,得到原始性能指标如表 2 所示。

表 2 资源网络原始数据

资源	特征				
	q_1	q_2	q_3	q_4	q_5
p_1	20.00	0.05	0.05	6.00	1.00
p_2	30.00	0.08	0.05	9.00	2.00
p_3	35.00	0.10	0.10	10.00	2.00
p_4	50.00	0.13	0.10	14.00	4.00
p_5	45.00	0.13	0.10	13.00	4.00
p_6	50.00	0.13	0.10	14.00	3.00
p_7	30.00	0.10	0.10	9.00	1.00
p_8	28.00	0.10	0.10	8.00	1.00
p_9	48.00	0.11	0.03	13.00	5.00
p_{10}	33.00	0.10	0.10	9.00	1.00
p_{11}	30.00	0.08	0.08	8.00	1.00
p_{12}	35.00	0.10	0.10	10.00	1.00
p_{13}	40.00	0.12	0.12	11.00	1.00

按照第 1.3 节的聚类方法,经过设定阈值和计算,最终将 13 个资源按照性能特点划分为以下聚类:$\{\{p_1\}, \{p_2\}, \{p_3, p_7, p_8, p_{10}, p_{12}\}, \{p_4, p_5, p_6\}, \{p_9\}, \{p_{11}\}, \{p_{13}\}\}$。按其综合性能从高到低排序为:$\{p_4, p_5, p_6\}, \{p_9\}, \{p_{13}\}, \{p_3, p_7, p_8, p_{10}, p_{12}\}, \{p_2\}, \{p_{11}\}, \{p_1\}$。

3)分配调度。

资源预处理完成后,将其模糊聚类结果按照综合性能分为可选集合和备选集合。当工作流任务调度优先级列表构建完成之后,计算任务在可选集合中完成时间最大最小差值,差值越大表示分配资源对其调度影响较大,故此任务要优先调度。

计算任务在可选集合中资源的最早开始时间和完成时间选择资源调度从而得到该任务调度时间。在调度过程中,如果可选集合没有空缺且所调度任务权值比较高时,则选择备选集合中综合性能最高的聚类资源进行分配,直至所有工作流任务调度完成。

2.2 算法分析

尽量缩短每个任务的调度完成时间,就能缩短整个工作流任务调度的执行跨度(*makespan*)。基于此,本文考虑到资源分配对后继任务执行情况具有一定程度的影响,提出了基于资源模糊聚类的任务调度算法,算法核心代码如下所示:

```
1)  Set the computation costs of the resource and communication
    costs of edges with mid-value.
2)  Compute priority for all tasks by traversing graph upward,
    starting from the exit task.
3)  Sort the workflow tasks in a scheduling list by descending order
    of priority(t_i) values called readylist.
4)  if (process ∈ firstProcess && ! isempty (firstProcess))
5)    for each t_i ∈ readylist          //computing priority(t_i);
6)      if (priority(t_i) > maxValue)
7)        maxValue = priority(t_i);
8)        task = t_i;
9)      end if
10)   end for each
11)   makespan = AFT*(task, process);
12) else
13)   if (AFT*(task, process) ≤ makespan)
14)     min (computing(task, process));
                          //put task in the minimize process
15)   else
16)     if (AFT*(task, process') < AFT*(task, process))
17)       makespan = AFT*(task, process');
18)     end if
19)   end if
20) end if
```

其中算法第 1)~3)步是通过计算资源的计算能力和通信能力的中值,得到工作流任务优先权值,设置调度就绪列表(readylist)。而模糊聚类作为调度之前的准备工作,极大地缩小了任务调度在选择资源时的空间和时间复杂度。第 4)~11)步是可选集合中有可用资源时,将优先级别高的任务调度到使其完成时间最小的资源上。第 12)~19)步则对备选集合的资源使用分配进行说明。

3 实验及结果分析

为了客观分析本文所提出算法 FCBWTS 的性能,实验设计了用于随机产生工作流任务图和资源网络图的相关程序,根据用户设定参数,产生相应实验图。

3.1 实验性能

为了评估本文所提出算法的性能,在 *makespan* 的基础上,采用了常用的性能比较参数:SLR(Schedule Length Ratio)和 Speedup。

SLR 是一种度量调度算法性能的主要方法,

$$SLR = \frac{makespan}{\sum_{t_i \in CP_{\min}} \min_{p_j \in P} \{w_{i,j}\}} \tag{8}$$

其中 $CP_{\min}$ 指当每个任务节点都使用最小计算时间的资源时所确定的关键路径上任务的集合。算法的 SLR 越小表明该算法具有越好的性能。

加速度是串行执行时间与并行执行时间之比,根据所有分配任务在单个资源上的最小执行时间计算出串行上执行时间。

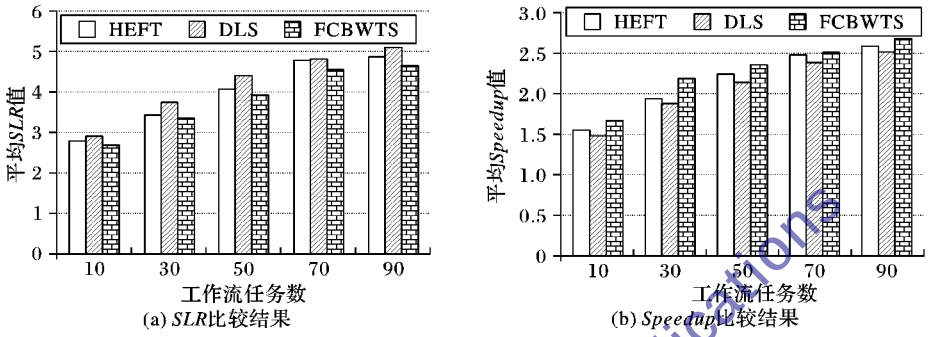
$$Speedup = \frac{1}{makespan} \min_{p_j \in P} \left\{ \sum_{t_i \in T} w_{i,j} \right\} \quad (9)$$

3.2　实验结果和分析

本文选择 HEFT 和 DLS(Dynamic Level Scheduling)作为对比算法进行实验。HEFT 在对任务进行排序时仅考虑了全局的信息,没有考虑当前任务对后继任务执行的通信开销的影响以及资源所带来的问题;而 DLS 计算得出预备任务和空闲资源相匹配的动态级别,也考虑了资源环境的异构性。与以上两者相比,本文对资源量化的处理能更准确地表现资源特性。

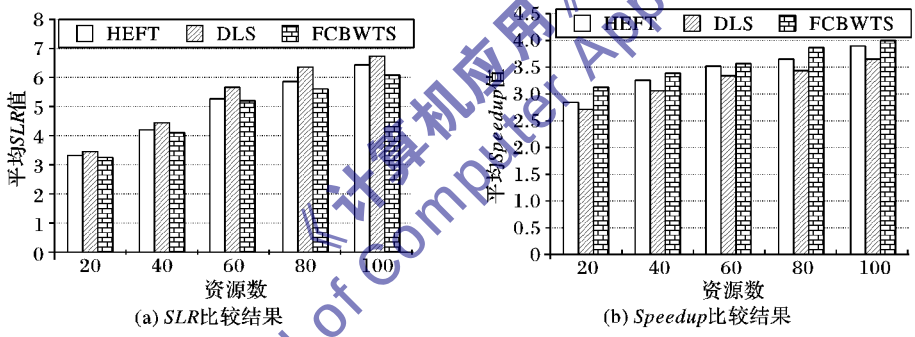
实验 1　设置工作流任务个数 $n = \{10,30,50,70,90,100\}$,资源数为 $m = 10$,任务计算量和通信量范围为[50,200],资源单位时间计算量的生成范围是[10,150],单位时间通信量在[10,100]范围中,最大出度为 3,由设计的随机任务、资源图生成程序进行 30 次实验,取得平均值,获得 *SLR*、*Speedup* 如图 2 所示。通过对比分析,本文 FCBWTS 算法的平均 *SLR* 比 HEFT 小 3.4%,比 DLS 小 9.9%;平均 *Speedup* 比 HEFT 大 5.9%,比 DLS 大 10.2%。

实验 2　设置工作流任务个数 $n = 20$,资源数 $m = \{20,40,60,80,100\}$,资源单位时间计算量的生成范围是[10,150],单位时间通信量范围为[10,100],最大出度为 3。实验随机取得 30 次平均值,获得 *SLR*、*Speedup* 如图 3 所示。通过对比分析,FCBWTS 算法的平均 *SLR* 比 HEFT 小 3.6%,比 DLS 小 9.7%;平均 *Speedup* 比 HEFT 大 4.5%,比 DLS 大 10.8%。



(a) *SLR*比较结果　(b) *Speedup*比较结果

图 2　工作流任务数对各算法性能的影响



(a) *SLR*比较结果　(b) *Speedup*比较结果

图 3　资源数对各算法性能的影响

4　结语

本文针对云计算环境下工作流任务调度问题进行研究,总结了传统的调度算法特点。通过深入分析云计算资源的特点,提出了一种对资源进行聚类的工作流任务调度算法 FCBWTS。该算法将资源特征矢量化,用来表示资源的综合性能,并基于这些特征对资源进行模糊聚类,缩小任务选择资源的范围,从而减小了所有任务的时间跨度。最后,将 FCBWTS 算法与已有算法 HEFT、DLS 进行对比分析,结果表明本文提出的算法提高了工作流任务的调度性能。

下一步,将在考虑云计算环境下通信能力的动态性、链路的可靠性等复杂情况下,对调度在安全性、动态性等作进一步完善和扩展,改进目前的调度策略。

参考文献:

[1] 陈全,邓倩妮. 云计算及其关键技术[J]. 计算机应用,2009,29(9):282 - 290.

[2] ARMBRUST M, FOX A, GRIFFITH R, *et al*. A view of cloud computing [J]. Communications of the ACM, 2010, 53(4): 50 - 58.

[3] LUIS M. V, LUIS R-M, JUAN C, *et al*. A break in the clouds: towards a cloud definition [J]. ACM SIGCOMM Computer Communications Review, 2009, 39(1): 50 - 55.

[4] 罗海滨,范玉顺,吴澄. 工作流技术综述[J]. 软件学报,2000,11(7): 899 - 907.

[5] 柴学智,曹健. 面向云计算的工作流技术[J]. 小型微型计算机系统,2012,33(1):90 - 95.

[6] TOPCUOGLU H, HARIRI S, WU M-Y. Performance-effective and low-complexity task scheduling for heterogeneous computing [J]. IEEE Transaction on Parallel and Distributed Systems, 2002, 13(3): 260 - 274.

[7] LAN Z, SUN S X. Scheduling algorithm based on critical tasks in heterogeneous environments [J]. Journal of Systems Engineering and Electronics, 2008, 19(2): 398 - 404.

[8] OMARA F A, ARAFA M M. Genetic algorithms for task scheduling problem [J]. Journal of Parallel and Distributed Computing, 2010, 70(1): 13 - 22.

[9] 宋娇,葛临东. 一种遗传模糊聚类算法及其应用[J]. 计算机应用,2008,28(5): 1197 - 1199.

[10] DORIGO M, GAMBARDELLA L M. Ant colony system: a cooperative learning approach to the traveling salesman problem [J]. IEEE Transactions on Evolutionary Computation, 1997, 1(1): 53 - 66.

[11] 张敏,于剑. 基于划分的模糊聚类算法[J]. 软件学报,2004,15(6):858 - 869.

[12] 陈志刚,杨博. 网格服务资源多维性能聚类任务调度[J]. 软件学报,2009,20(10): 2766 - 2775.

[13] ZADEH L A. Is there a need for fuzzy logic [J]. Information Sciences, 2008, 178(13): 2751 - 2779.

[14] 姚婧,何聚厚. 基于模糊聚类分析的云计算负载平衡策略[J]. 计算机应用,2012,32(1):213 - 217.