

## 基于模糊聚类的云任务调度算法

李文娟<sup>1,2</sup>, 张启飞<sup>1</sup>, 平玲娣<sup>1</sup>, 潘雪增<sup>1</sup>

(1. 浙江大学 计算机学院, 浙江 杭州 310018; 2. 杭州师范大学 钱江学院, 浙江 杭州 310012)

**摘要:** 为了更好地实践云计算提供廉价按需服务的宗旨, 提出了一种在模糊聚类基础上, 基于两级调度模式的任务调度 (FCTLBS, fuzzy clustering and two level based task scheduling) 算法, 新算法设置用户调度和任务调度 2 个等级。对资源进行性能模糊聚类; 根据任务参数计算资源偏好, 使不同偏好任务在不同聚类中选择, 缩小了选择范围, 更好地反映了任务需求。仿真实验表明, 本算法较之同类算法具备一定的优越性。

**关键词:** 云计算; 模糊聚类; 任务调度; 分配公平

中图分类号: TP301.6

文献标识码: B

文章编号: 1000-436X(2012)03-0146-09

## Cloud scheduling algorithm based on fuzzy clustering

LI Wen-juan<sup>1,2</sup>, ZHANG Qi-fei<sup>1</sup>, PING Ling-di<sup>1</sup>, PAN Xue-zeng<sup>1</sup>

(1. Computer Science and Technology, Zhejiang University, Hangzhou 310018, China;

2. Qianjiang College Hangzhou Normal University, Hangzhou 310012, China)

**Abstract:** In order to better serve cloud's objective of providing low-cost and on-demand services, a two-level and fuzzy clustering based cloud scheduling algorithm named FCTLBS(fuzzy clustering and two level based task scheduling algorithm) was proposed. Scheduling was divided into two levels: user scheduling and task scheduling. It used fuzzy clustering method to cluster resources according to their capabilities and computed tasks' resource bias coefficient according to their parameters. Thus tasks with different resource preference were able to choose different capability resources in the corresponding clusters. The new algorithm reduces the scale of resource choice, better reflects cloud tasks' service requirements. Simulation results show that the proposed algorithm has some advantages compared to other kindred algorithms.

**Key words:** cloud computing; fuzzy clustering; task scheduling; distribution justice

### 1 引言

云计算 (cloud computing) 是在网格计算、并行计算、P2P 和虚拟化等技术基础上发展起来的一种新型的共享信息基础设施的模式, 也是未来提供网络服务的重要形式<sup>[1-3]</sup>。云计算的宗旨是为企业和其他终端用户提供廉价按需服务, 因而特别强调 QoS。而任务调度模型的优劣直接影响云服务质量,

毋庸置疑是云计算研究的重要组成部分<sup>[4-6]</sup>。云计算研究目前仍处于初级阶段, 有关云调度的算法还非常稀少, 但传统的其他并行分布式系统中的调度模型<sup>[7,8]</sup>有很多。其中以最小化作业完成时间为目标的元任务调度算法有 OLB、贪心 (greedy) 算法、最大最小 (max-min)、最小最小 (min-Min)、Sufferage 算法<sup>[9-12]</sup>等以及基于人工智能的调度算法 (遗传算法、蚁群算法)<sup>[13,14]</sup>等。上述算法将任务

收稿日期: 2011-07-05; 修回日期: 2012-02-02

基金项目: 浙江省科技厅公益技术应用研究基金资助项目 (2010C31003)

Foundation Item: The Public Technology Application Research Project of Zhejiang Province (2010C31003)

与资源绑定时,更多的是考虑如何缩短任务执行时间,但云计算的目标是提供按需服务,这样无法反映用户真实的QoS要求。另外,传统的任务调度算法在选择资源时,其针对的对象是全体服务资源,未考虑资源本身的特性和用户或任务的偏好,导致资源选择的开销大且存在盲目性。因此找到一种合适的方法对资源进行一定程度的划分能够缩小资源选择范围,从而降低任务调度开销。聚类方法是对目标进行分类的有效手段。聚类的主要方法有:谱系聚类法、基于等价关系的聚类方法、图论聚类法和基于目标函数的聚类方法等<sup>[15-17]</sup>。资源模糊聚类<sup>[18-20]</sup>的研究集中于最近几年,比如杜晓丽等人提出的网格DAG任务图调度算法<sup>[18]</sup>、陈志刚等提出的网格服务资源多维性能聚类任务调度<sup>[19]</sup>等,为资源聚类和任务调度提供了有益的参考。但之前的聚类调度算法大多只考虑计算任务,并且仅对资源进行综合性能优劣的划分,主要目标是减少总执行时间,没有考虑资源的其他属性也没有考虑任务的期望。

为更好地体现云计算“廉价按需”服务的宗旨,反映资源公平公正分配的原则,在对云提供商的资源按性能进行模糊聚类的基础上,采用两级调度模式,按照任务的期望进行按需公平调度,本文提出了一种云环境下基于模糊聚类的两级任务调度(FCTLBS, fuzzy clustering and two level based task scheduling)算法。理论与实验分析表明,该算法在提高云系统整体效率和实现调度公平上具备一定的优越性。

本文的创新之处在于:1)用模糊聚类方法对云资源进行划分,在实际分配时减少了资源搜索的空间,降低了算法复杂度,同时更能充分有效发挥对应资源的优势;2)提出任务偏好的计算方法,能够更好分析任务的实际需求;3)提出一种模糊聚类基础上的两级任务调度算法,依据资源聚类结果及任务偏好,最大限度体现调度公平性。同时对用户实施整体调度,对任务实施微观调度,有助于减小问题规模,并为未来在调度中加入安全及其他机制奠定基础。

## 2 调度模型

### 2.1 用户模型

用户是指请求云服务、提交云任务的实体。使用 $U=\{u_0, u_1, u_2, \dots, u_{k-1}\}$ 表示用户集合, $k=|U|$ 代表用户数。其中第*i*个用户 $u_i(i \in [0, k-1])$ 又可以进一步表示为

$u_i=\{u_{ID}, u_{Name}, u_{UAid}, u_{TaskSet}\}$ ,各属性的含义如下。

- 1)  $u_{ID}$  表示用户编号,用户编号在云系统中是唯一的。
- 2)  $u_{Name}$  表示用户名称。
- 3)  $u_{UAid}$  表示用户代理的编号。
- 4)  $u_{TaskSet}$  表示用户提交的任务集。

### 2.2 任务模型

云计算植根于并行系统之上,云任务可以由有向无环图(DAG): $G=(T, E)$ 来表示。其中, $T=\{t_0, t_1, t_2, \dots, t_{n-1}\}$ 表示任务集, $n=|T|$ 代表任务数; $E$ 表示任务间执行先序关系的边集。第*i*个任务 $t_i(i \in [0, n-1])$ 又可以进一步刻画为 $t_i=\{t_{ID}, t_{Length}, t_{Status}, t_{RRes}, t_{ORSet}, t_{DeadLine}, t_{Data}\}$ ,各属性的含义如下。

- 1)  $t_{ID}$  表示任务编号,任务编号对于同一用户的任务而言是唯一的。
- 2)  $t_{Length}$  表示任务的长度,用于判断任务的计算量。
- 3)  $t_{Status}$  表示任务状态,任务可能的状态包括创建(CREATED)、就绪(READY)、执行(INEXEC)、完成(FINISH)、等待(WAITING)、撤销(CANCELED)、挂起(PAUSED)、唤醒(RESUMED)和失败(FALSE)。任务刚创建时被赋予CREATED状态;当执行条件满足时转为READY状态;获得处理机执行时进入INEXEC状态;因等待资源或其他协作任务完成转为WAITING状态;若从内存中调出的任务为PAUSED状态,相应地调入内存状态为RESUMED;任务执行完毕转为FINISH状态;若任务请求的资源在本系统中无法达成或缺乏执行可能则成为FALSE状态。

4)  $t_{RRes}$  表示任务的资源需求。资源需求 $t_{RRes}$ 又可以进一步刻画为 $t_{RRes}=\{t_{Comp}, t_{BW}, t_{Stor}\}$ ,其中 $t_{Comp}$ 、 $t_{BW}$ 、 $t_{Stor}$ 分别表示任务对资源计算能力、通信能力和存储能力的需求。

5)  $t_{ORSet}$  表示任务已分配的资源集合。

6)  $t_{DeadLine}$  表示任务的最晚完成时间,用于辅助评判调度策略的成功与否。

7)  $t_{Data}$  表示处理任务所需要的相关数据,包括 $t_{InputData}$ 和 $t_{OutputData}$ 。

### 2.3 资源模型

$R=\{r_0, r_1, r_2, \dots, r_{m-1}\}$ 表示云系统中的资源集合。 $m=|R|$ 代表资源数。其中,第*j*个资源 $r_j(j \in [0, m-1])$ 又可以进一步刻画为 $r_j=\{r_{ID}, r_{User}, r_{Cap}\}$ ,各属性的

含义如下。

- 1)  $r_{ID}$  表示资源编号。
- 2)  $r_{User}$  表示资源隶属的云提供商。
- 3)  $r_{Cap}$  表示资源的性能。本文使用 3 维向量来刻画资源的性能, 包括计算能力、网络传输能力、存储能力。

因此, 资源性能  $r_{Cap}$  又可以进一步刻画为  $r_{Cap} = \{r_{Comp}, r_{BW}, r_{Stor}\}$ ,  $r_{Comp}$ 、 $r_{BW}$ 、 $r_{Stor}$  分别表示资源的计算能力、通信能力和存储能力。

**定义 1** 资源综合性能  $r_{GP}$ : 由资源的各项性能需求综合计算得出, 计算公式如下:

$$r_{GP_j} = \sqrt{\frac{a(r_{Comp_j})^2 + b(r_{BW_j})^2 + d(r_{Stor_j})^2}{a + b + d}} \quad (1)$$

其中, 参数  $a$ 、 $\beta$ 、 $d$  分别表示计算、带宽和存储能力的经验系数。

**定义 2** 资源相似度  $r_{PDT_{ij}}$ : 表示资源  $i$  与资源  $j$  综合性能的相似度, 由资源之间综合性能的距离计算得出, 公式如下:

$$r_{PDT_{ij}} = |r_{GP_i} - r_{GP_j}| \quad (2)$$

**定义 3** 聚类资源综合性能  $c_{rGP}$ : 某一类型资源聚类的综合性能均值, 公式如下:

$$c_{rGP} = \frac{1}{n} \sum_{i=1}^n r_{GP_i} \quad (3)$$

其中, 参数  $n$  表示类内资源数。

### 2.4 资源模糊聚类

模糊  $C$  均值聚类方法 (FCM) 是 Bezdek 提出用来对数据进行模糊聚类的方法<sup>[21]</sup>。给定数据集  $X = \{x_1, x_2, \dots, x_n\}$ , FCM 算法的目标是找到该数据集的一个模糊划分聚类  $U_{c \times n}$ , 即划分原数据为  $c$  个模糊组, 并使得每组样本与其组聚类中心的非相似性指标达到最小。聚类目标函数的一般定义如下:

$$J(U, c_1, L, c_c) = \sum_{i=1}^c J_i = \sum_{i=1}^c \sum_j u_{ij}^m d_{ij}^2 \quad (4)$$

其中,  $u_{ij}$  表示数据  $j$  隶属于模糊组  $i$  的隶属度, 其值介于 0, 1 之间; 而  $c_i$  是模糊组  $i$  的聚类中心,  $d_{ij} = \|c_i - x_j\|$  为第  $i$  个聚类中心与第  $j$  个数据点间的欧氏距离;  $m \in [1, \infty)$  是一个加权指数。

本文使用 FCM 方式将资源按照性能模糊划分为 3 个类别。聚类的主要步骤如下。

- 1) 以系统中资源性能向量  $r_{Cap}$  建立初始化样

本矩阵。

- 2) 对原始数据进行标准化处理, 将数据压缩到 [0,1] 之间。

- 3) 将资源随机分为 4 个类别, 初始化模糊划分矩阵  $U$ 。

- 4) 计算聚类中心  $c_i, i \in \{1,2,3\}$ 。

5) 计算类内各节点与聚类中心的目标函数值。若小于某个确定的阈值, 或它相对上次目标函数值的改变量小于某个阈值, 则算法停止。

- 6) 计算新的划分矩阵  $U$ 。返回步骤 4)。

以包含 8 个资源节点的云系统为例, 假定初始状态下资源的各种性能指标如表 1 所示。

表 1 资源性能参数

资源号	计算能力	网络能力	存储能力
$r_1$	60	10	5
$r_2$	40	5	10
$r_3$	20	40	5
$r_4$	100	10	15
$r_5$	20	5	50
$r_6$	40	40	8
$r_7$	40	20	28
$r_8$	80	30	15

原始数据及经标准化处理后的资源性能数据如图 1 所示。

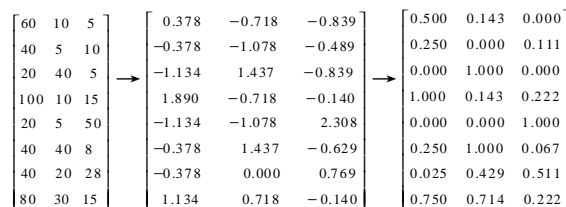


图 1 资源性能数据

数据标准化处理的公式如下:  $r'_{ij} = (r_{ij} - \bar{r}_j) / S_j$ , 其中,  $\bar{r}_j$  为各资源在第  $j$  个性能上的均值,  $S_j$  为资源在第  $j$  维性能上的标准差。再次, 以极值标准化方法将矩阵中的数据转化为闭区间 [0, 1] 上的数值。计算方法如下:  $r''_{ij} = (r'_{ij} - r'_{j \min}) / (r'_{j \max} - r'_{j \min})$ 。

经过设定阈值和若干次迭代运算, 最终将 8 个资源划分为 3 个聚类, 分别为  $Cluster0 = \{r_1, r_4, r_8\}$ ,  $Cluster1 = \{r_3, r_6\}$ ,  $Cluster2 = \{r_2, r_5, r_7\}$ 。其中  $Cluster0$  标识为计算型资源集,  $Cluster1$  标识为带宽型资源集, 而  $Cluster2$  标识为存储型资源集。

### 2.5 任务偏好系数

**定义 4** 任务综合资源期望  $t_{GR}$ : 由任务对资源的各项性能需求综合计算得出, 公式如下:

$$t_{GR_i} = \sqrt{\frac{a(t_{Comp_i})^2 + b(t_{BW_i})^2 + c(t_{Stor_i})^2}{a+b+c}} \quad (5)$$

其中, 参数  $a$ 、 $b$ 、 $c$  分别表示计算、带宽和存储能力的经验系数。

**定义 5** 任务偏好系数  $t_{RC}$ : 经过量化的任务对资源性能的敏感度。对应于资源的 3 种类别, 本文将任务区分为计算敏感型、带宽敏感型和存储敏感型。

任务偏好系数的计算: 任务综合资源期望  $t_{GR}$  与各种类型的资源之间综合性能的最短距离。

$$t_{RC_i} = \min_{j \in \{ \text{计算型, 带宽型, 存储型} \}} \{ t_{GR_i} - c_{rGP_j} \} \quad (6)$$

其中,  $c_{rGP_j}$  表示第  $j$  类资源聚类的综合性能。

## 3 FCTLBS 调度算法

在资源按性能模糊聚类的基础上, 设计了基于两级调度模型的调度算法。算法将云调度分为两级: 用户调度和任务调度。用户调度是指以云用户为单位的调度。用户调度模块根据用户的整体需求和当前云市场中各提供商的情况, 实现用户与若干提供商及资源集的绑定。任务调度实现同一用户内部任务与用户所获得的虚拟资源的绑定。用户调度的层面高于任务调度。

### 3.1 实施框图

新的调度模型在开放云系统中设立云信息服务中心 (CISC, cloud information service center) 以及云用户中心 (CUC, cloud user center)。以各提供商的单云平台为依据划分服务域, 域代理由提供商设立, 域代理代表提供商实施域管理策略, 包括资源模糊聚类算法的实施, 资源任务的绑定等。用户首次登录时向 CUC 注册, 而后每次登录时, 依据注册账号向 CUC 提交本次登录的各项需求, 包括服务类型、资源需求、时间期限和费用约束等。云提供商 (通过域代理) 向 CISC 注册。CISC 与 CUC 可以交互。因而用户可以获取提供商列表, 云提供商也可以向用户发布资源广告。

在模型实施过程中, 首先实现以云用户为单位的整体调度, 根据用户的整体需求为其选择云服务商或其组合。当用户与特定的几个提供商绑定之

后, 将获得提供商分配的若干虚拟资源以完成用户任务。在任务调度中, 先将云提供商的资源模糊划分为 3 个类别: 计算型、带宽型和存储型。当任务提交时计算任务的偏好系数, 根据任务偏好在合适的资源聚类中为其选择资源。模型的具体实施过程如图 2 所示。

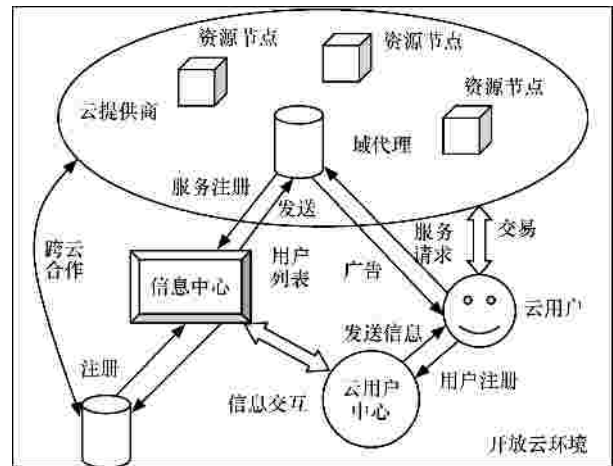


图 2 FCTLBS 算法实施

### 3.2 FCTLBS 算法流程

以各云提供商的单云平台为域进行资源模糊聚类。云资源模糊聚类的算法如下。

```

RCluster(resourceList rlist){
    设定停止阈值  $e$  和最大迭代次数  $MaxTimes$ ;
    迭代次数  $i=0$ ;
    根据输入的资源列表中资源性能为原始数据
    初始化并进行数据标准化处理
    将资源随机分入 3 个类别, 生成初始化划分矩阵  $U^0$ ;
    计算初始聚类中心;
    计算  $Jvalue^0$ 
    do{
         $i++$ ;
        //以下更新划分矩阵  $U^i$ 
        for(int  $x=0;x<3;x++$ )
            for(int  $y=0;y<rlist.size;y++$ )
                {
                    if( $d_{xy}^i > 0$ ){  $m_{xy}^i = \left\{ \sum_{z=0}^2 \left[ \left( \frac{d_{xy}^i}{d_{zy}^i} \right)^{\frac{2}{m-1}} \right] \right\}^{-1}$  };
                    if( $d_{xy}^i == 0$ ){
                         $\mu_{xy}^i = 1$ ;
                    }
                }
    }
    
```

```

        for(int k=0;k<rlist.size;k++)
        {   if(k<>y)  $\mu_{xy}^i=0$ ;}
    }
}
更新聚类中心
计算新的目标函数值  $Jvalue^i$ ;
} While((  $\|Jvalue^i - Jvalue^{i-1}\| > e$  )&&(迭代次数  $i < MaxTimes$ ))
return 划分矩阵  $U^i$  和聚类原型模式矩阵  $P^i$ 
}
用户调度的算法如下。
CUserS(userList ulist, providerList plist){ //用户调度实现用户与提供商的绑定
    While(ulist.iterator().hasNext()){ //需要分配资源的用户列表非空
        CloudUser cuser = (CloudUser) iter.next(); //取出列表中的一个用户
        //用户登录云系统时出示要求的资源类型和各项需求, 据此计算用户的整体资源需求
        requiredResource=cuser. computeRequireCapability ();
        List<CloudProvider> chosenPList=new ArrayList< CloudProvider >(); //创建选中的提供商列表
        /* findProvider 函数将根据该用户的服务类型、自身的等级、安全级别以及本次运行的资源需求, 在现有的提供商列表中寻找最合适的提供商或组合来为该用户服务。*/
        chosenPList=findProvider(requiredResource,plist);
        NodifyProvider(chosenPList, cuser); //通知云提供商需要服务的用户
        NodifyUser(cuser,chosenPList); //通知云用户为其分配的提供商列表
    }
}
在资源模糊聚类的基础上, 任务调度算法如下。
FCTLBS(taskList tlist, resourceList rlist){ //任务调度实现任务与资源的绑定
    RCluster(rlist); //实现某一云提供商资源的模糊聚类, 生成计算型、带宽型和存储型资源聚类
    Sort(comrlist); //对计算型资源聚类按照类内资源性能降序排列

```

```

    Sort(bwrlist); //对带宽型资源聚类按照类内资源性能降序排列
    Sort(storrlist); //对存储型资源聚类按照类内资源性能降序排列
     $c_{rGP1}$ =ComputClusterGP(comrlist); //按照式(3)计算计算型资源聚类的综合性能  $c_{rGP1}$ 
     $c_{rGP2}$ =ComputClusterGP(bwrlist); //按照式(3)计算带宽型资源聚类的综合性能  $c_{rGP2}$ 
     $c_{rGP3}$ =ComputClusterGP(storrlist); //按照式(3)计算存储型资源聚类的综合性能  $c_{rGP3}$ 
    While(tasklist.iterator().hasNext()){ //任务列表非空
        CloudTask cloudt= (CloudTask) iter.next(); //取出任务集中的一个任务
        /* ComputeTRC 函数根据任务的各项资源需求按照式(5)计算任务综合资源期望  $t_{GR}$ , 并根据式(6)确定任务的资源偏好*/
         $t_{RC}$ =ComputeTRC(cloudt,  $c_{rGP1}$ ,  $c_{rGP2}$ ,  $c_{rGP3}$ );
        //根据任务类型从虚拟机列表中寻找合适的虚拟机
        switch( $t_{RC}$ ){
        case 计算型任务:
            在计算型资源聚类中为其寻找合适资源;
            break;
        case 带宽型任务:
            在带宽型资源聚类中为其寻找合适资源;
            break;
        case 存储型任务:
            在存储型资源聚类中为其寻找合适资源;
            break;
        }
    }
}

```

### 3.3 算法性能分析

在云系统中, 某一云用户在一次登录时的完成时间取决于该用户所申请执行的云任务集的最终完成时间, 即完成时间最迟的任务的完成时间。使用  $u_{ct_i}$  表示用户  $i$  的完成时间,  $t_{ct_j}$  表示任务  $j$  的完成时间, 则  $u_{ct_i}$  可以表示为

$$u_{CT_j} = \max_{t_j \in t_{TaskSet_j}} \{CT_j\} \quad (7)$$

对于云任务而言, 其预期完成时间取决于任务的执行时间、通信时间和存储时间。任务  $j$  的通信时间如下:

$$t_{TT_j} = \sum_{k \in t_{ORSet_j}} \frac{t_{Data_{jk}}}{r_{BW_k}} = \sum_{k \in t_{ORSet_j}} \frac{t_{InputData_{jk}} + t_{OutputData_{jk}}}{r_{BW_k}} \quad (8)$$

其中,  $t_{ORSet_j}$  表示任务  $j$  已分配的资源集合,  $t_{Data_{jk}}$  表示任务  $j$  使用资源  $k$  传输的数据量, 相应地,  $t_{InputData_{jk}}$  和  $t_{OutputData_{jk}}$  表示通过  $k$  输入及输出的数据量。  $r_{BW_k}$  表示资源  $k$  的网络传输能力。

任务  $j$  的处理时间为

$$t_{PT_j} = \sum_{k \in t_{ORSet_j}} \frac{t_{Length_{jk}}}{r_{Comp_k}} \quad (9)$$

其中,  $t_{Length_{jk}}$  表示任务  $j$  在资源  $k$  上的计算量,  $r_{Comp_k}$  表示资源  $k$  的计算能力。

$$t_{ST_j} = \sum_{k \in t_{ORSet_j}} \frac{t_{SD_{jk}}}{r_{Stor_k}} \quad (10)$$

其中,  $t_{SD_{jk}}$  表示任务  $j$  在资源  $k$  上的存储量,  $r_{Stor_k}$  表示资源  $k$  的存储能力。因此任务的预期完成时间为

$$t_{CT_j} = t_{TT_j} + t_{PT_j} + t_{ST_j} = \sum_{k \in t_{ORSet_j}} \left( \frac{t_{Data_{jk}}}{r_{BW_k}} + \frac{t_{Length_{jk}}}{r_{Comp_k}} + \frac{t_{SD_{jk}}}{r_{Stor_k}} \right) \quad (11)$$

云系统的吞吐量取决于单位时间内系统服务的云用户总量, 缩短各用户完成时间可以达到此目标。用户完成时间取决于该用户隶属的用户任务集的完成时间。缩短任务完成时间应从缩短任务的计算、传输和存储时间入手。本文将资源分为计算型、带宽型和存储型 3 类, 同时将任务也按照其对资源需求的特点归为对应的计算偏好型、带宽偏好型和存储偏好型 3 个类别, 从而使得对资源某种能力要求高、需求量大的任务获得在此能力上性能较好的资源, 更好地体现了按需分配的原则, 也从一般意义上缩短了任务的完成时间。

RCluster 算法在最坏情况下的时间复杂度为  $o(m \times MaxTimes)$ ,  $m$  表示某一云提供商拥有的资源数,  $MaxTimes$  为最大迭代次数。用户调度程序 CUserS 在最坏情况下的时间复杂度为  $o(n \times k)$ , 其中  $n$  表示云用户数,  $k$  表示云提供商数。任务调度

程序 FCTLBS 是在 RCluster 基础上对任务与资源进行绑定, 因此其在最坏情况下的时间复杂度为  $o(x \times m \times MaxTimes)$ , 其中  $x$  表示任务数,  $m$  与  $MaxTimes$  的含义同前。

## 4 实验设计与分析

云系统中的任务调度算法目前还比较稀少, 在已有的分布式元任务调度算法中, Min-Min 是很多其他调度模型的基础, 另外, 考虑到本调度算法的目标之一是实现资源的按需分配, 也可以说是实践资源分配的公平性原则, 而赵春燕等人提出的基于伯格模型的云任务调度算法<sup>[22]</sup>其宗旨即分配公平, 因此本文将 FCTLBS 与 Min-Min 及基于伯格模型的任务调度算法相比较, 以获得相对客观的评价。

### 4.1 实验设计

实验模拟包含 2 个云提供商和若干云用户构成的云计算环境。以提供商的单云平台作为划分域的依据, 由提供商设立域代理节点, 域代理是代表提供商管理域资源和进行云交易的主体。仿真系统由任务生成器、资源发生器、云交互环境、用户调度器和任务调度器几部分构成。任务生成器按照输入的任务数、任务各项参数基值和对应的任务差异度随机生成任务, 并部署任务参数。资源发生器则按照要求的资源数目、资源性能基数以及资源差异度随机产生资源节点, 并部署资源性能。云交互环境主要实现仿真云环境中不同实体的交互和消息传递。交互系统中包含 3.1 节提到的云信息服务中心 CISC 和云用户中心 CUC。提供商 (通过域代理) 向 CISC 注册、获取用户信息, 用户向 CUC 注册并间接获取提供商的信息。用户调度器是实现用户与提供商绑定的程序, 而任务调度器是实现用户任务集中的任务与用户已经获得的资源绑定的程序。仿真系统模拟 50~1 500 个任务在 5~200 个资源节点上的执行情况。通过任务生成器的控制, 任务的长度控制在  $[t_{Length}, L_{dif} \times t_{Length}]$  之间的值, 任务的计算需求控制在  $[t_{Comp}, C_{dif} \times t_{Comp}]$  之间, 带宽需求控制在  $[t_{BW}, T_{dif} \times t_{BW}]$ , 存储需求控制在  $[t_{Stor}, S_{dif} \times t_{Stor}]$  之间。同理, 通过资源发生器的控制, 将资源的计算能力控制在  $[r_{Comp}, C_{dif} \times r_{Comp}]$  之间, 将资源的网络能力控制在  $[r_{BW}, T_{dif} \times r_{BW}]$  之间, 将资源的计算能力控制在  $[r_{Stor}, S_{dif} \times r_{Stor}]$  之间。其中,  $t_{Length}$  表示任务长度,  $L_{dif}$  表示任务长度差异度。  $t_{Comp}$  表示任务计算量,  $C_{dif}$  表示任务计算量差异度。  $t_{BW}$  表示任务带宽需求,  $T_{dif}$  表示任务带宽需求差异度。  $t_{Stor}$  表示

任务存储需求量,  $S_{dif}$  表示任务存储需求差异度。资源的各项指标类似。

### 4.2 评价指标

调度的目标是实现资源任务的合理匹配, 从而一方面提高系统的整体工作效率, 另一方面提高系统使用者的满意度。为了判断新的调度算法在实践调度目标上的性能, 仿真实验主要设置了 2 个评价指标: 一是任务集的最终完成时间, 用来评价系统的吞吐量; 二是用户满意度, 用于评价资源分配和调度的合理性。本文中用户满意度是根据任务的实际占有资源水平与期望占有水平的相比较计算得出的。

单个任务的满意度  $t_{jvalue_i}$  的计算方法如下所示:

$$t_{jvalue_i} = \ln \left( \frac{r(t_{Comp_i})}{\sum_{k \in t_{ORSet_j}} r_{Comp_k}} + \frac{w(t_{BW_i})}{\sum_{k \in t_{ORSet_j}} r_{BW_k}} + \frac{t(t_{Stor_i})}{\sum_{k \in t_{ORSet_j}} r_{Stor_k}} \right) \quad (12)$$

其中,  $t_{ORSet}$  如前所述表示任务实际分配得到的资源集合,  $r$ 、 $w$ 、 $t$  分别表示计算、带宽和存储资源需求的经验系数。由式 (12) 可以很容易推导出任务集的整体满意度, 即用户满意度  $u_{jvalue}$  的计算方法如下所示:

$$u_{jvalue} = \frac{\sum_{i \in u_{TaskSet}} t_{jvalue_i}}{n} \quad (13)$$

其中,  $u_{TaskSet}$  表示归属于该用户的任务集,  $n$  表示任务的个数。

### 4.3 实验结果分析

由于基于伯格模型的任务调度算法开销很大, 当任务数增加时, 其执行时间呈急速上升趋势, 因而对于任务完成时间的比较分两次进行。任务数较少时, 比较 3 种调度算法的性能; 任务数较多时, 比较 FCTLBS 和 Min-Min 的性能。比较结果如图 3 和图 4 所示。

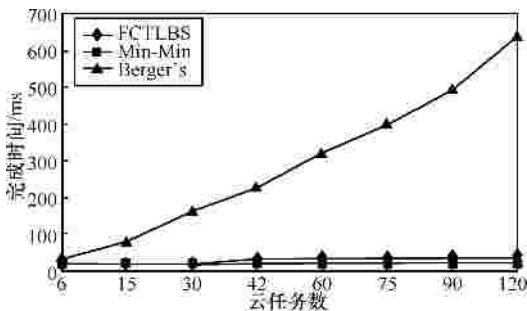


图 3 任务数较少时, 3 种算法完成时间比较

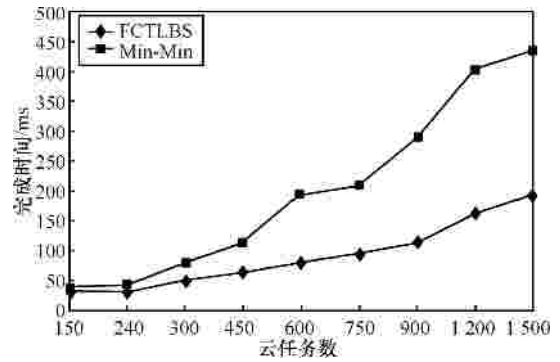


图 4 任务数较多时, 2 种算法完成时间比较

相应地, 对用户满意度的比较也分成两次进行, 比较结果如图 5 和图 6 所示。

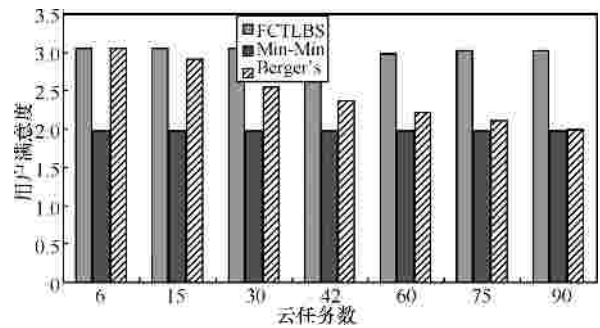


图 5 任务数较少时, 3 种算法满意度比较

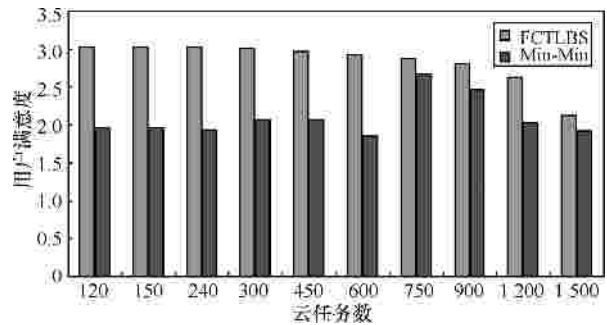


图 6 任务数较多时, 2 种算法满意度比较

从仿真实验结果可以看出, 当问题规模较小 (本文中 0~120 个任务) 时, FCTLBS 与 Min-Min 调度模型的时间效率相当, 优于伯格模型, 在用户满意度上 FCTLBS 与伯格模型相当, 优于 Min-Min; 当问题规模较大 (本文中大于 150 个任务) 时, FCTLBS 在完成时间和用户满意度上均较优。Min-Min 调度算法的主要思想是不断寻找并优先调度任务集中在所有资源上拥有最小完成时间最小的任务, 因而执行速度较快, 但由于其总是优先调度短任务, 无法保证整体用户满意度。伯格模型将欧氏距离最短的任务资源进行匹配, 以期达到用户期待分配资源与实际分配资源

的最大一致性, 整体用户满意度较高, 但无法保证系统吞吐量, 且实现全体任务与全体资源的匹配计算开销大, 在实际系统中实现困难。FCTLBS 算法采取两级调度模式, 以提供商的单云平台为单位实施资源聚类算法, 降低了问题的规模。同时在资源综合性能聚类的基础上, 按照任务的资源需求进行资源和任务的绑定, 更好地体现了按需分配的原则, 也从整体上缩短了任务集的完成时间。由此可见, FCTLBS 算法具备更好的性能和可操作性。

## 5 结束语

本文提出了一种云计算环境中使用的任务调度算法。该算法将调度分为用户调度和任务调度两级。以用户为单位进行的整体调度, 可以更好地发挥用户的自主资源选择权, 反映不同用户的需求, 也有利于后续安全措施的介绍。任务调度中, 采用模糊聚类的方法对不同提供商的资源按照其综合性能进行聚类, 并计算任务资源偏好, 以任务的偏好为依据在对应资源聚类中进行选择。聚类任务调度算法降低了资源选择的范围, 也保证了分配的公平合理性。最后, 通过实验与分析可知, 新的调度算法在确保系统整体吞吐量的同时保证了较高的用户满意度。

## 参考文献:

- [1] 刘鹏. 云计算[M]. 北京: 电子工业出版社, 2010.  
LIU P. Cloud Computing[M]. Beijing: Electronic Industry Press, 2010.
- [2] FOSTER I, KESSELMAN C. The Grid: Blueprint for a New Computing Infrastructure[M]. Morgan Kaufmann, San Francisco, 1999.
- [3] FOSTER I, ZHAO Y, IOAN R, *et al.* Cloud computing and grid computing 360-degree compared[A]. Grid Computing Environments Workshop, 2008. GCE '08 (2008)[C]. 2008.1-10.
- [4] 王鹏. 云计算的关键技术与应用实例[M]. 北京: 人民邮电出版社, 2010.  
WANG P. The Key Technologies of Cloud Computing and Its Application [M]. Beijing: Posts & Telecom Press, 2010.
- [5] 中国云计算论坛[EB/OL]. <http://bbs.chinacloud.cn>.  
Chinese cloud forum[EB/OL]. <http://bbs.chinacloud.cn>.
- [6] MLADEN A, VOU K. Cloud computing-issues research and implementations[A]. Proceedings of the ITI 2008, 30th Int Conf on Information Technology Interfaces[C]. 2008.31-40.
- [7] 朱福喜, 何炎祥. 并行分布式计算中的调度算法理论与设计[M]. 武汉: 武汉大学出版社, 2003.  
ZHU F X, HE Y X. Parallel Scheduling Algorithm in Distributed Computing Theory and Design[M]. Wuhan: Wuhan University Press, 2003.
- [8] CRREA R C. Scheduling multiprocessor tasks with genetic algorithms[J]. IEEE Transactions on Parallel and Distributed Systems, 1999, 10(8):825-837.
- [9] FREUND R, GHENITY M, AMBROSIUS S, *et al.* Scheduling resources in multi-user heterogeneous computing environments with SmartNet[A]. Proceedings of the 7th IEEE Heterogeneous Computing Workshop[C]. Orlando, Florida, USA, 1998.184-199.
- [10] HOU E S H, ANSARI N, REN H. A genetic algorithm for multiprocessor scheduling[J]. IEEE Trans on Parallel and Distributed Systems, 1994, 5(2):113-120.
- [11] IBARRA O H, KIM C E. Heuristic algorithms for scheduling independent tasks on non identical processors[J]. Journal of ACM, 1997, 24(2):280-289.
- [12] TRACY D B, HOWARD J S, NOAH B. A comparison of eleven static heuristics for mapping a class of independent tasks onto heterogeneous distributed computing systems[J]. Journal of Parallel and Distributed Computing, 2001, 61(6):810-837.
- [13] HOLLAND J H. Adaptation in Natural and Artificial Systems[M]. Michigan, USA: Ann Arbor University of Michigan Press, 1975.228-234.
- [14] GAMBARDELLA L M, DORIGO M. Solving symmetric and asymmetric TSPs by ant colonies[A]. Proceedings of the IEEE Conference on Evolutionary Computation[C]. 1996.622-627.
- [15] 曲绍云. 分布式异构系统中任务调度问题的研究[D]. 青岛: 青岛大学, 2005.  
QU S Y. Research on Task Scheduling Problems in Distributed and Heterogeneous Systems[D]. Qindao: Qindao University, 2005.
- [16] 高新波. 模糊聚类分析及其应用[M]. 西安: 西安电子科技大学出版社, 2004.  
GAO X B. Fuzzy Clustering Analysis and Applications[M]. Xi'an: Xi'an Electronic and Technology University Press, 2004.
- [17] 严骏, 姚敏. 模糊聚类算法应用研究[D]. 杭州: 浙江大学, 2006.  
YAN J, YAO M. Research on the Applications of Fuzzy Clustering Algorithms[D]. Hangzhou: Zhejiang University, 2006.
- [18] 杜晓丽, 蒋昌俊, 徐国荣等. 一种基于模糊聚类的网格 DAG 任务图调度算法[J]. 软件学报, 2006, 17(11): 2277-2288.  
DU X L, JIANG C J, XU G R, *et al.* A grid DAG scheduling algorithm based on fuzzy clustering[J]. Journal of Software, 2006, 17(11): 2277-2288.
- [19] 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, 2006, 17(11): 2277-2288.

2009,20(10):2766-2774.

CHEN Z G, YANG B. Task scheduling based on multidimensional performance clustering of grid service resources[J]. Journal of Software, 2009,20(10):2766-2774.

[20] 任大娟,张忠平. 基于模糊聚类的网格资源发现方法研究[D]. 秦皇岛: 燕山大学,2009.

REN D J, ZHANG Z P. Research on Grid Resource Discovery Based on Fuzzy Clustering[D]. Qinhuangdao: Yanshan University,2009.

[21] BEZDEK J C. Pattern Recognition with Fuzzy Objective Function Algorithms[M]. New York: Plenum Press, 1981.10-18.

[22] 赵春燕. 云环境下作业调度算法研究与实现[D]. 北京:北京交通大学, 2009.

ZHAO C Y. Research and Realization of Job Scheduling Algorithm in Cloud Environment[D]. Beijing: Beijing Jiaotong University, 2009.



**张启飞** (1984-), 男, 四川宣汉人, 浙江大学博士生, 主要研究方向为网络安全和云计算。



**平玲娣** (1946-), 女, 江苏常熟人, 浙江大学教授、博士生导师, 主要研究方向为网络安全和安全操作系统。

**作者简介:**



**李文娟** (1978-), 女, 福建福州人, 浙江大学博士生, 主要研究方向为网格计算安全和云计算。



**潘雪增** (1942-), 男, 浙江温岭人, 浙江大学教授、博士生导师, 主要研究方向为网络安全和云计算。

**(上接第 145 页)**

[12] ZHANG Q T, SONG S H. Exact expression for the coherence bandwidth of Rayleigh fading channels[J]. IEEE Transactions on Communications, 2007, 55(7): 1296-1299.

[13] 朱来义. 微积分[M]. 北京: 高等教育出版社, 2009.

ZHU L Y. Calculus[M]. Beijing: Higher Education Press, 2009.



**王俊** (1988-), 女, 重庆人, 电子科技大学硕士生, 主要研究方向为扩频通信、移动通信、通信抗干扰技术等。

**作者简介:**



**马万治** (1977-), 男, 四川都江堰人, 电子科技大学博士生, 主要研究方向为空时编码、MIMO 非相干检测技术、分布 MIMO 信道建模等。



**唐友喜** (1964-), 男, 河南潢川人, 电子科技大学教授、博士生导师, 主要研究方向为 OFDM、UWB、分布 MIMO、传感器网络等。