

广域网虚拟平台:网格计算和对等计算的融合^{*})

吴帆 叶懋 陈贵海

(南京大学软件新技术国家重点实验室 南京 210093)

摘要 近几年,分布式计算领域出现了两个研究热点:网格计算(Grid Computing)和对等计算(Peer to Peer Computing)。网格计算和对等计算以不同的方法组织大规模分布式的资源,包括计算能力、存储资源和带宽等。两者的研究领域有一定的重叠,又有很好的互补性,融合将是必然趋势。我们构想了一个广域网虚拟平台,全世界的计算机连接成一个整体,任何人随时随地都可以得到所需的资源和服务。正是这一理想,驱动着网格计算和对等计算技术的快速进步。

关键词 广域网虚拟平台,网格计算,对等计算

Internet Virtual Platform: Convergence of P2P and Grid Computing

WU Fan YE Mao CHEN Gui-Hai

(State Key Laboratory of Novel Software Technology, Nanjing University, Nanjing 210093)

Abstract Two supposedly new approaches to distributed computing have emerged in the past few years, both claiming to address the problem of organizing large scale and distributed resources including computational ability, storage and network bandwidth etc: peer-to-peer computing (P2P) and Grid computing. The P2P and Grid infrastructure communities are tackling an overlapping set of problems. Meanwhile the strengths and weaknesses of the two approaches suggest that the interests of the two communities are likely to grow closer over time. A vision of Internet Virtual Platform that a worldwide computer within which access to resources and services can be negotiated as and when needed motivates both Grid and P2P computing.

Keywords Internet virtual platform, Grid computing, P2P computing

1 引言

随着互联网和网络计算技术的迅猛发展,分布式计算领域出现了两个研究热点:网格计算(Grid Computing)和对等计算(Peer to Peer Computing)。两者从不同的角度提出分布式网络环境中大规模资源的收集和协同的解决方案,但具有共同的目标,即构建一个统一的广域网虚拟平台。

网格计算和对等计算联系紧密,主要体现在以下几点^[12]: (1) 研究的问题都是虚拟社区中共享资源的组织。(2) 采用的方法都是构建覆盖网络。(3) 都取得了实质性的进展,但仍然存在局限性, Ian Foster 这样概括目前的发展情况:“网格计算建立了体系结构,但没有解决结点失效(node failure)问题;对等计算解决了失效问题,但没有建立体系结构”。(4) 其互补性使两者的融合成为必然的趋势。

现在,对等计算与网格计算领域处于一种平行发展的局面。但这种平行状态并不是脑力与金钱的浪费。没有对等计算领域的有力支持,网格计算就不可能拥有完备牢靠的体系结构,就不可能具有如此多的重要性质,如检索、存储可扩展性、分布性、匿名性等等。

本文第2部分概要说明广域网虚拟平台的设计目标。第3部分阐述网格计算和对等计算的定义与体系结构,以及为实现虚拟平台的最终目标两者共同面临的问题。第4部分对现有的系统进行分析分类。最后,总结并勾勒未来的发展。

2 设计目标

我们的最终目标是建立一个全球化的广域网虚拟平台,

通过互联网加快分布式计算的速度,以获得更高的商业利益和提供更有效的科学研究手段。广域网虚拟平台可看作一个普适计算的基础设施,它将以前小规模的网络拓展成全球化的网格。该虚拟计算平台应满足以下几点基本要求:

可靠性: 虚拟平台提供的计算能力必须保证是持续、稳定和安全的,不应该因为虚拟平台内部个别资源的变化而对其应用造成影响,即虚拟平台内部局部资源的变动对其应用应该是透明的。虚拟计算平台还应该满足各种形式的安全要求,比如数据传输加密,权限的认证,避免非法入侵和非法使用等。

标准化: 一方面,虚拟计算平台资源之间应该有一个统一的可以互相访问的接口或者协议标准,因为只有这样才能实现虚拟平台资源的互操作从而实现充分的资源共享,标准化是共享的前提;另一方面,虚拟平台对用户提供的计算能力应该满足一定的标准,有一种比较统一的形式,从而便于以一种统一的方式进行访问,对于访问者来说,不能因为时间、地点、具体的访问系统等的不同而要求不断改变访问形式,访问形式应该具有一致性。

易访问性: 用户应该可以在任何时间、任何地点、以自己习惯的统一的形式访问和使用各种网络资源。在虚拟平台上没有资源处在什么位置的概念,无论你在何地,资源都在你的身边。人们以前在解决特定问题时或许不得不到特定的地点去进行,但在虚拟平台上解决问题时,不应该因为访问者或者资源所在地位置不同而受到限制。

廉价性: 这是虚拟计算平台被普遍接受和推广的前提。集中式系统的服务器有较高的性能需求,自然费用高昂。虚拟计

^{*}) 本课题得到国家 973 项目(No. 2002CB312002)和教育部高校青年教师奖资助。

算平台通过将资源充分共享,最大限度发挥资源的使用价值,可以将原来闲置的资源收集起来供平台用户使用,而且可以避免以前由于地理位置限制,所带来诸如搬运费、差旅费等各种额外的花费。

3 网格计算与对等计算

网格计算和对等计算具有共同的目标,即构建一个统一的广域网虚拟平台。两者从不同的角度提出分布式网络环境中大规模资源的收集和协同的解决方案。

3.1 网格计算(Grid Computing)

3.1.1 网格定义与目标 网格来源于大规模计算。Ian Foster 和 Carl Kesselman 在文[14]中这样描述网格:“网格是构筑在互联网上的一组新兴技术,它将高速互联网、高性能计算机、大型数据库、传感器、远程设备等融为一体,为科技人员和普通老百姓提供更多的资源、功能和交互性。互联网主要为人们提供电子邮件、网页浏览等通信功能,而网格功能则更多更强,让人们透明地使用计算、存储等其他资源。”2000年, Ian Foster 等在文[15]中将其进一步描述为“在动态变化的多个虚拟机构间共享资源和协同解决问题。”2002年7月, Ian Foster 在文[16]中,限定网格必须同时满足三个条件:(1)在分布式的网络环境中协调资源的使用;(2)使用标准的、开放的并且通用的协议和接口;(3)提供多样化的服务质量。

网格应该是自动配置(self-configuring)、自动调整(self-tuning)和自动恢复(self-healing)的,其目标是充分利用 Internet 中从中心到边缘的共享计算能力、存储空间、信息、软件等资源。

3.1.2 网格体系结构 网格现在采用五层沙漏结构。五层沙漏结构是一种影响十分广泛的结构,其中一个最重要的思想就是以“协议”为中心,也十分强调服务、API 和 SDK 的重要性。

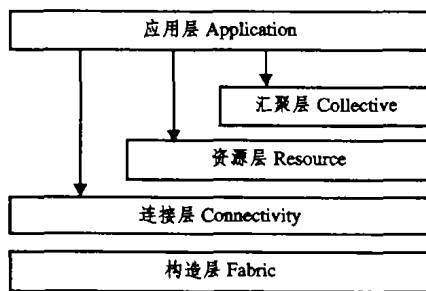


图1 网格五层协议体系结构

在五层沙漏结构中,最底层是构造层(Fabric),构造层面面对的是一个具体的物理(也可以是逻辑的)资源,它通过对这些局部资源的管理,向上层提供对这些资源的管理和控制接口。构造层的上面是连接层(Connectivity),主要是为下层的物理资源提供安全的数据通信能力,这是资源之间进行互操作的前提,连接层使得孤立的单个资源之间建立了联系。连接层的上面是资源层(Resource),它反映的是抽象的局部资源的特征,而资源层上面的汇聚层(Collective)完成的功能是如何将下面以单个资源形式表现出来的资源集中起来,协调解决多个资源之间的问题。最上面的应用层(Application)和资源的距离最远,它关心的是有什么样的资源可以由下面提供给虚拟组织,解决不同虚拟组织的具体问题。如图1所示^[15]。

开放网格服务结构 OGSA(Open Grid Services Architec-

ture)^[13]是 Global Grid Forum 的重要标准协议,是继五层沙漏结构之后最重要,也是目前最新的一种网络体系结构,被称为下一代的网格。

OGSA 定义了“网格服务”的概念。这里,服务指具有特定功能的网络化实体。在 OGSA 中,服务的概念更广,包括各种计算资源、存储资源、网络、程序、数据库等等,总之,一切都是服务。五层模型实现的是对资源的共享,而在 OGSA 中,实现的将是对服务的共享。从资源到服务,这种抽象,将资源、信息、数据等统一起来,十分有利于灵活的、一致的、动态的共享机制的实现,使得分布式系统管理有了标准的接口和行为。网格服务提供了一组接口,这些接口的定义明确并且遵守特定的惯例,解决服务发现、动态服务创建、生命周期管理、通知等问题。在 OGSA 中,网格服务可以以不同的方式聚集起来满足虚拟组织的需求,虚拟组织自身也可以部分地根据他们操作和共享的服务来定义。简言之,网格服务=接口/操作+服务数据。

以网格服务为中心的模型具有如下好处^[13]:1)由于网格环境中所有的组件都是虚拟的(这里的具体含义是指对相同接口不同实现的封装),因此,通过提供一组相对统一的核心接口,所有的网格服务都基于这些接口实现,就可以很容易地构造出具有层次结构的、更高级别的服务,这些服务可以跨越不同的抽象层次,以一种统一的方式来看待。2)虚拟化也使得将多个逻辑资源实例映射到相同的物理资源上成为可能,在对服务进行组合时不必考虑具体的实现,可以以底层资源组成为基础,在虚拟组织中进行资源管理。通过网格服务的虚拟化,可以将通用的服务语义和行为,无缝地映射到本地平台的基础设施之上。

3.2 对等计算(P2P Computing)

3.2.1 对等计算定义与目标 Ion Stoica 等人^[33]给出了对等计算的严格定义:“P2P 系统和应用是一种分布式的系统,它没有集中式的控制,或者分层组织结构,软件在任何节点上运行都能发挥相同的功能。”

P2P 研究的重点与 Grid 不尽相同,P2P 着重于系统的分布性(decentralization)、稳定性(stability)和容错性(fault tolerance)。这些方面,虽然非 Grid 所侧重,但随着系统规模的扩大,将越来越不容忽视。P2P 的目标是利用 Internet 边缘的共享存储空间、空闲处理器周期等资源。

3.2.2 对等计算体系结构 在 Dejan S. Milojicic 等人的技术报告^[29]中,提出了一个非标准化的 P2P 体系结构,如图2所示。在这一节中,我们讨论各构件的功能及其实现过程中遇到问题的解决办法。

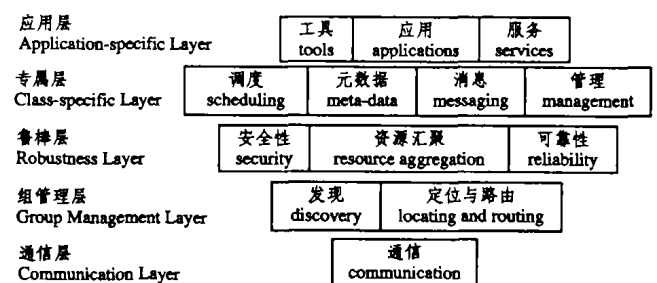


图2 P2P 系统体系结构

通信层(Communication Layer):P2P 模型以网络通信为基础,覆盖了广大的通信范畴。在 P2P 社区中通信,最根本的问题就是要解决端节点的动态性。系统应该能够很好地解决

某个节点或一组节点有意(用户关闭计算机)或无意(网络连接意外中断)地离开网络。如何在动态的环境中,为应用层提供可靠的连通性是 P2P 系统的开发者面临的挑战之一。

组管理层(Group Management Layer):组管理包括在虚拟组织中节点之间如何发现、定位和路由。如何发现未知节点方面,Napster[www.napster.com]采用高度集中的方式,Gnutella^[20]采用完全分布式的方式,还有一些系统采用折中的方式。

定位与路由算法是为了优化,或者说最短化节点间消息传递的路径。其基本目标是,尽量减小到达目标节点所经过的跳数,尽量缩小节点上路由表的大小。在这方面,基于 Hash 的路由方式被广泛采用,当前涌现的路由模型主要有 6 个:Chord^[33]、CAN^[30]、Tapestry^[38]、Pastry^[31]、Viceroy^[28]和 Cycloid^[18]。

鲁棒层(Robustness Layer):P2P 系统的鲁棒性需要有三个关键的构件来支持:安全性、资源汇聚和可靠性。安全性是实现 P2P 系统的又一挑战。P2P 的优势在于,它允许节点扮演客户和服务器双重角色。但是,从客户端向服务器的转化会给整个系统带来一系列的问题。一般来讲,只有可靠的或授权的用户才能够获取信息和服务。不幸的是,安全机制不是要求用户完成繁琐的认证程序,就是需要一个可靠的第三方提供证书。所以,这其中存在着一个矛盾,是保持 P2P 系统分布式的特点,还是采纳集中式的安全认证机制。目前的 P2P 系统不但采用了以前分布式系统成熟的安全机制,如信任链、会话密钥、数字摘要和数字签名,还提出了许多适用于 P2P 系统的新的安全机制,包括多重密钥加密^[7]、沙箱^[3,37]、数字签名^[24]、权限与义务挂钩、防火墙等。

资源汇聚是 P2P 模型提供的一个基本功能。从各个端节点汇聚来的资源种类繁多,一方面,资源包括计算机中存储的文件和其它信息;另一方面,端节点上所有可用的 CPU 计算能力、带宽、磁盘空间等都可被作为资源。

P2P 系统的可靠性是一个重要的问题。P2P 系统天然的动态性使得我们很难保证行为的可靠性。通常的解决办法,是利用冗余信息来提高可靠性。例如,在计算密集型应用中,当发现失败时,计算任务可由另一个可用节点重新执行。或者,同一个任务同时由多个节点各自独立地执行;在文件共享应用中,通过让一些节点保存文件的副本来提高可靠性;在通信型应用中,建立消息确认-重传机制或者从多条路径同时传递同一消息,都可提高可靠性;在有拓扑结构的 P2P 系统中,可靠性还应该考虑在结点的不断加入与离开情况下,如何动态维护覆盖网络的拓扑结构。

专属层(Class-specific Layer):以上讨论的构件适用于所有的 P2P 系统结构,而专属层构件是对各类 P2P 应用的功能抽象。调度应用于可并行执行的计算密集型应用,计算密集型任务被分割成小块,每一块由调度组件分配到可用节点执行。元数据适用于内容和文件管理型应用,元数据被用来描述节点提供的信息,并为资源定位提供有力的参考。消息构件应用于协作型应用。管理构件用来支持对 P2P 底层设施的管理。

应用层(Application-specific Layer):这一层包括由开发工具、应用程序和服务实现的运行于 P2P 覆盖层之上的各种具体应用,例如分布式计算(用在科学研究、金融业等),文件共享,及通信与协作系统。

3.3 共同面临的问题

3.3.1 体系结构(Formation) 拓扑结构和节点发现策

略常位于覆盖网络(overlay network)中,主要解决节点如何加入已知系统并获取其邻居信息的问题。Ian Foster 和 Jonathan Ledlie 分别提出在 Grid 和 P2P 环境中的成员管理协议^[21,26]。Grid 的体系结构的实现存在很大的困难,幸运的是可以借鉴 P2P 领域的研究成果,可见二者是相辅相成的。

3.3.2 可用性(Utilization) 资源发现:该策略决定了我们将如何找到所需资源,包括文件、计算机、服务和设备等等。这就需要在高速性、稳定性、有效性和自治性中找到合适的结合点。大多数情况下,在 P2P 领域中普遍使用的基于 Hash 的查找机制可以达到满意的效果。

资源管理和优化:为解决如何能够最好最有效地利用网络中的资源的问题。它包括如何分布元数据,由谁来执行计算任务,如何选择链路进行通信,以及如何加快热点资源的共享与传播等。Grid 方面,William H. Bell 等人提出并模拟了在 Grid 环境中的资源复制策略^[36];P2P 方面,Freenet^[10]作为代表,展示了一种分布式匿名的信息存储与获取系统。Grid 与 P2P 都致力于数据资源的复制和缓存算法,来提高系统的运行效率。

调度与冲突避免:P2P 将重点放在利用带宽方面,其解决方案大多是设计运行于各节点之上适应网络动态变化的应用程序。Grid 致力于有效的任务调度,并利用传统的技术,但这种技术需要中央服务器,容易形成瓶颈,缺乏服务质量和公平性保证^[6]。

负载均衡:P2P 和 Grid 采用的方法都是将大块的任务或数据划分为小块分配到多个节点来平衡负载或将大的数据流分割为多个小数据流并通过不同路径传输^[34]。Karger 等人^[23]改进了基于 Hash 的数据结构并提出 2 个复杂度为常数度的负载均衡策略。

3.3.3 鲁棒性(Robustness or Coping with Failure)

数据完整性:目前 P2P 系统模型往往采用冗余信息来保证数据完整性,并且允许有损的数据完整性。但 P2P 的这种思想无法应用于 Grid,试想,大型的科学计算项目,如 European Data Grid^[11],每个月产生的数据量以 PB 计,并对数据的完整性有极高的要求,冗余的数据将会带来巨大的资金开销。所以,P2P 模型要想适用于 Grid,就必须修改现有的模型。

安全性:包括认证(验证数据和计算,防范恶意攻击)、可用性(防范拒绝服务攻击)和授权(访问控制)。P2P 可以借鉴 Grid 的集中式体系结构来解决部分 Dos 的问题^[1,8]。Grid 方面,GSI(Grid Security Infrastructure)^[17]解决了认证的问题,但是 inter-testbed authentication^[19]仍然亟待解决。沿着类似 SPKI 的分布式认证机制研究也许是一个正确的方向^[9]。Welch 等人^[35]改进并标准化了基于 PKI 的 X.509 代理认证机制,为不同虚拟组织之间相互授权提供了更好的安全机制。

3.3.4 标准 在开发和管理方面,P2P 还没有实质上的标准或 API,除了 Gnutella,每一个版本的模型都采用一套全新的接口。另一方面,Globus 于 2002 年发布了面向 Web Services 的 API——Open Grid Services Architecture(OGSA),作为 Globus 的下一代网格计算平台的标准。

4 分类

网格计算和对等计算是分布式计算领域涌现出的两个新兴的研究热点。从系统特征来看,它们目前的应用系统大致可分为三类(如图 3 所示):网格计算系统、对等计算系统和兼具前两者特征的融合性系统。网格计算系统大都以 Globus 为实

现平台,包括以 European DataGrid^[11], NSF's National Technology Grid (NTG)^[32], NASA's Information Power Grid(NASA IPG)^[22]为代表的各类网格。对等计算系统可进一步划分为两类:文件共享(Napster, Gnutella, Freenet, Oceanstore^[25])和空闲 CPU 周期收集(SETI@home[setiathome.ssl.berkeley.edu]), SUN Microsystems 推出的 JXTA 作为 P2P 系统的开发包正逐渐为广大用户接受。网格计算和对等计算的融合是必然的趋势,但在这条道路上困难重重, Avaki^[2]迈出了难能可贵的第一步,相信不久将会有更多的融合性系统涌现出来。

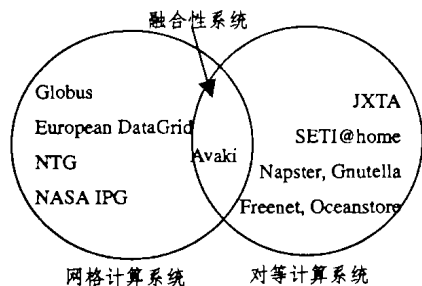


图3 分类(以系统特征为标准)

从应用领域看,网格计算和对等计算可分为高性能科学计算(Avaki, European DataGrid, NTG 等)、拾遗计算(SETI@home, Condor^[27], Entropia[www.entropia.com]等)、文件资源共享(Napster, Gnutella, Freenet, Oceanstore 等)、通信与协作(Magi^[5], Groove[www.grove.net], Jabber, AMIster 等)、支撑平台(Globus, JXTA, .NET My Services 等)。如图4所示。现行的商业系统都是位于某一坐标轴上,即针对某一特定应用;而支撑平台则对每一坐标轴领域提供技术支持。

高性能科学计算 高性能科学应用指以超级计算机为主体,整合互联网上从中心到边缘的计算能力,来完成目前单一的超级计算机无法解决的问题。高性能科学计算主要包括三个方面:分布式超级计算、高吞吐率计算和数据密集型计算。

1) 分布式超级计算。越来越多的科学与工程计算问题已无法在任何一台超级计算机上解决了,网格可以把分布式的超级计算机集中起来,协同解决复杂的大规模计算问题。包括 Avaki, GriPhyN [www.griphyn.org], NASA IPG, DOE's DISCOM^[4], NTG, NEESgrid [www.neesgrid.org]等等。

2) 高吞吐率计算。高吞吐率计算关注的是一段相对较长的时间内完成的计算量。目前网格将空闲的 CPU 资源收集起来,以获取强大的计算能力,提供给对时间不太敏感的计算问题。

3) 数据密集型计算。对于数据密集型问题,数据采集地点、处理地点、分析与结果存放地点、可视化设备位置等往往不在同一个地方。数据密集型问题的求解往往同时会产生很大的通信和计算需求,需要高性能网络的支持。许多高能物理实验、天气预报、数字化天空扫描等都是数据密集型问题,目前网格可以在这类问题的求解中发挥巨大作用,如欧洲数据网格 European Data Grid。

拾遗计算:这类应用将一个大的任务分割成小块,使其能够在相对独立的端节点上处理。以此为模型的应用集中在计算密集型应用。这类应用收集网络上空闲的计算能力,将其汇集成可观的计算能力。在这种情况下,每个端节点具有相同的功能,处理数据的不同子集。最具代表性的拾遗计算系统非搜

寻地外文明(SETI@Home)莫数。

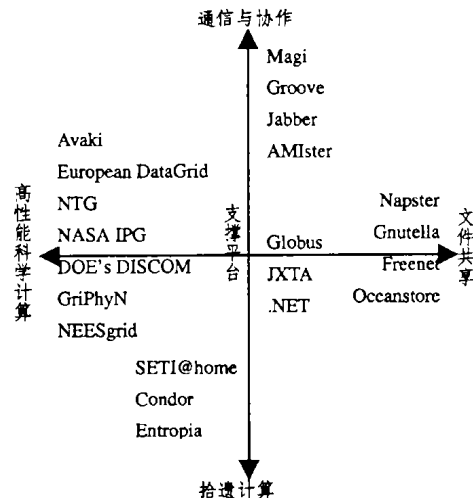


图4 分类(以应用领域为标准)

文件资源共享 这类应用集中在信息的存储与获取上,由内容交换模型驱动。像 Napster 和 Gnutella,每个客户端可以搜索其他用户共享的文件,并直接从对方的硬盘上下载。

通信与协作 这类应用支持用户进行实时的协作,无须依赖于服务器中转信息。包括即时通讯软件、远程协作软件、网络游戏等等。即时通讯软件已经深受网络用户的欢迎,如 Yahoo!、AOL 和 Jabber;远程协作软件可以帮助企业和客户,以及合作伙伴之间建立起一种安全的网上工作联系方式;可以使相隔万里的用户同时查阅并修改同一份文件;这项技术在网络游戏中发挥得淋漓尽致,游戏程序运行在所有的客户端,通讯完全通过 P2P 的方式,避免了中央服务器这一瓶颈。

支撑平台 支撑平台为开发 Grid 和 P2P 应用提供了一系列的服务、软件库和编程接口,有助于规划和组建大型的实验和应用,开发适合大型系统运行的应用程序。主要实现包括安全性、信息获取与分布、资源管理和远程数据传输在内的主要功能。当今代表性的支撑平台有 Globus, JXTA, .NET My Services。

结束语 让我们来想象一下,全世界的计算机连接成一个整体,任何人都可以得到所需的计算能力、存储能力,而不受地域的限制;我们在网络上工作,就像使用个人电脑一样方便快捷。我们把这个叫做虚拟计算平台。这依赖于网格计算和对等计算技术的成功结合。为此,我们需要构建一个稳定并且多用途的体系结构,以一种有组织的、分布式的方式实现在异构的网络环境中的鲁棒性、高性能和安全性。此外,还需建立多样化的发现、协商和维护协议来提供丰富并且高质量的服务。

目前网格计算和对等计算正在从不同的方向向这一目标而努力。随着时间的推移,网格计算在解决系统规模方面的技术将会逐渐成熟起来。网格计算对可扩展性、协商可靠性、自组织、自动问题检测和容错性的需求越来越强,幸运的是对等计算为其提供了大量可行方案。与此同时,对等计算领域,随着技术的飞速进步和越来越多资源的接入,研究者和开发者们越来越信心十足。我们期待着一整套标准的体系结构和开发工具的出现,网格计算和对等计算有共同的目标——构筑跨越互联网的虚拟计算平台,二者的融合正在逐渐成为一个研究热点。

(下转第 229 页)

- wan, 1994. 222~227
- 10 Bourhifir C, Dssouli R, Aboulhamid E M. Automatic Test Generation for EFSM-based System. <http://citeseer.nj.nec.com>
 - 11 Coga N. Comparing TorX, Autolink, TVG and UIO Test Algorithms. Springer-verlag Berlin Heikelberg, 2001
 - 12 [美]Jorgensen P C 著. 韩柯, 杜旭涛 译. 软件测试. 北京: 机械工业出版社, 2003
 - 13 [美]McGregor J D, Sykes D A 著. 杨文宏, 李新辉, 杨洁, 等译. 面向对象的软件测试. 北京: 机械工业出版社, 2002
 - 14 Kim Y G, Hong H S, Bae D H. Test Cases Generation from UML State Diagrams[J]. IEEE Proceeding on Software, 1999, 46(4): 187~192
 - 15 张毅坤, 等. 基于 UML 状态图的类测试用例自动生成方法. 计算机工程, 2003, 29(21): 91~93
 - 16 林宏, 曾一. 基于 UML 的面向对象软件测试框架. 重庆大学学报, 2003, 26(8): 43~46
 - 17 Katayama T, Furukawa Z, Ushijima K. Event Interactions Graph for Test-Case Generation of Concurrent Programs. In: Proc. 1995 Asia-Pacific Softw. Eng. Conf. (APSEC'95), 1995. 29~37
 - 18 Katayama T, Furukawa Z, Ushijima K. Design and Implementation of Test-Case Generation of Concurrent Programs. In: Proc. 1998 Asia-Pacific Softw. Eng. Conf. (APSEC'98), 1998. 262~269
 - 19 Katayama T, Itoh E, Furukawa Z, Ushijima K. Test-Case Generation of Concurrent Programs with the Testing Criteria Using Interactions Sequences. In: Proc. 1999 Asia-Pacific Softw. Eng. Conf. (APSEC'99), 1999. 590~597

(上接第26页)

参考文献

- 1 Adya A, Bolosky W, Castro M, et al. FARSITE: Federated, available, and reliable storage for an incompletely trusted environment. In OSDI '02, Boston, MA, 2002
- 2 AVAKI CORPORATION. Avaki 2.0 Concepts and Architecture. White paper, 2001
- 3 Ball T, Rajamani S K. Automatically validating temporal safety properties of interfaces. In: the Proc. of the Intl. SPIN Workshop on Model Checking of Software, May 2001. 333~344
- 4 Beiriger J, Johnson W, Bivens H, Humphreys S, Rhea R. Constructing the ASCII Grid. In: Proc. 9th IEEE Symposium on High Performance Distributed Computing, 2000
- 5 Bolcer G. Magi: An Architecture for Mobile and Disconnected Workflow. White paper, 2001
- 6 Buyya R, Stockinger H, Giddy J, Abramson D. Economic models for management of resources in peer-to-peer and grid computing. In ITCOM 2001, Aug. 2001
- 7 Castro M, Liskov B. Practical Byzantine Fault Tolerance. In: Proc. Usenix Symp. Operating Systems Design and Implementation (OSDI 99) Usenix Assoc., Berkeley, California. 1999
- 8 Castro M, Drushel P, Ganesh A, Rowstron A, Wallach D. Secure routing for structured peer-to-peer overlay networks. In OSDI '02, Boston, MA, 2002
- 9 Clarke D, Elien J, Ellison C, Fredette M, Morcos A, Rivest R. Certificate chain discovery in SPKI/SDSI. Journal of Computer Security, Jan. 2001
- 10 Clarke I, Sandberg O, Wiley B, Hong T. Freenet: A distributed anonymous information storage and retrieval system
- 11 European union data grid project. [http://eu-datagrid/](http://eu-datagrid.web.cern.ch/eu-datagrid/), 2001
- 12 Foster I, Iamnitchi A. On death, taxes, and the convergence of peer-to-peer and grid computing. In: 2nd Intl. Workshop on Peer-to-Peer Systems (IPTPS'03), Feb. 2003
- 13 Foster I, Kesselman C, Nick J, Tuecke S. The physiology of the grid: An open grid services architecture for distributed systems integration, 2002
- 14 Foster I, Kesselman C. The Grid: A New Infrastructure for 21st Century Science, 1998
- 15 Foster I, Kesselman C, Tuecke S. The Anatomy of the Grid: Enabling Scalable Virtual Organizations. International Journal of High Performance Computing Applications, 2001, 15(3): 200~222
- 16 Foster I, Tuecke S. What is the Grid? A Three Point Checklist. July 2002
- 17 Foster I, Kesselman C, Tsudik G, Tuecke S. A security architecture for computational grids. In: ACM Conf. on Computers and Security, Nov. 1998
- 18 Chen Guihai, Shen Haiying, Xu Cheng-Zhong. Cycloid: A Constant-Degree and Lookup Efficient P2P Overlay Network
- 19 Humphrey M, Thompson M. Security implications of typical grid computing usage scenarios. In HPDC 10, Aug. 2001
- 20 Houston B. The "GnutellaMandrill" virus. www.exocortex.org/gnutella, 2001
- 21 Iamnitchi A, Foster I. On fully decentralized resource discovery in grid environments. In: Intl. Workshop on Grid Computing. IEEE, Nov. 2001
- 22 Johnston W E, Gannon D, Nitzberg B. Grids as Production Computing Environments: The Engineering Aspects of NASA's Information Power Grid. In: Proc. 8th IEEE Symposium on High Performance Distributed Computing, IEEE Press, 1999
- 23 Karger D, Ruhl M. Simple Efficient Load Balancing Algorithms for Peer-to-Peer Systems. In: 3rd Intl. Workshop on Peer-to-Peer Systems (IPTPS'04), Feb. 2004
- 24 Katzenbeisser S. Information hiding techniques for steganography and digital watermarking. Artech House Books, 1999
- 25 Kubiawicz J, Bindel D, Chen Y, et al. OceanStore: An Architecture for Global-Scale Persistent Storage. In: Proc. of the Ninth Intl. Conf. on Architectural Support for Programming Languages and Operating Systems (ASPLOS 2000), Nov. 2000
- 26 Ledlie J, Taylor J, Serban L, Seltzer M. Selforganization in peer-to-peer systems. In 10th EW SIGOPS, Sep. 2002
- 27 Litzkow M, Livny M, Mutka M. Condor - A Hunter of Idle Workstations. In: Proc. 8th Intl. Conf. on Distributed Computing Systems, 1988
- 28 Malkhi D, Naor M, Ratajczak D. Viceroy: A Scalable and Dynamic Emulation of the Butterfly. In: Proc. of Principles of Distributed Computing (PODC 2002), 2002
- 29 Milojevic D S, Kalogeraki V, Lukose R, et al. Peer-to-Peer Computing, HP Technical Report, HPL-2002-57
- 30 Ratnasamy S, Francis P, Handley M, Karp R, Shenker S. A Scalable Content-Addressable Network. In: Proc. of the ACM SIGCOMM 2001 Technical Conference, San Diego, CA, USA, Aug. 2001
- 31 Rowstron A, Druschel P. Pastry: Scalable, decentralized object location, and routing for large-scale peer-to-peer systems. In Middleware, Nov. 2001
- 32 Stevens R, Woodward P, DeFanti T, Catlett C. From the I-WAY to the National Technology Grid. Communications of the ACM, 1997, 40(11): 50~61
- 33 Stoica I, Morris R, Liben-Nowell D, et al. Chord: A scalable peer-to-peer lookup service for internet applications. Research report, MIT, Jan. 2002
- 34 Padmanabhan V N, Wang H J, Chou P A. Supporting Heterogeneity and Congestion Control in Peer-to-Peer Multicast Streaming. In: 3rd Intl. Workshop on Peer-to-Peer Systems (IPTPS'04), Feb. 2004
- 35 Welch V, Foster I, Kesselman C, et al. X.509 Proxy Certificates for Dynamic Delegation. 3rd Annual PKI R&D Workshop, 2004
- 36 Bell W H, Cameron D G, Capozza L, et al. Simulation of dynamic grid replication strategies in optosim. In Grid 2002, Nov. 2002
- 37 Xu Z, Reps T, Miller B. Typestate Checking of Machine Code. In: Proc. of the 10th European Symposium on Programming, April 2001
- 38 Zhao B Y, Kubiawicz K D, Joseph A D. Tapestry: An Infrastructure for Fault-Resilient Wide-Area Location and Routing. Technical Report UCB//CSD-01-1141, University of California at Berkeley Technical Report, April 2001