

# 截止时间约束下独立网格任务的 协同调度模型

肖 鹏<sup>1,2</sup>, 胡志刚<sup>2</sup>

(1. 湖南工程学院计算机与通信系, 湖南湘潭 411104; 2. 中南大学信息科学与工程学院, 湖南长沙 410083)

**摘 要:** 针对网格环境中任务实时性难于保证的问题, 提出一种截止时间保证度的计算方法, 用于量化分析资源协同调度对任务实时约束的影响, 并设计和实现了一个可扩展的协同调度模型, 允许系统动态选择截止时间保证度最优的协同调度方案. 实验结果显示, 截止时间保证度指标可以较为准确地量化调度决策对任务实时约束的满足程度, 从而为实时网格任务调度决策的最优化提供可靠依据; 基于混合策略的调度模型则能综合已有协同调度策略的优点, 为实时网格任务提供可靠的服务质量保证.

**关键词:** 网格计算; 截止时间约束; 服务质量; 随机服务模型

**中图分类号:** TP393      **文献标识码:** A      **文章编号:** 0372-2112 (2011) 08-1852-06

## Co-Scheduling Model for Independent Tasks with Deadline Constraint in Computational Grid

XIAO Peng<sup>1,2</sup>, HU Zhi-gang<sup>2</sup>

(1. School of Computer and Communication, Hunan Institute of Engineering, Xiangtan, Hunan 411104, China;

2. School of Software, Central South University, Changsha, Hunan 410083, China)

**Abstract:** To address the issue of task co-schedule with deadline constraint in grids, a metric called probability of deadline guarantee is proposed to quantitatively evaluate the deadline-guarantee of a co-scheduling scheme. Based on this metric, an extensive co-scheduling framework is implemented, which combines multiple policies to obtain the optimal deadline guarantee. Extensive experiments are conducted to examine the performance of the proposed model. The results show that the probability of deadline guarantee metric can accurately quantify the degree of deadline guarantee provided by a specific co-scheduling scheme, which in turn provides an enhanced QoS for those applications with deadline constraint.

**Key words:** grid computing; deadline constraint; quality of service; stochastic service theory

## 1 引言

随着网格技术在新兴领域的应用, 大量应用对网格系统的实时响应能力提出了更高的要求, 而在动态性极强的网格环境中, 为实时任务提供截止时间保证始终是一个具有挑战性的课题<sup>[1~3]</sup>, 其难点主要集中在三个方面: (1) 资源可用性极不稳定, 系统无法向用户提供稳定的实时保证<sup>[4]</sup>; (2) 负载均衡策略容易导致低优先级任务难以获得明确的截止时间保证<sup>[5,6]</sup>; (3) 缺乏统一尺度来评估各类异构资源的动态服务能力<sup>[2,3,7]</sup>.

早期关于协同调度的研究大多回避实时性保证问题, 只针对任务的执行效率、响应时间、数据传输等指标进行优化, 典型的协同调度策略有随机选择策略<sup>[7]</sup>、数

据临近访问策略<sup>[8]</sup>、数据存储策略<sup>[9]</sup>、通信效率最优策略<sup>[10]</sup>等. 为了比较各种策略的性能表现, Bucur 等人<sup>[11]</sup>在网格测试床 DAS-2 下的性能测评分析结果显示, 具有负载感知能力的协同调度策略能获得较好平均响应时间, 也有利于系统实现负载均衡. 近期研究者已经开始关注实时任务的协同调度策略. 文献<sup>[12]</sup>提出了两种协同调度策略为实时网格任务提供最优截止时间违约率, 两种策略均采用数值逼近算法来求解最优协同调度方案. 针对基于计算经济的网格系统, Vanmechelen 等人<sup>[13]</sup>提出利用期货市场在一定时间段内锁定资源可获得性, 从而保证实时任务的截止时间. 文献<sup>[14]</sup>提出利用反馈理论将负载控制问题转换为一个约束优化问题, 并通过迭代逼近的方式来求解资源的最佳负载, 从而使资源节

点能提供最优的实时性保证;针对多性能尺度约束下的实时调度问题,Zhu 等人<sup>[15]</sup>提出了一个三阶段调度模型,将不同性能指标的优化策略以串联方式组合,从而叠加各种策略对实时保证度的正面效果.与以上研究不同,本文从截止时间保证度的量化计算方法入手,分析特定协同调度方案对任务截止时间约束的满足程度,然后综合已有各种协同调度策略的优点,为实时任务调度提供一个可扩展的调度模型.

## 2 问题描述与定义

**定义 1** 设网络系统由计算站点  $(CE_1, \dots, CE_N)$  构成,各个  $CE_i$  的服务模型表示为三元组  $\langle \lambda_i, \mu_i, c_i \rangle$ ,其中  $\lambda_i$  为任务到达时间间隔,  $\mu_i$  为  $CE_i$  的平均服务时间,  $c_i$  为  $CE_i$  的并行服务能力.

**定义 2** 实时网络任务表示为二元组  $\langle R, d \rangle$ ,其中  $R = \langle c_1, \dots, c_m \rangle$  为子任务的资源请求,  $d$  为任务的截止时间约束.

**定义 3** 协同调度方案是任务资源请求到计算站点集合的映射,表示为  $\bar{M}: R \times \{1, \dots, N\} \rightarrow \{0, 1\}$ .

**定义 4** 设随机变量  $E$  表示实时任务  $\langle R, d \rangle$  采用协同调度方案  $\bar{M}$  后的实际完成时间.

**定义 5** 给定任务  $\langle R, d \rangle$  和协同调度方案  $\bar{M}$ ,截止时间保证度表示为  $\Pr\{E \leq d | \bar{M}\}$ ,即采用该分配方案时,任务实际完成时间早于截止时间  $d$  的概率.

综合以上定义,截止时间约束下独立任务的协同调度问题可以一般化地描述为式(1)所示的约束优化问题,其中  $2^{R \times \{1 \dots N\}}$  表示所有协同调度方案集合.由于约束条件中的问题规模具有指数复杂度,因此该问题属于 NP 完全难题.

$$\begin{aligned} \max \quad & \Pr\{E \leq d | \bar{M}\} \\ \text{s.t.} \quad & \bar{M} \in 2^{R \times \{1 \dots N\}} \end{aligned} \quad (1)$$

本文采用有限枚举问题空间的策略,即用集合  $(\bar{M}_1, \bar{M}_2, \dots, \bar{M}_k)$  表示现有各种协同调度策略针对特定任务所生成的调度方案集合,用于替换原问题中的  $2^{R \times \{1 \dots N\}}$  集合,将原约束优化问题转化为式(2)所示的优化问题.

$$\begin{aligned} \max \quad & \Pr\{E \leq d | \bar{M}\} \\ \text{s.t.} \quad & \bar{M} \in \{\bar{M}_1, \bar{M}_2, \dots, \bar{M}_k\} \end{aligned} \quad (2)$$

显然,式(2)所示问题的时间复杂度可以表示为

$$O\left[\sum_{i=1}^k [O(\Pr\{E \leq d | \bar{M}_i\}) + O(\bar{M}_i)]\right]$$

$O(\Pr\{E \leq d | \bar{M}_i\})$  表示计算  $\Pr\{E \leq d | \bar{M}\}$  的时间复杂度,  $O(\bar{M}_i)$  表示生成  $\bar{M}_i$  方案的协同调度策略的固有时间复杂度.

## 3 截止时间保证度

**定理 1** 若计算站点  $CE_i$  采用随机服务模型  $\langle \lambda_i,$

$\mu_i, c_i \rangle$  描述,实时任务的相对截止时间需求为  $d$ ,则任务能在  $d$  时间段内完成的概率为

$$\Pr\{E \leq d\} = \sum_{k=0}^{c_i \mu_i d - 1} \Pr\{W = k\} \quad (3)$$

其中,  $E$  和  $W$  均为随机变量,分别表示任务的实际完成时间和  $CE_i$  中的等待任务数目.

**证明** 由随机服务理论<sup>[16]</sup>可知,计算站点  $CE_i$  在单位时间内能服务的任务数目为  $c_i \mu_i$ ,即在  $d$  时间段内  $CE_i$  能完成的任务数目为  $c_i \mu_i d$ .因此,当实时任务的相对截止时间为  $d$  时,为保证该任务能在  $d$  之内完成,  $CE_i$  中的等待队列长度必须小于等于  $c_i \mu_i d - 1$ ,即

$$\Pr\{E \leq d\} = \Pr\{W \leq c_i \mu_i d - 1\} \quad (4)$$

显然,概率  $\Pr\{W \leq c_i \mu_i d - 1\}$  的密度函数可以表示为  $\Pr\{W = k\}$ ,代表  $CE_i$  中等待队列长度为  $k$  的概率.由概率函数与其密度函数的关系可得

$$\Pr\{W \leq c_i \mu_i d - 1\} = \sum_{k=0}^{c_i \mu_i d - 1} \Pr\{W = k\} \quad (5)$$

综合式(4)和式(5),得证.

定理 1 给出了随机服务模型对实时任务截止时间保证度的通用计算方法.由于模型参数与限制条件不同,各种随机服务模型的  $\Pr\{W = k\}$  计算公式差别很大,下文的推论 1 和推论 2 将给出  $M/M/1$  和  $M/M/c_i$  两种服务模型的截止时间保证度计算公式,其余服务模型可以类似推导获得,本文不作过多赘述.

**推论 1** 若计算站点  $CE_i$  以串行方式提供服务并用  $M/M/1$  型队列系统描述,则  $CE_i$  能满足任务截止时间要求的概率如式(6)所示,其中  $\rho_i = \lambda_i / \mu_i$ .

$$\Pr\{E \leq d\} = \sum_{n=0}^{c_i \mu_i d - 1} (1 - \rho_i) \rho_i^n \quad (6)$$

**证明** 由随机服务理论可知,  $M/M/1$  模型的等待队列长度的概率密度函数为

$$\Pr\{W = k\} = (1 - \rho_i) \rho_i^k \quad (7)$$

由定理 1 直接可得证.

**推论 2** 若计算站点  $CE_i$  的负载和服务模型用  $M/M/c_i$  型队列系统描述,则  $CE_i$  能满足任务截止时间要求的概率为

$$\Pr\{E \leq d\} = \sum_{n=0}^{c_i} \delta \frac{(\rho_i c_i)^n}{n!} + \sum_{k=1}^{c_i \mu_i d - 1} \delta \frac{\rho_i^{k+c_i} c_i!}{c_i!} \quad (8)$$

其中  $\delta = \left[ \sum_{n=1}^{c_i} \frac{(\rho_i c_i)^n}{n!} + \frac{(\rho_i c_i)^{c_i}}{c_i!} \frac{1}{1 - \rho_i} \right]^{-1}$ ,  $\rho_i = \frac{\lambda_i}{(c_i \mu_i)}$ .

**证明**  $M/M/c_i$  模型的等待队列长度的概率密度函数为

$$\Pr\{W = k\} = \begin{cases} \delta \frac{\rho_i^{k+c_i} c_i!}{c_i!}, & k > 0 \\ \sum_{n=0}^{c_i} \delta \frac{(\rho_i c_i)^n}{n!}, & k = 0 \end{cases} \quad (9)$$

由定理 1 的推导和结论直接可得证。

以上定理和推论给出了计算站点对单一子任务的截止时间保证度  $\Pr\{E \leq d\}$ , 为求解式(2)所描述的约束问题, 需要进一步计算给定协同调度方案下的截止时间保证度, 即  $\Pr\{E \leq d | \bar{M}\}$ . 如前文所述, 网格系统是否采用冗余策略将影响截止时间保证度  $\Pr\{E \leq d | \bar{M}\}$  的计算结果, 算法 1 和算法 2 分别针对两类系统给出了  $\Pr\{E \leq d | \bar{M}\}$  的计算方法。

**算法 1** Calculating  $\Pr\{E \leq d | \bar{M}\}$  without Redundant Policy

```

Begin
1.  $\Pr\{E \leq d | \bar{M}\} := 1$ 
2. For  $j = 1$  to  $N$ 
3.    $WaitLength[j] := c_{\mu_j} d - 1$ 
4. End For
5. For  $i = 1$  to  $m$ 
6.   For  $j = 1$  to  $N$ 
7.     If  $S_{i,j} = 1$  Then
8.        $P_i^j := \sum_{k=0}^{WaitLength[j]} \Pr\{W = k\}$ 
9.        $\Pr\{E \leq d | \bar{M}\} := \Pr\{E \leq d | \bar{M}\} P_i^j$ 
10.       $WaitLength[j] := WaitLength[j] - 1$ 
11.      Continue
12.    End For
13.  End For
End

```

**算法 2** Calculate  $\Pr\{E \leq d | \bar{M}\}$  with Redundant Policy

```

1. For  $i = 1$  to  $m$ 
2.    $Max\_P_i^j = 0$ 
3.   For  $j = 1$  to  $N$ 
4.     If  $S_{i,j} = 1$  Then
5.        $P_i^j := \sum_{k=0}^{WaitLength[j]} \Pr\{W = k\}$ 
6.       If  $Max\_P_i^j < P_i^j$  Then  $Max\_P_i^j := P_i^j$ 
7.        $WaitLength[j] := WaitLength[j] - 1$ 
8.     End For
9.    $\Pr\{E \leq d | \bar{M}\} := \Pr\{E \leq d | \bar{M}\} \cdot Max\_P_i^j$ 
10. End For

```

在算法 1 和算法 2 中, 由于公式(6)和(8)没有考虑多个子任务同时被调度到同一个计算站点的情况, 算法利用  $WaitLength$  数组解决这个问题, 即当一个子任务被调度到一个计算站点, 其后被调度到该站点的子任务的截止时间保证将相应降低, 修正后的公式如算法 1 中 Step8 以及算法 2 中 Step5 所示. 算法 2 的描述部分只给出与算法 1 不同的部分. 显然, 算法 1 和算法 2 的时间复杂度均为  $O(m \times N)$ ,  $m$  为任务所包含的子任务数目,  $N$  为系统的资源站点数目, 约束问题(2)求解的总体复杂度为  $O[\sum_{i=1}^k [O(m \times N) + O(\bar{M}_i)]]$ .

## 4 实验分析与比较

本文设计的基于混合策略的协同调度体系 HPCF (Hybrid Policy Co-scheduling Framework) 工作方式如下: 系统分别采用不同的协同调度策略来生成多个分配方案  $(M_1, M_2, \dots, M_k)$ , 决策器则利用算法 1 或算法 2 来评估不同分配方案对任务截止时间的保证程度, 调度器最终选择保证度最高的分配方案为任务提供服务. HPCF 原型中实现了三种比较典型的协同调度策略: Round Robin<sup>[17]</sup> (RR), Capability-based Random<sup>[7]</sup> (CR), Cluster Minimized<sup>[8]</sup> (CM). 实验所用的多集群计算网格系统包括 12 个高性能计算站点, 集群节点之间通过 1G 的网络连接, 集群内部采用 100M 的局域网连接. 实验采用 Lublin-Feitelson 模型<sup>[18]</sup>生成任务负载, 每个任务负载由到达时间、资源请求量、截止时间三个参数表示。

### 4.1 HPCF 执行性能分析

本次实验主要分析和比较 HPCF 模型在截止时间违约率 (Deadline Miss Rate) 方面的性能表现. Casanova<sup>[19]</sup> 的研究显示, 对于 K-request 的冗余策略, K 取 2 或 3 时, 系统能在不显著增加额外开销的前提下获得较好的可靠性保证, 因此本实验中只关注  $K = 1, 2, 3$  的情况. 表 1 是各种策略在截止时间违约率方面的统计结果. 在前三种协同调度策略中, CR 对任务截止时间的保证最优, CM 的表现最差. 策略的实际执行情况显示, CM 将绝大部分任务调度到少数几个处理器数量较大的计算站点上. 对于单一任务而言, CM 倾向于将其多个子任务调度到同一个计算站点. 由于实验中的任务为独立网格任务, 这使得 CM 无法充分发挥其降低通讯开销的特长. RR 与 CM 刚好相反, 任务或子任务被均匀地调度到各个计算站点, 这使得任务获得了最大的并行性。

表 1 截止时间违约率对比

	Deadline Miss Rate		
	$K = 1$	$K = 2$	$K = 3$
RR_P	13.93%	11.25%	10.74%
CR_P	11.47%	9.28%	8.22%
CM_P	18.33%	12.64%	10.81%
HPCF	8.74%	6.68%	5.32%

详细实验显示, RR 策略产生的截止时间违约大多发生在少数几个低性能计算站点上, 这些站点的性能瓶颈制约了 RR 策略在协同调度过程中任务整体截止时间保证. CR 正好弥补了 RR 和 CM 的以上缺陷, 从而使得其性能明显优于前两者. 表 1 数据显示, 当采用冗余请求时, 三种策略的性能都得到了不同程度的提高. 其中 CM 的提高幅度最高 ( $K = 2$  时提高 5.96%,  $K = 3$  时提高 7.52%), 原因是冗余请求策略有效地克服了 CM 的负载过分集中的缺点. 当 HPCF 被开启后, 决策器

对三种策略的分配方案进行评估,选择对截止时间保证最好的策略作为最终分配方案.从表 1 数据显示, HPCF 的方法能有效提高协同调度过程中对任务截止时间的保证.为分析 HPCF 的决策情况,实验统计了 HPCF 的决策分布,如表 2 所示.

表 2 HPCF 决策分布情况对比

	Decision Counts (Deadline Miss Counts)		
	RR	CR	CM
$K = 1$	2253 (214)	4839 (357)	908 (128)
$K = 2$	1855 (139)	4774 (332)	1371 (73)
$K = 3$	2075 (152)	4116 (204)	1809 (64)

HPCF 决策分布情况显示,对于  $K = 1$  和  $K = 2$ , HPCF 对 8000 个任务的协同调度方案进行决策时,超过 50% 的任务都选用了 CR,这反映了 CR 对独立任务确实能提供较好的截止时间保证.另一方面,当采用冗余请求时,CM 被选中的次数有很大幅度提高.当  $K = 3$  时,CM 被选中的次数是不采用冗余请求策略时的 2 倍,而且其相应的截止时间违约率仅为 3.54%.结合表 1 和表 2 的实验结果与分析可以看出,HPCF 提供了一个可扩展的协同调度框架,不具备供截止时间保证能力的各类协同调度策略可以互为补充,为实时网络任务提供更好的截止时间保证.

#### 4.2 对比实验分析

本次实验主要以截止时间违约率为指标,横向比较混合策略 HPC (Hibrid Policy Co-allocation) 与已有的面向实时任务的协同调度策略的性能表现.实验选择了四种面向实时性保证的策略:ORT<sup>[12]</sup> (Optimal mean Response Time)、OMR<sup>[12]</sup> (Optimal mean Miss Rate)、UFC<sup>[14]</sup> (Utilization Feedback Control)、TSO<sup>[15]</sup> (Three-Step Optimization).实验将分四组进行, $\lambda$  分别设置为 0.05, 0.1, 1.0, 10, 每组实验针对不同的任务冗余度  $K$  进行 3 次,实验结果如图 1 所示.参数  $\lambda$  表示单位时间内到达系统的平均任务数目,即  $\lambda$  值越大则表示任务到达系统的时间越集中,越小则表示任务到达时间越分散.实验数据显示  $\lambda$  参数本身对系统的截止时间违约率有较为重要的影响,具体表现为:(1)当任务到达过程趋于分散时,不论采用何种策略,总体的截止时间违约率都低于 18%;(2)当  $\lambda$  的取值超过 1.0 时,截止时间违约率大多超过 35%,远远超过实际网络系统的容忍程度.

在与 HPC 策略对比的四种策略中,OMR 的性能表现最优,当不采用冗余机制时,OMR 与 HPC 的性能差异大约在 3% ~ 5%,尤其是当  $\lambda = 10$  时,OMR 的违约率比 HPC 策略还低 4.7%.当  $K = 2$  或 3 时,OMR 策略的性能改进不是很明显.从实验数据来看,HPC 策略的优势主要体现在系统的冗余度  $K = 2$  和 3 时,原因在于 HPC 策略的实现同时容纳了三种基本的调度策略,这三种策

略各自具有较强的互补性.因此,HPC 在  $K = 2$  和 3 时,相对其它四种策略的截止时间违约率能降低 29% ~ 47%.

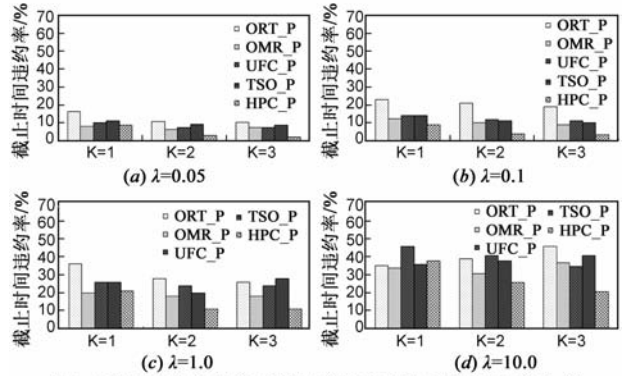


图 1 不同负载集中度下的网络任务截止时间违约率比较

#### 4.3 截止时间保证度分析

本次实验主要考察理论计算所得的  $\Pr\{E \leq d | \bar{M}\}$  与实际截止时间保证率的关系.实验分为三组,记录了每个调度方案的  $\Pr\{E \leq d | \bar{M}\}$  值,并将其取值范围分为 5 个区间:  $(0, 0.2]$ ,  $(0.2, 0.4]$ ,  $(0.4, 0.6]$ ,  $(0.6, 0.8]$ ,  $(0.8, 1.0]$ .对每一个区间,实验统计了其没有发生截止时间违约的次数和总执行次数,两者的比值为实际截止时间保证率 PDGR (Practical Deadline Guarantee Rate),图 2 为实验统计结果,折线图为  $\Pr\{E \leq d | \bar{M}\}$  区间对应的实际截止时间保证率,柱状图为各个  $\Pr\{E \leq d | \bar{M}\}$  区间对应的总执行次数的分布情况.

实验数据显示 PDGR 与理论计算得到的  $\Pr\{E \leq d | \bar{M}\}$  值具有相同的变化趋势,但并非成线性关系.例如,当  $\Pr\{E \leq d | \bar{M}\} \geq 0.6$  时,PDGR 非常接近  $\Pr\{E \leq d | \bar{M}\}$ .

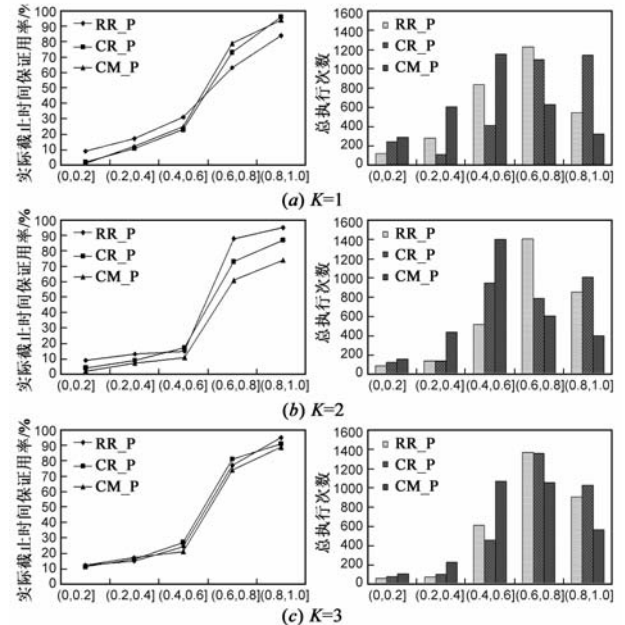


图 2 不同冗余度下  $\Pr\{E \leq d | \bar{M}\}$  值和实际截止时间保证率的关系及其决策分布

$\bar{M}$ ; 当  $\Pr\{E \leq d|\bar{M}\} \leq 0.4$  时,  $\Pr\{E \leq d|\bar{M}\}$  值相对 PDGR 值偏高. 从图 2 中执行次数分布图(柱状图)可以看出, 对  $K=1$  而言, 约有 50% 的任务能获得  $\Pr\{E \leq d|\bar{M}\}$  大于 0.6 的分配方案. 冗余度  $K$  对 PDGR 与  $\Pr\{E \leq d|\bar{M}\}$  的之间差异的影响主要体现在以下: (1) 在不采用冗余请求时且选择 RR 策略时, PDGR 值与  $\Pr\{E \leq d|\bar{M}\}$  最为接近; (2) 在使用 CR 和 CM 策略且 PDGR 值与  $\Pr\{E \leq d|\bar{M}\}$  的差异度值较小时, PDGR 值趋近  $\Pr\{E \leq d|\bar{M}\}$  的下限. 产生这一现象的原因是: 当使用 CR 和 CM 策略时,  $\Pr\{E \leq d|\bar{M}\}$  值较小的协同调度方案大多将多个子任务集中在一个计算节点上, 而这些协同调度方案往往无法在截止时间内完成任务. 实验数据同时显示, 在采用冗余请求时, 当  $\Pr\{E \leq d|\bar{M}\}$  值较小时, 三种调度策略的 PDGR 值都趋近其对应的  $\Pr\{E \leq d|\bar{M}\}$  下限, 其原因是: 由于采用了冗余请求策略, 系统的负载相对增加了,  $\Pr\{E \leq d|\bar{M}\}$  值较低的分配方案大多是在系统负载处相对较高时产生的, 加之任务对截止时间要求较为紧迫, 使得系统实际截止时间保证率普遍降低.

本次实验结论是: (1) 选择  $\Pr\{E \leq d|\bar{M}\}$  值较大的协同调度方案能够有效地保证任务的截止时间需求; (2) 当  $\Pr\{E \leq d|\bar{M}\}$  值较小时, 其准确率与实测的截止时间保证率有大约 7% ~ 12% 的偏差.

## 5 总结

本文研究了实时网络任务的截止时间保证度问题, 针对独立网络任务提出了一个截止时间保证度的计算方法, 并综合了冗余请求策略对计算结果的影响. 基于该方法, 本文设计和实现了一个基于混合策略的协同调度模型 HPCF. 实验比较了 HPCF 在截止时间保证率方面的表现以及其评估截止时间保证度的准确率. 结果显示, HPCF 是一个可扩展的协同调度框架, 使得不提供截止时间保证的各类协同调度策略可以互为补充, 为具有明确截止时间需求的网络应用提供更可靠的实时保证. 在今后的工作中, 我们将把资源失效的情况结合到实时保证度的计算方法中.

## 参考文献

- [1] Kacsuk P, Kovacs J, Farkas Z, et al. SZTAKI desktop grid (SZDG): A flexible and scalable desktop grid system[J]. Journal of Grid Computing, 2009, 7(4): 439 – 461.
- [2] Hussin M, Lee Y C, Zomaya A Y. Dynamic job-clustering with different computing priorities for computational resource allocation[A]. Proceedings of International Conference on Cluster Cloud and Grid Computing[C]. Washington USA: IEEE Computer Society Press, 2010. 589 – 590.

- [3] Qin X, Jiang H, Manzanares A, et al. Communication-aware load balancing for parallel applications on clusters[J]. IEEE Transaction on Computers, 2010, 59(1): 42 – 52.
- [4] Kondo D, Kindarji B, Fedak G, et al. Towards soft real-time applications on enterprise desktop grids[A]. Proceedings of International Symposium on Cluster Computing and the Grid[C]. Washington, USA: IEEE Computer Society Press, 2006. 65 – 72.
- [5] Czajkowski K, Foster I, Kesselman C. Resource co-allocation in computational grids[A]. Proceedings of International Symposium on Cluster Computing and the Grid[C]. Washington, USA: IEEE Computer Society, 1999. 219 – 228.
- [6] Sonmez O O, Mohamed H, Epema D H J. On the benefit of processor coallocation in multicluster grid systems[J]. IEEE Transactions on Parallel and Distributed Systems, 2010, 21(6): 778 – 789.
- [7] Berten V, Goossens J, Jeannot E. On the distribution of sequential jobs in random brokering for heterogeneous computational grids[J]. IEEE Transactions on Parallel and Distributed Systems, 2006, 17(2): 113 – 124.
- [8] Mohamed H H, Epema D H J. Experiences with the KOALA co-allocating scheduler in multiclusters[A]. Proceedings of International Symposium on Cluster Computing and the Grid[C]. Washington, USA: IEEE Computer Society Press, 2005. 784 – 791.
- [9] 艾丽华, 罗四维. 数据网格虚拟机动态存储层次的研究[J]. 电子学报, 2010, 38(11): 2680 – 2685.
- Ai Lihua, Luo Siwei. Research on dynamic storage hierarchy of data grid virtual machine[J]. Acta Electronica Sinica, 2010, 38(11): 2680 – 2685. (in Chinese)
- [10] 张伟哲, 田志宏, 张宏莉, 等. 虚拟计算环境中的多机群协同调度算法[J]. 软件学报, 2007, 18(8): 2027 – 2037.
- Zhang Weizhe, Tian Zhihong, Zhang Hongli, et al. Multi-cluster co-allocation scheduling algorithms in virtual computing environment[J]. Journal of Software, 2007, 18(8): 2027 – 2037. (in Chinese)
- [11] Bucur A I D, Epema D H J. Scheduling policies for processor coallocation in multicluster system[J]. IEEE Transactions on Parallel and Distributed Systems, 2007, 18(7): 958 – 962.
- [12] He L G, Jarvis S A, Spooner D P, et al. Allocating non-real-time and soft real-time jobs in multiclusters[J]. IEEE Transactions on Parallel and Distributed Systems, 2006, 17(2): 99 – 112.
- [13] Vanmechelen K, Depoorter W, Broeckhove J. Economic grid resource management for cpu bound applications with hard deadlines[A]. Proceedings of International Symposium on Cluster Computing and the Grid[C]. Washington, USA: IEEE Computer Society, 2008. 258 – 266.
- [14] 王洁, 王宏安, 傅勇, 等. 一种面向异构实时集群系统的

使用率反馈控制方法[J]. 计算机研究与发展, 2009, 46 (10):1626 – 1633.

Wang Jie, Wang Hongan, Fu Yong, et al. Feedback utilization control for heterogeneous real-time clusters [J]. Journal of Computer Research and Development, 2009, 46(10): 1626 – 1633. (in Chinese)

[15] Zhu X M, Lu P Z. Multi-dimensional scheduling for real-time Tasks on heterogeneous clusters[J]. Journal of Computer Science and Technology, 2009, 24(3): 434 – 446.

[16] Gross D, Harris C M. Fundamentals of Queuing Theory [M]. USA: John Wiley and Sons, 1998. 43 – 49.

[17] Dumitrescu C L, Raicu I, Foster I. The design, usage, and performance of GRUBER: a grid usage service level agreement based brokering infrastructure[J]. Journal of Grid Computing. 2007, 5(1): 99 – 126.

[18] Lublin U, Feitelson D G. The workload on parallel supercomputers: modeling the characteristics of rigid jobs[J]. Journal of Parallel and Distributed Computing. 2003, 63 (11): 1105 – 1122.

[19] Casanova H. Benefits and drawbacks of redundant batch requests[J]. Journal of Grid Computing, 2007, 5(2): 235 – 250.

作者简介



肖 鹏 男, 1979 年生于湖南湘潭, 讲师, 主要研究方向为网格计算, 并行与分布式系统, 可信计算.  
E-mail: xpeng4623@yahoo.com.cn



胡志刚 男, 1963 年生于山西孝义, 教授, 博导, 主要研究方向为分布式系统, 网格计算, 嵌入式系统.