

离散 PSO 结合模拟退火算法解决云调度问题

赵 轩¹, 蔚承建¹, 王 开², 刘 凯³

(1 南京工业大学 电子与信息工程学院, 江苏 南京 210009; 2 东南大学 信息科学与工程学院, 江苏 南京 210018;
3 解放军理工大学 国防工程学院, 江苏 南京 210007)

摘 要: 本文以最小化工作流程中任务完成所需的总的综合执行成本为目标, 用一种改进的离散形式的粒子群算法并引入模拟退火机制和混沌机制来完成调度工作, 并用实验证明该算法可行高效。

关键词: 云计算; 云调度; 离散 PSO; 模拟退火

中图分类号: TP301

文献标识码: A

文章编号: 1000-7180(2013)05-0137-04

Solving Scheduling Problem of Cloud Computing by Discrete PSO with Simulate Anneal Algorithm

ZHAO Xuan¹, WEI Cheng-jian¹, WANG Kai², LIU Kai³

(1 College of Electronics and Information Engineering, Nanjing University of Technology, Nanjing 210009, China;
2 Department of Information Science and Engineering, Southeast University, Nanjing 210018, China;
3 Defence College of Engineering, PLA University of Science and Technology, Nanjing 210007, China)

Abstract: This thesis will propose a modified discrete PSO with simulate anneal algorithm to solve the scheduling problem of cloud computing and to minimize the general cost, at the last, we provide experiments for proving the algorithm feasibility and efficiency.

Key words: computing; scheduling; discrete PSO; sumulate anneal

1 引言

云计算已成为全球 ICT (Information Communication Technology) 产业界发展重点, 研究云计算具有重要的现实意义和潜在价值。云调度是指在云平台 and 云技术下对用户提交的工作流做分布式动态资源分配与自动部署, 是云计算操作系统的核心任务, 也是云计算的核心技术之一, 本文, 借以工作流形式作为云调度分配的研究载体, 以最小化工作流程中任务完成所需的最大综合执行成本 (包括计算成本和计算资源之间的通信成本) 为目标, 用改进后的离散形式的 PSO 算法来达到此目的。

2 工作流调度问题 (workflow scheduling problem)

本文以应用工作流的综合执行成本为目标函数。把一个应用工作流用一个有向无环图^[1] (DAG, Directed Acyclic Graph) $G = (V, E)$ 表示, 其中 $V = \{T_1, \dots, T_n\}$ 为任务集, E 代表任务之间的数据依赖, 即 $f_{j,k} = (T_j, T_k) \in E$ 为 T_j 产生由 T_k 使用的数据。 $S = \{1, \dots, i\}$ 代表存储点的集合, $PC = \{1, \dots, j\}$ 代表计算点的集合, $T = \{1, \dots, k\}$ 代表任务集合。假设对于一定的输入大小, 任务 T_k 在计算资源 PC_j 上所花费的平均计算时间已知, 一个任务在一个计算机上的计算费用与该任务在这个计算机上所花费的时间成反比。同样, 也假设从一个资源

收稿日期: 2012-07-15; 修回日期: 2012-08-27

基金项目: 江苏省自然科学基金 (BK2011060, BK2010240); 教育部博士点基金 (20100092120012)

i 到一个资源 j 的单位数据访问成本 $d_{i,j}$ 为已知. 一般来说, 这个访问成本是由服务提供商固定的. 传输费用可以根据站点之间的带宽来计算. 在本文中假定数据访问成本都是非负, 对称且满足三角不等式, 即有: 对于所有的 $i, j \in N, d_{i,j} = d_{j,i}$, 并且对于所有的 $i, j, k \in N$, 有 $d_{i,j} + d_{j,k} \geq d_{i,k}$.^[2-7]

图 1(左) 描述了一个带有五个任务结点的工作流程图, 任务之间的依赖关系用箭头表示, 比如任务 T_2 的完成依赖于任务 T_1 . 由图可知, 根任务结点有一个输入文件(f_{in}) 且最后一个任务结点 T_5 有一个输出文件(f_{out}). 其中的每个数据(如 $f_{12}, f_{13}, \dots, f_{45}$) 都会在每个任务完成之后产生. 如果某个任务结点有子结点, 则它产生的数据将被其子结点使用. 两个任务之间数据的数值由一个权值 e_{k_1, k_2} 表示, 其中 $k_1 \in T$, 且 $k_2 \in T$. 同样, 图 1(右) 也描述了三个相互联系的计算机结点(PC_1, PC_2, PC_3) 和它们相对应的存储单元(S_1, S_2, S_3). 本文优化工作流的目标就是将工作流任务分配给计算机资源的同时, 使得所有分配的资源中最大综合执行费用(即计算资源时的计算成本和计算资源之间的通信成本之和)最少.

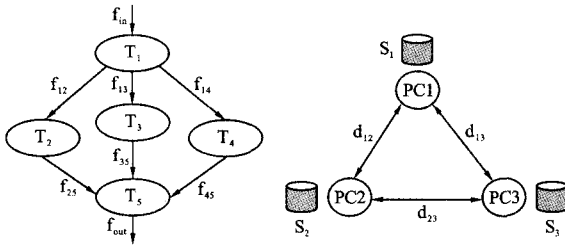


图 1 工作流实例, 计算结点(PC) 和存储结点(S)

这里设 $C_{cpt}(M)_j$ 为分配给计算资源 PC_j 的所有任务的总的计算成本(如公式 1). 在映射 M 中, 该值为分配给计算资源 PC_j 的所有任务的权值之和. 设 $C_{access}(M)_j$ 为在映射 M 中, 分配给计算资源 PC_j 与没有分配给该计算资源的任务之间的总的交互成本(如公式 2). 该值为从一个任务 $k_1 \in k$ 到任务 $k_2 \in k$ 的输出文件大小(由权值 e_{k_1, k_2} 给出) 与两个资源之间的单位数据量的平均通信成本(由 $d_{M(k_1), M(k_2)}$ 表示, 其中 k_1 映射为 $M(k_1)$ 且 k_2 映射为 $M(k_2)$) 的乘积. 仅当两个任务之间有文件依赖关系时(即 $e_{k_1, k_2} > 0$), 通信成本才有实际意义. 否则, 对于两个或者更多的任务, 如果是在同一个计算资源上执行, 则这些任务之间的通信费用为 0.

$$C_{cpt}(M)_j = \sum_k \omega_{kj} \quad \forall M(k) = j \quad (1)$$

$$C_{access}(M)_j = \sum_{k_1 \in T} \sum_{k_2 \in T} d_{M(k_1), M(k_2)} e_{k_1, k_2}$$

$$\forall M(k_1) = j \text{ 且 } M(k_2) \neq j \quad (2)$$

$$C_{total}(M)_j = C_{cpt}(M)_j + C_{access}(M)_j \quad (3)$$

$$Cost(M) = \max(C_{total}(M)_j) \quad \forall j \in P \quad (4)$$

$$\text{Minimize}(Cost(M) \quad \forall M) \quad (5)$$

公式(4) 确保分配方案不会考虑把所有的任务都映射到一个单一的计算结点的综合成本作为, 因为这样没有计算机资源之间的通信费用, 是难以做到 MAX 的. 初始时将任务分配给所有的计算机资源, 然后通过算法迭代求出最小化综合成本(由公式 5 限定), 使得综合成本在第一次初始分配之后呈下降趋势.

3 改进的离散粒子群算法

传统的 PSO 算法一般用来解决连续的函数空间领域, 本文通过将传统的 PSO 运算形式改进成离散形式用混沌机制更新随机参数并引入模拟退火算法接受迭代过程中产生的新解进而求出近似最优的最终解. PSO 改进过程如下:

假设有 Np 个粒子, 粒子维度为 $D = M \times N$ (M 为任务总数, N 为资源总数). 第 h 个粒子 ($h = 1, \dots, Np$), 位置 $X_h = \{X_{h11}, \dots, X_{hMN}\}$, $X_{hij} \in \{0, 1\}$, 表示第 h 个粒子将第 i 个任务分配给第 j 个资源 ($i = 1, \dots, M; j = 1, \dots, N$), 速度用 $V_h = \{V_{h11}, \dots, V_{hMN}\}$ 表示, 第 h 个粒子的局部历史最优解为 $L_h = \{L_{h11}, \dots, L_{hMN}\}$, 另外, $G = \{G_{11}, \dots, G_{MP}\}$ 则表示粒子群中全局最优解. 速度的更新公式为

$$V_{hij}^{new} = \omega V_{hij} + c_1 r_1 (L_{hij} - X_{hij}) + c_2 r_2 (G_{ij} - X_{hij}) \quad (6)$$

其中各个参数系数意义与传统粒子群速度更新公式相同, 我们对速度 V_{hij} 做适当的改进以适应本文离散空间运算. 由于粒子群算法的创始人 Kennedy 和 Eberhart 认为离散 PSO 算法中速度向量里的元素值越高, 则相应的位置向量里的元素越可能选择 1, 反之速度向量中的元素值越低, 则相应的位置向量里的元素越可能为 0, 因此, 对于速度做如下修改:

$$s(V_{hij}) = \frac{1}{1 + \exp(-V_{hij})} \quad (7)$$

$s(V_{hij}) \in (0, 1)$ 用来表示对应位置 X_{hij} 设置为 0 或 1 的概率, 其中 $V_{hij} \in [-V_{max}, +V_{max}]$, 为了避免 $s(V_{hij})$ 靠近 0 和 1, 常量 V_{max} 用来限定 V_{hij} 的取值范围. 文中设定 $V_{max} = 4$, 则 $s(V_{hij})$ 取值范围大致介于 0.018 和 0.9820 之间, 按照原先 Kennedy 和 Eberhart 的思想则位置按照下式更新:

$$\begin{cases} X_{hij} = 1, & \text{if } \text{rand}() < s(V_{hij}) \\ X_{hij} = 0, & \text{else} \end{cases} \quad (8)$$

但考虑到本文调度问题的约束性,即每个任务仅仅只能分配给一个资源,因此将式(7)进行标准化概率形式

$$q_h(i, j) = \frac{s(V_{hij})}{\sum_{j \in U} s(V_{hij})} \quad U \text{ 为资源集合} \quad (9)$$

然后位置 X_{hij} 更新由式(8)改为基于式(9) 概率采用“轮转选择法”分配任务 i 给资源 j 。

4 算法流程

```

初始化参数并输入调度任务;
计算各个计算资源上对各任务的平均计算成本;
计算资源之间的平均通信成本;
设定边的权值  $e_{k1, k2}$  为任务间传输文件大小;
初始化粒子群, 包括各粒子的速度矩阵, 位置矩阵,  $L_h$ 
 $= X_h$ ,  $h=1, \dots, N_p$ ,  $G = X_1$ ;
Loop
For  $h=1$  to  $N_p$  //找出群中最优解
If  $f(X_h) < f(G)$  then //  $f()$  为目标函数在此为综合执行成本
 $g = h$  //  $g$  为群中最优解的粒子序号
End if
Next  $h$ 
For  $h=1$  to  $N_p$ 
//更新速度矩阵, 用混沌处理等式(1)中的系数
Update the velocity matrix  $V_h$  by chaotic random coefficient
subject to  $V_{hij} \in [-V_{max}, +V_{max}]$ 
map  $V_{hij}$  to  $s(V_{hij})$  //等式(7)
normalized probability  $q_h(i, j)$  //等式(9)
assign task  $i$  to resource  $j$  ( $X_{hij}$ ) by roulette wheel selection rule based on  $q_h(i, j)$ 
//下面用模拟退火算法更新  $X_h$  矩阵
 $\Delta E = f(X_h) - f(L_h)$ 
if  $\Delta E < 0$  then  $L_h = X_h$ 
else Compute  $\delta = \exp(-\Delta T'/T)$ 
If a generated random number  $Pa < \delta$  then
 $L_h = X_h$ 
End if
Next  $h$ 
decrease the temperature  $T$ 
Until the end of criterion is reached.

```

5 实验结果与分析

设定算法初始参数如下: 模拟退火算法的初始

温度 $T=100$, 冷却系数 $r=0.99$, $\omega=0.8$, $c_1=c_2=1$, $V_{max}=4$, 迭代数为 100. 任务数为 7, 计算资源数为 5, 7 个任务的处理时间为: $\text{process time} = \{15, 10, 20, 30, 15, 25, 35\}$, 内存需求为: $\text{memoryrequirement} = \{20, 30, 25, 20, 10, 35, 5\}$, 5 个计算资源的可用内存为 $\text{memoryavailable} = \{45, 40, 50, 55, 35\}$, 任务与任务之间的数据依赖(权值)矩阵 taskdataflow 及计算资源(cloud)之间的通信费用矩阵 linkcost 如下:

$$\text{taskdataflow} = \begin{Bmatrix} 0, 5, 0, 4, 1, 1, 3 \\ 5, 0, 1, 0, 6, 3, 1 \\ 0, 1, 0, 3, 0, 0, 0 \\ 4, 0, 3, 0, 0, 1, 3 \\ 1, 3, 0, 3, 0, 2, 3 \\ 1, 0, 2, 0, 0, 0, 3 \\ 1, 3, 0, 0, 3, 2, 0 \end{Bmatrix}$$

$$\text{linkcost} = \begin{Bmatrix} 0, 2, 3, 0, 2 \\ 2, 0, 0, 1, 2 \\ 3, 0, 0, 3, 3 \\ 0, 1, 3, 0, 3 \\ 2, 2, 3, 3, 0 \end{Bmatrix}$$

实例任务调度示意图如图 2、图 3 所示。

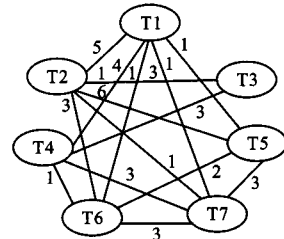


图 2 调度实例示意图

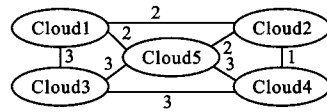


图 3 调度实例示意图

任务分配结果矩阵如表 1 所示。

总的计算成本为 54, 可以看出, 改进的 PSO 算法得到的结果显示花费比较合理。

还可以通过处理不同大小的任务来说明用改进的 PSO 算法可以高效充分利用所有资源。图 4 为各个资源的负载情况。

图中纵坐标表示各个资源处理总任务量的百分比, 横坐标表示任务处理时间的倍数, 我们假设任务处理时间越长表明任务总量越大, 在此试验中我们以上个实验中的 7 个任务的处理时间“process time”

表 1 任务分配结果

OUTPUT			
Cloudlet ID	STATUS	Data center ID	VM ID
4	SUCCESS	2	0
7	SUCCESS	2	2
6	SUCCESS	2	4
2	SUCCESS	2	1
5	SUCCESS	2	1
1	SUCCESS	2	3
3	SUCCESS	2	3
***** Total Debt *****			
BestFitness 54			

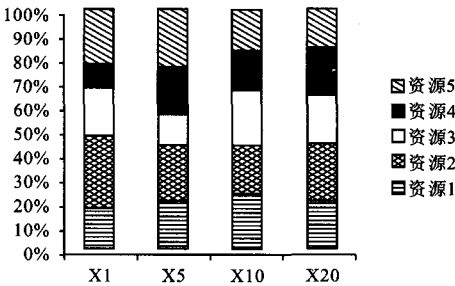


图 4 资源负载图示

为基数 X1, X5 则是 5 倍的该处理时间“process time”(即 process time 中各个分量乘以 5), X10、X20 以此类推。该图表明了,在改进的算法下,所有的任务都有选择的分布在各资源上,力求资源最大化利用,当数据量相对较小时(例如 X1),任务主要分配到某个综合费用较低的资源上,而当数据量不断变大上升时,任务却逐渐均匀地分布到各个资源上。显然这样的分配是比较合理的。

在算法还未真正执行时,一般得到的初始总成本较高,随着算法的推进,收敛性变得越来越强烈,并且很快就能搜索到一个近似全局最小值。在本实验的运算环境中,一般通过迭代 30~50 次左右,便收敛到一个稳定值,计算机资源的分配也趋于平衡。

6 结束语

本文提出用离散的粒子群算法结合模拟退火算法以及混沌理论可以较好的实现云工作流的调度

分配问题。随着云计算相关技术的发展,云结构和云处理将可能更加复杂,分散式的算法会逐步代替集中式来处理大规模的工作流调度问题,为适应需求今后的研究将考虑更加高效的调度算法。

参考文献:

[1] Chang F, Dean J. Ghemawat S, et al Bigtable: A distributed structured data storage system[C]// proceeding of the 7th OSDI. [s. l.]:IEEE,2006;205-218.

[2] Frank R Kschischang, Brendan J Frey, Hans—Andrea Loeliger. Factor Graphs and the sum—product algorithm [J]. IEEE Transactions on Information Theory, 2001, 47(2):498-519.

[3] Mathaisel D,Comm, C (1991). Course and classroom scheduling: an interactive computer graphics approach [J]. Journal of Systems and Software,1991(15):149-157.

[4] Jia Yu, Rajkumar Buyya. Workflow schdeduling algorithms for grid computing[EB/OL]. [2012—04—20]. <http://www.buyya.com/papers/MHS—Springer—Jia2008.pdf>, pp:110-149.

[5] Kennedy J, Eberhart R C. Particle swarm optimization. Proceedings of IEEE Int'l. Conf on Neural Networks;IEEE,1995;1942-1948.

[6] Y Gil, E Deelman, M Ellisman, et al. Examining the challenges of scientific workflows [J]. Computer, 2007;368-373. 40(12).

[7] Suraj Pandey. Scheduling and management of data intensive application workflows in grid and cloud computing[D]. Melbourne: The University of Melbourne, 2010.

作者简介:

赵 轩 男,(1984—),硕士.研究方向为分布式计算、多代理与博弈论方面的研究;

蔚承建 男,(1957—),博士,教授,硕士生导师.研究方向为云计算、演化计算、人工智能多代理博弈论方面的研究.

王 开 男,(1980—),博士,副教授.研究方向为非线性信号处理的研究.

刘 凯 男,(1959—),博士,教授.研究方向为计算机应用.