

# 云环境下基于节能和负载均衡的混沌粒子群资源优化调度

何丹丹

(大连科技学院 信息科学系, 辽宁 大连 116052)

**摘要:** 针对传统云计算资源调度方法仅关注任务的最大完成时间, 没有考虑到节能和资源负载均衡的问题, 提出了一种基于混沌粒子群算法实现云资源优化调度的方法: 首先, 定义了以节能和负载均衡为目标的多目标数学模型, 然后设计了一组靠近最优 Pareto 前沿的解作为初始种群, 采用改进的粒子群算法来搜索最优调度方案, 当最优解连续两代未发生变化时, 通过混沌遍历法对粒子进行局部寻优, 以加快获取全局最优解; 在 CloudSim 仿真环境下结合 Matlab 工具进行实验, 结果表明: 文中方法负载均衡离差平均值为 0.156, 且较其它方法, 具有较好的负载均衡能力和较低的能耗, 具有很强的可行性。

**关键词:** 资源调度; 粒子; 云计算; 混沌搜索

## Chaos Particle Swarm Optimizing Scheduling Based on Power-aware and Load Balance in Cloud Computing

He Dandan

(Department of Information Science, Dalian Institute of Science and Technology, Dalian 116052, China)

**Abstract:** Aiming at the traditional cloud computing resource scheduling method only considering the latest finishing time of task, concerning the power-aware and resource load balance, a method based on chaos particle swarm was proposed. Firstly, the mathematical model based on energy-saving and load balance was built, then a set of solution near Pareto front as the initial population, the improved particle swarm algorithm was used to search the optimal scheduling scheme. When the optimal solution was not changed for two iterations, the chaos search was used to search to get the global optimum solution. The simulation in the CloudSim and Matlab environment shows the method in this paper has the average load balance distance as 0.156, and compared with the other methods, it has the good load balance performance and low energy consumption. It has strong feasibility.

**Key words:** resource scheduling; particle swarm; cloud computing; chaos search

### 0 引言

美国国家标准技术研究院 (NIST) 将云计算定义为<sup>[1-2]</sup>: 云计算是通过互联网按需访问的可配置的计算资源池, 具备按需自助服务、网络访问广泛、资源共享方便和快速弹性能力等特征。

资源调度是云计算研究领域的重点问题, 其与传统的网格计算相比, 具有很多不同之处<sup>[3]</sup>, 因此, 传统的针对网格计算的资源调度模型<sup>[4-5]</sup>如 Min-Min 算法和 Max-Min 算法等算法往往以任务的最大完成时间作为目标进行资源调度, 导致各物理资源的负载不均衡, 同时没有考虑到节能问题<sup>[6]</sup>。

因此, 如何在云计算环境下进行合理的资源调度, 不仅能实现云计算各服务器的负载均衡, 同时又能最大化地减少能量消耗是云计算资源调度的一个目标<sup>[7]</sup>。

但是目前国内外对云计算资源调度的研究主要集中在减少任务执行时间和提高用户服务质量上。文献 [8] 以最大化效用作为调度目标, 通过求解 CUM 优化问题得到虚拟机和物理机之间的映射关系。文献 [9] 在 Hadoop 平台的基础上设计一种资源调度算法。文献 [10] 设计了一种基于元胞自动机遗传算法的云资源调度方法。文献 [11] 设计了一种基于蚁群算法的云资源调度方法。

上述方法均针对云计算环境下的资源优化调度问题<sup>[13]</sup>, 具有重要意义。为了实现云计算环境下的节能和负载均衡, 文中设计了一种基于混沌粒子群的云资源调度方法。

### 1 云计算资源优化调度模型

云计算平台是由应用层、平台层和基础设施层构成, 用户通过应用层提交服务请求并接受最终的结果。平台层是由可重用的软件资源组成。基础设施层主要包含一些硬件设备和相应的管理软件。由于虚拟化技术的引入, 一台物理机可以被实例化为多台虚拟机, 多台物理机的剩余资源也可以被虚拟化为一台虚拟机, 此时, 物理主机可以抽象为多个虚拟机资源, 因此, 用户提交的任务可以交由虚拟机进行处理, 因此, 云计算环境下的资源调度模型不再是主机-任务两层模式, 而是在中间加上了一层虚拟机, 形成主机-虚拟机-任务的三层模式, 如图 1 所示。

### 2 云计算资源调度目标

#### 2.1 负载均衡目标函数

假设云计算环境下的  $n$  个用户提交的任务构成的任务集为  $T = \{t_1, t_2, \dots, t_n\}$ , 任务之间无依赖关系, 虚拟机集合为  $R = \{r_1, r_2, \dots, r_m\}$ , 虚拟机在同一时刻只能处理一个任务, 其它任务处于等待状态, 虚拟机是独占和非抢占式的, 即任务一旦执行就不可中断, 必须等待任务完成或失败后, 该资源才能被重新分配。

预期执行完成时间矩阵  $ETC$  (Excepted Time) 为一个  $n \times m$  的矩阵, 其中, 元素  $ECT_{ij}$  表示任务  $t_i$  在虚拟机  $r_j$  上执

收稿日期: 2013-11-27; 修回日期: 2014-01-18。

作者简介: 何丹丹 (1979-), 女, 内蒙古赤峰市人, 工学硕士, 讲师, 主要从事软件工程理论, 计算机算法方向的研究。

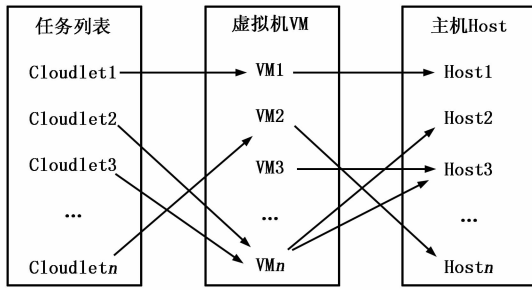


图 1 云计算资源调度模型

行所需要的时间，如下所示：

$$ECT = \begin{bmatrix} ECT_{00} & ECT_{01} & \dots & ECT_{0m} \\ ECT_{10} & ECT_{11} & \dots & ECT_{1m} \\ \vdots & \vdots & \vdots & \vdots \\ ECT_{n0} & ECT_{n1} & \dots & ECT_{nm} \end{bmatrix} \quad (1)$$

式中， $ECT_{ij}$  可以通过下式进行计算：

$$ECT_{ij} = \frac{l(i)}{u_{ij}c(j)} \quad (2)$$

式中， $l(i)$  表示任务  $t_i$  的计算量， $c(j)$  表示虚拟机的处理能力， $u_{ij}$  表示虚拟机分配给任务  $t_i$  的计算资源比率。

采用 **TA** (Task Assign, TA) 表示任务分配矩阵，由于每个任务只能分配到唯一的虚拟机上运行，因此，对于任意  $t_i$ ，仅对应唯一的  $r_j$ ，即每行仅有一个元素  $TA_{ij}$  值为 1。

$$TA = \begin{bmatrix} TA_{00} & TA_{01} & \dots & TA_{0m} \\ TA_{10} & TA_{11} & \dots & TA_{1m} \\ \vdots & \vdots & \vdots & \vdots \\ TA_{n0} & TA_{n1} & \dots & TA_{nm} \end{bmatrix} \quad (3)$$

云计算中虚拟机  $r_j$  的负载均衡因子可以通过下式计算：

$$LB_j = \frac{\sum_{i=1}^n ECT_{ij} * TA_{ij}}{\sum_{i=1}^n \sum_{j=1}^m ECT_{ij} * TA_{ij}} / \frac{l(i)}{\sum_{i=1}^n l(i)} \quad (4)$$

式 (4) 中， $\sum_{i=1}^n ECT_{ij} * TA_{ij}$  表示在虚拟机  $r_j$  上执行的所有任务的执行时间之和。

为了实现负载均衡，采用负载均衡离差来衡量虚拟资源的负载均衡程度，可以表示为：

$$\varphi = \sqrt{\frac{\sum_{j=1}^m (LB_j - \bar{LB}_j)^2}{m-1}} \quad (5)$$

式中，虚拟机  $r_j$  的负载均衡因子  $LB_j$  可通过式 (4) 获得， $\bar{LB}_j$  为虚拟机的负载均衡因子  $LB_j$  的均值。

因此，虚拟资源负载均衡度的目标函数为：

$$G_1 = \min \varphi \quad (6)$$

### 2.2 节能目标函数

节能目标即使得所有虚拟机的能耗最低，可以表示为：

$$G_2 = \min \sum_{j=1}^m (p_j^{\min} + \sum_{i=1}^n (u_{ij})(p_j^{\max} - p_j^{\min})) \quad (7)$$

式中， $p_j^{\min}$  表示虚拟机  $r_j$  的最低能耗，

即虚拟机  $r_j$  的处于空负荷工作状态的耗能， $p_j^{\max}$  表示虚拟机  $r_j$  处于满负荷工作状态的耗能， $p_j^{\min}$  和  $p_j^{\max}$  为大于 0 的常数。

## 3 基于混沌粒子群的资源优化调度

### 3.1 粒子群算法概述

粒子群算法 (PSO, particle swarm optimization) 源于对鸟群系统觅食行为研究，问题空间的每一个潜在的解都可以通过  $d$  维空间的一个点即粒子表示，粒子具有位置和速度两个特征，粒子  $i$  的位置  $P_i$  和速度  $V_i$  可以表示为式 (8)：

$$P_i = \{p_{i1}, p_{i2}, \dots, p_{iD}\} \\ V_i = \{v_{i1}, v_{i2}, \dots, v_{iD}\} \quad (8)$$

式中，粒子  $i$  的个体最优位置和全局最优位置分别为  $pbst_i$  和  $gbst_i$ ，在每轮迭代程中，粒子  $i$  通过下式对自身的位置  $x_i$  和速度  $v_i$  进行不断的更新：

$$v_i(t+1) = \omega v_i(t) + c_1 \lambda_1 (pbst_i - p_i(t)) + c_2 \lambda_2 (gbst_i - p_i(t)) \\ x_i(t+1) = x_i(t) + v_i(t+1) \quad (9)$$

式 (9) 中， $\omega$  被称为惯性权重，这。  $c_1$  和  $c_2$  为学习因子，其取值一般为 1~2 之间的随机数。 $\lambda_1$  和  $\lambda_2$  为 (0, 1) 之间的随机数。

### 3.2 粒子群初始化

云环境下  $n$  个任务到  $m$  个虚拟机的映射可以表示为一个自然数编码的向量，即采用向量来对粒子编码：

$$P = \{p_1, p_2, \dots, p_n\} \quad (10)$$

其中，任意一个任务  $t_i$  对应的虚拟机即为  $p_i$  对应的虚拟机。

由于文中方法对应的 2 个目标的优化，通常较高的负载均衡度对应了较多的能耗，因此，只能寻找其 Pareto 最优解，在保持负载均衡度的同时降低能耗。

经典粒子群算法采用随机方法初始化种群，可能导致初始解分布在远离 Pareto 前沿，从而降低算法的执行效率，因此，为了使得初始解更贴近 Pareto 前沿并减少搜索区域和提高算法效率，文中采用的初始解为高能耗和低负载均衡性能，如下所示：

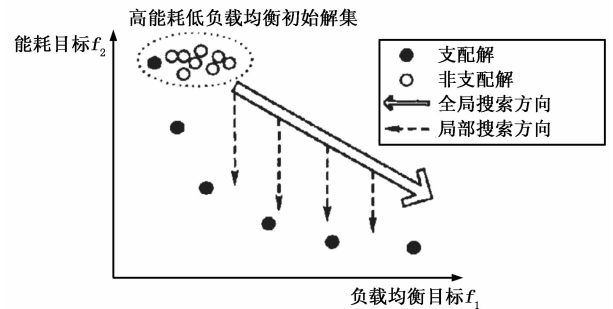


图 2 种群初始化解集

### 3.3 动态惯性权重因子

粒子运动过程中对速度的更新依赖惯性因子  $\omega$  的大小，当  $\omega$  值较大时，粒子具有较强的局部搜索能力，在当前位置邻域内进行搜索，若  $\omega$  较小，则算法的全局寻优能力较强，可以增加解的多样性。因此，设计动态变化的权重计算公式，如式 (10) 所示：

$$\omega = \begin{cases} \omega_{\min} - \frac{(\omega_{\max} - \omega_{\min}) \cdot (f - f_{\min})}{f_{\text{avg}} - f_{\min}} & f \leq f_{\text{avg}} \\ \omega_{\max} & f > f_{\text{avg}} \end{cases}$$

(11) 式 (10) 中， $\omega_{\max}$  和  $\omega_{\min}$  为惯性权值的最大值和最小值， $f_{\text{avg}}$  和  $f_{\min}$  表示所有粒子的平均适应度和最小适应度。

### 3.4 混沌粒子群局部寻优

当全局最优解在连续 2 次迭代过程不发生变化时, 加入 Tent 映射作为混沌扰动<sup>[12]</sup>, 使得粒子往低能耗和高负载均衡的方向发展, 假设最优解对应的粒子位置为  $x_i$ , 则对  $x_i$  的每维分量  $x_{ik} (e_k \leq x_{ik} \leq f_k)$  通过如下步骤进行混沌扰动:

(1) 将  $x_i$  通过下式映射到区间 (0, 1) 上, 得到  $y_{ik}$ :

$$y_{ik} = \frac{x_{ik} - e_k}{f_k - e_k} \quad (12)$$

(2) 对  $y_{ik}$  通过下式进行混沌扰动, 得到  $z_{ik}$ :

$$z_{ik} = \begin{cases} 2y_{ik} & 0 \leq y_{ik} \leq 1/2 \\ 2(1 - y_{ik}) & 1/2 \leq y_{ik} \leq 1 \end{cases} \quad (13)$$

(3) 对将  $z_{ik}$  通过下式重新映射到原空间:

$$x_{ik} = a_k + z_{ik}(b_k - a_k) \quad (14)$$

## 4 资源优化调度算法描述

初始化: 粒子群规模  $M$ , 惯性权重因子的最大值  $\omega_{\max}$ 、最小值  $\omega_{\min}$ , 学习系数  $c_1$  和  $c_2$ , 最优值未变化的迭代次数  $u$ , 迭代次数的最大值  $T$ ;

(1) 最优值未变化的迭代次数  $u = 0$ , 当前迭代次数  $t = 1$ ;

(2) 根据图 2 产生种群为  $M$  高能耗低负载均衡度的初始种群;

(3) 对于粒子群中的每个粒子  $i$ , 根据公式 (6) 和 (7) 计算粒子的适应度:

如果粒子  $i$  的适应度大于其历史最优适应度, 则对当前粒子的个体最优值  $pbest_i$  进行更新;

如果粒子  $i$  的适应度大于所有粒子的历史最优适应度  $gbst_i$ , 则对所有粒子的全局最优值  $gbst_i$  进行更新,  $u = u + 1$ 。

(4) 判断  $u \geq 2$  是否成立:

如果成立, 则对粒子采用 3.4 节的混沌遍历法进行局部搜索;

(5) 根据公式 (9) 对粒子的速度和位置进行更新;

(6) 当前迭代次数  $t = t + 1$  并判断是否达到最大值  $T$ , 如果达到最大值, 则算法结束, 输入全局最优解  $gbest_i$  作为最终的资源调度方案, 否则转入步骤 (3) 继续执行;

## 5 仿真实验

采用仿真工具 CloudSim 来模拟实验环境, 并采用 Matlab 进行仿真结果输出, 实验主要参数设置如下: 任务个数为 0~500 个, 虚拟机个数为 0~50 个, 粒子种群规模为 50, 惯性权重因子的最大值  $\omega_{\max} = 0.9$ 、最小值  $\omega_{\min} = 0.3$ , 学习系数  $c_1 = 0.91$ 、 $c_2 = 0.78$ , 迭代次数最大值  $T = 100$ 。

为了验证文中方法的有效性, 从能耗和负载均衡度两个方面来进行验证, 并与文献 [10] 基于元胞自动机的方法进行比较, 文献 [10] 方法参数设置如下: 种群规模为 50, 交叉概率 0.65, 变异概率 0.58, 迭代次数最大值  $T = 100$ , 采用两种方法得到的能耗和负载均衡离差分别如图 3 和图 4 所示:

从图 3 中可以看出, 文中方法的虚拟机能耗在仿真期间一直低于文献 [10] 对应的方法, 文中方法的平均能耗为 262.8, 文献 [10] 方法的平均能耗为 306.2, 平均越低 14.17%, 这是因为文中方法考虑虚拟机节能因素, 因此, 具有较低的能耗消耗。

从图 4 中可以看出, 文中方法的负载均衡离差较低, 平均值为 0.156, 文献 [10] 方法对应的负载均衡离差平均值为

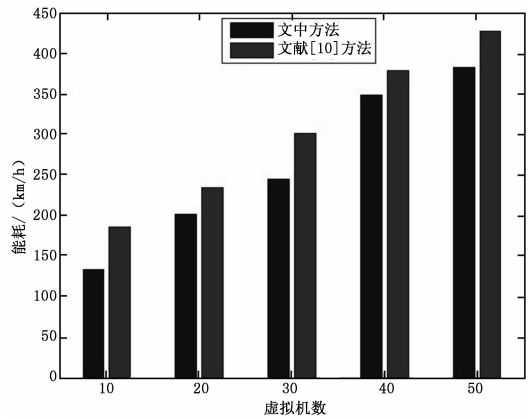


图 3 算法能耗比较

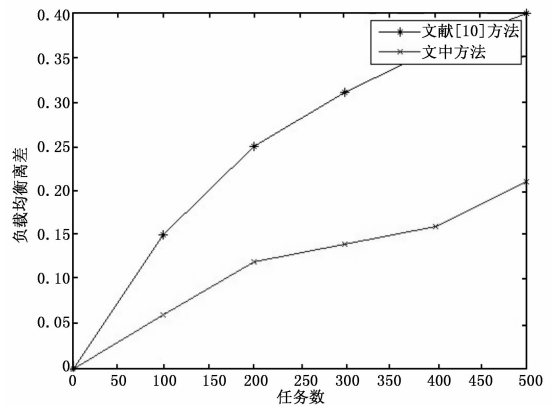


图 4 负载均衡离差比较

0.294, 显然, 文中方法的负载均衡程度远远高于文献 [10] 方法, 平均约高 53.1%, 因此, 文中方法更优。

## 6 结语

为了实现云计算环境下的资源优化调度, 在实现节能的同时具有较好的负载均衡性能, 设计了一种基于混沌粒子群算法的云环境虚拟资源调度方法, 以能耗目标和负载均衡目标作为双目标, 通过粒子在空间中的全局搜索和局部搜索, 寻求最优的 Pareto 最优解。通过 CloudSim 工具进行仿真实验, 实验结果证明文中方法实现了节能调度和资源负载均衡, 具有良好的收敛性能, 是一种适合云计算环境的有效调度方法。

### 参考文献:

- [1] Von L G, Wang L, Younge A J, et al. Power-Aware Scheduling of Virtual machines in DVFS-enabled clusters [A]. Proc. of IEEE international conference on cluster computing 2009 [C], New Orleans, LA, USA, 2009: 1-10.
- [2] 林伟伟, 齐德昱. 云计算资源调度研究综述 [J]. 计算机科学, 2012, 10 (39): 1-7.
- [3] 孙大为, 常桂然, 李风云, 等. 一种基于免疫克隆的偏好多维 QoS 云资源调度优化算法 [J]. 电子学报, 2011, 8 (30): 1824-1831.
- [4] Maheswaran M. Dynamic Mapping of a Class of Independent Tasks onto Heterogeneous Computing Systems [A]. Proc. of the 8th IEEE Heterogeneous Computing Workshop [C], IEEE Press, 1999.

(下转第 1631 页)

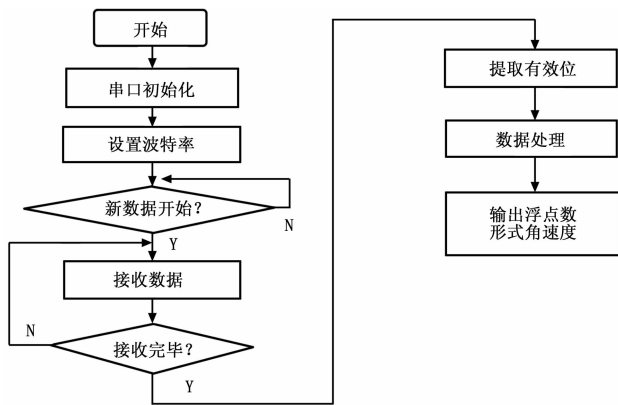


图 4 光纤陀螺数采及处理软件流程图

表 3 实验数据抽样结果

时间 (s)	姿态角估计值 (deg)	陀螺常值漂移估计值 (deg/s)
20	5.012 359	0.000 969
40	5.124 270	-5.90e-05
60	5.006 294	5.70e-05
80	5.192 151	-0.000 396
100	4.592 440	0.001 695
120	5.250 754	-0.000 462
140	5.099 051	0.000 977
160	4.951 328	0.000 433
180	5.002 199	6.90e-05
200	5.179 318	-4.90e-05

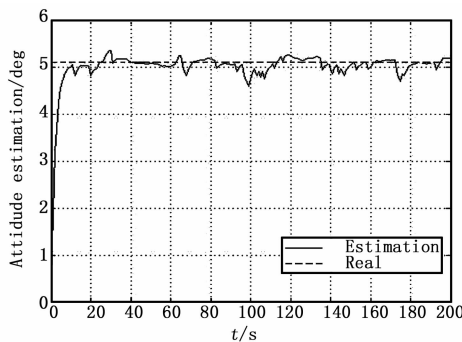


图 5 姿态估计曲线

通过实验结果, 可以得到如下的结论:

(1) 基于太阳传感器和光纤陀螺的姿态确定实验系统的设计是可行的, 姿态确定算法有效, 达到了预期的效果;

(2) 物理仿真姿态确定精度优于  $0.5^\circ$ 。陀螺常值漂移估计的平均值为  $3.625\ 758e-4^\circ/s$ 。

(3) 从实验结果可以看出, 实验还存在误差。误差来源一方面是由于太阳传感器对实验环境要求较高, 杂光源对于太阳传感器的测量精度会产生较大影响; 陀螺漂移修正后也会存在误差, 算法也会产生计算误差等。

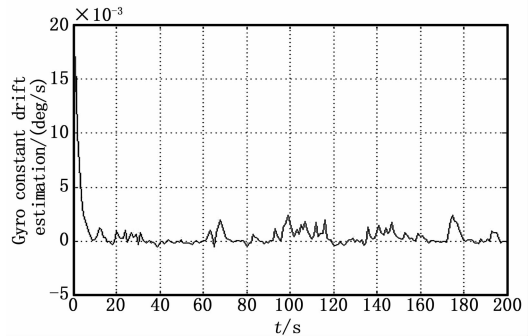


图 6 陀螺常值漂移估计曲线

### 5 结论

设计和实现了卫星组合姿态确定物理仿真系统。应用了扩展卡尔曼滤波定姿算法, 采用光纤陀螺和太阳传感器作为组合定姿传感器, 控制器为 DSP, 实验平台为单轴气浮转台。物理仿真实验结果说明, 该项系统设计是可行的, 能够实现太阳传感器和陀螺的优势互补, 提高了定姿精度, 该项技术对从事卫星组合姿态确定系统的工程实现和相关科研工作有一定的参考价值。

#### 参考文献:

[1] 施少范. 国外对地观测卫星高精度姿态控制系统研究 [J]. 上海航天, 2000, (6): 49-53.  
 [2] 秦永元, 张洪钺, 汪叔华. 卡尔曼滤波与组合导航原理 [M]. 西安: 西北工业大学出版社, 1998.  
 [3] 高 桦, 钟 昊. 飞行器姿态大角度机动联合控制物理仿真技术研究 [J]. 计算机测量与控制, 2010, 18 (10): 2315-2317.  
 [4] 任丽香, 马淑芬, 李方慧. TMS320C6000 系列 DSPs 的原理和应用 [M]. 北京: 电子工业出版社, 2000.

(上接第 1628 页)

[5] 陈 晶, 孔令富, 潘 勋. 结合预测机制和 QoS 约束的网络资源调度算法的研究 [J]. 计算机研究与发展, 2008, 45 (s): 11-16.  
 [6] Koomey J G. Estimating total power consumption by servers in the US and the world [R], 2007.  
 [7] 刘 永, 王新华, 王 朕, 等. 节能及信任驱动的虚拟机资源调度 [J]. 计算机应用研究, 2012, 29 (7): 2479-2483.  
 [8] 师雪霖, 徐 恪. 云虚拟机资源分配的效用最大化模型 [J]. 计算机学报, 2013, 36 (2): 252-262.  
 [9] 邓传华, 范通让, 高 峰. Hadoop 下基于统计最优的资源调度算

法 [J]. 计算机应用研究, 2013, 30 (2): 417-422.  
 [10] 张水平, 邹海艳. 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.  
 [11] 周文俊, 曹 健. 基于预测及蚁群算法的云资源调度策略 [J]. 2012, 29 (9): 239-242.  
 [12] 单 良, 强 浩, 李 军. 基于 Tent 映射的混沌优化算法 [J]. 控制与决策, 2005, 20 (2): 179-182.  
 [13] 王永贵, 韩瑞莲. 基于改进蚁群算法的云环境任务调度研究 [J]. 计算机测量与控制, 2011, 19 (5): 1203-1211.