

基于平衡定价和成本梯度的科学工作流调度策略

马 艳¹, 龚 斌¹, 邹立达²

(1. 山东大学计算机科学与技术学院, 山东济南 250101; 2. 山东女子学院信息技术学院, 山东济南 250300)

摘 要: 科学工作流调度在自治网格和竞争市场环境变得极具挑战性. 为了兼顾科学用户和服务提供者的需要, 提出效用网格环境下市场驱动的科学工作流调度框架; 使用平衡定价机制推导出多个资源约束情况下的服务价格, 最大化服务提供者的利润和实现资源的最优分配; 成本优化映射策略引入成本梯度因子作为服务选择的标准, 提高调度算法的优化能力和优化速度. 性能模拟表明该框架不仅实现了较高的资源节点收益和资源利用率, 还可在保证用户 QoS 要求的前提下优化不同类型科学工作流的执行成本.

关键词: 网格计算; 任务调度; 科学工作流; 服务定价

中图分类号: TP393

文献标识码: A

文章编号: 0372-2112 (2010) 10-2416-06

Equilibrium Pricing and Cost Gradient Based Scheduling Strategy of Scientific Workflow

MA Yan¹, GONG Bin¹, ZOU Li-da²

(1. School of Computer Science and Technology, Shandong University, Jinan, Shandong 250101, China;

2. Information Technology School, Shandong Women's University, Jinan, Shandong 250300, China)

Abstract: Scheduling of scientific workflow is becoming more challenging for autonomous grid and egoism of competitive market. We proposed market-driven scientific workflow scheduling framework (MSWSF) in utility grid to consider the benefits of both scientific users and service providers. Equilibrium pricing scheme derives service price under multiple resource constraints to achieve optimal allocation of resources and maximal profits of service providers. Cost optimization mapping strategy introduces a cost gradient metric as a criterion for service selection in order to improve optimization capability and speed. Performance simulation proves MSWSF not only can achieve high average profits of resource nodes and resource utilization rate, but also can optimize usage cost of different types of scientific workflows while satisfying QoS requirements.

Key words: grid computing; task scheduling; scientific workflow; service pricing

1 引言

科学工作流管理系统根据调度策略分两类^[1]: Pegasus, DAGMan, Taverna 等用性能驱动调度, 目标为最小化科学工作流执行时间, 运行在社区网格; Nimrod-G, Gridbus 等用市场驱动调度, 目标为在 QoS 约束下最小化执行成本, 运行在效用网格. 效用网格把网格环境下所有资源看作可测量和可解释的效用服务. 文献[2]将效用服务分为资源服务和应用服务. 本文研究应用服务, 其可独立执行任务且价格由服务提供者直接给出. 服务定价对资源分配、提供者利润分布和 QoS 约束调度有重要影响, 许多文献[3~7]对此展开研究. 文献[3]证明动态定价优于固定定价和随机定价, 且调度算法对参

数扫描和线性工作流有效, 但仅考虑了基于供需的单资源定价和简单应用调度. 文献[4]研究有限资源基于定价的调度来满足多维 QoS 需求, 通过多次固定定价实现任务和资源代理双赢, 但需多次迭代才能将价格调至最优. 文献[5~7]同样研究服务定价及网格资源分配, 但其关注单个资源定价或者忽略了资源的有限性. 本文根据边际效益不断递减的规律, 利用投入产出的平衡点为服务提供者制定取得最高效用的服务价格, 既考虑了多个有限资源的定价, 还在兼顾提供者利益的同时快速给出最优价格. 本文将服务提供者看作拥有有限资源的经济独立体, 为了最大化效用, 它有部署和定价服务的权利; 科学工作流则要在给定时限(deadline)内最小化执行成本.

目前给定时限的市场驱动 workflows 调度的研究各有优缺点. Buyya^[8]最早提出可分配资源和调整供需的计算经济框架 CEF, 用商品市场模型决定价格, 并在服务等级协议 SLA 下为任务选择最便宜资源调度, 既有单纯供需定价的滞后性, 又有局部实时调度的低效性. Yu^[2]将时限分到各子 workflow, 再用马尔科夫决策过程求解, 提出 Deadline Distribution (简称 DD) 调度算法, 但文献^[9]证明 workflow 划分策略并不优于 full-graph 调度, 尤其对不平衡 workflow. 文献^[10]提出基于有效路径的费用优化 DAG 调度算法, 忽略了通信时间和通信费用. Menasci^[11]提出 Back Tracking 算法, 考虑了几乎所有资源的调度, 但不适合于面向服务的体系架构. 本文以不同市场角色考虑从市场分析、服务定价、服务选择到任务映射的整个过程, 提出效用网格环境下市场驱动的科学 workflows 调度框架. 基于成本梯度的调度以动态 full-graph 方式实施, 具有较强的成本优化能力.

2 系统模型

2.1 效用网格模型

市场驱动的效用网格模型如图 1 示. 效用网格由多个自治的服务提供者 SP 组成, 每个 SP 有一套市场运作机制来保证效益最大化. 市场分析模块分析服务历史需求, 绘出需求与供给函数, 界定需求对象和规模, 对下个周期的服务定价起决定作用; 服务定价模块从市场分析模块得到各个服务的需求数据, 以最大化 SP 利润为目标, 对各个服务的价格进行界定; 根据市场分析模块的决策, 如果某项服务达不到预期收益或某项未部署服务有较好创益前景, 服务部署模块负责对服务进行撤销或配置.

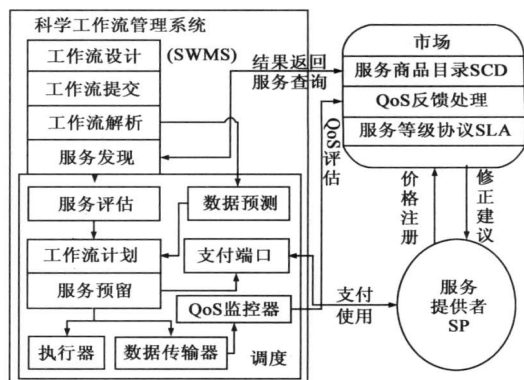


图2 市场驱动的科学 workflows 调度框架

2.2 市场驱动的科学 workflows 调度框架

市场驱动的科学 workflows 调度框架 MSWSF 由科学 workflows 管理系统 SWMS, 服务提供者 SP 和市场组成, 如图 2 示. SP 把本域的服务及属性, 包括最大化效益的服务价格, 注册到市场的服务商品目录 SCD. 市场把 SWMS

传来的 QoS 评价通过 QoS 反馈处理反馈给 SP, 作为提高服务质量和调整价格的参考. 服务等级协议 SLA 定义 SP 和 SWMS 约定的服务类型、服务质量和用户付款等要求.



图1 效用网格体系结构

MSWSF 框架的 SWMS 通过对 WFE (Work Flow Enactment Engine)^[12]改进来适应市场驱动机制. 用户通过设计模块描述科学 workflows 及 QoS 约束; 提交模块负责接收执行请求; 解析模块将 workflow 划分为单个的任务、对象和数据依赖; 数据预测负责预测数据传输量, 为计算通信成本做准备; 服务发现模块根据设计模块提交的 QoS 请求为任务在 SCD 中找到可用服务集; 服务评估根据服务发现信息, 按一定原则对服务质量和价格进行评估排序; 计划模块结合数据预测对服务反复筛选, 找出满足约束的最优解; 服务预留模块估计任务开始时间, 预留服务某时间段并通过支付端口缴纳税金; 执行器和数据传输器控制任务执行和数据传输; QoS 监控器记录任务执行的序列、服务状态、参数设置、数据起源等信息, 并可消费者的评价信息反馈给市场.

3 平衡定价

根据经济学边际分析, 追加支出等于追加收入时为临界点, 即投入资金所得利益与输出损失相等时, 经济个体取得最大利润. MSWSF 框架中, SP 收入指消费者对其支付价格和, 支出指占用资源量. 为说明价格对收入支出的影响, 给出需求函数的定义. 需求函数是商品需求数量和影响该需求的各因素之间的相互关系. 对某个 SP, 当其部署服务的 QoS 确定, 价格将是影响该需求的唯一因素. 设需求函数为 $d=f(p)$, p, d 为服务价格和服务需求, 其数学模型可用常见的需求函数^[13], 也可用回归分析法求得.

若 SP 部署 n 个服务 S_1, S_2, \dots, S_n , 则 SP 收益为

$$B = \sum p_i d_i, p_i \text{ 是 } S_i \text{ 的价格, } d_i \text{ 是 } S_i \text{ 的需求. 部署服务}$$

需花费一定的资源,且资源总量有限.考虑 m 种有限资源,则收益约束于 $\sum_{j=1}^n dx_j^i = c_j, j=1, 2, \dots, m, v_j^i$ 为服务 S_i 单位时间内消耗第 j 种资源 r_j 的量, c_j 为资源 r_j 总量.本文先选 $d_i = f(p_i) = (a_i - p_i) / b_i$ ^[13] 作服务 S_i 的需求函数,其是最基本也是较常见的以价格为变量的需求函数,而且本文比较对象 CEF 也采用该函数.参数 a_i, b_i 值可根据历史需求数据用线性回归法得到.因此, B 可简化为 p_i 的函数, $B = g(p_1, p_2, \dots, p_n)$, 约束也是 p_i 的函数 $c_j = u_j(p_1, p_2, \dots, p_n), j=1, 2, \dots, m$, 求解目标为 $\max B$. 用拉格朗日乘数法求解如下

$$\max B = g(p_1, p_2, \dots, p_n) + \sum_{j=1}^m \lambda_j (c_j - u_j(p_1, p_2, \dots, p_n)) \quad (1)$$

则 B 的极值的一阶条件为

$$B_{\lambda_j} = c_j - u_j(p_1, p_2, \dots, p_n) = 0, j=1, 2, \dots, m \quad (2)$$

$$B_{p_i} = g_{p_i} - \sum_{j=1}^m \lambda_j u_{p_i} = 0, i=1, 2, \dots, n \quad (3)$$

$$g_{p_i} = \frac{\partial g(p_1, p_2, \dots, p_n)}{\partial p_i} = \frac{(a_i - 2p_i)}{b_i} \quad (4)$$

$$u_{p_i} = \frac{\partial u_j(p_1, p_2, \dots, p_n)}{\partial p_i} = -\frac{v_j^i}{b_i} \quad (5)$$

将式(4)和式(5)带入式(3)得到

$$a_i - 2p_i = \sum_{j=1}^m \lambda_j v_j^i, i=1, 2, \dots, n \quad (6)$$

化简式(2)得到

$$\sum_{i=1}^n v_i^j (a_i - p_i) / b_i = c_j \Rightarrow \sum_{i=1}^n M_i^j \cdot p_i = N_j, j=1, 2, \dots, m, \quad (7)$$

$$N_j = \sum_{i=1}^n a_i v_i^j / b_i - c_j, M_i^j = \frac{v_i^j}{b_i}$$

若 $m \geq n$, 且增广矩阵行列式不为 0, 可直接由式(7)求 p_i ; 否则, 联立式(6)(7)得到有 $m+n$ 个变量的 $m+n$ 个方程, 用线性代数理论和 Matlab 求解 p_i . g_{p_i} 是每增加单位 p_i 的收益增量, u_{p_i} 是每增加单位 p_i 的资源耗费增量, 式(6)(7)是约束最优化问题的平衡点等式, 表明各方向上每增加单位资源带来的边际效益都相等时, 资源分配的总效益 B 最优.

4 成本优化映射策略

4.1 问题描述

科学工作流与业务工作流的最大区别是: 它面向数据流, 任务间存在海量数据依赖. 科学工作流可抽象为有向无环图 $G=(T, E)$, T 是任务集, E 是数据依赖集. 数据传输成本为 $c \times \sum_{k=1}^{|E|} d_k^{rs}$, c 是单位数据传输价格 (\$ \backslash \text{Mb}), d_k^{rs} 是任务 T_r 和 T_s 的数据传输量. 科学工

作流执行在一定时限 D 内. 因此, 科学工作流调度形式化为: $\min(\sum_{j=1}^{|T|} p_i^j t_j^i + c \times \sum_{k=1}^{|E|} d_k^{rs})$

$$\text{s.t.} \quad \sum_{j=1}^n t_j^i + \sum_{k=1}^m \frac{d_k^{rs}}{B^{rs}} \leq D \quad (8)$$

$|T|$ 是任务数, $|E|$ 是数据依赖数, n 和 m 是关键任务数和关键依赖数, B^{rs} 是关键任务 T_r, T_s 执行主机间带宽, p_i^j 是任务 T_j 的可用服务 S_i 的单位时间执行价格, t_j^i 是使用时间. 算法 1 给出调度算法的通用框架.

算法 1: 调度算法通用框架

Procedure S che (G, D, set)

// G is scientific workflow, S is set of services, D is deadline

1: for $j=1$ to $|T|$

2: $Set^j \leftarrow$ get available n_j services for task T_j

3: sort $s_i^j, i=1, 2, \dots, n_j$ in Set^j in non-descending order of $p_i^j t_j^i$ value;

4: select $s_{i_j}^j$ whose $p_i^j t_j^i$ is min in Set^j for T_j and record in rb

5: end for

6: Flag = true; $ct \leftarrow$ critical tasks;

7: while (Flag) {heu S che(G, S, D, ct, rb)}

4.2 基于成本梯度的启发式调度算法

据前面分析提出三点观察: ①总数据传输量不变导致数据传输成本固定且独立于映射策略, 目标函数简化为 $\min \sum_{j=1}^{|T|} p_i^j t_j^i$; ②数据传输速度瓶颈取决于一方, 无论数据发到哪里, 发送速度均不变; ③调度器始终拥有服务的精确信息及其的完整控制.

任务 T_j 可用服务集为 $\{S_0^j, \dots, S_i^j, \dots, S_{n_j}^j\}$, 执行成本向量 $V_j = (p_0^j t_j^0, \dots, p_i^j t_j^i, \dots, p_{n_j}^j t_j^{n_j})$ 按 pt 升序排列. 关键路径执行成本向量为 $W = (V_1, \dots, V_i, \dots, V_n)$, 散列开如图 3 示. 每行是关键任务执行成本向量, 行间顺序是关键任务执行顺序. 第 j 行第 i 列的节点值是 T_j 映射到 S_i 的 $p_i^j t_j^i$, 边值是时间值 $(t_j^i + t_d^i)$, t_d^i 是运行在 S_i 的 T_j 到 T_{j+1} 的通信时间. 实线连接表示任务映射到 $\min pt$ 服务; 虚线表示存在次小 pt 层服务的调用; 点线表示树结构继续广度扩展. 为找到以最大时间降低获得最小成本增加的服务, 引入成本梯度因子. 两个服务 S_i 和 S_j 的成本梯度定义为 $\Delta pt / \Delta st = (p_i^j t_j^i - p_j^i t_j^j) / ((t_j^i + t_d^i) - (t_j^j + t_d^j))$, $p_i^j t_j^i - p_j^i t_j^j$ 是计算成本差, $(t_j^i + t_d^i) - (t_j^j + t_d^j)$ 是任务在服务 S_i 和 S_j 的运行时间差.

图3 调度算法的树形扩展分析

算法 2 给出使用成本梯度的启发式调度算法. 首先为每个任务选择 $\min pt$ 值的服务并检查是否超出时限, 即计算图 3 实线部分边值之和是否超出 D . 若超出 D 在次小 pt 组中选择. 首先根据 $(t_{q_i} + t_{d_i})_{\text{new}}$ 和 $(t_{q_j} + t_{d_j})_{\text{new}}$

$t_{c,old} < 0$ 排除部分服务, $(t_e + t_c)_{new}$ 是新替换边权值, $(t_e + t_c)_{old}$ 是原先边权值;再选择 $\min \Delta pt$ 且满足 D 的服务. 若未找到最优解, 则向深度扩展, 即替换不止一个服务. 如果所有服务均被删除, 使用成本梯度选择标准, 即每次按照成本梯度值的非降序选择前 k 个服务进行替换. 成本梯度因子是对图 3 成本矩阵的树状扩展. 显然, 当替换陷入困境时, 成本梯度因子可帮助用户快速地找到最优或次优服务.

成本优化映射的时间复杂性为 $O(|T| * l * \log l) + O(c_1 * (|T| + |E|)) + O(c_2 * n * l)$; $l = \max\{l_1, l_2, \dots, l_n\}$, l_i 是任务 T_i 的可用服务数; n 是关键任务数; 所有任务的排序时间是 $O(|T| * l * \log l)$; $O(|T| + |E|)$ 是求关键路径的时间, c_1 是执行次数; $O(c_2 * n * l)$ 是调度函数 $heuSche()$ 的时间, c_2 是使用成本梯度数量的调度次数. 另外, 实际分配操作时间是 $O(|T|)$ ($heuSche()$ 的 Step2); 为服务作比较的时间是 $O(n)$ ($heuSche()$ 的 Step4); 深度扩展时间为 $O(c * |rc|)$, 因 $|rc| \leq n$ 可化简为 $O(c * n)$, c 是常数 ($heuSche()$ 的 Step7).

算法 2: 启发式调度算法

Procedure $heuSche(G, S, D, \alpha, rb)$

```

1: if  $(s^* = \sum_{i=1}^n t_i + \sum_{k=1}^m \frac{d_{ij}}{B_{ij}} \leq D)$  // not exceeding deadline
2: then scheduling  $T_j, j=1, \dots, |T|$  on  $rb$ : flag=false; return
3: else foreach  $j$  in  $\alpha$  // exceeding deadline
4:   if  $(t_j^2 + t_e^2) - (t_j^k + (t_d)^k) \leq D$  {record service in  $rc$ :}
5:   end for each
6:   if  $rc!$  = null //finding services with less time
7:   the for  $k=1$  to  $|rc|$  {select  $k$  services with the first  $k$  smallest  $\Delta pt$  in
    $rc$  to replace old service in the same row in  $rb$ }
8:   end for
9:   else select services with  $\Delta pt / \Delta st < 0$  and record in  $rd$ :
10:   for  $k=1$  to  $|rd|$  {select first  $k$  min service with cost gradient
   in  $rd$ . use selected services to replace old service in  $rb$ }
11:   end for

```

5 实验和性能分析

平衡定价和成本优化调度的性能和效率需要验证. 实验环境为浪潮天梭 10000 IA32 机群, IBM p690 和 5 台 PC. 1PC 作 UDDI 服务器, 2PC 作模拟服务请求, 2PC 负责部署网格节点. Globus Toolkit 4.0.2 作网格中间件, juddi-0.9rc4 负责服务注册. 基于 WFEE 的 MSWSF 框架在 java 平台开发, 新增加用户操作界面、数据传输预测和服务评估模块. 实验中, MSWSF 部署 15 种服务, 每个 SP 部署 Random(10) 种服务, 不同 SP 可提供相同类型不同处理能力的服务. 不同科学 workflows 实例具有不同数

目的服务请求.

MSWSF 先由平衡定价(EquiPricing) 确定价格, 再用 CostGradient 为任务选择最佳服务. 本文选 CEF 框架和 DD 作比较. CEF 框架是典型的经济调度框架, 先用供求驱动定价 SDDP (Supply-Demand Driven Pricing) 确定价格, 再用贪婪算法 Greedy 选择最便宜服务调度任务. DD 性能比 Back-tracking 优^[2], 且考虑通信成本, 本文选 DD 作比较. 选择资源节点平均收益、资源平均使用率、优化能力和调度计划时间作评价标准, 并在合成 workflows (Synthetic Workflow) 和实际应用 (Real Application) 上验证其性能, 合成 workflows 又分为合成并行和合成随机 workflows.

5.1 合成并行 workflows

提交不同节点数的并行 workflows 验证平衡定价性能. SDDP 受市场供求影响, 此环境供求指节点单位时间处理能力与应用服务请求数目间的关系. MSWSF 规定资源节点单位平均处理能力为 500. 图 4 给出 EquiPricing 与 SDDP 的资源节点收益均值比较. 供大于求时 EquiPricing 劣于 SDDP, 因 EquiPricing 针对资源有限情况设计, 符合实际情况. 供小于求时 EquiPricing 明显优于 SDDP, 因 EquiPricing 以最大化资源节点收益为目标给出服务合理定价, SDDP 只是在供小于求时简单提高价格, 造成交易成功率降低. 图 5 统计 EquiPricing 和 SDDP 不同实例的资源占用率. 供大于求时两者的资源占用随着服务请求量增加稳步上升. SDDP 在供求平衡时占用率达到峰值, 但在供小于求时其呈递减趋势, 因 SDDP 有调节滞后性; 而 EquiPricing 能找到追加收入和支出平衡点, 可在增加收益的同时稳定提高资源占用率.

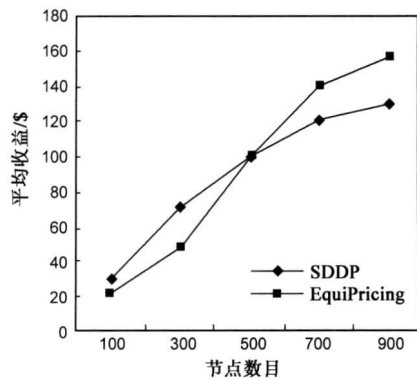


图 4 不同请求数目时的平均资源节点收益

为验证 MSWSF 成本优化能力, 提交 500 个节点和 150 条边的并行 workflows. 图 6 给出不同时限的执行成本, S 是关键路径长度. 时限较大时 CostGradient 可找到最优解或次优解. DD 成本也随时限增加而下降, 但因使用划分策略成本下降幅度比 CostGradient 小. Greedy 执行成本最高, 因其优先选择便宜资源导致时限在调度早期过度消耗, 后期必须使用昂贵服务加快速度, 图

7 给出 16s 时限和不同节点数的调度响应时间占总体执行时间的比例。CEF 框架采用实时策略,计划时间忽略不计。DD 和 MSWSF 使用静态调度策略,故其需计划调度时间。DD 的计划时间据 workflow 实例结构的变化而变化,其百分比不稳定。而 CostGradient 的计划时间一般与节点数目成正比。

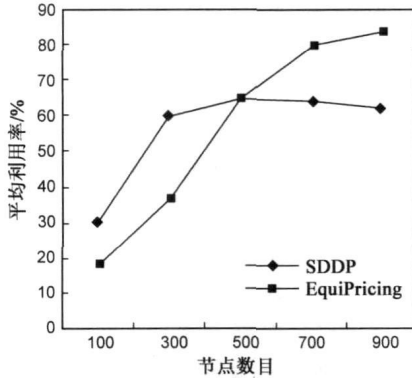


图5 不同请求数目时的资源平均利用率
合成并行工作流

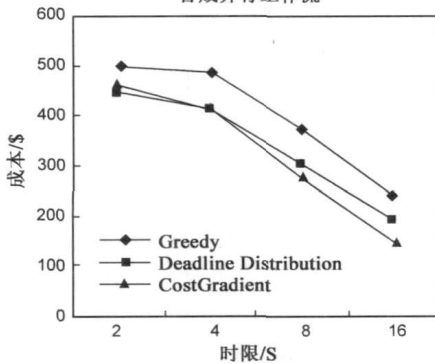


图6 不同deadline约束下的执行成本

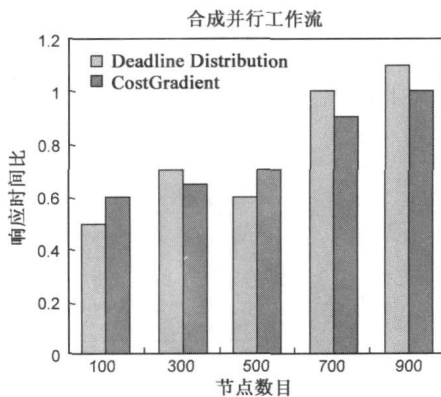


图7 不同节点数目时的计划调度时间百分比

5.2 合成随机工作流

平衡定价性能优劣与工作流结构无直接关系,不再赘述合成随机工作流和实际应用的资源节点收益及资源利用率评估。产生 5 个随机工作流,节点数为 200, 400, 600, 800 和 1000。图 8 给出 16s 时限 5 个工作流的执行成本,时限较大时 CostGradient 可较快且准确找到

最优解。特别是节点数较多时, CostGradient 明显优于 DD, 主要因为节点数越多, DD 产生子工作流越多, 子工作流优化值之和与整体最优解的误差就越大。图 9 给出 DD 和 CostGradient 的调度计划时间。该实验使用随机工作流进一步验证 5.1 节结论。对 DD, 某些随机工作流计划时间比并行工作流短, 这主要由其特殊划分方式^[2]决定。

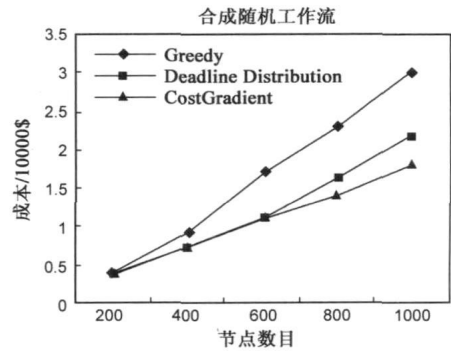


图8 不同节点数目时的执行成本

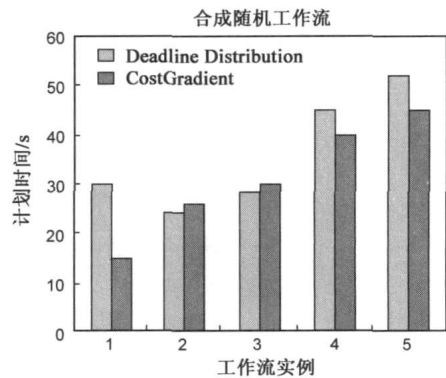


图9 不同工作流实例时的计划调度时间

5.3 实际应用工作流

最后,使用实际应用 Montage 评估 MSWSF 框架和算法。Montage 实际属于并行工作流。图 10 表明不同时限各种算法的成本优化能力。CostGradient 对实际应用也有较好性能。由于任务并行度不同, DD 在特定情况下会表现出比 CostGradient 优的性能。

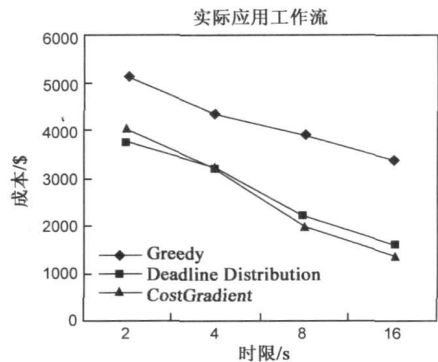


图10 不同deadline约束下的执行成本比较

6 结论

本文提出效用网格环境下市场驱动的科学 workflow 调度框架:平衡定价利用边际原理实现服务提供者利益优化和资源最优分配,不仅考虑了服务消耗多种资源的情况,还兼顾资源的有限使用性;成本优化映射策略优先排除非优解,较快实现了执行成本最小化,当查找陷入困境时成本梯度因子又能快速有效地为当前任务找到新的最佳服务.通过与 DD 和 CEF 比较,平衡定价可实现较高的资源利用率和资源节点收益,尤其适合于供小于求的市场;基于成本梯度的调度能在较短计划时间内找到成本最优或次优的调度方案,尤其是在 loose deadline 的情况下.

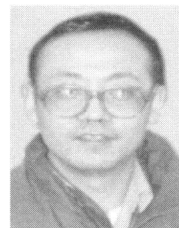
参考文献:

- [1] J Yu, R Buyya. A Taxonomy of workflow management systems for grid computing[J]. *Journal of Grid Computing*, 2005, 3(3—4): 171—200.
- [2] J Yu, R Buyya, C K Tham. Cost-based scheduling of scientific workflow applications on utility grids[A]. *Proceedings of the 1st IEEE International Conference on e-Science and Grid Computing*[C]. Melbourne Australia: IEEE, 2005. 140—147.
- [3] O O Sonmez, A Gursay. Comparison of pricing policies for a computational grid market[A]. *Parallel Processing and Applied Mathematics* [M]. Berlin/Heidelberg: Springer, 2006. 766—773.
- [4] C L Li, L Y Li. An optimization approach for decentralized QoS-based scheduling based on utility and pricing in grid computing [J]. *Concurrency and Computation: Practice & Experience*, 2007, 19(1): 107—128.
- [5] 吴健, 邓磊, 胡正国. 网格计算经济模型中的价格协商和管制[J]. *西北工业大学学报*, 2008, 26(4): 497—502.
Wu Jian, Hu Zhengguo, Deng Lei. Pricing strategies for grid computational economy [J]. *Journal of Northwestern Polytechnical University*, 2008, 26(4): 497—502. (in Chinese)
- [6] 李立, 刘元安, 马晓雷. 基于组合双向拍卖的网格资源分配[J]. *电子学报*, 2009, 37(1): 165—169.
Li Li, Liu Yuanan, Ma Xiaolei. Grid resource allocation based on the combinatorial double auction[J]. *Acta Electronica Sinica*, 2009, 37(1): 165—169. (in Chinese)
- [7] H Li, Y Zhong, Joan Lu. A banking based grid recourse allocation scheduling[A]. *Proceedings of the 3rd International Conference on Grid and Pervasive Computing* [C]. Kunming: IEEE, 2008. 239—244.
- [8] R Buyya, D Abramson, J Giddy, et al. Economic models for resource management and scheduling in grid computing[J]. *Journal of Concurrency: Practice and Experience. Grid computing special issue*, 2002, 14(13—15): 1507—1542.
- [9] M Wiecek, R Prodan, T Fahringer. Comparison of workflow scheduling strategies on the grid[A]. *Parallel Processing and Applied Mathematics* [M]. Berlin/Heidelberg: Springer, 2006. 792—800.
- [10] 陈宏伟, 王汝传. 费用-时间优化的网格有向无环图调度算法[J]. *电子学报*, 2005, 33(8): 1375—1380.
Chen Hongwei, Wang Ruchuan. A grid DAG scheduling algorithm for Cost-Time optimization[J]. *Acta Electronica Sinica*, 2005, 33(8): 1375—1380. (in Chinese)
- [11] D A Menascio, E Casalicchio. A framework for resource allocation in grid computing[A]. *Proceedings of 12th Annual International Symposium on Modeling, Analysis, and Simulation of Computer and Telecommunications Systems* [C]. Volendam: IEEE, 2004. 259—267.
- [12] J Yu, R Buyya. A novel architecture for realizing grid workflow using tuple spaces[A]. *Proceedings of 5th IEEE/ACM International Workshop on Grid Computing* [C]. Pittsburgh: IEEE, 2004. 119—128.
- [13] Luo W Y, Liu J H. *Microeconomics: Implications And Applications of Principles* [M]. Beijing: Economic Publishing House, 2006. 70—78.

作者简介:



马 艳 女, 1982 年 10 月生于山东泰安. 博士研究生, 就读于山东大学计算机科学与技术学院, 从事网格与高性能计算、功耗感知计算方面的研究.



龚 斌 男, 1964 年 10 月生于山东济南. 教授、博士生导师. 现为山东大学计算中心主任、山东省高性能计算中心副主任, 主要从事网格与高性能计算、机群计算方面的研究工作.
E-mail: gb@sdu.edu.cn