

基于知识网格的数据挖掘^{*}

魏定国^{1,2} 彭宏²

(华南理工大学计算机学院 广州 510641)¹ (广东商学院 广州 510320)²

摘要 工业、科学、商务等领域的数据库通常分布在不同的地方,需要在不同的地点对其进行分布式维护,只有使用计算功能超强的分布式、并行处理系统才能分析这些领域所产生的超大规模数据集。网格为分布式知识发现应用中的计算提供了有效支持。为了在网格上进行数据挖掘的开发,本文提供了一个称之为知识网格的系统,讨论如何应用知识网格设计实施数据挖掘应用,并说明如何搜索网格资源、编制软件和数据组件,以及数据挖掘应用在网格上的执行过程。

关键词 数据挖掘,知识网格,网格计算,网格调度

Data Mining Based on Knowledge-Grid

WEI Ding-Guo^{1,2} PENG Hong²

(South China University of Technology, Guangzhou 510640)¹ (Guangdong Commerce College, Guangzhou 510640)²

Abstract Since the data in such fields as industry, science and commerce are usually distributed in different places, and distributed maintenance is needed for them in different sites, the super large-scale data sets produced in these fields can be analyzed by only using the distributed and parallel processing systems that have a super strong computing function. The grid offers effective support for the computing in the distributed knowledge discovery applications. For the development of data mining application on grids, this article provides a system called knowledge grid, and discusses how to design and implement data mining applications by using the knowledge grid. This article illustrates in detail how to search for grid resources, compose software and data components, and the process that the mining applications execute on grids.

Keywords Data mining, Knowledge grid, Grid computing, Grid scheduling

1 引言

网格计算是一个被广泛关注的新颖的计算模型。它与传统的分布式计算模型不同,主要着眼于需要大规模资源共享领域的应用,当然,同样适合高性能计算方面的应用。现在网格已经成为高性能计算和分布式处理的一个有效基础。网格是用户能够用简单的接口对其进行访问的、由异构型机器组成的分布式分析计算体系结构。因此,网格通过由公共和个体的共享资源所组成的虚拟组织提供资源访问技术和操作服务。

现在,实施网格应用的工具包和软件环境(如 Globus^[7]、Legion^[8]、Condor^[9]和 Unicore^[10]),已经可以投入商业使用了,特别是 Foster 和 Kesselman 的 Globus 工具包已经广泛应用于科学和数据密集型的网格应用的中间件,已经变成了一个实施网格系统的事实上的标准了。工具包主要解决了安全、信息发现、资源和数据的管理、通信、错误检测和可移动等问题,它通过由服务袋组成的机制来执行网格中的操作。现在, Globus 和其他网格工具已经广泛应用于各种工程项目。尽管这些工程大多数还是科学与技术方面的分析与计算,但网格在教育、工业和商务中的应用也在迅速增长。

数据网格中间件主要集中于网格上的数据移动和复制管理。在很多的科学和商业领域使用工具和软件环境中,对可利用的数据进行分析、推理和发现是非常必要的。科学家与工程师能够应用这些环境进行虚拟科学实验,能够在挖掘巨

大量数据时遵循同样的方法以获得决策支持。因此,数据网格是使用知识网格^[9]提供的工具和模型对网格上的数据资源进行基于知识的分布式挖掘与析取,使自己不断发展进化。

知识网格是一个并行、分布式的、集数据挖掘技术和网格技术于一体的软件体系结构。在知识网格的体系结构中,数据挖掘工具集成了数据网格的所有机制与服务。因此使用知识网格能够对网格上的巨大数据集进行数据挖掘,以实现科学发现,改进工业流程和组织模型,揭示有价值的商业信息。

本文的工作主要是基于这些相关的研究基础,将论述分布式高性能知识发现的应用环境的设计与实施,并讨论基于知识网格的数据挖掘应用的设计与执行过程。

2 知识网格

知识网格体系结构使用基本的网格机制,在网格工具箱和网格服务基础上建立具体的知识发现服务系统。我们能够应用可用的网格环境以不同的方式开发这些服务系统。当今的实施方法是基于 Globus 的工具箱。与 Globus 一样,知识网格以局部服务的协作和联合为基础提供全局的服务。我们设计的知识网格的体系结构使得很多专门化的数据挖掘工具能够在低级的网格机制和数据网格服务上兼容。使用该方法就能够充分利用各种可用的“标准”网格服务资源,保证能够拥有一个开放式的并行、分布式的知识发现体系结构,能够在网格中间件级别上以简单的方式进行配置。

^{*} 本文的工作受广东科技攻关项目(2004A10202001)和广州市科攻关项目(2004Z2-D0091)的资助。魏定国 教授,博士后,主要研究方向为计算机网络技术、数据库和电子商务;彭宏 教授,博士生导师,主要研究方向为智能网络技术、智能商务与数据挖掘、人工智能应用技术。

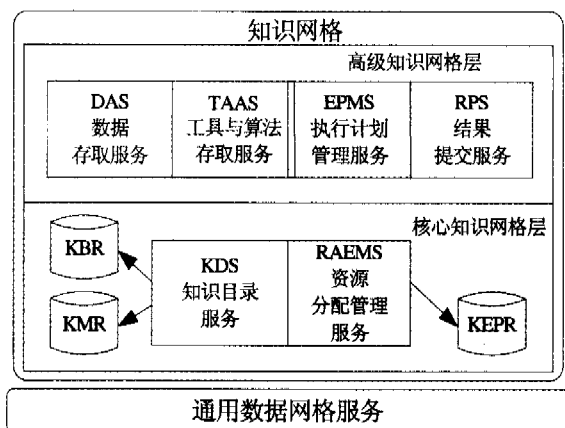


图1 知识网络的体系结构

知识网络由两个层次构成,核心知识网格层和高级知识网格层。前者只涉及一般的网格服务的顶层直接实施的服务,而后者是用于在知识网络上进行设计、组建和执行分布式知识发现计算。图1中给出了两层的结构知识网络的体系结构,并包括知识网络数据仓库和元数据仓库。

2.1 核心知识网格层

核心知识网格层是在网络上为分布式知识发现应用的定义、组建和执行提供最基本的服务。它主要负责对描述数据源特性的元数据、第三方提供的数据挖掘工具、数据管理工具、数据可视化工具和算法进行管理。此外,该层通过试图利用网格资源满足应用需求,以协调应用程序的执行。核心知识网格层主要包含两个主要的服务。

(1) 知识目录服务(KDS)

知识目录服务(KDS)将基本的 Glous 服务扩展到监控与发现服务,并管理描述数据的元数据和在知识网络中应用的工具。这主要包括:

- 被挖掘的数据仓库(数据源)。
- 用于抽取、过滤和操纵数据的工具与算法;用于挖掘数据、可视化和存放挖掘结果的工具。
- 分布式执行计划。一个执行计划是分布数据挖掘应用的一个抽象说明,也就是描述人机交互、数据源之间的数据流、数据挖掘工具、可视化工具和结果存储设备的图示说明。
- 将所获知识作为挖掘处理的结果,也就是学习模型和发现模式。

所有的元数据用可扩展标记语言(XML)文档来说明,并存放在知识元数据仓库(KMR)之中。要维护在一个特定的数据仓库中被挖掘的数据是不切实际的。然而,要维护已经发现的知识库却是可能的,也是非常有用的。这些信息被存放在知识仓库(KBR)之中,其关联的元数据是由知识目录服务(KDS)来管理。因此知识目录服务不但用来搜索和访问原始数据,而且用来查找以前发现了的知识,能够将其用于给定的挖掘计算结果与不同的数据源之间进行比较,或者以递增的方式应用数据挖掘工具。

对知识网络来说,通常可以直接使用现有数据管理、分析和可视化工具(也就是说它们驻留在文件系统或代码库之中)。知识执行计划仓库(KEPR)用来存放数据挖掘处理的执行计划。

(2) 资源分配与执行管理服务(RAEMS)

RAEMS 用于查找执行计划与可用资源之间的一个适当的映射,以满足应用需求(计算能力、存储、内存、数据库、编译器、网络带宽和延迟)和网格约束的目标。在执行计划激活之

后,该服务就管理、协调应用的执行。在当前知识网络实施中,是直接基于 Globus 资源分配管理器(GAMR)的服务,而不再使用知识目录服务和 Globus MDS 服务。每一个数据挖掘作业请求都是用 Globus 资源规格说明语言(RSL)^[4]来表达的。执行计划的分析与处理将生成全局的资源请求,就可以为本地 GRAM 将全局的资源请求依次翻译成 RSL 请求。

2.2 高级知识网格层

高级知识网格层包括用于组建、验证和执行并行、分布式知识发现计算的服务。而且该层还提供存储、分析已发现的知识的的服务。其主要服务如下:

(1) 数据存取服务(DAS)

数据存取服务负责搜索、选择、抽取、转换和交付为挖掘的数据。搜索和选择是基于核心知识目录服务。以用户需求和约束为基础,数据访问服务使用数据挖掘(DM)工具自动(或在用户的帮助下)搜索和查找要分析的数据源。

(2) 工具和算法存取服务(TAAS)

该服务负责搜索、选择和下载数据挖掘工具与算法。还和以前一样,关于它们的可用性、位置和配置的元数据是存放在 KMR 中,由知识目录服务来管理,然而工具与算法却是存放在每一个知识网络结点的本地存储设备中。希望将数据挖掘工具输出到其他用户就必须用知识目录服务发布它们,它们是以元数据的形式存放在 KMR 的本地部分。

(3) 执行计划管理服务(EPMS)

执行计划是用描述交互和数据资源之间的数据流的图形来表示的。在简单的情况下,用户能够直接使用可视化的组建工具设计执行计划,其程序联接到数据资源。然而,由于数据存取服务(DAS)以及工具和算法存取服务(TAAS)产生的结果的多样性,会根据数据和工具的位置,移动的策略和中间结果等产生出不同的执行计划。因此,执行计划管理服务是用半自动工具来实施的,采用的数据和程序由用户的选择生成一个描述所设计的计算的抽象执行计划,映射到具体网格资源(参看 3.4 节)执行计划并存放在知识执行计划仓库中(KEPR)。

(4) 结果表示服务(RPS)

结果的可视化在知识发现处理中是非常重要的,帮助用户理解或解释已发现的模式。存放在知识仓库中之后,这个服务说明如何生成、表示和可视化所析取的知识模型(如关联规则、群集模型和类模型),其元数据存放在由 KDS 管理的知识元仓库(KMR)之中。

3 网格数据挖掘应用的设计

图2给出了在知识网络上分布式知识发现的设计过程中的步骤。这个设计过程是从搜索和选择需要组建应用的资源开始的。这一步是由 DAS 和 DAAS 工具完成的,分析表示参与知识网络结点的可用资源的 XML 元数据文档,并将其存放在它们的 KMR 之中。这种分析试图查找关于可用资源的特殊信息(如实施特殊算法、特殊资源等的软件),其工作的完成是基于用户选择的搜索参数和选择的过滤器。然后将为该计算所选择的资源的元数据存放在任务元数据仓库之(TMR)中,它是一个包含关于选择完成某个计算资源信息(计算结点、数据资源和软件)的本地存储空间。

数据挖掘计算的设计由执行计划管理服务(EPMS)完成。下面是用户建立简单的方式,这就当然需要一个称之为可视化的网格应用环境(VEGA),VEGA 实质是一个工具集。图3是 VEGA 的体系结构,它集成了 EPMS 和其它知识网格服务的功能,应该能够提供以下执行计划管理服务(EPMS)

操作:

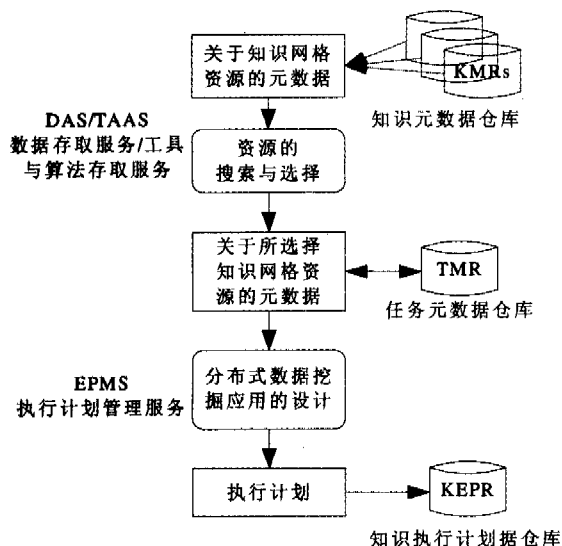


图2 数据挖掘计算的设计过程

- 任务的组建,也就是计算中涉及实体的定义以及它们之间关联的规格说明;
- 计划的计算的一致性检测;
- 执行计划的生成。

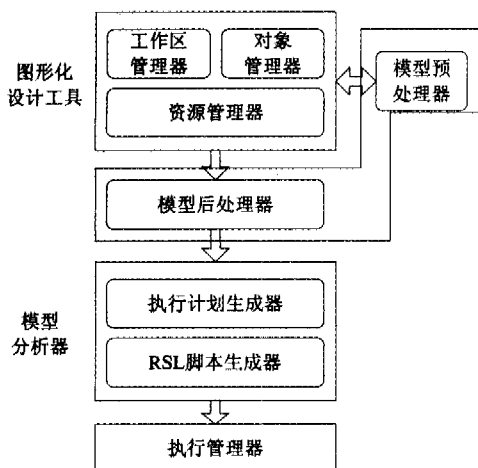


图3 网络应用的可视化环境的体系结构

3.1 任务组建

任务的组建是由图形化接口完成的,它提供给用户表示可应用的网格结点资源(如数据集、数据挖掘工具)的图形化对象的集合。这些对象能够通过可视化工具组建,允许用户在它们之间插入链接,产生代表这个计算的图形表示。

这个阶段可由下列软件组件来完成:资源管理器;对象管理器;工作区管理器。

资源管理器为了搜索和选择用于该计算中的资源,支持任务元数据仓库的浏览。所选择的主机被显示在主机屏幕上,用户能够通过点击其标签检索它们每一个的资源。通过分类,将资源显示到资源屏幕上。

对象管理器在可视化组建的过程中处理图形化对象。每一个图形化对象都与它相关资源的信息相关联;这种信息用于内部模型的创建和执行计划的生成。对象管理器处理着3种类型的对象:数据、软件和主机。它允许用户把主机和资源屏幕中的对象拖动到工作区。然后,用户能够链接这些对象,标明它们之间的交互。在组建阶段,对象涉及几种操作,如将其插入和移动到工作区中、与其他对象联接等等。链接能够

表示不同的动作,如数据传输、程序的执行和输入与输出的关联。对象管理器完成链接的标记,完成其它关联特性的属性说明。数据传输链接是用于移动网格上不同位置的资源,输入和输出链接是用于说明程序的输入和输出数据。对于每一种类型的链接,设置相关的参数是可能的(如协议、数据传输目标路径、执行的作业管理器等)。

一个复杂的计算通常是由几个作业组成的,设计环境是由多个工作区组成的。在给定的工作区中,作业都会试图并发执行,而工作区是顺序执行的,因此在工作区之间优先次序需要进行维护。这个工作是由工作区管理器完成的。

3.2 任务的一致性检测

这一步的目标是为了获得一个正确的、一致性的计算模型。整个验证过程通过两个组件完成:模型预处理器和模型后处理器。

计算模型的预处理是在图形化组建期间完成的。模型预处理检测模型组件的一致性,由于采用上下文敏感控制,只有当它们表示实际执行的动作才允许创建链接。例如,它允许用户在一个软件对象和数据对象之间插入一个输入或输出链接,但它绝对不允许在一个主机对象和一个数据对象之间插入一个执行链接。

检测由模型后的处理器来完成,它负责捕捉预处理阶段不能识别的错误。例如它验证一个工作区至少包含一个主机。

3.3 执行计划的生成

在这个阶段计算模型将翻译成用XML文档表示的执行计划。这个任务由执行计划生成器来完成。

执行计划生成器基本上是一个分析程序,它分析图形化的计算模型,生成等效的XML文档。当调用执行计划生成器时,它依据所涉及资源的特性和链接的参数完成其任务。XML文档执行计划是在高层次上描述的一个数据挖掘计算,既不包含关于资源(用元数据识别)的物理信息,也不包含这些资源的状态和当前的可用性。实际上关于所涉及资源的特殊信息只在RSL脚本中体现出来。

3.4 面向应用的调度

在实际的网格应用中,组建时要具体说明所有的应用需求通常是不可能的。正如前面所说的,我们增加到知识网格程序模型的主机通常是抽象主机,只是定义其可能性,也就是它们的特性只能部分地说明,可以和不同的具体主机相匹配。抽象作业(也就是那些只涉及抽象主机)的具体主机的指派是由一个调度程序(它是资源分配的执行管理服务的一个部分)来完成。这个调度程序的任务是要检测由抽象作业组成的执行计划,并以计算开销和输入与输出开销的预测(知识)为基础,以改善应用性能为目标,进行合理的调度。此外,调度程序能够针对作业的状态和可用资源方面的信息调整相应的调度。

调度程序提供一个开放式的Java接口,允许用户自定义调度策略。此外,用户能够提供以自己的方式估算计算开销与I/O开销,也就是软件组件的计算时间,如输入时间、主机处理时间、传输数据的通信时间等。

调度策略由映射组件实施。映射组件的输入,除抽象执行计划和资源说明外,还包含开销估计功能。映射的输出由抽象作业到主机的分配和计时组成,对每一个作业在应用程序执行时必须启动计时功能进行计时。计算调度可以是部分的,也就是说可包含未分配的作业(即所谓挂起),以后再调度,但必须满足几个严格的要求。首先,资源约束必须满足,也就是说每一个作业必须是真正可以执行的(考虑组成它的软件、数据和主机的特性)。其次,优先约束也必须得到满足,

也就是如果在输入计划中作业 j_i 优先作业 j_k , 那么作业 j_k 的启动时间必须选择在作业 j_i 完成之后(并在其它可能数据移动操作之后)。最后整个完成的时间必须尽可能地少。

映射组件必须处理几个非常具有挑战性的问题。即使假设在完全链接好的资源池中, 不论是处理单元还是网络链接, 所有的资源都具有相同的性能, 资源数量不变。如果要考虑一个以上的主机, 那么产生的问题也是一个 NP 难题。因此, 精确、理想的技术, 如整数线性或约束编程似乎不再适用, 因为它们将导致调度处理持续时间达到指数级别的规模。

映射组件, 先前使用试探的方法以减少搜索空间规模为目标完成预处理阶段。预处理阶段由以下几步组成:

- 删除那些不知输入大小的作业简化抽象计划, 以适合估算。这个简化是切实可行的, 因为处理的是数据密集型的应用。

- 抽象计划还要简化到只包含登录作业(也就在调度时间内其所有输入都就绪的作业)和只依赖它们的作业, 直到某个深度 p 为止。

- 依据相关文献^[12]中的思想, 根据计算能力和网络距离对主机进行“相似”分组。

- 最后, 为了做到切实可行的分配, 通过查看资源说明, 对每一个抽象作业, 找出能在它上面进行调度的主机, 建立一个相应的主机集合。

在完成搜索空间的简化之后, 映射组件应用一个或多个试探方法, 如 Min-Min 试探法、Max-Min 试探法、最小完成时间试探法、机会主义负载平衡法、模拟退火法、禁忌搜索法、遗传算法等。

此外, 为了在资源请求与规格说明之间尽可能找到部分匹配, 当然也为了拥有抽象软件、数据资源等可能性, 我们计划将在知识目录服务 KDS 中融入本体概念。本体可帮助调度任务的实施, 因为在资源请求与规格说明之间找到完全一致通常是困难的。创建的本体能够清晰地描述资源和作业请求, 允许使用本体中定义的术语实现严格的语义匹配。资源规格说明与作业请求可以表示成本体概念, 当一个资源与一个作业规格说明(也就是请求)相匹配时, 就允许说明成匹配规则。因此, 要实现基于本体的资源/请求匹配, 就应当提供描述请求、请求的性质(如请求的所有者)、请求特性(如作业请求的类型)和资源需求的本体。此外, 能捕获资源权限和使用策略(如授权访问指定的计算机系统)的本体也是必需的。我们现在正试图为数据挖掘领域建立本体, 用于数据挖掘软件工具的分类。它将简化在网络的分布式知识发现应用的开发, 为不同类型的数据挖掘任务、方法学、解决给定问题的应用软件提供一个参考模型, 帮助用户查找最合适解决方案。

4 网络数据挖掘应用的执行

图 4 中给出在网络上数据挖掘计算执行步骤。执行计划的优化与翻译是通过资源定位和执行管理(RAEMS)来完成的, 是由 VEGA 组件与调度程序完成的(参看图 3)。

VEAG 当然要集成 RSL 生成器模块, 用于产生资源规格语言脚本, 能够直接提交给运行 Globus 工程箱的网格结点上的资源分配管理器。与 XML 执行计划相反, RSL 脚本能够完全描述所完成计算的实例, 也就是说, 它能说明执行需要的所有物理信息(如资源的名称与位置、软件参数等)。

挖掘计算的执行是通过 VEGA 执行管理模块完成的。通过使用 Globus 网格基本的安全体系结构(GSI)服务系统, 执行管理器允许系统验证用户。为执行挖掘计算, 需要向 Globus GRAM 提交 RSL 脚本。在整个生命周期中, 执行管理器也负责监控组成整个数据挖掘计算的作业。最后, 执行

管理器收集分布数据挖掘计算的结果, 随后将它们传送给结果表示服务(RPS)系统, 再由 RPS 将结果呈现给用户。

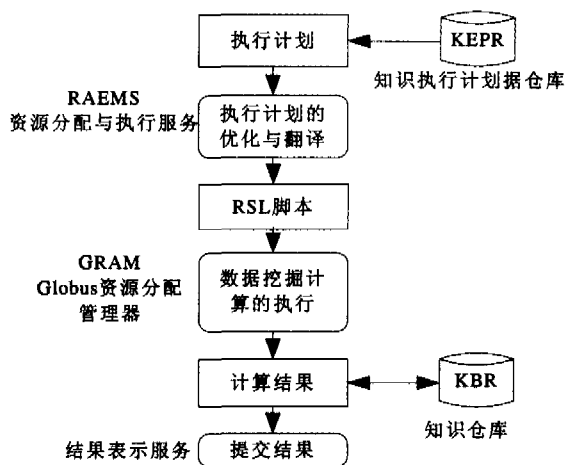


图 4 数据挖掘计算的执行过程

结束语 数据挖掘已广泛应用于大规模企业数据与科学数据的分析与处理。分布式数据挖掘的方法与工具和可计算的网格系统是数据密集实验的科学计算、分布式在线服务的商务活动、虚拟团队和虚拟企业中的虚拟组织支持等高性能计算环境的两个至关重要的元素。

网格的基本结构正在快速增长, 不论是工具数量上还是支持应用的多样性上, 将越来越完善和复杂化。有了知识网格机制, 就能充分地利用网格工具箱和网格服务资源, 建立具体的实用的知识发现服务系统。虽然目前已经有非常多的并行分布式数据挖掘系统, 但很少发现像本文中这样能够充分利用网络上已有的工具与服务资源的系统。

限于篇幅, 本文并没有提供知识网格的实验原型, 但我们已经用实验证明了应用知识网格对分布数据挖掘知识发现应用的实施是能够带来好处的, 很适合面向数据服务和高性能信息管理服务的海量数据处理, 因为数据挖掘应用的执行时间基本上是随网络的结点数据线性减少, 执行的速度是随结点线性增加。

我们相信知识网格是网格系统发展的一个重要组成部分, 因为它支持分布式数据分析与知识发现和知识管理服务, 能够完善数据网格的服务。

参考文献

- 1 Foster C I, Kesselman C, Salisbury C, et al. The data grid: Toward an architecture for the distributed management and analysis of large scientific datasets. *Computing Applications*, 2001, 23: 187~200
- 2 Cannataro M, Talia D. The knowledge grid. *Communication ACM*, 2003, 46(1): 89~93
- 3 Foster I, Kesselman C. Globus, A metacomputing infrastructure toolkit. *Int J Supercomputer Applications*, 1997, 11: 115~128
- 4 <http://www.globus.org/gram/rsl-spec1.html>
- 5 Kwok Y, Ahmad I. Static scheduling algorithms for allocating directed task graphs to multiprocessors. *ACM Computing Surveys*, 1999, 31(4): 406~471
- 6 Cannataro M, Comito C. A data mining ontology for grid programming. In: *Proc. 1st Int Workshop Semantics Peer-to Peer Grid Computing (SemPGrid)*, 2003. 113~134
- 7 <http://www.globus.org/toolkit>
- 8 <http://legion.virginia.edu>
- 9 <http://www.cs.wisc.edu/condor>
- 10 <http://www.unicore.org>
- 11 Hoschek W, Jaen-Martinez J, Samar A, et al. Data management in an international data grid project. In: *Proc. IEEE/ACM Int Workshop Grid Computing Grid*, 2000. 77~90
- 12 Dail H, Casanova H, Berman F. A Modular Scheduling Approach for Grid Application Development Environments, [UCSD Technical Report], CS2002-0698
- 13 刘洁, 郑丽萍. 中国织女星知识网格研究进展. *计算机研究与发展*, 2003, 40(12): 1672~1676