CROWN: 面向服务的网格中间件系统与信任管理 *

怀进鹏** 胡春明 李建欣 孙海龙 沃天宇

(北京航空航天大学计算机学院, 北京 100083)

摘要 针对大量网格资源的分布、自治等特点, 给出了基于 SOA 的服务网格体系结构, 提出了基于层叠网的分布式网格资源组织与管理机制、访问控制策略的自动协商、信任管理和信任协商机制, 并研制了中间件系统 CROWN; 通过部署 CROWN 中间件系统建立了广域试验环境并部署了中尺度天气系统降水预报、海量多媒体数据处理平台、血液流动温度场显示、科学数据网格以及数字巡天图检索等多个网格应用, 应用经验表明, 该中间件系统能够支撑以计算密集型、数据密集型和海量信息分析与处理为特征的典型应用.

关键词 网格计算 服务网格中间件 资源管理 分布式访问控制 信任管理 CROWN

1 引言

网格计算(Grid computing)[1]是近年来出现的一种面向Internet的分布式计算模式, 其目的是使分布在网络上的各类异构、自治的资源按照应用需要进行能力集成、共享和协同工作, 形成动态虚拟组织, 实现跨自治域的可信资源组织、管理和利用, 提高资源的综合利用效率, 有效地满足面向互联网的复杂应用对大规模计算能力、海量数据处理和信息服务的需求, 并最终实现跨领域、多学科研究与应用的资源共享与集成.

近年来, 网格计算得到了许多政府机构、学术界和产业界的共同关注, 美国自然科学基金会自 1997 年起先后启动了先进计算伙伴计划PACI、分布式万亿级

收稿日期: 2006-04-15; 接受日期: 2006-06-21

* 国家自然科学基金(批准号: 90412011)、国家杰出青年科学基金(批准号: 60525209)、国家重点基础研究发展计划(批准号: 2005CB321803)和国家高技术研究发展计划(批准号: 2005AA119010) 资助项目

** E-mail: huaijp@buaa.edu.cn

计算设施DTF和增强的分布式万亿级计算设施EDTF计划[2]，美国能源部 2001 年启动了Science Grid计划[3]，美国国防部 20 世纪 90 年代末启动了规模宏大的全球信息网格GIG计划，拟于 2020 年完成；欧盟于 2000 年和 2001 年先后开展了包括欧洲网格计划(EuroGrid)、数据网格计划(DataGrid)，以及EGEE, CoreGRID等在内的数十个网格重大项目；英国 2001 年启动了e-Science计划[4]. 目前美欧的主要大学和科研机构间已基本形成了基于网络进行资源共享与协作的实验环境，开展了许多典型应用. 自 20 世纪末以来，我国也实施了多项网格研究计划或重大项目，如CNGrid, ChinaGrid和NSFCGrid等，投入力度和应用领域不断加强，取得了显著的结果.

随着网格技术研究和应用的深入，不同网格系统的互操作性和松耦合集成问题逐步得到重视；同时，Web服务技术应用及相关标准化工作逐步开展，面向服务的体系结构(service oriented architecture，简称SOA)得到了学术界和工业界的广泛共识，直接促进了网格与Web服务技术的融合，尤其是开放网格服务体系结构(open Grid service architecture，简称OGSA)[5]和Web服务资源框架(Web service resource framework，简称WSRF)[6]的出现进一步推进了网格技术的发展，使基于SOA的服务网格体系结构和中间件技术迅速兴起，并成为网格计算研究与应用的基础之一. SOA和相应的标准化工作为网格技术研究实践提供了一种重要的方法，它通过将资源封装为具有统一接口的服务，支持一致的服务管理协议，一定程度上解决了资源的异构性问题；同时，它通过对资源服务的发现和动态绑定来访问资源，也在一定程度上解决了资源的动态性问题. 但是，由于网络资源的异构性、局部自治性和动态分布性，现有的资源服务组织管理机制仍难以满足应用需求，已有的安全机制亦难以适应隐私保护、“陌生”者间资源共享所提出的动态信任等需求.

事实上，无论是网格计算，还是对等计算(peer-to-peer computing)和泛在计算(ubiquitous computing)均从不同侧面对面向网络的分布计算模式的积极探索，其共同之处是如何基于开放、动态的网络计算环境，实现可信、协同的资源(包括计算资源、数据资源、软件资源以及服务资源等)共享和有效利用. 进一步，由于包括无线移动网络在内的互联网是一个不断成长的无序系统，节点行为表现出相当的随机性，网络上各类分布异构资源缺乏有效的安全控制和有效的组织管理机制，尤其是大量分布的异构资源不断更新与扩展，自治系统之间的关联关系不断动态变化，使网络资源的共享和协作难度增大，可协同、可管理和可信任是其 3 个基本问题. 具体地，可协同性问题是指如何跨自治域进行资源共享和协同工作；可管理性问题是指如何将异构、庞大的网络资源进行有效地聚合与管理，这是建立面向 Internet 的分布式计算环境的前提条件；可信任性问题是指面对大量不可控和缺乏信任基础的服务资源，如何保证共享和协同的资源之间所需建立

的可靠和相互信赖关系问题.

自 2002 年以来, 我们在Web服务应用支撑环境(WebSASE)[7]的研究基础上, 先后研制了符合OGSA/OGSI规范的服务网格中间件WebSASE4G[8]和服务网格系统CROWN[1)][9,10], 尤其是围绕上述 3 个基本问题重点研究了服务网格资源管理、分布式访问控制与跨自治域信任管理、网格服务工作流, 以及基于服务组合的网络软件开发等服务网格关键技术, 建立了覆盖北京、长沙、重庆、香港和英国利兹等 5 个城市 11 个科研机构的广域网格试验平台, 有效地支撑了以计算密集型、数据密集型和海量信息分析与处理为特征的网格应用, 如中尺度天气系统降水预报(AREM)、海量多媒体数据处理平台(MDP)、血液流动温度场显示(gViz)、科学数据网格(SDG), 以及数字巡天图检索等多个网格应用.

本文首先介绍了基于 SOA 的服务网格体系结构及 CROWN 中间件, 针对网格资源的组织管理、访问控制与信任管理问题, 重点探讨了基于层叠网的分布式网格资源组织与管理机制、访问控制策略的自动协商、信任管理和信任协商机制. 为方便, 除非特殊说明, 本文中 CROWN 中间件是指一组支持服务网格运行管理的中间件组件; CROWN 系统(环境)则指基于 CROWN 中间件构建起来的包括计算资源、数据资源、存储资源以及仪器设备等在内的网格计算环境.

2　CROWN 体系结构与设计原理

2.1　体系结构

网格计算产生于 20 世纪 90 年代中期的元计算概念 [1]. 10 年来, 网格技术经历了从元计算到计算网格和服务网格的转变, 网格功能和技术也在不断拓展和泛化, 但网格的基本体系结构并未出现本质的变化, 其中 2001 年提出的五层沙漏结构 [11]得到了学术界的认同. 随着网格应用和标准化工作的开展, 网格体系结构及其支撑技术成为一个重要研究内容, 其中OGSA[5]是一个面向服务的体系结构, 它采用服务作为资源的统一封装形式, 通过定义标准的网格服务访问协议和统一的服务扩展方式使网格具有较好的开放性和服务扩展能力; WSRF[6]是对OGSA框架中服务接口和服务交互协议的优化和重新定义, 进一步使OGSA和Web服务规范兼容, 实现网格技术与Web服务技术的平滑融合.

实际上, 五层沙漏结构仅给出了服务网格的一个抽象功能结构, 而OGSA/WSRF 则可视为在此基础上提出的一种基于服务概念的实现框架和基本网格服务接口, 它们均未给出构造服务网格的设计原理、中间件组件及其功能定义, 亦未讨论服务网格的访问控制和信任协商机制. 本文工作针对基于

1) CROWN 的英文全称是 China Research and Development Environment Over Wide-area Network, 即以网络为基础的科学活动环境. 该项目是国家自然科学基金委员会 2004 年网格重大专项中的一个重大项目

OGSA/WSRF 的服务网格展开，通过分析网格应用方式，讨论服务网格中间件 CROWN 的体系结构、设计原理与关键技术.

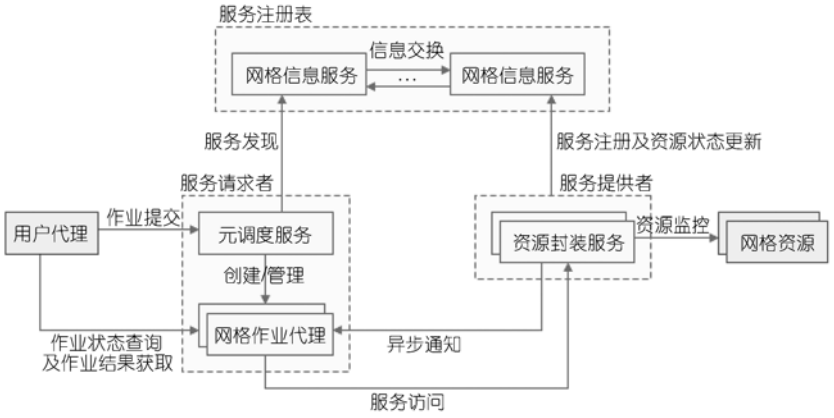

图 1 基于 SOA 的服务网格应用模式

通常，基于 OGSA/WSRF 的服务网格包括共性服务、资源封装服务和应用特定服务等三类，其中共性服务是网格中间件的主要组成部分，包括网格作业代理、元调度服务，以及网格信息服务等一系列基础组件. 在实际网格应用中(见图 1)，用户首先通过代理向元调度提交作业，并可通过访问网格作业代理查询作业状态、获取作业结果；元调度服务解析作业描述语言中给出的资源需求，并通过网格信息服务进行服务发现和作业调度，同时调用资源服务并跟踪状态.

基于上述应用模式，我们给出了 CROWN 的层次体系结构及其与五层沙漏结构的比较(如图 2 所示)，其中服务网格中间件主要覆盖图中的资源层、汇聚层和应用层. 在系统设计实现中，资源层中的资源封装服务、汇聚层中的网格信息服务、元调度服务、网格作业代理服务，以及部分领域相关服务均与 OGSA 兼容，并通过网格信息服务提供注册与发现机制；通过网格应用支持工具实现汇聚层与应用层的交互(如针对领域或特定应用的开发框架、专用程序，以及基于 Web 的网格门户等).

2.2 CROWN 中间件的设计原理

早期的网格研发工作大多针对确定的应用，如 Cactus[12]，AppLeS[13]，Nimrod[14]以及 Ninf[15]等，但一些工作已开始尝试基于通用中间件为不同应用构造底层软件基础设施，如 Condor[16]，Globus[17]和 Legion[18]等；自 OGSA 提出后，许多研究工作的重点开始转移，OGSA/WSRF 体系结构成为服务网格中间件构建的主要参考标准，并形成了两种典型的服务网格中间件：一种是支持服务运行和管

理的基本软件服务器和模块, 如基于Web服务的GT3/4 和OMII、基于CORBA的GriT[19]和基于Jini的ICENI[20]等; 另一种则是针对特定功能设计的通用网格服务, 如用于异构数据库互连的OGSA-DAI[21]以及用于群组授权的CAS[22]等.

图 2　基于 CROWN 环境的服务网格层次结构

CROWN 中间件是应用网格开发与运行的关键层, 其主要特点是: 遵循 OGSA/WSRF 体系结构, 结合应用需求和 OGSA/WSRF 安全结构的局限性, 在中间件设计中突出考虑了网格资源组织与动态管理机制, 实现对网格服务的开发、部署、运行、监控和管理等全生命周期的支持; 提出了与之适应的安全结构以及分布式资源的访问控制技术与信任管理机制, 有效地支撑松耦合环境下异构资源的共享与协同. 它由如下两大类 11 种模块组成:

(1) 节点服务器(node server)　提供网格服务的基本运行环境, 它不仅对底层的异构资源进行服务化封装, 还为网格服务提供了运行基础(如处理 SOAP 消息、管理服务实例、提供生命期管理与通知机制等); 在系统实现上, Node Server 集成了远程服务热部署、系统资源监控、日志管理与服务器远程管理等功能, 并通过配置安全处理模块, 提供细粒度的用户身份鉴别、资源访问控制、信任管理与信任协商等功能, 可有效地保护资源使用的安全性和用户的隐私信息.

(2) 资源定位与描述服务(resource locating & description service, 简称 RLDS)是一个提供服务注册与发现功能的分布式信息服务系统, 它通过信息交换和拓

扑维护协议建立多个 RLDS 实例之间层次结构, 动态构成逻辑独立的层叠网, 在网格资源动态变化时提高资源管理和服务发现的效率(详见 3.2 节).

(3) 调度服务(CROWN scheduler) 按照一定的策略对用户提交的作业进行排队和调度, 并通过 RLDS 获取当前网格系统的服务部署和作业状态等信息, 进行服务发现, 并按照预定义的一组策略(如随机策略、负载均衡策略、最近距离策略等)进行服务匹配, 代理用户完成服务调用; CROWN Scheduler 支持 POSIX 应用程序和网格服务调用两类作业类型, 并通过 JSDL 统一描述作业, 支持作业的服务质量和安全性要求.

(4) 安全通信与身份鉴别(CROWN CommSec) 为网格服务提供基础安全通信, 作为 CROWN Node Server 的可选插件, 表现为一个共性的身份鉴别和证书验证 Web 服务, 同时为用户提供了证书链的构造和验证功能, 管理人员可通过配置预定义的策略文件, 灵活地定义网格服务复杂多变的安全需求, 具有独立性、可扩展性和灵活性.

(5) 授权服务(CROWN Authz) 设计了基于 XACML 的授权策略描述语言, 具有较强的授权策略表达能力, 为网格服务提供集中的授权决策和策略管理功能, 支持多粒度的访问控制和域授权策略的定制, 并与服务授权策略相结合进行多级授权.

(6) 信任证管理子系统(CROWN CredMan) 支持用户信任证管理, 它通过代理证书的方式, 解决基于 Portal 的作业提交与执行和移动场景中的身份信息获取问题; 同时可满足多资源主体参与的网格协同与共享时的身份一致性需求.

(7) 身份映射与信任证转换系统(CROWN CredFed) 用于解决网格中异构自治域间安全基础设施导致的信任证类型不一致问题, 允许管理员通过配置信任证转换策略, 实现跨域的身份映射, 支持 KerberosV5 和 X.509 两种信任证的相互转换与映射.

(8) 信任管理与协商子系统(CROWN ATN) 用于建立开放网络中陌生方信任关系, 服务于主体自治性和隐私信息保护, 尤其是提供了无法依赖可信第三方时的安全决策与信任管理机制.

(9) 服务网格门户(CROWN Portal)与客户端框架(rich client framework) 它通过提供统一的用户界面, 支持应用网格的参数配置与作业提交, 并生成基于 JSDL 的任务描述提交给服务代理, 显示处理结果. 其中 CROWN Portal 支持基于 Web 的交互需求较弱的应用模式, 而应用客户端框架(rich client framework)支持复杂的应用需求(如可视化处理), 并可进行定制扩充, 支持用户快速开发和部署特定应用展现形式.

(10) 网格服务集成开发环境(CROWN designer) 是基于 Eclipse 设计的网格服务开发和部署工具, 提供了一组开发向导, 支持易用的服务封装与配置, 利用

Node Server 中远程热部署功能实现服务封装单元(GAR 文件)的拖放式部署, 并提供了网格服务工作流的图形化建模工具, 以提高网格服务的开发效率. 进一步, 我们还将集成基于服务组合的软件设计开发工具, 支持更多的网格应用开发.

(11) 监控工具(CROWN monitor)　获取、存储和分析各种不同网格实体的信息(事件), 并提供基于 Eclipse RCP 的客户端交互方式, 通过图形方式直观地显示当前网格环境的节点状态信息, 并可通过调整其控制参数对网格系统的运行进行动态可视化监控.

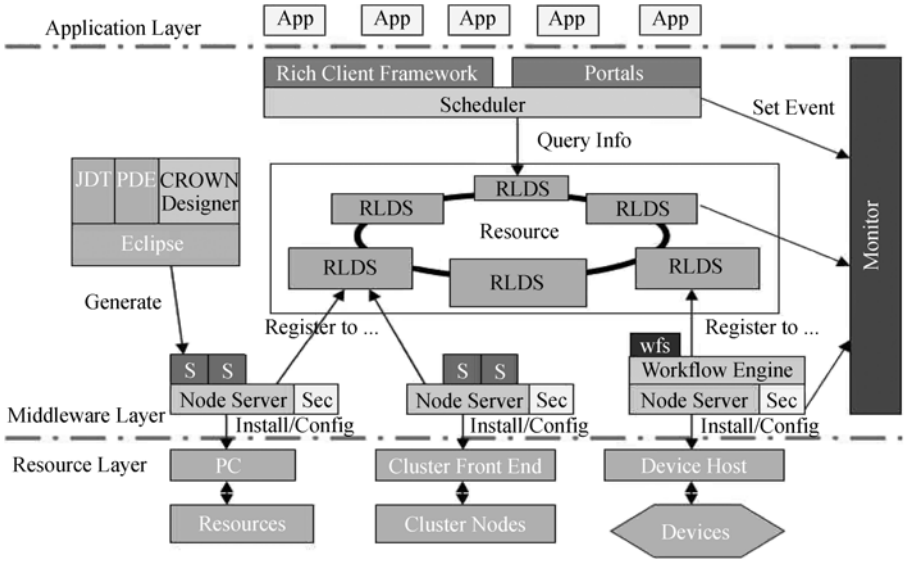

图 3　基于 CROWN 中间件的服务网格设计原理

基于上述中间件模块, 服务网格 CROWN 的设计原理(见图 3), 而服务网格环境的建立则通过 CROWN 中间件连接资源层的各类资源, 其中最上层的应用网格通过 Web Portal 或定制的客户端界面向网格提交作业, 求解用户问题. 具体应用时, 首先在各类异构资源之上部署 Node Server, 支持资源服务的部署和运行管理; 其次, 根据网格资源的分布特点确定自治域的划分并部署多个 RLDS 实例, 通过高效的拓扑管理协议进行实例间交互以形成分布式的网格信息服务系统; 第三, 部署 CROWN 调度服务, 从网格客户端接受作业请求, 根据用户需求进行服务发现与作业分配, 完成问题求解; 最后, 通过监控工具与辅助开发工具, 简化服务网格的构建以及面向服务网格的应用开发.

限于篇幅, 本文重点讨论 CROWN 环境中的资源组织与管理结构、安全机制.

3 基于层叠网的网格资源管理机制

3.1 CROWN 环境的资源组织与管理结构

由于网格系统连接了大量分布异构、动态变化的资源, 为实现高效地资源检索、共享和聚合, 资源组织和管理机制自然地成为服务网格的一个基本问题. 在 CROWN 中, 资源来自不同的自治域, 具有一定的组织管理关系, 为提高资源的组织管理效率, 我们提出了基于区域、域、节点和资源信息网关的资源组织与管理模式(见图 4), 其中:

(1) 节点(node): 各类分散的异构资源封装为若干 CROWN 节点, 通过在节点上部署各类服务以实现对异构资源的封装与一致访问;

(2) 域(domain): 根据现实中的资源隶属或使用关系, 将 CROWN 节点划分为若干的管理域, 每个域内通过部署信息服务, 提供域内资源/服务的注册与发现功能; 域间资源可按照隶属关系进行注册, 形成层次结构.

(3) 区域(region): 采用不同的资源组织和管理协议以及安全基础设施的域可划分为不同的区域, 多个区域之间可按更灵活的方式交互信息、共享资源, 但区域之间通过资源信息网关(resource information gateway)进行一对一的互联, 实现不同网格系统之间的资源互操作.

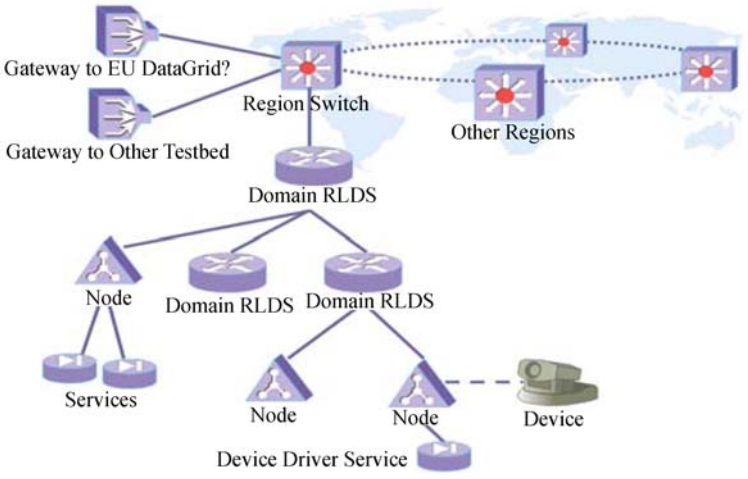

图 4 CROWN 资源组织与管理的拓扑结构

图 4 给出了一种静态的组织域管理结构, 但随着网格规模的逐步扩大, 参与网格环境的自治域数量、资源总数和种类随之增长, 因此如何在网格规模增长和动态更新的同时设计动态高效地资源组织管理机制, 这是网格资源管理的一个关键问题, 下面我们讨论基于层叠网的资源组织管理.

3.2　S-Club: 基于层叠网的网格资源组织与管理机制

近年来, 为了提高物理网络的性能和服务质量, 基于现有底层网络和特定的逻辑准则构建新“网络”——层叠网(overlay)——的技术得到重视和应用, 实验结果表明层叠网拓扑结构对路由服务的性能有直接的影响 [23]. 在提高网络性能(如可靠性等)和扩展功能方面, 出现了弹性层叠网(RON, resilient overlay network)[24]、服务层叠网(SON, service overlay network)[25]和OverQoS[26]等. 通常, 层叠网的主要设计方法是(1) 按需确立节点: 按照一定的逻辑准则寻找物理网络的特殊节点作为层叠网节点, 并可基于这些准则定义适用于上层网络的协议; (2) 建立虚拟链接结构: 通过建立若干个层叠网链接, 将分散的层叠网节点连接起来, 形成有序的拓扑结构, 并在层叠网的生命期内维护这种拓扑结构; (3) 设计通信路由协议: 主要是基于层叠网节点的消息转发实现节点间通信, 通过定义层叠网节点之间路由协议实现预定的功能.

20 世纪 90 年代后期, 网络复杂系统的研究引起很大反响 [27~29], 其中的研究表明, 虽然大量自主节点组成的互联网行为呈现了相当的随机性, 但在一定的应用目标聚合下, 其行为有一定的内在规律, 如小世界(small-world)和尺度无关(Scale-free)等特性. 通过分析上述规律, 不仅对理解和研究资源聚合及其行为具有意义, 而且对资源组织管理也有重要的价值. 我们也注意到, 在使用网格服务时用户通常依据特定的服务类别进行服务发现, 它说明网格服务发现的过程不是盲目的, 可以通过网格服务的分类来建立一种有效的结构和高效的服务发现算法. 基于上述分析, 我们提出了基于层叠网的网格信息服务管理机制, 即建立Service-Club的层叠网(简称S-Club).

为了提高服务发现的效率, 减少网络的综合开销, 需要合理地限定服务发现的请求检索范围. 在 S-Club 的设计中, 我们基于现有 GIS 所构成的底层网络, 依据每个 GIS 实例注册的服务类别信息, 将 GIS 纳入不同的服务类别组, 建立 GIS 实例之间的逻辑连接形成层叠网. 每个 GIS 实例所构成的组将含有一类服务的 GIS 聚合在一起, 其成员具有共同的特征, 类似人类社会中“俱乐部”, 这也是称为 Service-Club 的原因(如图 5).

图 5 所示的 S-Club 中, 空心和实心节点代表不同的 GIS 实例, 实心的 GIS 实例(*C*, *D*, *E* 和 *G*)表示其服务注册信息中由含有类型为 *T* 的服务构成一个层叠网(如图中虚线所示). 层叠网建立后, 服务发现请求被直接转发到 Club 的成员, 从而降低了查询的响应时间和查询的综合网络开销. 但是, Club 建立后, 由于服务部署的动态性和各自治域管理的独立性, 新的 GIS 实例可以加入 Club, 一些 GIS 实例可能退出, 需要动态地维护 Club 结构, 这将产生额外的网络开销. 因此, 如何平衡检索响应时间和综合网络开销是保证系统和网络性能的关键, 而所需构

建的 S-Club 数量及其示例数是影响性能的基本因素.

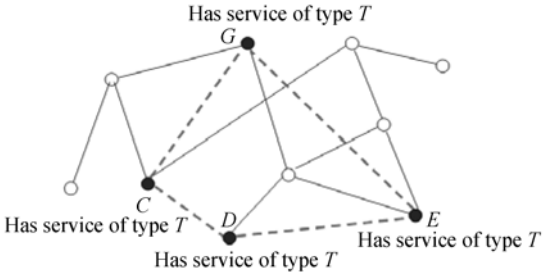

图 5 S-Club 示意

我们注意到: 由于网格环境中不同服务类型的使用频度不同, 因此, 我们基于服务类型的使用频度构建和维护同类 S-Club. 为了记录不同服务的调用情况, 识别出分布环境中服务的请求频度, 被调用的每个服务向其注册的信息服务进行报告, 每个 GIS 实例在本地为每个注册的服务类型维护其使用频度, 为此, 我们有:

定义 3.1 对任意一个确定 GIS 实例, 其服务类型 T 的使用频度 UF (usage frequency)是服务类型 T 被调用的次数占该 GIS 实例记录到的服务调用总数的比值, 即

$$UF(T)=\frac{N(T)}{M}.$$

实际上, 网格服务发现的过程并不是盲目的, 而是呈现一定的规律, 通过建立基于层叠网的网格服务分类结构, 可以合理地控制和选择服务检索范围, 提高服务发现效率. 因此, 在计算 GIS 示例的服务类型的 UF 后, 按照降序排列, 然后按其频度高的服务类型创建相应的 Club, 为此, 我们通过频度及其阈值可以引导 Club 构建方式和服务检索的范围.

定义 3.2 设 UF 是某 GIS 示例中服务类型 T 的使用频度, 频度阈值 (frequency threshold) α%指按降序排列 T 的所有 UF 值后, 建立 T 的 S-Club 的门限值, 即选取前 α%所对应的服务类型建立 Club.

定义 3.1 中 $UF(T)$体现了服务类型 T 的局部使用频度, 而定义 3.2 给出了建立 Club 的依据. 为基于服务类型的局部使用频度估算其在全局的使用频度, 我们设计了一种投票机制, 允许 GIS 间交互本地服务类型的使用频度信息, GIS 实例依据投票过程计算支持率δ, 并与设定的构造阈值比较, 决定是否为其服务类型创建 Club, Club 构建方法如下:

(1) 创建 Club 前: 首先每个 GIS 在尝试为 UF 排在前 α%的服务类型建立 Club,

并询问其他 GIS 实例是否也愿意为其建立 Club; 其他 GIS 实例根据本地 *UF* 排序队列, 确认 *T* 是否排序在前 *α*%, 并作出响应; 依据这些投票结果, 发起创建 Club 的 GIS 实例计算支持率, 并与预先设定的构造阈值进行比较, 决定是否为 *T* 创建 Club.

(2) 创建Club时: Club成员将独立探测与其他成员之间的网络距离(如两点间的延迟)作为建立层叠网拓扑结构的依据. 我们的CROWN采用最小生成树(minimum spanning tree, 简称MST)[30]作为S-Club的层叠网内部拓扑结构(见图 6), MST具有*O*(*n*)的时间复杂度为: 可以确保层叠网节点的连通性、层叠网内部请求转发和资源状态更新时代价最小.

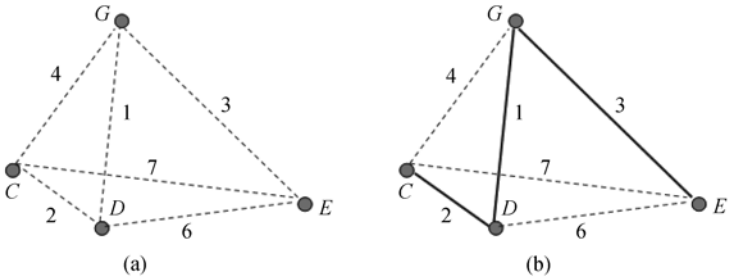

图 6　基于 MST 的 S-Club 的拓扑构造

(3) S-Club创建后: 借鉴Narada[31], 我们采用成员管理协议进行区块检测(partition detection)和拓扑恢复(topology recovery), 其基本方法是: 每个成员维护一个列表, 记录其他成员当前状态, 当新成员加入或原有成员退出Club时, 同步更新该列表; 每个Club成员周期性产生刷新消息(refresh message), 并在Club拓扑中传播这些变化信息, 刷新消息包含一个顺序增加的序号, 由GIS实例进行比较来确定信息的有效性, 具体地: (a) 当GIS实例*i*收到来自*j*的刷新消息后, 基于文献[31]的算法, *i*更新自己的成员状态列表; (b) 当一个成员主动离开时, 将其直接邻居列表发送给其他邻居节点, 这些邻居节点为在其同列表中的其他成员间建立层叠网连接, 以确保Club的内部拓扑不会因为某一成员的离开而分割成两个子图; (c) 当因网络或主机故障使成员被动离开时, *i*无法通知其他邻居节点, 需要Club基于刷新消息进行故障检测和恢复.

由于上述成员进入和退出的机制无法确保成员间的拓扑结构的性质, 成员的动态变化会降低 Club 内消息传递性能, 我们给出了 Club 定期进行拓扑重构机制, 即重新执行分布式拓扑构造算法, 以恢复到预定的拓扑结构, 提高层叠网内部的消息传递效率.

(4) S-Club 的删除: 当一个服务的访问频度下降到一定程度后, 需要删除为

其创建的 Club, 其原理类似于创建过程, 通过本地频度监控与全局投票相结合的方法实现.

(5) 在服务发现时, 用户将请求发给最近邻的 GIS 服务, 若该类型的 Club 已建立, 则直接将请求转发给 Club 中任何一个成员, 该成员将请求转发给 Club 中其他成员; 若尚未建立, 则在底层的 GIS 网络中进行洪泛(flooding)查询.

在仿真实验中, 我们将服务发现过程建模为基于泊松分布(Poisson distribution)的随机过程, 使用平均响应时间ART和总通信开销TTO这两个性能指标, 对分析比S-Club和MDS-like(采用MDS2[32]描述的服务发现机制)系统的性能.

定义 3.3 平均响应时间(average response time, 简称 ART)是指系统从收到请求到完成处理并返回结果的总时间的平均值.

定义 3.4 总通信开销(total traffic overhead, 简称 TTO)指某一时段内进行服务发现所需的网络通信和维护 Club 所需的网络通信的报文长度总和.

首先, 我们选取 n = 100, 200 和 400 三个不同的网络规模, 通过分别执行 10000 到 60000 个服务发现请求比较其性能变化(试验结果如图 7 所示), 结果表明: 随着请求数量的上升, S-Club 与 MDS-like 的总体网络开销基本呈线性上升趋势, 且规模网络规模越大上升速度越快; 除 S-Club 机制在开始阶段略有下降外, 平均响应时间并不随总请求数的上升而变化. 和 MDS-like 系统相比, S-Club 可有效减少网络消息量和响应时间, 如在 n = 200 且请求数为 60000 时, S-Club 降低了 55%的消息量和 27%的平均响应时间.

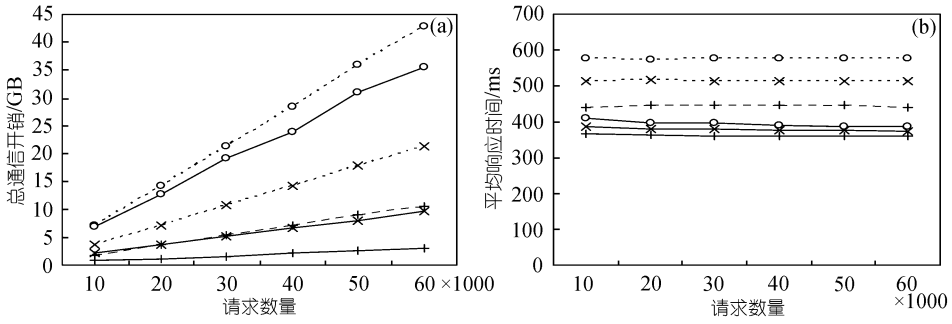

图 7 S-Club 与 MDS-like 系统的性能随请求数量的变化

--+-- MDS-like, n = 100; --×-- MDS-like, n = 200; --○-- MDS-like, n = 400; —┼— S-Club, n = 100; —×— S-Club, n = 200; —○— S-Club, n = 400

另一个试验研究 S-Club 机制的伸缩性. 我们分别选取 n = 100, 200 直到 1000 等 10 组不同的网络规模, 比较其平均响应时间(试验结果如图 8(a)所示), 结果表明: 随着网络规模的增加, MDS-like 与 S-Club 系统的平均响应时间都在增长, 但 S-Club 的增长幅度远小于 MDS-like 的增长幅度; 图 8(b)表示 S-Club 同 MDS-like

的系统比较后的平均响应时间优化率，$OptimizationRatio_{ART} = \frac{ART_M - ART_S}{ART_M}$，其中 ARTM 和 ARTS 分别是 MDS-like 和 S-Club 系统的平均响应时间，从图中可以看出，随着网络规模的扩大，S-Club 机制相对与 MDS-like 系统的优势更为明显，在网络规模为 n = 1000 时, S-Club 仍能减少 27% ~ 35%的平均响应时间.

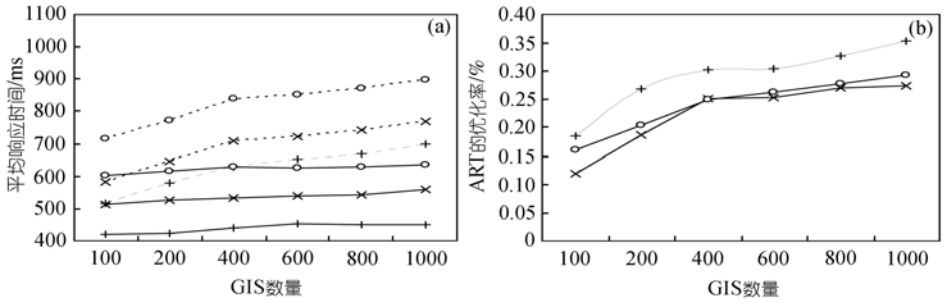

图 8　S-Club 的可伸缩性及相对 MDS-like 系统的优化率

(a) --+-- MDS-like, p = 0.2; --×-- MDS-like, p = 0.4; --○-- MDS-like, p = 0.8; —┼— S-Club, p = 0.2; —×— S-Club, p = 0.4; —○— S-Club, p = 0.8. (b) --+-- p = 0.2; —×— p = 0.4; —○— p = 0.8

在Web服务与网格环境中，进行服务注册与发现是构建松耦合系统的重要内容之一，如前所述，UDDI[33]，UDDIe[34]，R-GMA[35]和Hawkeye[36]等均采用集中式或多镜像的集中式信息存储和检索，结构简单，维护成本小，但这些方法易于形成单点性能瓶颈；Globus所采用的MDS-1/MDS-2[32]将资源信息组织成层状结构，并维护一个全局的分布式目录信息树，这种方案可以分散查询负载，避免造成单点故障，但是当系统规模扩大时，层状结构的扇出度较大，从而降低了服务发现的效率；K*Grid中GAIS[37]是与OGSA/OGSI兼容的分布式网格信息服务系统，它基于MDS构建的层叠网实现GIS实例间的VO信息交换与共享，但其层叠网仍是一个静态结构且仅用来交换VO信息，且若所需的VO信息不在接到用户请求的GAIS中，该GAIS实例会通过洪泛机制询问其他的GAIS实例；资源路由器 [38]是织女星(VEGA)网格采用基于层次结构的分布式信息服务体系结构，将全局信息分散在主干网上的多个BGRNS节点上，能够在一定程度上适应信息的规模变化. 和这些已有工作相比，我们提出的基于S-Club的资源管理与服务发现机制通过识别服务的访问频度，可动态地在信息服务实例间建立层叠网，能有效地提高服务发现的效率.

3.3　ROST: 可信的远程服务热部署机制

由于网格环境中资源和服务的动态性较强，不仅体现在资源的自主进入或退出导致的相应服务需要进行部署和反部署，而且也体现在用户对某类资源或

服务的访问需求的动态变化以及安全需求. 因此, 基于层叠网的网格资源组织模式 S-Club, 我们提出了服务的远程部署机制, 以适应这种动态管理需求, 并能动态实现网格负载的迁移与负载均衡.

目前, 远程服务热部署作为服务网格中一个重要问题还未被充分研究. 已有的远程服务部署均以“冷部署”的方式存在, 即当部署一个服务时, 需要重新启动服务容器才能使服务生效, 但同时必须停止正在运行的服务, 当服务容器重新启动之后, 这些服务才能继续甚至重新启动作业的处理过程, 开销很大. 而有效的服务热部署机制可以提高系统的效率, 实现负载均衡和作业迁移等性能优化, 例如, 在基于 CROWN 的生物信息应用中, 有许多计算密集型应用程序服务(如 BLAST)需要被部署到远程的服务容器中, 当应用中多个作业在短时间内聚集时, 计算节点易于超载, 如果节点能够将这些服务的副本部署到其他的远程节点, 则作业负载可被均衡.

为方便, 我们称部署服务的节点或用户为部署者(deployer), 将远程的服务运行环境称为目标服务容器, 负责运行和管理所部署的服务. 除了上述需求外, 当服务被部署到远程的服务容器过程中, 还面临着许多安全问题, 因为部署者提供的服务可能包括恶意代码, 目标服务容器也可能提供虚假服务; 同时, 部署者的安全策略和服务容器的安全策略可能不兼容. 因此, 在动态开放的网格环境中, 我们无法期待服务部署者及目标服务容器均已建立了所需的信任关系, 而基于典型的 PKI 建立部署中的信任关系代价较高.

针对上述问题, 我们提出了一种可信的远程服务热部署(remote and hot service deployment with trustworthiness, 简称 ROST)机制(见图 9), 它通过动态更新运行

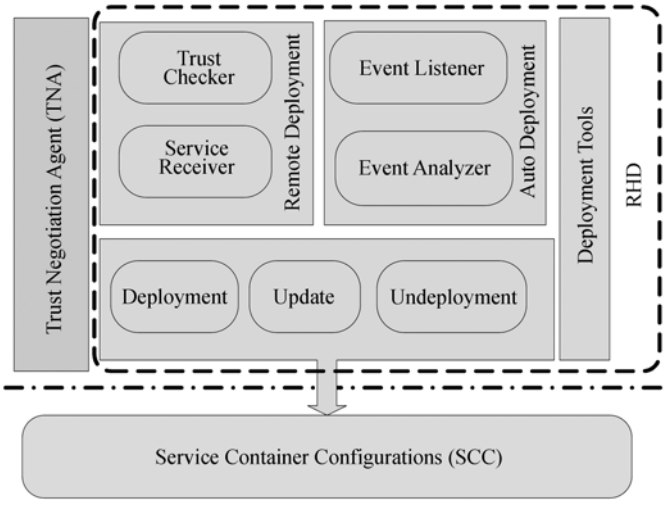

图 9 ROST 功能结构

环境的配置来实现热部署, 避免了服务部署过程中重新启动服务容器, 并基于 CROWN 的自动信任协商机制为服务部署提供安全保障, 图中的信任协商代理 TNA(trust negotiation agent)和远程热部署 RHD(remote and hot deployment)是 ROST 的两个核心组件, 其中 TNA 负责在部署者和服务容器之间建立信任关系(其实现信息详见 4.2 节), 而 RHD 负责远程服务热部署, 服务容器配置 SCC (service container configuration) 是对服务容器各种配置信息的抽象.

(1) 远程服务部署.

在建立相互信任关系之后, 目标服务容器接收并解压 GAR 文件; 然后调用底层部署功能执行相应的操作; 服务容器存储了已部署服务的各种配置信息, 而热部署的关键是动态更新 SCC 配置信息, 包括可执行程序、WSDL 描述信息、WSDD 和 JNDI 配置信息等(如服务用 Java 实现时, 就需要装载服务的 Java 类).

但是, 应谨慎对待已存在的服务执行重部署或反部署, 因为服务可能正在被其他的服务或者用户使用. 如果没有有效的控制机制, 只是简单地重部署或者反部署一个服务, 可能会使用户遭受不可预料的服务中断. 为此, 我们在每一个已部署的服务中设置了引用计数, 初始值为 0, 服务调用开始时, 计数值加 1, 服务调用结束时, 计数值减 1; 当请求重部署或者反部署时, 首先检查服务的引用计数, 仅当引用计数为 0 时, 服务才能被重部署或者反部署.

(2) 本地自动服务部署.

除了远程热部署之外, RHD 组件也提供了将服务方便地部署到本地服务容器的机制, 我们通过指定的文件夹接收 GAR 文件, EventListener 持续侦听该文件夹相关的事件, 主要分析新文件的到来和已存在文件的删除这两类事件, 它捕获了事件 e 后, 交给 EventAnalyzer 进行分析和处理; 基于事件的内容, EventAnalyzer 调用不同的底层部署功能. 因此, 当用户通过向文件夹增加/替换/删除 GAR 文件, 即可相应地实现部署/重部署/反部署操作, 而底层的服务部署过程对用户是完全透明的.

基于 CROWN 上部署的生物计算应用 BLAST, 我们对 ROST 性能进行了测试, 通过 Internet 连接两个安全自治域, 服务部署者位于清华大学, 目标服务容器 CROWN node server 位于北京航空航天大学. 首先, 我们测试了部署的平均响应时间随并发请求数的变化趋势(见图 10), 服务部署的时间开销基本可以接受, 所需的时间随着并发数量的增大而增大, 其中 SOAP 附件方式的远程部署效率略低于 FTP 方式, 因为 SOAP 附件的编解码开销较大; 其次, 我们测试了基于 ROST 的服务负载均衡能力, 图 11 给出了任务执行时间和初始部署的服务数的测量结果, 在不使用 ROST 的情况下, 在初始部署的服务数较多时, 请求被分别调度到不同的服务实例上, 任务可在较短时间内得到执行, 但如果初始部署的服务少, 大量的任务顺序执行, 从而导致任务的整体执行时间迅速增大. 如果使用了

ROST 机制, 在初始部署的服务少时, 系统可利用 ROST 机制向其他服务容器远程部署新的服务实例, 服务请求因此可以分散到多个服务实例处, 因而具有较好的作业执行效率.

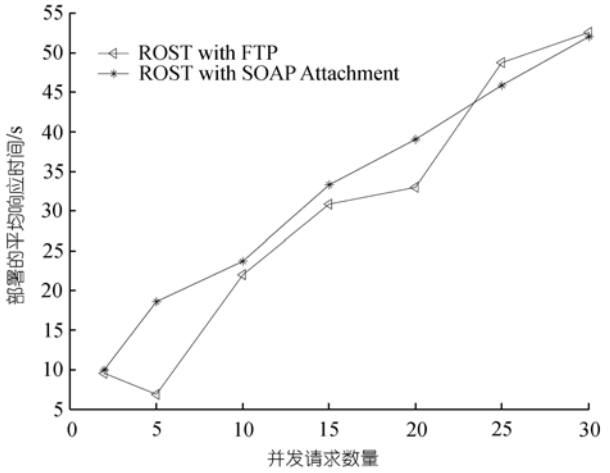

图 10 部署响应时间与并发部署请求数

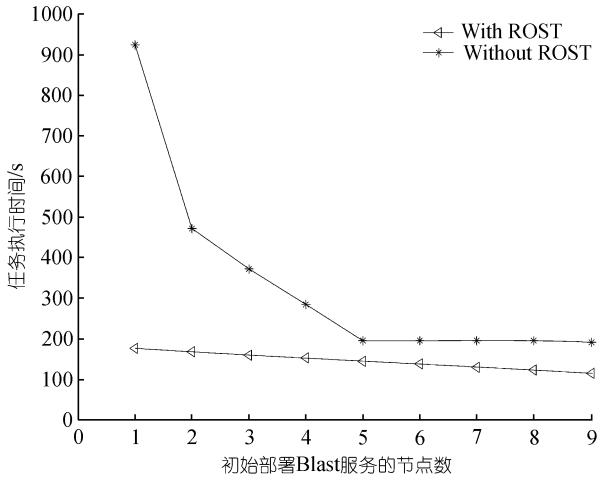

图 11 任务执行时间与初始部署服务数量

通过汇聚广域网络的资源求解应用问题是 CROWN 的目标, 因此, 我们对远程服务热部署具有较大的应用需求, 通过 ROST, 服务可动态部署到位于不同安全自治域的远程服务容器中, 提高了服务的效率和质量.

3.4　信任-激励相容的资源分配机制

如前所述, 为了最大限度地实现资源共享和协同工作, 我们仍面临许多实际问题, 例如, 多数节点希望能更多地使用其他节点的资源, 而不愿共享本地的资源; 每个理性的用户在追求自身利益最大化的同时, 也会严重影响协作整体的运行效率; 此外, 在实际应用中, 大量资源的使用并不是无偿的, 要吸引资源的拥有者加入资源协作, 就必须保证其利益和安全. 因此, 面对多种变化的资源供求关系, 资源的价格、安全因素也是十分重要的. 我们通过分析经济和信任因素, 基于S-Club的资源组织模式, 提出了信任-激励相容的分配机制(trust-incentive compatible resource management)[39], 它使资源分配兼顾投标者的报价和信任度, 激励节点共享有价值的资源, 获取更多的虚拟货币和更高的信任度; 反之, 当协作需求变化导致资源分配不均衡时, 激励机制促进了共享资源的增加, 使得协作重新达到一个新的供需平衡. 此外, 我们还基于经济学中的一般均衡理论, 给出了资源提供者的动态价格调整策略, 以保障网络共享资源的安全与供需均衡.

4　分布式访问控制技术与信任管理

长期以来, 访问控制一直是确保信息系统安全性的主要技术手段之一, 经过数十年的发展, 尽管信息系统结构和规模发生了巨大变化, 但可靠高效的访问控制对于保护服务网格环境仍至关重要, 而基于策略的分布式访问控制技术和信任管理技术是实现这一目标的基础.

作为一种新型的分布式计算模型, 服务网格的分布、异构、自治和动态等特征都对访问控制和信任管理技术提出了不同的挑战. 其分布性体现在用户和资源数量庞大且高度分散, 要求访问控制策略管理和强制机制具有良好的伸缩性, 支持委托授权和单一登录; 其异构性体现在网格资源类型多、结构差异大, 且各自治域可采用不同的安全基础设施, 要求访问控制系统具有足够的开放性和灵活性, 支持各类自治域间身份映射和信任证转换机制; 其自治性体现在网格用户和节点隶属于不同的自治域, 跨域的资源共享和协同需要遵循各方的安全策略, 要求访问控制策略协商机制, 支持策略的协商制定, 支持在保护各方隐私的约束下通过协商方式建立信任关系; 其动态性体现在网格中所有用户和资源聚合的构成、状态、部署位置和安全需求随时而变, 虚拟组织是动态形成和不断变化的, 要求采用独立于应用业务逻辑的方式来规范和强制访问控制策略, 降低策略修改的开销, 适应频繁变化的需求, 以及基于信任管理的分布式授权机制.

根据安全、可信资源共享和协作环境的需求, 下面, 我们讨论 CROWN 的安全体系结构, 包括基本的安全通信、身份鉴别与授权, 以及身份映射与信任证转换技术, 以支持基本的安全功能; 同时, 我们针对网格环境新的安全需求, 重点研究了访问控制策略协商机制、信任管理与信任协商技术.

4.1 CROWN 的安全体系结构

由于网格应用环境可能涉及多个不同的安全自治域, 为有效、一致地支持分布式、跨自治域资源的安全可信管理和共享, 基于资源组织与管理的拓扑结构, 面向异构环境设计了 CROWN 的安全体系结构和解决方案(见图 12), 它包括基于 PKI 和 Kerberos 的安全基础设施、Node Server 消息处理链扩展, 以及一组用于认证、授权和信任管理的安全服务; 同时, 设计了基于策略的分布式资源访问控制机制, 以支持全局统一强身份认证机制、全局一致的授权和访问控制策略及其安全信任域管理等. 其主要特点如下:

(1) 域内资源访问的安全处理: 引入域认证服务和域授权服务, 以支持域内资源的集中认证和授权管理, 所有安全处理过程(包括用户身份鉴别和访问控制)均通过高度灵活的安全策略进行配置, 并允许管理员使用强表达能力的描述语言 XACML 定制细粒度的访问控制策略;

(2) 域内安全基础设施的部署: 在同一自治域内, 部署同构的安全基础设施;

(3) 域间的信任联盟: 在不同自治域间, 允许采用不同的安全基础设施(如 PKI, Kerberos 等), 通过身份映射和信任证转换服务实现域间用户身份的共享, 使 CROWN 用户可以共享和集成更大范围的资源;

(4) 域间的访问控制策略的自动协商: 当服务部署者和 Node Server 位于不同的自治域, 服务的访问策略可以通过双方协商来制定;

(5) 域间的信任管理与协商: 为了支持大规模分布式环境中陌生主体间信任的建立与管理, 采用了基于属性的访问控制和自动信任协商等技术.

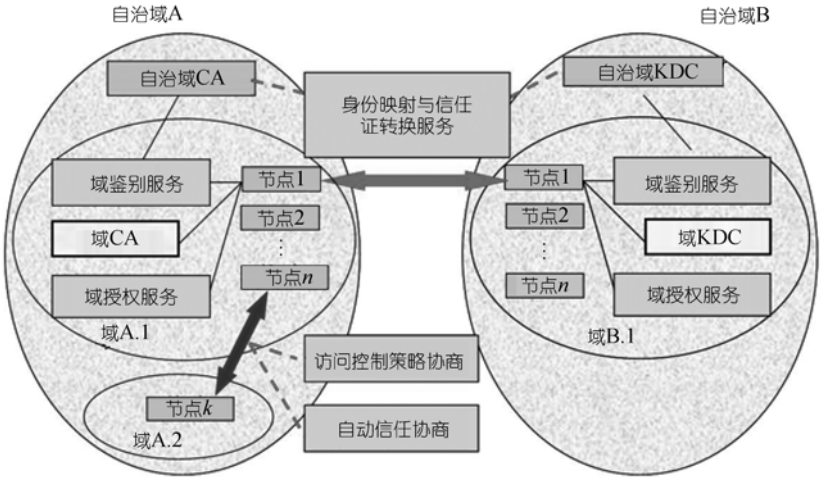

图 12 CROWN 安全体系结构

4.2　访问控制策略的自动协商机制

在服务网格中, 服务部署者和目标服务容器可能位于不同的自治域, 尤其是支持服务运行的物理资源通常由运行服务容器的网格节点提供, 而实现服务的软件资源往往由服务部署者提供, 在提供服务前远程传输并安装在目标网格节点上, 服务部署者和容器方都为服务的实现提供了资源, 因此需要由双方协商制定控制服务访问的策略; 此外, 网格环境所支持的各种协作中出现多方参与的应用场景时(如联合科学研究中产生的实验数据是由参与研究的各方共同拥有), 其访问控制策略应由参与研究的各方协商制定.

在分布式资源的访问控制技术研究中, 已有的网格环境的分布式访问控制系统大多采用委托作为主要的访问控制策略管理方法(如PRIMA[40], GSI[41], CAS[42]以及Akenti[43]等), 这些方法通常假设资源有明确的归属, 即所有授权都有唯一的终极权威源, 没有考虑策略协商的问题; 此外, 一些用于安全策略调和(reconciliation)与协商的方案, 如Patrick McDaniel等人主要是针对群组通信安全策略协商制定问题, 提出的高效的两策略协调算法, 其采用的策略语言(Ismene)无法描述细粒度的访问控制策略; 而且在合成安全策略时采用了保守方法, 其本质是否定优先, 不能动态选择不同的合成方式; Khurana[44]面向动态联合提出了用于刻画访问控制策略协商的状态转移模型, 使访问控制策略协商问题转化为满足各联合成员的目标和一组指定的协商约束的问题, 但其协商方式完全基于RBAC模型, 未提供自动化机制; Bharadwaj等 [45]基于半环的约束逻辑程序, 提出了自治域间访问控制策略自动协商的数学框架, 将多方策略协商过程转化为基于半环的约束满足问题的解空间搜索过程, 但仍处于理论研究探索中, 协商结果的约束缺乏有效性查询验证, 难以支持实用的自动协商方案.

通过分析已有工作和应用需求, 我们提出了一种有效的访问控制策略协商方案, 它基于非递归、分层且带约束的Datalog访问控制策略语言, 采用前向链式(forward chaining)规则的元策略来控制各方的交互行为, 其策略协商过程 [46]如图 13 所示, 在协商过程中, 双方的行为均由元策略所控制; 同时, 该方案具有良好的可扩展性, 可支持更复杂的资源协同场景. 其中元策略采用前向链式规则语言和引擎实现, 主要包括如下三类:

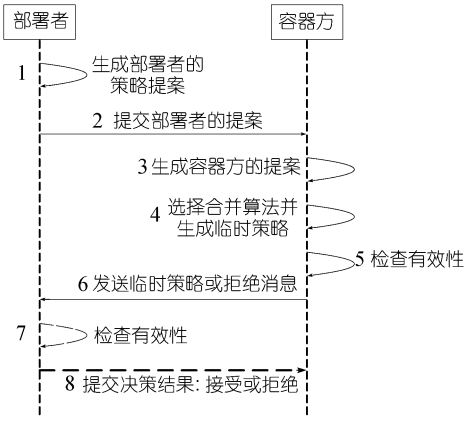

图 13　访问控制策略的协商过程

(1) 提案生成元策略. 主要用于

容器方生成策略提案，基于部署者和容器方间预订的服务等级协定(service level agreement，简称 SLA)或其他协作约定而制定；

(2) 合并算法选择元策略. 实际上，不存在单一的合并算法适用于所有的冲突消解问题，因此需要使用这类元策略动态地选择适当的合并算法，使部署者、被部署服务和原始资源的各种性质都可用于定义这类元策略，例如，可根据部署者的可信度等级选择不同的合并算法，如典型的拒绝优先、许可优先和基于显式优先级等算法；

(3) 有效性验证元策略. 用于检查合并后生成临时策略的有效性评估，根据其逻辑查询有效性接受临时策略，除了对职责分离 [47]的授权约束进行验证外，可通过这类元策略检查临时策略利用SLA或协作规定的服务质量情况.

在 CROWN 环境中，我们部署了访问控制策略协商原型系统，并进行了一系列的应用实验，评价了访问控制策略的自动协商过程.

图 14 和 15 中，我们以并发线程数和临时策略的尺寸作为实验的参数，其中临时策略的尺寸又进一步划分下四个参数，即每个基本授权规则中原语约束数(记作 *PC*)、原语约束中变量数(原语约束的大小，记作 *PCS*)、临时策略中变量数(记作 *VAR*)，以及临时策略中基本授权规则数(记作 *R*).

我们将整个协商过程所用时间作为性能评估的指标，但不包括用户交互和加密/解密所花的时间. 首先，我们使用了四组随机生成的测试用例，每组包含 30 个用相同策略尺寸参数生成的测试用例. 为了便于比较，根据 *PC* 和 *PCS* 取值的不同，我们将这四组测试用例分为两部分(如图 14 和 15 所示)，图中每个点代表了一个测试用例所用的时间，且具有相同尺寸参数的测试用例所使用的协商时间

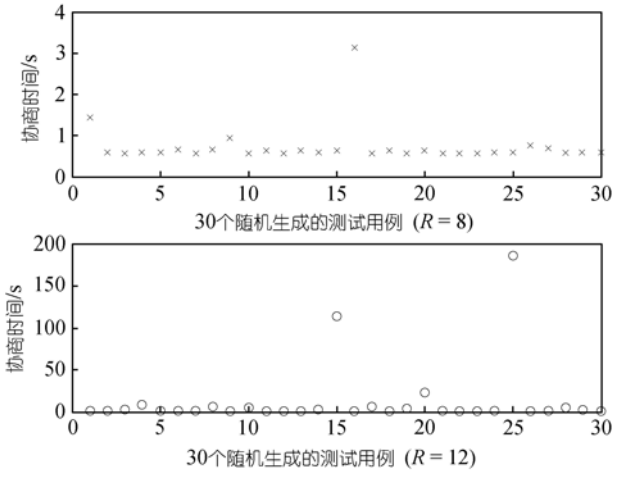

图 14 30 个测试用例的协商时间
(*PC*=3, *PCS*=3, *VAR*=8)

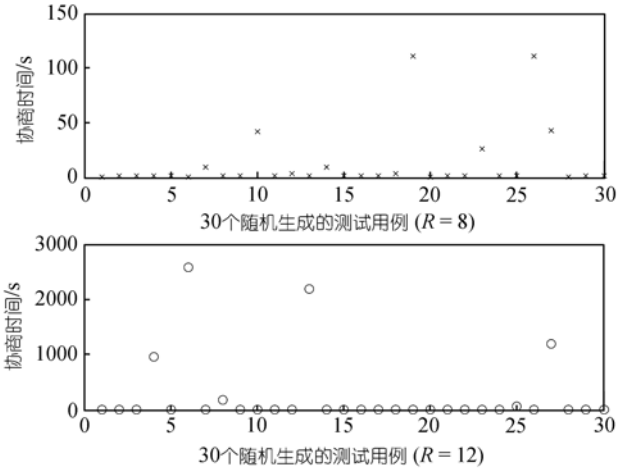

图 15　30 个测试用例的协商时间
(*PC*=5, *PCS*=5, *VAR*=8)

可能差别很大, 因为测试用例的策略和查询结构对协商时间有重要影响, 如查询中约束文字的数量和形式、策略中组合规则形式均会影响协商时间. 最坏情形的复杂度至少是 co-NP-hard, 但从图中也可看出, 实际应用中大多数情形的复杂度并不高.

如果将 30 个测试用例的策略协商时间的峰值视为最坏情形复杂度, 从图 14 和 15 中可以看出, 当 *PC* 和 *PCS* 相对较小, 且规则数目适度时, 测试用例的协商耗时则在可以接受范围内, 因此, 这种策略协商方案具有的实用价值较高.

图 16(*PC*=5, *PCS*=3, *VAR*=8 和 *R*=8)显示了平均协商时间随并发请求数变化

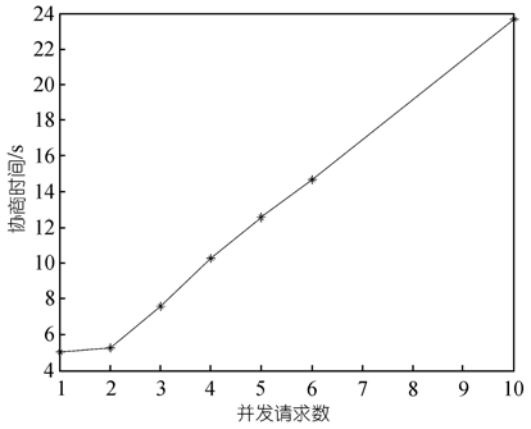

图 16　协商时间与并发请求数的关系

情况，表明了随着并发请求数增加时，平均协商时间基本上线性增加，进一步说明了该策略协商方案的可用性.

4.3 信任管理与信任协商

基于 Internet 的分布式应用对访问控制技术提出了新的要求，尤其是已有的机制大多都基于资源请求者身份进行授权决策，无法适用于广域分布式应用系统. 为解决开放网络环境中无法部署集中身份管理问题，信任管理 [48,49]引入了非集中授权的原则; Winsborough等 [50]和李建欣等 [51]结合信任管理提出了自动信任协商(automated trust negotiation，简称ATN)概念，它通过信任证、访问控制策略的交互披露，资源的请求方和提供方自动地建立信任关系；基于IBM TE(trust establishment)系统 [52]的信任管理语言TPL(trust policy language)和X.509 v3 属性证书，美国UIUC和BYU完成的TrustBuilder[53]是目前唯一可用的自动信任协商中间件，它提供了对敏感X.509 v3 证书的保护，但缺乏灵活的委托授权机制，无法实现跨安全域间基于身份和属性的权限委托，而且基于X.509 v3 证书的表达能力存在局限性，无法描述复杂约束信息；GT4.x遵循了一系列的Web Service安全规范，但未考虑保护敏感信任证和访问控制策略中隐私内容，无法动态地建立基于协商的信任关系.

在 CROWN 中，我们将信任管理和协商技术作为其安全支撑基础实施的一部分，以支持动态建立相互的信任关系(如 3.3 节的 ROST)，信任协商的引擎中主要包含如下 4 个主要功能模块(如图 17 所示):

(1) 策略强制模块：基于协商中对方披露的信息和本地会话信息，调用信任管理引擎提供的信任链构造和一致性验证算法，更新信任图，得到最终可披露的消息集合;

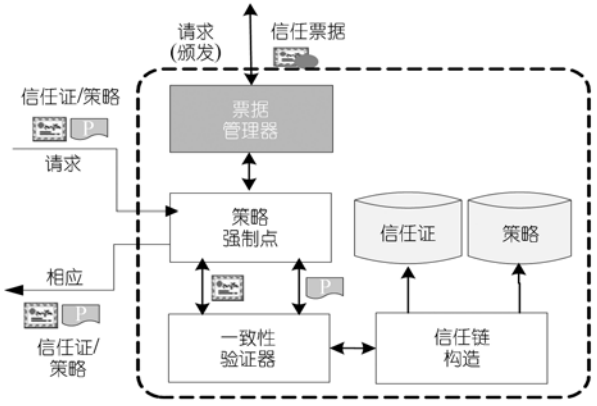

图 17 协商引擎的功能结构

(2) 一致性验证模块: 用于判定信任证集合是否满足访问控制策略的要求, 或者根据访问控制策略生成满足该策略的所有信任证集合;

(3) 信任链构造模块: 从本地或远程的属性权威(attribute authority, 简称 AA)服务中心收集、验证信任证, 并构造信任证链;

(4) 信任票据管理模块: 用于颁发或验证有效时间较短的信任票据, 避免服务请求方多次调用同一目标服务时引起的重复信任协商.

CROWN 的信任协商服务的运行原理如图 18 所示, 通过信任证和访问控制策略的交互披露, 服务请求方和提供方逐渐建立双方的信任关系; 最后在授权决策安全处理链中, 根据协商结果对用户的服务访问进行授权.

当服务请求访问目标服务时, 首先通过握手交互过程双方确定是否能够进行协商; 握手成功后, 遵照信任协商策略双方不断披露自己的信任证和访问控制策略, 直到协商成功(即协商服务判断请求方披露的信任证足以证明其有权访问目标服务), 或者双方拒绝披露进一步的信任证而导致协商失败; 在协商成功后, 服务方授权链节点(AuthzHandler)检查该会话的协商是否成功, 以决定是否授权用户访问目标服务.

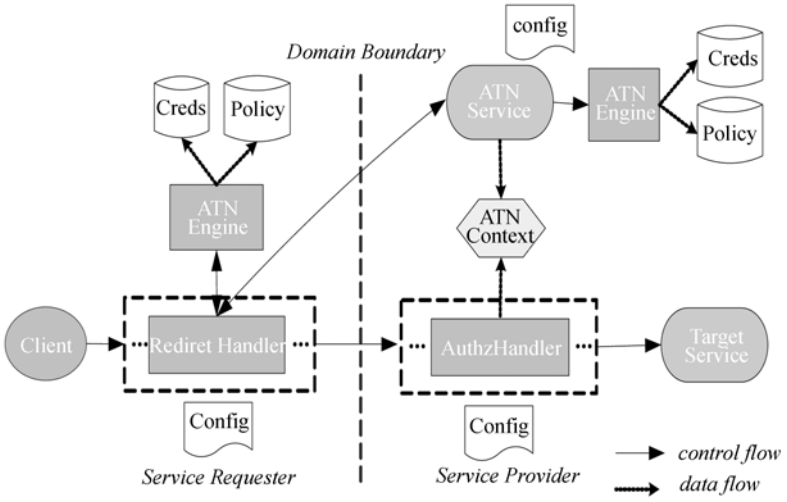

图 18　协商服务在 CROWN 中的运行原理

为提高服务性能, 在服务容器的安全上下文管理器中, 我们通过设立一个队列缓冲已解析的信任证和访问控制策略, 避免特定服务频繁地对引擎初始化; 针对CROWN中信任协商引擎和服务, 我们设计了相关试验分析系统的性能和可用性. 由于协商服务运行时间与具体协商场景的复杂度紧密相关, 为了便于合理比较, 我们给出一个递进的试验示例, 首先选择一个需 3 轮交互的协商实例, 增加委托深度约束后, 扩展生成实例 2, 进一步增加角色参数约束, 得到实例 3, 并为

TrustBuilder v1.0[53]设计了类似于实例 1 的配置；同时，通过多个客户端发起并发请求，测试在无信任票据支持(具有策略缓存机制)、使用信任票据和无策略缓冲机制三种方式下服务的运行总时间.

图 19 中, 3 个实例中协商引擎每轮响应时间大约介于 15 ~ 40 ms 之间，由实验结果可知: 第 1 是协商过程各轮的响应时间变化较小，表明引擎处理性能的稳定性；第 2，在实例中增加各类约束信息后，虽然处理时间会增加，但增长幅度是可接受的；第 3，与 TrustBuilder 相比，我们的引擎响应时间较短，因为 TrustBuilder 存在许多冗余消息验证，例如需验证 X.509v3 证书中的所有字段.

图 20 中，在系统的三种配置方式下，当有 25 个并发请求时，服务的运行总

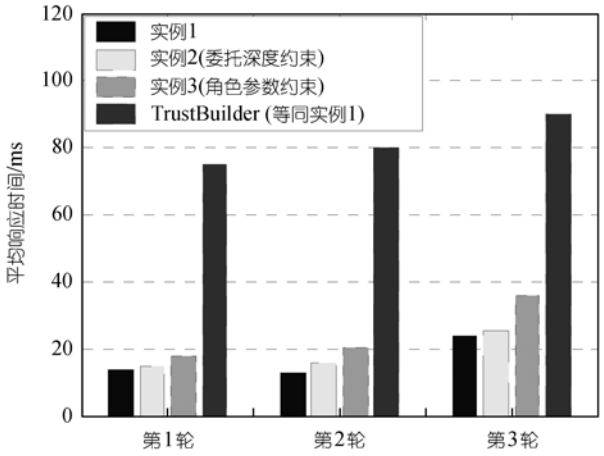

图 19 信任引擎在每轮平均响应时间

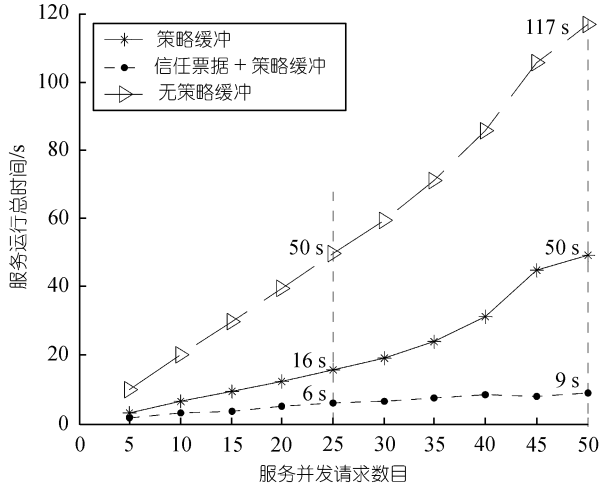

图 20 协商服务对并发请求处理性能

时间大约分别为 6, 16 和 50 s; 而有 50 个并发请求时, 大约为 9, 50 和 117 s. 通过该组实验同样可得到3个结论: (1) 在不同配置下, 协商服务运行的总时间和并发客户端数目大致成线性关系, 表明协商服务性能的稳定性; (2) 协商服务的信任票据机制能够有效降低服务负载, 从图中可以看到总的运行时间增长比较平缓; (3) 协商服务的策略缓冲机制非常有效, 能避免每次由于配置文件的读取和解析给协商服务带来的额外负载, 这也提示我们在管理服务容器时, 需要限定频繁被调用服务的数量, 避免策略缓冲队列内信息的频繁更替.

5　CROWN 试验平台与应用

为了验证上述技术和 CROWN 中间件的有效性, 建立基于网络的多学科科研活动环境, 我们联合清华大学、北京大学、中国科学院计算机网络信息中心、中国科学院大气物理研究所、中国科学院自动化研究所、国防科学技术大学、香港科技大学以及英国利兹(Leeds)大学等单位共同构建了 CROWN 试验网络, 并开展了多个领域的应用示范. 截至 2005 年 8 月底, CROWN 试验网络已经覆盖了分布在 5 个城市的 11 个科研机构, 形成一个由 40 余台高性能服务器或集群系统组成的核心物理资源集合, 并通过中间件系统在逻辑上组织成 5 个区域, 16 个自治域的拓扑结构. 在所建立的 CROWN 广域试验环境中, 我们充分体现了网格的动态、异构、分布、自治等特性, 使得我们可以在极大程度上模拟和观测真实环境中网格应用的运行情况, 验证相关技术和算法. 随着研究的不断开展, 试验环境将在规模上不断扩大, 更加逼近真实应用环境, 并逐渐向生产、研究并重的广域分布网格环境过渡.

同时, 基于 CROWN 试验网络, 我们部署了中尺度天气系统降水预报(AREM)、海量多媒体数据处理平台(MDP)、血液流动温度场显示(gViz)、科学数据网格(SDG), 以及数字巡天图检索等多个网格应用, 对多种需求给出了应用验证.

AREM 将网格应用于数值天气预报模型的研究. 气象学科研人员和天气预报业务工作者在开展生产研究工作的同时, 总结出若干数值模型, 这些模型利用国家气象中心转发的原始气象资料进行处理, 根据大气及流体物理规律进行演化模拟, 从而推断未来时刻的天气情况. 这些模拟以复杂的数值计算为基础, 需要大量计算机处理周期及数据存储空间. 通过和中国科学院大气物理研究所合作, 我们将 Fortran 编译器、可视化后处理软件 GrADS 以及 AREM 数值模式进行服务化封装, 并部署了统一的原始数据仓库, 利用 CROWN 所提供的服务化资源管理, 网格作业调度机制以及试验床集成的计算和存储能力, 实现了中尺度暴雨数值预报模式的研究系统. 目前, 气象学科研人员可通过 CROWN Portal 提交数值模拟运算作业, 并根据模拟结果不断改进和优化数值模型, 简化了相关研究的

过程, 缩短了数值模型运算的时间和改进的周期, 提高了科研效率.

在以语音、视频内容识别为代表的多媒体数据处理应用中, 需要大量的存储资源和计算能力. 传统处理过程主要采用集中处理的方式, 将分散采集的信息收集并统一进行处理. 当数据来源增加时, 集中处理的方式难以提供良好的可扩展性, 特别是在对处理实时性具有一定要求的场景下, 依赖于少量计算资源的能力往往不能满足处理需要. 我们将服务网格技术与海量数据处理技术相结合, 针对多媒体信息处理中计算量大, 存储要求高, 并行处理需求突出的特点, 与中国科学院自动化所共同研制了基于 CROWN 的海量多媒体信息处理平台 MDP, 并已在 CROWN 试验床中部署运行. 通过将相关算法封装为服务, 部署在多个网格节点上. 用户可以将多路的多媒体数据输入与网格作业调度结合起来, 通过分析网格节点负载, 发现可用资源并进行调度, 完成处理过程; 同时, 由于在广域网络中进行服务部署, 我们将信任管理与协商技术应用于该平台, 保障了用户数据和处理过程的可信性. MDP 有效增加了多媒体数据处理应用可用资源的数量, 提高了信息处理的准确程度, 提高了整个系统的可靠性和处理过程的实时性.

此外, CROWN 还提供良好的开放性和可集成性, 可通过标准协议规范与其他网格中间件和试验床进行互联互通互操作. 例如, 我们在 CROWN 和英国国家网格(NGS)主要节点(WRG)上同时部署了血液流动温度场可视化(gViz)应用, 实现了远程、异构和独立自治的两个网格系统的资源共享与协同处理, 并在 2005 年英国网格大会(AHM2005)上作为重点介绍和演示.

上述应用开发和部署经验表明, 通过利用 CROWN 提供的资源管理、分布式访问控制和信任协商等技术, 平台能够很好地支撑以密集计算、海量信息处理为特征的典型应用. 截至 2006 年 4 月, CROWN 平台已经成功部署 11 种应用, 接受访问 25000 余次, 有效的支持了广域网络为基础的科研活动. CROWN 中间件的初步应用经验和研究实验表明: 该实验床可以有效地支撑以计算密集型、数据密集型和海量信息分析与处理为特征的典型应用领域.

6 结束语

本文重点讨论了基于服务网格体系结构及中间件系统 CROWN, 针对大量网格资源的分布动态、异构和可信等特点, 提出了基于层叠网的分布网格资源组织与管理机制; 针对网格资源的自治性特点, 提出并实现了支持访问控制策略的自动协商、信任管理和协商机制; 并通过部署 CROWN 中间件系统, 建立了广域试验环境, 开展了中尺度天气系统降水预报、海量多媒体数据处理平台、血液流动温度场显示、科学数据网格以及数字巡天图检索等多个实际网格应用. 研究和应用结果表明: CROWN 中间件可以有效地支撑计算密集型、数据密集型和海量信息分析与处理等为特征的应用领域.

为了建立可信的虚拟服务计算环境, 目前, 我们主要围绕可协同、可管理和可信任这三个基本问题, 正在进一步研究基于层叠网和契约机制的资源动态组织管理与分配模型、服务质量管理, 以及基于协议计算的跨域信任管理与访问控制机制; 尤其是为了有效地支撑应用快速开发, 我们结合 CROWN 的优化和应用部署, 我们正在重点研发基于服务组合的可信软件开发技术与工具, 进一步扩大应用范围和领域, 以深化和发展我们的研究应用.

参　考　文　献

1 Foster I, Kesselman C. The Grid: Blueprint for a New Computing Infrastructure. 2nd ed. San Fransisco: Morgan Kaufmann, 2004. 279—310

2 Daniel A R. The TeraGrid: Cyber infrastructure for 21st Century Science and Engineering. Arlington: National Science Foundation, 2001. http://www.teragrid.org/

3 Vision for the DOE Science Grid. http://doesciencegrid.org

4 Computer Challenges to Emerge from eScience. http://www.escience-grid.org.uk

5 Foster I. The Physiology of the Grid —— An Open Grid Service Architecture for Distributed Systems Integration. Open Grid Service Infrastructure WG, Global Grid Forum. 2002

6 Web Service Resource Framework (Version 1.0). http://www.globus.org/wsrf/specs/ ws-wsrf.pdf

7 舒剑, 胡春明, 葛声, 等. Web Service 运行管理平台的研究与实现. 计算机研究与发展, 2004, 41(3): 442—450

8 胡春明, 怀进鹏, 孙海龙, 基于 Web 服务的网格体系结构与支撑系统研究. 软件学报, 2004, 15(7): 1064—1073

9 Hu C, Huai J, Zhu Y, et al. Efficient information service management using service club in CROWN Grid. In: Proceedings of 2005 IEEE International Conference on Service Computing (SCC 2005). Washington DC:IEEE Computer Society, 2005. 5—12

10 Sun H, Zhu Y, Hu C, et al. Early experience of remote and hot service deployment with trustworthiness in CROWN Grid. In: Proceedings of 6th International Workshop on Advanced Parallel Processing Technologies (APPT 2005). Belin: Springer, 2005. 301—312

11 Foster I, Kesselman C, Tuecke S. The anatomy of the Grid: Enabling scalable virtual organization. IJHPCA, 2001, 15(3): 200—222

12 Allen G, Benger W, Hege C, et al. Solving Einstein's equations on supercomputers. IEEE Computer Magzine, 1999, 32(12): 52—58

13 Casanova H, Obertelli G, Berman F, et al. The AppLeS parameter sweep template: User-level middleware for the Grid. In: Proceedings of Supercomputing 2000. Washington DC: IEEE Computer Society, 2000

14 Abramson D, Sosic R, Giddy J, et al. Nimrod: A tool for performing parameterized simulations using distributed workstations. In: Proceedings of 4th IEEE Symposium on High Performance Distributed Computing. Washington DC:IEEE Computer Society, 1995

15 Nakada H, Sato M, Sekiguchi S. Design and implementations of Ninf: Toward a global computing infrastructure. Future Generation Computing Systems, 1999, 15: 649—658[DOI]

16 Litzkow M, Livny M, Mutka M. Condor——A hunter of idle workstations. In: Proceedings of 8th IEEE International Conference on Distributed Computing Systems, 1998. Washington DC: IEEE Computer Society, 1998. 104—111

17 Foster I, Kesselman C. Globus: A metacomputing infrastructure Toolkit. IJSA, 1997, 11(2): 115—129

18 Chapin S, Katramatos D, Karpovich J, et al. The Legion Resource Management System. In: Proceedings of the 5th Workshop on Job Scheduling Strategies for Parallel Processing (JSSPP'99), in conjunction with the

Int'l Parallel and Distributed Proceesing Symposium (IPDPS'99). Washington DC:IEEE Computer Society, 1999. 162—178

19 Gokhale A S, Natarajan B. GriT: A CORBA-based Grid middleware architecture. In:Proceedings of the 36th Hawaii Int'l Conference on System Sciences (HICSS'03). Washington DC :IEEE Computer Society, 2002. 319—322

20 Furmento N, Lee W, Mayer A, et al. ICENI: An open Grid service architecture implemented with Jini. Parallel Computing (PARCO), 2002, 28(12): 1753—1772[DOI]

21 Antonioletti M, Atkinson M P, Baxter R, et al. The design and implementation of Grid database services in OGSA-DAI. Concurr Comput: Prac Exp, 2005, 17(2-4): 357—376[DOI]

22 Pearlman L, Welch V, Foster I, et al. A community authorization service for group collaboration. In: Proceedings of the 3rd IEEE International Workshop on Policies for Distributed Systems and Networks. Los Alamitos: IEEE Computer Society Press, 2002

23 Li Z, Mohapatra P. The impact of topology on overlay routing service. In: Proceedings of IEEE INFOCOM 2004. Washington DC:IEEE Computer Society, 2004

24 Andersen D G, Balakrishnan H, Kaashoek M, et al. Resilient overlay networks. In: Proceedings of 18th ACM Symposium on Operating Systems Principles (SOSP'01), Banff, Canada. New York:ACM Press, 2001. 131—145

25 Duan Z, Zhang Z L, Hou Y T. Service overlay networks: SLAs, QoS and bandwidth provisioning. In: Proceedings of 10th IEEE International Conference on Network Protocol (ICNP'02). Los Alamitos: IEEE Computer Society Press, 2002

26 Subramanian L, Stoica I, Balakrishnan H, et al. OverQoS: An overlay based architecture for enhancing internet QoS. In: Proceedings of USENIX 1st Symposium on Networked System Design and Implementation (NSDI 2004). San Francisco:USENIX Press, 2004. 71—84

27 Albert R, Barabasi A L. Statistical mechanics of complex network. Rev Mod Phys, 2002, (74): 47—97

28 Dorogovtsec S N, Mendes J F F. Evaluation of Network. Adv Phys, 2002, 51:1079—1187[DOI]

29 Strogatz S H. Exploring complex networks. Nature, 2001, 410(6825): 268—276[DOI]

30 Abdel-Wahab H, Stoica I, Sultan F, et al. A simple and fast distributed algorithm to compute a minimum spanning tree in the internet. In:Proceedings of Joint Conference on Information Science '95.Washington DC:IEEE Computer Society, 1995. 429—433

31 Chu Y, Rao S G, Seshan S, et al. A case for end system multicast. In:Proceedings of ACM Sigmetrics 2000. New York:ACM Press, 2000

32 Czajkowski K, Fitzgerald S, Foster I, et al. Grid information services for distributed resource sharing. In: Proceedings of 10th IEEE Int'l Symposium on High Performance Distributed Computing (2001). Washington DC: IEEE Computer Society, 2001. 181—184

33 Universal Description, Discovery and Integration of Web Services (UDDI) Version 2.0. http://www.uddi.org. OASIS. 2002

34 Shaikh-Ali A, Rana O, Al-Ali R, et al. UDDIe: An extended registry for Web service. In: Proceedings of Workshop on Service Oriented Computing Models, Architectures and Applications. Washington DC: IEEE Computer Society, 2003

35 DataGrid Information and Monitoring Services Architecture: Design, Requirements and Evaluation Crieteria. Technical Report, Data Grid Project. http://hepunx.rl.ac.uk/ egee/jra1-uk/glite/doc/java.pdf. 2002

36 Raman R, Livny M, Solomon M. Matchmaking: Distributed resource management for high throughput computing. In: Proceedings of IEEE International Symposium on High Performance Distributed Computing (HPDC-7). Washington DC: IEEE Computer Society, 1998

37 Hong W, Lim M, Kim E, et al. GAIS: Grid advanced information service based on P2P mechanism. In: Proceedings of the 13th IEEE International Symposium on High Performance Distributed Computing

(HPDC-13). Washington DC: IEEE Computer Society, 2004. 276－277
38 董方鹏, 龚奕利, 李伟, 等. 网格环境中资源发现机制研究. 计算机研究与发展, 2003, 40(12): 1749－1755
39 Huai J, Zhang Y, Li X, et al. Distributed access control in CROWN groups. In: Proceeding of the 34th International Conference on Parallel Processing (ICPP 2005). Washington DC: IEEE Computer Society, 2005
40 Lorch M, Adams D, Kafura D, et al. The PRIMA system for privilege management, authorization and enforcement in Grid environments. In: Proceedings of The 4th International Workshop on Grid Computing (Grid 2003). Los Alamitos: IEEE Computer Society, 2003
41 Foster I, Kesselman C, Tsudik G, et al. A security architecture for computational Grids. In: Proceedings of the 5th ACM Conference on Computer and Communications Security. New York: ACM Press, 1998. 83-92
42 Pearlman L, Welch V, Foster I, et al. A community authorization service for group collaboration. In: Proceedings of IEEE 3rd International Workshop on Policies for Distributed Systems and Networks. Washington DC: IEEE Computer Society, 2002. 50－59
43 Thompson M R, Mudumbai S. Certificate-based authorization policy in a PKI environment. ACM Transactions on Information and System Security (TISSEC), 2003, 6(4): 566－588[DOI]
44 Khurana H. Negotiation and management of coalition resources. PhD thesis. University of Maryland, 2002
45 Bharadwaj V G, Baras J S. Towards automated negotiation of access controls policies. In:Proceedings of the 4th IEEE International Workshop on Policies for Distributed Systems and Networks. Washington DC: IEEE Computer Society, 2003. 111－119
46 Xue W, Huai J, Liu Y. Access control policy negotiation for remote hot-deployed grid services. In: the first International Conference on e-Science and Grid Computing, (eScience2005). Melbourne: IEEE Computer Society, 2005
47 Clark D D, Wilson D R. A comparison of commercial and military computer security policies. In: Proceedings of 1987 IEEE Symposium on Security and Privacy. Los Alamitos: IEEE Computer Society, 1987. 184－195
48 Blaze M, Feigenbaum J, Lacy J. Decentralized trust management. In: IEEE Symposium on Security and Privacy. Los Alamitos:IEEE Computer Society, 1996
49 徐锋, 吕建. Web 安全中的信任管理研究与进展. 软件学报, 2002, 11(13): 2057－2064
50 Winsborough W H, Seamons K E, Jones V E. Automated trust negotiation. In: DARPA Information Survivability Conference and Exposition, 2000
51 李建欣, 怀进鹏, 李先贤. 自动信任协商研究.软件学报, 2006, 17
52 Herzberg A, Mass Y, Michaeli J, et al. Access control meets public key infrastructure, Or: assigning roles to strangers. In: IEEE Symposium on Security and Privacy (S&P 2000). Los Alamitos:IEEE Computer Society, 2000
53 Winslett M, Yu T, Seamons K E, et al. The trustbuilder architecture for trust negotiation. IEEE Internet Comput, 2002, 6(6): 30－37[DOI]