

DPSO 算法在服务网格资源调度中的应用

廖 宁, 刘建勋, 王俊年
(湖南科技大学, 湖南 湘潭 411201)

摘 要: 服务网格结合了 Web 服务技术和网格技术, 代表了应用网格的发展方向。在这类网格中, 有大量的应用在运行, 这些应用又共享网格的各种资源, 资源优化调度能使这些应用获得最大的性能, 也是影响网格计算能否成功的最重要的因素之一。在建立服务网格资源调度数学模型的基础上, 应用离散微粒群优化算法(DPSO), 在给定约束下寻找服务网格环境下资源和任务间的最优的映射关系, 达到资源优化调度的目的。通过 MATLAB 进行模拟仿真实验, 验证了该方法的可行性。

关键词: 服务网格; 资源调度; 离散微粒群优化算法; 多目标优化

中图分类号: TP301.6

文献标识码: A

文章编号: 1673-629X(2009)08-0104-03

Application of Discrete Particle Swarm Optimization Algorithm to Service Grid Resource Optimization Scheduling

LIAO Ning, LIU Jian-xun, WANG Jun-nian
(Hunan University of Science and Technology, Xiangtan 411201, China)

Abstract: Service grid is a combination of Web services technology and grid technology, which is representative of the direction of the application grid's development. In these grids there are vast running applications, which are sharing all kinds of resources of grids. Resource optimization scheduling will enable these applications to maximize performance, the impact of grid computing is also the success of one of the most important factor. Presents a mathematical model based on service grid resource scheduling. Discrete particle swarm optimization algorithm (DPSO) is proposed, which search the optimal mapped relation of serving grid resource and task according to restriction. The MATLAB simulation experiment is validated, and the method is feasible.

Key words: service grid; resource scheduling; discrete particle swarm optimization; multi-objective optimization

0 引 言

服务网格^[1](Service Grid)是结合了 Web 服务技术和网格技术的一种智能化的技术,它强调应用服务集成,能够提供面向服务的创建、过滤、集群、组织、定位及选择的机制。服务网格平台是一种高层应用中间件,实现以服务为核心元素的虚拟计算环境,可将网络上需要共享的各种计算资源、存储资源、通信资源、信息资源等统一以服务的形式进行封装和接入,通过集成服务来实现所有资源共享,这些服务来自于单个机构或外部组织中完全不同的资源所组成的分布式、各

种各样的、动态的虚拟组织,它们在物理上保持分布自治的同时实现逻辑上的统一管理,以透明的方式进行资源的优化选取、按需中介和有效访问,并能够支持用户参与应用配置的新兴应用开发模式。

1 相关研究

服务网格最突出的思想就是以“服务”为中心。服务的调用实际上就是资源管理和调度^[2],服务请求者(资源使用者)各自具有不同的目标、目的、策略和需求,而服务提供商(资源拥有者)分布在世界各地,处在不同的时区,有不同的工作体制、不同的计费模型、不一样的负载能力,因此,面对大量的服务请求如何快速地处理服务请求,生成服务实例,返回服务数据。要达到这一点就必须通过有效的作业调度系统来实现对网格资源的优化调度,实现系统的全局最优化,资源调度算法的选择就显得尤为重要。

网格资源调度^[3]问题已被证明是一个 NP-hard

收稿日期:2008-12-19;修回日期:2009-03-02

基金项目:国家自然科学基金项目(60673119,90818004);国家 863/CIMS 项目(2006AA04Z152);南京大学计算机软件新技术国家重点实验室开放基金项目

作者简介:廖 宁(1982-),女,湖南湘潭人,硕士研究生,研究方向为服务计算;刘建勋,博士,教授,研究方向为工作流、服务计算、语义与知识网络;王俊年,博士,教授,研究方向为智能控制、数据挖掘。

问题,具有多变量、多约束、多阶段和多目标的特点。解决含多目标和多约束的优化问题,一般情况下,满足所有目标都是最优的结果是不存在的,大部分的研究都是采用具有相同的量纲的 n 个目标的多目标问题,根据某效用函数将对多目标进行权重分配转化为单目标问题。这种方法具有时间复杂度低、便于实现的特点。但需要较多的先验知识,而且计算效率低,缺乏灵活性与可扩展性。文中采用微粒群优化算法从平衡时间和费用两方面的约束出发,采用多目标组合最优化算法来解决这个问题,达到资源的优化调度的目的。

2 网格资源调度数学模型

网格将分散在全球范围内的闲散资源和优势资源封装为网格节点,向用户提供统一的资源访问入口和透明的资源服务,来快速形成资源配置系统,满足用户对资源共享的要求。在定义数学模型之前,对多个不同长度的任务在多个不同资源上的调度问题进行下列约定^[2]: (1)任务在调度过程中是互相独立的; (2)各种资源具有不同的属性,如运算性能、体系结构、使用费用等; (3)一个资源在同一时刻只能处理一个任务; (4)一个任务不能同时在两个资源上处理; (5)任务一旦运行,运行该任务的资源被独占,只能等到任务完成后,再执行别的任务。基于以上的约定,资源分配问题数学模型定义如下:

1) 假设网格中有 M 个资源可以使用,记作 $S = \{s_1, s_2, \dots, s_m\}$, 用户递交了 N 个任务,记作 $X = \{x_1, x_2, \dots, x_n\}$, 用户指定了任务的最长执行时间 Deadline, 以及可以支付的最大开销 Budget 值。

2) 任务 x_i 在资源 s_j 的执行需要支付的费用 ec_{ij} 和预计执行的时间 et_{ij} , 构成的集合分别对应矩阵

$$EC = \begin{bmatrix} ec_{11} & \cdots & ec_{1n} \\ \vdots & & \vdots \\ ec_{m1} & \cdots & ec_{nm} \end{bmatrix} \quad ET = \begin{bmatrix} et_{11} & \cdots & et_{1n} \\ \vdots & & \vdots \\ et_{m1} & \cdots & et_{nm} \end{bmatrix}$$

3) 任务分配给资源执行,每次分配就确定了一个调度方案,它们构成了 $X \rightarrow S$ 资源分配映射方案的集合 $MAP = \{\text{map}_1, \text{map}_2, \dots, \text{map}_k\}$, $k = m^n$, 每个映射对应一个 $n \times m$ 矩阵 R , 由 0 和 1 组成, 当任务 x_i 分配给资源 s_j 处理时 R_{ij} 的值取 1, 否则取 0。不同的方案运行总的时间开销 T 和总的费用 C 存在差异, 优化调度的目的就是在任务分配给各个资源执行过程中, 总的“耗费”最小。引入效用函数来描述用户的需求偏好, 其中 $u_T(x_i, s_j)$ 为任务 x_i 在资源 s_j 上执行的时间效用函数值, $u_C(x_i, s_j)$ 为任务 x_i 在资源 s_j 上执行的费用效用函数值。对于某个映射 map_i , 所有任务效用值和为:

$$U_C(\text{map}) = \sum_{i=1}^n u_C(i, j), U_T(\text{map}) = \sum_{i=1}^n u_T(i, j) \quad (1)$$

4) 基于费用时间约束的网格资源调度问题, 根据给定的 S, X, EC, ET 和 U , 求一个映射方案 map , 使得

$$Z = \max[U_C(\text{map}), U_T(\text{map})] \quad (2)$$

s. t. : $T \leq \text{Deadline}, C \leq \text{Budget}$

其中 $T = ET \cdot R, C = EC \cdot R$, 可以看出这是一个多目标优化问题, 文中采用离散微粒群算法来解决这个问题。

3 PSO 算法在资源调度中的应用

目前在多目标优化问题上, 多目标演化算法是其中较为成熟的方法之一, 主要采用遗传编码进行求解, 因为遗传算法并行处理各个目标, 避免了目标间求和或者优先排序处理。但是遗传算法复杂, 参数较多, 而微粒群优化算法^[4,5] (Particle Swarm Optimization, PSO 算法) 作为一种新兴的进化计算技术, 易于实现, 需要调整的参数少, 收敛速度快, 已经在许多领域得到了广泛的应用^[6,7]。

3.1 基本微粒群算法

微粒群优化算法 (PSO) 是基于群体智能理论的优化算法, 通过群体中微粒间的协作与竞争产生的群体智能指导优化搜索。PSO 保留了基于种群的全局搜索策略, 采用简单的速度位移模型, 微粒在目标搜索空间 D 中飞行, 通过跟踪两个“极值”来更新自己, 第一个就是个体当前的最优值 p_i ; 另一个是整个种群目前找到的全局最优极值 p_g 。找到这两个最优值后, 微粒根据如下的两个公式更新微粒的速度和新的位置:

$$v_{i+1} = wv_i + c_1 R_1 (P_i - x_i) + c_2 R_2 (P_g - x_i) \quad (3)$$

$$x_{i+1} = x_i + v_{i+1} \quad 1 \leq i \leq D \quad (4)$$

式中: w 称为惯性因子, 较大的 w 适于对解范围进行大范围探查, 较小的 w 有利于算法的收敛。 c_1 和 c_2 为正常数, 称为加速因子, 通常在 $[0, 2]$ 之间取值。 R_1, R_2 是 $[0, 1]$ 区间内相互独立的随机数, $v_i \in [-v_{\max}, v_{\max}]$, $x_i \in [-x_{\max}, x_{\max}]$ 。

3.2 DPSO 算法对资源调度寻优步骤

基本 PSO 算法主要适用于连续问题的优化求解, 文中根据网格资源调度这类离散型组合优化问题的特点, 对微粒的编码和速度位移模型进行了改进, 采用离散微粒群优化算法^[8] (Discrete Particle Swarm Optimization, DPSO) 进行求解。

首先资源调度问题中所有的映射集合 MAP 为整个微粒的种群, 种群中的每个微粒被看作问题的潜在

解,种群中第 i 个微粒用四元组 (map_i, x_i, v_i, p_i) 表示,其中 map_i 为任务分配给资源执行的一个调度方案, x_i 表示该方案在映射集合 MAP 的空间位置; v_i 表示粒子 i 当前的速度; p_i 表示粒子 i 自身搜索过的个体最好位置。用 DPSO 实现网格调度的资源调度步骤为:

步骤 1:初始化微粒群中的所有微粒,设置微粒的初始位置与初始速度。计算每个微粒的适应值,将第 i 个微粒的当前位置作为初始个体极值 p_i ,将所有微粒中适应值最优的微粒位置作为 p_g 。

步骤 2:根据式(3)计算出粒子 x_i 的速度,式(4)计算出粒子 x_i 位置偏移量,若微粒飞出边界,则 x_i 等于边界值,最后通过向上取整来求得 x'_i ,即微粒飞向第 x'_i 个映射 $map_{x'_i}$ 。

步骤 3:计算出当前微粒的适应值,如果微粒 i 的适应值优于 p_i 的适应值,则该微粒的 p_i 更新为当前位置,如果某微粒的适应值优于 p_g 的适应值,则 p_g 更新为该微粒的当前位置。

步骤 4:判断算法是否满足停止条件,如未满足,则返回步骤 3 继续搜索。

步骤 5:输出全局最优适应值及其位置,将寻到的微粒最优位置,赋给网格调度模型。

4 实验仿真与分析

文中提出基于微粒群算法的网格资源分配优化方法,以达到满足用户时间和费用的两方面服务质量要求,算法用 MATLAB7.0 在 P43.0G, 512M 计算机上进行了仿真实验。根据仿真的需要生成不同的资源数量、任务数量、每个任务的预测执行时间及费用等参数。运用 DPSO 寻找最佳方案时,种群规模为 20 个微粒,迭代次数为 100,学习因子取为 $c_1 = c_2 = 1.5$,权重数 w 为 $0.4 \sim 0.9$;当任务个数越少时 w 就取小值,个数越大, w 取值就越大。选取标准遗传算法^[9]作为对比算法。图 1 反映了两种算法在资源数固定的情况下,随任务数量的变化,算法运行时间的差别,实验结果表明 PSO 算法相较于 GA 算法能够较快收敛。表 1

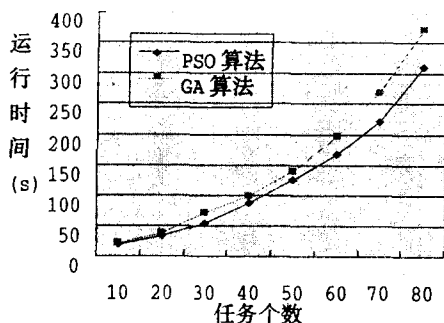


图 1 在资源数固定情况下两种算法的运行时间比较

和表 2 分别反映两种算法在资源数和任务数固定情况下,总执行时间和总执行费用的差别。实验结果表明 PSO 算法的寻优结果优于 GA。

表 1 资源数为 15 时的任务执行情况
(时间/s, 费用/cent)

任务数	算法	PSO	GA
60	总执行时间	27.3	28.5
	总执行费用	266.5	270.1
80	总执行时间	39.5	40.3
	总运行费用	326.3	337.2
100	总执行时间	48.3	49.4
	总运行费用	424.7	443.2

表 2 任务数为 100 时的任务执行情况
(时间/s, 费用/cent)

资源数	算法	PSO	GA
15	总执行时间	48.3	49.4
	总执行费用	424.7	443.2
25	总执行时间	37.3	41
	总运行费用	417.4	423.5
35	总执行时间	30.9	35.4
	总运行费用	414.6	424.2

5 结束语

在分析网格资源调度间各种约束关系的基础上,建立了调度问题模型和目标模型,并采用离散微粒群优化算法队问题模型进行求解,通过实验验证了该算法在解决多目标优化调度的优越性。但是由于网格资源调度问题在实际执行时,除了考虑时间和费用开销外,还应考虑负载均衡等问题,因此,在后面的工作中,还将针对这些问题开展进一步的研究。

参考文献:

- [1] Ding Jingbo, Tong Weiqin. A Service Scheduling System in Service Grid Environment [C] // Proceedings of the IEEE Congress on Industrial Informatics. [s.l.]: [s.n.], 2003: 413-418.
- [2] Zou Deqing, Qiang Weizhong, Shi Xuanhua. A Formal General Framework and Service Access Model Service Grid [C] // Proceedings of the IEEE Congress on Engineering of Complex Computer Systems. [s.l.]: [s.n.], 2005: 349-356.
- [3] Buyya R, Abramson D. Economic models for resource management and scheduling in Grid computing [J]. The Journal of Concurrency and Computation: Practice and experience, 2002, 14(13-15): 1507-1542.
- [4] Clerc M, Kennedy J. The particle swarm: explosion, stability, and convergence in a multi-dimensional complex space [J]. IEEE Transaction on Evolutionary Computation, 2004 (6): 58-73.

(下转第 109 页)

$B - \{a_i\}$; 否则 a_i 不能被约简, B 不变。

最后得到的 B 即为信息系统的约简。

定理4 IQBARK 算法的时间复杂度是 $O(|A|^3 \times |U|^2)$ 。

证明:

(1) 计算约简需要计算一次 $I(A)$ 。

(2) 最坏情况下需要计算 $|A|$ 次 $I(a_i)$ 和 $|A|$ 次 $I(A - \{a_i\})$ 。

(3) 为了计算 $I(A)$ (计算 $I(A)$ 和计算 $I(A - \{a_i\})$ 的时间复杂性相同), 需要求下列计算:

① 计算 $|A|$ 个覆盖, 所需时间复杂性为 $O(|A| \times |U|^2)$;

② 为了计算 $I(A)$ 和 $I(A - \{a_i\})$, 需计算 $|A|$ 和 $|A| - 1$ 次交运算, 所需时间复杂性为 $(|A| + |A| - 1) \times O(|U|^2)$ 。因此, 计算一次 $I(A)$ 的时间复杂性为 $O(|A|^2 \times |U|^2)$ 。

所以, 整个算法的时间复杂性为: $(|A| + |A| - 1) \times O(|A|^2 \times |U|^2) = O(|A|^3 \times |U|^2)$

例1 表1给出了一个信息系统。

表1 信息系统

U	a_1	a_2	a_3
x_1	1	2	1
x_2	3	2	2
x_3	1	1	2
x_4	2	1	3
x_5	3	3	2
x_6	3	2	3

解: 下面使用所给的 IQBARK 算法求解约简。

步骤1: 计算 $I(A)$ 。

由于 $J_A = \{\{x_1\}, \{x_1, x_2, x_3\}, \{x_3\}, \{x_3, x_4\}, \{x_1, x_2, x_3, x_5\}, \{x_1, x_2, x_3, x_4, x_6\}\}$, 所以 $I(A) = 40/9$ 。

步骤2: 计算 $I(a_i)$ 。

$I(a_1) = 91/36, I(a_2) = 97/36, I(a_3) = 95/36$,

所以, $I(a_1) \leq I(a_2) \leq I(a_3)$, 把 a_1 选入;

步骤3: 令 $B = A$, 计算 $I(B - \{a_1\})$, 由于 $I(B -$

$\{a_1\}) = I(\{a_2, a_3\}) = I(A)$, 所以 a_1 约简, 令 $B = B - \{a_1\} = \{a_2, a_3\}$; 而对其余的 $a_i, I(B - \{a_1\}) \neq I(A)$, 所以不能约简。因此, 最后约简为 $\{a_2, a_3\}$ 。

4 结束语

从粗糙集理论出发, 研究了优势信息系统的属性约简算法, 给出了优势关系下信息量的概念, 提出了知识粒度与信息量之间的关系, 即信息量随着知识粒度的变小而单调增加, 并对其进行了量化分析。在此基础上, 提出了基于信息量的属性约简算法, 为进一步研究约简算法提供了理论依据。

参考文献:

- [1] Miao Duoqian, Hu Guirong. A heuristic algorithm for reduction of knowledge[J]. Journal of Computer Research and Development, 1999, 36(6): 681-684.
- [2] 张文修, 梁怡, 吴伟志. 信息系统与知识发现[M]. 北京: 科学出版社, 2003.
- [3] 李鸿. 一种基于粗糙熵的知识约简算法[J]. 计算机工程与应用, 2005, 36(14): 78-80.
- [4] 李红梅, 周桂红, 王克俭. 基于粗糙集和遗传算法的知识发现方法[J]. 计算机技术与发展, 2007, 17(8): 76-79.
- [5] 纪滨. 粗糙集理论及进展的研究[J]. 计算机技术与发展, 2007, 17(3): 69-72.
- [6] Miao Duoqian, Wang Jue. An information-based algorithm for reduction of knowledge[R]. Beijing: Inst. of Autom., Acad. Sinica, 1997: 1155-1158.
- [7] Matarazzo G S, Slowinski R. Rough approximation of a preference relation by dominance relations[J]. European Journal of Operational Research, 1999, 117: 63-83.
- [8] 贾戎莉. 信息系统上的优势关系与保序关系[J]. 山西师范大学学报, 2005, 19(2): 14-16.
- [9] Liang Jiye, Li Deyu. Information Measures of Roughness of Knowledge and Significance of Attribute in Rough Set Theory[J]. Journal of Engineering Mathematics, 2005, 17: 106-108.

(上接第106页)

- [5] Kennedy J, Eberhart R. Particle swarm optimization[C]// Proc IEEE Int Conf on Neural Networks. Perth: [s. n.], 1995: 1942-1948.
- [6] 卜艳萍, 俞金寿. 离散微粒群优化算法在网格任务中的应用[J]. 计算机仿真, 2008, 25(4): 175-178.
- [7] 王雅琳, 王宁, 阳春华, 等. 求解任务分配问题的一种离

散微粒群算法[J]. 中南大学学报, 2008, 29(3): 571-576.

- [8] 马驰, 阮秋琦. 基于离散微粒群优化算法的 SVM 参数选择[J]. 计算机技术与发展, 2007, 17(12): 20-23.
- [9] 李钧, 王忠群, 刘涛. 基于遗传编程的网格资源调度算法[J]. 计算机技术与发展, 2008, 18(2): 129-132.