

网格:面向虚拟组织的资源共享技术*)

高全泉

(中国科学院数学研究所 北京 100080)

Grid:Resources Sharing Technologies for VO(Virtual Organizations)-Oriented

GAO Quan-Quan

(Institute of Mathematics, Chinese Academy of Sciences, Beijing 100080)

Abstract The "Grid" was coined in the mid1990s to denote a proposed distributed computing infrastructure for advanced science and engineering. Now, it has emerged as an important new field. Grid computing can be defined as coordinated resources sharing and problem solving in dynamic, multi-institutional virtual organizations. In this paper, we introduce the idea of virtual organizations, present characteristics and requirements of virtual organizations resources sharing, some important terms in Grid technologies are defined, and principle fundamentals and critical ideas of Grid architecture are discussed.

Keywords Grid, Grid computing, Virtual organizations, Resources sharing, Distributed computing, High performance computing, Supper computing

1. 引言

在过去的几年里,“网格”(Grid)一词主要在学术界使用。如今,它已从幕后走到了前台,在IT界引起人们的普遍关注。术语网格源于学术界的学派描述共享互联网的所有资源(从PC到超级计算机),以共同地解决超级计算任务。一般认为,有关网格的实质性研究始于1995年。当时美国Argonne国家实验室的Ian Foster博士和南加州大学信息科学研究所的Carl Kesselman博士共同领导了美国政府(能源部、NASA等)支持的高性能分布式计算项目Globus^[1,2]。在此项目中,为了标记现代科学和工程方面的分布式计算基础结构,首次使用了具有实际内涵的“网格”,提出了一种前所未有的网格理念并实施了相关研究与发展。作为一种能够广泛利用Internet上的各种资源,实现虚拟组织的资源共享和问题求解技术,网格从提出到现在,一直受到各国政府、学术界和著名高科技企业的高度重视,恰似一石激起千重浪,掀起了Internet的新的技术浪潮。在构造这种基础结构方面,重要的进展不断获得,令人颇受鼓舞。特别是Globus所做的贡献,包括他们提出的网格理论框架、体系结构以及实现资源共享的协议和软件“工具包”,为网格发展奠定了坚实的基础。

如今,美国政府用于网格技术基础研究的经费达5亿美元,军方正在规划实施为期达二十年,投资数百亿美元的“全球信息网格”。英国政府已投资1亿英镑研制“英国国家网格(UK National Grid)”。在学术界,英国爱丁堡大学的并行计算中心正在开发网格和高性能计算方法,美国的休斯顿大学开始建立关于地质应用,如寻找石油储油层的大学校园网格。网格对互联网的深远影响更为众多国际著名的IT公司所关注,他们纷纷支持Globus计划所做的努力,来将网格的处理(如计算)和存贮资源的用法标准化。比如,IBM公司在2001年8月宣布,投资四十亿美元实施“网格计算计划”。SUN与

Globus 结盟,最近正在升级源代码网格引擎软件(代码超过5万行),并帮助推广Globus的网格“工具包”,双方联手开发将该“工具包”连接到SUN的网格引擎软件的接口,等等。所有这些活动表明,一项当今互联网上最重要的技术——网格,已经产生并已渐入佳境。它是互联网上继Web之后新的技术浪潮。在回答“互联网(Web)之后最重要的技术是什么?”这一问题时,IBM负责互联网策略的副总裁帕托克说道“现在还没有答案,但我相信是网格”^[3]。这代表了当今的一种共识。

当年,在构造科学和工程的分布计算的基础结构中,将术语“Grid”用于标识这种基础结构,使得二者融为一体,涵盖从先进的网络到人工智能的一切事物。现在,网格可谓一枝红杏出墙来。尽管如此,作为一种脱颖而出的新技术,网格的理念及内涵自然需要有一个为人所知的过程。客观上,系统地介绍和分析网格技术的专文也较少,已有的相关文章也或多或少存在不够系统和对网格理解的局限,以至于人们对网格领域或本质知之不多或不得要领。于是,就“Grid”是否真有一些实际要领和意义存在如下一些疑问:

问题一:现实中,是否存在一类明显的“网格问题”,以及由此而存在对新的网格技术的需要?

问题二:若然,这些技术的性质是什么?它们的应用域是什么?

问题三:网格的技术内涵是什么?

问题四:网格与当前其它主要的技术趋势有何区别?

虽然多数组织在网格概念及共享,乃至到更有意义的网格的体系结构的共同看法上有兴趣,但是,迄今尚未看到对这些问题的一致答案。对网格的理解与认识可谓众说纷纭,莫衷一是。既有仁者见仁,智者见智,也有误解与谬传。特别是,近年来随着网格影响的扩大,对网格的提法难免有不实和不准确之处,与创始者的本意并不一致。显然,这不利于网格的研究、发展、应用及传播。网格作为一种新的技术,如果缺乏

*)国家自然科学基金项目,批准号60072006。高全泉 研究员,研究方向为人工智能、知识工程、软件体系结构、并行/分布计算(中间件)技术等。

这一领域的相关定义、专业用语的解释以及比较鉴别,也就很难成为一门公认的学科。

为了解答和澄清以上问题, Ian Foster, Carl Kesselman, Steven Tuecke 在 2001 年的 Intl J. Supper-computer Application 15 上发表专文^[4],论证了网格问题既非凭空杜撰,也非闭门造车,它确实由实际和专门的问题所驱动,并且存在一个正在形成中的解决此类问题的有意义的各个方面的网格技术基础。此前,他们开发了关于当前和今后的网格技术的详细的网格体系结构和路线图。值得一提的是,该文是他们数年研究和实践的总结和归纳,定义了网格研究的领域和目标,提出了网格体系结构,论述了网格与其它分布计算技术的关系。如此,使得网格理念和思路更清晰,提法和概念更科学、准确。该文作者认为,尽管网格技术与当前其它主要技术趋势,如 Internet, enterprise, distributed 以及 Peer-to-peer 计算性质不同,但通过将这些技术拓展到网格技术所解决的问题领域,这些技术能从中获益匪浅。毋庸置疑,该文对网格研究领域的界定和论述是全面和系统的,体现了网格创始人的真实思想;提出的网格体系结构和核心技术具有科学性和可行性,已在实践中得到检验;通过与已有技术的比较,有利于正确理解网格技术,纠正一些模糊和错误的观点。我们根据这一文献及其它相关文献,围绕以上问题介绍和分析网格技术。由于涉及内容较多且范围较广,为方便阅读,我们分三个专题进行综述。本文主要围绕上述一、二两个问题,讨论网格的需求背景,给出相关概念定义和虚拟组织资源共享特性和需求,介绍网格一般技术理念/框架。其余两个专题分别是“网格体系结构详解”和“关于网格及其它分布计算的若干问题的讨论”,分别讨论三、四两个问题。

2. 虚拟组织的意义及网格问题的提出

说到“网格”,当然不能不提到虚拟组织。网格所指的实际和专门的问题是:在动态的、多机构的虚拟组织中可调整的资源共享和问题求解。它所关心的共享是直接访问计算机、软件、数据和其它资源,如在工业、科学和工程中出现的协作性问题求解和资源代理中所要求的资源,而不同于某种特定的资源(如基本的文件共享)。这里,关于共享的规则是:

- 由资源提供者和消费者高度可控;
- 资源提供者和消费者明确、详细地定义什么被共享,允许谁被共享以及共享出现的条件。

2.1 虚拟组织概念及说明性例子

网络时代,虚拟组织的例子俯拾皆是。虚拟组织是指由以上共享规则所定义的一组个体与/或机构。虚拟组织的活动是跨实际组织的资源共享与协作求解。为简洁,我们用 RO 和 VO 分别代表实际组织和虚拟组织。为理解网格的产生背景,我们援引 Ian Foster 等列举出的一些经典的 VO 实例及其活动内容。这些例子有的是网格实际应用项目,在我们关于网格的专题介绍中也以它们为背景。

例 1 某汽车制造企业计划筹建一个新的工厂,该企业以及由它约请的对新建厂的前景做评估的顾问、应用服务提供者(ASP)、存储服务提供者(SSP)以及相关系列提供者构成一个虚拟组织。

在做建立新厂的决策时,企业从 ASP 那里调用精细的金融预测模型并为之提供对源于企业数据库的适合的私有数据的访问能力,该企业数据库处于由 SSP 操作的存储系统中。在决策会议期间,what-if 情景协作地、交互地运行,即便参与决策的部门领导位于不同的城市。在详细的询问进行期间,ASP 与一个有衔接决策全过程能力的系列提供者签约,并要求系列提供者满足所期望的安全性和性能要求。

例 2 在一个新的航空器项目上投资的工业协会。

一个工业协会组织开发下一代超音速航空器,承担整个航空器的高精度多学科模拟。该模拟集成由不同的合作伙伴开发的专用的软件组件。这些组件在合作伙伴各自的计算机上操作,并可访问适当的设计数据库以及其它由别的成员使得可用于该协会的数据。

例 3 应急管理小组、数据库及用于对一个紧急情况作出响应的模拟系统。

一个应急小组通过使用本地的气候和土壤模型,在对化学泄露作出响应时,估算泄露的范围,确定基于人口居住位置以及地理特点(如河流和供水)的影响,建立一个短期的减轻灾害影响的计划,并通过安排和协调疏散、通知医院等手段,为应急响应的全体人员分派工作。

例 4 大规模国际性的、多年的高能物理协会。

在世界范围内数以百计的实验室和大学内工作的上千名物理学家在 CERN(欧洲高能物理实验室)会合,创建、操作并分析一个主要的探测器产品。在分析阶段,他们统筹其计算、存储以及网络资源以便建立具有分析天文数字级的数据能力的“数据网格”。

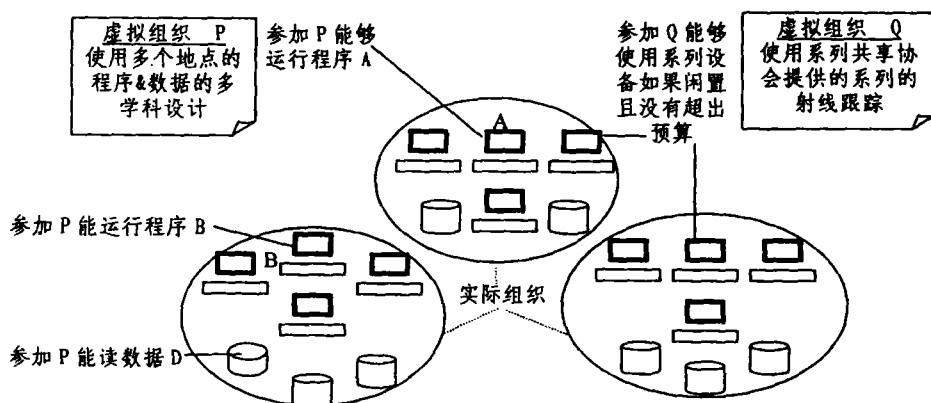


图 1 虚拟组织与实际组织的关系示意图

以上的每个例子表示一种计算与问题求解解决方法,这种方法基于在计算与数据丰富的环境中的协作。通过共享某

些或全部资源,一个实际组织能够参加一个或多个 VO。比如,在图 1 中,给出 3 个 RO 和 2 个 VO(P 和 Q),分别用椭圆

和矩形表示:P 连接空间设计协会中的伙伴;Q 连接已经同意将闲置的计算系列供他人共享的大学。左边的 RO 参加 P,右边的 RO 参加 Q,中间的 RO 既是 P 的成员又是 Q 的成员。对访问资源进行管理的策略随 RO、资源和所涉及的 VO 而不同,这在图 1 中作了归纳。

2.2 VO 资源共享的特性及需求

由 2.1 节的例子可知,VO 在目的、作用域、大小、持久性、共有性和社会性方面有很大差异。上述四个例子的参与者的成员和类型不同,从事的活动的类型、交互的时间和规模以及被共享的资源也不相同。

然而,经过研究发现,VO 的共同点更多。它们有共同关心的问题 and 相同的需求,本质的问题是对跨组织的协作和资源共享的需求。这种协作性资源共享的基本特性和共同需求如下:

- 资源多样化:从程序、文件、数据到计算机、传感器、网络等。
- 关系通常不是简单的 Client-Server,而是 Peer-to-Peer (对等)。

提供者可以是消费者,共享关系可以在伙伴的任何子集间存在。共享关系可以被合并,以便于协调使用跨许多不同的组织所拥有的资源。例如,在 VO Q 中,在一个轮回的计算资源上开始的计算可以在其后访问数据或开始实施别处的子计算。

这种协调跨多个资源的操作所要求的机制是能以可控的方式委派权力。

- 共享关系灵活并高度可控。表现在:

1) 依据所调用的资源,所允许的访问性质以及被允许访问的参加者,共享关系可随时间动态地改变。这些关系并不必然地需要一个明显的个体的命名集,而是可由对资源的访问进行管理的策略隐含地定义。例如,一组织能被任何人访问只要来人能证明自己是“客户”或“学生”。

此种动态性要求即时发现并将一个特别的点上存在的性质进行特征化。例如,参加 VO Q 的一个新伙伴必须能确定能够访问什么资源,这些资源的质量以及对访问进行管理的策略。

2) 共享受共享条件的控制,每个资源拥有者使得资源可用,使共享受时间、地点以及所能做的事情的限制。例如,在图 1 中的 VO P 的一参加者可以允许 VO 参与者只调用其“简单”问题的模拟服务。资源消费者也并非没有选择权,可在他们的打算与其它资源一起工作的资源上安排限制。比如,VO Q 的一个参加者可以只接受已经验证为“安全的”入池的计算资源。

实现这种限制要求如下机制:表示策略、建立消费者/资源(认可)的标识、确定一操作是否与可用的共享关系(认可)相一致。

- 相同资源以不同模式使用的可协调性。根据在共享和共享目标上所安排的限制,即便是相同的资源也能以不同的方式使用。这是因为资源的使用与活动的目标密切相关,目标不同,用法也有所不同。比如,在一个共享安排中,一计算机可以仅用于运行某个软件的片段;但在另一个安排中,它可以提供一般的计算系列(执行活动的启承转合)。其它用法如,以单用户还是多用户使用,是以注重性能还是注重成本使用。

这种可变的用法需要有资源被如何使用的先验知识。为了不使问题复杂化,可以在资源共享或用法上安排性能度量、

期望、限制(即服务的质量)等来实现。

- 需要具备如何使用共享资源的精细和精确的控制层。包括粒度良好的和多个栈持有者访问、委派、局部和全局策略的应用。这种控制层是 VO 资源共享特性所要求的至关重要的需求。

2.3 “网格”问题的提出

在科学、工程及商业等许多不同的学科和活动中,协作性处理是最基本的需要。由于能够使组织与/或个体的团体(或小组)能以一种可控的风格共享资源,其成员能够协作完成共享的目标,因此 VO 正在成为当今诸多学科和活动,特别是现代计算不可或缺的概念。

综观当前的分布式计算技术,虽然门类很多,但它们并不能解决 VO 问题,即不具备 VO 要求的特性和需求。主要表现在,或者不能适应资源类型的范围,或者不提供建立 VO 所需的共享关系之上的灵活性和控制能力。这可通过分析以下几种主流技术的主要特性看出:

- 当前的 Internet 技术解决计算机之间的通信和信息交换,并不提供关于计算的多站点上资源的可协调使用解决方法。

- Business-to-business 交换着眼于信息共享(通常通过中央控制的服务器)。

- 虚拟企业技术与 Business-to-business 类似,虽然共享最终能扩展到应用和物理设备。

- 企业分布式计算技术,如 CORBA,Enterprise Java 使得仅能在单一组织内共享资源。

- 开放组织的分布式计算环境 DCE 支持跨站点的安全的资源共享,但大多数 VO 觉得它过于烦琐而令人费解,且缺乏灵活性。

- 存储服务提供者 SSP,应用服务提供者 ASP,虽然允许组织将存储和计算机需求输出到其它伙伴,但仅仅是以一种受限的静态配置方式使用。例如,SSP 资源通过一个虚拟的私有网络 VPN,典型地连接到一个顾客。

- 正在形成的“分布式计算”公司寻求利用国际范围的闲置的计算机,但时至今日,仅支持对这些资源的高度集权控制的访问。

正是由于当前的其它技术不能解决动态的和跨组织的资源共享,因此建立可伸缩 VO 并实现 VO 间资源共享就成为新的一类问题,这就是所谓的“网格问题”。

“网格问题”的解决面临惟一性认可、认证、资源访问、资源发现等诸多挑战。在过去的几年里,经过网格共同体的研究和努力,已经产生了一系列能精确地解决在寻求建立可伸缩 VO 时所出现的各种挑战的协议、服务和工具,我们称之为网格技术。网格技术的基本内容包括:

- ✓ 安全性解决方案,支持当计算跨越多个机构时的信任和策略的管理

- ✓ 资源管理协议,支持安全的远程访问计算和数据资源及多种资源的合作分配

- ✓ 信息查询协议和服务,提供关于资源、组织及服务的配置和状态信息

- ✓ 数据管理服务,确定存储系统和应用的位置并在二者间传输数据集

Ian Foster 认为,VO 有潜力戏剧性地改变我们求解问题使用计算机的方式,正如 Web 已经改变了我们交换信息的方式。“网格”是面向 VO 的学科,它使得建立可伸缩的 VO 成为

可能,因而成为 Internet 之上新的技术浪潮。

3. 若干基本概念的定义

在网格技术的描述中,频繁用到一些基本的概念。这些概念是:协议、服务、API 和 SDK。事实上,它们既非新概念,也非仅出现在网格中,早就在 Internet 及分布计算领域中普遍出现。然而,在不同场合,对它们的定义和解释却不尽相同。为了有助于网格的理解,这里给出它们的一般性定义及用法。

3.1 协议

协议是远程通信系统中,末端节点在交换信息时所使用的一组规则。

具体来说,一协议定义在两个系统间发送的信息的格式,信息包括被交换的数据本身和如何交换数据的“元数据”,即消息。因此,协议定义被交换数据的格式以及消息的文法、字符集以及消息的次序。例如,FTP 协议定义指出用于协商一个文件传输的消息的格式,但并不规定接受实体如何管理它的文件。协议也不规定信息发送是通过调用 Java 方法,调用 C 语言函数,还是通过在终端前的人员输入响应完成的。协议可由其它协议定义^[5]。

协议的一个重要性质是可供多个实现之用:两个端点只需实现相同的协议就能够通信。因此,在分布式计算环境中,标准的协议是达到互操作性的根本。例如,它允许在 Sun 工作站上运行的 Java 应用与在 Cray-3 上运行的 C 应用一起工作,正如 TCP/IP 使得异构的操作系统和计算机集能在 Internet 上通信那样。

协议举例:

- Internet Protocol(IP)定义一个不可靠的包传输协议;
- 建立在 IP 之上的 Transmission Control Protocol (TCP)定义一个可靠的数据交付协议;
- Transport Layer Security(TLS)协议定义在两个通信的应用间,提供保密和数据完整性的协议。它位于可靠传输协议(如 TCP)的上层。
- 建立在 TCP 之上的 Lightweight Directory Access Protocol(LDAP)定义查询远程数据库状态的查询-响应协议。

3.2 服务

一个服务是通过网络能提供专门能力的实体。例如,移动文件能力,建立进程或验证访问权的能力。一个服务由与之交互所使用的协议和响应不同的协议消息交换所期望的行为所定义,可表作:“服务=协议|行为”。

一服务允许各种实现。例如:

- 一个 FTP 服务器讲文件传输协议,并支持对文件集的远程读写。一个 FTP 服务器实现可以简单地从本地磁盘读出和写入,而另一个实现则从一个巨大的存储系统读出和写

入,并在此进程中自动地压缩和解压文件。从基础结构级的观点来看,此二种服务器响应一个存储请求的行为完全不同,然而,在要求服务的客户看来,其行为并无差异。因为,存储一个文件然后检索该文件将得到相同的结果,不在意使用了哪个服务器实现。

• 一 LDAP 服务器讲 LDAP 协议并支持对查询的响应。一个 LDAP 服务器实现可以使用一个信息的数据库响应查询,而另一个则可以通过动态地形成 SNMP 调用产生运行中必需的信息。

一个服务可以是持久性的,也可以不是持久性的,它能够探测一定的错误与/或由一定的错误做恢复,以特权运行,与/或有提高了可伸缩性的分布式的实现。如果变体是可能的,则需要允许客户确定一服务的特例的机制。能够定义讲相同协议的不同服务,如在 Globus 中,备份目录和信息服务都使用 LDAP 协议。

技术上,服务是一些这样的进程,它们通常自动地启动并对其它进程提供基本的功能性。它们也被称为用户代理和守护神(daemons)。来看 E-Mail 服务的情形。E-Mail 服务或许是最为大家所熟知的服务,它在后台运行。一个 E-Mail 程序(客户程序)简单地使用一个规定的协议与服务器通信并为服务器提供数据(地址表,正文,附加等)。Email 服务实际看管在网上传输邮件,另一端的服务通知邮件已经到达的适当的客户。服务的优点是使得应用程序开发更快和程序更小巧,因为每个客户不需给出服务代码而可直接使用。由于客户不必编写更为错综复杂的服务代码,因而应用程序也更加稳定可靠。

3.3 应用程序接口 API

一个 API 是为了调用一个规定的功能集而定义的标准的函数。它规定应用程序员遵循的能用来编写代码的接口。这种接口包括功能性,名字,数据类型,参数序列及返回类型。一般的 API 是子程序调用形式,面向对象的 API 是对象和方法调用的形式。

一个 API 可以定义多语言联编或使用一种接口定义语言。这种语言可以是传统的编程语言,如 C 或 Java;也可以是外壳语言,此时 API 将参照命令行自变量的具体定义,程序的输入/输出以及程序的出口状态。一个 API 通常规定一个标准的行为,同一个 API 可以有多个实现存在。例如,如果在各种异构的平台上支持相同的 API,并以此 API 作为不同的存储系统的接口,则程序员的任务将大大简化。

理解 API 与协议之间的关系是重要的。一个协议并不规定可在程序里面调用以便产生协议消息的 APIs 的任何事情。一个单独的协议可以有許多 APIs;一个单独的 API 可以有针对不同协议目标的多个实现。概括地说来,标准的 API 使得可移植性成为可能并能提高效率;标准的协议使得互操作性成为可能。

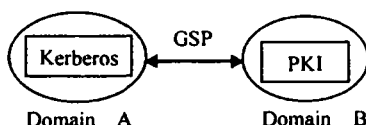
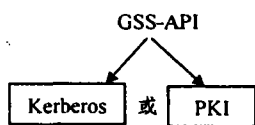


图 2 左边,一个 API 被用于开发能将 Kerberos 或 PKI 安全性机制作为目标的应用。右边,GSP(由 Globus 提供的网络安全协议)协议被用于使得 Kerberos 和 PKI 域之间能够互操作。

例如,Public Key 和 Kerberos 联编二者已被定义于 GSS-API(GSS 代表一般的安全性服务),因此,使用 GSS-API^[6]调用认可操作的程序,可以或者在 Public Key 中操作,

或者在 Kerberos 环境中操作而不需改变。另一方面,如果我们希望一个程序同时能在 Public Key 和 Kerberos 环境中操作的话,则需要支持此二环境的互操作性的标准协议。如图 2

所示。

以下是 API 的例子:

- 一般的安全性服务 (GSS) API 定义验证通信方的标识, 加密消息等有关的标准功能。

- 消息传递接口 API 用数种语言定义了一个并行的计算系统中, 有关用于在进程间传输数据的功能的标准接口。

3.4 软件开发工具包 SDK

术语软件开发包标记一组打算与应用程序链接在一起, 并从应用程序里面调用以提供规定的功能的代码集。一个 SDK 典型地实现一个 API。如果一个 API 准许为多个实现所用, 则对此 API 将有多 SDK。某些 SDK 通过特别的协议提供对服务的访问。例如,

- OpenLDAP 版本包括一个 LDAP 客户 SDK, 该 SDK 包含能由 C 或 C++ 应用使用的执行查询 LDAP 服务的函数库。

- JINI 是一个 Java SDK, 它包含能用于执行查询一个 LDAP 服务的函数。

- 不同的 SDK 分别使用 TLS 和 Kerberos 协议实现 GSS-API。

可能有多个, 比如来自多个厂商的 SDK, 它们实现一个特别的协议。更进一步, 对面向客户-服务器的协议, 可以有分开的客户 SDK 和服务 SDK。前者由希望访问一个服务的应用使用, 后者由希望实现特别的、定制的服务行为的服务实现者使用。

一个 SDK 不需讲任何协议。例如, 提供数字功能的 SDK 可以完全本地化地行动而不需谈及执行该操作的任何服务。

4. 网格体系结构的设计思想及其核心理念

实现 VO 间的资源共享是一个系列活动, 包括动态的跨组织的 VO 共享关系的建立、管理和利用 3 个方面。既然当今的其它主流技术不能解决这些问题, 就要求新的技术, 需要寻求突破。网格技术就此登场并扮演这种新技术的角色。

从功能上来看, 网格是一种建立在已有基础平台之上的一种基础设施, 本质上是一种中间件系统(中间件是在分布式网络环境中支持应用程序的公共集所需要的服务)。IBM 的帕托克认为, “现有的资源, 诸如网络、超级计算机、服务器、操作系统、数据库、文件系统等都是网格计算的底层设备, 网格计算之上是应用程序, 这些应用程序通过网格计算调用、共享网上资源来完成任务。因此, 将网格计算看成一种中间件应该算是恰如其分”^[3]。为了有别于已有的中间件, 可将网格称为超级中间件。作为一种超级中间件系统, 首要的问题是体系结构的定义与设计。因此以具体的“网格体系结构”形式构造对这一技术的讨论是最恰当的。“网格体系结构”给出基本的系统组件, 规定这些组件的目的与功能, 并指出这些组件如何与别的组件交互。

在定义网格体系结构时, 基于这样的出发点: 即, 通过有效的 VO 操作, 使得能够在任意潜在的伙伴间建立共享关系。不同资源对象的共享关系实质上是一种互动能力, 一资源既可操作别的资源, 也可被别的资源所操作, 实现这种能力的技术手段需要通过互操作完成。因此, 互操作性是需要解决的核心问题。更进一步, 在网络互联的环境中, 互操作性意味着共同的协议。因此, Ian Foster 等认为, 网格体系结构的首要且最根本的体系结构是协议体系结构。用协议来定义由 VO 用户与资源协商, 建立、管理并利用共享关系所使用的基本机

制。一个基于标准的开放的体系结构有利于可扩展性、互操作性和代码共享; 标准化的协议能使定义增强能力的标准服务更加容易。我们也能构建应用程序接口 APIs 和软件开发工具包 SDK 来提供建立一个可用网格所要求的程序设计抽象。这些技术与体系结构一起, 构成了通常所说的中间件概念^[7]。在我们的专题综述中, 因中间件概念过于含糊不清且为了有别于已有的技术, 尽量避免使用/少用该术语。

由 Ian Foster 等提出的网格体系结构是一种层次式的协议体系结构, 共有 5 层。这种体系结构涉及的内容较多, 涉及基本的系统组件, 详细规定这些组件的目的与功能, 并指出这些组件如何与别的组件交互。既有理论框架和一般性描述, 也有实践模型, 且已在世界范围的众多项目中得到应用。该体系结构事实上集结了网格相关主要技术, 对网格领域的研发具有普遍指导意义, 我们将专文予以完整准确的介绍。这里, 为了内容的完整, 我们按照由底向上的次序, 对这一体系结构做如下简介:

- 基础结构层, 它们是物理或逻辑实体, 作用是向上提供可供共享的资源。

- 连通层, 是网络事务处理通信和授权控制的核心协议。基础结构层提交的各种资源间的数据交换都在这一层的控制下实现, 并实现各资源间的授权验证、安全控制。

- 资源层, 共享单独的资源。定义在个别资源上的共享操作的安全协商、创始、监控、控制、记账和付费方面的协议。这些协议的实现调用基础结构层的功能来访问和控制本地的资源。资源层只关心单独的资源, 不考虑跨分布收集的全局性和原子性(如事务)问题。

- 汇集层, 协调多个资源。定义全局的、跨所有资源捕捉相互作用的协议和服务等, 不与任何一个专门的资源关联。实现资源的汇集和各种共享行为。

- 应用层, 由在 VO 环境中操作的应用组成。应用通过在任意层中定义的服务以及在这些服务上的调用的形式构成。在体系结构的每一层, 有定义良好的提供访问某些有用服务的协议, 如: 资源管理, 数据访问, 资源发现等。

在网格体系结构的构造中, 首先强调解决互操作性的协议的标识和定义, 其次强调 API 和 SDK。这些概念使用我们在第 3 节给出的定义。网格体系结构设计的主要理念及相应支持如下:

1) 互操作性是 VO 资源共享最根本的问题 为了保证能在任意方创始共享关系, 跨不同的平台、语言和程序设计环境, 动态地接纳新参加者, 不一致正是我们所需要的。在这种上下文中, 机制若未被定义且实现成能跨组织界限、操作策略和资源类型可互操作的, 则几乎没有意义。如果没有互操作性, 必然迫使 VO 应用伙伴参加一个双边的共享安排, 因为缺少将任意两方间所使用的机制扩展到任何其它方的保证。缺乏这种保证, 动态的 VO 构成到此为止且根本就不可能。即便能, 所构成的 VO 的类型也被严格限制。正如 Web 通过提供信息交换的通用协议 HTTP 和文法 HTML 改革了信息共享那样, 对一般化的资源共享, 要求标准的协议和文法。

2) 协议是解决互操作性的关键所在 协议定义了为完成一个规定的行为, 分布式系统元素怎样相互交互, 以及在交互期间交换的信息的结构。协议着眼于外部(交互)而不是内部(软件, 资源特性), 因而对互操作有重要的实际益处。我们知道, VO 往往是不固定的。因此, 要求用于发现资源、建立标

(下转第 11 页)

- 2 Foster I, Kesselman C. The Globus Project: A Status Report. In: Proc. Heterogeneous Computing Workshop, IEEE Press, 1998. 4~18
- 3 Foster I, Kesselman C, Tuecke S. The Anatomy of the Grid: Enabling Scalable Virtual Organizations. Intl. J. Supercomputer Applications, 2001, 15(3)
- 4 Realizing the Information Future: The Internet and Beyond. National Academy Press, 1994. <http://www.nap.edu/readin-groom/books/rtif/>
- 5 Steverson R, et al. From the I-WAY to the National Technology Grid. Communications of the ACM, 1997, 40(II): 50~61
- 6 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
- 7 Beiriger J, et al. Constructing the ASCI Grid. In: Proc. 9th IEEE Symposium on High Performance Distributed Computing, IEEE Press, 2000
- 8 <http://griphyn.org>
- 9 <http://www.neesgrid.org>
- 10 <http://www.ppdg.net>
- 11 <http://www.en-datagrid.org>
- 12 Foster I, Roy A, Sander V. A Quality of Service Architecture that Combines Resource Reservation and Application Adaption. In: Proc. 8th International Workshop on Quality of Service, 2000
- 13 Czajkowski K, Fitzgerald S, Foster I, Kesselman C. Grid Information Services for Distributed Resource Sharing. 2001
- 14 Allock W, Foster I, Kesselman C, Tuecke S. Protocol and Services for Distributed Data-Intensive Science. ACAT2000 proceedings, 2000. 161~163
- 15 Allcock B, et al. Secure, Efficient Data Transport and Replica Management for High-Performance Data-Intensive Computing. In Mass Storage Conference. 2001
- 16 Czajkowski K, et al. Resource Management Architecture for Metacomputing Systems. In The 4th Workshop on Job Scheduling Strategies for Parallel Processing, 1998. 62~82
- 17 李国杰. 信息服务网格——第三代 Internet. 计算机世界, 2001. 11. 15
- 18 徐志伟. 因特网之后是什么? ——网格技术探讨. 计算机世界, 2001. 11. 22
- 19 严世宗译. IBM 预测 Grid Computing 是因特网的未来. 计算机世界, 2001. 10. 15
- 20 高全泉. 网格: 面向虚拟组织的资源共享技术. 计算机科学, 2002, 30(1)

(上接第 5 页)

识、确定认证及创制共享的机制必须是灵活的和轻量级的,这样才能快速地建立和改变资源共享安排。由于 VO 与现有的机构/技术相辅相成,而不是取而代之,所以不能要求对本地的策略有本质的改变,并且必须允许单独的机构在他们拥有的资源上保留最终的控制权。由于协议管理组件之间的交互,而不是管理组件的实现,所以保留了本地的控制。

3) 服务是重要的(通过服务向 VO 伙伴提供协议行为)——服务完全由它所讲的协议和它实现的行为定义。对于访问计算、访问数据、资源发现、合作安排、数据备份等行为来说,标准的服务定义允许我们增强提供给 VO 伙伴的服务,并将那些妨碍 VO 应用开发的资源专门的细节抽象掉。

4) API(应用程序设计接口)和 SDK(软件开发包)是协议的重要辅助手段。对于 VO 来说,当然有比互操作性、协议和服务更进一步和更细的要求。对开发者来说,他们必须能够开发在复杂的和动态的执行环境中的精细的应用;对用户来说,他们必须能够操作这些应用。因此,应用的鲁棒性、正确性、开发成本及维护成本都是一些重要的需要关心的问题。标准的抽象,API 以及 SDK 是协议的辅助而不是替代者。如果没有标准的协议,互操作性在 API 层能被完成只能通过以下两种方式完成:或者到处使用一个单独的实现,而这在许多有趣的 VO 中根本行不通;或者通过使得每个实现知道其它实现的细节。

结语 本文对公众关心的网格问题进行了一般性的综述。为了理解网格的产生背景,我们从虚拟组织及它们从事的活动出发,概括了虚拟组织的建立及虚拟组织成员间资源共享的特性和需求。并且指出,因已有技术不具备这些特性,不能满足这类需求,故现实中确实存在一类“网格问题”。本文也给出网格讨论中的基本术语/概念的定义,标识了建立虚拟组织共享所要求的基本特性和需求,概括介绍了网格的基本设计思想及关键理念。这些内容基本回答了在本文引言中提出

的前两个问题。至此,网格的定义可以一言以蔽之,即动态的、可伸缩的虚拟组织的可控且可调的资源共享和资源使用。

Ian Foster 等在文[4]中,对网格研究的领域,关注的主要问题及关键技术作了界定,这些定义、术语/概念、原理、关键技术和体系结构,既提供了网格讨论的基本公共语言和交流基础,也提供了实际可以运用和借鉴的理论框架和参考模型。这对全面准确认识网格是必要的和及时的。对网格深层技术,即网格体系结构的需求和功能描述、网格与同时代其它分布计算技术/中间件的区别与关系(以及网格的不同观点的)讨论,我们另文专述。

参 考 文 献

- 1 The Globus Project, <http://www.globus.org>
- 2 Foster I, Kesselman C. The Globus Project: A Status Report. In: Proc. Heterogeneous Computing Workshop, IEEE Press, 1998. 4~18
- 3 孙定. 从理论到实践——Globus 网格计算理论及其应用. 计算机世界, 2001. 11. 5
- 4 Foster I, Kesselman C, Tuecke S. The Anatomy of the Grid: Enabling Scalable Virtual Organizations. Intl. J. Supercomputer Applications, 2001, 15(3)
- 5 Allock W, et al. Data Management and Transfer in High-Performance Computational Grid Environments. Parallel Computing, 2001
- 6 Linn J. Generic Security Service Application Program Interface Version 2. Update 1, IETF, RFC 2743, 2000. <http://www.ietf.org/rfe/rfe2743>
- 7 Aiken R, et al. Network Policy and Services: A Report of a Workshop on Middleware. IETF, RFC2768, 2000. <http://www.ietf.org/rfc/rfc2768.txt>
- 8 高全泉. 网格: 面向虚拟组织的资源共享技术. 计算机科学, 2003, 30(1)