

基于信任 QoS 增强的网格服务调度算法

张伟哲¹⁾ 方滨兴^{1),2)} 胡铭曾¹⁾ 张宏莉¹⁾

¹⁾(哈尔滨工业大学计算机科学与技术学院 哈尔滨 150001)

²⁾(国家计算机网络应急技术处理协调中心 北京 100029)

摘要 针对目前服务网格资源管理中存在的信任机制与调度机制分离的缺陷,基于网格信任模型与信任效益函数,讨论了信任 QoS 增强的计算服务调度问题。分析传统调度算法的缺陷,提出了基于信任关系的网格服务调度算法。该算法在保证服务性能 QoS 要求的同时兼顾信任 QoS 的要求。对基于信任关系的网格服务调度算法性能进行了多角度分析和大规模仿真实验。结果表明:该算法较传统的基于性能 QoS 的调度启发式和另外两种基于信任 QoS 的调度启发式而言,不仅具有较优的平均信任效益、总信任效益和较小的最终服务期,而且在失效服务数和系统吞吐率等方面具有较好的综合性能。

关键词 服务网格;服务调度;服务质量;信任模型;信任关系;信任效益函数

中图法分类号 TP393

A Trust-QoS Enhanced Grid Service Scheduling

ZHANG Wei-Zhe¹⁾ FANG Bin-Xing^{1),2)} HU Ming-Zeng¹⁾ ZHANG Hong-Li¹⁾

¹⁾(School of Computer Science and Technology, Harbin Institute of Technology, Harbin 150001)

²⁾(National Computer Network Emergency Response Technical Team/Coordination Center of China, Beijing 100029)

Abstract At present existing scheduling algorithms in service grid largely ignore the impact of trust mechanism. It is rather doubtful to adopt these algorithms in a realistic environment. Based on the grid trust model and trust utilization functions, a computational service scheduling problem based on trust QoS enhancement is proposed. A novel trust-QoS enhanced heuristic based on trust relationship is put forward. The algorithms are evaluated with large scale simulation. Simulation results demonstrate trust-driven heuristics perform better than time-driven heuristic. The algorithm based on trust relationship achieves better integrative performance on performance-QoS, trust-QoS, etc. than trust-driven Min-Min and Sufferage heuristic algorithms.

Keywords service grid; service scheduling; QoS; trust model; trust relation; trust utility function

1 引言

随着网络与通信水平的不断提高,并行计算系统由封闭的超级计算机向开放的网络计算系统(network computing system)发展。服务网格(service

grid)^[1]作为大规模的网络计算系统,将开放网格服务体系结构 OGSA 和 Web 服务资源框架 WSRF 相结合,以“服务”的方式来包装多机构虚拟组织中各种资源,统一以网格服务的形式提供给外界使用,解决动态、多机构的虚拟组织中资源共享与协同问题。

服务网格系统根据其设计目的广义上可以分为

收稿日期:2006-02-16;修改稿收到日期:2006-04-09. 本课题得到国家“九七三”重点基础研究发展规划项目基金(G2005CB321806)和国家自然科学基金(90412001)资助。张伟哲,男,1976 年生,博士研究生,讲师,主要研究方向为网络计算和信息安全。E-mail: wzzhang@hit.edu.cn. 方滨兴,男,1960 年生,博士,教授,博士生导师,CCF 高级会员,中国工程院院士,主要研究领域为信息安全和网络计算。胡铭曾,男,1935 年生,教授,博士生导师,主要研究领域为计算机体系结构和网络计算。张宏莉,女,1973 年生,博士,教授,主要研究领域为信息安全、网络计算。

计算服务网格、数据服务网格、信息服务网格、知识服务网格和设备服务网格等。其中计算服务网格通过计算服务接口对计算程序进行抽象、对服务使用者屏蔽计算服务实现细节,协调不同组织与机构高性能机群的计算能力解决大规模科学计算和工程应用问题。计算服务网格具备处理挑战性应用的潜力,然而良好的服务质量(QoS)保证和用户满意度必须借助于有效的网格服务管理(service manage),尤其是面向QoS的服务调度机制^[2,3]。

在大规模计算服务管理中,可量化的用户QoS要求分为性能QoS要求和信任QoS要求^[4]。其中,性能QoS包括最终服务期限等与时间相关的参数和精度(precision)等与准确性相关的参数,而信任QoS是用来评价服务信息的真正可信程度。然而,当前计算服务网格中调度机制与信任机制分离,多数基于QoS的服务调度机制和算法仅仅考虑性能QoS^[6~12],忽视了信任QoS的影响。事实上,服务请求者或提供者中任何一方由于信任缺失引发的恶意行为均会降低性能服务质量。例如:若服务请求者对服务资源恶意使用,将极大消耗服务提供者的资源,降低其提供服务的能力;反之,若服务提供者所提供的服务易失效、不能保障安全性,会降低服务请求的最终执行期限。因此,在保证性能QoS需求同时兼顾信任QoS要求、设计信任QoS增强的计算服务调度算法、形成融入信任机制的网格资源管理策略等方面亟待深入研究。

本文将针对此问题,在前期工作的基础上^[5],详细阐述网格中信任的基本概念,确立服务请求与提供者间的信任关系,采用信任效益函数量化不同信任关系下调度结果对服务质量的影响,并讨论信任QoS增强的计算服务调度问题。同时,提出一种基于信任QoS增强的计算服务调度策略,该策略在追求较好的性能QoS同时兼顾服务的信任QoS需求。

本文第2节介绍相关工作,同时进一步阐明本文与相关工作的差异和深入研究的意义;第3节介绍涉及的基本概念,讨论信任QoS增强的计算服务调度问题;第4节详细阐述一种基于信任关系的启发式服务调度算法;第5节给出本文算法实验设置和与实验结果;在最后一节中,我们对本文工作进行了总结并对后续工作进行了展望。

2 相关工作

本节首先介绍基于性能QoS的网格调度研究,

而后阐述与信任QoS相关的研究工作,最后指出当前面向QoS服务调度机制的局限性。

基于性能QoS的计算服务调度方面:Maheswaran等人^[6]在多管理域、大规模网络计算系统中引入了QoS驱动的资源管理策略,利用效益函数量化服务质量等级,提出作业级动态调度算法来优化性能QoS中最终服务期限和优先级。文献[7,8]提出了映射一组相互独立作业的策略。该策略采用重复映射以更好地适应网格计算环境下的动态性和自治性。算法通过定义效益函数,追求最终服务期限等性能QoS需求最优化。He等人^[9]通过改进传统的Min-Min作业调度启发式,提出了一个自适应、性能QoS驱动的Min-Min启发式算法。同样,Weng等人^[10]以最终服务期限和平均响应时间作为性能QoS要求,提出以Sufferage为基算法的Qsufferage算法,此算法性能在最终服务期限和平均响应时间方面显著优于Xsufferage、Max-Min等算法。此外,除上述面向作业级应用的计算服务调度,文献[11,12]针对服务网格中工作流(workflow)应用,深入研究了基于性能QoS的服务管理与调度算法。

基于信任QoS的计算服务调度方面:初期的网格服务安全研究中,Foster等人设计了面向计算服务网格和面向服务发现的安全架构^[13,14],为服务提供者的安全提出了细粒度访问控制模型^[15]。然而,人们仅着眼于服务提供者的安全,忽视服务提供者同样会存在服务信息缺乏可信度等行为,应为服务使用者提供必要的服务保障机制。因此,融入信任机制已经被公认为大规模网格服务安全管理的关键因素。Azzedin与Maheswaran^[16]等人首次将“信任”融入网格资源管理,提出考虑信任QoS的作业调度会引入额外负载并设计了负载最小化算法。Humphrey^[17]等人对具有信任意识的服务计算模型进行了深入研究。Abawajy^[18]等人提出的DFTS分布式失效容忍调度策略,通过复制作业的多个副本到不同站点来保证作业在网格环境下可靠运行。Song^[19]和Li^[20]提出了信任QoS驱动的网格信任调度框架和算法。此外,我们在基于信任QoS的计算服务调度方面取得了一些初步成果^[5],提出了两种信任驱动的网格作业调度启发式。

当前计算服务调度算法的研究存在局限性。服务使用者与服务提供者分布于网格不同管理域,任何一方的恶意行为均会严重影响服务质量。基于性能QoS的服务调度算法^[6~12]由于信任机制缺失而难以在真实的网格环境下获得理想的结果。然而,目

前基于信任 QoS 的多数研究工作着眼于安全与信任架构的确立^[14~18],扩展传统的作业调度算法来形成信任 QoS 驱动的调度算法^[5,18~20],对服务请求者与服务提供者间信任关系缺乏把握,导致算法难以同时兼顾性能 QoS 与信任 QoS 要求.

3 基本概念与问题描述

3.1 信任 QoS 定义

信任是一个非常复杂的主观概念,目前没有一致的定义^[21,22].本文给出信任的定义如下.

定义 1. 信任.由信任值表征的客观实体的身份和行为的可信度评估,信任值取决于实体可靠性、诚信和性能等.

基于信任定义,我们给出了计算服务网格系统中形式化的信任模型.

定义 2. 信任模型.计算服务网格中的信任模型为一个四元组 (O, A, R, f) ,其中 O 代表服务网格中参与计算的服务提供者(provider)与服务请求者(demander)的集合, A 为信任属性集合, R 代表服务计算主客体间的信任关系集合, f 代表不同信任属性与信任关系下的效益函数的集合.

定义 3. 信任属性.信任属性包含如下两个方面:(1)安全性(security).衡量服务提供者对服务请求者数据的真实性、保密性和完整性的保障程度.真实性保证运行数据未被恶意用户篡改,数据保密性保证某个用户所请求的数据不会传到别的用户手中,而数据的完整性使作业运行中间数据及结果无缺失;(2)可靠性(reliability).长时间执行的作业有可能因为某个服务失效导致运行失败甚至重启,造成系统资源浪费和系统性能低下,因此服务可靠性是网格作业可信运行的又一重要衡量手段.本文分别采用 PS 和 PR 量化服务提供者的安全与可靠属性, DS 和 DR 量化服务请求者的安全与可靠需求.

定义 4. 信任关系.根据调度过程中服务请求者对服务提供者信任值的要求,两者间的信任关系可以分为强(strong)信任关系、弱(weak)信任关系与无信任关系.

(1) 强信任关系是指调度时用户请求的安全与可靠性需求级别必须高于所分配服务的固有属性值,即 $PS > DS$ 且 $PR > DR$,如果不存在满足条件的服务,则此请求将被放弃.其最终效益值或者为最大,或者为零.

(2) 弱信任关系指调度算法尽量保证请求者信

任属性值高于提供者的信任属性值,此时作业可获得最大效益;否则可降低请求者的信任需求,但其信任效益值随之下降.

(3) 无信任关系指调度算法在调度过程中不考虑服务请求与提供者间的信任关系,仅以性能 QoS 为目标.

定义 5. 信任属性隶属度函数.表征不同信任属性与信任关系下用户对调度服务质量的满意程度,本文根据定义 4 中不同信任关系对最终效益值的要求,给出信任关系为 strong, weak, no 的服务请求 d_i 在服务提供者 p_j 上执行的安全性隶属度函数如下:

$$SecUtil_{\text{strong}}(i, j) = \begin{cases} 1, & \text{如果 } DS_i \leq PS_j \\ 0, & \text{其它} \end{cases} \quad (1)$$

$$SecUtil_{\text{weak}}(i, j) = \begin{cases} 1, & \text{如果 } DS_i \leq PS_j \\ 1 - (DS_i - PS_j) / (\max_level - PS_j), & \text{其它} \end{cases} \quad (2)$$

$$SecUtil_{\text{no}}(i, j) = DS_i / \max_level \quad (3)$$

式中, DS 和 PS 为枚举对象 $\{poor=0, low, better, \dots, max_level\}$.

同理,可靠性隶属度函数定义如下:

$$ReliUtil_{\text{strong}}(i, j) = \begin{cases} 1, & \text{如果 } DR_i \leq PR_{i,j} \\ 0, & \text{其它} \end{cases} \quad (4)$$

$$ReliUtil_{\text{weak}}(i, j) = \begin{cases} 1, & \text{如果 } DR_i \leq PR_{i,j} \\ \frac{1 - \frac{\exp(-(PR_{i,j} + 1 - DR_i))}{\exp(-(1 - DR_i))}}{1 - \frac{\exp(-1)}{\exp(-(1 - DR_i))}}, & \text{其它} \end{cases} \quad (5)$$

$$ReliUtil_{\text{no}}(i, j) = \frac{1 - \exp(-PR_{i,j})}{1 - \exp(-1)} \quad (6)$$

式中,服务请求者的可靠性需求 $DR_i \in (0, 1)$,而服务提供者可靠性 $PR_{i,j}$ 由服务请求 d_i 和服务提供者 p_j 共同决定.设开始时每个服务提供者固有失效率为 $FP_{j,0}$.随服务运行时间增加,服务提供者失效的概率逐渐增大,可靠度降低.设服务请求 d_i 在服务提供者 p_j 上的完成时间为 $CT_{i,j}$,则此服务的可靠度为 $PR_{i,j} = \exp(-CT_{i,j} \times FP_{j,0})$.

定义 6. 信任 QoS 效益函数.服务请求 d_i 对服务提供者 p_j 信任 QoS 满意度采用如下的函数评价:

$$TrustUtil = W \cdot R \quad (7)$$

式中, R 为安全与可靠性隶属度函数产生的评价矩阵 $(SecUtil(i, j), ReliUtil(i, j))^T$. $W = (w_1, w_2)$ 为权重矩阵($0 < w_1, w_2 < 1$), w_1, w_2 分别代表安全性和可靠性等在信任 QoS 要求中的不同权重,

$$w_1 + w_2 = 1.$$

3.2 信任 QoS 增强的服务调度问题

目前,国内外计算服务网格环境下的调度策略研究根据研究对象不同可以分为两类^[23]: (1) 工作流调度(workflow scheduling): 由传统机群环境下基于任务图(DAG)的调度问题演化而来。通过将服务请求抽象为粗粒度约束任务图,采用经济模型和数学规划策略构建可增值的复合服务,目的是优化网格环境下某个或某类服务请求的性能 QoS 需求; (2) 作业级调度(job scheduling): 其研究对象是相互独立的服务请求集合,用户提出的请求无前驱后继关系,目的是宏观上优化计算服务网格中多用户请求的整体服务质量,是高性能机群作业调度研究在计算服务网格环境下的延伸。本文研究对象为后者,即面向服务质量的作业级计算服务调度。有别于传统的作业级服务调度,本文提出的信任 QoS 增强的调度问题在优化性能 QoS 的同时兼顾信任 QoS 需求。

定义 7. 信任 QoS 增强的网格作业调度问题。设服务网格中存在 m 个计算服务提供者 $P = \{p_1, p_2, \dots, p_m\}$, n 个相互独立的服务请求 $D = \{d_1, d_2, \dots, d_n\}$, 服务请求在不同服务提供者上的预期运行时间记录在 ETC 矩阵中,矩阵中第 i 行第 j 列分量 $CT_{i,j}$ 表示作业请求 d_i 在资源 p_j 上的预估执行时间。求映射方案 $map = (a, s)$ (其中 $a: D \rightarrow P$ 为表示服务分配的映射, $a(i) = j$ 表示将请求 d_i 分配到服务 p_j 上, $s: \{(i, a(i)) | i \in T\} \rightarrow N = \{1, 2, 3, \dots\}$ 为调度函数, $s(i, j) = k$ 表示在服务请求 d_j 将在服务提供者 P_j 上第 k 个执行,使得

$$\text{Maximize } TrustUtil(map) \quad \& \&$$

$$\text{Minimize PerformUtil}(map) \quad (8)$$

式中信任 QoS 效益函数值越大表示作业映射后执行越稳定,执行结果愈加可信,而性能 QoS 效益函数可采用最终服务期限与平均响应时间等指标衡量,本文采用最终服务期限(makespan)表示用户的性能 QoS 满意度。

4 基于信任关系的启发式调度算法

信任机制与网格信息服务、安全服务和资源管理服务的融合在相关文献^[16, 24]中得到深入研究,限于篇幅,本文重点研究信任增强问题的服务调度算法。

本文提出的启发式算法基于如下基本思想:

(1) 服务请求者与服务提供者间存在强信任、弱信任与无信任三种信任关系,不同信任关系对效益函数影响不同。事实上,首先调度具有强信任需求的请求能够使其信任 QoS 需求在最大程度上得到满足。因此,有必要为三种信任关系设置调度优先级。另外,还设法为后分配的作业寻找可行位置,以期满足其信任 QoS 需求;(2) 显然,信任效益函数值越大表示映射后服务执行越稳定,执行结果愈加可信。然而,最大化作业集合的信任效益函数同时,应当保证作业集最终服务期限较小,为此给出基于简单加权法的优化加权综合效益函数,以保证映射方案满足综合服务质量约束:

$$\begin{aligned} WeightUtil(i, j) = \\ w_1 \times TrustUtil(i, j) / TrustUtil(i, \max) + \\ w_2 \times CT_{i,\min} / CT_{i,j} \\ \text{st. } w_1 + w_2 = 1, \end{aligned} \quad (9)$$

式中 w_1 和 w_2 是在信任和性能上的重要性权值。 $TrustUtil(i, \max)$ 就是请求 d_i 在所有服务提供者上最大信任效益。 $CT_{i,\min}$ 就是 d_i 在所有服务上的最小完成时间。

算法 1. 基于信任关系的计算服务调度启发式(TD Trust-Relation 算法)。

输入: 服务请求 D 与服务提供者 P 信息, ETC 矩阵

输出: 作业映射方案(map)

1. 初始化: 令 D 为所有请求的集合, P 为所有服务提供者的集合, 集合 CT 保存服务请求者-提供者对, 变量 k 用于计数;

2. 根据可靠性信任关系(strong, weak, no)将 D 分成三个不相交的子集合 $Cluster_1, Cluster_2, Cluster_3$;

3. 令 $k=1$;
4. Repeat
5. If ($Cluster_k$ 不为空)
6. CT 置为空;
7. For $Cluster_k$ 中每一个请求 d_i
8. For P 中每一个服务提供者 p_j
9. If p_j 的作业队列上存在一个可行位置满足 d_i 的信任需求
10. 计算 d_i 和在 p_j 上的加权综合效益值;
11. Endif
12. Endfor.
13. If 所有资源都无法满足 d_i 的信任需求
14. 将 d_i 从 $Cluster_k$ 中除去;
15. Endif.
16. Endfor.;
17. 从 CT 中选择最大加权综合效益值的服务请求者-服务提供者对 (d_i^*, p_j^*) ;

18. 将 d_i^* 分配到 p_j^* 的服务作业队列可用位置上，并从 $Cluster_k$ 中除去 d_i^* ;
19. Endif.
20. If ($Cluster_k$ 为空)
21. $k = k + 1$;
22. Endif.
23. Until ($k > 3$).

算法首先将集合 T 被分成 3 个服务请求子集： $Cluster_1, Cluster_2, Cluster_3$ ，分别由强、弱和无信任需求的服务请求构成。接下来，对每一个子集重复执行步 5~步 22，直到子集中的所有作业请求都被分配或放弃。步 9 检查如果将 d_i 分配到 p_j 服务队列上的可行位置， d_i 的最低信任要求能否得到满足。可行位置定义为：如果信任关系为无信任，则可行位置就是 p_j 的服务队列末尾；其它情况下，可行位置在最靠近 p_j 服务队列末尾的某个位置（在该位置上， d_i 的强或弱信任需求能够被满足，且该处以后位置上的所有服务请求强弱信任需求，不会因为 d_i 的插入而无法满足）。该算法时间复杂度为 $O(NM)$ ，其中 N 为服务请求 D 的数目， M 为服务提供者 P 的数目。

5 仿真实验结果及性能分析

本节首先给出仿真实验环境及算法设置，而后将本文提出的基于信任关系的计算服务调度启发式算法 TD Trust-Relation 与 He 等人提出的基于性能 QoS 的调度算法 QoS Min-Min^[9]、信任驱动的 TD Min-Min 和 TD Sufferage 启发式^[5,19,20]比较，验证本文算法在信任 QoS、性能 QoS 和服务失效率等方面的有效性。

5.1 仿真实验环境及算法设置

在仿真实验中，本文采用网格模拟器仿真由 $m=20$ 与 50 个计算服务提供者组成的网格系统，研究 40~500 个独立服务请求集合的调度情况。实验的硬件环境为 PIV 2.8GHz 的 CPU 和 512MB 内存，软件环境为 Windows XP Professional 操作系统。

服务提供者属性。 安全级别 PS 设置为四个级别 $\{poor, low, medium, high\}$ ，每个服务的安全级别在此区间内随机分布。资源的单位时间失效率 FP 在区间 $[0.0001, 0.0015]$ 上随机分布。

服务请求者属性。 服务安全需求级别 DS 设置与服务提供者安全级别设置相同。服务可靠性需求级别 DR 在强、弱信任关系的情况下根据如下分布

生成： $DR = (0.9 + 0.1 \times rand) \exp\{10^{-4} \times (\text{服务请求数}/\text{服务提供者数})\}$ ，其中 $rand$ 均匀分布在 $[0, 1]$ 。预期执行时间 ETC 根据文献[25]中方案取 $\mu_{task} = \mu_{mach} = 100, V_{task} = V_{mach} = 0.6$ 。

信任关系。 设置变量 $1 \leq V_q \leq 4$ 控制服务请求者与服务提供者间的信任关系；生成一个 $[0, 1]$ 间随机数，如果该数小于 $0.25V_q$ ，则称两者具有强信任关系；该数小于 $0.5V_q$ ，具有弱信任关系；否则，为无信任关系。信任效益函数(式(7))与加权信任效益函数(式(9))中 w_1, w_2 取值均为 0.5。

5.2 实验结果及性能分析

本节从信任 QoS(总信任效益值、平均信任效益值)、性能 QoS(最终服务期)、失效服务数和系统吞吐率等多个性能指标对上述启发式算法进行综合评价。

总信任效益是系统对调度结果中所有成功调度的服务请求所获得的信任效益之和，即

$$TotalTrustUtil = \sum_{i=1}^n TrustUtil(d_i, a(d_i)) \quad (10)$$

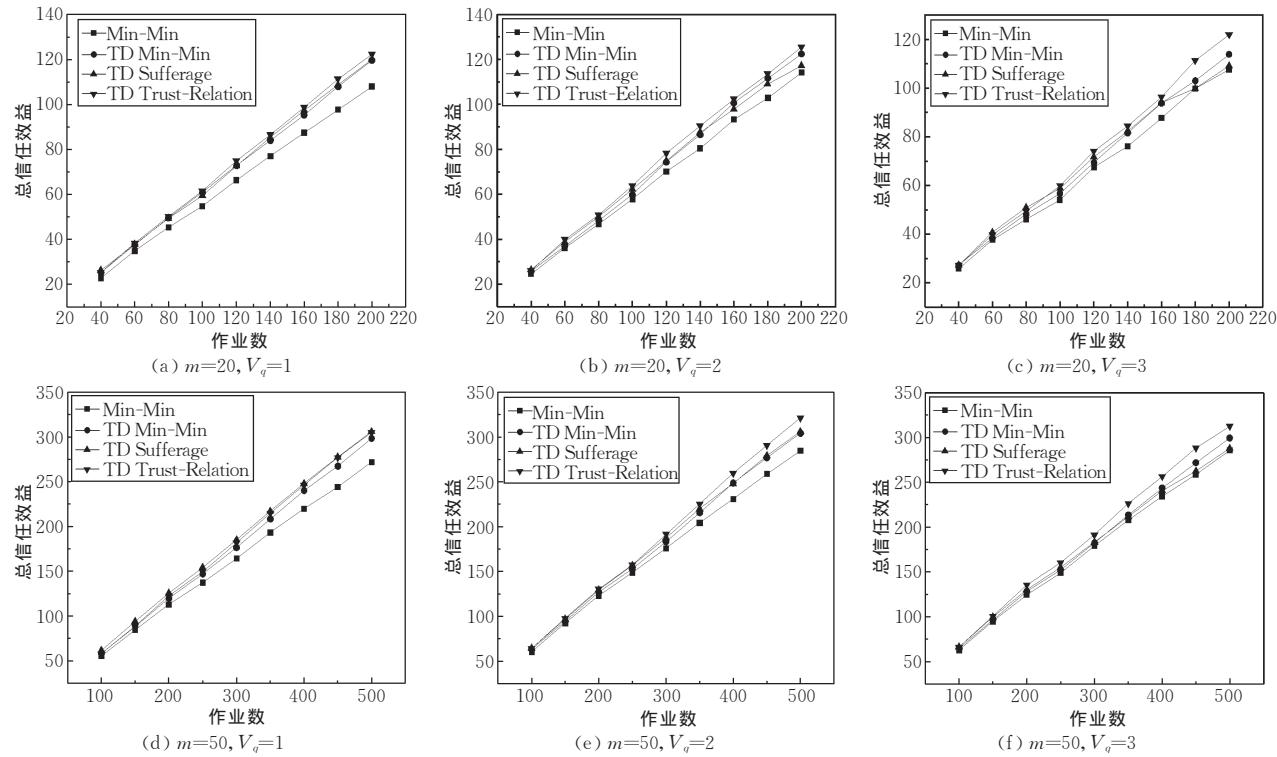
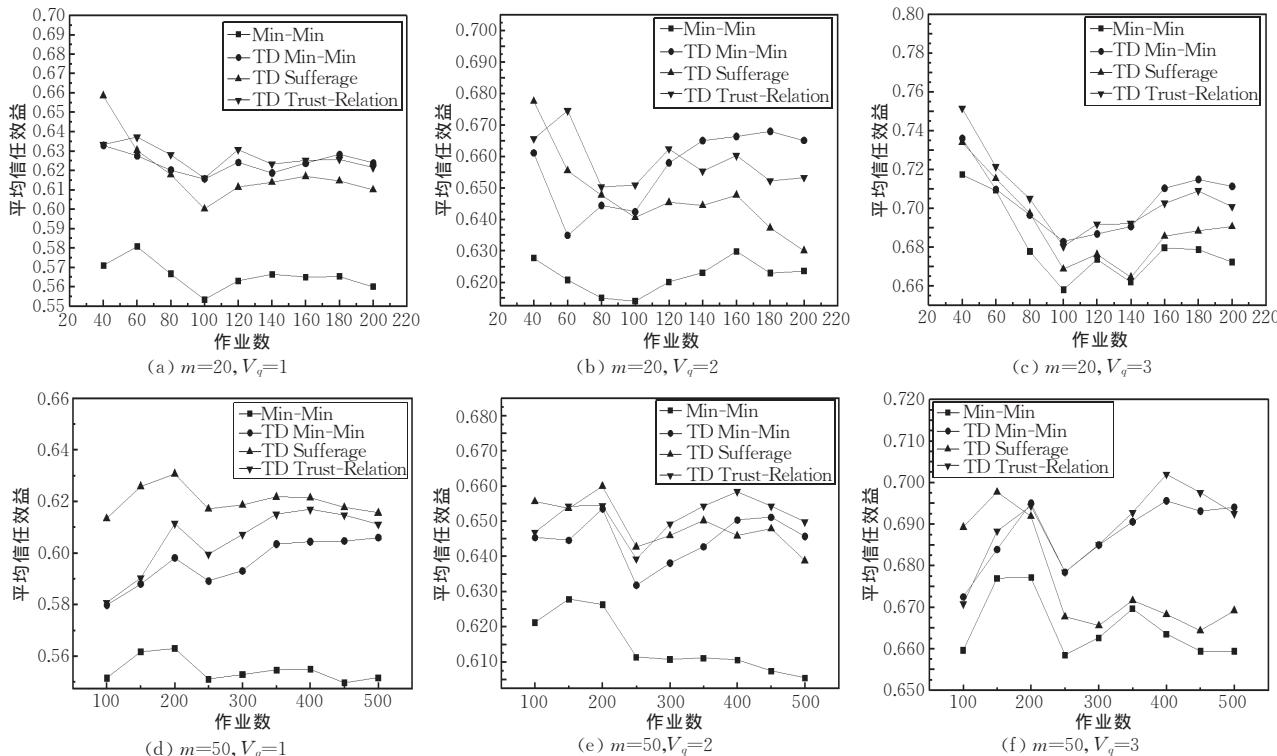
是调度算法性能系统级评价指标。获得总信任效益大的调度算法能够为系统提供较好的信任服务。

从图 1 中可以看出，三种信任驱动的启发式所获得的总信任效益要明显地优于无信任驱动的 Min-Min。三种信任驱动启发式算法中，TD Trust-Relation 在总信任效益较 TD Min-Min 和 TD Sufferage 有较为明显的优势。TD Trust-Relation 的优异表现主要得益于将所有服务请求按照其信任关系从系统角度进行规划。

TD Sufferage 总信任效益受作业数和 V_q 影响较大。在 V_q 较小时，TD Sufferage 的总信任效益优于 TD Min-Min，某些情况下甚至优于 TD Trust-Relation ($m=50, V_q=1$)，但随着 V_q 的增加 TD Sufferage 的性能下降较快甚至接近于 TD Min-Min ($m=50, V_q=3$)。TD Min-Min 获得的总信任效益较为稳定，它与 TD Trust-Relation 保持着 2.4% 到 5.1% 的差距。

平均信任效益是所有被成功调度的服务请求所获得的信任效益均值。

从图 2 中可以看出信任驱动的启发式所获得的平均信任效益要明显地优于无信任驱动的 Min-Min。三种启发式算法中 TD Trust-Relation 和 TD Min-Min 在各种情况下均能获得较大的平均信任效益。总服务请求数和 V_q 的变化对它们的平均信任效益

图 1 $m=20$ 与 50 时 V_q 变化对总信任效益的影响图 2 $m=20$ 与 50 时 V_q 变化对平均信任效益的影响

影响很小。某些情况下 TD Trust-Relation 显著优于 TD Min-Min 的平均信任效益(如 $m=50, V_q=2$)，而在其它的情况下，这两个启发式则差别不大。

TD Sufferage 在一些情况下能够获得很好的平均信任效益(如 $m=50, V_q=1$)。但是 TD Sufferage 对总服务请求数和 V_q 的变化较为敏感，当总作业数和

V_q 增大时, TD Sufferage 性能下降较为明显。

最终服务期为所有服务完成时间的最大值即

$$Makespan = \max_{i=1}^n \{CT_{i,a(t_i)}\} \quad (11)$$

Min-Min 只对服务的完成时间进行优化, 因此它能够获得最小的 Makespan。但它的优势相对于 TD Min-Min 和 TD Trust-Relation 并不明显。TD Min-Min 使用了加权综合效益函数, 在选择服务提供者时, 同时考虑了信任效益和服务的完成时间, 因此能获得较好的 Makespan。TD Trust-Relation 也

采用了加权信任效益函数对服务进行选择, 但由于它不考虑时间因素而优先分配信任关系较强的服务请求, 因而获得的 Makespan 较 TD Min-Min 略大, 在 0.7%~1% 之间。TD Sufferage 较少地考虑了服务的完成时间因素(只在两个服务具有相同的亏损值时, 才选择其中具有较小完成时间的服务进行调度), 因此它获得的 Makespan 最差, 平均为其它三种启发式的 2 倍左右。

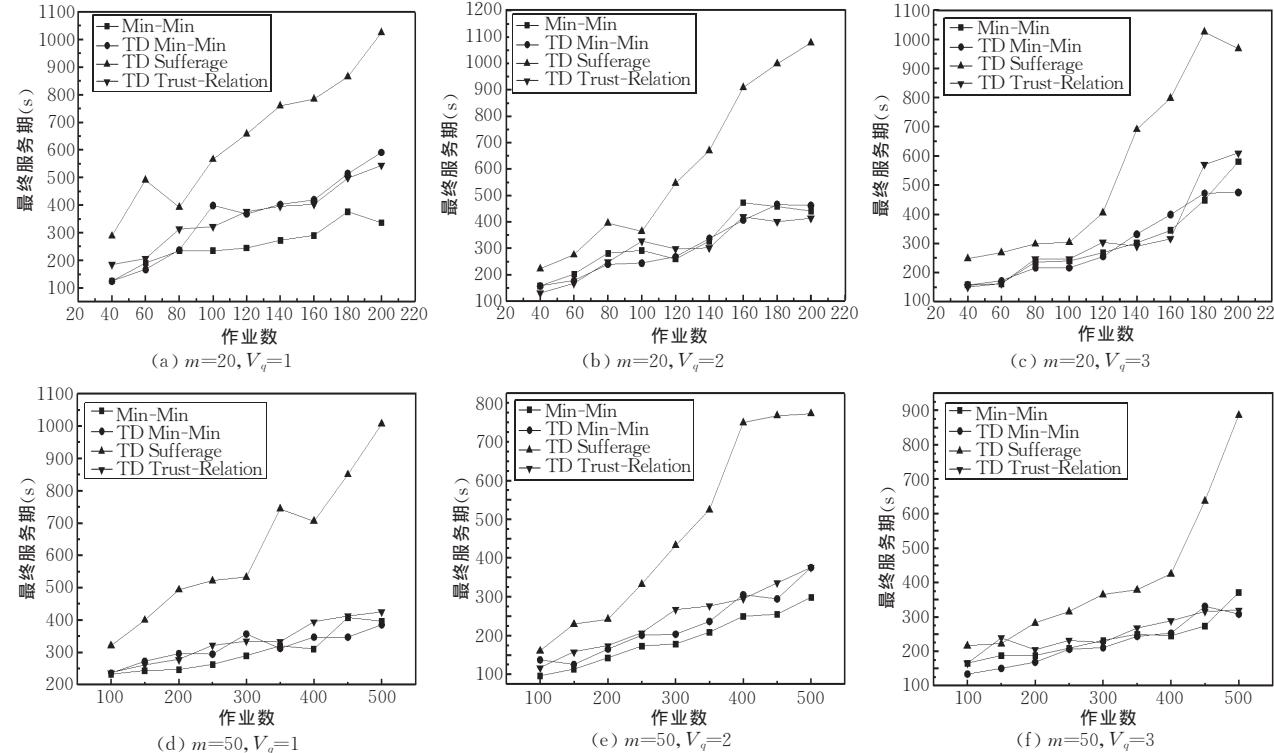


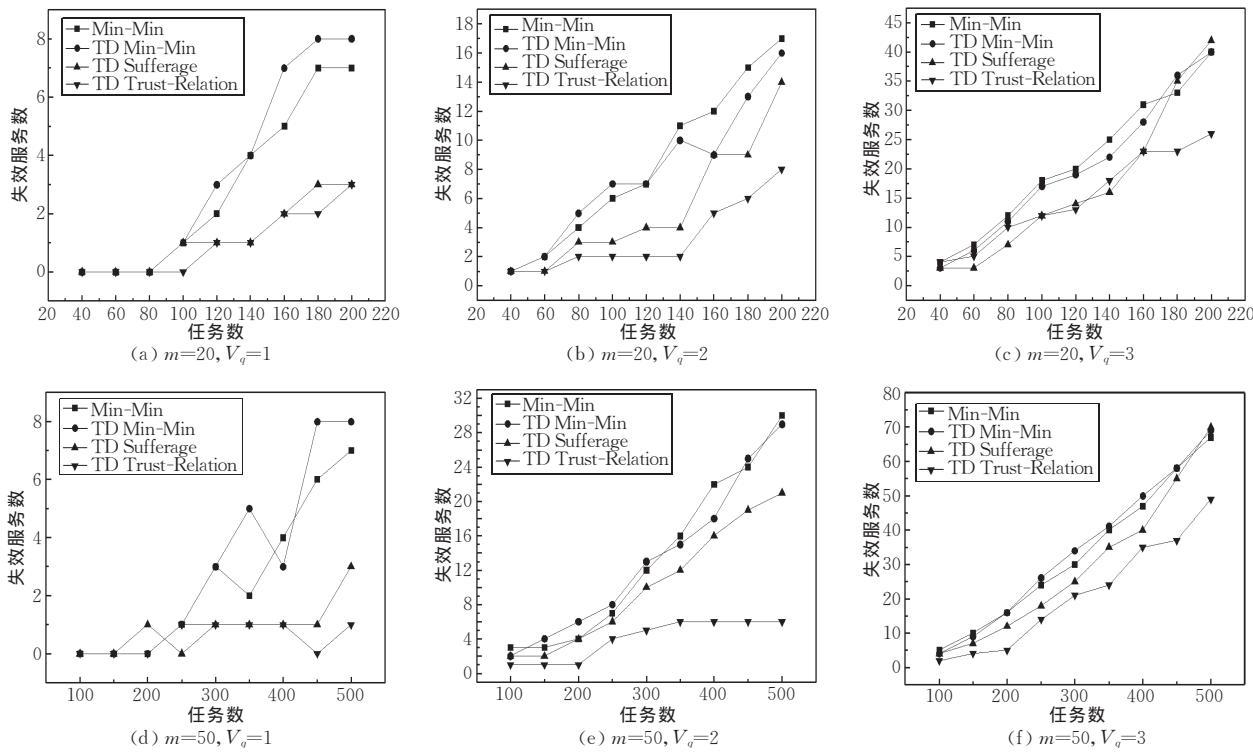
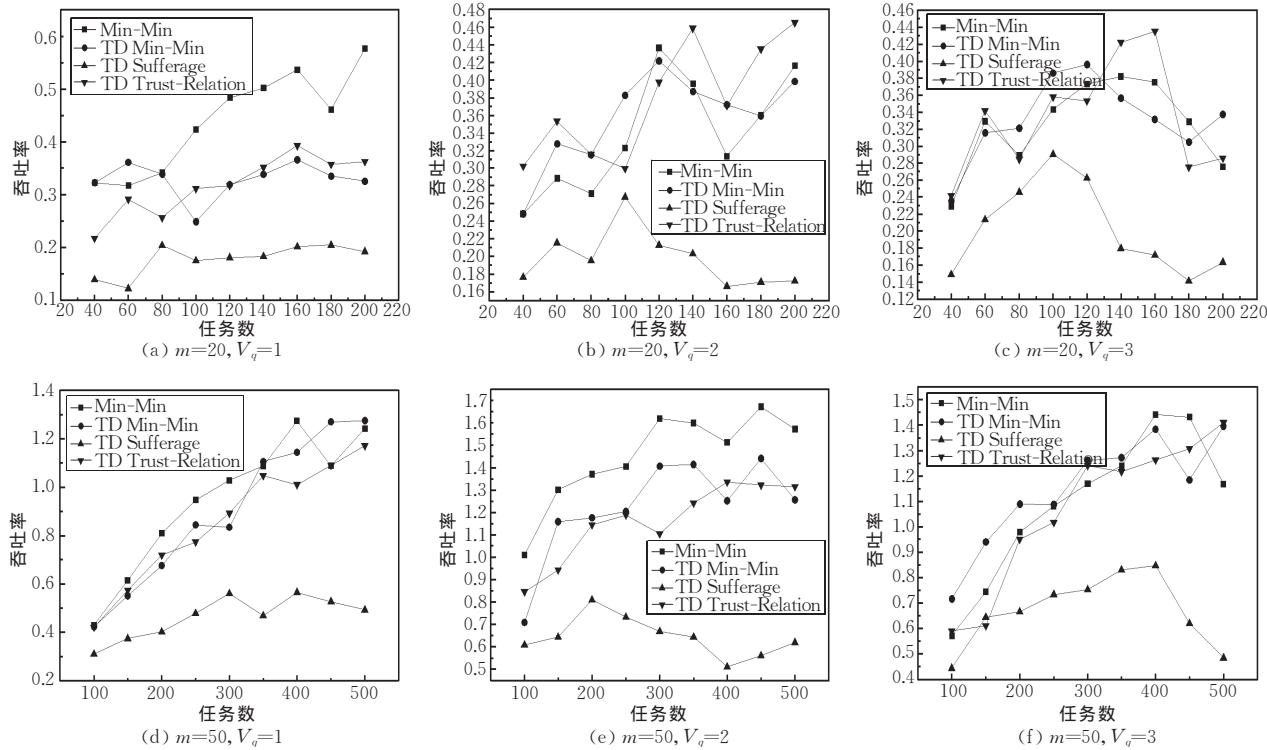
图 3 $m=20$ 与 50 时 V_q 变化对最终服务期的影响

失效服务数是指调度过程中由于信任关系无法满足而被迫丢弃的服务请求数目。

从图 4 中可以看出 TD Trust-Relation 在几乎所有的情况下, 都能获得最小的失效服务数。最好的情况下失效服务数目仅为其它启发式的 1/2 左右($m=20, V_q=2; m=50, V_q=2$)。主要由于 TD Trust-Relation 使用两种方法来减少失效服务数, 增大信任效益:(1)具有较强信任需求的服务请求得以优先调度;(2)为后分配的请求寻找可行位置, 使得由于被分配在服务队列的后端而使其信任需求无法满足的请求也可能被成功调度。此外, TD Trust-Relation 在 $V_q=2$ 时的优势最为明显。这是由于当所有服务中具有强信任需求的请求较少时, 通过以上两种方法而受益的服务数较少, 作用并不明显。随着具有强信

任关系的请求数的增加, TD Trust-Relation 的优势逐渐增大。但当具有强信任关系的请求占所有请求比例过大时, 根据信任关系类型划分服务的作用逐渐减小, TD Trust-Relation 的优势变小。因而, TD Trust-Relation 更适合应用在强信任关系服务比例适中的情况。这也较为符合实际的计算服务网格中的信任关系。

TD Sufferage 也能丢弃较少的请求, 这是由于它优先调度紧迫服务请求, 使得它们能够及时得以调度, 从而有效地减少了失效服务数。Min-Min 和 TD Min-Min 丢弃的请求较多是由于不区分服务的信任关系, 可能是根据加权信任效益优先调度具有较小完成时间和较大信任效益但信任关系较弱的请求, 因而延误对信任关系较强的服务的调度, 使得这些服务请求由于信任关系无法被满足而被丢弃。

图 4 $m=20$ 与 50 时 V_q 变化对失效服务数的影响图 5 $m=20$ 与 50 时 V_q 变化对吞吐率影响

吞吐率为单位时间内系统处理的服务请求数, 定义为成功调度的服务请求数与最终服务期的值之比。启发式可能由于成功地调度了较多请求而产生较大的最终服务期的值, 通过吞吐率可以对启发式的效率从更为客观的角度进行评价。

尽管 Min-Min 可能丢弃较多的请求, 但由于它对服务完成时间的充分关注使得它在大多数的情况下能够获得最大的吞吐率。TD Min-Min 通过加权综合效益对完成时间加以考虑, 也能获得较好的吞吐率。虽然 TD Trust-Relation 采用的划分

服务集合和为请求寻找可行位置的方法可能使其获得的 *Makespan* 较大,但由于它丢弃较少的服务请求,使得更多的请求得以成功分配,因此它的吞吐率与 TD Min-Min 相比,差别不大。TD Sufferage 的 *Makespan* 较大,因此它的吞吐率最小。

6 结束语

在服务网格环境中,信任机制与服务调度机制有效融合是大规模分布式资源管理难题之一。本文通过引入网格环境下的信任机制,量化服务提供者与服务请求者间不同信任关系下信任效益函数,提出了信任 QoS 增强的计算服务网格调度问题。在此基础上,本文提出了一种基于信任关系的计算服务调度启发式算法,算法同时兼顾网格服务的性能 QoS 与信任 QoS 要求。仿真实验证明三种信任驱动的启发式调度算法在信任 QoS、性能 QoS、失效服务数目、吞吐量等方面要优于传统的仅基于性能 QoS 的 Min-Min 算法,其中本文提出的考虑信任关系的计算服务调度启发式算法具有最好的综合性能。下一步的研究方向是根据不同用户的多种 QoS 需求,开发基于多目标优化的服务调度算法。

参 考 文 献

- 1 Foster I., Kesselman C., Nick M., Tuecke S.. The physiology of the grid: An open Grid Services architecture for distributed systems integration. Open Grid Service Infrastructure WG, Global Grid Forum, June 22, 2002 (extended version of Grid Services for Distributed System Integration). <http://www.globus.org/alliance/publications/papers/chervenakGrid2005.pdf>
- 2 Papazoglou M. P., Georgakopoulos D.. Service-oriented computing. Communications of the ACM, 2003, 46(10): 24~28
- 3 Papazoglou M. P.. Service-oriented computing: Concepts characteristics, and directions. In: Proceedings of the 4th International Conference on Web Information System Engineering, Roma, Italy, 2003, 3~10
- 4 Sabata B., Chatterjee S., Davis M., Sydir J., Lawrence T.. Taxonomy for QoS specifications. In: Proceedings of the IEEE Computer Society 3rd International Workshop on Object-Oriented Real-Time Dependable Systems, Newport Beach, California, 1997, 100~107
- 5 Zhang Wei-Zhe, Liu Xin-Ran, Yun Xiao-Chun, Zhang Hong-Li, Hu Ming-Ceng, Liu Kai-Peng. Trust-driven job scheduling heuristics for computing grid. Journal of Communications, 2006, 27(2): 73~79(in Chinese)
- (张伟哲,刘欣然,云晓春,张宏莉,胡铭曾,刘凯鹏. 信任驱动的网格作业调度算法. 通信学报, 2006, 27(2): 73~79)
- 6 Maheswaran M.. Quality of service driven resource management algorithms for network computing. In: Proceedings of the International Conference on Parallel and Distributed Processing Technology, Nevada, USA, 1999, 1090~1096
- 7 Ding Qing, Chen Guo-Liang, Gu Jun. A unified resource mapping strategy in computational grid environments. Journal of Software, 2002, 13(7): 1303~1308(in Chinese)
(丁 篓,陈国良,顾 钧. 计算网格环境下一个统一的资源映射策略. 软件学报, 2002, 13(7): 1303~1308)
- 8 Ding Q., Chen G.. A benefit function mapping heuristic for a class of meta-tasks in grid environments. In: Proceedings of the 1st IEEE International Symposium on Cluster Computing and the Grid, Brisbane, Australia, 2001, 654~658
- 9 He X., Sun X., Gregor V. L.. QoS guided Min-min heuristic for grid task scheduling. Journal of Computer Science and Technology, 2003, 18(4): 442~451
- 10 Weng C., Lu X.. Heuristic scheduling for bag-of-tasks applications in combination with QoS in the computational grid. Future Generation Computer Systems, 2005, 21(2): 271~280
- 11 Gu Qing-Fan, Wu Jie-Yi, Zhang Sa-Bing. Scheduling mechanism of composite grid service with multiple QoS constraints. Journal of Computer Integrated Manufacturing Systems, 2005, 11(12): 1775~1780(in Chinese)
(谷清范,吴介一,张帆兵. 网格环境下多服务质量约束的复合服务调度机制. 计算机集成制造系统(CIMS), 2005, 11(12): 1775~1780)
- 12 Jia Yu, Rajkumar Buyya, Chen Khong Tham. QoS-based scheduling of workflow applications on service grids. In: Proceedings of the 1st IEEE International Conference on e-Science and Grid Computing, Melbourne, Australia, 2005, 1~9
- 13 Foster I., Kesselman C., Tsudik G., Tuecke S.. A security architecture for computational grids. In: Proceedings of the 5th ACM Conference on Computer and Communication Security, NY, USA, 1998, 83~92
- 14 Czerwinski S. E., Zhao B. Y., Hodes T. D., Joseph A. D., Katz R. H.. An architecture for a secure service discovery service. In: Proceedings of the 5th Annual International Conference on Mobile Computing and Networks, Seattle, Wash, 1999, 24~35
- 15 Butt A. R., Adabala S., Kapadia N. H., Figueiredo R., Jose A. B. Fortes. Fine-Grain access control for securing shared resources in computational grids. In: Proceedings of the 16th International Parallel and Distributed Processing Symposium, Florida, USA, 2002, 22~29
- 16 Azzedin F., Maheswaran M.. Integrating trust into grid resource management systems. In: Proceedings of the 2002 International Conference on Parallel Processing, Vancouver, British Columbia, Canada, 2002, 47~54
- 17 Humphrey M., Thompson M. R.. Security implications of typical grid computing usage scenario. In: Proceedings of the IEEE HPDC, San Francisco, CA, 2001, 95~103

- 18 Abawajy J. H.. Fault-tolerant scheduling policy for grid computing systems. In: Proceedings of the 18th IEEE International Parallel & Distributed Processing Symposium, Santa Fe, New Mexico, 2004, 50~58
- 19 Song S. , Kwok Y. K. , Hwang K.. Trusted job scheduling in open computational grids: Security-driven heuristics and a fast genetic algorithm. In: Proceedings of the 19th IEEE International Parallel & Distributed Processing Symposium, Denver, CO, USA, 2005, 33~40
- 20 Li K. , He Y. , Liu X.. Security-driven scheduling algorithms based on eigentrust in grid. In: Proceedings of the 6th International Conference of Parallel and Distributed Computing Applications and Technologies, Denver, USA, 2005, 1068~1072
- 21 Misztal B.. Trust in Modern Societies. Cambridge MA: Polity Press, 1996
- 22 Grandison T. , Sloman M.. A survey of trust in internet appli-
- cations. IEEE Communications Surveys & Tutorials, 2000, 3 (4): 2~16
- 23 Sabin G. , Kettimuthu R. , Rajan A. , Sadayappan P.. Scheduling of parallel jobs in a heterogeneous multisite environment. In: Feitelson D. G. , Rudolph L. , Schwiegelshohn U. eds. Job Scheduling Strategies for Parallel Processing, Lecture Notes Computer Science, Berlin: Springer-Verlag, 2003, 2862: 87~104
- 24 Ching Lin, Vijay Varadharajan, Yan Wang, Vineet Pruthi. Enhancing grid security with trust management. In: Proceedings of the IEEE SCC, Shanghai, 2004, 303~310
- 25 Ali S. , Siegel H. J. , Maheswaran M. , Hengsen D.. Task execution time modeling for heterogeneous computing systems. In: Proceedings of the IPDPS Workshop on Heterogeneous Computing. Cancun, Mexico, 2000, 185~199



ZHANG Wei-Zhe, born in 1976, Ph. D. candidate, lecturer. His research interests include network computing and network security.

FANG Bin-Xing, born in 1960, Ph. D. , professor, Ph. D. supervisor, member of Chinese Academy of Engi-

neering. His research interests include network security and network computing.

HU Ming-Zeng, born in 1935, professor and Ph. D. supervisor. His research interests include computer architecture and network computing.

ZHANG Hong-Li, born in 1973, Ph. D. , professor and Ph. D. supervisor. Her research interests include information security and network computing.

Background

This work is supported by the National Natural Science Foundation of China under grant No. 90412001 and the National Basic Research Program(973 Program) of China under grant No. G2005CB321806. The key problem in both of the programs is the network resource management and task scheduling in large-scale computational service grid environment.

Traditionally the aim of task scheduling is to optimize the performance QoS of tasks such as makespan. However,

grid resources are distributed in different grid-domains and out of control, which requires that trust mechanisms are incorporated into different scheduling policies. Thus, we should not only focus on the performance QoS but also the trust QoS when we design the tasks scheduling systems. This paper provides a trust-QoS enhanced computational service scheduling to harmonize the performance and trust QoS as a pre-research to enhance the computational service scheduling systems.