

## PSO 应用于 QoS 偏好感知的云存储任务调度

王娟, 李飞, 张路桥

(成都信息工程学院 信息安全工程学院, 四川 成都 610225)

**摘 要:** 研究有 QoS 偏好要求的云存储任务调度。首先, 分析云存储与云计算的差异, 用存在矩阵避免无效解的产生。其次, 归纳云存储的 QoS 需求为时间、代价与质量 3 大类, 并据此修改 PSO 算法的适应度函数用权重因子调节 QoS 偏好。实验发现, 在不同优先级任务分布不均的情况下, 分布广的任务的偏好会掩盖其他任务的偏好, 因而不适宜用 PSO 进行整体性调度, 而必须进行分级调度。实验证明, 改进后的分级 PSO 算法对 QoS 偏好具有较好的感知能力。

**关键词:** 云存储; 粒子群算法; 任务调度; 偏好感知

中图分类号: TP393

文献标识码: A

文章编号: 1000-436X(2014)03-0231-08

## Apply PSO into cloud storage task scheduling with QoS preference awareness

WANG Juan, LI Fei, ZHANG Lu-qiao

(School of Information Security Engineering, Chengdu University of Information Technology, Chengdu 610225, China)

**Abstract:** The task scheduling algorithm in cloud storage area with QoS preference awareness was studied. Firstly, the differences between cloud storage and cloud computing were pointed, and exist matrix was used to avoid the creation of useless solutions. Secondly, QoS factors were summarized into three classes: time, cost and quality classes. Based on the three classes, the fitness function of the PSO was improved by using weighted factors to adjust the QoS preference. The simulation results release that PSO is not fit for the overall scheduling when the task distributions of different degrees are unevenness because the QoS demand of wide distribution tasks may cover the QoS demand of other tasks. So a hierarchical scheduling based on degree is necessary. The simulation results show that the improved hierarchical PSO scheduling algorithm can aware the QoS preference relevantly.

**Key words:** cloud storage; PSO algorithm; task scheduling; preference awareness

### 1 引言

近年来, 随着云计算<sup>[1,2]</sup>技术和软件即服务(SaaS)<sup>[3]</sup>思想的兴起, 云存储成为信息存储领域的一个研究热点。云存储系统的任务调度前期的研究多脱胎于数据网格和云计算领域。这些工作<sup>[4-8]</sup>的主要目的是选择哪些响应时间最短的副本。但是云系统与传统网格最大的不同在于“一切都是服务”,

更加关注服务质量(QoS, quality of service)。既然一切都是服务, 那么用户的感受无疑是最重要的。因此, 近年来的云存储任务调度的研究热点在于满足系统 QoS 的要求。但是, 很遗憾现有算法提供的 QoS 普遍存在 2 个致命缺点。

1) QoS 大多是针对系统而言, 而不是使用它的用户。这与“以服务为中心”的宗旨相违背。如果用户的需求得不到满足, 即便使得整个系统拥有最

收稿日期: 2012-08-17; 修回日期: 2013-04-15

基金项目: 四川省应用基础计划基金资助项目(2013JY0064); 四川省科技支撑计划基金资助项目(2011GZ0195); 四川省教育厅科研(青年)基金资助项目(10ZB093)

**Foundation Items:** The Application Basic Research Project of Sichuan Province (2013JY0064); The Science and Technology Support Plan of Sichuan Province (2011GZ0195); The Scientific Research Project of Education Department of Sichuan Province (10ZB093)

大的吞吐率，也应该是失败的，因为结果不被用户认可，系统的价值得不到实现。

2) 针对用户提供的 QoS 几乎都是统一的，没有意识到用户的需求存在差异。这也很不符合实际情况。用户天然存在需求的差别，提供统一的 QoS 服务不但不能满足所有用户的要求，而且造成了不必要的浪费。

举个很简单的例子：某云存储系统提供用户各种各样的数据。系统采用的调度算法以使整个系统的吞吐量最大为目标，从系统角度看这是无可厚非的。但是用户的 QoS 要求是有差异的：其中一些用户不希望多花钱，为此宁愿降低自己的优先级多等待一些时间；而另一些用户，宁愿多花钱来获得最短的响应时间。系统如果不能提供满足用户偏好（时间或代价）的任务调度方案，先满足了愿意等待的用户而延迟了不愿意等待用户的要求，显然会造成 2 种用户都不满意。前者认为自己多付了钱（快的响应往往代表高的花费），后者则觉得浪费了自己的时间。尽管整个系统的吞吐率最高，资源利用率和负载均衡率都较好，但是明显不符合现实情况的需要。

在云计算领域已有少量对 QoS 偏好性的探讨，例如文献[9]。但是，云计算和云存储系统存在很大的区别，这些方法都不适合直接应用于云存储系统，需要做较大修改。云存储系统与云计算的不同主要表现在以下两方面。

1) 云计算的任务是计算型任务，任务可以指派到系统的任意节点执行，区别仅仅在于在高效率（CPU 性能高）节点上执行时间短，而在低效率节点上执行时间长，不会有任务不能由某个节点执行的情况。但是云存储的任务大多是数据传输任务，首要的前提就是节点有任务所需要的数据，因此某任务不能由某个节点提供数据的情况大量存在，这点导致很多从云计算领域移植的算法生成的结果对云存储系统是无效的。

2) 云计算领域的任务之间存在先后关系，即某些任务必须等到另一些的结果后才能执行。而云存储系统的任务一般不存在这种先后关系，数据传输不存在必须先传哪部分后传哪部分。这样使得原有云计算领域中的调度算法可以适当简化，不必考虑任务的先后关系。

以上两点不同提醒大家，现有从传统云计算领域移植的算法应该考虑这两大不同，做出改进以适

应云存储环境的特点。例如：云计算中占据重要地位的 CPU 的计算能力等参数将不再起主导作用，而带宽、网络延迟等性能成为任务调度的重要参考。

由于任务调度是一个 NP 问题，因而以粒子群优化(PSO, partical swarm optimization)算法为代表的启发式算法近年来在云任务调度领域受到关注，取得了不错的系统吞吐率。本文就利用 PSO 算法解决带偏好的云存储任务调度问题进行研究，对 PSO 怎样解决偏好不同、解决的效果和原因进行实验和分析。

## 2 云存储系统任务调度抽象

为了描述方便，先对云存储中的任务调度进行抽象。

1) 用  $R=\{r_1, r_2, \dots, r_i, \dots, r_m\}$  表示  $m$  个资源，其中， $r_i$  表示第  $i$  个资源， $i \in 1, 2, \dots, m$ 。

2) 用  $J=\{j_1, j_2, \dots, j_i, \dots, j_n\}$  表示  $n$  个任务，其中， $j_i$  表示第  $i$  个任务， $i \in 1, 2, \dots, n$ 。

3) 定义  $M \times N$  阶矩阵  $A[m, n]$  为任务分配

(assignment)决策矩阵， $A = \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1m} \\ a_{21} & a_{22} & \dots & a_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \dots & a_{nm} \end{bmatrix}$  中

行是任务，列是资源，即

$$a_{ij} = \begin{cases} 1, & \text{如果 } J_i \text{ 由资源 } R_j \text{ 提供给请求者} \\ 0, & \text{如果 } J_i \text{ 没有由资源 } R_j \text{ 提供给请求者} \end{cases}$$

云存储系统任务调度的目标就是在适应度函数的指引下找到最符合目标函数的调度方案。这里的适应度函数可以简单指任务完成时间，也可以是使用资源的代价、可靠性等。而目标函数则只有 2 种：一种是取适应度函数的最大值，另一种是取适应度函数的最小值。很多情况下，通过改变适应度函数的定义，目标函数可以相互转换。例如：取任务完成时间为适应度函数，一般的目标函数就是取完成时间的最小值。这个定义等同于取任务完成时间的倒数为适应度函数，而目标函数取该倒数的最小值。因此，本文不设专门的目标函数，只使用适应度函数  $F$ ，目标就是取使得  $F$  最小的调度方案。

已有的研究已经证明多因素下的任务调度是一个 NP 难题，没有最优解，只能通过启发式算法寻找比较优化的解<sup>[10]</sup>。当前常用的启发式算法中，粒子群优化算法以其算法简单、计算方便、求解速度快受到任务调度研究者的重视<sup>[11]</sup>。下面研究如何

改进 PSO 算法以感知用户偏好差异。

### 3 适应用户 QoS 偏好的 PSO 云存储任务调度

本节先介绍 PSO 算法及其应用到云存储系统中的编码方式,进而从解空间、适应度等方面对现有云计算中的 PSO 调度算法进行改进,首先保证不产生对云存储系统来说无意义的解;其次,修改适应度函数定义来感知用户偏好。

#### 3.1 标准粒子群算法简介

PSO<sup>[12]</sup>是一种基于种群搜索策略的自适应随机优化算法。在基本粒子群算法中,生物的个体被抽象为没有质量和体积的粒子,生物群体被抽象为由  $e$  个粒子组成的群体。粒子  $i$  代表优化问题在  $D$  维搜索空间中可能的解,表示为位置矢量  $X_i = (x_1, x_2, \dots, x_N)$ ,位置可以通过飞行速度矢量更新  $V_i = (v_1, v_2, \dots, v_N)$ 。每个粒子都有一个由目标函数决定的适应值(fitness value),并且知道自己到目前为止发现的最好位置( $pbest$ )和现在的位置  $X_i$ 。这个可以看作是粒子自己的飞行经验。除此之外,每个粒子还知道到目前为止整个群体中所有粒子发现的最好位置( $gbest$ )( $gbest$  是  $pbest$  中的最好值)。这个可以看作是粒子同伴的经验。粒子就是通过自己的经验和同伴中最好的经验来决定下一步的运动。

PSO 的初始化为一群随机粒子(随机解)。然后通过迭代找到最优解。在每一次的迭代中,粒子通过跟踪 2 个“极值”( $pbest, gbest$ )来更新自己。在找到这 2 个最优值后,粒子通过下面的公式来更新自己的速度和位置。

$$V_i = V_i + c_1 \cdot rand() \cdot (pbest_i - X_i) + c_2 \cdot rand() \cdot (gbest_i - X_i) \quad (1)$$

$$X_i = X_i + V_i \quad (2)$$

在式(1)和式(2)中,  $i=1,2,\dots,m$ ,  $m$  是该群体中粒子的总数,  $V_i$  是粒子的速度;  $pbest$  和  $gbest$  如前定义;  $rand()$  是介于 (0, 1) 之间的随机数;  $X_i$  是粒子的当前位置。 $c_1$  和  $c_2$  是学习因子,通常取  $c_1=c_2=2$ 。在每一维,粒子都有一个最大限制速度  $V_{max}$ ,如果某一维的速度超过设定的  $V_{max}$ ,那么这一维的速度就被限定为  $V_{max}(V_{max}>0)$ 。

1998 年 Shi 等<sup>[13]</sup>对前面的式(2)进行了修正。引入惯性权重因子,如式(3)所示。

$$V_i = w \cdot V_i + c_1 \cdot rand() \cdot (pbest_i - X_i) + c_2 \cdot rand() \cdot (gbest_i - X_i) \quad (3)$$

$$X_i = X_i + V_i \quad (4)$$

其中,  $w$  为非负,称为惯性因子。式(3)和式(4)被视为标准 PSO 算法。 $w$  值较大,全局寻优能力强,局部寻优能力弱;  $w$  值较小,反之  $w$  值较大。

#### 3.2 编码策略

很显然以上算法适合求解实数问题,但对离散化的任务调度不能直接应用。需要进行转化,一般方法是进行编码。对云存储系统来说,一个编码就代表一个任务调度序列。出于描述方便采用十进制编码,假设有  $m$  个资源,  $n$  个任务,用  $Code=e_1e_2\cdots e_i\cdots e_n$  表示任务调度向量,即一个调度方案。对云存储系统来说,  $e_i$  代表第  $i$  个任务的数据由  $e_i$  的值代表的资源节点提供,而该向量的长度即是单位时间内要调度任务的总量  $n$ 。例如,某任务调度向量为 [2, 4, 3, 4, 1, 5, 6], 这个向量的长度是 7,代表该单位时间需要调度的任务数为 7。序号 1 的位置上的值是 2,代表任务 1 的数据由系统的 2 号节点提供。以此类推,任务 2 和 4 的数据由节点 4 提供;任务 3 的数据由节点 3 提供;任务 5 的数据由节点 1 提供;任务 6 的数据由节点 5 提供;任务 7 的数据由节点 6 提供。

#### 3.3 限制解空间

传统云计算领域的 PSO 算法,初始解和解的更新都是整个解空间,即云计算任务可以调度到任何一个节点计算。但是如引言中所述,在云存储系统中,由于任务资源只在特定节点存储,因此任务不能被调度到任意节点执行,即云存储中的解必然是受限制的。

针对这个问题,引入存在矩阵(EM, exist matrix)<sup>[14]</sup>来指示任务和资源节点间的对应关系。 $EM$  是一个  $n \times m$  的矩阵,  $n$  是调度任务数目,  $m$  是系统资源节点数目。矩阵的项  $e_{ij}$  代表节点  $j$  是否有  $i$  任务的数据,有则为 1,无则为 0。该矩阵可以从云存储系统的资源列表中生成。资源列表记录了数据块 ID 和存放的资源节点的对应关系,为了保证数据的安全,数据块往往被冗余存储在多个资源节点上,因此,存在矩阵中代表任务的一行中会存在多个 1 的情况,对应的列数就是该拥有任务数据的节点 ID。

在存在矩阵的限制下,对传统云计算中基于 PSO 的任务调度算法做以下 3 点改进。

1) PSO 的初始化解不再随机产生,而是根据存在矩阵生成;生成算法简单来说是从  $EM$  中获取

每行值为 1 的序号, 然后从这些序号中随机选择一个作为该行代表任务的解,  $n$  个任务的解构成一个任务调度解  $V$ 。

2) 在根据式(3)和式(4)产生新的解时, 先不进行适应度的计算和比较, 先进行存在性检测, 即产生的新解是否符合存在矩阵, 如果不符合则另外产生一个解直到满足存在矩阵, 之后再进入适应性检测。

3) 在算法陷入局部最优时, 同样从存在矩阵  $EM$  中产生新的解, 而不是像传统算法一样随机在整个解空间产生新解。

### 3.4 用户 QoS 偏好适应

从前面标准 PSO 流程可以看出, 决定 PSO 解优劣程度的是其目标函数和由目标函数决定的适应度。为了适应不同的用户 QoS 需求的差异, 下面对 PSO 的适应度定义进行改进, 加入对云存储系统非常重要的 QoS 因素, 并对这些因素进行分类, 用权重因子来指示这些因素的重要程度, 即可以通过权重因子的调节来达到适应不同用户不同 QoS 偏好的作用。

#### 3.4.1 云存储中的 QoS 因素归纳与效用函数定义

##### 1) 云存储中常见的 QoS 因素

适应度函数  $F$  由多个约束因素决定, 这些约束因素就是决定任务调度 QoS 的因素。根据现有研究<sup>[1~9,15,16]</sup>和自身经验, 总结出以下主要影响因素。

$s_i$  代表任务  $i$  所需要传输的数据量的大小(size, 单位: Megabyte)。

$b_{ij}$  代表任务  $i$  的请求者和资源  $j$  之间的带宽(bandwidth, 单位: Mbit/s, 如果有多个带宽值取最小值)。

以上二者的商其实就是任务  $i$  由资源  $j$  提供时要耗费的传输时间  $t_{ij}$ 。无论对云计算还是云存储系统。一般时间以秒为单位。

$c_j$  代表使用资源  $j$  所花费的代价, 例如, 使用服务器的代价明显和使用一般的主机是不同的, 中心节点使用的代价和边缘节点使用代价也肯定是有差异的。各个节点的代价构成了代价向量  $C = [c_1, c_2, \dots, c_m]$ 。

$l_j$  代表资源  $j$  在本调度周期前的负载 (CPU 负载、内存负载、数据存储负载等或者是综合)。各个节点的负载构成了负载向量  $L = [l_1, l_2, \dots, l_m]$ 。

$d_{ij}$  代表任务  $i$  的请求者和资源  $j$  之间的网络延迟 (可以通过历史数据预测, 包括等待时间、预热

时间等), 一般来说以秒为单位。各个节点的延迟量构成了代价向量  $D = [d_1, d_2, \dots, d_m]$ 。

$q_j$  代表资源  $j$  的质量, 对于云存储系统来说, 主要考虑该资源节点的故障率, 各个节点的质量评价构成了质量向量  $Q = [q_1, q_2, \dots, q_m]$ 。

$IO_j$  代表资源  $j$  的 IO 并发数, 该数目有上限, 超过的话, 服务器对新到的请求会拒绝。

##### 2) QoS 效用函数

由于各个 QoS 要求的物理意义不完全相同, 计量单位也不一定相同, 从而使得 QoS 值的量纲和数量级可能不同。需要利用 QoS 效用函数将每个候选资源节点的 QoS 属性映射到一个实数值, 通过该值对每个候选节点进行评估和比较, 以便选择到满足 QoS 约束的资源节点。

本文采用的效用函数是归一化函数, 其构造方法是, 将候选节点某 QoS 属性与其对应的最大值或最小值进行比较, 从而将多个 QoS 属性值进行归一化处理 (范围在 0~1), 使其转化到一个综合衡量的实数值 (独立于每个具体属性的单位或范围), 如下所示。

当  $q_i$  是效益型属性, 效益型的属性值越大, 表明属性质量越优。

$$q_i = \begin{cases} (q_i - \min q_i) / (\max q_i - \min q_i), & \max q_i \neq \min q_i \\ 1, & \max q_i = \min q_i \end{cases} \quad (5)$$

当  $q_i$  是成本型属性, 成本型的属性值越小, 表明属性质量越优。

$$q_i = \begin{cases} 1, & \max q_i = \min q_i \\ (\max q_i - q_i) / (\max q_i - \min q_i), & \max q_i \neq \min q_i \end{cases} \quad (6)$$

其中,  $\min q_i$  和  $\max q_i$  分别表示在服务组 QoS 属性  $q$  的最小值和最大值。

#### 3.4.2 QoS 偏好感知的适应度定义

定义以下的适应度函数  $f_{ij}$ , 代表任务  $i$  由资源  $j$  执行的适应度为

$$f_{ij} = \lambda_t \delta_t + \lambda_c \delta_c + \lambda_r \delta_r \quad (7)$$

其中,  $\lambda_t + \lambda_c + \lambda_r = 1$ 。

因素的性质各有不同, 大致包括时间、代价、和质量 3 大方面。例如, 文件大小  $s_i$  除以带宽  $b_{ij}$  就是时间; 费用  $c_j$ 、负载  $l_j$  属于代价衡量范围, 而网络延迟  $d_{ij}$ 、网络节点的质量  $q_j$ 、服务器 IO 数是否充足可以看作是服务质量衡量范畴。因此, 适应度函数可以被化为 3 大部分, 由各自的因子协调重要性,  $\lambda_t$ 、 $\lambda_c$  和  $\lambda_r$  分别被称为时间因子、代价因子

和可靠性因子。它们值的大小反映了对时间、代价和质量的重视程度。

代价、时间与质量是 3 类 QoS 要求, 每类包含多个技术性 QoS 指标。设  $c_i$  类中有  $n$  个指标  $index$ , 这些指标都经过归一化处理。节点  $j$  的  $c_i$  类指标为指标和的效用值, 用  $w^i$  表示本类各因素的权重调节各因素的重要性 (专业性评价), 且  $w^i \in [0,1]$ ,  $\sum_i w^i = 1$ 。

$$c_i = w^1 index_i^1 + w^2 index_i^2 + \dots + w^n index_i^n \quad (8)$$

该权重和适应度权重 ( $\lambda_t$ ,  $\lambda_c$  和  $\lambda_r$ ) 设定本文采用优序判定法, 减少主观判断。首先构建判断尺度, 重要程度判断尺度用 1、2、3、4、5 五级表示, 数字越大, 表明重要性越大。当 2 个属性对比时, 如果一个属性重要性为 5, 则另一属性重要性为 0; 如果一个属性为 3, 则另一个属性为 2。

据此构建判断矩阵, 矩阵为  $n \times n$  的方阵,  $n$  为待评判属性数。第  $i$  行  $j$  列值  $w_{ij}$  即是属性  $i$  与  $j$  相比的重要性, 例如,  $w_{ij} = 3$ , 则对应的  $j$  与  $i$  相比的重要性为  $w_{ji} = 5 - 3 = 2$ ,  $i \neq j$ 。行之和  $\sum_i w_i$  为属性  $i$  在整个属性中的重要性。以全部属性重要性之和  $\sum_j \sum_i w_{ij}$  为分母, 属性重要性为分子, 两者之商即是该属性的权重。

$$W_i = (\sum_i w_i) / (\sum_j \sum_i w_{ij}) \quad (9)$$

该适应度函数定义具有以下优点。

1) 具有很好的可扩展性: 本文所总结的因素仅仅是针对云存储系统而言比较重要的一部分。在计算网格或其他系统中, 还会有其他一些因素。当需要的时候, 这些因素可以融入已有框架中, 加入时间、代价或可靠性部分。需要的时候现有的因子也可以被剔除。即仅仅需要根据实际情况对适应度函数  $f$  做相应修改, 其他部分不变, 就可以满足不同系统的需要。

2) 具有偏好适应性: 可以根据各因子的调整, 满足不同偏好的要求。例如, 某些用户对数据的传输时间并不看重, 它们宁愿用更多的等待时间来换取更低的服务费用。对这类任务调高代价因子、调低时间因子能获得更加有针对性的调度方案。

### 3.4.3 用户 QoS 偏好感知算法流程

#### 1) 任务优先度决定因子设定值

前文提到 QoS 可以归为 3 类, 任务对 QoS

偏好的组合, 根据对用户的调查 (主要是询问了本校网络专业 10 级的学生) 主要可以分为以下 5 类。

最高级为同时看重质量和时间的任务, 如此高要求代价上就不能要求太多, 等级标记为 5 (组合中三方面同时要求的情况实际中不允许, 而学生们对得到高效有质量保证的服务必须付出的代价也有共识)。其次, 对质量、时间、代价其一做要求, 其中, 由于时间要求比较紧迫, 设定等级为 4; 要求质量的等级设定为 3; 仅仅对代价有要求的设定为 2 (要求少花钱, 那么对效率和质量就不能有太高要求, 这类划分也受到被询问学生的赞同)。最后, 没有要求的被设置为等级 1。

用 1~5 代表不同的程度, 根据等级设定时间因子、代价因子和可靠性因子的取值。例如, 设定等级 5 代表用户希望任务耗时短、可靠性高、不计代价, 那么时间和可靠性因子则设为高值, 代价因子设为低值; 等级 2 则代表用户希望代价越低越好, 不计较消耗的时间和可靠性, 对应时间和可靠性因子则设为低值, 代价因子设为高值。等级 1 比较特殊, 没有要求, 这里视为对代价有最高要求。不同于等级 2, 虽然对代价看重但对时间和质量还有一定的期待。等级 1 代价取 1, 其他因素取 0。

对于因子的具体取值, 本文采用优序对比法进行设定, 能够一定程度上减弱用户和系统管理员的主观影响, 具体如下。

以任务优先级为 5 时的各因子值设定优序对比为例。

表 1 基于优序判定法的因子值设定 (当任务优先级为 5 时)

因子类型	时间	代价	可靠性	合计	最终权数
时间	—	5	2	7	0.47
代价	0	—	0	0	0
可靠性	3	5	—	8	0.53
总计	—	—	—	15	1

当任务为其他优先级时的设定方法以此类推, 最终的设定值如表 2 所示。

表 2 任务优先级对应的各因子取值

优先级	1	2	3	4	5
时间因子值	0	0.27	0.33	0.40	0.47
代价因子值	1	0.40	0.27	0.13	0
可靠性因子值	0	0.33	0.40	0.47	0.53

于是，一个等待调度的任务实际上包含 2 个维度的数据（数据量大小  $s$ ，任务优先级  $p$ ）。计算适应度时，根据  $p$  取 3 个因子的值。

而代价向量  $C = [c_1, c_2, \dots, c_m]$  是一个预定义的向量，序号代表系统节点的编号，即  $c_1$  代表节点 1 的使用代价。代价的程度值是系统的管理人员根据节点本身的配置花费、能耗花费等评估而得。

适应度计算还需要  $l$ 、 $d$ 、 $q$  等值，可以实时从系统节点日志记录中获取，与代价类似构成负载向量  $L$ 、延迟向量  $D$ 、质量向量  $Q$  等。再根据定义和式(5)、式(6)计算需要的归一化值。本文的实验系统设计并实现有专门记录和计算这些值的模块，随着调度的进行，各项值会按当前值与记录的最大最小值进行更新。

以上的计算是针对某一个确定解的，即任务在某个具体的节点执行已经确定，则以上各项值都可以获取到。最后比较所得所有解的适应度值，取最小适应度对应的解为最终确定解。

### 2) 算法流程

依照以上定义和改进，适应用户 QoS 偏好的 PSO 云存储任务调度算法流程示意如图 1 所示。限于篇幅标准 PSO 流程省略，仅仅突出了改进之处。

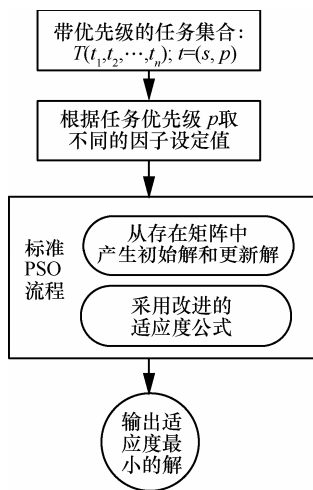


图 1 带 QoS 偏好感知的 PSO 任务调度

## 4 实验与分析

为了分析本文所提偏好感知任务调度算法的有效性，设计并用 MATLAB 实现了一个云存储偏好感知仿真平台，模拟云存储从提交任务请求到调度指定节点提供任务资源的过程。仿真平台由 3 大模块组成，其中，模拟条件生成模块负责模拟云存

储系统的各种条件，包括网络的带宽、网络线路的延迟、节点的负载、节点故障、节点的 IO 并发数等；资源定位模块负责根据任务需求和系统本身的资源记录表实时生成资源存在矩阵  $EM$ ；任务调度模块则负责根据本文的算法对任务进行调度。

实验统一设置最大迭代次数为 100 次，为避免干扰导致的某些个别情况，每个实验重复进行 10 次，每次实验记录“运行时间”、“迭代次数”等参数以待对比。运行时间就是指算法从开始接受输入需要调度的任务向量开始，到最后输出一个可以接受的调度序列的时间。迭代次数记录的是最后输出的调度序列是第几次迭代产生的，这个值可以考察算法的收敛速度。偏好适应率则是考察生成的调度序列多大程度上满足了用户的偏好。

下面就传统 PSO 算法和改进后的带偏好感知的 PSO 算法对偏好感知的具体效果进行考察（都有存在矩阵限制）。随机生成的  $Task$  向量自带对 QoS 因素的要求（代价、时间、质量），对应的值在 0~1 之间。调度结果提供节点的对应值大于或等于要求则视为满足了用户 QoS 要求，反之视为不满足。用一个偏好评价函数进行自动检查，给出任务偏好满意百分比，计算方法如下

$$\eta = \frac{\text{满足偏好的任务数}}{\text{任务总数}} \% \quad (10)$$

需要说明的是，IO 并发数对质量的影响，已有并发数多并不一定就比已有并发数少的服务器提供更差的服务，这点在实验时表现出对调度评价的影响并不符合实际。更多情况下要考虑剩余并发数的多少。因此，将 IO 改为剩余并发数，即从成本型改为效益型，值越大越好。调度模块为某节点调度一个任务后，更新节点的各项值，对 IO 来说减少了一个并发数。

对比结果如表 3 所示，可以看到，传统 PSO 对有偏好要求的任务是完全没办法满足的，平均只有 22% 的满足率，全凭运气。而本文改进的 PSO 算法偏好满足率上升 40% 达到了 60%。说明改进后的 PSO 算法确实带有一定的偏好感知能力。但是距离预计 80% 以上的偏好满足率还有一定差距。

开始以为是因子设置问题，但是无论怎么调整因子的值，实验的偏好满足率都只有约 50%~70%。后来发现，问题不在于因子值，以上优序判定的因子已经比较合适。真正的原因是各个等级任务所占百分比。实验模拟是一般的金字塔型的任务分布，

即高等级的任务比例比较小，大量的低等级任务的情况，中间等级的任务中等数量，这也是一般系统的任务分布。第 3.4.2 节设计的感知偏好的适应度具有对单个任务 QoS 感知识别的能力。但是 PSO 算法的选择是对总体 QoS 的评估。于是出现了占 80% 以上的优先级只有 1、2 的任务的偏好左右了总体 QoS 偏好，导致整体偏好满意率不高。这 50%~70% 被满足的任务大部分是占有率高的低等级任务。这种情况在笔者调整了各个等级任务的分布百分比后有所改善，特别当各等级任务所占百分比相同的情况下，占有率对整体偏好影响达到最小，偏好满足率基本在 80% 以上。但是这种情况在现实中是很少的。大量存在的其实是表 3 代表的金字塔型的任务分布。因此，这次实验说明对于偏好不同且分布相差大的任务调度其实不适合用 PSO 算法进行整体调度。

表 3 带偏好感知的 PSO 云存储调度算法

实验次数	传统 PSO 偏好满足率/%	改进 PSO 偏好满足率/%
1	31	52
2	25	61
3	21	71
4	52	63
5	10	59
6	14	52
7	11	63
8	33	51
9	13	72
10	12	63
平均	≈22	≈60

最后，笔者把这些任务按优先级进行分级调度（按优先级先后顺序调度），即将 3.4.3 节算法里的

“任务矩阵”按优先级先后输入，高优先级完成后，再进行低优先级的调度，其余不变，得到了 90%~100% 的偏好满足率如表 4 所示。虽然总体满意率较高，但是优先级低的满意率居然高于优先级高的。分析发现原因在于：由于是按优先级从高到底顺序调度，优秀的资源优先调度给了优先级较高的任务。但是资源总是比任务少，高优先级的任务意味着高资源要求，在总体资源受限的情况下，无法完全满足高优先级任务。而低优先级任务唯一的的就是代价要求，因此在其他任务将代价高的资源先占用的情况下反而可以保证其代价低的要求，因而其满足率最高。

表 4 分级的带偏好感知的 PSO 云存储调度算法

实验次数	优先级 5 的任务 偏好满足率/%	优先级 4 的任务 偏好满足率/%	优先级 3 的任务 偏好满足率/%	优先级 2 的任务 偏好满足率/%	优先级 1 的任务 偏好满足率/%
1	90	94	85	89	100
2	88	92	90	91	100
3	96	88	91	95	100
4	92	96	95	96	100
5	90	95	96	92	100
6	93	90	92	97	100
7	91	91	92	96	100
8	88	95	94	97	100
9	95	88	97	95	100
10	99	87	96	95	100
平均	92.2	91.6	92.4	94.3	100

因而算法流程改变为图 2 所示按优先级先后调度，调度算法内部流程不变，输入改为按任务优先级顺序输入。不建议并发调度，因为并发过程优秀

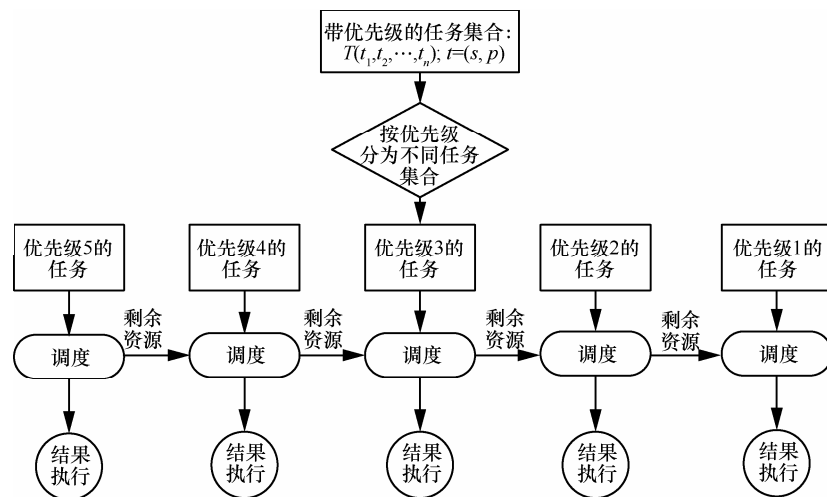


图 2 分级的带偏好感知的 PSO 调度

资源可能被低级任务占用。

## 5 结束语

本文对云存储任务调度进行建模, 总结并归纳出云存储任务 QoS 要求主要有 3 个方面, 即时间、代价和质量, 每个方面又包含若干具体因素。修改 PSO 算法的适应度定义, 用权重因子调节对不同 QoS 偏好的要求。实验表明, 通过对适应度的修改, 确实可以使得 PSO 调度具备一定的偏好感知能力。但是, 由于 PSO 是整体评价算法, 则在各优先级任务分布差异较大的情况下, 占有率较高的任务的 QoS 偏好会掩盖其他占有率较低的任务的偏好, 影响这些任务的偏好满足率。解决方案是按优先级分别调度这些任务。

总体来说, 本文最大的发现是对于任务分布不均的系统, 不适宜用 PSO 进行有偏好感知的整体调度, 一定要进行分级的调度。本文将 PSO 算法引入云存储领域并进行偏好感知的一些经验, 特别是不成功的经验希望能给后来者以参考。

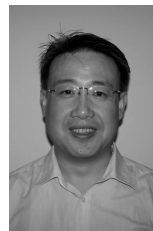
## 参考文献:

- [1] HAYES B. Cloud computing[J]. Communications of the ACM, 2008, 51(7):9-11.
- [2] LIN G, DASMALCHI G, ZHU J. Cloud computing and IT as a service: opportunities and challenges[A]. Proceedings of 2008 IEEE International Conference on Web Services[C]. Beijing, China, 2008.5.
- [3] NAMJOSHI J, GUPTE A. Service oriented architecture for cloud based travel reservation software as a service[A]. Proceedings of the 2009 IEEE International Conference on Cloud Computing(CLOUD'09)[C]. Bangalore, India, 2009.147-150.
- [4] VAZHKUDAI S, TUECKE S, FOSTER I. Replica selection in the globus data grid[A]. Proceedings the First IEEE/ACM International Symposium on Cluster Computing and the Grid[C]. Brisbane, Qld, 2001. 106-113.
- [5] VAZHKUDAI S, SCHOPF J M, FOSTER I. Predicting the performance of wide area data transfers[A]. Proceedings of International Parallel and Distributed Processing Symposium, Marriott Marina[C]. Fort Lauderdale, FL, USA, 2002.34-43.
- [6] CHANG R S, CHEN P H. Complete and fragmented replica selection and retrieval in data grids[J]. Future Generation Computer Systems, 2007, 23(4): 536-546.
- [7] RAHMAN R M, ALHAJJ R, BARKER K. Replica selection strategies in data grid[J]. Journal of Parallel and Distributed Computing, 2008, 68(12): 1561-1574.
- [8] JIN J, ROTHROCK L, MCDERMOTT P L, *et al.* Using the analytic hierarchy process to examine judgment consistency in a complex multi-attribute task[J]. IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans, 2010, 40(5): 1105-1115.
- [9] 熊润群, 罗军舟, 宋爱波等. 云计算环境下 QoS 偏好感知的副本选择策略[J]. 通信学报, 2011, 32(7):94-102.
- XIONG R Q, LUO J Z, SONG A B, *et al.* QoS preference-aware replica selection strategy in cloud computing[J]. Journal on Communications, 2011, 32(7):94-102.
- [10] MA T H, YAN Q Q, LIU W J, *et al.* Grid task scheduling: algorithm review[J]. The Institution of Electronics and Telecommunication Engineers, 2011, 28(2):158-167.
- [11] CHEN R M, WANG C M. Project scheduling heuristics-based standard PSO for task-resource assignment in heterogeneous grid[J]. Hindawi Publishing Corporation Abstract and Applied Analysis, 2011, 2011:1-20.
- [12] KENNEDY J, EBERHART R C. Particle swarm optimization[A]. Proceedings of the IEEE International Conference on Neural Networks, Path[C]. Australia, 1995.1942-1948.
- [13] SHI Y H, RUSSELL E B. A modified particle swarm optimizer[A]. Proceedings of IEEE International Conference on Evolutionary Computation[C]. Anchorage, AK, 1998.69-73.
- [14] 王娟, 李飞, 张路桥. 限制解空间的 PSO 云存储任务调度算法[J]. 计算机应用研究, 2013, 30(1):127-129.
- WANG J, LI F, ZHANG L Q. Task scheduling algorithm in cloud storage system using PSO with limited solution domain[J]. Application Research of Computers, 2013, 30(1):127-129.
- [15] HE X S, SUN X H, GREGOR V L. QoS guided min-min heuristic for grid task scheduling[J]. Journal of Computer Science and Technology, 2003, 18(4):442-451.
- [16] CHAUHAN S S, JOSHI R C. QoS guided heuristic algorithms for grid task scheduling[J]. International Journal of Computer Applications, 2010, 2(9):24-31.

## 作者简介:



王娟 (1981-), 女, 四川崇州人, 博士, 成都信息工程学院副教授, 主要研究方向为云计算、物联网技术、复杂网络。



李飞 (1966-), 男, 湖南常德人, 成都信息工程学院教授, 主要研究方向为网格计算和物联网技术研究。



张路桥 (1982-), 男, 四川自贡人, 博士, 成都信息工程学院副教授, 主要研究方向为无线网络、云计算。