

1-3-2019

## Cloud Job Scheduling Model Based on Improved Plant Growth Algorithm

Li Qiang

*1. Department of Information, College of Finance & Economics, Tai yuan University of Technology, Taiyuan 030024, China; ;2. College of Information, Shanxi Finance & Taxation College, Taiyuan 030024, China; ;*

Xiaofeng Liu

*3. College of Data Science, Taiyuan University of Technology, Taiyuan 030024, China;*

Follow this and additional works at: <https://dc-china-simulation.researchcommons.org/journal>



Part of the Artificial Intelligence and Robotics Commons, Computer Engineering Commons, Numerical Analysis and Scientific Computing Commons, Operations Research, Systems Engineering and Industrial Engineering Commons, and the Systems Science Commons

---

This Paper is brought to you for free and open access by Journal of System Simulation. It has been accepted for inclusion in Journal of System Simulation by an authorized editor of Journal of System Simulation.

---

## Cloud Job Scheduling Model Based on Improved Plant Growth Algorithm

### Abstract

**Abstract:** The performance of cloud job scheduling algorithm has a great importance to the whole cloud system. The key factors that affect cloud operation scheduling are found out, and a resource constraint model is established. *The existing simulation plant growth algorithm is improved based on the Logistic model of plant growth law, so that the plant growth way was made to change according to the energy power.* The comparison of four different plant models was carried out and their different features were analyzed. Compared with 6 typical cloud job scheduling algorithms, it is concluded that the improved simulation plant growth algorithm based on Logistic model has better job scheduling efficiency.

### Keywords

Hadoop, cloud computing, scheduling, plant growth simulation algorithm, MapReduce

### Recommended Citation

Li Qiang, Liu Xiaofeng. Cloud Job Scheduling Model Based on Improved Plant Growth Algorithm[J]. Journal of System Simulation, 2018, 30(12): 4649-4658.

# 基于模拟植物生长算法的云作业调度模型

李强<sup>1,2</sup>, 刘晓峰<sup>3\*</sup>

(1.太原理工大学 财经学院, 太原 030024; 2.山西省财政税务专科学校 信息学院, 太原 030024;  
3.太原理工大学 大数据学院, 太原 030024)

**摘要:** 云作业的调度算法的优良对整个云系统的工作效率具有重要意义。首先, 找出影响云作业调度的关键因素, 建立资源约束模型; 随后, 通过植物生长规律的 Logistic 模型来改进现有的模拟植物生长算法, 使其生长依据能量动力改变植物生长的方式; 最后, 通过 4 种不同的植物模型进行比较, 分析其不同特点, 并与 6 个典型的云作业调度算法比较, 得出结论: 基于 Logistic 模型改进的模拟植物生长算法具有更好的作业调度效率。

**关键词:** Hadoop; 云计算; 调度; 模拟植物生长算法; MapReduce

中图分类号: TP393; TP181 文献标识码: A 文章编号: 1004-731X (2018) 12-4649-10  
DOI: 10.16182/j.issn1004731x.joss.201812020

## Cloud Job Scheduling Model Based on Improved Plant Growth Algorithm

Li Qiang<sup>1,2</sup>, Liu Xiaofeng<sup>3\*</sup>

(1.Department of Information, College of Finance & Economics, Tai yuan University of Technology, Taiyuan 030024, China;  
2.College of Information, Shanxi Finance &Taxation College, Taiyuan 030024, China;  
3. College of Data Science, Taiyuan University of Technology, Taiyuan 030024, China)

**Abstract:** The performance of cloud job scheduling algorithm has a great importance to the whole cloud system. The key factors that affect cloud operation scheduling are found out, and a resource constraint model is established. *The existing simulation plant growth algorithm is improved based on the Logistic model of plant growth law, so that the plant growth way was made to change according to the energy power.* The comparison of four different plant models was carried out and their different features were analyzed. Compared with 6 typical cloud job scheduling algorithms, it is concluded that the improved simulation plant growth algorithm based on Logistic model has better job scheduling efficiency.

**Keywords:** Hadoop; cloud computing; scheduling; plant growth simulation algorithm; MapReduce

## 引言

MapReduce 作为云计算事实上的典型代表, 以突出的数据处理能力得到了学者们的广泛关注。然而, MapReduce 的缺省作业调度方案只是关注节点

的位置特征的两个因素(本地性、机架本地性和远程性), 以及该三者之间的关系, 很多影响其作业调度性能的因素没有考虑进来, 如节点的处理能力、作业的存储状态和环境的异构性等等, 所以默认的调度算法没有能发挥出其系统最大的潜力<sup>[1]</sup>。

随着对云作业调度研究的深入, 大量改进的新算法涌现出来, 成为研究云计算的一个方向例如文献[2]中, 为了减少作业调度中人工参数的设置, 提出一种可以调度参数自动适应的作业调度算法, 方便了用户的管理。



收稿日期: 2018-05-05 修回日期: 2018-07-14;  
基金项目: 国家自然科学基金(61502330), 山西省高等学校科技创新项目(20171119), 山西省软科学计划(2016041008-5);  
作者简介: 李强(1980-), 男, 山西太原, 硕士, 副教授, 高工, 研究方向为植物生长算法; 刘晓峰(通讯作者 1979-), 男, 山西怀仁, 博士, 研究方向为云计算。

<http://www.china-simulation.com>

• 4649 •

在文献[3]中,将 workflow 调度方式和负载平衡机理相结合实现云作业的调度,实现了 MapReduce 作业调度安排的细化。还有文献[4]改进文献[2]中的模型,使用假设分析和优化器相融合的方式细化各个参数对作业调度的作用。还有学者根据云作业的特点将作业划分为处理机型和 I/O 型,分别对待以优化处理作业调度队伍<sup>[5]</sup>。也有研究者以云作业输入参数为依据,使用静态数据过滤策略来优化调度模型<sup>[6]</sup>。

针对异构环境上的云作业调度,文献[7]通过在云作业的映射阶段和混洗阶段,使用 Tarazu 算法调度作业。若混洗阶段资源紧张,则该算法倾向更多的任务在本地运行;否则使用资源平衡策略运行各个任务。也有文献以执行效率为依据,决策任务被分配到加速池或非加速池。该模型通过比较两个池的性能决定哪个池更适合任务的分配,实现任务数量的总体平衡<sup>[8]</sup>。还有文献[9]使用 MARLA 算法在异构环境中来替代 HDFS,测试节点的执行效率实现任务的动态分配。此方法通过主节点在运行过程中判断从节点的执行效率,决定性能高的节点获得更多的任务,实现不同性能节点之间的任务均衡分配。宋杰等研究者以降低系统的能耗为出发点,使用能耗算法来自动调节任务的数量,实现资源的动态分配。

云作业的调度问题其本质属于多条件限制的资源分配问题,可对应于数学上的 NP 问题。为了解决该类问题,大量的算法不断被学者们提出,如退火算法、PSO 算法、蜂群算法和神经网络等。例如使用 Hopfield 神经网络(HNN)解决云存储的负载均衡和短波协同侦收设备调度等问题<sup>[11-12]</sup>。基于模拟植物生长算法(plant growth simulation algorithm, PGSA)在近些年得到了改进<sup>[13]</sup>和广泛应用,特别在车辆调度<sup>[14]</sup>、包装件配送<sup>[15]</sup>和 GPU 的并行处理<sup>[16]</sup>得到了突出的表现。

多数算法主要以某一角度为出发点,使用改进的算法来实现效率的提升。然而,云作业的环境参数是随运行条件不断变化,无法事先测算。

模拟植物生长算法虽然表现优秀,但是原有的生长模型是在一切等概率的情况下发展,与事实的生长环境差异还很大。本文首创诱导函数模拟自然生长环境,使植物生长规律更加客观,并且通过 4 种不同植物的模型比较,找到了可用于云作业调度的大针茅模型。

## 1 云作业调度模型

采用基于植物生长算法实现云作业调度的优化过程其实质是一个求解多项条件约束函数极值的过程,该问题对应于有限条件的整数规划问题<sup>[16]</sup>。为了简化问题,这里采用拉格朗日乘法将多条件约束函数求极值的问题转化为附加条件的多元函数极值问题。

### 1.1 约束条件

被处理的作业集合表示为  $J = \{j_1, j_2, \dots, j_n\}$ ,  $j_k$  表示第  $k$  个作业,  $k=1, 2, \dots, n$ ; 机器节点集合表示为  $N = \{n_1, n_2, \dots, n_m\}$ ,  $n_i$  表示第  $i$  个节点。

**定义 1** 目标函数: 求在所有方案中用时最小的方案,其对应的函数可描述为:

$$f_1 = \min(\max_T \sum_{i=1}^N \sum_{j=1}^M \sum_{k=1}^T X_{ijk}) \quad (1)$$

设变量  $X_{ijk}$  表示作业  $i$  在节点  $j$  的第  $k$  个时间片执行的状态。如果该变量值为 1, 在第  $k$  个时间片时表示在第  $j$  号节点上执行  $i$  号作业, 否则该值为 0。其中,  $1 < i < N$ ,  $N$  为最大作业数;  $1 < j < M$ ,  $M$  为节点数量;  $1 < k < T$ ,  $T$  为最大时间片数。每个作业按时间片分配机器节点。

**定义 2** 作业最长时间限制:

$$f_2 = \min(\sqrt{\sum_{i=1}^N (\sum_{j=1}^M \sum_{k=1}^T X_{ijk} - P_i)^2}) \quad (2)$$

式中:  $P_i$  为第  $i$  个作业运行估计完成时间。如果式中所有的作业都接近估计完成时间, 该式趋近于 0。

**定义 3** 节点时间约束, 限制任意节点上的一个执行时间片内至多有一个作业执行, 可表示为:

$$\varphi_1 = \sum_{i=1}^N \sum_{j=1}^M \sum_{k=1}^T \sum_{\substack{l=1 \\ l \neq i}}^N X_{ijk} \cdot X_{ljk} = 0 \quad (3)$$

**定义 4** 节点作业互斥条件, 一个作业在单位时间内仅可以在一个节点上执行, 可描述为:

$$\varphi_2 = \sum_{i=1}^N \sum_{j=1}^M \sum_{k=1}^T \sum_{l=1}^N \sum_{h=1}^T X_{ijk} \cdot X_{ilh} = 0 \quad (4)$$

**定义 5** 无空闲节点要求, 在任务序列执行周期内没有节点产生空闲周期。该要求可以保证系统整体的性能最佳。

$$\varphi_3 = \sum_{j=1}^M \sum_{k=1}^T (\sum_{i=1}^N X_{ijk} - 1)^2 = 0 \quad (5)$$

**定义 6** 单作业最大周期限制, 可表述为:

$$\varphi_4 = \sum_{i=1}^N \sum_{j=1}^M \sum_{k=1}^T X_{ijk} \cdot K_{ijk}^2 \cdot G(K_{ijk})$$

$$K_{ijk} = k - p_i, \quad \text{and} \quad (6)$$

$$G(K_{ijk}) = \begin{cases} 1 & K_{ijk} > 0 \\ 0 & K_{ijk} \leq 0 \end{cases}$$

式中: 第  $i$  个作业的执行最大时间限制长度为  $p_i$ ;  $G(K_{ijk})$  为单位计算函数。

**定义 7** 资源限制约束。保证任意两个作业不会同时使用相同的资源, 即资源不可抢占条件。

$$\varphi_5 = \sum_{i=1}^N \sum_{j=1}^M \sum_{k=1}^T \sum_{h=1}^N \sum_{l=1}^M \sum_{w=1}^F \sum_{\substack{h \neq i \\ l \neq j}} X_{ijk} \cdot R_{ykw} \cdot X_{hlk} \cdot R_{hlkw} = 0 \quad (7)$$

该问题就是在满足各项有限资源环境中, 对作业的调度顺序进行优化组合, 针对该问题通过设计算法求得最优解的作业调度顺序使得整个任务的完成时间最短。

## 1.2 问题的数学表达

以上约束条件项被满足时, 每项约束条件值为 0。引入拉格朗日乘数因子  $\lambda_i$ , 利用拉格朗日乘数法将约束条件和目标函数按式(8)表达:

$$F = f_1 + f_2 + \sum_{l=1}^5 \lambda_l \cdot \varphi_l \quad (8)$$

通过构造拉格朗日函数, 也就是统一函数表达, 将有限多条件多元函数问题转化为简单多元函数的极值问题。再通过模拟植物生长算法来求解该问题, 就可以实现受资源约束的云作业优化调度。

## 2 模拟植物生长算法

模拟植物生长算法(PGSA)在近些年智能优化算法中独树一帜, 得到了令人瞩目的效果, 特别是求解路径规划和组合优化问题中, 在全局性和稳定性方面表现出比流行的粒子群算法、蚁群算法等具有更好的优势<sup>[14]</sup>。

该算法将求解问题的约束环境映射为植物的生长环境, 依据植物向光性理论生长的规律, 模拟植物生长过程来演绎问题的求解。

### 2.1 植物的生长方式

植物的生长过程可以简单描述为: 首先, 植物的茎从播种点(原始点)发芽出来, 同时在该植物的茎上一些位置发出新芽, 这些可以发芽的位置被称“生长点”; 然后, 在这些生长点上长出新枝, 同时, 这些新枝上也会发出自己的生长点, 这些生长点又会根据环境条件长出新的枝; 最后, 反复枝上生枝, 以类似或相同的结构发展。

依据生物学实验, 植物的茎干和枝上的生长点并不是一定能长出新枝, 而是与生长点上的生长激素(形态素)的浓度有关, 其浓度越高越能发出新枝, 而该浓度的高低与所处的环境有关, 其中阳光是影响作用最大的一个主要因素<sup>[13]</sup>。

### 2.2 植物生长规律的数学描述

根据文献[16], 每个可生长的生长点并不是都能均衡发展成为树枝的, 而是依据生物学规律, 每个生长点所携带细胞形态素的浓度决定, 第  $t$  个生长点的形态素浓度记为  $p_t(0 < p_t < 1)$ 。其单个形态素的浓度计算的积分形式可按(9)式表达。

$$P_t = \frac{f(a) - f(t)}{\int_0^k (f(a) - f(x)) dx + \int_0^p \int_0^q (f(a) - f(x)) dx} \quad (9)$$

我们这里简化树枝的生长过程,假设茎干的长度为  $M$ , 并携带有  $k$  个生长点, 记为  $S_M = \{S_{M1}, S_{M2}, \dots, S_{Mk}\}$ , 同时, 每个对应的生长点附有的形态素浓度记为  $P_M = \{P_{M1}, P_{M2}, \dots, P_{Mk}\}$ ; 且设每一个树枝长度均为单位长度相等, 记为  $m(m < N)$ , 在每个单位长度上均匀分布  $q$  个生长点, 记为  $S_m = \{S_{m1}, S_{m2}, \dots, S_{mq}\}$ , 对应的生长点形态素浓度记为  $P_m = \{P_{m1}, P_{m2}, \dots, P_{mq}\}$ , 树干和树枝上生长点上的形态素浓度计算值可按下式计算:

$$P_{Mi} = \frac{f(a) - f(S_{Mi})}{\sum_{i=1}^k (f(a) - f(S_{Mi})) + \sum_{j=1}^q (f(a) - f(S_{mj}))} \quad (10)$$

$$P_{mi} = \frac{f(a) - f(S_{mi})}{\sum_{i=1}^k (f(a) - f(S_{Mi})) + \sum_{j=1}^q (f(a) - f(S_{mj}))} \quad (11)$$

式中:  $a$  代表种子的生长点, 即原始点;  $f^*$  是目标函数。这里树根和树枝的生长点共有  $k+q$  个, 对应共有  $k+q$  个形态素浓度, 且每次生长, 会有新的生长点产生, 且原有的形态素浓度将被更新。

这里要求所有的生长点的形态素浓度之和为 1, 记为:

$$\sum_{i=1}^k P_{Mi} + \sum_{j=1}^q P_{mj} = 1 \quad (12)$$

式(12)中, 描述了生长点的形态素浓度的分配, 以及浓度越大生长点将拥有更多的机会长出新枝, 而其浓度值的确定由植物的向光性决定。将式(12)中的状态空间映射为概率空间可由图 1 表示。

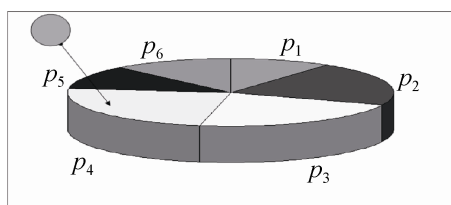


图 1 形态素浓度对应的概率空间  
Fig. 1 Probability space corresponding to the speciation concentration

经过在  $[0,1]$  之间产生随机数, 并映射为扇区大小, 整个小球的面积为 1。该小球落到哪个扇区空间, 其对应的生长点将获得相应的生长概率。依据植物生长的动力学机理, 反复按此规律生长, 直到第  $N$  条枝生长出来或计算结果满足精度条件, 算法结束。

### 2.3 算法改进

#### 2.3.1 诱导生长函数的设计

众所周知, 植物的生长并不是按固定长度增长的, 而是主要受到阳光、水分和环境温度等环境因素综合影响的。所以, 为了提升原有算法的科学性, 这里我们引进能量动力函数来作为植物生长的动力源, 提升原有算法的执行效率和有效性<sup>[17]</sup>。

$$E_j(t) = S(t)W(t)T(t); \quad (13)$$

$$S(t) = \xi\eta(t)$$

式中:  $E_j(t)$  是一定环境植物生长条件下光、水分和环境热量综合作用的强度;  $W(t)$  代表水分作用函数;  $T(t)$  是环境热量作用函数;  $S(t)$  为光能作用函数;  $\xi$  是光利用系数;  $\eta(t)$  是光辐射时间函数。

自然界植物生长规律可按 Logistic 曲线函数描述, 即:

$$L = \frac{L_0}{1 + e^{\alpha - \beta \int_0^t \varphi(x) dx}} \quad (14)$$

式中:  $L$  是物质积累量, 即生长长度;  $L_0$  为最大物质积累常数;  $\alpha$  和  $\beta$  是生长系数; 其中,  $\varphi(x)$  是生长作用函数, 该积分式是求在一定时间  $t$  内的热量之和。使用函数  $L$  时对时间  $t$  求导, 可得:

$$V_t = \frac{dL}{dt} = \frac{\beta L_0 \varphi(x) \exp(\alpha - \beta \int_0^t \varphi(x) dx)}{(1 + \exp(\alpha - \beta \int_0^t \varphi(x) dx))^2} \quad (15)$$

式(15)中:  $V_t$  是单位时间生长速度函数;  $\varphi(x)$  是综合影响函数。

令  $W^*$  水分作用为常数;  $T^*$  环境热量作用为常数, 即  $W(t)T(t) = \lambda$ ; 这里生长影响函数为

$$\varphi(x) = E_j(x); \text{ 则定周期内的植物生长长度为:}$$

$$\begin{aligned} \Delta T &= \omega t; G(t) = \int_0^t \varphi(x) dx; \\ \Delta L &= V_t \Delta T \approx \left\| \frac{\beta L_0 \varphi(x) \exp(\alpha - \beta \int_0^t \varphi(x) dx)}{(1 + \exp(\alpha - \beta \int_0^t \varphi(x) dx))^2} \right\| \Delta T = \\ &= \left\| \frac{\beta L_0 \varphi(\omega t) \exp(\alpha - \beta \int_0^{\omega t} \varphi(x) dx)}{(1 + \exp(\alpha - \beta \int_0^{\omega t} \varphi(x) dx))^2} \right\| \omega t = \\ &= \left\| \frac{\beta L_0 \varphi(\omega t) \exp(\alpha - \beta G(\omega t))}{(1 + \exp(\alpha - \beta G(\omega t)))^2} \right\| \omega t \end{aligned} \quad (16)$$

根据世界气象组织 1981 年的太阳常数值为 1 638 W/m<sup>2</sup>, 所以这里令  $\eta(t)=1\ 638$ ; 光利用系数按图 1 的要求随机变化  $\xi=p_t$ ; 一个迭代周期约为 1 s, 则  $\Delta T=\omega t=1t=1\text{ s}$ ;  $\delta$  为修正系数, 式(16)可进一步修正为:

$$\begin{aligned} \Delta L &\approx \delta \xi \eta \beta L_0 \left\| \frac{\exp(\alpha - \beta G(\omega t))}{(1 + \exp(\alpha - \beta G(\omega t)))^2} \right\| \omega t = \\ \delta L_0 \beta \eta \omega &\left\| \frac{\exp(\alpha - \beta G(\omega t))}{(1 + \exp(\alpha - \beta G(\omega t)))^2} \right\| p_t = \\ 1638 \times \delta \beta L_0 &\left\| \frac{\exp(\alpha - \beta G(\omega t))}{(1 + \exp(\alpha - \beta G(\omega t)))^2} \right\| p_t \end{aligned} \quad (17)$$

### 2.3.2 动力函数分析

为了分析该动力函数的特征, 我们这里继续对该函数求二阶导数, 即求增长的加速度函数的如式(18)表示:

$$\begin{aligned} \alpha &= \frac{\Delta L}{dt} \approx \\ 1683 \times \delta \beta L_0 p_t &\frac{d \left\| \frac{\exp(\alpha - \beta G(\omega t))}{(1 + \exp(\alpha - \beta G(\omega t)))^2} \right\|}{dt} = \\ 1638 \times \delta \beta L_0 p_t &\frac{\beta \exp(\alpha - \beta t) (2 \exp(\alpha - \beta t) - 1)}{1 + \exp(\alpha - \beta G(\omega t))} \end{aligned} \quad (18)$$

令  $a=0$ , 可得:

$$t_0 = \frac{\alpha - \ln \frac{1}{2}}{\beta} L_0 \gamma \quad (19)$$

很明显, Logistic 函数的一阶导数恒大于 0 时, 该函数持续增加。当  $t_0$  点前二阶导数大于 0, 该 Logistic 函数呈现是越快越快态势; 当到  $t_0$  点后二阶导数小于 0, 且一阶导数恒大于零, 该函数的增长开始变缓慢。此外, 式(18)中  $t_0$  点的位置主要与  $\alpha$ 、 $\beta$  和  $L_0$  有关, 不同植物的模型参数  $\alpha$ 、 $\beta$  和  $L_0$  不同, 也就是不同植物模型的增长方式、算法的收敛速度和位置是不同的。

### 3 算法描述

本文的云作业调度算法步骤描述如下:

#### 算法 1. Cloud\_Scheduling\_Based\_Improved\_PGSA

**输入:** 资源互斥矩阵  $R_{ij}=\{r_{ij}\}$ , 节点数量  $n$ , 作业数  $m$ , 每个作业的估计完成时间  $P_i$ , 最大迭代次数  $K$ 。

**输出:** 路径  $path=\{S_0, S_i, S_j, \dots, S_k\}$

第一步: 初始化  $V_{ijk}=1$ ; 初始化树干的有  $k$  个生长点; 树枝的生长点数  $q=0$ ; 确定原始点  $S_0$ , 并将  $S_0$  添加到集合  $S$  中, 即  $S=\{S_0\}$ 。

第二步: 更新树干生长节点, 为树干生成  $k$  个生长点  $S_M=\{S_{M1}, S_{M2}, \dots, S_{Mk}\}$ , 依据公式(10)为每个生长点计算生长概率, 为  $P_M=\{P_{M1}, P_{M2}, \dots, P_{Mk}\}$ 。

第三步: 更新新树枝生长节点, 为树枝共分配  $q$  个生长点, 记为  $S_m=\{S_{m1}, S_{m2}, \dots, S_{mq}\}$ , 依据公式(11)计算生长点的浓度  $p_m=\{p_{m1}, p_{m2}, \dots, p_{mq}\}$ 。

第四步: 确定  $k+q$  个生长点的位置, 并依据公式(12)计算  $k+q$  个生长点的浓度及关系。

第五步: 依据公式(16), 计算选举出来的生长点的生长长度和位置, 这里位置是将新枝按随机正负  $45^\circ$  方向调整, 形成新的端点  $S_{t+1}$ 。

第六步: 更新集合  $S$ , 即将  $S=\{S_0, S_1, S_2, \dots, S_t, S_{t+1}\}$ ; 求最新的树枝端点到原点的路径  $p=\{S_0, S_1, S_2, \dots, S_t, S_{t+1}\}$ ; 如果  $p < path$ , 则更新  $path=p$ ; 将该路径代入公式(8)判断是否达到最优路径, 若满足

最优条件或达到迭代上限, 进入下一步, 否则转到第二步。

第七步: 若满足最优条件, 则输出路径  $path$ , 即任务安排方案; 否则提示用户未找到最优解。

以上步骤如图 2 所示。

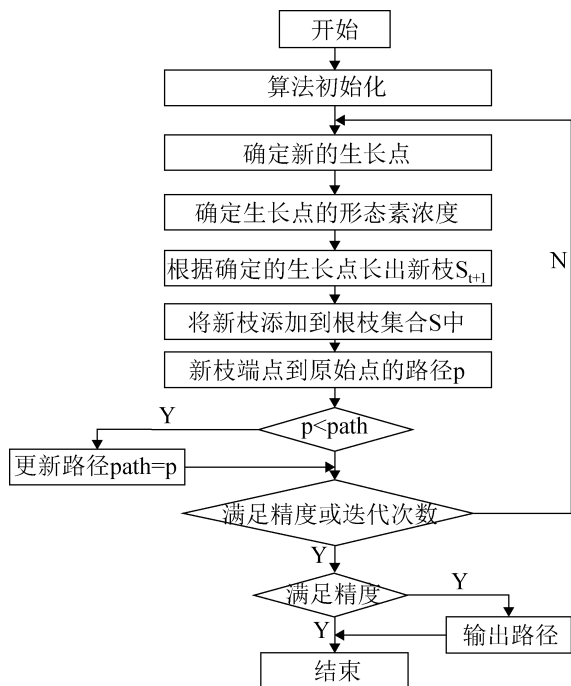


图 2 算法步骤  
Fig. 2 Algorithm steps

## 4 实验与分析

### 4.1 实验环境与参数

该算法将在具有 7 台 PC 机的 Hadoop 集群上运行, 每台计算机的配置为联想 3.5 GHz 处理器, 并配有 4 G 内存, 1TB 硬盘, 千兆网络带宽。其中 1 台为 Master 点, 其它 6 台为计算节点。

该实验使用标准云测试用例 Wordcount、Terasort、Pi Estimator, 随机方式运行, 并设置 map 和 reduce 的任务数都为 9。其中, Wordcount 和 Pi Estimator 测试是属于 CPU 密集型用例, 而 Terasort 属于 CPU 密集型和 I/O 密集型的综合代表。

本文采用静态和动态折中的方式定期采集各个计算节点的资源状态, 包括 CPU、内存、I/O 和

网络利用率的状态, Master 采集周期为  $T=10$  s。该方式可以避免静态调度不能实现资源负载平衡, 又可以防止动态调度过于频繁地在节点间传输任务和状态信息, 增加系统额外的开销。

## 4.2 结果分析

### 4.2.1 不同植物模型比较

为了更好的反应植物自然增长规律, 本文采用文献[18]中的 4 种实际植物生长参数来设置本实验的参数, 如表 1 所示。

表 1 实验参数设置  
Tab. 1 Parameters for experiment

植物类型	$L_0$	$\alpha$	$\beta$
大针茅	1.205	2.608	0.042
冷蒿	3.177	2.770	0.077
羊草	0.200	2.032	0.060
冰草	0.156	1.858	0.040

如表 2 所描述, 在标准云测试用例中大针茅、冷蒿、羊草和冰草 4 种植物模型被用来实验和比较, 可以发现, 这 4 种不同植物模型的系统开销分别为 734 s、813 s、855 s 和 921 s。其中, 在该测试用例下, 大针茅模型的性能最优。

表 2 不同植物模型的执行性能  
Tab. 2 Performance of different models

植物模型	执行时间/s	Map 计时/s	Reduce 计时/s
大针茅	734	677	713
冷蒿	813	653	797
羊草	855	676	832
冰草	921	781	913

在实验中, 采用四种不同生长参数模拟四种不同植物的生长, 来观察问题的求解过程见图 3。可以通过该问题求解的形态我们可以观测到, 这 4 种模型共有的生长态势基本上呈现出越长越短, 且越来越散的趋势, 该情况符合本文采用 Logistic 动力函数诱导根枝生长方式的结果, 更加符合植物生长的方式, 有助于提升问题空间的搜索效率, 并且在植物的顶端加速算法的收敛。



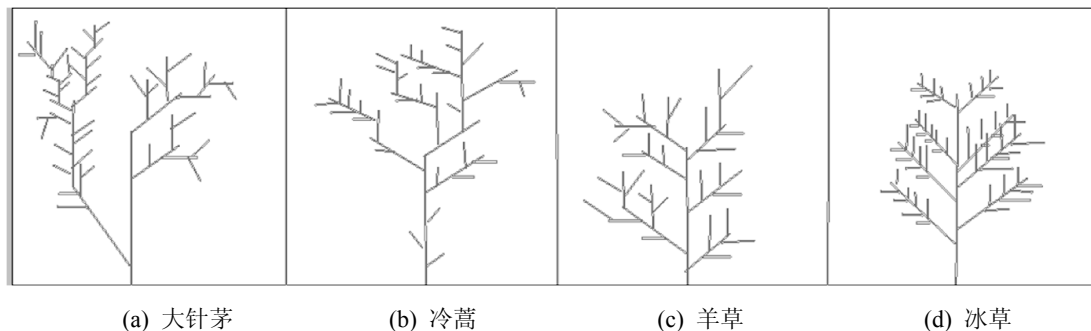


图 3 不同植物模型的形态  
Fig. 3 Morphology of different plant models

图 3 中, 大针茅、冷蒿的形态属于占用空间比较散, 具有形态偏高, 枝点偏稀松的特征; 而羊草和冰草的形态属于占用空间比较集中, 具有形态偏矮, 枝点偏密的特征。

根据反复实验的经验总结, 我们这里得出了“小空间多元变量易采用短杆多枝的植被模型, 大空间少元变量易采用长杆少枝的植被模型”经验规律。

4.2.2 不同算法比较

表 3 中展示的是大针茅模型的具体执行时间分布, 可以观测到 3 个标准测试用例 Wordcount、Terasort 和 Pi Estimator 同时随机执行的周期分别是 734 s、633 s 和 93 s, 系统总开销为 734 s。

表 3 大针茅模型的执行性能  
Tab. 3 Performance of the stipa gris model

用例序列	执行时间/s	Map 计时/s	Reduce 计时/s
Wordcount	734	677	713
Terasort	633	578	614
PiEstimator	93	73	87
总执行时间	734		

在本实验中, 我们分别采用 IPGSA(大针茅植物模型)、PGSA、蚁群算法、PSO 算法、动态调用算法、能量算法和 Hadoop 默认的公平调度算法来测试比较。以上 7 种算法的系统开销如表 4 所示, 分别为 734 s、894 s、974 s、937 s、1443 s、1535 s 和 3923 s, 前 6 种算法比 Hadoop 默认算法分别提升 81%、77%、75%、76%、67%和 61%, 我们可以归纳出 Hadoop 的默认算法具有很高的提升空间, 该算法并不能发挥系统的最佳优势, 有很多的局

限性。同时, 我们可以发现改进的植物模拟算法的性能比其它典型算法更高。

表 4 典型算法比较  
Tab. 4 Comparison of different algorithms

算法名称	执行时间/s	Map 计时/s	Reduce 计时/s	提升率/%
IPGSA	734	677	713	81
PGSA	894	774	881	77
蚁群算法	974	847	919	75
PSO	937	811	923	76
动态调用算法	1 443	1 380	1 437	67
能耗算法	1 535	1 497	1 513	61
公平调度算法	3 923	3 733	3 830	-

注: 这里改进的植物生长模拟算法(IPGSA)采用的是大针茅植物模型。

通过系统监控程序, 可以从图 4 中观测 7 个算法的 4 个性能指标, 包括 CPU、内存、I/O 和网络 4 个方面的平均利用率。在 Hadoop 缺省的公平调度算法中, 以上 4 个指标比其它算法都高, 而本文推荐的改进的植物生长模拟算法(IPGSA), 其 4 个指标基本上接近最低, 说明该算法可以较好的分配任务, 降低了作业调度的频次, 减少了不必要的系统开销。

从计算的角度出发, 可以观察到除公平调度算法 CPU 和内存的平均利用率超过 90%外, 其它 6 个算法的这两个指标都接近 80%, 可以得出这 6 个算法花费在计算中的代价基本相同。

从系统外部开销的角度观察, 按 I/O 和网络的平均利用率由高到低排列, 分别是公平调度算法、动态调度算法、能耗算法、蚁群算法、PSO 算法和

IPGSA, 其中公平调度算法的两个指标均超过 85%, 开销最大。而 IPGSA 和 PGSA 的两个指标接近 45%, 开销最小, 说明该算法可以合理地按需调度作业, 资源利用充分, 没有过度地在节点间传输任务, 减少了系统外部开销。同时, 可以观察到剩余其它 4 个算法的这两个指标约在 69%~81%之间, 可以得出该 4 个算法的外部开销基本接近。

此外, 从表 4 中可以进一步观测到改进的植物生长模拟算法(IPGSA)比传统的植物生长模拟算法(PGSA)的执行效率还可以进一步提升 4%。结合图 3, 还能发现 IPGSA 比 PGSA 的 CPU 利用率和内存利用率略更低一点, 分析其原因为 IPGSA 由于采用 Logistic 动力函数减少了整体的计算量和算法

的迭代次数, 计算的开销所以降低了。因此, 本文推荐的 IPGSA 算法比其它算法在性能方面更具有显著的优势。

图 5 中表现的是本文推荐的改进的植物模拟生长算法(IPGSA)在 7 个节点上运行的 4 个性能指标, 包括 CPU 利用率、内存利用率、I/O 利用率和网络利用率。其中, 第 1 个节点是 Master 节点, 执行实验算法并负责作业调度, 而其它 6 个是计算节点, 负责任务执行。从该图可以看出, Master 节点上的四个指标均超过 80%, Master 节点不仅负责完成本算法的执行, 而且实现任务调度和其它计算节点状态检测, 所以 Master 节点任务最重。

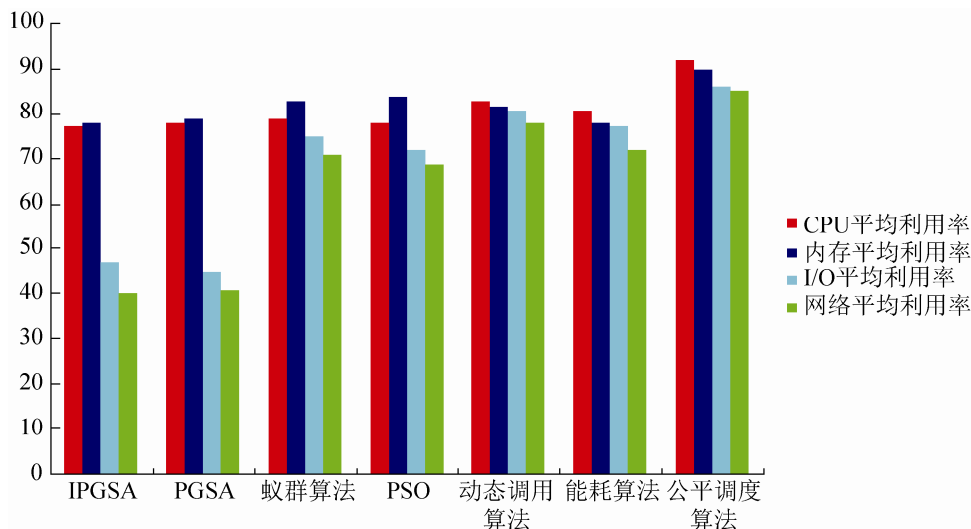


图 4 典型算法的性能  
Fig. 4 Performances of classic methods

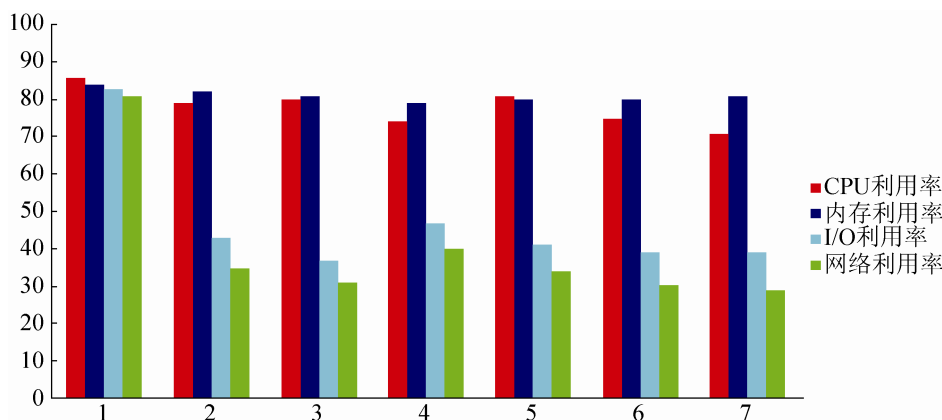


图 5 节点的性能  
Fig. 5 Performances of nodes

<http://www.china-simulation.com>

同时, 我们还可以发现其它 6 个计算节点的 CPU 利用率和内存利用率均接近 80%, 可以得出计算负载可以均匀布置, 没有低效的计算节点。此外, 还可以观察到该 6 个节点的 I/O 利用率和网络利用率分布在 29%~47%之间, 说明这些节点的外部开销基本接近, 资源负载比较均衡。

经过上述分析, 可以得出不论内部计算开销, 还是外部传输开销, 本文所推荐的算法可以比较显著地实现系统整体资源的合理分配, 且可以比较高效地完成云作业的调度。

## 5 结论

本文将光能量函数作为植物生长的动力来提升模拟植物生长算法的性能, 提出一种可变速生长速度的植物模拟算法来实现云作业的调度策略, 并通过实验证明该方法的有效性。本文中通过自然环境对植物生长的动力学分析, 设计能量动力诱导函数作为植物生长的依据, 更加科学地模拟了植物生长的发育。

在实验中, 采用 4 种不同生长参数的植物模型, 模拟 4 种不同植物的生长, 来观察问题的求解, 并得出了“小空间多元变量易采用短杆多枝的植被模型, 大空间少元变量易采用长杆少枝的植被模型”经验规律。

与现有的云作业调度算法、粒子群算法、蚁群算算法、基于能量的调度算法和动态调度算法实验比较和分析, 说明该算法具有资源分配更加合理、性能更高的特点。此外, 起点对计算结果和效率的影响, 以及植物模型如何选择等问题将是我们下一个研究的任务。

## 参考文献:

- [1] 黄山, 王波涛, 王国仁, 等. MapReduce 优化技术综述[J]. 计算机科学与探索, 2013, 7(10): 885-905.  
Huang Shan, Wang Botao, Wang Guoren, et al. A Survey on MapReduce Optimization Technologies[J]. Journal of Frontiers of Computer Science & Technology, 2013, 7(10): 885-905.
- [2] Shivnath B. Towards automatic optimization of MapReduce programs[C]// Proceedings of the 1st ACM Symposium on Cloud Computing (SoCC '10), Indianapolis, USA, Jun 10- 11, 2010. New York: NY, USA: ACM, 2010: 137-142.
- [3] Herodotou H, Lim H, Luo Gang, et al. Starfish: a self-tuning system for big data analytics[C]// Proceedings of the 5th Biennial Conference on Innovative Data Systems Research (CIDR 11), Asilomar, USA: Jan 9-12, 2011: 261-272.
- [4] Herodotos H, Shivnath B. Profiling, what- if analysis, and cost-based optimization of MapReduce programs[C]// Proceedings of the 36th International Conference on Very Large Data Bases (VLDB 10), Singapore, 2010: 1111-1122.
- [5] Tian Chao, Zhou Haojie, He Yongqiang, et al. A dynamic MapReduce scheduler for heterogeneous workloads[C]// Proceedings of the 8th International Conference on Grid and Cooperative Computing (GCC 09), Lanzhou: 2009: 218-224.
- [6] Jahani E, Cafarella M J, Re C. Automatic optimization for MapReduce programs[C]// Proceedings of the 37th International Conference on Very Large Data Bases (VLDB 11), Seattle, USA: 2011: 385-396.
- [7] Ahmad F, Chakradhar S T, Raghunathan A, et al. Tarazu: optimizing MapReduce on heterogeneous clusters[C]// Proceedings of the 7th International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS 12), London: UK, 2012: 61-74.
- [8] Polo J, Carrera D, Becerra Y, et al. Performance management of accelerated MapReduce workloads in heterogeneous clusters[C]// Proceedings of the 2010 39th International Conference on Parallel Processing (ICPP 10), San Diego, USA, 2010. Washington, DC, USA: IEEE Computer Society, 2010: 653-662.
- [9] Z Fadika, E Dede, J Hartog, et al. MARLA: MapReduce for heterogeneous clusters[C]// Proceedings of the 12th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (CC-Grid'12), Ottawa, Canada, 2012: 49-56.
- [10] 宋杰, 徐澍, 郭朝鹏, 等. 一种优化 MapReduce 系统能耗的任务分发算法[J]. 计算机学报, 2016, 39(2): 323-338.  
Song Jie, Xu Shu, Guo Chaopen, et al. A Task Distribution Algorithm for Energy Consumption Optimization of MapReduce System[J]. Chinese Journal of Computers, 2016, 39(2): 323-338.
- [11] 李强, 刘晓峰. 基于 Hopfield 神经网络的云存储负载

- 均衡策略[J]. 计算机应用, 2017, 37(8): 2214-2217.
- Li Qiang, Liu Xiaofeng. Load balancing strategy of cloud storage based on Hopfield neural network[J]. Journal of Computer Applications, 2017, 37(8): 2214-2217.
- [12] 刘洋, 王伦文. 基于 Hopfield 网络的短波协同侦收设备调度技术[J]. 火力与指挥控制, 2017, 42(12): 135-138, 143.
- Liu Yang, Wang Lunwen. Research on Technology of Resource Scheduling for HF Cooperative Reception Based on Hopfield Neural Network[J]. Fire Control & Command Control, 2017, 42(12): 135-138, 143.
- [13] 王莉, 秦勇, 徐杰, 等. 植物多向生长模拟算法[J]. 系统工程理论与实践, 2014, 34(4): 1018-1027.
- Wang Li, Qin Yong, Xu Jie, et al. Plant multi-direction growth simulation algorithm[J]. Systems Engineering-Theory & Practice, 2014, 34(4): 1018-1027.
- [14] 曹庆奎, 刘新雨, 任向阳. 基于模拟植物生长算法的车辆调度问题[J]. 系统工程理论与实践, 2015, 35(6): 1449-1456.
- Cao Qingkui, Liu Xinyu, Ren Xiangyang. Vehicle scheduling problem based on plant growth simulation algorithm [J]. Systems Engineering-Theory & Practice, 2015, 35(6): 1449-1456.
- [15] 樊贵香. 混合模拟植物生长算法在包装件配送中的应用[J]. 包装工程, 2016, 37(13): 43-49.
- Fan Guixiang. Application of Hybrid Plant Growth Simulation Algorithm in Packaged Products Distribution[J]. Packaging Engineering, 2016, 37(13): 43-49.
- [16] 杨磊, 龚学余. 基于 GPU 的并行植物模拟生长算法[J]. 计算机工程与设计, 2016, 37(12): 3271-3279.
- Yang Lei, Gong Xueyu. Parallel plant growth simulation algorithm based on GPU [J]. Computer Engineering and Design, 2016, 37(12): 3271-3279.
- [17] 唐卫东, 李萍萍, 李金忠. 基于生长动力学的芦苇属植株虚拟生长模型[J]. 计算机应用, 2015, 35(4): 1110-1115.
- Tang Weidong, Li Pingping, Li Jinzhong. Virtual development model of plant-reed based on growth kinetics[J]. Journal of Computer Applications, 2015, 35(4): 1110-1115.
- [18] 张彩琴, 杨持. 内蒙古典型草原生长季内不同植物生长动态的模拟[J]. 生态学报, 2007, 27(9): 3618-3629.
- Zhang Caiqin, Yang Chi. Simulation of growth dynamics of four plants in the typical steppe of Inner Mongolia in growing season[J]. Acta Ecologica Sinica, 2007, 27(9): 3618-3629.

## 《系统仿真学报》荣获“2017 中国国际影响力优秀学术期刊”证书

由中国学术期刊（光盘版）电子杂志社与清华大学图书馆联合成立的中国学术文献国际评价研究中心，发布了 2017 版《中国学术期刊国际引证年报》，《系统仿真学报》荣获“2017 中国国际影响力优秀学术期刊”。

《年报》（2017 版）采用的统计源期刊为 20192 种，涵盖 WoS 收录的 SCI 期刊 8874 种、SSCI 和 A&HCI 期刊 4645 种，ESCI 期刊 5578 种；增补期刊 1762 种。参照中外文学术期刊总被引频次、影响因子、半衰期等各项国际引证指标，计算期刊影响力指数(CI)，对国内 6210 种学术期刊排序，遴选了人文社科、自然科学与工程技术两个类别的 TOP10%为国际影响力品牌学术期刊。TOP5%以内的期刊为“最具国际影响力学术期刊”、TOP5-10%之间的为“国际影响力优秀学术期刊”。