

一种多 QoS 需求的资源优化算法

孙志刚

(浙江传媒学院, 浙江 杭州 310018)

摘 要: 用户的多 QoS 需求,使得网络服务中的动态资源属性值变化频繁,传统方法难于对这些动态资源进行分配.为解决这一问题,提出了一种处理多 QoS 需求的动态资源优化算法.算法引入混沌理论,结合优化算法与逼近理想解排序思想,将多个动态资源形成一个链表在这些资源的各个 QoS 需求间进行选择调度,并在迭代过程中使用并行算法来降低算法陷入局部最优解的可能性.实例验证了该算法求解该问题的可行性和有效性,具有较好的实际应用价值.

关键词: QoS; 资源优化; 混沌理论; 优化算法; 逼近理想解排序

中图分类号: TN402

文献标识码: A

文章编号: 1000-7180(2013)05-0153-04

A Resource Optimization Algorithm Based on Multi-QoS Requirements

SUN Zhi-gang

(Zhejiang University of Media and Communications, Zhejiang 310018, China)

Abstract: Research the multi-QoS requirements of users. This paper puts forward a dynamic resource optimization algorithm based on multi-QoS requirements. Algorithm with chaos theory, combined with optimization algorithm and approximating the ideal solution sort thought. More dynamic resource form a linked list which is selected scheduling among QoS requirements of these resources, and parallel algorithms use iterative process to reduce the algorithm into a local optimum solution. The experiment results show that the improved algorithm is feasible and effective to solve the multi-QoS requirements problem, and has a certain practical value.

Key words: QoS resource optimization; chaos theory; optimization algorithm; approximating the ideal solution sort

1 引言

随着网络的普及和信息技术的飞速发展,网络信息资源容量呈级数式增长,但信息来源分散,质量参差不齐,缺乏必要的组织与控制,严重影响了信息资源的获取及使用,另外用户的 QoS 需求也呈现出多样性,为保障用户服务质量,资源优化配置成为信息管理者所关注的问题.

为了解决这一问题,文献[1]结合公平调度算法,提出了一种预测非线性任务工作量的方法,虽然其优化算法简单,但容易产生碎片,资源利用率较低,尤其对于大量的动态资源.文献[2]通过绘制网格来提供服务的流程,并能根据用户定义的参数和

首选项提供出 peer-to-peer 的服务质量;文献[3]依据计算几何概念制定了一套满足服务质量要求的资源调度策略,解决了资源提前预留机制中产生的碎片问题,但它们是仅基于某单一任务在较少的任务完成时间内,兼顾到服务质量需求的.随着用户 QoS 需求的增多,这些技术已经难以实现对网络资源的优化分配.

本文针对多任务下多 QoS 需求的资源优化问题,从服务角度出发,将计算资源看成服务,结合不同用户对 QoS 需求的不同倾向性,描述了一种求解资源服务分配的 QoS 优化算法,以满足不同用户的 QoS 需求,并保障全局服务质量.

2 算法基本思想

对于多任务多 QoS 需求的资源分配问题,首先根据 QoS 参数的满意度将多个服务任务划分为三类:硬性级,软性级和尽力级^[4-6],完成资源的初次调度.其中硬性级服务任务的 QoS 需求必须首先得到满足. QoS 需求参数选取安全性(security)、时效性(timeliness)、精准性(precision).引入混沌理论,采用混沌粒子群优化算法对 QoS 需求进行多目标优化求解.然后对优化 QoS 需求后的同级别服务任务,根据其所需资源对 QoS 需求的满意度进行逼近理想解排序,得到一个资源优化链表,从而实现对多 QoS 需求的动态资源优化调度.

2.1 多 QoS 需求满意度

(1) 安全性满意度

资源 r_j 分配到任务 t_i 上第 k 个执行时,该任务的安全效益为:

$$U_{sec}(i, j) = \begin{cases} 1, & T_{t-ufe}(i) \leq T_{r-ufe}(j) \\ 1 - \frac{T_{t-ufe}(i) - T_{r-ufe}(j)}{T_{t-ufe}(i)}, & \text{else} \end{cases} \quad (1)$$

其中: $T_{t-ufe}(i)$ 、 $T_{r-ufe}(j)$ 分别表示任务 t_i 的安全性需求及资源 r_j 提供的安全性水平.

(2) 时效性满意度

当用户提交某一任务 $t_i (t_i \in T, i = 1, 2, \dots, M)$ 时,该任务被选择调用到资源 $r_j (r_j \in R, j = 1, 2, \dots, N)$ 上运行的执行效率满意度.由运行时间满意度与计算规模满意度共同确定.

① 设任务的最迟完成时间为 $timel_{max}(i)$ 、任务分配到资源上的期待执行时间为 $timel_{expect}(i, j)$, 则其运行时间满意度为:

$$U_{makespan}(i, j) = \begin{cases} 1, & timel_{expect}(i, j) \leq timel_{max}(i) \\ power(a, timel_{max}(i) - timel_{expect}(i, j)), & (2) \\ \text{else} \end{cases}$$

其中, $power()$ 为指数函数, $a > 1$ 为常数.

② 设任务 t_i 自身的计算规模为 $T_{self}(i)$, 资源 r_j 的最大处理规模为 $R_{c-max}(j)$, 则计算规模的满意度为:

$$U_{\sigma} = \begin{cases} power[a, (T_{self}(i) - R_{c-max}(j))], & T_{self}(i) \leq R_{c-max}(j) \\ 0, & \text{else} \end{cases} \quad (3)$$

采用加权方法得出时效性的综合满意度为:

$$U_{timel}(i, j) = \begin{cases} \omega_1 \times U_{makespan}(i, j) + \omega_2 \times U_{\sigma} \\ \omega_1 + \omega_2 = 1 \end{cases} \quad (4)$$

取 $\omega_1 = 0.8, \omega_2 = 0.2$.

(3) 精准性满意度

将任务 t_i 分配到资源 r_j 上时,资源能够保障任务 QoS 需求的精度程度而给用户带来的满意度.假设任务 t_i 的最小精度 QoS 需求为 $T_{t-precision}(i)$, 资源 r_j 提供的精度 QoS 需求为 $T_{r-precision}(j)$, 则精准性满意度为:

$$U_{precision}(i, j) = \begin{cases} 1, & T_{t-precision}(i) \leq T_{r-precision}(j) \\ 1 - \frac{T_{t-precision}(i) - T_{r-precision}(j)}{T_{t-precision}(i)}, & \text{else} \end{cases} \quad (5)$$

2.2 评价函数

假设所有任务完成的总时间为 Time, CPU 的使用总费用为 Cost.

$$\text{Time} = \max\{NC_1/V_1, NC_2/V_2, \dots, NC_n/V_n\} \quad (6)$$

$$\text{Cost} = \sum_{i=1}^n C_i \times NC_i/V_i \quad (7)$$

其中, NC_i 为资源 r_j 的总 CPU 周期, V_i 为该资源的访问速率, C_i 为使用该资源的费用.

3 算法实现

3.1 混沌粒子群优化算法

资源优化问题属于一个目标规划问题.求解规划问题的优化算法有很多,这里选用混沌粒子群优化算法(CPSO)来实现.为求出较优的粒子作为初始粒子,在初始化时事先进行混沌序列化.本文采用 logistic 方法得出混沌化序列:

$$x_{n+1} = \mu x_n(1 - x_n) \quad n = 0, 1, \dots \quad (8)$$

其中, μ 为控制参数; x 为状态变量; x_n 记录第 n 代的状态; $x_0 \in [0, 1]$ 为初始值.

算法流程如下:

① 混沌初始化形成粒子位置和粒子速度,计算初始粒子个体极值和粒子群全局极值.选取最优的个数混沌变量作为粒子初始种群.

② 以目前粒子位置作为混沌变量,更新混沌序列,并计算、评价适应度 F_1 .

③ 以粒子初始种群为基础更新粒子群,并计算评价规则下满足资源约束的适应度 F_2 .

④ 比较两个适应度 F_1 和 F_2 ,保留较优的解.更新粒子局部最优和全局最优.

⑤ 若达到最大迭代次数,则返回全局最优解 $gbest$;否则跳至步骤 ②。

3.2 逼近理想解排序算法

考虑到各个资源对 QoS 满意度的重要性程度不同,构造出每个任务的相应资源,即所有候选资源集 R 对各个 QoS 需求的满意度矩阵 U_{TR} ,计算正理想解 $E^+ = \{E_1^+, E_2^+, \dots, E_N^+\}$ 和负理想解 $E^- = \{E_1^-, E_2^-, \dots, E_N^-\}$,其中 $E_j^+ = \max_{1 \leq i \leq M} U(i, j)$, $E_j^- = \min_{1 \leq i \leq M} U(i, j)$,则候选资源与正、负理想解之间的距离分别为:

$$d_i^+ = \sum_{j=1}^N d(U_{ij}, E_j^+) = \sum_{j=1}^N \sum_{j=1}^N \sqrt{\frac{1}{3}} (U_{ij} - E_j^+)$$
$$i = 1, 2, \dots, M \tag{9}$$

$$d_i^- = \sum_{j=1}^N d(U_{ij}, E_j^-) = \sum_{j=1}^N \sum_{j=1}^N \sqrt{\frac{1}{3}} (U_{ij} - E_j^-)$$
$$i = 1, 2, \dots, M \tag{10}$$

则候选资源与理想解的相对贴近度为:

$$RU_i = \frac{d_i^-}{d_i^+ + d_i^-} \quad i = 1, 2, \dots, M \tag{11}$$

按照 RU_i 的大小对各任务排序,依次按照任务的初次资源调度进行执行,达到最后的优化资源分配目的。

4 仿真实验与结果分析

实验在仿真平台 GridSim 上进行,参数设置如下: $T_{wait}(j)$ 、 $T_{self}(i)$ 分别为 $[0, 10^5]$ 、 $[10^1, 10^5]$ 上的随机数, $R_{max}(j)$ 为 $[10^1, 10^5]$ 上的随机数. 为控制任务与资源间的服务级别,设置变量 $V_Q(1 \leq V_Q \leq 4)$ 和 $r_1(0 \leq r_1 \leq 1)$. 若 $r_1 < 0.25V_Q$,则为硬性级服务任务. 若 $r_1 < 0.5V_Q$,则为软性级;否则为尽力级. $U(prio, time, precision, sec, reli) - min(i)$ 满意度指标下限设为 0.3.

(1) 将资源总数设置为 5,任务数分别取 10、20、30、40、50、60、70、80、90、100,依次递增,其实验结果如表 1 所示。

表 1 资源数一定而任务数不同的实验结果

任务数	Time	Cost
10	650.179	224.786
20	1637.583	368.169
30	2150.389	565.174
40	3225.498	815.937
50	4114.889	1053.597
60	4829.276	1198.645
70	5300.299	1400.268
80	5901.256	1647.947
90	6752.789	1864.278
100	8091.795	2036.592

由表 1 数据得到图 1. 可以看出,在资源数一定的情况下,随着任务数目的增大,完成所有任务所花费的总费用,以及总的运行时间都将增大,这与理论相一致。

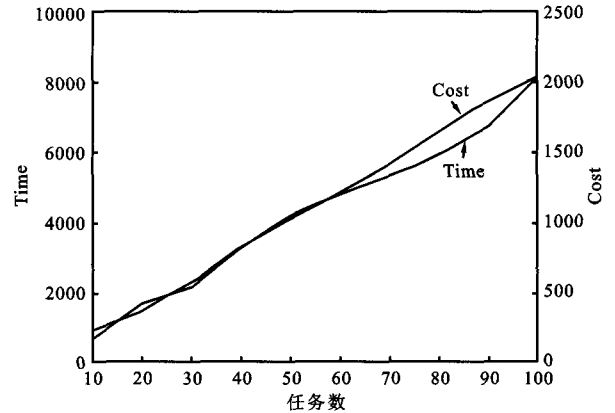


图 1 资源数一定而任务数不同的实验结果图

(2) 当任务数固定为 50,资源数分别取 10、15、20、25、30 时,其实验结果如表 2 所示。

表 2 任务数一定而资源数不同的实验结果

任务数	Time	Cost
10	3017.337	941.809
15	2879.483	892.245
20	2813.809	886.824
25	2987.382	907.849
30	2803.765	901.172

将表 2 数据绘制成如图 2 所示. 可以看出,当任务数固定不变时,随着资源数目的增多,完成所有任务所花费的总费用和总时间都呈现出先减后增的趋势. 这是因为,任务调度与资源匹配是一个相互选择的动态过程:如果资源少,则增加资源可以较好的节约运行时间与费用;但当资源增加到一定程度时,就会增加任务和资源的匹配过程,从而使得总运行时间和总费用出现增大现象。

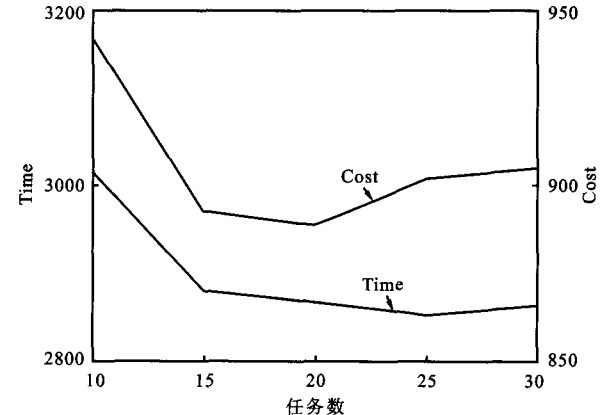


图 2 任务数一定而资源数不同的实验结果图

通过以上分析,可以得出结论:在资源优化问题上,本文算法能够很好的结合用户不同 QoS 需求的变化,及时调整资源调度方案,更有效的利用系统资源,它不仅能完成多任务在多资源上的分配,还能保障用户的多 QoS 需求.

5 结束语

本文针对多任务多 QoS 需求的动态资源优化分配问题,提出了一种优于传统算法的资源调度策略,为解决网络个性化服务提供了一个有效方法. 实验说明,本文算法(CPSOD)比传统以期待执行时间进行调度的 Min-Min 算法更具优势,取得了较为理想的结果. 该算法不仅可以有效地分配资源,还能满足用户 QoS 需求的多样性.

参考文献:

- [1] Doulamis N, Doulamis A, Litke A, et al. Adjusted fair scheduling and non linear work load prediction for QoS guarantees in grid computing[J]. Computer Com-

munications, 2007, 30(3): 499-515.

- [2] Kyriazis D, Tesrpas K, Menychtas A, et al. An innovative workflow mapping mechanism for Grids in the frame of quality of service [J]. Future Generation Computer Systems, 2008, 24(6): 498-511.
- [3] Castillo C, Rouskas G N, Harfoush K. Online algorithms for advance resource reservations [J]. Journal of Parallel and Distributed Computing, 2011, 71(7): 963-973.
- [4] 丁菁, 陈国良, 顾钧. 计算网格环境下一个统一的资源映射策略[J]. 软件学报, 2002, 13(7): 1303-1306.
- [5] 莫赞, 谢娜, 贾功祥, 等. 基于多 QoS 需求驱动的网络资源调度研究[J]. 计算机应用研究, 2012, 29(10): 3904-3907.
- [6] 曾建潮, 介婧, 崔志华. 微粒群算法[M]. 北京: 科学出版社, 2004.

作者简介:

孙志刚 男, (1979-), 硕士. 研究方向为计算机网络.

(上接第 152 页)

序关系分析法较其他权重确定方法而言, 具有简单、准确、可靠等特点. 本文采用序关系分析法确定房地产预警指标的主观权重. 实验结果进一步表明: 采用序关系分析法确定房地产预警指标的权重是合理的, 计算过程简单, 也容易计算机编程.

(2) 本文采用向量投影法对传统的 TOPSIS 法进行改进, 改进后的 TOPSIS 法能有效避免传统的 TOPSIS 的不足, 使改进后的方法 TOPSIS 法更加科学、合理性, 且模型计算相对简单.

本文的方法计算简便、思路清晰、计算结果合理, 还可应用于项目决策、其他方面的性能评价等方面, 具有广泛的应用前景.

参考文献:

- [1] 张先起, 梁川. 改进的 TOSPSIS 模型及其在黄河置换

水量分配中的应用[J]. 四川大学学报: 工程科学版, 2006.

- [2] 郭亚军. 综合评价理论、方法及应用[M]. 北京: 科学出版社, 2007.
- [3] 岳超源. 决策理论与方法[M]. 北京: 科学出版社, 2009.
- [4] 王永兰, 孙雷. 改进的 TOSPSIS 法在油气田开发中的应用[J]. 西南石油大学学报: 自然科学版, 2008.
- [5] 周荣喜, 徐建荣. 基于离差最大化和 OWGA 算子的多属性群决策方法[J]. 统计与决策: 理论版, 2007(1): 132-133.

作者简介:

张彦峰 男, (1967-), 博士研究生. 研究方向为决策支持系统、知识系统、管理信息系统工程.

鲁艳玲 女, (1982-), 博士研究生. 研究方向为管理信息系统工程.