

粒子群优化算法在网格工作流调度中的应用

徐志立, 王红霞

(北京青年政治学院 计算机系, 北京 100102)

摘 要: 为了提高网格工作流管理系统的性能, 将粒子群优化算法(PSO)引入到网格工作流的调度策略中. 分析算法的基本原理, 根据网格工作流调度的问题对其进行变形, 提出基于粒子群优化算法的网格工作流调度策略, 并与基于 Dijkstra 的网格工作流调度算法进行对比实验. 实验数据表明, 粒子群优化算法在网格工作流调度中的性能较好.

关键词: 网格工作流, 粒子群优化, 调度

中图分类号: TN911. 22

文献标识码: A

文章编号: 1000-7180(2011)11-0065-04

Application of Particle Swarm Optimization Algorithm to Grid Workflow Scheduling

XU Zhi-li, WANG Hong-xia

(Computer Science Department, Beijing Youth Politics College, Beijing 100102, China)

Abstract: The particle swarm optimization algorithm has been applied to grid workflow scheduling in order to improve the property of the grid workflow management system. We proposed the grid workflow scheduling strategy based on particle swarm optimization algorithm. And the contrast experiment with grid workflow scheduling algorithm based on Dijkstra was done to test the swarm optimization algorithm's better property.

Key words: grid workflow; particle swarm optimization; scheduling

1 引言

网格工作流是一种新型工作流管理系统, 它旨在利用网格环境中丰富的、异构的和分布的资源, 创建一种管理网格中的高性能计算能力的协同管理机制, 实现动态变化的多个虚拟机构间的资源共享和协同工作^[1]. 由于网格工作流与传统工作流存在较大差异, 在网格环境下, 工作流负责处理更加复杂庞大的科学计算, 任务的调度机制尤为重要, 调度策略的优化能够大大提高网格工作流管理系统的性能.

一个工作流的调度过程是一种映射, 调度的目的是找到一个调度实例, 以达到调用执行服务时间最短的目的. 通过深入研究粒子群优化(Particle Swarm Optimization, PSO)算法的进化原理, 并根

据近年来 PSO 算法在求解旅行商(TSP)问题中的应用^[2-3], 将 PSO 算法应用于网格工作流的调度.

2 粒子群算法的基本原理

PSO 算法的求解过程就是通过个体极值(P_{Best})和全局极值(G_{Best})不断调整飞行来寻求最好位置的过程. 所谓个体极值就是每个粒子在飞行过程中所经历过的最好位置, 即粒子本身找到的最优解; 全局极值就是整个群体所经历过的最好位置, 即整个群体目前找到的最优解. 通过由优化问题所决定的适应度函数值(fitness value)来评价粒子的“好坏”程度. 每个粒子都通过个体极值和全局极值不断更新自己, 从而产生新一代群体. 在求解过程中, 每个粒子的行为总是追随着当前的最优粒子在解空间中

搜索^[4-5].

在 PSO 算法中,每一个可行解都对应着粒子具有的速度和位置.粒子通过不断调整自己的位置来搜索新解,搜索到的最好解记作 P_{id} ,整个粒子群经历过的最好位置记作 P_{gd} ,即目前搜索到的最优解.每个粒子都有一个速度和位置,速度记作 V ,位置记作 X . 设:

$X_i^t = (x_{i1}^t, x_{i2}^t, \dots, x_{iD}^t)$ 为粒子 i 在 t 代的位置;

$V_i^t = (v_{i1}^t, v_{i2}^t, \dots, v_{iD}^t)$ 为粒子 i 在 t 代的飞行速度;

$P_i^t = (p_{i1}^t, p_{i2}^t, \dots, p_{iD}^t)$ 为粒子 i 所经历过的具有最好适应值的位置,也就是粒子 i 在 t 代前所经历的最好位置,称为个体最好位置.对于最小化问题,目标函数值越小,对应的适应值越好.

$P_g^t = (p_{g1}^t, p_{g2}^t, \dots, p_{gD}^t)$ 为整个粒子群在 t 代前所经历的最好位置,也称为全局最好位置.

粒子根据下面的公式来更新自己的速度和位置:

$$\begin{cases} v_{id}^t = \omega v_{id}^{t-1} + c_1 r_1 (p_{id}^t - x_{id}^{t-1}) + c_2 r_2 (p_{gd}^t - x_{id}^{t-1}) \\ x_{id}^t = x_{id}^{t-1} + v_{id}^t \end{cases} \quad (1)$$

其中, v_{id} 为第 i 个粒子在第 d 维上的速度, x_{id} 为第 i 个粒子在 d 维上的位置, ω 为惯性权重, r_1 和 r_2 为均匀分布于 $[0, 1]$ 区间的随机数, c_1 和 c_2 代表加速度限定因子.

公式 (1) 由粒子先前的速度、认知部分 (Cognition Modal)、社会部分 (Social Modal) 三部分组成:粒子先前的速度说明了粒子目前的状态,起到了平衡全局和局部搜索的能力;认知部分表示从当前点指向此粒子自身最好点的一个矢量,使粒子有了足够强的全局搜索能力,避免局部极小;社会部分表示从当前点指向种群最好点的一个矢量,反映了粒子间的协同合作和知识的共享,体现了粒子间的信息共享.在这三部的共同作用下粒子才能有效到达最好位置,共同决定了粒子的空间搜索能力.

对 PSO 算法全局和局部搜索能力的平衡主要由惯性权重 ω 控制,较大的 ω 值有利于全局搜索,较小的 ω 值有利于局部搜索.选择一个合适的 ω 值能够平衡全局和局部搜索能力,以最少迭代次数找到最优解.

c_1 和 c_2 用来控制粒子自身记忆和同伴记忆之间的相互影响,选择合适 c_1 和 c_2 值可以避免陷入局部极小值,提高算法速度.

粒子在最大速度 ($V_{\max}$) 和最小速度 ($V_{\min}$) 的限

制范围内不断根据速度调整自己的位置时.当 V_i 超过 $V_{\max}$ 时将被限定为 $V_{\max}$,当 V_i 小于 $V_{\min}$ 时将被限定为 $V_{\min}$.如果 $V_{\max}$ 太大,粒子可能飞过最优解的位置;如果 $V_{\max}$ 太小,可能降低粒子的全局搜索能力,所以 $V_{\max}$ 的选择不应超过粒子的宽度范围.

3 PSO 算法在网格工作流调度中的应用

网格工作流的过程建模将网格工作流表示为一个 DAG (Directed Acyclic Graph) 模型,定义 $W = (\text{Nodes}, \text{Edge})$, Nodes 为工作流的节点集合 $\{n_1, n_2, \dots, n_m\}$,其中每一个节点代表一个网格服务;Edges 为工作流中有向边的集合 $\{e_1, e_2, \dots, e_m\}$,指明了有向图服务的依赖关系, $e_m = (n_i, n_j)$ 构成了有向图的一条带权有向边.

根据 PSO 基本原理的定义,将粒子群优化算法引入到网格工作流的调度策略中,将寻找最佳调度实例的过程抽象化为粒子互相“学习”、不断更新寻找最优粒子的过程.在初始化的粒子群体中,对每次更新的群体都能找到一个全局最优粒子,对每个粒子的更新轨迹都存在一个自身历史最优粒子,算法的本质含义即为粒子不断向自身最优与全局最优粒子“学习”的过程.

下面针对 DAG 诠释“学习”的含义:一个粒子在一定条件下被定义为最优,即假设该粒子中的一个单元段也为最优段,粒子更新迭代的过程即为粒子向最优粒子最优段学习的过程.具体定义如下:

(1) 粒子的位置

根据网格工作流的形式化定义,若工作流的节点数(即网格服务)共有 N 个,根据公式(1),设定位置 $X = (n_1, n_2, \dots, n_i, \dots, n_N)$ 为某种调度方案,即从有向图源节点到目标节点服务执行的一个有向序列,则 $n_i \in \text{Nodes}, i = 1, 2, \dots, N$.

(2) 粒子的速度

速度 V 用来改变粒子位置,定义 $V = (i_k, j_k)$, $i_k, j_k \in (n_1, n_2, \dots, n_i, \dots, n_N)$, k 表示迭代次数.速度 V 是未知的,是按照随机概率 r 选取自满足条件(即后面定义的适应值)最小的最优 X 的单元段. V 实现的是插入替换一条邻接边的操作,定义将速度作用于某一个位置,则表示在该位置执行插入替换的操作;加入速度边,则修改了服务执行路径序列.

(3) 加法操作

位置 X 与速度 V 的加法操作实现了粒子的位置移动,使粒子更新进入下一个位置状态,表示为 $X' = X + V$.

针对于网格工作流的 DAG 描述形式来实现这种特殊的加法操作,将原始抽象的粒子群优化算法实例化为具体的迭代置换操作,正如粒子的速度定义.当实现插入替换操作时,需要考虑以下几种情况:

① 当随即选取的速度 V 中两个节点包含在位置 X 中时,直接替换边 (i_k, j_k) 为 X 中的一条边;

② 当速度 V 中节点包含在 X 中时,可以实现边的直接衔接.当 V 中节点不包含在 X 中时,则搜索该节点的邻接点,即对于 $V = (i_k, j_k)$,当 X 中不包含 i_k 与 j_k 时,搜索 i_k 的前驱节点与 j_k 的后继节点,当其前驱节点或继节点包含在 X 中时,实现前驱入边及后继出边的插入替换;当其前驱节点或继节点不包含在 X 中时,将 V 的最优位置直接替换到当前位置中,以确保粒子的更新;

在 DAG 有向图内, $(n_j, n_j + 1) \in E$ 为一条有向边,定义 L_{ij} 为服务节点 n_i 到服务节点 n_j 的工作流执行时间 T_c ,根据网格工作流语言的描述,在只考虑单一参数的前提下, L_{ij} 即有向图边的权值.预寻找一条服务执行路径,则位置 X 的适应值函数可表示为:

$$f(x) = \sum_{j=1}^n L n_j n_{j+1} \quad (2)$$

速度与位置相加经过一次迭代更新后,位置得到更新变换,定义粒子当前最好的位置 P_i 为:

$$p_i^{k+1} = \begin{cases} p_i^k & \text{若 } f(x_i^{k+1}) \geq f(p_i^k) \\ x_i^{k+1} & \text{若 } f(x_i^{k+1}) < f(p_i^k) \end{cases} \quad (3)$$

定义全局粒子最好位置 P_g 为:

$$p_g^k \in \{p_i^k\} = \{p_0^k, p_1^k, \dots, p_m^k\}$$

则有:

$$f(p_g^k) = \min\{f(p_0^k), f(p_1^k), \dots, f(p_m^k)\} \quad (4)$$

(4) 减法操作

位置 X_2 与位置 X_1 的减法操作结果为速度 V ,表示为:

$$V = X_2 + X_1 \quad (5)$$

位置间的减法操作表示为把速度 V 作用在粒子 X_1 上得到新位置 X_2 ,在此相当于位置与速度相加的变相形式.

(4) 乘法操作

速度的乘法操作具有概率的意义,表示为, $V' = r(X' - X)$, $r \in (0, 1)$.实现的是按照 r 的概率来截取个体或全局最好粒子列表中一条边作为变换后的速度 V' ,并执行每个粒子的置换操作.

(5) 合并操作

若干个置换序列可以合并成一个新的置换序列,定义 $\oplus$ 为两个置换序列的合并操作.

由于离散量运算的特殊性,对粒子的运动方程作了修改,取消了原有的惯性项,即公式(1)中右边的第一项,因为速度的定义使位置变化一步到位,因此它的意义不大.粒子的运动方程可描述如下

$$V = r_1(P_i - X) + r_2(P_g - X) \quad (6)$$

$$X' = X + V \quad (7)$$

那么,根据 PSO 优化算法的基本原理,每个粒子搜索迭代的过程为寻找最优位置的过程,即为寻求网格工作流最佳调度方案的过程,根据公式定义满足最好适应值记录粒子最佳位置,将此适应值定义为工作流最优调度策略下实现网格服务的总体最短执行时间,即优化目标的约束条件为:

$$s = \min(f(x)) = \min\left(\sum_{j=1}^N L n_j n_{j+1}\right) \quad (8)$$

4 算法与实验

根据前文定义,利用粒子群优化算法思想解决网格工作流的调度问题,粒子的速度根据式(6)定义来计算,即根据粒子现在当前位置与个体极值之间的置换及当前位置与群体全局极值之间的置换的迭加和.粒子位置根据公式(7)移动到一个新位置上,实现位置置换.基于粒子群算法的工作流调度实现步骤如下:

(1) 初始化微粒子群,设定参数初值与迭代次数,随机取每个粒子的位置向量 X ;

(2) 计算每个粒子的适应度值,将最初始的适应度作为个体微粒的当前最优解 P_i ,同时按公式(4)求得所有微粒的全局最优解 P_g ;

(3) 按照公式(6)与(7)更新每个粒子的速度 V 和位置 X ,并计算每个粒子新的适应度值;

(4) 对每个粒子比较当前位置与个体粒子当前局部最优位置的适应度值,如果更好,则更新 P_i ;

(5) 对每个粒子比较当前位置与群体最优位置的适应度值,如果更好,则更新 P_g ;

(6) 根据公式(6)与(7)更新粒子速度与位置;

(7) 满足终止条件,即找到足够好的位置或完成最大迭代次数时,终止;否则返回步骤(3);

(8) 选取最后一个 P_g 作为最优结果输出.

为了测试粒子群优化调度方法的正确性与适用性,分别定义了 5、10、15、20、25、30 个网格服务组成的工作流,由 3 个网格节点执行实现.随着网格服务即有向图节点数的不断增加及服务间依赖关系复杂

性的增强,测定算法找到最优执行服务序列的完成时间.并将粒子群优化算法 PSO 与 Dijkstra 算法的网格 workflow 调度算法进行比较,将两个算法的执行时间进行比对,将服务数量分别从 5 增加至 30,在不断增加网格 workflow 流程图复杂性的基础上,将两种算法执行时间与实现效果进行比较.如表 1 所示.

表 1 网格 workflow 调度算法比较

网格服务个数	5	10	15	20	25	30
PSO 算法执行结果(权重)	4	10	29	36	38	51
Dijkstra 算法执行结果(权重)	4	10	29	36	38	48
PSO 算法执行时间(ms)	8	13	15	19	22	28
Dijkstra 算法执行时间(ms)	6	15	24	36	47	55

从算法的执行结果可见,当服务节点个数较少,工作流的 DAG 结构相对比较简单时,基于 Dijkstra 的网格 workflow 调度算法执行时间相对较少,但是随着网格服务数目的增加即网格 workflow 节点增加,服务间关系复杂性增强,运用基于粒子群的网格 workflow 调度算法的执行时间明显少于基于 Dijkstra 的网格 workflow 调度算法的执行时间;另外,随着服务节点数目的增加,根据 PSO 算法所得到的最优调度序列可能为局部最优解,而并非全局最优解,所以在服务节点数为 30 的网格 workflow 的调度中,基于粒子群的调度算法得到的服务执行序列权重值稍高于最优结果.

5 结束语

文中提出一种基于粒子群的网格 workflow 调度策

略,取得了较好的系统性能.随着网格服务节点的增加和 workflow 有向图复杂度的增大,粒子群算法 PSO 体现出较明显的优势.随着网络技术的发展与网格环境的不断扩大,利用粒子群算法来解决网格 workflow 的调度问题具有十分重要的意义.

参考文献:

[1] Jinjun Chen, Yun Yang. Key research issues in grid workflow verification and validation [D]. Australia: Swinburne University of Technology, 2006: 1445-1336.

[2] Clerc W. Discrete particle swarm optimization illustrated by the traveling salesman problem [EB/OL]. [2011-01-25]. <http://www.mauriceclerc.net>.

[3] Hong Y L, Chen G L, Guo W Z. A new particle swarm optimization for TSP [C]// Proceedings of 2006 Asian Fuzzy Systems Society International Conference. Indonesia: Bali, 2006: 297-301.

[4] Kennedy J, Eberhart R C. Particle swarm optimization [C]// Proceedings of IEEE International Conference on Neural Networks. Australia, 1995: 1942-1948.

[5] Eberhart R C, Kennedy J. Swarm intelligence [M]. San Francisco: Morgan Kaufmanns, 2001.

作者简介:

徐志立 男, (1969-), 硕士, 副教授. 研究方向为图像压缩、网络安全.

王红霞 女, (1979-), 博士, 讲师. 研究方向为图像压缩、软件开发.

(上接第 64 页)

[3] Jiawei H, Micheline K. 数据挖掘概念与技术[M]. 范明, 孟小峰, 译. 北京: 机械工业出版社, 2001.

[4] 刘靖明, 韩丽川. 一种新的聚类算法—粒子群聚类算法[J]. 计算机工程与应用, 2005(20): 183-185.

[5] Kennedy J, Eberhart R. Particle swarm optimization [C]//Proc of IEEE International Conference on Neural Network. Perth, Australia. Piscataway, NJ: IEEE Service Center, 1995: 1942-1948.

[6] 黄少荣. 基于随机加速系数的粒子群优化算法[J]. 微电子学与计算机, 2010, 27(6), 114-117.

[7] Lee K Y, ElzSharkawi M A. Fundamentals of particle

swarm techniques. Modern heuristic optimization techniques with applications to power systems [C]// IEEE Power Engineering Society, 2002. USA: New York, 2002: 45-51.

[8] 何静, 周鲜成, 李桂梅. 一种新形式的微粒群优化算法[J]. 微电子学与计算机, 2010, 27(10), 117-119.

作者简介:

陈应显 男, (1975-), 博士, 讲师. 研究方向为空间数据挖掘、矿山信息管理.