

摘 要

网络是一种新的信息基础设施。在网络计算中，系统资源具有动态性、异构性、通信延迟的不确定性，因此在网络环境里如何有效的管理资源就是网络计算成功与否的重要因素之一。网络资源发现则是网络资源共享的基础，在网络环境下能否高效的发现资源，已经成为衡量网络系统好坏的重要标准之一。资源发现是把资源和资源请求者联系起来的重要环节，有了资源发现机制，请求者才能使用请求的资源，否则，即使在网络上存在大量的资源，请求者也很难找到合适的资源。

论文首先对网络的特点、体系结构进行了简要介绍，并对网络资源的特点、网络资源管理模型进行了说明，总结了集中式资源发现机制和分布式资源发现机制的优点和缺点。在此基础上，本文将集中式资源发现机制和分布式资源发现机制结合起来，提出一种基于分层树型网络资源发现模型，将注册有同类资源的网络信息节点组织在一起形成社区，资源发现请求及转发都被限制在相应的社区内，从而缩小了搜索空间，提高了资源的发现速度，同时也克服了传统资源发现方法的单点失效和网络瓶颈的缺点，改善了资源发现的性能。在资源发现的工作过程中，分析了资源节点加入和离开的情况，给出了资源发现算法，并以模拟实验的形式验证了分层树型网络资源发现方法的高效性。

关键词：网络；资源管理；资源发现；分层树

Abstract

Grid is a new kind of information infrastructure. In grid computing, the system resources have the characteristics of dynamics, heterogeneity and indeterminacy of communication delay. Then, how to manage these resources efficiently becomes one of the most essential factors which affect the eventual success of grid computing. The grid resources discovery is the base of grid resources sharing. Whether discovering resources efficiently in network circumstance or not becomes one of the basic standards to evaluate the grid system. The grid resources discovery stands as an important bridge which links resources requester with resources provider. Only with the help of scheme of resources discovery can requesters use the resources of the grid. Otherwise, requesters have difficulties in finding proper resources even though there are abundant resources in the grid.

First, the characteristics and structure of grid are described briefly. And the characteristics of grid resources and the models of resources management are illustrated in this thesis. The advantages and disadvantages of central and distributed scheme are presented. Then this thesis puts forward a method of grid resources discovery based on hierarchical tree combining the centralized scheme with distributed one. In which information nodes with the same type of registered resources are organized into communities. The requests and forwards of the resources discovery are restricted in the corresponding communities. By which the search space can be reduced and the efficiency of resource discovery can be improved evidently, at the same time, the shortcoming of traditional method of resource discovery include server lose efficiency and Bottleneck of the network can be overcome. In this way the performance of resources discovery method is improved. In the work process of resources discovery, the joint and leaving of the

recourse nodes are analyzed. The resource search algorithm is given. And the high-efficiency of the hierarchical resource discovery method is validated by simulation experimentation.

Key words: Grid; Resources Management; Resource discovery; Hierarchical tree

哈尔滨工程大学 学位论文原创性声明

本人郑重声明：本论文的所有工作，是在导师的指导下，由作者本人独立完成的。有关观点、方法、数据和文献的引用已在文中指出，并与参考文献相对应。除文中已注明引用的内容外，本论文不包含任何其他个人或集体已经公开发表的作品成果。对本文的研究做出重要贡献的个人和集体，均已在文中以明确方式标明。本人完全意识到本声明的法律结果由本人承担。

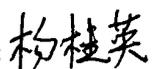
作者（签字）：

日期： 2009年6月14日

哈尔滨工程大学 学位论文授权使用声明

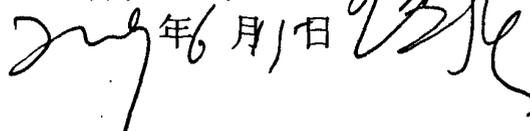
本人完全了解学校保护知识产权的有关规定，即研究生在校攻读学位期间论文工作的知识产权属于哈尔滨工程大学。哈尔滨工程大学有权保留并向国家有关部门或机构送交论文的复印件。本人允许哈尔滨工程大学将论文的部分或全部内容编入有关数据库进行检索，可采用影印、缩印或扫描等复制手段保存和汇编本学位论文，可以公布论文的全部内容。同时本人保证毕业后结合学位论文研究课题再撰写的论文一律注明作者第一署名单位为哈尔滨工程大学。涉密学位论文待解密后适用本声明。

本论文（在授予学位后即可 在授予学位12个月后 解密后）由哈尔滨工程大学送交有关部门进行保存、汇编等。

作者（签字）：

日期： 2009年6月14日

导师（签字）：

 2009年6月14日

第 1 章 绪论

1.1 研究背景

近年来,计算机与网络技术取得飞速进步,具有强大计算能力的巨型机与服务器应用正在普及,高速率、高带宽的网络已经在广域环境中应用,同时,个人计算机功能也日益强大,基于网络的应用,比如Web,已经非常广泛和成熟,在这些技术前提下,为了解决复杂的计算问题和推动新的计算应用,网格计算便应运而生。网格计算^[1](Grid Computing)一般指把通过高速网络连接广域分布的计算设备(大型机到个人计算机,甚至小型信息终端)、存储设备、以及专用外围设备(如天文、气象观测系统,虚拟现实装置)等资源在逻辑上连为一个整体,实现资源共享,向用户提供透明而便利的服务。

为了实现网格的资源共享与融合,需要许多技术的支撑,资源发现^[2]就是其中之一。它主要解决如何根据网格用户资源请求来发现相应资源的问题。资源发现问题的有效解决是网格资源实现共享与融合的前提,同时为网格系统其它中间件的应用执行奠定了基础,如资源监控,资源分配,资源调度等。因此,资源发现机制的研究与实现是网格计算领域的一个重要研究课题。由于网格系统中包含各种各样的资源,这些资源又具有动态变化、广域分布、系统异构等特性。因此,网格环境下资源发现机制的研究难度更大、面临的挑战更多。

1.2 相关领域国内外研究现状

目前,网格技术的研究在全球范围内得到了各研究组织机构的极大重视,许多国家和地区都投入了巨资进行网格技术的研究和网格基础设施建设。由于资源发现是实现有效的网格计算需要解决的重要问题之一。许多网格项目都对其进行了专门的研究,提出了许多资源发现的方法和策略。

网格资源发现方法已经取得了较多的研究成果。其中比较有影响的是Globus的集中式资源发现方法和我国中科院开发的织女星网格的基于路由转

发的资源发现方法。

下面针对当前国内外一些典型的网格项目，综合分析网格资源发现机制及方法的研究现状。

1.2.1 国外的研究状况

Globus^[3]是一个由美国的一些大学、科研院所，政府机构和IBM、Microsoft以及Cisco发起，由美国Argonne国家实验室进行实施的研究项目。它旨在为计算网格搭建最基本的基础设施以及高级网格服务。Globus中的监控和发现服务组件MDS包含两个基本元素：1) 信息提供者 (information Provider) 提供有关单个实体的信息；2) 聚合目录服务 (aggregate Directory services)，搜集、管理、索引由多个信息提供者提供的信息。信息提供者提供有关网格中各资源 (实体) 的信息；而聚合目录则提供了与VO有关的网格资源的特定视图。其中网格资源注册协议 (Grid Resource Registering Protocol, GRRP) 具有资源的发现和查询两种基本功能，资源的发现通过搜索操作进行。MDS并没有提出适应于整个网格范围的资源发现机制。MDS中的聚合目录只适合于提供某个特定VO内的资源信息，各个聚合目录之间尽管可以通过标准的协议进行交互，进行信息复制等操作，但MDS并没有对这些信息结点之间的组织和交互定义通用的协议和规范。

OGSA是由Globus项目组提出的面向服务的网格体系结构。OGSA中的资源发现主要指服务发现。OGSA的基本前提是将一切都视为服务：提供一定功能的可通过网络访问的实体。OGSA中的服务发现是根据服务的属性，如提供的接口、负载等，来确定一组GSH的过程。支持服务发现的网格服务称为注册服务 (registry)，它提供并实现注册GSH和获取注册的GSH信息的接口。OGSI中并未指明各注册服务实例之间该如何交互来支持大规模的资源 (服务) 发现。

Condor将大量地理分布、属于不同所有者的空闲计算资源聚合起来支持高吞吐率的计算。Condor中的计算资源由于只是在空闲时才对Condor可用，

Condor中与资源发现和调度有关的一个重要部件是匹配器 (Matchmaker)，它负责将任务请求和周期性提交上来的资源状况作匹配，并通知匹配的双方进行谈判和协作。匹配器做的匹配只是为相互匹配的任务和资源做一种“介绍”，而不管任务的执行。Condor会周期性重新匹配任务，若发现更好的资源(机器)，则正在进行的任务会被迁移到新的机器上执行。总的来说，Condor中依赖于集中式的匹配器来实现资源发现和任务的调度。

Legion是一个面向对象的元计算环境，试图将包括从PC到并行超级计算机在内的大量主机连接起来构成一个虚拟的计算机。Legion中定义了Collection对象用于存放组成VO的资源的状态信息，其中每一条记录是一组Host对象的属性集合。如果使用“推”(push)的方式搜集对象信息，Legion对象将会把这些信息存放到指定的Collection中，而Collection同样也会从资源处以“拉”(pull)的方式获取数据。用户或其资源发现代理则通过查询Collection来获取信息。Collection支持的查询语言支持典型的字段匹配、语义比较以及布尔组合等操作。由此可见，Legion中的资源发现与Globus中的方法有些相似。用户可以直接与各个实体交互获取资源信息，或者通过集中式的方式(Globus中是聚合目录、而Legion中则是collection)查找所需的资源。

UDDI是Web Service中的服务发现和集成机制。UDDI先后推出了不同的版本，目前广泛使用的是UDDI3.0的版本。UDDI3.0基本上是一个集中式的服务注册和查找机制，其核心是Web Service数据和元数据的表示。一个UDDI注册项提供了分类、建目录和管理Web Service的标准机制，使Web Service能够被发现和使用。UDDI服务器支持多种不同的服务查找方式，如:根据实现的接口来查找Web Service实例；根据已知分类来查找Web Service的提供者；询问一个给定Web Service所采用的安全和传输协议；根据关键词搜索服务等。在UDDI.20中，少数几个UDDI服务器之间采用互为备份的全连接方式。

1.2.2 国内的研究状况

从1995年开始，中国科学院计算机技术研究所就建立了专门的网格研究

队伍，开始研究与网格技术相关的技术，并成功开发了多个版本的信息一体化平台。正在开发的网格服务协议和标准有：网格公共信息协议（GPIP, Grid Public Information Protocol），网格系统信息协议（GSIP, Grid System Information Protocol）和网格计算协议（GCP, Grid Computing Protocol）等。

2001年中科院计算所提出织女星网格计划，该计划包括从低到高的网格操作系统、信息网格、知识网格结构层次。在织女星网格中，资源发现机制主要通过两个技术来解决资源发现的问题：基于资源信息的路由转发的资源定位模型和三层资源（将网格资源分成物理资源、逻辑资源和虚拟资源三个层次）表示模型。资源定位模型主要在资源路由器上实现，是资源发现机制的核心。

1.3 研究的目的是和意义

网格资源发现服务是网格系统的一个基本服务，就是查找网格系统中与应用程序匹配的可用资源，网格资源发现是把资源和资源请求者联系起来的重要环节，有了资源发现机制资源请求者才能使用自己请求的资源，否则，大量的资源放在网格上，请求者不知道自己能够使用哪些资源。由此可见网格资源发现的研究可以说是网格研究的重中之中，只有实现了高效的网格资源的查找才可能实现其它的具体网格技术模块，才有可能实现网格发展的宏伟蓝图。换言之，资源发现功能的强弱，直接决定了网格的使用效率和友好程度。

目前许多网格的项目在资源发现的机制及方法上已做了不少有益的研究工作，但是仍然存在不少问题和限制，比如集中网格资源发现的瓶颈问题和分布式网格资源发现的洪泛问题。因此，对网格环境下资源发现机制及方法进行进一步研究是完全必要的。针对以上问题，本文提出了一种基于分层树型的网格资源发现机制，以适应网格应用开发阶段的新要求。

1.4 论文研究的内容

资源发现是网格计算需要解决的首要问题，虽然许多网格项目在资源发

现方面已经做了不少有益的研究工作，但是仍然存在不少问题和限制。因此，对网格环境中资源发现机制的研究是必要的并且具有重大意义。本文针对网格资源发现机制及方法进行了一定的研究，主要内容如下：

1. 研究现有的网格环境下的资源发现，总结资源发现的各种机制，把分层树引入到网格资源发现。建立基于分层树型算法的网格资源发现模型，在模型中进行社区的划分，使得资源发现限定在所属社区范围内，缩小了搜索空间，提高资源发现的效率，同时为了较好地解决瓶颈问题，在模型的上层设置索引节点，映射下层的信息节点，这样可以在一定程度上实现网格中动态资源的发现。

2. 以分层树网格资源发现模型为基础，详细阐述分层树的网格资源发现方法，给出这一资源发现方法的资源发现过程和算法。

3. 以模拟实验的方式来验证本文所提出的网格资源发现方法，给出实验配置和相关结果。

1.5 论文结构

全论文共分4章，其中详细介绍了网格、网格资源发现理论，并对网格环境下的资源发现方法进行了一定研究，全文各章内容安排如下：

第1章，介绍了本论文的研究背景和国内外在网格资源发现方面的研究现状以及研究的目的、范围和意义，并提出了本论文要研究的内容。

第2章，介绍了网格的基本理论，针对网格的一些特点、网格资源的定义和特点、以及网格资源管理等进行了详细的阐述。

第3章，给出分层树型网格资源发现模型，并对这一模型进行了详细论述。

第4章，提出了一种分层树型网格资源发现算法，给出了实验仿真结果对比分析。

第2章 网格资源发现的相关理论

为了更好的展开对分层树型网格资源发现的研究，在论述分层树型网格资源发现方法之前，本章先对网格资源发现涉及到的相关基础理论进行一下简要介绍。

2.1 网格的定义及特点

网格计算系统是一种无缝、集成的计算和协作环境。网格的目的^[4]是利用互联网把分散在不同地理位置的电脑组织成一台虚拟的超级计算机，实现计算资源、存储资源、数据资源、信息资源、软件资源、通信资源、贵重科学仪器资源、知识资源、专家资源等的全面共享与协调管理，其中每一台参与的计算机就是一个结点。传统互联网实现了计算机硬件间的连通，Web实现了网页的连通，而网格试图实现互联网上所有资源的全面连通。整个网络如同一台巨大无比的计算机，向每个用户提供一体化的服务。

简单地讲，网格就是将位于全球不同地方的研究机构所拥有的硬件和软件设施联合起来以解决极端复杂的科学研究和计算问题。在硬件方面，网格需要将地理上分布在不同地方的硬件资源，如网络、计算机、数据仓库以及可视化虚拟现实显示等有机地联系起来。在软件方面网格的作用就象一个中间件，它将需要的各个硬件和软件资源有机地集成起来以便这些不同的部件能够协调运作。

网格的特点^[5]包括：

1. 分布性——网格上的各类资源（硬资源和软资源）通常类型复杂、规模较大、跨越地理范围较广。
2. 共享性——尽管网格资源是分布的，但是它们却是可以充分共享的。
3. 自相似性——网格的局部和整体之间存在着一定的相似性，局部往往在许多地方具有全局的某些特征，而全局的特征在局部也有一定的体现。
4. 动态性——随着时间的推移，网格拥有的资源或功能可以增加或减少。

5. 异构性——网格可以包含多种异构资源，包括跨越地理分布的多个管理域。

6. 多级管理域——网格系统的资源通常属于不同的机构或组织，并且使用不同的安全机制，因此既需要各个机构或组织对其拥有的资源有自主的管理能力，又需要他们共同参与解决多级管理域的问题，以实现资源共享和互操作。

2.1.1 网格与其它网络技术的比较

网格技术与当前存在的一些网络技术如万维网、分布式计算技术、P2P技术等解决问题的着重点上有所区别。

1. 网格与万维网

网格实际上是继传统因特网、万维网之后的第三代因特网应用。传统因特网实现了计算机硬件的联通及相互之间的通信，万维网使用统一的超文本协议实现了网页的连通，仍然停留在信息交换的阶段。而网格试图实现互联网上所有资源的全面联通，包括计算资源、存储资源、通信资源、软件资源、信息资源、知识资源等等，最终实现网络虚拟环境上的资源共享和协同工作，消除信息孤岛和资源孤岛。网格的目标就是让人们使用网络资源像用电一样简单。

2. 网格计算与分布式计算

网格计算的核心问题是资源共享，属于分布式计算领域研究的范畴，它们在许多方面具有共同点，但由于网格计算强调资源共享和不同资源间的互操作性，而各种分布式计算技术更强调为用户提供一个逻辑或虚拟的集中环境。因此，网格计算区别于一般的分布式计算技术^[6]。

首先，网格中资源共享是大规模的，资源类型具有普遍性。网格下的一个资源结点可以代表文件服务器、数据库、程序、传感器等种种连接于网络的软硬件设施。而目前的分布式计算技术处理的资源类型十分有限。Web发布，B2B exchange 以及 virtual enterprise 技术主要解决信息资源的共享，但是

并没有涉及应用程序的绑定调用和远端物理设备（如显示设备和传感器）的控制使用等问题。

其次，网格下的关系更加灵活，充分考虑到了网格结点的自治性。虚拟组织（VO）具有开放的体系结构和灵活的共享机制和访问机制，对于参与成员没有过多的限制。与之相比，现有的分布式计算技术对于计算单元的种种限制，束缚了它们的灵活性和扩展性。企业分布式计算技术，如 CORBA 和企业级 Java 仅允许同一组织内部的资源共享。SSP（Storage service Provider）允许不同组织间的存储资源和计算资源的共享，但是必须加以一定的条件约束，比如 SSP 要求用户必须通过 VPN（Virtual Private Network）与服务提供者进行连接。

分布式计算本质上是一种网络环境下的资源协同共享，因此这些技术都可以运用网格概念加以解释，使用网格技术实现。

2.1.2 网格体系结构

网格体系结构给出了网格的基本组成与功能，描述了网格各组成部分的关系以及它们的集成方式和方法以及支持网格有效运转的机制。

1. 五层沙漏模型

最经典的网格体系结构是 Ian Foster 等 2001 年提出的五层网格体系结构^[7]，五层沙漏结构的五层从下到上分别是构造层，连接层，资源层，汇聚层，以及应用层（如图 2.1）。

构造层（fabric）：是网格中可以被共享的资源所在的层。构造层面对的是一系列具体的物理（也可以是逻辑的）资源，包括计算设备、目录、文件系统、计算机群集、网络、贵重仪器等其它设备。网格构造层的基本功能就是控制本地的资源，向上提供访问这些资源的接口。构造层应该实现的基本功能包括：查询机制（发现资源的结构和状态等信息）、控制服务质量的资源管理能力等。

连接层（connectivity）：是网格中处理通信与授权控制的核心协议层，支

持便利安全的通信。其基本功能就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。它定义了核心的通信和认证协议，用于网格的网络事务处理之中。通信协议允许在构造层资源之间交换数据，要求包括传输，路由，命名等功能。

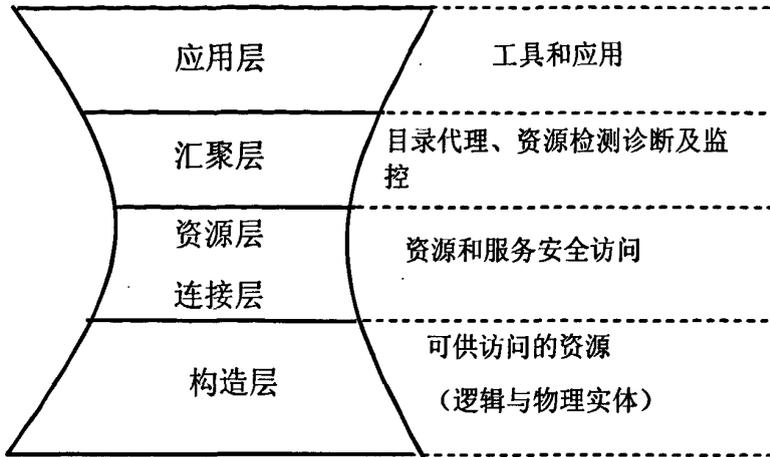


图2.1 沙漏模型

资源层 (resource)：主要功能就是实现对单个资源的共享控制。资源层建立在连接层的通信和认证协议之上，定义的协议包括安全初始化、监视、控制单个资源的共享操作、审计以及付费等。

汇聚层 (collective)：协调各种资源。其主要功能是协调多种资源的共享，将资源层提交的受控资源汇集在一起供应用程序使用。汇聚层协议与服务描述的是资源的共性，并不涉及资源的具体特性，说明不同资源集合之间是怎样交互的。汇聚层在资源层通用协议基础上实现更高级的应用，并能够面向特定领域，这些功能或者协议可以只在特定的虚拟组织中存在。

应用层 (application)：包括用户代码和网格调用两部分，和资源的距离最远，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。应用层是在虚拟组织环境中存在的。

这种模型的一个显著特点就是沙漏形状——“两头大，中间细”，突出体现了用中间较细的部分把两头较大的部分连接为一个整体的特点。其内在含义就是因为各部分协议的数量是不同的，对于其最核心的部分（沙漏的狭窄部

分), 要能够实现上层各种协议向核心协议的映射, 同时实现核心协议向下层其它各种协议的映射, 核心协议在所有支持网格计算的地点都应该得到支持, 因此核心协议的数量不应该太多, 这样核心协议就形成了协议层次结构中的一个瓶颈, 在五层结构中, 资源层和连接层共同组成这一核心的瓶颈部分, 因此这个协议集合要小, 而且尽量标准化。这些协议要能够抓住涵盖不同资源类型的基本共享机制, 但是又不能够对高层协议的类型和性能有约束。其好处是一个很细的中间核心部分, 把大量的两端对象连通, 避免了用不同的方法把两端不同的对象连接起来需要付出的大量工作和繁重劳动。在减少连通工作量的同时, 有利于防止出现错误, 也有利于系统的调试和修改。

2. 面向Web服务的网格模型

2002年2月, 在加拿大多伦多市召开的全球网格论坛GGF会议上, Globus项目组和IBM共同倡议了一个全新的网格标准OGSA。OGSA全称为开放网格服务体系结构^[8], 它把Globus标准与以商用为主的Web Services的标准结合起来, 网格服务统一以Services的方式对外界提供。2003年符合OGSA规范的Globus Toolkit3.0 (GT3) 发布, 这标志着OGSA已经从一种理念、一种体系结构, 走到付诸实践的阶段。GT3提供了一个完整的开放网格服务基础设施(OGSI)实现, 它的许多功能重构成与OGSI兼容的服务。GT3将服务发现、程序执行作业的提交、监控和可靠的文件传输, 定义成与OGSI兼容的服务。其它如数据传递、副本定位和授权等服务也尽量构建成与OGSI相兼容。并且GT3定义了一组关于使用Web服务描述语言(WSDL)和扩展标识语言(XML)模式的约定与扩展, 以便启用有状态服务。虽然OGSI的概念很重要, 但是也存在一些自身的缺陷, 需要一些新的结构来解决这些问题。所以采用新的结构代替OGSI是有必要的, 通过转变可以获得网络服务强有力的支持。

2004年1月, 美国Akamai Technologies、美国The Globus Alliance、惠普、IBM、美国Sonic Software和美国TIBCO Software六公司公布了统一网格计算和Web服务的新标准“WS-Notification”和“WS-Resource Framework”。Web服务资源框架(WSRF)是OGSI的重构和发展。WSRF基本保留了OGSI中的

所有功能,同时更改了一些语法,并且还在其表示中采用了不同的技术。Web服务通知(WSN)为Web服务提供基于消息发布和预定的能力。WSRF和WSN都是建立在已存在的Web服务定义和技术基础上的,帮助实现了网格计算、系统管理和Web服务的统一。2005年1月31日发布的Globus Toolkit4(GT4),实现了WSRF和WSN标准。GT4提供API来构建有状态的Web服务,其目标是建立分布式异构计算环境。所有知名的GT3协议都被重新设计为可以使用WSRF。并且GT4也在其中增添了一些新的Web服务的组件。

2.2 网格资源

资源是网格中所有可以被主体请求使用的实体的总称。网格中包含有多种形式、功能各异的资源,除了万维网上传统意义上的资源之外,还包括天文望远镜、电子显微镜、环境监测探测器、无线通信设备等。

2.2.1 网格资源的定义、特点及分类

网格资源^[9]指所有能够通过网格远程使用的实体,包括计算机软件、计算机硬件、设备和仪器等。计算机软件资源包括系统软件、应用数学、数据等;计算机硬件资源包括处理器、存储器、硬盘以及其它计算机设施;设备和仪器包含通信介质、天文望远镜、显微镜、传感器等。除了这些物理资源之外,在网格设备旁工作的人员也是一种重要的资源,为了和其它资源加以区别,称为人类资源。人类资源包括人的知识、能力等多种因素,是网络上最具有伸缩性的资源。

网格的资源具有面向用户和透明性的特点,用户可以在不考虑资源物理位置的情况下,方便的使用资源。此外,资源也具有动态演化的特性。总的来说,网格资源有如下特点:

1. 异构性:资源工作在异构平台上,并且由不同的管理策略控制。资源种类繁多,功能各异,访问接口也不尽相同,本地管理系统不同,共享规则不同。

2. 动态性:包括资源属性的变化,以及在网格内的复制和迁移等。网格

中的资源可以自由随时地加入和离开网络系统，网络资源的可获得性、贡献给网格用户使用的能力、以及它的负载都是随时间而动态变化的。

3. 广域性。资源的地理分布极广，资源类型和数量巨大，而且要求一定程度的协同工作。资源之间、资源和客户以及客户之间往往通过广域网（如 Internet）连接。

4. 自治性:网络资源有自己的本地管理机构或处在本地管理机构的管理之下，有着或强或弱的本地自治能力。网络管理系统必须尊重本地管理者，服从本地管理策略，不能把网格的意愿加到资源上，更不能替换本地管理系统。

5. 二分特性:除了少部分网格资源是专门提供给网格用户使用的之外，大部分的资源都是被网格用户和本地用户共享使用的，本地用户对资源拥有绝对的控制权，网格用户不能损害本地用户的利益。

资源的种类很多，功能差异也很大。可以从不同的角度将资源分成不同的类别^[9]。

1. 根据资源能否移动的特性可将资源分为可移动资源和不可移动资源。可移动资源包括数据、程序、代码等，它们可以从网格中的一个位置移动到另一个位置。不可移动资源是硬件、设备等不使用机械力量，仅仅使用网络的命令、操作等无法移动的资源。可移动资源会导致备份管理、一致性等许多不可移动资源所没有的问题。在网格中，可移动资源的合理移动是提高网格性能的一种有效途径，资源的移动管理是网格资源管理的基本功能。

2. 根据资源是否可重复使用的特性可将资源分为可重复使用的资源和不可重复使用的资源。可以被多次使用的资源是可重复使用的资源，如内存、存储介质等是可重复使用的资源，一个用户使用之后别人还可以继续使用它们。不可重复使用的资源是计算周期、通信带宽等资源，这种资源一旦被一个用户使用，将会消耗掉它，别人不能再使用了。不可重复使用的资源是网格中最重要的资源，必须合理、有效地利用这种资源，才能发挥网格的能力。

3. 根据资源是否可复制的特性可将网格资源分为可复制资源和不可复制资源，可复制资源可以通过指令或服务请求，把一份资源变成多份，用同一

个名字或相同的属性访问该类资源，可以找到多个相同的物理实体。网络上的数据、应用程序、服务等是可复制资源。不可复制资源不能通过指令或服务请求产生副本，设备、仪器属于不可复制资源。按名称访问不可复制资源，找到的物理实体是唯一的。

4. 根据资源的变化频度，可以分为静态资源和动态资源。静态资源如：本地节点名、主机名、主机地址、操作系统名等等，它们的变化频率比较小。动态资源，如：本地内存利用率、CPU利用率、硬盘利用率等等，它们的变化频率比较大。我们将动态资源本地化存储，而将静态资源集中化存储。

2.2.2 网络资源的描述

网络资源描述是资源发现的基础，通过某种资源描述方法来定义资源的属性信息，作为资源发现的输入输出数据。网络资源描述是资源属性描述的集合，资源描述可包含零个或多个资源属性。

网络资源管理的核心目的就是要在资源提供者和资源请求者之间建立一种共同的协议，通过该协议资源提供者同意提供用于代表资源请求者执行某项任务的能力。资源请求者需要根据资源的性质、数量或配置等来申请资源，这些需求都通过资源描述语言来表达。因此，网络资源描述技术是网络资源管理技术的基础。网络资源描述语言需要提供以下最基本的功能：

1. 使得资源请求者能够描述他所期望的资源能力；
2. 能让资源提供者描述他所能提供的资源的能力，以及限制条件。

可见，资源描述有两种不同的作用，一种是联系资源请求者和资源中介者，请求者用资源描述机制描述自己的资源需求，中介者根据描述做出响应；一种是联系资源和资源中介者，资源把描述自己的信息告诉中介者。合理的描述资源是资源管理的前提。因此，一般的资源描述语言通常包括如下基本的资源描述原语：

1. 参数化的资源属性度量，它用来描述资源的特定性质。这些度量包括：带宽、延迟、空间等。

2. 资源组合操作符,它允许将复杂资源表示为简单的资源组合。这些操作符可以包括“与”、“交”、“异或”等传统的集合运算符,以及描述相同元素集合的“数组”、命名特定资源集的集合操作符。

3. 资源生命周期度量:复合资源实体要求在某一特定时间间隔内保持有效,复合实体中的每个子实体的生命周期都应该满足复合资源实体的生命周期。

除了上述基本的资源描述原语外,高级的资源描述语言还包括有资源控制、配置等原语,这样可以方便的将资源描述和资源管理有机的结合起来。

网格资源描述可以分为3类:资源描述、请求描述、响应回答描述。资源描述由登记在网格资源注册服务器中的所有资源属性组成;请求描述用于描述用户的资源服务需求;响应回答描述由资源请求描述和资源描述的并集组成,它是资源请求的最终结果。

目前,大多数网格系统中的资源描述方法都是专有的,它们都定义自己的资源描述框架或资源描述语言。下面简单介绍几种典型的、使用广泛的资源描述方法^[10]。

1. RSL描述方式(资源规范语言 Resource Specification Language),是 Globus设计的一种资源描述语言,为不同平台的资源描述提供一种公共的描述交换语言。它是在早期网格环境中广泛使用的描述资源的通用可交换语言,它提供了一个框架性的语法描述,可用来组成复杂的资源描述。Globus资源管理框架中的不同组件之间通过RSL的描述完成管理功能,并与系统中的其他组件协调地工作。RSL语法的核心是关系。关系由属性名字和它的值组成。提供一个可以描述复杂资源的语法框架RSL的基本语法是关系,通过关系把属性和值联系起来。源管理组件采用这种特定的<attribute, value>属性/值对通用结构,每个属性/值对作为控制参数以控制对资源的各种操作。

2. WSDL描述方式(Web Service Description Language),是Web Service发展过程中提出的服务描述语言,由微软、IBM首先联合制定,并由W3C作了标准化。它是一种用于描述Web服务的XML语言,用于提高分布式应用之间的互操作性。WSDL将Web服务描述为一组对消息进行操作的网络端点。

WSDL保持协议中立。一个WSDL服务描述包含对一组操作和消息的一个抽象定义，绑定到这些操作和消息的一个具体协议，和这个绑定的一个网络端点规范。通过检查WSDL，有可能推断在客户机和服务器之间的网络上流动的请求和应答消息。WSDL描述了Web服务的三个基本属性：

- 1) 服务做些什么——服务所提供的操作（方法）。
- 2) 如何访问服务——和服务交互的数据格式以及必要协议。
- 3) 服务位于何处——协议相关的地址，如URL。

用WSDL描述网络资源的优点:可以描述复杂的资源，具有良好的可扩展性，技术比较成熟，并且已经是一个工业标准。但互操作性、可用性、安全性等方面有待完善。

3. RDF描述方式（资源描述框架RDF Resource Description Framework）是W3C制定的基于XML语言的一个描述资源的推荐标准，用于表示web上信息的框架。其目的是利用当前存在的多种元数据标准来描述各种网络资源，形成人机可读的并可由计算机自动处理的文件（即提供一个供多种元数据标准共存的框架）。在RDF中所有在Web上被命名、具有URI（Unified Resource Identifier 统一资源标识符，）的东西，如网页、XML文档中的元素等，都可以看作资源，所有URI都具有共同的特征：即不同的人或组织可以彼此独立地创建并使用URI来标识事物。RDF定义了一种与资源无关的通用模型，即框架（Framework），通过这种“资源——属性——值”的三元组，来描述Web上的各种资源。它描述和处理的是元数据。凡是被RDF描述的事物都叫资源，一个RDF文件包含多个资源描述，而一个资源描述是由多个语句构成，每个语句又是由资源、属性类型、值构成的三元组。RDF具有简单、通用、易扩展、开放性、易交换、易综合等特点。

2.2.3 网络资源的存储

1. 基于XML的信息存储：XML本质上是一种描述性的标记语言，是一种更好地表示信息的手段。同时，这也意味着XML可以是用于信息存储和管

理的一种方式，正如同传统的数据库技术一样。XML文件就是数据库，它是数据的集合。在许多方面看起来它和其他文件没什么区别。无论如何，每个文件都含有某种类型的数据。作为一种“数据库”格式，XML有一些优势：例如，它是自描述的（所用的标记描述了数据的结构和类型，尽管缺乏语义），可交换的，能够以树型或图形结构描述数据。

XML提供了许多数据库所具备的东西：存储（XML文档），模式（DTD，XML schema等等），查询语言（XQuery，XPath，XQL，XML-QL等等），编程接口（SAX，DOM，JDOM）等等。而且它在可扩展性、可移植性和结构性等方面都有突出优点，能够很好的满足网络对信息管理和信息交换的需求。

2. 基于LDAP目录服务的信息存储：目录是一种专门被优化用于执行读、浏览、搜索等操作的数据库，可以包含网络以及在网络上运行的应用程序所需的信息。目录倾向于包含具有描述性的、基于属性的信息，并且支持高度复杂的过滤搜索功能。最初的网格信息服务系统基于LDAP的信息模型构建。迄今为止，LDAP依然是网格信息服务实现方案的主要支撑技术。这方面的代表性成果为Globus项目组设计开发的MDS（Metacomputing Directory services/Monitoring and Discovering services），它属于层次模型的网格信息服务，MDS是Globus工具包的核心组件之一。在实现上，MDS主要使用LDAP（Lightweight Directory Access Protocol）作为网格信息访问与存储的统一界面，将网格中的资源信息存储在LDAP目录中。此外，MDS提供了一个可配置的信息提供者（information provider）组件，称为GRIS（Grid Resource Information Service）（它基于LDAP协议，提供资源的相关信息）和一个可配置的集合目录组件，称为 GIIS（Grid Index Information Service）。

3. 基于RDBMS关系数据库的信息存储：关系型数据库管理系统（RDBMS）由于其能够表达复杂对象之间的关系和支持复杂信息查询操作而被一些网格研究团体采用为网格信息服务的基础结构，他们坚信基于RDBMS的网格信息服务系统会更适合于未来网格技术的发展，尽管目前还没有重要的基于RDBMS的网格信息服务软件包出现。基于RDBMS的关系数据库的信

息存储具有如下的优点:基于RDBMS的网格信息服务系统在实现时要求尽可能地不改变传统的关系数据库管理系统,保持其原子性、一致性、独立性和持久性。关系数据模型应用于网格环境的缺点是分布式管理能力较弱,因为它不像层次数据模型那样很自然地按照管理域对网格信息进行组织,例如在LDAP的层次模型中,可以利用其委托机制对某一子“树”进行管理和访问。传统的RDBMS并没有专门针对分布式管理域来设计,甚至没有多个管理域的概念。这个问题正在研究和讨论之中。

2.3 网络资源管理

资源管理^[11]关注的不是资源和服务的核心功能(即资源能够为用户做什么),而是控制网络资源和服务怎样向用户、应用或服务在内的其他实体提供可用能力的一系列操作,即资源的功能执行方式,如被请求的操作何时开始执行,或者它需要多长时间完成等。与集群系统相比,网络的资源管理系统面临着不同的环境。网络在广域范围内分布于多个域,规模庞大,资源种类多且异构性更强,拥有非常多的用户,而且网络中同时拥有共享的和私有的资源。网络的这些特点导致了网络资源管理比先前的集群等系统的资源管理更为复杂。

网络资源的特点决定了网络资源管理系统应当隐藏异构性,为用户提供统一的访问接口;要屏蔽动态性,保证用户使用的质量;要尊重资源的本地管理机制和策略;要仔细审查网格用户的请求,确保网格资源的安全和资源拥有者的权益。

2.3.1 网络资源管理的任务

网络资源管理的任务就是把网格中分散的各种资源管理起来,使多个资源请求者可以共享使用网格中的同一个资源,资源请求者可以根据业务需要同时或先后使用网格中的多个资源,而不需要资源请求者付出额外的劳动。资源管理的目的包括以下三个:

1. 为用户提供访问资源的简单接口。资源管理模块隐藏资源实际使用的

复杂技术细节，将物理资源抽象为逻辑资源，并向用户提供。

2. 协调资源的共享使用。资源管理模块采用排队策略、分时共享策略或其他策略决定多个请求者如何使用同一个资源，这些策略是根据资源本身的特性和拥有者制定的策略确定。同时资源管理模块还应支持一个请求者请求使用多个资源的需要。

3. 代替请求者去使用资源，并建立安全的网格资源使用机制。资源管理器作为超级用户，代替网格用户在资源上进行工作，用户请求时，资源管理器为该用户在资源本地建立一个进行活动的场所——用户容器，用户在容器内使用资源。容器严格定义了用户拥有的权限和可进行的操作等。通过这种方式避免了多个网格用户在资源的同一个本地账号下活动的隐患。容器可在请求时动态地建立并在请求结束后撤销，同一资源上用户容器的数目很少，不会带来大的管理负担。

2.3.2 网格资源管理的基本操作

网格资源管理的基本操作^[12]有以下七个，实际的资源管理可以在这些基本操作的基础上提供更复杂的操作。

1. 资源信息收集：资源管理器主要收集和存储两类信息。一类是资源在加入网格时报告自己的相关信息，如资源名称、类型、拥有者信息等，资源管理器将其记录下来，供使用该资源的应用或用户使用。另一类是网格内动态产生的资源信息，如资源使用情况等。

2. 资源信息更新：资源信息经常会随时间而变化，如可用CPU数目的变动、资源负载、使用情况等。资源管理器周期性地更新这些信息，以免过期信息造成资源使用故障。信息更新频率的确定至关重要，频繁的更新可以及时反映资源的实际信息，但会增加通信的负担。

3. 资源发现：资源发现是资源拥有者和资源请求者之间的纽带，通过该机制，资源请求者才能从数目巨大的资源中发现并使用自己请求的资源。资源发现机制根据资源请求者的资源请求描述，从网格上为请求者找到满足该

描述要求的合适资源，并返回该资源的唯一标识符。本论文重点研究资源发现，所以会在后面重点加以介绍。

4. 资源分配: 资源分配的依据是作业提交者用作业描述语言声明的参数，以及资源拥有者对资源使用所制定的策略。在拥有多资源和多用户的动态网格环境中，资源的分配需要考虑以下两种情况，一是如何从多个可用的资源中选择合适的一个或多个资源分配给请求用户使用，二是如何从请求同一资源的众多请求者中决定哪个或哪些允许使用资源。通常而言，资源分配的输出结果是请求者的作业与资源的匹配关系，以及使用资源的时段、资源能力（CPU数目、存储空间、软件使用许可证数目等）、使用权限等等。

5. 资源定位: 资源定位是根据资源的属性描述获得相应资源物理地址的过程。网格中每个资源都有唯一的物理地址，用户通过该地址实现对资源的访问，但该地址是供机器使用的，不易被人理解。实际上，用户使用网格资源时，不需要知道物理地址，而是用属性描述的方式指定所需资源，并把描述提交给网格，网格中的转换机制再把资源的属性描述转换成用户可以访问的资源的实际物理地址。

6. 资源迁移: 资源迁移是可移动网格资源从一个位置移动到另一个位置的过程，包括服务迁移、作业迁移、数据迁移、软件迁移等。资源迁移的依据是资源的使用情况和网格中可承载该资源的节点运行情况。资源迁移的目的是提高用户访问资源的速度与效率，以及网格负载的平衡。

7. 资源预约: 资源预约是资源请求者在正式使用资源之前，向资源拥有者请求其使用时段内把所需资源预约给自己使用，并保证所需的服务质量。具体的预约请求包括CPU数目、存储空间、网络带宽、软件使用许可证数目等等。资源预约可分为提前预约和立即预约两种，提前预约是预约时间在开始使用时间之前的预约，立即预约的预约时间等于开始使用时间，即预约后马上使用。按预约资源数目分类，资源预约包括单资源预约和多资源联合预约，多资源联合预约是常见的形式，它需要对用户应用所需的多个资源，如远程数据传输所需的存储空间和网络传输带宽，全部进行预约。

2.3.3 现有的资源管理模型

资源管理的结构模型主要依赖于要调度的作业和管理的资源的数量，也依赖于资源处在单个域中还是处在多个域中。目前网格资源管理结构主要有如下几种模型^[13]：

1. 集中式的资源管理方法：如 Globus 计算网格中的 MDS 实现了基于 LDAP 的树状元数据目录服务；Condor 实现了不依赖全局资源命名，而依靠属性匹配的集中式的资源共享系统；Web 服务中的 UDDI 实现了集中式的服务实体的统一描述、注册和发现；但这种集中式的方法是无法满足全球性的 Web 服务的应用。在实践中，目前只有 IBM 公司和微软公司提供 UDDI 商业注册主机服务，且只有极少数企业登记过黄页信息（Web 服务），也就是说，它其实并没有真正意义上的实现互联网服务。

2. 采用大范围的 P2P 系统来解决资源发现的方法，P2P 环境中常采用类分布式资源定位方法：每个结点存储整个资源 ID 空间的一个子空间，并负责本子空间内的资源 ID 到其物理位置的映射，结点间通过特定的协议维护状态和转发查询请求，典型的代表有 pastry 等；但由于全球范围 Web 服务资源的复杂性，服务资源的一致性维护与查找值得进一步研究。

3. 采用目前 Internet 广泛使用的路由机制，只是在这种路由机制中路由器存储与转发的是资源信息（或资源的抽象 ID 号），这种思想借鉴了当前 Internet 的 IP 路由的成功机制，但采用资源路由信息来解决资源查找与组织方式也存在很大的不足之处：首先，IP 路由与资源路由相差太大，IP 是有组织分配的，是唯一的，不会动态产生；而资源是无组织的，重复的，且动态产生与消失。其次，IP 路由能有效地人为组织，按地理（或物理）区域划分，而 Web 服务资源是不行的。因此要得到真正的运用还有许多工作值得研究。

2.4 资源发现

现有的资源发现机制主要是通过一定方法组织和分布资源信息使之形成一个资源信息空间，并在这个信息空间内通过相应的资源发现方法来实现资

源发现, 本节从资源信息空间构造(资源信息组织、信息结点分布拓扑)以及资源发现方法的角度探讨资源发现机制, 并对每种机制的资源信息空间的特点以及在其中查询路由方式进行总结分析。

2.4.1 资源发现的定义

网络应该为用户提供一种功能, 即网络资源发现功能, 它能够根据用户的请求从网络资源中找到满足用户请求的资源。

资源发现是根据资源请求者的资源请求描述, 从网络上为请求者找到满足请求描述要求的合适资源的过程。资源发现可以形式化地描述为一个函数^[15]:

$$S_{id}=F(S_{req})$$

该函数的输入变量是资源需求, 输出变量是一个资源的唯一标识符。函数F是资源发现功能的表示。

资源发现将网络中不被用户所知道的资源和请求使用资源的用户联系起来。资源发现功能的强弱, 直接决定了网络的使用效率和友好程度。

对于资源发现有着不同的定义, 给出详细的定义很困难, 但网络环境中一个比较完整的资源发现机制, 一般应回答以下五个方面的问题:

1. 成员协议: 负责收集和更新资源和网络结点的信息, 约定资源或结点以何种方式加入网络, 如何获取其他结点或资源的信息, 并与之通信。

2. 覆盖构造: 负责从已知成员中选取最优的, 并构成结点覆盖网络的拓扑结构。在实际应用中, 构造覆盖的策略是多种多样的。

3. 预处理: 为获得更好的搜索性能而预先采取的措施。这些措施是独立于资源请求的, 由请求处理完成最终的匹配和定位。因此, 对资源信息的缓存不是预处理, 而是预取。

4. 请求处理, 包括本地处理和远程处理两个部分:

- (1) 本地处理是在本地存储的信息中对请求进行匹配, 根据特定策略对请求进行处理。

(2) 远程处理是指请求转发的规则：根据不同的策略将请求发送到其他的结点上。

5. 资源表示方式：结出用户请求和资源信息的格式。

资源发现是网格计算的一个非常重要的方面，通过资源发现机制将返回一组与预想的资源描述相匹配的资源。

2.4.2 资源发现方法

资源发现方法可分为两大类，主动发现和被动发现，根本区别在于资源提供者和资源发现者之间的交互关系。在主动发现过程中，资源发现者通过发送广播消息进行探询，在一定范围内的资源收到该消息后发送回应，通知发现者其当前的属性信息；被动发现过程则由资源提供者通过注册/注销机制通知发现者其状态的变化。

网格资源发现方法基本上有以下几种^[18]：

1. 泛洪 (flooding)：是基于图论的资源发现算法，把网格看作有向图，网格中的资源看作图中的节点，资源之间的联系看作边，节点间形成无结构网络或具有某种结构的网络，如树状结构或超立方结构。接收到资源发现请求的节点除了搜索本地注册资源信息外，还向所有邻节点转发请求（请求进入的邻节点除外），直至满足结束条件。

2. 路由转发 (RT)：网格中每个资源节点中都存放着所有资源的路由信息，而不是资源的直接位置。即节点是以系统中其他部分节点为邻节点，且知道到所有异地资源的路由信息。在查找某个资源时，资源发现请求通过路由函数计算搜索的方向指向的节点，将被接收节点转发到相应邻节点，并最终被路由到注册有满足条件资源所在的节点。在这种方式下，资源发现过程就是从起点到资源所在节点的路径，资源信息的更新需要传播到每个节点。

3. NEVRLATE：是节点被分为若干组，每一个资源都必须在所有组中至少一个节点上进行注册，因此资源发现的时候只要搜索任意一组即可，但对于资源更新，则需要将其散布到每一个组视为多个类型惟一的不同资源。

2.4.3 资源发现机制

目前网格中的资源发现机制从本质上说可以分为集中式与分布式两种。下面分别对这两种发现机制进行简要介绍^[24]。

1. 分布式的资源发现机制

分布式的资源发现机制属于信息服务自身提供的方式，采用分布方式，不存在任何集中成份，在分布式资源发现机制中，每个结点既是资源提供者又是资源使用者，也是资源发现服务提供者，结点之间的关系是对等的，资源只向“本地”结点注册其描述信息，而结点之间的连接是随机的，整个系统的拓扑结构也是随机的、不固定的，每个结点只存储两种类型的信息：本地资源的信息和其邻结点的位置信息。分布式资源发现机制不存在中央服务器，所有状态信息都被分布于网络中的各个节点上，因而能克服集中式结构的主要缺点，适应动态大范围的环境。然而由于没有中央服务器，系统无法得知资源的分布情形，而且不可能存在逻辑上的全局资源信息分布视图。用户直接向“本地”结点发起资源发现请求，只能依靠存储的本地资源的信息集进行查询，返回可能的满足需求资源的信息，同时采用泛洪（flooding）或随机搜索的形式盲目地向它的邻结点转发请求。

分布式资源发现机制的主要缺点是：

(1) 资源信息空间的无序性和无结构性使得资源发现具有一定盲目性，基于泛洪或广播的资源发现效率较低；

(2) 结点可随时加入或离开使得系统的安全性也难以控制。

但其最大的优点是可扩展性和可靠性，结点可随时加入网络并将自己的资源共享给其它的用户；同时任何结点的关闭或故障都不会影响系统中其他结点的正常工作。

2. 集中式的发现机制

目前有很多的资源发现机制都采用了集中方式，其属于信息服务第三方提供的方式，如 Globus 的 MDS1.0 和欧洲数据网络的复制定位机制 Giggie 等都使用了集中的中心服务器来实现资源发现和资源定位功能，其拓扑结构如

图 3.1 所示。在集中式资源发现机制中，只存在一个信息结点即中心服务器，所有的资源向中心服务器注册其信息，且资源信息被组织形成了一定结构的信息空间，信息空间及资源发现服务都集中在中心服务器上。用户直接向中心服务器发起资源发现请求，通过对资源信息空间的查询，获取所需资源的信息，并通过中心服务器调用该资源，这使得服务方要保证足够的带宽和处理器周期。另外，目前许多 P2P 的系统虽然资源共享采用了 P2P 的方式，客户结点直接调用资源结点上的资源，但是资源信息空间构造仍然采用了集中方式，如 Napster 文件共享系统。

集中式资源发现机制优点：系统拓扑结构简单，构建和维护容易且消耗较小；资源信息空间集中，不存在资源信息不一致问题，而且资源发现效率较高；服务集中，易资源共享并发控制，而且系统安全性较好。

但其有两个最大的缺点：系统可靠性很差，当这个中心服务器出现故障时，系统不能够正常工作，系统没有容错功能；系统可扩展性比较差，受限于中央信息结点的能力，因为系统中的资源信息必须集中存储在中心服务结点，并且所有服务也必须通过中心服务结点，使得系统不能够很容易地增加大量资源。

考虑到以上的资源发现机制都有一定的局限性，本文提出一种新的网格资源发现机制，那就是基于分层树型的网格资源发现机制。

2.5 本章小结

本章主要阐述了网格资源发现的基础理论，包括网格技术、网格的体系结构，给出网格资源的定义，特点及其分类，给出资源发现的定义及特点、资源发现在网格系统中的地位、资源发现的关键技术以及目前资源发现存在的问题分析，说明了各种发现机制的优缺点。接下来的章节将阐述本文所提出的网格资源发现的模型及实现。

第3章 分层树型网格资源发现的模型设计

在前面对网格资源发现的一些基本理论知识进行了概述，本章将提出一个分层树型网格资源发现的框架结构——分层树型网格资源发现模型，给出这个模型的设计过程，并分析实现模型的关键问题。

3.1 管理系统

管理系统由受管对象和管理者构成，受管对象是网格上的一个个资源，管理者是网格资源管理器，网格资源处在资源管理器的管理之下。

按照功能进行角色划分，网格环境中与资源发现相关的主要有三个角色：资源提供者，资源请求者和资源中介者（也可称作资源发现系统），由它们构成资源管理系统。资源提供者是控制资源的智能主体，向外界提供一些可访问的资源，通过注册到网格资源系统，使这些资源有清晰的访问语义，可以被外界发现；资源请求者是使用资源的用户，通常即是虚拟组织成员或需要使用资源的应用。它通过资源发现系统发现所需要的资源，通过匹配过程和资源提供者联系并使用对方提供资源或服务。资源管理系统是为网格用户提供发现服务的基础设施，负责搜集资源的注册信息并进行有效的组织和管理以及接收资源查询请求并返回查询结果。

根据资源管理过程中信息流动路径的不同，资源管理系统可以划分为三种形式^[25]。

1. 直线形

资源请求者向资源中介者提出资源请求，资源中介者为用户寻找合适的资源并驱动资源工作，为用户提供服务。资源为用户服务的结果仍然是通过资源中介者返回给资源请求者。

2. 折线形

用户向资源中介者提出请求，资源中介者为用户找到合适的资源并把资源标识和使用资源的接口信息返回给用户。用户根据返回信息，组织消息和数据，驱动资源工作，得到资源提供的服务。

3. 三角形

用户向资源中介者提出请求,资源中介者为用户寻找合适的资源并驱动资源工作,并告诉资源把服务结果用什么形式、向哪个地址返回给请求者。资源开始工作,服务结束后按照中介者提供的消息把服务结果告诉资源请求者。

由于网格环境是由跨多个组织,每个组织采用不同的管理策略的自治域共同组成的,因为必需要采用一个合理高效的管理系统,本文使用的是三角形的管理系统

3.2 资源发现模型

3.2.1 设计目标

研究网格资源发现模型的目的是为了给用户提供一个高效可靠的资源利用环境,在该环境下用户可以享受到透明的资源发现服务。因此,资源发现系统必须满足以下功能需求^[13]:

1. 能有效地注册、注销和更新资源信息。

网格中存在大量的不同类型的资源,包括处理能力、存储系统、分布式文件系统等等。资源提供者必须将资源信息有效地注册到网格系统中,资源才能被网格资源请求者利用,当资源提供者停止提供某项资源服务时,相应的网格资源信息要从网格系统中及时删除。同样为确保资源信息的精确性,系统要定期更新资源信息。

2. 能有效地发现资源信息。

快速、有效地查询到与资源请求者提交的资源查询请求相匹配的资源信息是资源发现服务需要解决的首要问题。

3. 实现分布式和集中式共同控制。

由于分布式和集中式的发现机制都有各自的缺点,通过采用集中式和分布式相结合的方法,能够提高系统的发现效率和可靠性,又避免了中心服务器工作负荷过大、负载不均衡等问题,并且具有良好的可扩展性。

基于对以上问题的分析,我们提出基于分层树的网格资源发现模型。该

模型的基本思想是：把整个网络中分布的节点从逻辑上分成上下两层：下层由拥有各种网格资源和服务的网格资源节点和信息节点组成；上层为索引节点组成的网络，负责信息节点的组织、管理和资源定位，这里的索引节点由下层信息节点在逻辑上映射过来，构成一个逻辑上的网络组织。

3.2.2 框架结构

著名的 Stanley Milgram 实验发现，通过平均六人次的熟人传递就可以把社会中任意两个人联系起来，这种现象称为“小世界”现象^[32]。“小世界”的实验揭示了两个发现，一个是短链效应普遍存在，另一个是人们可以找到短链。这二个发现说明，当网络呈现某种拓扑结构时，仅利用局部信息就可以有效地找到短链。“小世界”现象普遍存在于各种社会、自然界网络中。网格也存在“小世界”现象，“小世界”网络也有两个特点：任意两个节点间的特征路径短；聚集系数大。前者表明借助“小世界”查找只有很小的网络延迟；后者表明借助网络可以指引节点快速发现最短路径。可以利用“小世界”网络高聚集度的特点来组织网格资源发现模型。

为了充分利用网格资源节点的特性，分层树型的网格资源发现模型的下层由拥有各种网格资源和服务的网格资源节点组成，按照节点网络邻近性和“小世界”理论将这些网格节点及资源划分为若干社区，每个社区选择性能最强的节点作为信息节点，信息节点的信息服务模块作为其所在社区内的资源信息数据库，记录社区内各个可用资源的资源属性信息，负责社区内的资源信息管理和资源查找。作为信息搜索的中心，它负责收集某一区域内多个资源节点的信息，同时各个信息节点之间构成一个基于小世界的网络。上层是由负责注册网格资源的信息节点映射的索引，在社区层之上形成一个个逻辑单元，构成一种分布网络，用以实现对信息节点的组织、管理和资源定位。

上层网络中的每个节点是下层的信息节点，承担在上层网络中信息节点间的资源路由和消息传递任务。其结构见图3.1，

分层树型结构可以保证系统的可扩展性，而每个非树根节点保存了左右邻

居节点的连接，可以方便处理父节点异常失效的情况，提高了系统的可靠性。

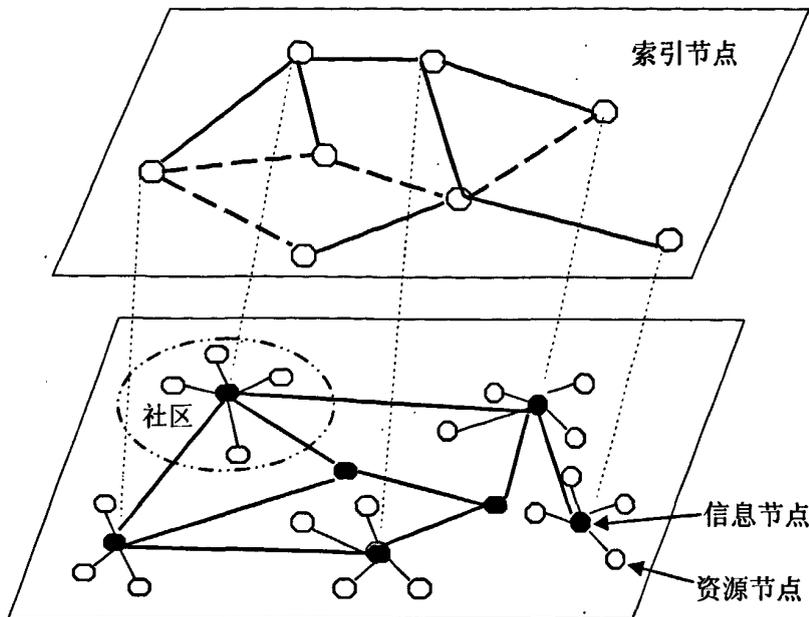


图3.1 基于分层树型的网格资源发现模型

3.2.3 社区的划分

网格社区^[33]是参考人类社会中的社区而建立的概念。目的是便于管理网格中的各种资源，快速实时地把资源请求者和资源提供者联系起来。

社区划分的原则是：将网格系统中地理位置相邻或属于同一局域网的资源信息节点组织在一起，按照一定的协议或规则，构成一个虚拟组织。在这个组织中，所有资源信息、节点形成一个团体，相互合作，相互获取资源，共同完成某些任务。一个社区类似于一个小的网格，而一个庞大的网格系统可以看作是由若干个社区组成，用网络把各种社区连接起来，形成一个没有边际的动态网。当存在一个社区无法完成的任务时，可以由各个社区之间相互合作，相互提供资源，共同完成一些在一个社区上无法完成的任务。

在本模型中，社区是可以嵌套的，社区之中还可以有社区。不同的社区之间可以相互覆盖，一个用户或资源可以根据需要同时属于不同的社区。

对于每一个社区，都至少有一个管理结点，我们称为社区的信息节点，

在一个社区内集中具有相同类型的资源或相同需求的用户，使其组织在一起形成社区成员。资源发现请求转发以及资源信息的扩散都被限制在相应的社区内，因为节点根据注册资源类型加入相应的社区，资源信息复制或链接局限于同社区的节点，只向与请求资源类型相同的社区内的节点转发请求。从而改善了资源发现的性能。

3.2.4 模型的特点

通过上述分析，可以发现分层树型网格资源发现模型方案的设计比较合理，这可以从模型的特点上看出。对本模型的特点总结如下：

1. 整个网络没有一个唯一的中央服务器，而是把中央服务器的功能分散在各个社区中，由各个社区的信息节点完成，并在逻辑上构成索引节点，从而避免了整个网络的单点失效。

2. 每个社区具有一个信息节点用于存储本社区的信息库和资源邻接表。资源发现请求转发具有目的性，有效地减少一次资源发现引发的网络通信量，使得每一层的路由范围很小，有效缩小资源搜索范围。

3. 系统资源信息分布在网络的各信息节点上，每个信息节点只负责维护本地资源信息，以分布方式组织信息节点解决了资源的更新和维护问题，适应网格资源的动态变化社区的规模要控制在一定范围之内，若过大则进行分解，过小则进行合并。

4. 分层结构在底层可灵活地接入现有的其他网格实现系统，使资源发现算法也具有多样性，而且机制的资源发现能力可动态根据资源发现需求的变化进行调整，使系统具有较好的伸缩性。

5. 分层及社区划分不仅省去了上层路由查找目标的消耗，大大加速了路由，而且可能出现大量的直接跨域请求转发，从一定程度上缓解了访问瓶颈问题。

3.3 系统的建立与维护

由于网格中的资源是不确定的，从而导致存在的资源数也不确定。这样

就可能会产生某些社区的资源节点数过少或过多的情况。如果社区中的资源数过少，则资源分布比较分散，不利于资源发现，从而降低了资源发现效率；如果社区中的资源数过多，资源过度集中，则会令信息节点负担过重，导致性能瓶颈、单点失效等缺点。

为解决上述问题，可以把社区中的资源节点数控制在一定的范围。当社区中的资源节点数达到所设上界时，只允许资源节点离开，而不允许资源节点加入，欲加入的资源节点可以构建一个新的社区，或选择其他社区加入；如果社区中资源节点数小于所设下界，可以根据资源所在的物理地址，将相近的社区合并，形成新的社区，使社区保持一定规模。

当一个普通网格节点准备加入网格系统，也就是该网格节点准备将其资源加入到网格系统中，以便其它节点可以共享资源，称该节点为资源节点。任何网格资源节点都可以动态地加入和离开网格系统，资源维护主要就是资源的加入、更新和离开。当资源信息变化时，资源节点要向信息节点提交资源更新信息，以使信息节点更新当前资源状态。信息节点必须尽量保持其信息表中资源信息的正确性和实时性，一个信息节点将定期地收到它的邻居节点向其发送的更新报文，并以此为根据更新本节点的信息表中的相关内容。相应地，该节点也将本地的资源信息周期性的报告给邻居节点。如果一个信息节点在某一段时间内没有收到来自邻居的更新，则认为该邻居节点发生了故障，则不再将请求报文和更新发送给该邻居，并根据特定策略更新拓扑。各信息节点在这一过程中运行一个资源更新守护进程，其作用是每隔一个周期查询所在社区的状态。

3.3.1 节点的加入

当一个资源节点准备加入网格系统时，它将首先联系与之最近的信息节点，并申请加入该资源区域，在收到信息节点确认消息后，则与之建立连接。

在这一过程中，资源节点必须和一个或一个以上的信息节点取得联系，并且只能选择其中之一作为自己的父节点。它选择这个节点作为被请求节点，

然后向被请求节点发送一个加入请求。被请求节点可以是上层索引节点，也可以是下层任意信息节点。这样就同时有多个节点都可以处理资源节点的加入请求，以避免出现性能瓶颈和单点失效问题。

被请求节点收到请求加入消息后，首先向它的父节点和邻居节点并行地转发请求加入消息，然后发送一个应答消息给请求节点。当其它节点收到消息后，也同样先向它的父节点和各邻居节点并行地转发请求加入消息，然后发送一个应答消息给请求节点。请求节点从发出请求消息后就一直等待应答消息，当收到第一个应答消息后，将发出应答消息的节点作为它的父节点，并与它建立连接，至此，请求节点加入网格中。

如果本节点的性能高于其它节点，则自己充当信息节点，并接受其它节点的连接请求，从而形成一个新的区域。该节点可以和其它信息节点取得联系，从而建立区域间的联系。根据上述过程整个系统就可以动态地建立起来，形成一个相对独立但又相互联系的社区结构。

3.3.2 节点的离开

节点可以随时从系统中离开，有两种情况。一种是当节点主动离开时，此时要考虑是信息节点的离开还是资源节点的离开：

1. 如果是资源节点的离开，则删除该节点，并更新在社区上信息节点中的相关数据；

2. 如果是信息节点的离开，在离开前要向它的邻居节点和子节点发送离开消息。若其它节点收到的是来自父节点请求离开消息，则发送消息请求加入到其相邻的信息节点，并等待节点的加入确认消息，一旦收到确认消息就与它建立连接。

另一种是节点异常失效时，若资源节点失效，则直接删除该子节点；若信息节点失效，此时不发送任何消息给邻居节点，但其下属的资源节点需要处理父节点异常离开的情况。方法有两个，一是通过左右邻居节点找到其它信息节点作为父节点，二是在资源节点中重新选择一个性能最强的节点作为

信息节点，然后加入这个作为父节点的信息节点，成为它的子节点。

3.3.3 信息节点的选取策略

每个社区中的信息节点具有双重角色，在上层覆盖网络中负责资源路由和拓扑维护工作，在下层本区域内负责资源信息管理和资源查找工作。为了提高系统的容错性，在选取信息节点的同时选取一个备份节点，用于备份信息节点的数据，当信息节点失效时可以替代它。初始的信息节点由系统指定，但随着社区内资源节点的加入，可能有更强的节点来替代现有的信息节点，或者当信息节点异常失效时，也需要重新选择信息节点。因此，信息节点的选取分两种情况。第一种称为正常选取：当社区内有资源节点需要注册和加入网格系统时，信息节点获取新加入节点的一些参数，如内存大小、CPU速度和网络带宽；然后，根据这些参数分配的权值来计算得到新加入节点的能力值；如果新加入节点的能力值高出原有信息节点一定程度，则选取新加入节点作为信息节点，并将原信息节点作为备份节点。第二种称为异常选取：当信息节点失效或者退出网格系统时，直接将备份节点作为新的信息节点，并在本社区的资源节点中选取一个能力值最大的作为新的备份节点。

3.4 本章小结

本章借助于小世界理论和社区的概念建立了一个分层树型网格资源发现的模型，给出资源发现模型的框架结构，把整个网络中分布的节点从逻辑上分成上下两层，提出索引节点、信息节点和资源节点的概念，把索引节点作为信息节点的映射放在上层，构成一个逻辑上的分布式网络，把信息节点和资源节点放在下层，构成一个个的社区，社区之间又构成一个大的网络。在这个基础上，给出了模型的构建过程。

第4章 分层树型的网格资源发现方法

本章将以第3章分层树网格资源发现模型为基础,在这个基础进行论述,逐步说明资源发现的过程及其方法,并在最后给出实验,进而验证分层树网格资源发现方法的可行性和高效性。

4.1 资源发现的设计思想

在基于分层树的网格资源发现模型中,任何节点都可以处理资源查找请求。

资源提供者必须首先注册到一个信息节点,并周期性地向其发送自己最近的状态。当一个资源提供者收到一个资源请求时,如果本地资源可以满足请求,则向请求者发送相应消息,否则将该请求丢弃。

用户将自己的资源请求发给与自己最近的信息节点上。该请求包含了所需资源的类型和属性值。信息节点维护着邻居节点及信息节点资源状态的信息表,当收到一个请求时,信息节点就检查本地的信息表,看能不能匹配用户的请求,若能匹配,将该资源地址回发给用户,否则请求转发。

4.2 资源发现机制的描述

当用户或应用在某个节点上查找需要的资源信息时,资源的查找和定位可设计如下:

步聚1:首先在本地资源缓存表中查找所需的资源,如果找到满足应用要求的资源,则立即返回资源信息,否则进入步骤2;

步聚2:节点将查找请求发送到信息节点,并在其资源信息库中查找,如果存在相应的资源,则返回相应的资源信息给请求节点,否则,进入步骤3;

步聚3:本地信息节点将资源查找请求发送到索引节点上,并以并行方式进行资源定位,找到所需资源则返回资源信息。

资源定位是整个资源发现的关键,由于该模型以分层树型的网络来组织节点,资源发现请求转发具有一定的目的性,在资源查找请求在信息节点间

的路由过程中不会像完全分布资源发现那样产生大量泛洪消息。当资源发现请求在社区的信息节点间进行资源匹配查找时，资源请求者将所需资源信息通过代理与索引信息层之间的接口进行交互，进而形成资源请求，并向社区提交该请求。

资源发现过程要考虑如下问题：

1. 资源请求在社区内的资源节点查找匹配。

资源请求者首先在所在社区内寻找是否存在能够满足要求的资源，若满足，则查询结束，资源发现返回查询结果到资源请求者，若无法满足，则资源发现返回“无结果”信息。

2. 资源请求信息在社区间转发。

将无法在本社区内执行的资源请求发送到本社区的信息节点，由信息节点转发并选择转发机制，可以根据资源信息节点中同类的社区名称、位置等索引信息将资源请求有选择地发送至一个或多个社区上，接收到资源请求的社区的信息节点与本地的资源发现交互协调实现本社区内的资源的查找。重复这一过程，使资源请求在网格全局内传递。

3. 资源请求信息在索引结点上转发。

在资源发现过程中，只有在本地不存在所需资源的前提下才将资源请求信息提交到索引信息层，并进行查找发现。如果不能在本地找到满足需求的资源，就会将请求转发到相连的索引节点上。从而在索引节点网络上转发，找到所需资源所在的社区，进而找到对应的资源节点。由于不缓存索引节点上的资源信息，因此具有开销小，不会返回很可能是过时的资源信息。

对索引节点的选择是基于以往的经验。每次成功得到邻居命中返回的匹配都会将相关信息缓存到信息表里，其维护了有关所有索引节点的经验，包括邻居的地址、成功返回命中资源的次数、平均响应时间等，如果没有相应邻居的经验值可以参考，则会随机选择一个索引节点。

通过以上分析，可以得到资源发现的流程图，见图4.1。

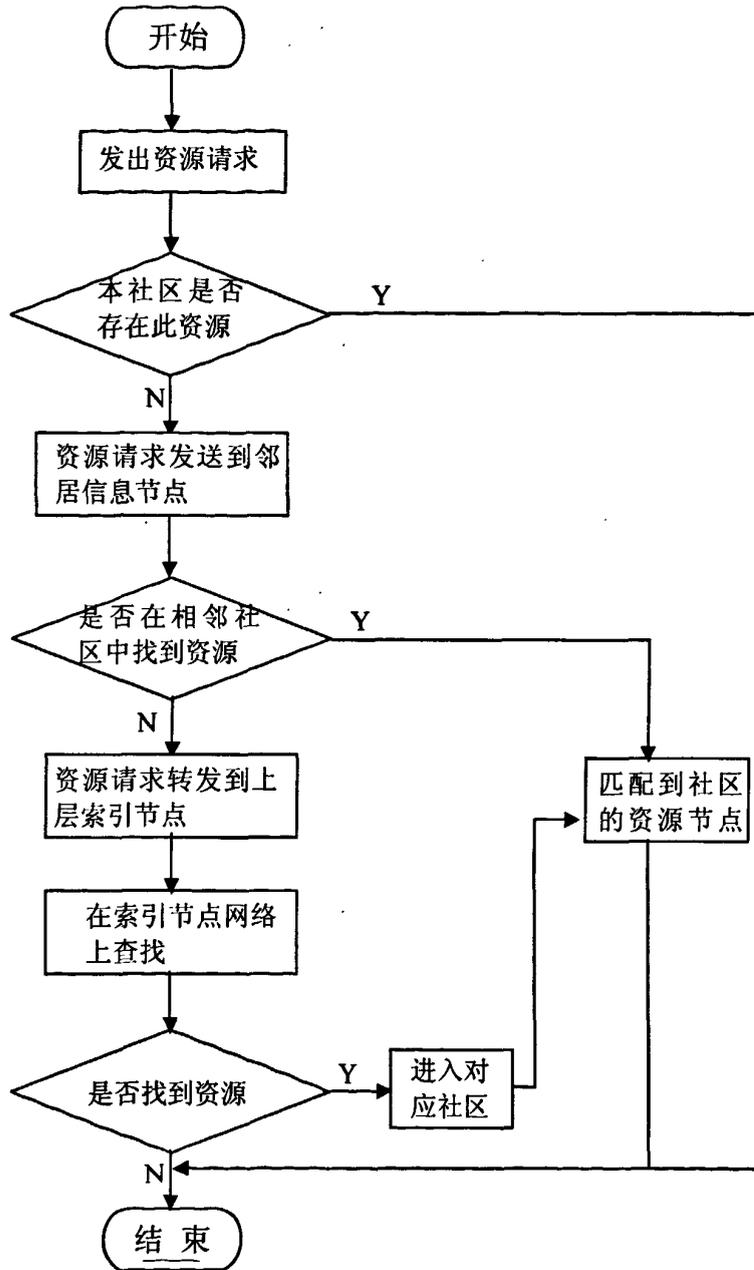


图4.1 分层树型网络资源发现流程图

4.3 资源发现算法

资源发现是资源提供者和请求者进行联系的重要环节，资源发现机制能

够根据用户请求从网格资源中找到满足用户所请求的资源，使得请求者可以发现、定位和使用所需要的资源。完成这一过程，就要设计一个算法，下面就对这一算法进行分析。

4.3.1 数据结构

数据结构是用户查找所需要的网格资源的依据，是联系网格资源提供者和资源请求者的纽带。为了完成算法的实现，要对运算过程中涉及到的存储单元进行分析，本文要用到两个这样的数据结构，下面进行说明。

1. 邻接表：在进行资源发现时，根据信息表的内容可以很快确定所需资源存在概率最大的社区，但确定所选社区后，由于社区中的资源数量也比较大，所以还要为每个社区建立相应的邻接表，并将邻接表存储在信息节点中，用来存放与其它社区信息节点的邻接关系。如下所示：

name	type	num	IP	TTL	Y/N
------	------	-----	----	-----	-----

其中 name 代表社区名称；type 表示资源类型；num 表示社区中存在的资源类型数量；IP 表示管理本社区的信息节点的地址；TTL 是为每个社区设置的一个域，用以进行节点的通信管理。TTL 的设置可以适时地将节点中过时的资源信息从邻接表中清除，它在节点退出系统的时候也起着决定性作用；Y/N 表示本社区是否存在 type 类型的资源。

2. 信息表：在新资源加入的时候要附带资源的固有信息包括：资源ID、资源类型、所在网络地址等信息。信息表可以是以数据结构的形式存放在内存中，也可以是以表的形式放在数据库中，当有新的资源信息需要加入而信息表本身已经存满时，可采用近期最少使用算法进行替换。其结构如下所示：

type	ID	num	IP	time	Y/N
------	----	-----	----	------	-----

其中type是资源类型；ID是资源所在节点的编号；IP是资源所在节点的IP地址，也就是资源节点的IP地址；num是ID和IP所在节点中对应的type类型的资源数量；Y/N表示是否存在type类型的资源；time表示最近一次成功访问该资源的时间。

社区中存储着每个资源的具体信息,信息节点就是这样一个特殊的节点,存储社区中的资源信息的信息表和邻接表。

4.3.2 算法描述

为了描述的方便和通俗性,本系统中的算法描述采用自然语言描述。其方法如下:

Step1. 提交资源发现信息,在本社区内查找是否存在所需资源;若有,则执行Step6,若没有所需资源,则将资源发现信息发送至相邻社区的信息节点上;

Step2. 根据所需资源类型,和相邻信息节点中信息表里对应资源的信息进行比较;

Step3. 根据比较的结果,对一个或几个信息节点中的邻接表进行遍历,对已遍历的资源节点作已遍历的标志;

Step4. 如果在邻接表里找到匹配项,则转到匹配项所对应的社区,再根据这个社区信息节点中的信息表找到对应的资源节点,执行Step6;如果在邻接表里未找到匹配项,则执行Step5。

Step5. 在索引节点中查找可能存在需求资源的信息节点,若找到满足需求资源的要求的节点,则返回该资源的地址给资源请求者,执行Step4,否则,执行Step6。

Step6. 查找结束。

根据算法的需要,需要对系统中节点本身以及节点之间的相互关系作一些参数化的运算^[36]。

(1) 节点编号

对每一个节点,需要计算它的编号。具体的计算方法如下:

$$\{i(s)\} = \{p=i+t^s | i < p < m, s \geq 0\} \cup \{p=i-t^s | 0 < p < i, s \geq 0\}$$

其中, t 为预设的一个常数, i 为节点的编号, m 为网络的规模,可以预先限定一个值,节点 $i(s)$ 就是对应各个节点的编号。

(2) 节点间距离

任何两个节点 i 和 j 的编号之差设为这两个节点的距离: $d(i, j) = |i - j|$ 。

当系统中的某节点准备发送资源查询时, 消息的发送以及转发可以按照如下方式进行:

(1) 设发送查询消息的节点为 Node (i), 并设置一个全局唯一标识 F 。节点 Node (i) 作如下计算:

$$\text{Node (1)} = i + m/t;$$

$$\text{Node (2)} = i + (\text{Node (1)} - 1) / t;$$

...

$$\text{Node (n)} = i + (\text{Node (n-1)} - 1) / t;$$

直到 Node (n) 为 i ;

(2) 设 Node (j) 为接收到 Node (i) 查询消息的某个节点。Node (j) 首先查询在自己所在的社区内是否存在满足查询条件的资源, 如果存在, 则将资源的相关信息返回给 Node (i)。

(3) 否则, 如果节点 Node (j) 已经接收过标识为 F 的消息, 则 Node (j) 不再转发这个消息。

(4) 如果没有接收过, 则节点 Node (j) 作如下运算: $q = j + d(j, m) / t$ 。并在自己的邻居列表中找到使 $|k - q|$ 值最小的节点 Node (k)。如果 Node (k) $\in q$, 则 Node (j) 不转发查询消息, 否则 Node (j) 将查询消息连同集合 $q \cup \{\text{Node (k)}\}$ 一起发送给 Node (k)。

(5) Node (k) 按照 (1) 的方式进行, 区别是如果 Node (k) 为资源节点, 就不再进行消息的转发, 并且如果没有查询到满足条件的资源, Node (k) 就向 Node (i) 发送查询失败的消息。

4.3.3 算法复杂度分析

算法的复杂度可以从时间复杂度和空间复杂度上进行分析^[29]。

1. 时间复杂度。假设网格中有 N 个社区, 相应就有 N 个信息节点, 其中

M个社区中有所查找资源，设第t个社区中有 T_i 个资源节点，各个资源节点存在所需资源的概率相同，则时间复杂度要由两部分组成：

(1) 如果对各信息库中的信息进行排序，选择合适的社区。则利用最优的排序方法，平均时间复杂度与最坏时间复杂度均为 $O(M \cdot \ln^M)$ 。

(2) 对所选社区的邻接表进行遍历以查找合适资源。那么在该社区中进行查找的最坏时间复杂性为 $O(T_i)$ ；平均时间复杂性为 $O(\ln^{T_i})$ 。因此，整个资源发现算法的平均时间复杂性为：

$$S = O(M \cdot \ln^M) + O(\ln^{T_i})$$

2. 空间复杂度。每个社区中都有一个信息节点用来存储信息表和邻接表，它们反映了社区中资源的大体情况，而每个资源节点的具体信息是存储在相应资源节点中的。这样所需存储空间由信息节点的信息表和邻接表所占用空间，以及资源节点存储信息所占用空间的组成。

将本方法与其他方法进行比较，可以看出分层树的网格资源发现方法有一定的优越性：

1. 社区中的邻接表是按照资源类型进行建立的，并且不存放资源内容而只存放资源的相关信息，因此在资源发现和查找的过程中，不会访问与所需资源无关的节点，从而避免了资源发现的盲目性。

2. 由于即不是采用集中的也不是采用分布的控制方式，而是采用集中和分布相结合的方式：将网格中所有资源分割成若干个整体使这在下层形成一个个相对集中的社区，社区中的信息节点在逻辑上构成一个分布的系统，避免了因中央服务器而引起的单点失效和性能瓶颈问题。

3. 由于将与所需资源相关的资源按照优先级和所含资源的数量排序放到了一个信息表中，因此发现是有的放矢的，避免了穷举法的盲目性。

4.4 资源发现中的错误处理

由于网格资源是动态变化的，而资源状态的变化是由资源管理器逐次更新的，因此在实际应用中，会有信息表不一致的情况发生。信息表不一致可

能会导致错误的请求判定。如果请求方向上的资源状态在请求被转发期间发生变化,或变化在确定转发方向前已发生,但还没有来得及传播到当前持有的请求,那么请求转发时依据的是实效的信息,其结果可能是,当请求到达原先认为的目标资源时,资源已经不存,也就不能满足用户请求。对于这种情况,可以降低信息更新周期与资源状态变化周期之间的比值来提高资源查找的正确性。

由于请求的最大转发次数由TTL决定,因此会有TTL已经到期,但请求尚未到达能满足的资源的异常情况,此时路由器会丢弃请求,而用户在等待了一段时间后没有收到请求,会认为请求失败。而且分层树弄算法,请求的转发不会形成环,所以只要请求的TTL大于或等于网络拓扑中的最大路径长度,就可以避免该异常发生。

新的资源加入,即资源注册过程,系统中保存多个同步更新的结构信息服务器,新的资源加入时,系统随机选择一个最佳结点作为该资源的父结点加入系统,同时更新所有相关结点的资源信息。此时在系统中的数据量并不大,因此不会导致消息的无控制传播。资源的变更一般不会导致系统结构的变化,而只是改变相关的数据。资源的退出,由于结点上正在执行的任务,则有可能导致结构的调整^[40]。

4.5 实验与分析

实验主要用于验证本文提出的分层树型的网格资源发现,即通过分层树算法实现网格资源查找,以及验证分层树算法在网格中资源发现中性能的提高。由于条件所限,本文对分层树型网格资源发现模型采用了软件的方法进行仿真实验。

4.5.1 模拟器的选择

目前,世界各国的一些研究机构针对网络的特征,开发了一些网格模拟器,模拟器的作用是模拟一个网格环境,通过配置参数,可以更加真实的模拟出现实环境中的各种应用场景,使得模拟结果更具真实性;通过分析在模

拟器上试验的结果，可以不断改进设计。当前比较有代表性的网格模拟器有 Bricks, MicroGrid, SimGrid, GridSim, ChicSim, EDGSim, GridNet, OporSim.

根据真实的网格系统可以抽象出四类实体，如图 5.1，它们是构成网格环境的主要组件：网络用户、网络应用、网络中间件和网络资源。因此，网格模拟器可以从这四个方面来进行网格环境的模拟。

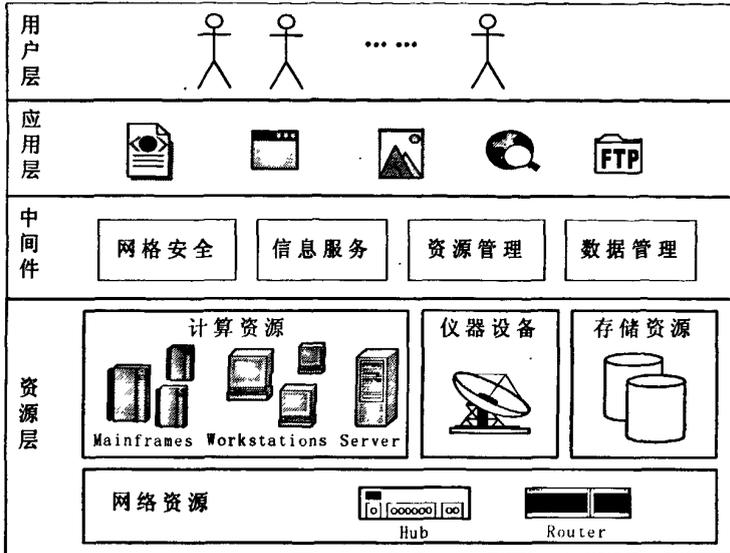


图 4.2 网络系统

通过对比分析各种模拟器在网格用户、网络应用、网络中间件和网络资源上的模拟情况，本文所使用的网格模拟器是 GridSim。GridSim 是在 SimJava^[38]的基础上开发的，它提供丰富的函数库以支持模拟网格环境中的异构资源（时间共享和空间共享）、用户、应用程序、用户代理和调度器。除了通过手工编程来实现模拟外，GridSim 还提供了一套图形界面工具 Visual Modeler (VM) 帮助用户配置网格环境并产生相应的代码。模拟结束后，用户可以调用 GridSim 中的称为 GridStatistics 的库函数来收集各种模拟的统计数据。GridSim 是用 Java 开发的，因此具有跨平台特性。这也是本文选择 GridSim 的原因，GridSim 也是一套 API 的工具包，它也可以产生资源如处理器，也可以产生计算任务。这些都由 API 来完成，使用 J2SDK1.42 编译，然后输出结果到文件中。

4.5.2 实验环境

由于 GridSim 具有跨平台性，因为它可以应用于 windows 操作系统上。下面对 windowsXP 上的实验环境安装过程加以说明。

1. 首先确认机器上是否有 J2SDK，若没有则安装。
2. 设置 Java 环境变量：

在环境变量 Path 中加入下列语句：;C:\j2sdk1.4.1_01\bin;然后添加 classpath 项，值为：

C:\j2sdk1.4.1_01\lib\dt.jar;C:\j2sdk1.4.1_01\lib\tools.jar;

3. 安装 GridSim，点击 gridsimtoolkit 安装文件
4. 编译与运行设置：

在 Classpath 加入：.\gridsimtoolkit-2.2\jars\all.jar，这样可以在任意目录运行 Gridsim 的 Java 程序

5. 测试运行结果。测试文件是软件包中的 Example。

4.5.3 实验结果分析

由于网格资源发现机制涉及到资源模型，信息组织模型，网络模型，请求查询模型等多个方面，因此目前对网格资源发现机制的性能影响参数还没有统一标准。本文认为资源数量、资源分布情况、网络拓扑、每个请求的生存周期以及带宽等几种参数是影响资源发现机制的重要因素。

为了简化问题，忽略在一次资源发现过程中信息结点拓扑以及资源的变化，这里设置 TTL=5，评价指标主要从效率和质量两个方面来考虑，请求的平均响应时间体现了模型的效率，平均返回的匹配结果数量体现了模型的质量。

图 5.2 所示的实验结果说明，在一定的网络规模内，匹配资源数量不断增多的情况下，返回的匹配结果数量增多。

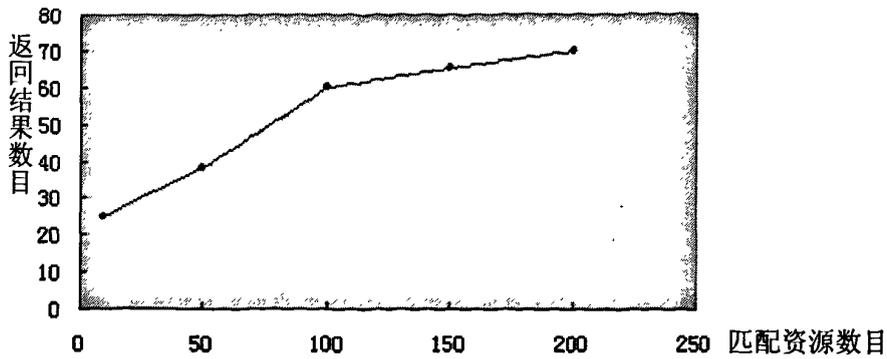


图 4.3 匹配资源与返回结果的关系

图 4.3 所示的实验结果说明，请求的平均响应时间会随着网络规模不断增大而减少，这是因为请求只被转发到较少的节点上，减少了网络开销。

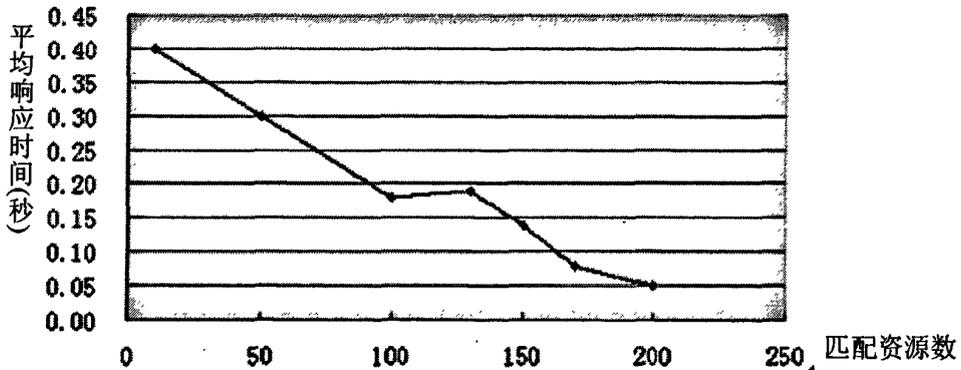


图 4.4 匹配资源数和平均响应时间的关系

在模拟网络仿真实验中可以从另一个角度考察分层树算法：首先测试网络系统中一个资源节点能够多快多准地找到相符的其它资源节点（即查询命中），最终形成资源虚拟组织；然后测试系统中分散的网络资源节点要经过多少次重连。

对系统中的查询命中能力做的仿真试验结果图 5.4 所示。图 5.4 说明运行该算法时，只需要产生很少的消息数，随着算法的运行，系统中查询消息的路由会迅速减小，不会产生过多的消息数来消耗系统的带宽；这一结果说明分层树算法具有较好的有效性。

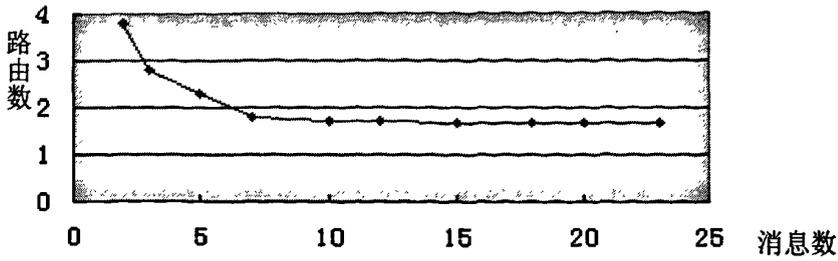


图 4.5 分层树算法中消息和路由的关系

通过模拟实验观察不同规模下分散的网格资源结点采用分层树算法构建资源组织所需重连次数的结果如图 4.5 所示。结果显示随着系统规模的增加，重连次数的增加幅度会变小，并且当规模较大达到 800 结点时的重连次数不超过 6 次，说明采用分层树算法将网格系统中的资源节点组织成资源虚拟社区的系统开销是比较小的。

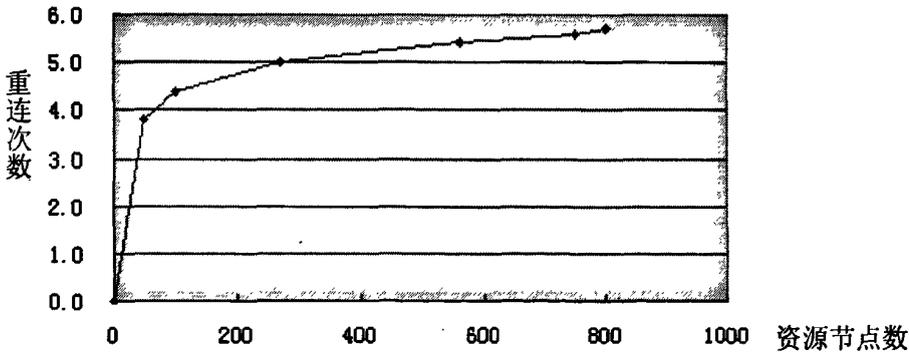


图 4.6 不同规模下的重连次数

4.6 本章小结

本章提出了分层树型网格资源发现的定义和资源发现的工作过程以及在资源发现过程中遇到的异常问题解决。

对基于分层树的网格资源发现方法进行实验模拟，从资源数与返回结果、资源数与平均响应时间的关系，以及消息与路由的关系、不同规模下重连次数等方面，反应出分层树方法具有较好的资源发现效率。

结 论

网格是建立在 Internet 技术、Web 技术和高性能计算等技术之上的综合软、硬件基础设施。采用开放标准，为协同完成科学、工程或信息处理中的应用，提供可扩展、安全、一致、高效的大规模资源共享。由于网格资源种类繁多，网格用户自己可以决定共享或者不共享自己的资源而且随时可以离开或者加入到网格系统中，具有很强的动态性。这就决定了网格的资源发现具有很强的挑战性，虽然目前的网格资源发现有大量的研究，但仍存在许多不足之处。

本文在参阅国内外大量相关文献的基础上，对目前网格项目中比较流行的网格资源发现方法进行了大量的分析和研究工作，提出一种基于分层树型的网格资源发现机制。围绕网格资源发现这一中心，设计了资源发现模型和资源发现算法。主要完成了以下几个方面的工作。

1. 总结了现有的和网格资源发现相关的理论，在这个基础上分析了现有网格资源发现机制的不足。

2. 设计了分层树网格资源发现模型，将网格在逻辑上分成上下两层，下层由拥有各种网格资源和服务的网格资源节点和信息节点组成，将这些节点及资源划分为若干社区，每个区域选举性能最强的节点作为信息节点，负责社区内的资源信息管理和资源查找。上层由负责注册网格资源的索引节点组成，实现对信息节点的组织、管理和资源定位。该网格资源发现模型克服了集中式资源发现模型的瓶颈问题，又解决分布式资源发现模型的洪泛及盲目性，并且有较好的扩展性。

3. 根据分层树型网格资源发现模型，讨论了资源发现的过程及方法，分析数据结构并给出资源发现的算法。又对算法的复杂度进行了分析。

4. 在现有的实验环境下，在实验室机器上利用软件模拟器模拟了分层树网格资源发现模型和资源发现算法，验证了分层树模型的高效性。

由于时间、精力和实验环境的限制，本文的网格资源发现在许多方面还

需要进一步完善。本文目前所做的工作仅在于理论上的探讨，实验环境也仅限于网络模拟器，并且在假定条件时进行了一些简化，没有考虑网络的复杂情况，因此，后续的工作中，将结合网络技术的最新发展，在真实的网络环境和更复杂的网络结构上研究本文方法的有效性。

参考文献

- [1] W. Li, Z. Xu, F. Dong and J. Zhang. Grid Resource Discovery Based on a Routing-Transferring Model. In Proc. 3rd International Workshop on Grid Computing, Baltimore, MD, 2002: 145-156P
- [2] M. Doar, A better model for generating test networks. IEEE Global Internet, 1996: 86-93P
- [3] Rajive Bagrodia, Richard Meyer, Mineo Takai and Yu-an Chen. Parsec: A Parallel Simulation Environment for Complex Systems. IEEE Computer, Vol. 31 (10) , 1998:77-85P
- [4] Henri Casanova. SimGrid: A Toolkit for the Simulation of Application Scheduling. Proceedings of the First IEEE/ACM International Symposium on Cluster Computing and the Grid, 2001: 430-437P
- [5] M. Van Steen, F. Hauck, P. Homburg and A.S. Tanenbaum: Locating Objects in Wide-Area Systems, IEEE Commun. Mag., vol.36, nr.1, Jan. 1998: 104-108P
- [6] G. Ballintijn, M. Steen and A. Tanenbaum, Scalable User-Friendly Resource Names, IEEE Internet Computing, vol. 5 (5) , 2001: 20-27P
- [7] Ian. Foster, Carl Kesselman, and Steven Tueeke. The Anatomy of the Grid: Enabling Scalable Virtual Organizations. International J.SuPercomputer Applications 2001.15 (3) :200-222P
- [8] I Foster, J Geisler, C Kesselman et al. Managing multiple communication methods in high-performance networked computing systems. Journal of Parallel and Distributed Computing, 1997, 40 (1) :35-48P
- [9] 于游. 网络资源发现机制的研究. 燕山大学硕士学位论文. 2007
- [10] 朱凌, 黄德才等著. 一种基于 P2P 分层结构的网络资源发现模型. 浙江工业大学学报 37 卷 1 期. 2009 (2): 79-82 页

- [11] Fred Howell and Ross McNab. SimJava: A Discrete Event Simulation Package For Java With Applications In Computer Systems Modelling. First International Conference on Web-based Modelling and Simulation, Society for Computer Simulation, 1998: 91-114P
- [12] M. Steen , P. Homburg and A. Tanenbaum: Globe: A Wide-Area Distributed System, IEEE Concurrency (1999) : 70-78P
- [13] 夏靖波.网络原理与开发.西安:西安电子科技大学出版社, 2006: 201-208 页
- [14] 孙帅, 董小社等著.基于三层 P2P 结构的网络资源发现模型.微电子学与计算机. 2005 (22.8) : 127-129 页
- [15] 徐志伟, 冯百明, 李伟.网络计算技术.北京:电子工业出版社, 2005:14-46 页
- [16] 都志辉, 陈渝, 刘鹏.网络计算.清华大学出版社, 2002: 17-40 页
- [17] <http://www.globus.org/toolkit/gt3-factsheet.htm>
- [18] R. Raman, M. Livny and M. Solomon, Matchmaking: Distributed Resource Management for High Throughput Computing, In Proc. of IEEE Intl. Symp. High Performance Distributed Computing, July 1998
- [19] S. Fitzgerald, G. Laszewski and M. Swany: GOSv2: A Data Definition Language for Grid Information Services, Feb. 2001
- [20] I. Foster, C. Kesselman, J. Nick, S. Tuecke, 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
- [21] J. Apgar, A. Grimshaw, S. Harris, M. Humphrey and A. Nguyen-Tuong: Secure Grid Naming Protocol (SGNP), Avaki Corporation, University of Virginia, GGF Working Group Chairs and Steering Group, January, 2002

- [22] 杨冬梅, 胡正国, 郑继川, 张小芳. 基于层次结构的网格资源管理系统, 计算机研究与发展. 2002
- [23] 刘星, 肖卫东等著. 基于复合拓扑的网格资源发现机制. 计算机工程与应用. 2005 (9) : 132-136页
- [24] William H. Bell, David G. Cameron, Luigi Capozza, A. Paul Millar, Kurt Stockinger, Floriano Zini. OptorSim - A Grid Simulator for Studying Dynamic Data Replication Strategies. International Journal of High Performance Computing Applications, 2003
- [25] 邹洵. 基于小世界特性的网格资源发现算法. 研究与开发. 2008 (12): 59-62 页
- [26] 朱承, 刘忠等著. 基于信息服务社区的网格资源发现方法. 计算机科学. 2004 (3): 10-14 页
- [27] 肖国强, 邹洵. 基于兴趣聚类的网格资源发现算法. 计算机应用研究. 2007 (11) : 274-277 页
- [28] OGSA information: <http://www.globus.org/ogsa>
- [29] 印玉兰. 基于层次模式的网格资源发现研究. 山东科技大学硕士学位论文. 2006: 33-37 页, 41-43 页
- [30] 邓小清. 基于蚁群算法的语义网格资源发现研究. 西南大学硕士学位论文. 2008
- [31] 刘国清. 网格环境下层次式资源发现方法研究. 湖南大学工程硕士学位论文. 2007
- [32] 谭树斐. 网格环境中资源发现机制的研究. 中南大学硕士学位论文. 2005
- [33] 杨清平, 邓小清等著. 网格资源的组织与发现研究. 重庆师范大学学报. 2008 (10): 70-73 页
- [34] 刑长明, 刘方爱著. 基于 P2P 的网格资源发现机制研究. 计算机技术与发展. 陕西西安. 2006 (8) :42-43 页, 23-24 页, 30 页

- [35] 林伟伟, 齐德昱. 基于分层结构和树型覆盖网络的网格资源发现模型. 华南理工大学学报. 2008 (5) : 6-12 页
- [36] 孙逸风. 一种基于分层的网格资源发现模型. 吉林大学硕士学位论文. 2006: 27-39 页
- [37] Houda Lamehamedi, Zujun Shentu, Boleslaw Szymanski, and Ewa Deelman. Simulation of Dynamic Data Replication Strategies in Data Grids. Proceeding of 12th Heterogeneous Computing Workshop, 2003
- [38] 吴思远. 基于分布式资源覆盖树的P2P网格资源发现机制. 计算机应用与软件. 2007 (10) :23-25 页
- [39] 张晓红, 刘治国, 张德育. 基于分层分布式网格的资源发现机制研究. 大众科技. 2007 (11): 62-63 页
- [40] 刘磊. 网格环境下资源管理. 黑龙江大学硕士学位论文. 2005: 49-58 页
- [41] 李恳. 分布式网格资源发现系统设计与实现. 华中科技大学硕士学位论文. 2007:25-42 页

致 谢

首先向我的导师王慧强教授致以最诚挚的谢意!王老师严谨的治学态度、踏实的工作作风、和蔼可亲的为人给我留下深刻的印象。

其次是感谢我的同学秦英楠,她在我哈尔滨学习期间于生活上给了我很大的帮助。

还要感谢我的家人,读研的时候女儿刚出生几个月,公公婆婆虽然是已经 70 多岁的老人,可他们在我学习期间,还是义无反顾的承担了帮我照顾女儿的重担。丈夫虽然因为纪检工作经常出差,也尽一切可能的帮我分担家务和照顾女儿。

再次感谢所有关心帮助过我的老师、同学和朋友们!

个人简历

1992年9月~1996年7月，就读于黑龙江省富锦市第一中学

1996年9月~2001年7月，就读于黑龙江大学计算机系，获学士学位

2001年8月至今，在黑龙江大学伊春分校(伊春职业学院)工作，担任教学任务