第 35 卷 第 7 期 计 算 机 工 程 2009 年 4 月
Vol.35 No.7 Computer Engineering April 2009
·软件技术与数据库· 文章编号：1000—3428(2009)07—0052—03 文献标识码：A 中图分类号：TP393

用户 QoS 及系统指标指导的计算网格任务调度

龚红翠[1]，于 炯[1,2]，侯 勇[1]，刘洪伟[1]
(1. 新疆大学信息科学与工程学院，乌鲁木齐 830046；2. 北京理工大学计算机学院，北京 100081)

摘 要：针对网格环境中的动态性特点，特别是用户服务质量(QoS)需求的动态变化性，通过定义任务的效益函数来评估任务的多维 QoS 需求。同时为了兼顾任务完成时间、负载平衡等系统指标，引入系统效益的概念，给出负载平衡度的定义用来指导调度及评价调度性能。针对一组具有 QoS 需求的相互独立的计算任务提出一种用户 QoS 及系统指标指导的计算网格任务调度算法——UQSI。模拟实验结果显示，该算法能较好地满足用户的多维 QoS 需求，更加适合开放复杂的网格环境。

关键词：网格计算；任务调度；服务质量；时间跨度；负载平衡

User QoS and System Index Guided Task Scheduling in Computing Grid

GONG Hong-cui[1], YU Jiong[1,2], HOU Yong[1], LIU Hong-wei[1]
(1. School of Information Science and Engineering, Xinjiang University, Urumqi 830046;
2. School of Computer Science and Technology, Beijing Institute of Technology, Beijing 100081)

【Abstract】According to dynamic characteristic of grids, especially to the dynamic QoS request of the users, this paper describes multiple QoS attributes as utility functions. Meanwhile, considering the system indexes such as makespan, load balancing, the notion of system utility is introduced, and the load balance degree is defined to guide and evaluate the scheduling. A user QoS and system index guided task scheduling algorithm, UQSI, is presented to schedule independent tasks with multiple QoS. Simulation results reveal that the algorithm can satisfy the user's demands and improve the system's performance, and it is more suitable for the complex grid environments.

【Key words】grid computing; task scheduling; QoS; makespan; load balancing

网格技术是当今计算机领域的研究热点之一。由于资源具有广域分布、异构、动态等特征，如何对任务进行调度以满足用户的需求是一个极具挑战性的问题。任务调度是网格计算系统的一个重要组成部分，关系到网格能否高效利用资源、快速完成任务。同时它也是一个 NP-hard 问题，目前多采用启发式算法。

1 相关工作

异构计算环境下适用于元任务(meta task)调度的经典的批模式的启发算法有 Min-min, Max-min, Sufferage 等，它们的评价标准为时间跨度(makespan)和负载平衡(load balancing)，没有考虑用户的 QoS。

文献[1]提出了一种以 QoS 为指导的 Min-min 启发式网格任务调度算法。该算法基于传统的 Min-min 算法，并加入了用户 QoS 的要求。但是它只考虑了一维 QoS 的情况，并且其最终的评价标准也与传统方法类似。

文献[2-3]在 Nimrod-G 模型中提出基于时间和成本限制下的优化调度算法(DBC)，比较有效的算法有完成时间优先调度算法、成本优先调度算法、限定时间成本保守时间最优化算法等。这些方法是基于经济模型的调度算法，但通常都过分注重以用户的指标为中心并且只考虑了用户的二维 QoS 要求。

文献[4]等采用效益模型的方法，利用效益函数来衡量用户任务的 QoS，它们考虑了用户的多维 QoS 要求，但没有考虑系统指标。

而网格计算任务调度的主要目标[5]就是要对用户提交的任务实现最优调度，并设法提高网格系统的总体吞吐率。具体的目标包括最优跨度、服务质量、负载均衡等。其中，跨度是一个最主要、最常见的目标，指的是从第一个任务开始运行到最后一个任务运行完毕所经历的时间。跨度越短说明调度策略越好。当用户向网格系统提交任务后，最大的愿望是网格系统尽快完成自己的任务。可见，实现最优跨度是用户和网格系统的共同目标。

用户对资源的需求情况是通过 QoS 形式反映出来的。任务管理与调度系统在进行分配调度任务时，保障网格应用的 QoS 是完全应当的。并且判断网格的标准之一就是提供非凡的服务质量。

在开发并行和分布计算应用时，负载平衡是一个关键问题。网格系统更进一步扩展了这个问题。网格任务调度是涉及交叉域和大规模应用的调度。解决好系统的负载均衡是一个非常重要的问题。

并且资源的可用性都是基于预测模型的，如果一个计算

基金项目：国家自然科学基金资助项目(60563002)；教育部春晖计划基金资助项目(Z2005-1-65009)；新疆工业高等专科学校科研基金资助项目(WGZ2008K05)

作者简介：龚红翠(1983－)，女，硕士研究生，主研方向：分布式计算，网格计算；于 炯，教授；侯 勇，博士研究生；刘洪伟，硕士研究生

收稿日期：2008-07-29 **E-mail：**gonghc@xju.edu.cn

网格节点被分配了太多的任务而负载过度，那么一个任务的响应时间不容易得到保证。

本文采用效益函数来量化用户多维 QoS 需求及系统指标，提出了一个综合的调度标准和启发式调度算法，并通过模拟实验对其进行了验证。

2 模型建立

2.1 任务调度模型

在网格计算中，任务调度就是根据任务信息，采用适当的策略将用户任务集合 $T=\{t_1,t_2,\cdots,t_n\}$中 n 个相互独立的任务分配到异构计算资源集合 $M=\{m_1,m_2,\cdots,m_m\}$中 m 个异构可用资源上。

为此进行如下的约定:

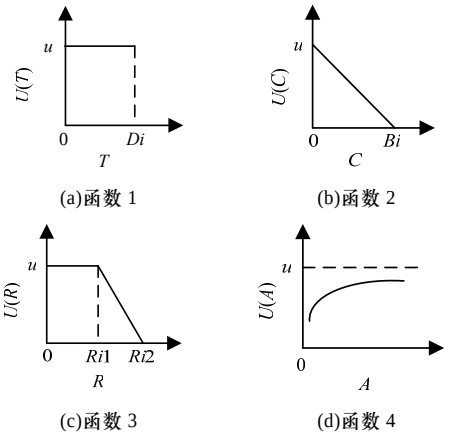
(1)进行调度的一组任务是相互独立的，即任务之间没有通信和数据依赖；一个任务不能同时在 2 个资源上处理；任务一旦运行，运行该任务的资源被独占，只能等到运行完后再执行别的任务。

(2)假设任务在每个计算节点上的运行时间已经预测得出。在网格任务调度过程中，任务在计算节点上的实际执行时间是无法提前确定的，需要对任务的执行时间进行预测，目前存在针对网格环境中任务执行时间的短期和长期预测方法，如文献[6-7]中的方法。

2.2 用户QoS 需求模型及效益模型

用户可能有多维 QoS 需求，主要包括截止期限、费用限制、可靠性、优先权等。假设对于任务集合中的每一个任务 t_i，均有 QoS 需求 $Q_i=\{Q^1_i, Q^2_i,\cdots, Q^{di}_i\}$ $(1\leqslant i\leqslant n, d_i=|Q_i|)$。同时对于每一维 QoS 需求，用效益函数表示用户从完成任务中获得的 QoS 满意程度。QoS 的度量指标包括消极度量和积极度量。对于消极度量，其值越大，效益越低，如时间；对于积极度量，其值越大，效益越高，如可靠性。

图 1 描述了不同 QoS 度量的具有代表性的效益函数图形。



(a)函数 1　(b)函数 2　(c)函数 3　(d)函数 4

图 1　QoS 的效益函数

对任务 $t_i\in T$，定义它的第 k 维 QoS 需求的效益函数为 u^k_i。

用 $u_k(i,j)$代表任务 t_i在资源 m_j上执行时的第 k 维 QoS 效益值。

对于消极 QoS 度量，归一化后的效益值为

$$U_k(i,j)=\frac{\min(u_i^k)}{u_k(i,j)} \tag{1}$$

其中，$\min(u^k_i)$代表任务 t_i在所有满足其 QoS 需求的资源上第 k 维 QoS 的最小效益值。

对于积极 QoS 度量，归一化后的效益值为

$$U_k(i,j)=\frac{u_k(i,j)}{\max(u_i^k)} \tag{2}$$

其中，$\max(u_i^k)$代表任务 t_i在所有满足其 QoS 需求的资源上第 k 维 QoS 的最大效益值。

根据优先级计算出的优先级影响的效益值作为最终的第 k 维 QoS 效益值:

$$PU_k(i,j)=(P_i/p_{\max})\times U_k(i,j) \tag{3}$$

其中，P_i为任务 t_i的优先级；$p_{\max}$为优先级的最大值。

$$U_{\text{user}}(i,j)=\sum_{k=1}^{d_i}W_{ik}PU_k(i,j) \tag{4}$$

式(4)代表任务 t_i 在资源 m_j 上的归一化的总的效益值。W_{ik}为任务 t_i的第 k 维 QoS 度量的权重且各维权重之和为 1。效益函数及权重由用户在提交任务时给定。

2.3 系统效益模型

定义 负载平衡度(load balance degree): 假设有 m 个资源，每个资源消耗的 CPU 时间分别为 C_i，则这些资源的负载平衡度

$$LBD=\frac{\sum_{j=1}^{m}c_j}{\max(c_j)\times m}$$

U_T为系统时间效益值:

$$U_T(i,j)=\frac{time_{\min}}{time(i,j)} \tag{5}$$

U_L为负载平衡效益值:

$$U_L(i,j)=\frac{\sum load_k+time(i,j)}{load_{\max}\times n} \tag{6}$$

其中，$time_{\min}$代表任务 t_i在各个满足其需求的资源上的最小执行时间;$time(i,j)$代表任务 t_i在资源 m_j上的执行时间;$\sum load_k$代表所有满足其需求的资源的当前负载之和；$load_{\max}$代表将任务预分配到资源 j 上后负载最大的那一资源;n 代表满足需求的资源的数目。显然，$U_L\leqslant 1$，并且 U_L 越逼近 1，负载越平衡。

系统的效用值:

$$U_{\text{system}}(i,j)=W_1U_T(i,j)+W_2U_L(i,j) \tag{7}$$

其中，W_1, W_2为权重，且 $W_1+W_2=1$。

例如假设 t_1在资源 m_1, m_2, m_3, m_4 上的预期执行时间为 2, 4, 6, 8，且各资源当前负载为 8, 6, 4, 2。则

$U_T(1,1)=2/2=1$, $U_L(1,1)=(8+6+4+2+2)/(10\times4)=0.55$

3 用户QoS及系统指标指导的UQSI调度算法

首先根据任务的要求对资源进行筛选，然后以式(8)作为目标函数，查找使该目标函数最大的任务/资源对进行映射。如果所有的资源都不能合格地完成该任务，则将该任务拒绝。

$$U(i,j)=W\times U_{\text{user}}(i,j)+(1-W)\times U_{\text{system}}(i,j) \tag{8}$$

其中，偏好因子 $W(0\leqslant W\leqslant 1)$的变化能够改变目标函数中 2 项权重，通过改变偏好因子来追求用户 QoS 和系统指标的不同目标。

对于被系统拒绝服务的任务可以反馈给用户。用户通过修改任务的 QoS 以使系统能够接受其服务请求，或者参与下一次调度。因为网格是一个动态的环境，各种资源随时都在变化，所以下一次就可能有资源能满足其 QoS。

UQSI 算法的详细步骤如下:

输入 任务与资源信息，***ETC*** 矩阵

输出 任务/资源映射方案

初始化：$I(i, j)$=1 表示任务 t_i 可以在资源 m_j 上运行，初始化为全 1；$Load_j$ 代表资源 j 的当前负载，初始化为全 0

收集新到达的任务到任务集 T

收集上次已经映射但未调度执行的任务到 T

查询网格资源信息

repeat

for each task t_i in T

for each machine m_j in M

if I(i, j)=1

if m_j satisfy QoS of t_i

计算 $u_k(i,j)$ $(1 \leqslant k \leqslant d_i)$, time(i, j)

else

I(i, j)=0

endfor

for each machine m_j in M

if I(i, j)=1

计算 $U_k(i,j)$

计算 $PU_k(i,j)$

计算 $U_{user}(i,j)$

endfor

if (I(i, j)=0 or $U_{user}(i,j) \leqslant 0$)

remove t_i from T

选择 $time_{min}$

预分配任务 t_i 到资源 m_j 上后选择 $Load_{max}$

计算 U(i,j)

endfor

在所有任务中找出使 U(i,j)最大的任务/资源映射对

调度该任务到该资源，并更新该资源状态和信息，包括其 $Load_j$

从任务组 T 删除该任务

until(T 为空)

4 仿真实验

4.1 实验设置

在仿真实验中考察由 20 个资源对 50~250 个独立任务构成的任务集合的调度情况。任务在资源上的预期执行时间矩阵 ***ETC*** 以 μ_{task}= 100, V_{task}=V_{mach}=0.6 生成[8]。选取了截止期限、可靠性、优先级这三维 QoS 需求，对截止期限、可靠性提供了许多常用的效用函数以便从中选取，优先级从 1, 2, 4 中随机选取。将 W, W_1, W_2 设为 0.5，在实际的网格系统中可根据情况对其进行设置。

资源的单位时间失效率 FR_j 在区间[0.000 1, 0.001 5]上随机生成。

4.2 实验结果及分析

从调度方案的用户总 QoS 效益值、makespan、负载平衡度等几个方面对本文提出的算法 UQSI 以及不考虑系统指标的多维 QoS(MQoS)算法进行比较分析。

如图 2 所示，当资源比较充足时，用户的总 QoS 效益值 MQoS 比 UQSI 稍高，但随着任务集合中任务的增多，UQSI 却较之性能有明显的提高。这是因为当任务数增多时，由于 MQoS 每次都选择用户最大的效益值优先进行调度，有可能造成任务在此资源上的执行时间较长，使得任务集合中有些待调度任务超过了其截止时间而被丢弃，被成功调度任务率降低，从而用户总 QoS 效益值降低。

如图 3、图 4 所示，由于 UQSI 算法在调度时考虑了时间跨度、负载平衡度等系统指标，每次调度时选择用户效益值和系统效益值之和最大的任务资源对进行调度，从而其时间跨度、负载平衡度较 MQoS 有很大提高。

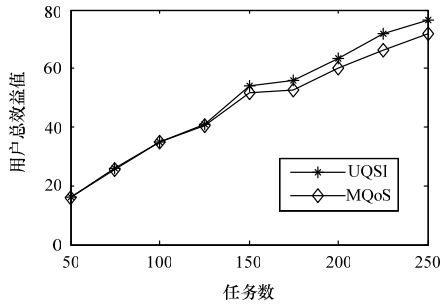


图 2 用户总效益值比较

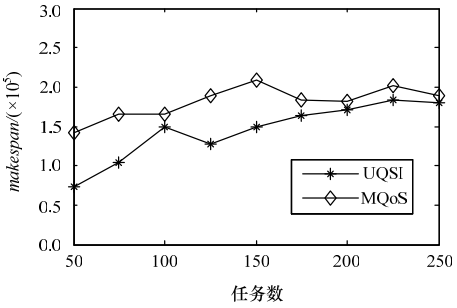


图 3 时间跨度比较

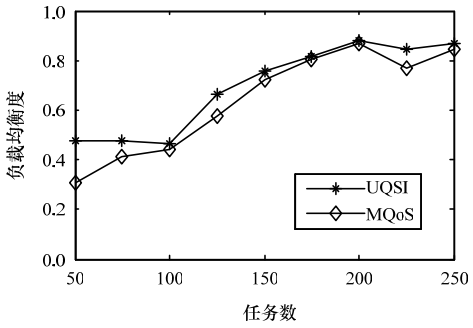


图 4 负载平衡度比较

5 结束语

用户的 QoS 要求随着调度过程的进行而变化。用户在提交作业的同时，不是单单设置了其 QoS 需求的参数，而是根据这些参数构造了效益函数。同时针对现有网格任务调度算法所存在的不足，引入系统效益的概念，综合考虑用户 QoS 以及系统指标因素，提出了一个能提高调度系统整体调度性能的调度算法。仿真实验结果表明，该算法不仅能很好地满足用户的需求，并且与不考虑系统指标的多维 QoS 调度算法相比，在时间跨度和负载平衡度方面有很大的提高。

未来的工作主要包括以下几个方面：(1)系统地建立 QoS 描述模型，包括 QoS 描述的分类以及更准确的效益函数表示。(2)用户在提交任务时各 QoS 需求权重的确定方法。

参考文献

[1] He Xiaoshan, Sun Xiaohe, Laszewski G. QoS Guided Min-min Heuristicfor Grid Task Scheduling[J]. Journal of Computer Science and Technology, 2003, 18(4): 442-451.

[2] Buyya R, Murshed M, Abramson D. A Deadline and Budget Constrained Cost-time Optimization Algorithm for Scheduling Task Farming Applications on Global Grids[C]//Proc. of the International Conference on Parallel and Distributed Processing Techniques and Applications. LasVegas,USA: [s. n.], 2002: 2183-2189.

（下转第 58 页）