

清 华 大 学

综 合 论 文 训 练

题目：基于网格技术的异构数据库整合在农业信息集成中的应用

系 别：自动化系

专 业：自动化

姓 名：万宇鑫

指导教师：曹军威 研究员

2007 年 6 月 20 日

关于学位论文使用授权的说明

本人完全了解清华大学有关保留、使用学位论文的规定，即：学校有权保留学位论文的复印件，允许该论文被查阅和借阅；学校可以公布该论文的全部或部分内 容，可以采用影印、缩印或其他复制手段保存该论文。

(涉密的学位论文在解密后应遵守此规定)

签 名：_____ 导师签名：_____ 日 期：_____

中文摘要

近十年，全国已累计投资 10 亿元左右进行农业知识资源的收集、分析、处理与应用，农业知识与数据资源已达到 TB 量级。而农业知识的最底层资源分布在不同区域、不同单位，如何集成分布在不同区域的多个农业知识集，增强海量信息处理的能力，是当今农业知识信息服务面临的一个瓶颈问题。

自网格概念提出以来，基于网格的资源共享技术成为解决信息集成问题的重要途径。本文主要探讨的是如何将新兴的网格技术与传统的数据库集成思想结合起来，以解决异构农业数据的集成问题。在此基础之上，本文提供了一套可行的系统构建方案并在实验环境下进行了实现。

本文首先描述了传统的分布式数据库系统组成结构以及将其应用于农业数据集成的主要问题，并介绍了网格技术及相应中间件在数据集成方面的发展现状。然后结合上述两种技术提出了一种基于网格技术的异构数据库集成方法，并结合 Globus 网格中间件及 ODBC 技术设计了一种网格环境下异构数据资源整合的框架。在文章的后半部分，则具体描述了集成系统的搭建过程。

关键词：农业数据、分布式、异构性、网格、数据库集成

ABSTRACT

During the past 10 years, billions of national investment has been spent on agricultural information system for data collecting, analyzing and processing, agricultural knowledge and data resources have reached TB magnitude. However, as the bottom data source are distributed in different area and different organization, it comes a bottleneck problem for agricultural information services that how to integrate all these high distributed agriculture data and provide mass information processing ability.

With putting forward of Grid concept, grid-based resource sharing mechanism has become an good solution of information integration. This article focus on how to combine the grid technique with the traditional database integration mechanism for an new data source integration framework. And then we provided a scheme of structure of integration system and realized it in experimental environment.

We first introduced the structure of traditional distributed database integration system and its deficiency when used in agricultural information system. Then we summarized the development status of grid-based database integration and its middleware. After that, we put forward a heterogeneous database integration method based on data grid concept and provided an realizable framework based on Globus toolkit and ODBC technique. At last, we detailed the installation and usage of this system.

Keywords: Agricultural Data Distributed Heterogeneous Grid
Data Integratio

目 录

第 1 章 序言	1
1.1 项目背景	1
1.2 技术背景	2
1.2.1 网格技术介绍	2
1.2.2 异构数据库系统简介	4
1.2.3 文件结构安排	4
第 2 章 关键技术分析	5
2.1 分布式数据库系统	5
2.1.1 分布式数据库系统介绍	5
2.1.2 分布式数据库系统组成形式及设计框架	6
2.1.3 分布式数据库系统应用于农业信息集成的问题	7
2.2 网格环境下数据整合相关技术	8
2.2.1 Globus Toolkit	8
2.2.2 OGSA-DAI	9
2.2.3 OGSA-DAI 及其类似项目用于农业信息集成	13
第 3 章 农业信息集成系统设计	15
3.1 我们的目标	15
3.2 系统框架设计	16
3.3 实际系统模型	16
第 4 章 农业信息集成系统的实现	19

4.1 底层异构数据库整合----ODBC 数据库访问接口	19
4.1.1 ODBC 接口标准及开发原理	19
4.1.2 用 ODBC 实现的数据库集成	20
4.2 利用 Globus 工具包实现的数据网格	24
4.2.1 Globus 中的安全认证	25
4.2.2 Globus 中的用户代理机制	26
4.3 Apache 与 PHP 构成服务平台	28
4.4 系统工作流程	28
4.5 目前实现的功能	30
第 5 章 系统评估及未来工作展望	33
5.1 系统性能评估	33
5.2 未来工作展望	33
插图索引	35
表格索引	36
参考文献	37
致 谢	39
声 明	40
附录 A 外文资料调研报告	41

第1章 序言

1.1 项目背景

近十年，全国已累计投资 10 亿元左右进行农业知识资源的收集、分析、处理与应用，农业知识与数据资源已达到 TB 量级。到目前为止，知识信息服务已全面应用于全国农业种植、养殖业的生产经营全过程，全国各省市地方管理机构正在逐步建立以农业知识资源为核心的农业科技支撑，采用农业知识来指导高效生产和结构调整。农业知识的最底层资源分布在不同区域、不同单位，如何集成分布在不同区域的多个农业知识集，增强海量信息处理的能力，是当今农业知识信息服务面临的一个瓶颈问题。从事农业知识科学研究和应用的组织、团体，迫切需要在知识密集型计算领域应对交叉和大规模协作式的科研活动，有效解决在时间、空间上高度耦合、交叉的农业生产问题。

农业数据的主要特点在于数据的海量性以及地理上的分散性，而农业知识集成系统的目标就在于利用目前的网络技术及软件技术将地理位置分散的数据资源及计算资源整合在一起，构建一种可扩展的基础设施框架，对外提供统一的访问接口，解决建立农业信息系统中的包括集成、共享、存储和利用的问题，为农业领域不同专业技术人员提供方便、高效的综合农业知识服务。其中将涉及不同领域模型的知识进行整合，使农业知识在一种协同计算框架支持下进行有序管理和协同工作，为分布用户提供大规模、大范围和跨地域的农业资源与服务共享，改变农业领域知识的研究方式和传播方式，降低农业信息化成本，提供有效服务。

农业知识集成系统可划分为三个层次，第一层是实现农业异构数据的转换和集成，即实现异地不同数据库结构下异质数据统一虚拟访问，提供异质数据统一虚拟访问中间件，屏蔽数据的访问的差异性，进一步可以研究数据传输策略，建立数据传输虚拟代理数量与并发用户数目映射关系的最优模型，提高数据访问效率。第二层是实现农业信息的无缝共享，包括信息数据库的构建、信息的发现、联通及处理等。第三层是替提供基于语义的信息搜寻和检索，从底层的数据和信息中发掘知识、处理知识、应用知识，同时研究各数据资源之间的联系和关系，实现业务流程的组合调度。

本项目属于农业信息集成系统中的一部分，主要目标在于解决基本农业数据资源的整合问题，消除由于地理原因形成的信息孤岛，充分发掘已有的农业信息资源，构建农业信息集成的基本框架，实现异构数据资源的集成，为上层的知识提取及数据挖掘提供基础。

1.2 技术背景

1.2.1 网格技术介绍

网格是继 Internet、Web 技术之后的第三次互联网技术浪潮，Internet 及 Web 技术实现了计算机和网络的连接，构建了信息传递的网络技术，其主要关注的是网络的传输速度、信息流量大小及传输的安全性。而网格的关注的是如何有效的管理和利用互联网上分散的各种资源，利用 Internet 把地理上广泛分布计算资源、存储资源、带宽资源、软件资源、数据资源、信息资源、知识资源等连成一个逻辑整体，就像一台超级计算机一样，为用户提供一体化信息和应用服务（计算、存储、访问等），最终实现这个虚拟环境下的资源共享和协同工作，彻底消除资源“孤岛”，最充分的实现信息共享。网格技术将实现 WWW（World Wide Web）到 GGG（Great Global Grid）的变革。

网格技术源自于 20 世纪 90 年代初由美国政府资助的分布式超级计算（Distributed Supercomputing）项目 I-WAY，其概念借鉴于电力网提出，网格的最终目的是希望用户在使用网格计算能力解决问题时像使用电力一样方便，不用去考虑得到的服务来自于哪个地理位置，由什么样的计算设施提供。除去资源的分布性外，换个系统还有另外四个特点^[1]。一是异构性，组成网格的资源是异构的，例如计算资源有不同类型的计算机，不同的计算方式，不同的计算接口，不同的系统架构，而相应的存储资源也有类似特点，因此，网格既要具有利用资源的异构特点进行处理的能力，也要具有提供一致资源管理的能力；二是自治性，网格上的资源首先是属于某一本地的个人或者组织，网格资源的拥有者对资源具有最高级别的管理权限，网格应该允许资源拥有者对其资源有自主的管理能力；三是动态性，由于网格中的资源具有自治性，因此网格资源可能动态地加入或者退出网格，也可能出现故障而导致不可用；另外，资源的性能情况也可能发生较大的变化，使得供网格使用的资源也会发生相应的变化，由于网格没有集中控制能力，因此，对于这种动态性需要有一种机制来保障网格应用的运行不会遭受比较大的影响；四是自相似性。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，网络的构建通过小的局部网格可以形成更大的网格，其构成方式具有相似性。

网格技术最初主要用于计算领域，如分布式超级计算、高吞吐率计算以及数据密集型计算等，而现在网格技术已经被推广至应用服务领域，例如广泛的协同工作、环境保护、培训和教育、信息集成等。同时，企业和组织根据一定的目标和规划构成虚拟组织，共享应用和数据，整合计算能力、存储和其他资源，通过对这些资源进行共享、有效优化和整体管理，降低计算和存储的总成本。根据网格客体对象的不同，可以把网格分为数据网格、计算网格和服务网格。数据网格中共享的基本单位是数据，主要解决数据的共享问题；计算网格中共享的基本单位是计算资源，为用户提供共享资源的良好接口和机制；服务网络中共享的对象是服务，以服务的形式提供共享的手段。在服务网格中，人们可以把不同的资源经过封装，用服务的形式提供给网格主体使用。

就目前网格技术在世界范围内的发展来看，美国仍然是走在最前列的国家，多家研究机构开展了与网格相关的研究工作，开展了 Condor、Legion、Globus 等比较有影响的网格项目。欧洲也是网格研究和建设非常活跃的地区，已经启动了多个网格研究项目，其中的欧洲网格和数据网格是由多个国家参加的研究项目。而亚洲的中国、日本、韩国等国都先后举办了网格方面的高级国际论坛或研讨会，网格的研究和应用已经在亚洲推广起来。我国于 96 年开始开展网格方面相关技术的研究，目前已得到迅速发展，有中国网格 CNGrid、中国教育科研网格 ChinaGrid、中国国家自然科学基金网格 Crown 等项目^[2]。

1.2.2 异构数据库系统简介

异构数据库系统是相关的多个数据库系统的集合，可以实现数据的共享和透明访问，每个数据库系统在加入异构数据库系统之前本身就已经存在，拥有自己的 DBMS。异构数据库的各个组成部分具有自身的自治性，实现数据共享的同时，每个数据库系统仍保有自己的应用特性、完整性控制和安全性控制。异构数据库系统的异构性主要体现在三个方面。第一是计算机体系结构的异构，各个参与的数据库可以分别运行在大型机、小型机、工作站、PC 或嵌入式系统中。第二是基础操作系统的异构，各个数据库系统的基础操作系统可以是 Unix、Windows NT、Linux 等。DBMS 本身的异构，可以是同为关系型数据库系统的 Oracle、SQL Server 等，也可以是不同数据模型的数据库，如关系、模式、层次、网络、面向对象，函数型数据库共同组成一个异构数据库系统。

1.3 本文内容及目标

本文主要关注的是如何将数据网格的思想与分布式数据库系统结合起来，利用已有的成熟的数据库集成技术并结合网格技术的动态性及可对大量数据进行处理的特点设计农业数据集成的框架，构成一个松耦合的农业信息平台。其中第一方面是异构数据库系统的整合，包括操作系统、数据库供应商、数据表列命名不同等三方面的异构问题，目标是屏蔽上述异构问题，对外提供统一的查询接口。文章的第二部分则是数据网格技术在信息集成中的应用，将已有的一些网格中间件应用于数据库系统中，使得海量数据的传输及处理得到实现，利用网格技术的动态性特点构造松耦合的集成系统。

1.4 文章结构安排

在本文的第 1 章中，我们介绍了该项目的实际工程应用背景及相关的网格技术、异构数据集成方面的背景。

在第 2 章中，我们将对数据集成方面两个关键的技术：分布式数据库系统以及网格环境下数据集成技术进行分析。我们将结合农业信息集成的特点分析目前已有技术的优劣。

在第 3 章中，我们将结合第 2 章分析中提出的问题设计农业信息集成的框架及实现流程图，将具体讨论系统实现过程中主要的技术难点和需要解决的问题，我们将结合框架图具体分析说明每一部分的作用。

在第 4 章中，我们将介绍农业信息集成系统的实现过程，运用了哪些软件及技术手段，分别是怎么解决第 3 章中提出的技术问题的。

在最后一章中，我们将介绍该项目之后的需要继续开展的工作以及最后需要达到的目标。

第2章 关键技术分析

2.1 分布式数据库系统

2.1.1 分布式数据库系统介绍

分布式数据库系统研究起始于 20 世纪 70 年代末，是随着网络技术的发展而诞生的。它的发展源于实际生产生活中对于分散数据资源整合的需求。早期的数据库系统多为集中式系统，即由一台中央系统进行所有的数据储存和维护，对于客户机而言只是通过网络连接到中央系统进行查询或其他应用操作，客户机本身仅作为输入输出设备，并不进行数据的存储及处理。而随着计算机技术的发展及普及，一方面需要存储的数据量变得越来越大，集中式的数据库系统为满足对于海量数据的存储和处理必然对硬件提出了很高的要求，使得成本上升；另一方面，分散于各地的计算及存储资源也变得越来越多样，对于地理上分散的公司、组织或团体而言，如何将这些分布的数据资源整合在一起变成了一个迫切的需求。分布式数据库系统就是为了解决这些地理上分散而又逻辑上相关联的数据库系统的整合问题。

分布式数据库系统有三大特性。一是数据的分布性，即存储数据的单元物理上存在分散性，数据往往分布在多个节点上，由网络将这些节点连接在一起；二是逻辑整体性，这些分布在多个节点上的数据往往是关联在一起的，构成一个逻辑上的整体，由逻辑上的整体对外提供信息服务；三是站点自治性，即在集中控制机制协调处理个节点数据库的同时，各个数据节点仍然享有一定的自治权限，可以由节点的 DBMS 对本身的数据进行处理操作。根据以上三个特性又可以推出分布式数据库系统的其他一些特点，例如由于分布性造成的数据独立性（包括数据逻辑独立性、物理独立性与最重要的分布透明性等三方面）、站点自治性形成的“集中与自治相结合的控制结构”以及可以适当增加数据冗余等。1987 年，C. J. Date 提出了完全的分布式数据库管理系统应遵循的如下 12 条原则：

- (1) 场地自治性(Local Autonomy)
- (2) 不依靠中心站点(Nonreliance On Central Site)
- (3) 持续操作性(Continuous Operation)
- (4) 位置透明和独立性(Local Transparency and Local independence)

- (5)数据分割独立性(Fragmentation Independence)
- (6)分布式独立性(Replication Independence)
- (7)分布式查询处理(Distributed Query Processing)
- (8)分布式事务管理(Distributed Transaction Management)
- (9)硬件独立性(Hardware Independence)
- (10)操作系统独立性(Operating System Independence)
- (11)网络独立性(Network Independence)
- (12)数据库管理系统独立性(DBMS Independence)

2.1.2 分布式数据库系统组成形式及设计框架

分布式数据库系统有三类组成方式^[3]，第一种是紧耦合式的 DDBS，全局控制信息放在一个中心站点上，必须通过中心节点才能访问到各个分片上的数据资源，其特点是容易实现数据完整性和一致性，但系统的鲁棒性不高，一旦中心节点出现问题，系统就无法工作；第二种是联邦式 DDBS，对于每个分片来说都包含有全局控制信息的一个副本，不存在一个特殊的中心节点，但这类的系统缺陷是在保持数据一致性方面难度很大；第三类则是上述两个类型的综合，设计两类节点，一类是具有全局控制信息的主节点，另一类是为主节点提供数据资源的节点，但这种系统的缺陷在于选取主节点的过程较为复杂且很难与实际情况相吻合。

集中式的数据库系统由数据库（DB）、数据库管理系统（DBMS）以及用户（包括 DB 管理员及一般访问用户）三部分组成，而分布式数据库系统在此基础上对于上述组件进行了扩充，数据库分为全局 DB 及局部 DB，相应管理系统 DBMS 也被区分为全局 DBMS 和局部 DBMS。一个分布式数据库系统的全局 DBMS 需要定义三方面的结构关系，第一是定义分布式数据库中数据的整体逻辑结构，达到数据的“分布透明性”^[4]；二是需要将逻辑上整体的数据库分为不相交的分片，定义分片与全局关系的映射；三是需要定义具体的分片与物理存储节点的对应关系，一个分片可能对应多个节点，这时就是冗余式的分布式数据库，反之若一个分片只对应于一个特定的物理存储节点，则是非冗余的分布式数据库。分布式数据库系统的模型结构如图 2.1 示：

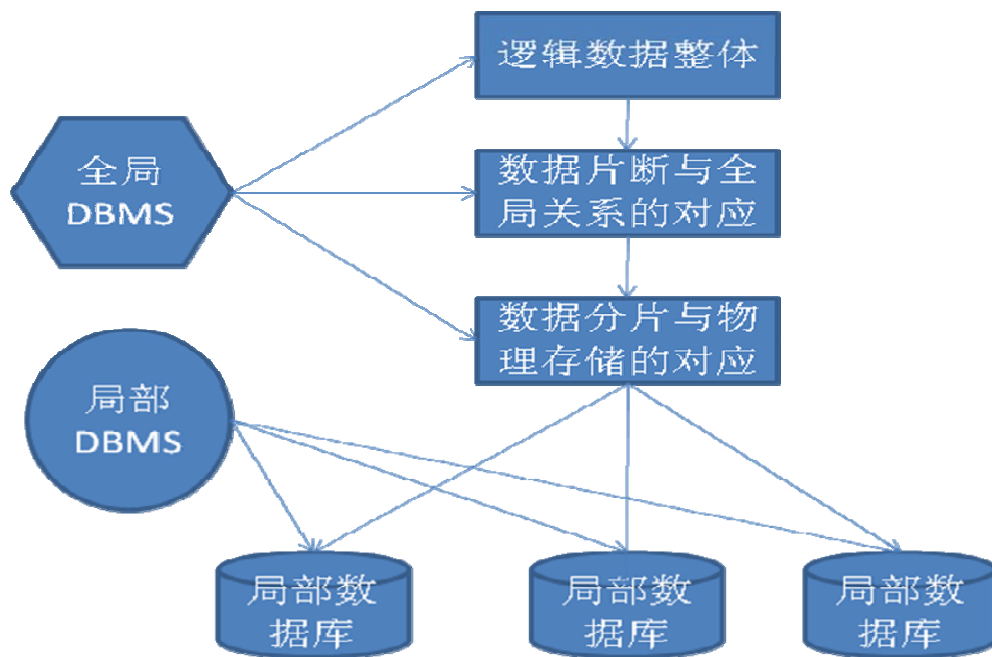


图2.1 分布式数据库系统模型结构

2.1.3 分布式数据库系统应用于农业信息集成的问题

分布式数据库系统是一直以来解决数据库集成问题的一个重要手段，目前已经有一些完善的系统构建体系及用于数据库整合的中间件。而底层农业数据就是以数据库的形式存储的，一个很直接的想法就是直接将分布式数据库系统用于农业信息集成，构建一个全局模式的逻辑数据集合，屏蔽底层物理分散存储的数据分片，粗略看技术层面不存在任何难点且可以很好的解决信息集成的问题，但为什么到目前为止还没有人采用这种方式呢，实际上这是与农业数据的几个特点有关的。

前面谈到农业数据信息集成中最大的一个问题就是数据的海量性，可以假设一下，如果我们采用分布式数据库系统的话，首先排除的就是联邦式的 DDBS 构建方式，因为联邦式的 DDBS 为了实现系统的稳定，需要在每个存储节点布置全局控制信息的副本，我国目前有六百多个市，如果其中的一半有农业数据的分散节点的话，需要构建 300 个左右全局数据库，即使我们采用非冗余的分布式数据库系统（即不在某个数据节点上保存其他节点数据的副本，普通计算机不可能维护 TB 量级的数据库）也不可能投入实际运营，因为这种框架不仅在设计分布式系统的逻辑数据集合时需要耗费大量成本，维护起来的开销也是不可接受的。

农业信息集成的另一个难点是伴随着农业信息化程度不断提高，数据节点的个数也再不断的增加，节点的数量及地理位置是动态变化的。而分布式数据库系统基本属于一个紧耦合的系统，一旦系统构建完成，再往其中添加新的节点会非常麻烦，因为需要更改全局 DBMS 的内容，重新构建逻辑数据集合，同样会造成巨大的系统维护成本。所以紧耦合的分布式数据库系统不可直接用于农业信息技术集成。

另一方面，分布式数据库系统中全局系统对于子数据节点有增、删、改的全部权限，而实际应用中农业信息集成只需要提供数据的查询和共享就可以了，不应该让全局系统享有过高的权限。

上面谈到的是制约分布式数据库系统直接应用于农业数据集成的几个问题，但同时我们必须做的一件事是底层农业数据库的整合，而目前这方面又只有分布式数据库技术可以利用，所以我们必须引入其他的技术并结合分布式数据库技术才能实现目标。这个其他的技术就是数据网格。

2.2 网格环境下数据整合相关技术

2.2.1 Globus Toolkit

Globus Toolkit 源于 Globus 项目，后者是目前国际上网格技术方面最具影响力的项目之一。Globus Toolkit 是一个网格技术基础构件，其目标是在保证单个系统自治性的前提下实现网络上计算资源、存储资源及其他资源的安全共享^[5]。Globus 工具包中包含用于资源发现、资源管理、系统监测、安全认证以及文件系统管理等多种服务构件及其程序库。它是一个开源软件，任何人都可以从其官网上下载其源代码并进行研究改进，可以被使用在商业及非商业的任何领域。Globus 工具包尤其适用于应用服务开发和系统集成方面，事实上 Globus Toolkit 就是一个为网格应用提供中间服务及程序库的网格中间件。

Globus 工具包的实现主要有四方面的内容^[6]：

(1) 网络安全，即网格技术中数据传递及消息认证方面的安全性保证，Globus 中运用了目前许多成熟的分布式安全技术，并结合自身应用进行了一些扩张，目前广泛使用的数据传输安全协议 SSL 及 TLS 都被应用于 Globus 工具包中，还有 Globus 工具包中包含的基于代理的消息认证机制 Grid Security infrastructure(GSI)，GSI 负责在网格环境下的安全认证和加密通信，提供单点登录功能、远地身份鉴别功能、数据传输加密功能。

(2) 网络信息获取与分布，即在网络计算环境中发布资源信息，查询、检索资源信息。具体实现是 Metacomputing Directory Service(MDS), MDS 基于 Lightweight Directory Access Protocol(LDAP)协议，主要完成对网络计算环境中信息的发现、注册、查询、修改等工作，反映网络环境下各种资源的和服务主体的运行情况。

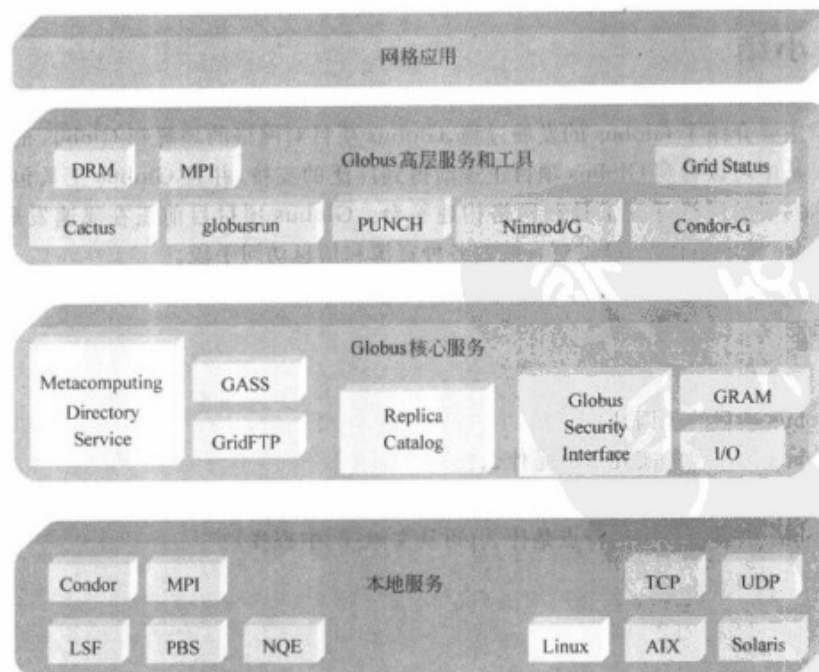


图2.2 Globus框架图

(3) 网络资源管理，由于网络环境中的资源主要分布在广域网环境中，已有的局域网资源管理技术不能有效地对其进行管理，因此 Globus 在局域网资源管理之上实现了更高层次的资源管理技术 GRAM (Globus Resource Allocation Manager)，负责资源请求、任务调度以及远程任务管理。

(4) 网络数据传输，实现网络环境下大量数据的安全可靠传输，以及对于远程文件 I/O 的访问，Globus 工具包中嵌套了 Grid Ftp 工具包以及 GASS (Global Access to Secondary Storage)，实现了数据传输及 I/O 访问方面的需求。

2.2.2 OGSA-DAI

OGSA-DAI 是 e-Science 领导下一个项目并且利用了 Globus 的相关技术，OGSA 是指目前的开放网格体系(Open Grid Service Architecture) 而 DAI 代表数据接入及整合(Data Access and Integration)。类似于分布式数据库系统，OGSA-DAI

将异构的数据资源整合为逻辑上统一的整体，目前支持的数据资源包括关系数据库、XML 数据库及文件系统。按照 OGSA-DAI 官方网站上的描述，该项目的研究目标如下^[7]：

- 1) 发掘网格环境中的各种数据资源，包括关系数据库、XML 数据库及文件系统等
- 2) 提供查询、更新、转换和传递数据所需的网络服务
- 3) 提供一种统一的数据库接入方式，屏蔽数据库的异构性
- 4) 支持元数据的概念，并且可以连接到元数据所处的数据资源
- 5) 支持多种数据资源的数据基层
- 6) 提供中间网络服务，该服务可以用于组成更高级得网络服务用于数据集成和分布式问题处理
- 7) 使得未来的科学家不再关心数据的地点、结构，数据传输及整合，而专注于应用数据的分析及处理

OGSA-DAI 定义了三种主要的动作用于与数据资源进行交互^[7]，用户可以在此框架之上添加自己的新的应用。这三种动作分别是：语句综合(Statement activities) 将用户提交的查询任务转换为底层数据库可以“理解”的语句格式，然后再进行数据的查询和上传；格式转换(Transformation activities) 将获得的

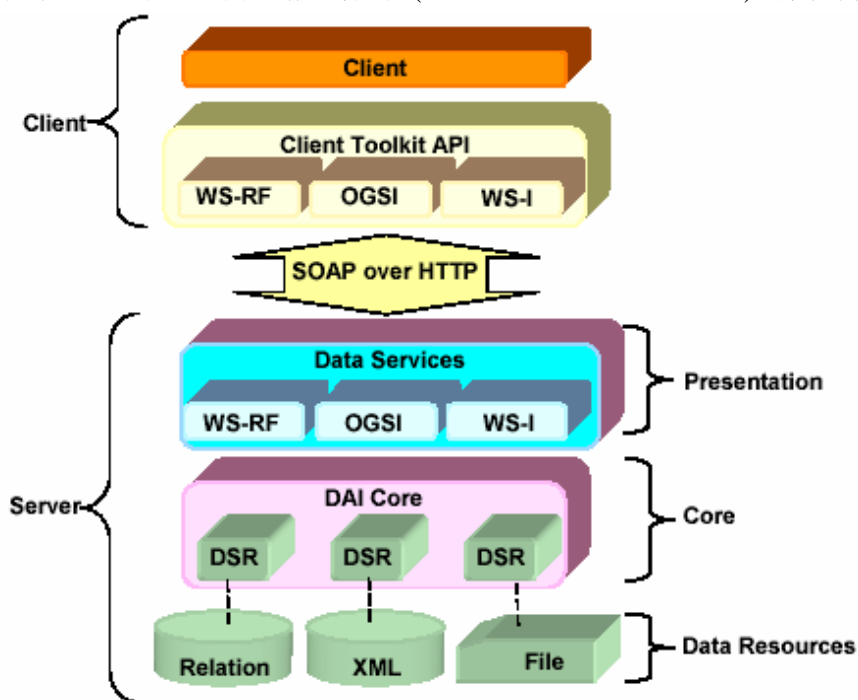


图2.3 OGSA-DAI 组成结构

数据转换为更合适于用户接收的格式，比如说将原始的 XML 数据结果转换输出到 Web 页面上或是进行压缩；传输方式(Delivery activities) 定义传递数据的方式比如 Grid Ftp 或是 SMTP。早期的 OGSA-DAI 版本都是基于 OGS (Open Grid Services and Infrastructure) 框架构建的，之后陆续增加了对于 WS-I, WSRF 平台的支持，目前最新的 OGSA-DAI6.0 版本包含了上述三种平台结构下的实现。OGSA-DAI 系统由服务端和客户端两部分组成，其中服务端主要设计了数据库访问、连接和作业调度方面的结构；客户端部分则主要定义了用户提交请求到提交标准查询文件这一过程。OGSA-DAI 的组成框架如图 2.3 所示^[8](该图取自 OGSA-DAI 工作组发布的《OGSA-DAI Status and Benchmarks》):

1) 客户机工具箱(Client Toolkit API) 屏蔽了不同的消息传递结构，在获取用户提交的申请之后，形成标准的执行文件并加上所用平台的类型信息，通过 SOAP 协议将其提交给表示层(Presentation layer)。

2) 表示层由一系列数据服务组成，主要完成两方面的功能。一是从客户端接收请求信息，这些请求可能来自于 OGS、WSRF 或是 WS-I 等几种不同的服务，表示层需要将这些不同服务提交的请求转换成一种标准格式。另一方面表示层需要选择核心层中合适的数据服务资源(Data Services Resource)并将任务执行档(Perform Document)提交给特定的 DSR。同时从表示

3) 核心层又被称为处理层，是由一组数据服务资源 DSR 构成的，DSR 表现为实际数据源的映射。每个 DSR 都通过特定的 DAI 函数与实际的数据资源一一对应，这个数据资源可以是数据库也可以是简单的文件系统。在这一层中 OGSA-DAI 将调用 JDBC (Java Data Base Connectivity, 是一种用于执行 SQL 语句的 Java API, 可以为多种关系数据库提供统一访问) 驱动管理具体的数据源。DSR 也可以增加其它的服务比如在数据传输时由第三方保存一个数据的 cash(如果需要支持异步的数据传输)。另一方面，DSR 还需要将查询返回的数据信息以反馈文档(response document)的形式返回给表示层。三种不同的服务平台 WSRF、OGSI 或 WS-I 对应的 DSR 都是一样的但每个 DSR 都对应于一个特定的数据源类型。

4) 数据层定义了 OGSA-DAI 可以访问的数据资源，包括关系数据库、XML 数据库及文件系统。

上面是整个 OGSA-DAI 的组成结构，而图 2.4 则显示了一个典型的 OGSA-DAI 交互过程^[9](取自 OGSA-DAI 工作组发布的《<<Introduction to OGSA-DAI Services >>》):

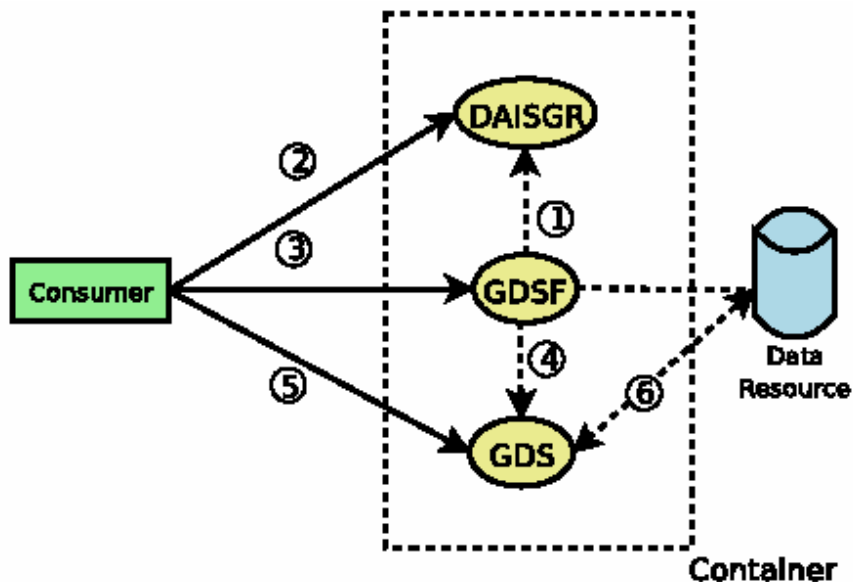


图2.4 典型OGSA-DAI交互过程

DAISGR: 数据接入及集成服务注册(Data Access and Integration Service Group Registry), 用于发布元数据以及元数据所包含的具体数据资源及容量。客户端可以利用 DAISGR 查找到最符合其需求的资源。

GDSF: 网格数据服务工作站(Grid Data Service Factory), GDSF 保持与数据源的持续连接并且包含 DAISGR 不可达的额外相关元数据。GDSF 产生 GDS 进行与数据源的连接和操作。

GDS: 网格数据服务(Grid Data Service), 是一个即发即用的连接, 客户端通过 GDS 与具体的数据源进行连接。

- 1) GDSF 将其元数据及容量信息注册到 DAISGR
- 2) 客户端通过询问 DAISGR 获取可以到达的数据资源信息
- 3) 客户端可以向一个特定的 GDSF 询问详细的资源信息
- 4) GDSF 表现为一个持续的数据元的封装但它不提供直接的数据接入服务, GDSF 创建一个 GDS, 通过 GDS 客户端与所需的数据资源进行连接
- 5) 客户端向 GDS 提交一个执行文档(perform document), 其中包含需要在数据源上进行处理动作和需要返回的数据信息
- 6) GDS 执行具体的查询动作

2.2.3 OGSA-DAI 及其类似项目用于农业信息集成

上面介绍了 OGSA-DAI 的软件框架及工作流程,事实上在网格环境下进行数据整合方面的技术或软件并不仅仅是 OGSA-DAI 这一个,类似的还有美国 DARRA(Defense Advanced Research Projects Agency)开发的 SRB(Storage Source Broker)项目以及 Avaki 公司开发的 Avaki Grid 软件,OGSA-DAI 只是其中的一个代表。

从前两节介绍我们可以看出 OGSA-DAI 构建的是一个松散耦合系统,具体的数据资源可以随时地加入或者退出网络,这当然是很好了,但对于农业信息集成而言却又略显得多余,因为在农业信息系统中一旦数据节点被添加入系统再退出的可能性就比较小了,即使退出,也不会是经常性的,所以 OGSA-DAI 系统构架中关于数据资源发掘这部分没多大应用价值。同时,数据资源发掘部分繁琐的机制会造成查询时间比较长(如前述需要经过六个步骤客户端才能连接到具体的数据库),而实际使用中也有人反映 OGSA-DAI 反应时间慢这一问题。

另一方面,OGSA-DAI 系统开发时面向的主要是广域网环境下所有数据资源的整合,所以在数据资源整合方面包含了关系数据库、XML 数据库及文件系统等三方面的数据资源,而实际分布的农业数据节点主要是以关系数据库的形式存在的,所以 OGSA-DAI 对于农业数据资源的针对性还不是很强,我们需要应用的只是其中的一部分功能,另外,我们可以看到 OGSA-DAI 在进行异构数据集成方面运用的技术是属于分布式数据库系统的 JDBC 技术。

总的来看,网格环境下的数据整合软件重点的工作还是在于广域网环境下动态数据资源发布、数据资源发掘以及与之相关的消息认证方面,在解决底层异构数据集成方面并没有新的工作,而对于农业信息系统而言,这套数据资源发掘的机制并没有太大的实用价值,所以将已有的软件直接应用于农业信息集成还是不太合适,针对性不强。但同时,已有技术中在海量数据传输以及安全认证方面的一些工作却是可以被我们借鉴的,在下一章中,我们将介绍自己设计的系统模型框架。

第3章 农业信息集成系统设计

3.1 我们的目标

在进行系统搭建之前，我们首先来看一下需要解决的主要问题：

1) 实现数据资源整合，或者说是数据透明性，这是整个系统构建的基础。从下至上这又有四层含义：首先是需要屏蔽数据库类型不同造成的数据异构性，参照分布式数据库的定义，我们需要建立一个逻辑上属于同一整体的虚拟数据库并提供标准的接入及查询方式。第二是屏蔽由于地理位置造成的数据库异构性。第三是提供数据库验证的透明性，即用户不需要关心具体的数据分片信息而可以直接获取底层的数据资源，最高一层是实现数据表属性的透明性，即屏蔽由于数据表列名命名不同造成的异构。

2) 实现安全认证，底层的数据资源并不是可以随便访问的，经过授权的组织 and 程序才可以调用底层的数据资源；同时也不是任意的数据资源都可以添加进网络在中。在上面讨论分布式系统时提到过如何将一个新的数据库节点添加进已有的系统的问题，而安全认证机制就是为了解决这一问题的。另一方面，由于我们需要构建的是一个松耦合系统，认证的过程不应该过于复杂。

3) 实现数据节点的方便接入，这需要建立在上述安全认证的基础上。

4) 实现海量数据的可靠传输，前面提到农业数据突出的一个特点就是数据的海量性，这也是为什么分布式数据库系统不适合用于农业数据集成的瓶颈，我们需要确保在目前有限的带宽上数据安全可靠的传输，同时对于传输的速度还有一定的要求。

5) 实现海量数据处理，在具体应用中，有时不仅仅是获取原始数据就可以了，还需要对获取的大量数据进行运算处理，同样我们不能依靠集中式的处理模式，而必须实现分布式计算。

6) 用户任务获取及分析，类似于 OGSA-DAI 系统中的客户端程序的功能，我们需要将用户提交的申请转换为执行文本，其中包含底层数据源可以识别和执行的内容。

7) 提供安全保证，由于客户提交的请求需要以执行文本的形式传递给底层数据资源，这些执行文本的权限必须进行严格的控制，防止安全漏洞

8) 提供方便的客户端接入, 用户可以方便的提交任务, 查询获取所需的资源。

上面主要描述了在构建农业信息集成系统中我们需要解决的八个问题, 而在这八个问题中, 又数异构数据库集成和海量数据处理这两个问题最为核心。

3.2 系统框架设计

根据上述的目标及要求, 我们设计了一个三层体系结构的系统框架:

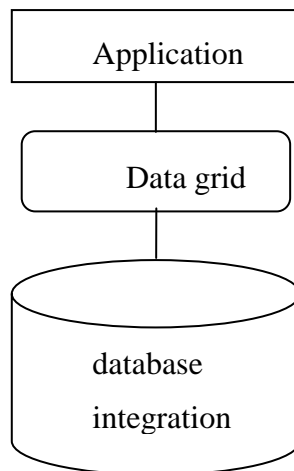


图3.1 农业信息集成系统框架图

1) 最底层是异构数据库集成层, 完成异构数据库的整合并且提供标准的查询接入方式, 对比与 OGSA-DAI, 这一层完成核心层和数据资源层的功能

2) 第二层需要构建一个数据网格系统, 这一层用于确保海量数据的传输及处理, 同时数据节点接入及安全认证机制将在这一层中实现。另一方面, 这一层将作为系统的中央处理器, 根据从应用层获取的执行文档, 选取操作合适的数据源。

3) 最上层, 我们需要构建一个应用服务层, 用于接受用户提交的申请, 并将其转换为可被识别的执行文档, 这一层对应于 OGSA-DAI 系统的客户端及表示层。

3.3 实际系统模型

以上述上层结构为蓝本的实际系统模型如图 3.2 示:

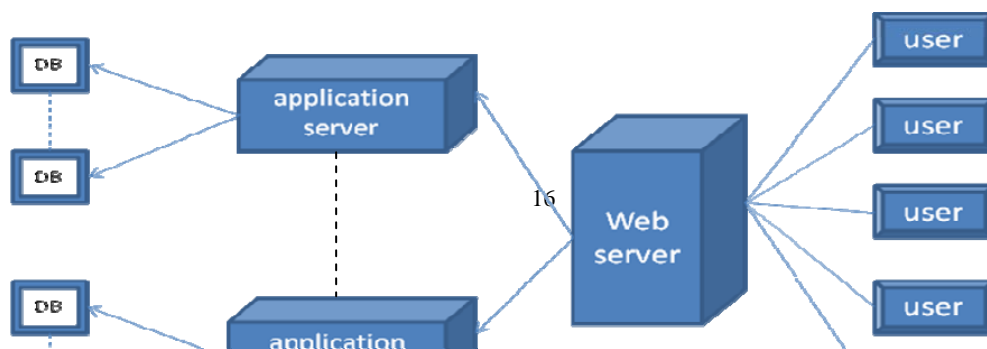


图3.2 实际系统模型

实际系统由四部分组成，第一部分是具体的数据库资源，其中可以包含 ORACLE、SQL Server、MYSQL 等多种数据库，其中数据库转载的操作系统可以是多种操作系统，常用的是 Windows 及 Linux 两种。

第二部分是应用服务器，应用服务器主要完成的功能是数据库整合以及部分的应用层数据处理，它构成了整体框架中异构数据库集成层及部分的数据网格层。事实上这里我们采用了分布式数据库系统的构建方法，一个应用服务器集成部分的数据节点资源构成一个非冗余的紧藕式分布式系统，而整个农业信息集成系统又由许多类似的应用服务器组成一个数据网格。上一章中谈到由于农业数据节点数量较多导致直接采用 DDBS 会使得系统设计及维护成本过高，但如果采用“分布的” DDBS 则会使得问题的复杂程度下降。应用服务器主要是依据地理因素建立的，每个应用服务器负责整合与其地理位置相近一些数据库资源。同时，采用应用服务器的另外一个好处是我们可以将分布式计算的思想引入系统。上面谈到农业信息集成一个和核心的问题是解决海量数据的处理，如果仅仅依靠一个中心处理器对查询的原始数据进行运算处理则会造成处理时间过长以及对硬件配置要求过高。而将应用服务器的概念引入系统后，我们可以将运算处理的过程分配到每一个应用服务器上，而中心服务器仅仅用作数据综合，这样时间成本势必大幅下降。在应用服务器上，我们主要采用 ODBC 技术用于进行异构数据库基层，同时该层部署 Globus 工具用于构建数据网格，应用程服务器我们采用 Linux 系统进行构建，主要原因是 Globus 工具包最初是在 Unix 环境下开发的而且 UNIX ODBC 是一个开源软件。

第三部分是一个 Web 服务器，用于和用户进行交互，将用户提交的申请转换为执行文档提交给特定的应用服务器，这包含两方面的工作，一是将用户提交的任务转换为标准的查询语言，另一方面是根据用户提交的申请选取合适的应用服务器。Web 服务器同时还是整个数据网格层的核心，所有的应用服务器在工作过

程中都是 Web 服务器的底层资源。而安全认证机制也被部署在 Web 服务器上，只要通过安全认证的应用服务器就可以被 Web 服务器访问到，关于这一点我们将在下面具体介绍。当一个新的数据节点需要添加进系统时将有两种方式，一种是直接将数据节点添加进与其相邻的应用服务器，即直接加入某个 DDBS；另一种方式是新建一个应用服务器，将数据库加入改 DDBS，然后再将应用服务器添加进网络。我们采用了 apache 服务器用于提供 web 交互，同时利用 PHP 脚本语言对用户提交的申请进行处理，在这一层上也部署了 Globus 工具包用于进行数据网络的构建，Web 服务器我们仍然采用 Linux 服务器进行实现。

上面我们已经了解了整个农业数据系统的组成框架及具体实现模型，下面一将详细介绍系统的实现。

第4章 农业信息集成系统的实现

4.1 底层异构数据库整合----ODBC 数据库访问标准

4.1.1 ODBC 接口标准及开发原理

在介绍 OGSA-DAI 时我们曾提到 OGSA-DAI 项目执行底层异构数据库集成时用到的是 JDBC 接口标准，ODBC 是与 JDBC 类似的一个数据库访问标准，不同的是 ODBC 基于 C 语言实现。ODBC 是开放式数据互连 Open Database Connectivity 的缩写，它是微软公司开放服务体系 WOSA(Windows Open Services Architecture)中关于数据库的一个组成部分，它建立了一组规范及访问数据库的标准 API，一个基于 ODBC 的用户程序对数据库的操作将不依赖于具体的 DBMS，而是通过 ODBC 驱动的形式与数据库进行交互。使用 ODBC 开发应用系统的结构图如图 4.1^[10]：

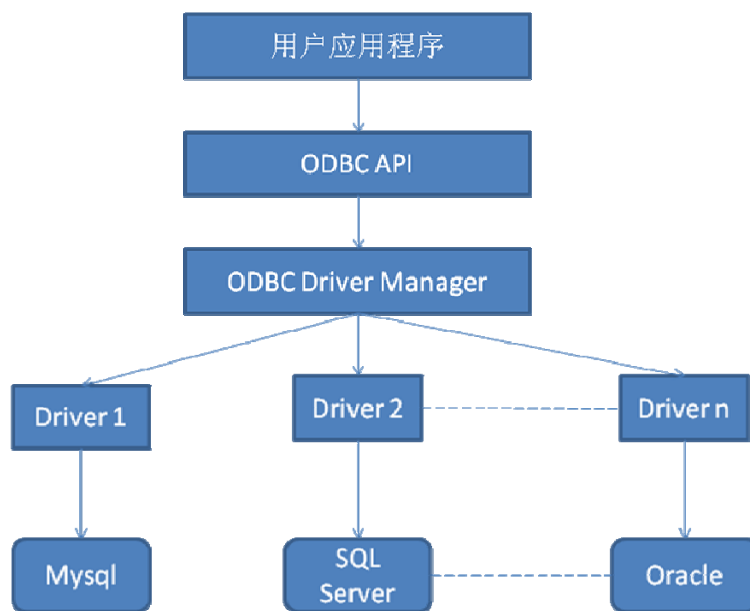


图4.1 ODBC应用系统体系结构图

整个系统由用户应用程序、驱动程序管理器、数据库驱动程序及数据源四部分组成。

应用程序主要用于提供用户界面和进行服务处理，应用服务程序调用标准 ODBC API 接口及 SQL 语句对底层数据库进行操作。一个完整的 ODBC 程序需

要包含六方面的内容：一是连接数据库；二是发送查询语句；三是分配查询返回结果的存储空间及数据格式；四是获取数据库查询结果；五是对结果进行处理并返回用户；六是断开连接。

驱动程序器是用于管理具体到数据库驱动程序的，主要管理应用程序与驱动程序之间的通信。同时驱动管理器需要转载 ODBC 驱动程序、选择和连接正确的驱动程序、管理数据源、检查 ODBC 调用参数的合法性及记录 ODBC 函数的调用等。ODBC 驱动程序管理器可以建立、配置或删除数据源并察看当前系统安装的数据库驱动程序。

数据库驱动程序是应用程序操作底层数据库的通道。实际上应用程序不能直接操作底层的数据库，所有的请求都是通过调用具体 DBMS 对应的驱动程序来实现的，每个不同的数据库都会有自己的数据库驱动程序，其中包含一些与数据库相关的库文件，如果需要操作不同的数据库，这就需要动态的链接到不同的驱动程序上。

最底层的是具体的数据源，ODBC 给每个被访问的数据库分配一个唯一的数据源名称 DSN(Data Source Name)并且自动的匹配所需的底层软件，在一个连接过程中，DSN 就是实际数据库资源的代理，实际的资源对于用户来说是透明的，用户无需知道具体的 DBMS 及有关的驱动程序。

使用 ODBC 进行数据集成开发的优点在于 ODBC API 可以在所有的主流操作系统上运行；同时 ODBC 提供了数据库的动态绑定，这使得客户端可以轻松的连接到一个数据源上并且在使用过程中不用担心重新连接的问题。

4.1.2 用 ODBC 实现的数据库集成

在实验环境下，我们利用 ODBC 搭建了一个应用服务器，实现了一些主流数据库的集成。实验用机器共两台，一台在 Windows 操作系统下用于部署数据库，在该系统中我们安装了三类不同的关系数据库：Oracle、SQL server 及 MYSQL。另一台在 Linux 系统下配置了 Unix ODBC 用于进行数据级联。具体的软件及系统版本表 4.1 示。

先以简单的 SQL Server 的 ODBC 驱动安装为例介绍一下数据库驱动的安装及测试过程：

- 1、在 Unix ODBC 主页 <http://www.unixodbc.org> 查找并下载 Sql Server 的驱动程序 freetds-0.62.4

表4.1 系统配置表

Windows 系统	Linux 系统
MS professional Service pack 2	Fedral core 6 Linux 内核版本 2.6.20-1.3002.fc6xen Unix ODBC 版本 2-2-11
SQL Server 2000	ODBC 驱动: freetds-0.62.4
Mysql 5.0.20-nt	ODBC 驱动: Mysql-connector-odbc-3.51.24-0.i386.rpm
Oracle 10.1.0.2.0	Oracle 简单客户端及 ODBC 驱动: oracle-instantclient-basic-10.2.0.3-1.x86_64.rpm oracle-instantclient-sqlplus-10.2.0.3-1.x86_64.rpm odbc-oracle-3.1.0-linux-x86-64

2、进行驱动程序的安装，注意在安装之前输入参数

```
./configure--prefix=/usr/local/freetd--with-unixodbc=/usr/local/unixODBC--with-tdsver=8.0
```

3、在 Windows 系统防火墙例外中添加端口 1433

4、修改 freetds 的配置文件 freetds.conf 将数据库的信息添加进文件中：

```
[MyServer2k]
```

```
host=166.111.137.19 (host 代表 SQL Server 服务器的 IP 地址)
```

```
port=1433
```

```
tds version=8.0
```

5、利用 tsql 测试与数据库的连接，-U 后的参数是数据库使用人名称，-P 后是密码，use master 表示使用的数据库名称是 master，结果如图 2.4 示：

A terminal window titled 'root@zhihui:~' with a menu bar containing '文件(E)', '编辑(E)', '查看(V)', '终端(T)', '标签(B)', and '帮助(H)'. The terminal content shows the execution of 'tsq1 -H 166.111.137.19 -p 1433 -U wyx -P minipc', locale information, and a successful SQL query: '1> use master', '2> select * from person', '3> go'. The query result is a table with columns 'name', 'id', and 'credit', containing one row: 'wanyxu', '2004011582', '123124563'. The session ends with '1> exit' and the prompt '[root@zhihui ~]#'.

```
[root@zhihui ~]# tsq1 -H 166.111.137.19 -p 1433 -U wyx -P minipc
locale is "zh_CN.UTF-8"
locale charset is "UTF-8"
1> use master
2> select * from person
3> go
name    id          credit
wanyxu  2004011582  123124563
1> exit
[root@zhihui ~]#
```

图4.2 tsq1测试图

6、修改 Unix ODBC 中的配置文件，将 SQL 数据库的信息及驱动程序信息添加进 ODBC 驱动管理器中，ODBC 管理配置文件共有两个，分别是 `odbcinst.ini` 与 `odbc.ini` 文件，在 `odbcinst.ini` 写入如下内容：

```
[TDS]
Description = MS-SQLServer
Driver = /usr/local/freetds/lib/libtdsodbc.so
Setup = /usr/local/freetds/lib/libtds.so
FileUsage = 1
```

在 `odbc.ini` 文件写入如下内容

```
[tst]
Driver = TDS
Server = 166.111.137.19
Database = master
Port = 1433
Socket =
Option =
Stmt =
```

7、利用 `tsq1` 测试通过 Unix ODBC 驱动程序与数据库的连接，查看 SQL Server 的驱动程序是否正常工作，结果如图 4.3 所示：


```

root@zhihui:~
文件(F) 编辑(E) 查看(V) 终端(T) 标签(B) 帮助(H)
[root@zhihui ~]# isql -v tst wyx minipc
+-----+
| Connected!
|
| sql-statement
| help [tablename]
| quit
|
+-----+
SQL> select * from person
+-----+
| name      | id          | credit  |
+-----+
| wanyxu    | 2004011582 | 123124563 |
+-----+
SQLRowCount returns 1
1 rows fetched
SQL>

```

图4.3 测试ODBC驱动程序是否正常工作

经过上述步骤安装及测试则证明 SQL Server 的 ODBC 的驱动程序及数据库都是正常工作的，下面就可以利用 ODBC API 开发数据库的管理程序了。我们实现了对于上述数据库资源的管理程序，下图显示了一个数据资源的查询过程，manage 是一个 ODBC 管理程序，可以对用户提交的 SQL 查询申请进行处理，图 4.4 显示了不同查询语言的输出解果。

```

root@zhihui:/home/zhangwen/testprogram
文件(F) 编辑(E) 查看(V) 终端(T) 标签(B) 帮助(H)
[root@zhihui testprogram]# ./manage "select * from person"
2004011582 wanyxu
connect database tst successfully!
2004011580 shenhu
20031231 wangzhen
connect database mysqltst successfully!
[root@zhihui testprogram]# ./manage "select * from person where name='wanyxu'"
2004011582 wanyxu
connect database tst successfully!
connect database mysqltst successfully!
[root@zhihui testprogram]#

```

图4.4 ODBC管理程序查询示例

从上面的例子我们可以看出管理程序已经实现了对底层数据源的集成，管理程序是使用 ODBC API 函数进行开发的，其工作流程如下图示^[11]：

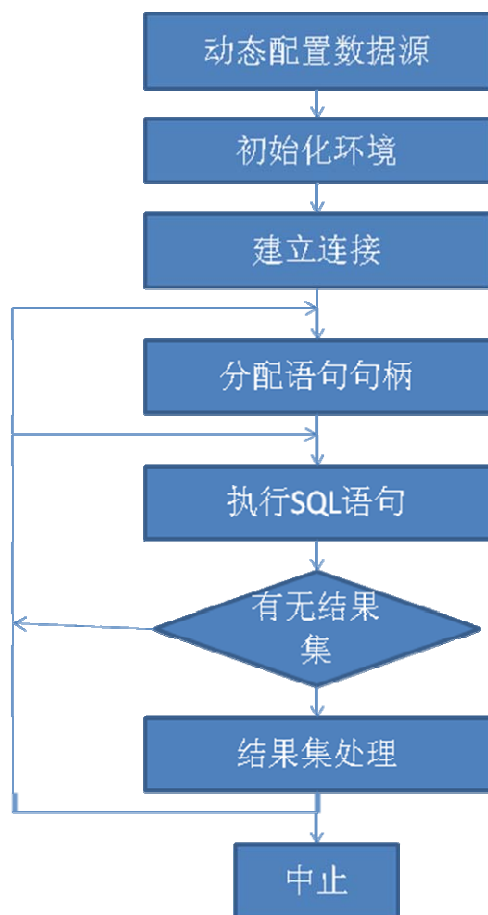


图4.5 ODBC管理程序工作流程

这一部分主要介绍了底层异构数据库集成方面的工作，这是整个农业信息集成系统的基础，下面一节将介绍如何利用 Globus 工具包实现松散耦合的数据网格系统。

4.2 利用 Globus 工具包实现的数据网格

在第二章中我们介绍并分析了 Globus Toolkit 的项目背景及实现的功能，这一部分将重点讨论 Globus 工具包中集成的工具在我们设计的农业信息系统中的应用。其中最主要的有两个工具，其一是 Globus 提供的用户认证，其二是 Globus 嵌套的 Grid Ftp 工具。

4.2.1 Globus 中的安全认证

前面曾多次提到我们需要构建的系统是一个松散耦合的系统，也即是数据资源可以方便的加入系统中去，而这里的问题是如何对新加入的资源进行认证同时资源又如何确认访问者的权限，Globus Toolkit 中嵌入了基于 X.509 数字签名标准的认证机制可以很好的解决这一问题。数字签名采用的是公钥算法，也被称为非对称密钥算法^[12]。非对称密钥算法有两个密钥，一个被称为公钥，一个被称为私钥，公钥就是发布给所有人的密钥，而私钥是自己保管的密钥。公钥和私钥是算术上相关的两个数，如果密文是通过其中一个密钥加密的，接收方都可以利用另一个对密文进行解密。当一个用户 A 用私钥对自己的信息进行加密时就形成了 A 的数字签名，因为接收方可以用 A 的公钥对密文进行解密，如果解密成功，则证明信息确实是由 A 发的，这样就实现了对 A 的身份认证。Globus 实现的网络安全构架 GSI (Grid Security Infrastructure) 采用的是就基于数字签名的认证方式，主要实现了以下三方面的目标：

- 1) 保证计算网格节点之间的安全通信，包括认证和秘密通信
- 2) 支持跨组织边界的安全通信，从而避免了中央控制的安全系统
- 3) 支持网格用户的单点申请，包括对多种计算资源或对象进行授权的证书机制

Globus GSI 中最重要的就是证书的概念，每个网格用户或服务都是通过证书相互认证的，证书中包含用于确认用户或服务的主要信息。一个 GSI 证书中包含以下四项内容：

- 1) 对象名称，用于认证证书声明的用户或对象
- 2) 一个属于对象的公钥
- 3) 为对象的公钥和身份颁发证书的证书认证机构(Certificate Authority CA)的身份信息。
- 4) CA 的数字签名

在 GSI 中引入了第三方认证机构 CA 的概念，CA 用于构建公钥与被认证的对象之间的联系，也就是由第三方颁发对象的公钥和私钥。如果要使得对象的证书和其内容被信任，CA 的数字证书者应该首先被信任，CA 及其证书必须通过非密码方式进行传递，否则系统就无法被信任了。

如果通信的双方都是由同一个受信任的 CA 颁布的证书，那么双方的认证过程如下：

- 1) 首先双方必须获取 CA 的数字证书并且信任该 CA

- 2) 假设由 A 发起对 B 的连接，A 将自己的证书传递给 B
- 3) B 首先必须通过检查 CA 的数字证书对 A 发来的证书进行认证，确认 A 的证书确实是由 CA 签发的且没有经过篡改
- 4) 一旦 B 确认证书是 A 的（也就是 A 的公钥），则 B 需要将 A 的证书发过来的那个人确实是 A，B 会用 A 的公钥加密一段信息然后将其发给 A
- 5) A 用自己的私钥对信息进行解密并用自己的私钥对信息进行加密后发回给 B
- 6) B 用 A 的公钥对返回的密文进行解密，如果这与之前 B 发出的信息一致的话，则 B 对 A 的认证过程结束。
- 7) 同样 A 需要对 B 的身份进行确认，确认的过程与上述相同。
- 8) 经过对于证书的确认之后，A 与 B 知道对方都是可以信任的。

以上便是 Globus 安全认证机制的实现过程，通过这套认证机制数据节点就可以动态的添加进网络中去了。

4.2.2 Globus 用户代理机制

在建立了 GSI 安全认证机制的基础之上，Globus 还对标准的 SSL 协议作了扩展，开发了 Globus 用户代理认证，减少了用户需要输入口令的次数。当一个用户 A 需要登入到用户 B 的机器上使用 B 的资源时，在基于 SSL 的 SSH 连接方式下用户 A 必须输入 B 的口令才可以登入，但由于有了证书认证的机制之后，A 只要创建一个代理就可以省去输入 B 密码这一步骤。A 创建的代理包含一个新的证书及私钥，证书中包含 A 的身份信息。整个证书的签发过程如下图示^[5]：

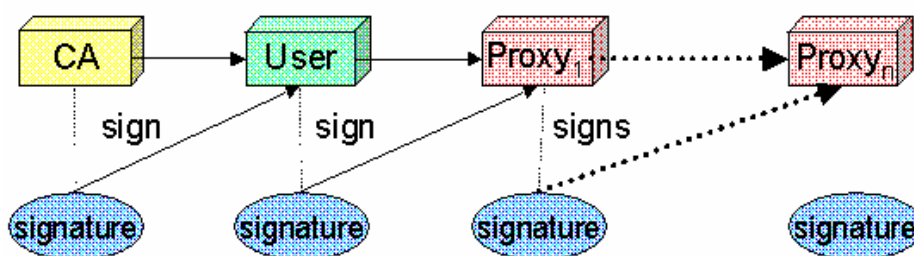


图4.6 用户证书及代理签发过程

- 1) 先由 CA 中心（我们定为 Web Server）生成一个数字证书，即 CA 证书，该证书包含 CA 的具体信息将作为整个网格搭建的基础，该证书会以一个压缩文件的形式存在。

2) 用户通过电子邮件的形式向第三方 CA 申请加入网格 (或是通过管理员直接配置), 经过管理员认证用户信息后, 由 CA 中心将压缩的数字证书发给用户。

3) 用户首先以 root 身份登入系统, 将 CA 证书安装在特定的文件路径中。

4) 安装完 CA 证书后, 用户需要以 root 身份申请一个主机证书, 即让你的主机能添加进网格中并且让网格中心能够识别出来你是谁。

5) 用户将申请主机证书的文件传送到 CA 中心, 由 CA 中心对申请进行处理, 产生用户的主机证书并将其发还给用户。

6) 用户以 root 身份将签发的主机证书安装在特定的文件夹中, 该证书信息对于其他用户来讲只具有读的权限。

7) 用户以一般用户的身份申请一个用户证书, 申请用户证书的身份就是以后申请代理的身份, 该证书的作用主要规定了使用代理的用户, 同样, 用户需要将申请提交给 CA 中心, 由 CA 中心对代理进行签发。

8) 用户以申请者的身份将签发后的用户证书安装在制定文件夹中, 下面就可以申请用户代理了。

9) 在 Globus 中有一个 Grid-mapfile 文件可以对代理访问的权限进行配置, 比如 A 的代理将访问 B, 则 B 的 Grid-mapfile 文件将会设置 A 的代理将会以什么登入 B 使用其资源。

在实际搭建系统中, 我们以 Web 服务器作为 CA 中心, 各台应用服务器分别具有自己的主机证书、用户证书。同时在 Web 服务器中以非 root 的身份向 CA 中心 (即本机) 申请主机证书及用户证书, 那么就可以实现 Web 服务器使用代理对各个应用服务器的资源进行访问, 在各台应用服务器中可以对本机的 grid-mapfile 文件进行配置, 设置两个用户: 一是 Web 服务器登入本机的用户, 举例来说 Web 服务器上有三个用户, 分别是 root、wan 以及张, 在前期申请 Globus 代理时是用 wan 用户申请的, 那么访问本机的 Globus 代理就必须设置为 wan 用户; 二是设置代理在本机上具有的用户权限, 该权限一般为部署分布式数据库查询处理程序的用户权限, 这样 Web 服务器就可以以代理的身份直接操作数据库处理程序了, 就完成了数据网格的动态搭建于配置, 应用服务器可以方便的加入网格中, Web 服务器可以直接调用底层的数据资源。

4.3 Apache 与 PHP 构成服务平台

Apache 是目前世界上市场占有率最高的 Web 服务器，它可以适用于任何平台。我们利用 Apache 服务器提供基本的用户交互，通过 Html 网页的形式接收用户提交的申请。PHP 是一种脚本语言，与微软的 ASP 类似，PHP 可以嵌入于 Html 文件中在服务端执行一定的代码或程序。PHP 代码本身对于用户是隐藏的，用户看不到 PHP 文件执行的过程，当用户在 Html 文件上执行特定的操作时会触发 PHP 脚本的执行，比如对于表单的提交动作。

在上一节中我们描述了 Globus 证书代理机制在农业信息集成系统中的应用，其功作主要是为 Web 服务平台服务的，首先需要注意的一个问题是 php 脚本的使用用户必须与 Globus 的代理用户一致。上面谈到我们实际是通过 Web 服务器上部署的 Globus 代理去操作底层的数据资源的，而实际访问应用服务器的程序是写在 PHP 脚本中的，PHP 脚本通过 Globus 代理调用 Globus-job-run 命令将用户执行文档提交到应用服务器，直接操作其分布式数据处理程度执行底层数据库的查询操作。而前面谈到应用服务器在 Grid-mapfile 对于访问用户进行设置，期中之一需要设置访问用户。故可以对应用服务器进行访问的 Web 服务器用户帐号是确定的，如上一节提到的 wan 用户，也只有该用户可以利用代理将执行文档提交给应用服务器，故 PHP 文件的执行用户必须与 Globus 代理用户一致，否则就无法运用代理机制访问下一层应用服务器。

4.4 系统工作流程

经过上述的描述，我们已经实现了整个农业信息集成系统的搭建，这个系

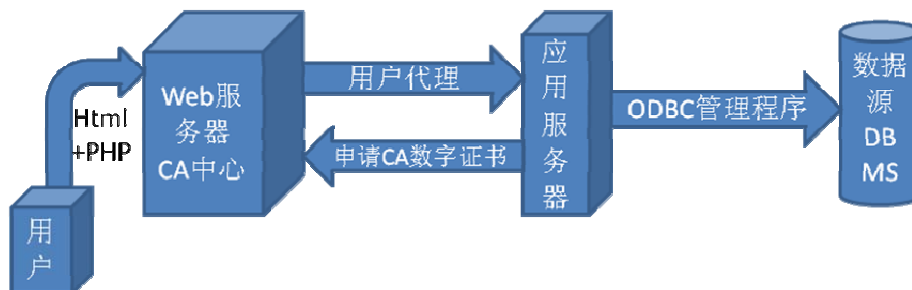


图4.7 系统实现框架图

统包括一个布置有 Globus 的 Web 服务器，一个用于接受用户任务申请的主页

以及包含处理程序的 PHP 文档。Web 服务器与应用服务器组成数据网格，Web 服务器作为证书认证的 CA，每个应用服务器都拥有 Web 服务器颁发的数字证书，Web 服务器上一个特定用户申请一个代理，通过这个代理 Web 服务器可以调用每一个应用服务器的资源。应用服务器通过 Unix ODBC 管理一些具体的数据库资源，构成一个简单的分布式数据库系统，系统实现框架图如下：

从用户提交申请到得到查询结果的流程图如下：

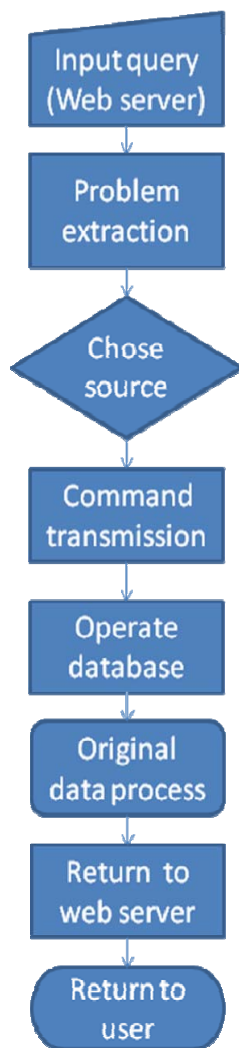


图4.8 农业信息集成系统工作流程图

- 1) 类似于一般的搜索引擎，客户通过 Web 服务器提供的主页提交任务申请。

2) 隐藏于 Html 文档中的 PHP 程序对用户提交的任务申请进行处理产生执行文档并根据用户需求选取合适的应用服务器。

3) PHP 程序利用 Globus 工具将任务提交给下一层的应用服务器,这一过程需要用到 Web 服务器的代理,所以必须在 Web 服务器中还有一个后台程序以固定的周期产生代理,通常是每 12 小时申请一次(因为 Globus 默认的代理有效时间是 12 小时)

4) 当应用服务器接受到查询任务后,应用服务器执行本机上部署的处理程序调用底层的数据库资源。实际上执行处理程序的是 Web 服务器本身,通过代理证书的方式,Web 服务器被授权调用应用服务器上的资源,包括数据库的处理程序。在应用服务器上部署一个动态获取其数据库资源信息的程序,可以动态的取得由 ODBC 分配的 DSN 名、数据库用户帐户以及密码等信息,这些数据库信息会在应用服务器处理程序执行的过程中动态的添加进程序中。

5) 应用服务器根据不同的服务对获取的原始数据进行处理,然后通过 Grid Ftp 将数据返回给 Web 服务器。

6) Web 服务器对收到的各个应用服务器返回的数据进行综合后以 Html 页面的形式返回给用户。

4.5 目前实现的功能

目前底层的数据库级联以及数据网络的搭建已经完成,由于 Oracle 数据库驱动程序是付费软件,试用期已过,所以可用于级联的数据库为 SQL 及 Mysql 两个。在 Web 应用层实现的功能还比较少,实现了一个基本的 Web 搜索的功能,其页面如下页图 4.9 所示,在输入框中可以输入要查找的信息,然后服务器会调用底层资源搜索相关的信息,当什么都不输入时,服务器默认输出所有的底层数据源信息,在图 4.10 中显示了什么都不输入的返回结果,而图 4.11 则显示了输入定查找对象的输出结果:

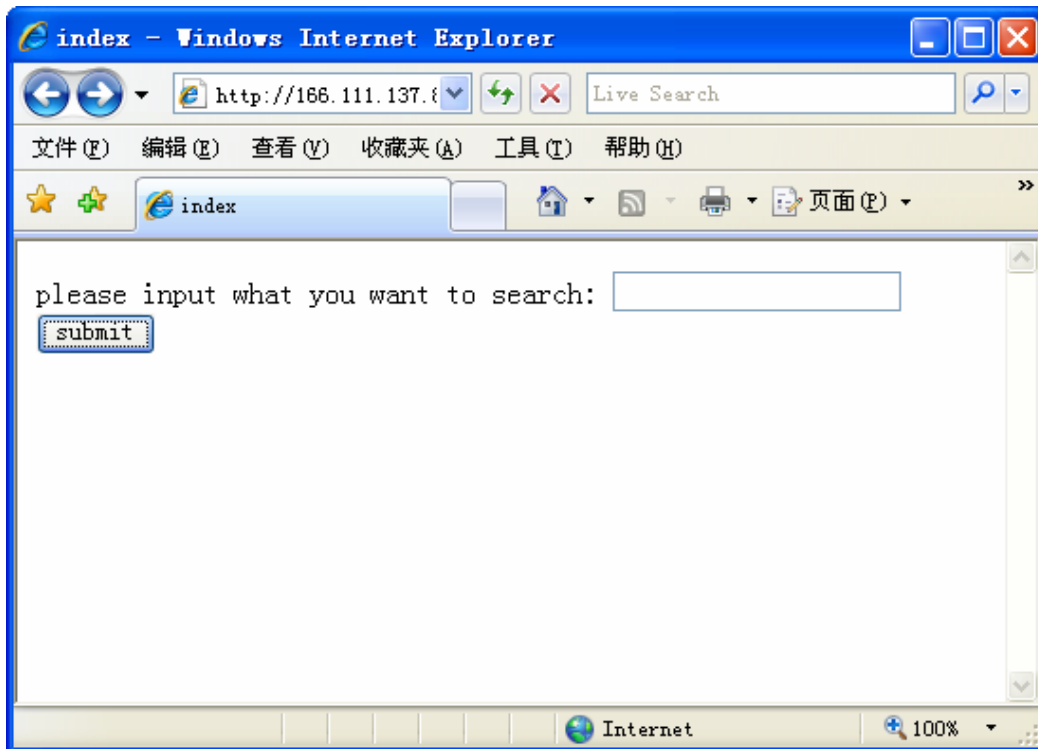


图4.9 输入界面

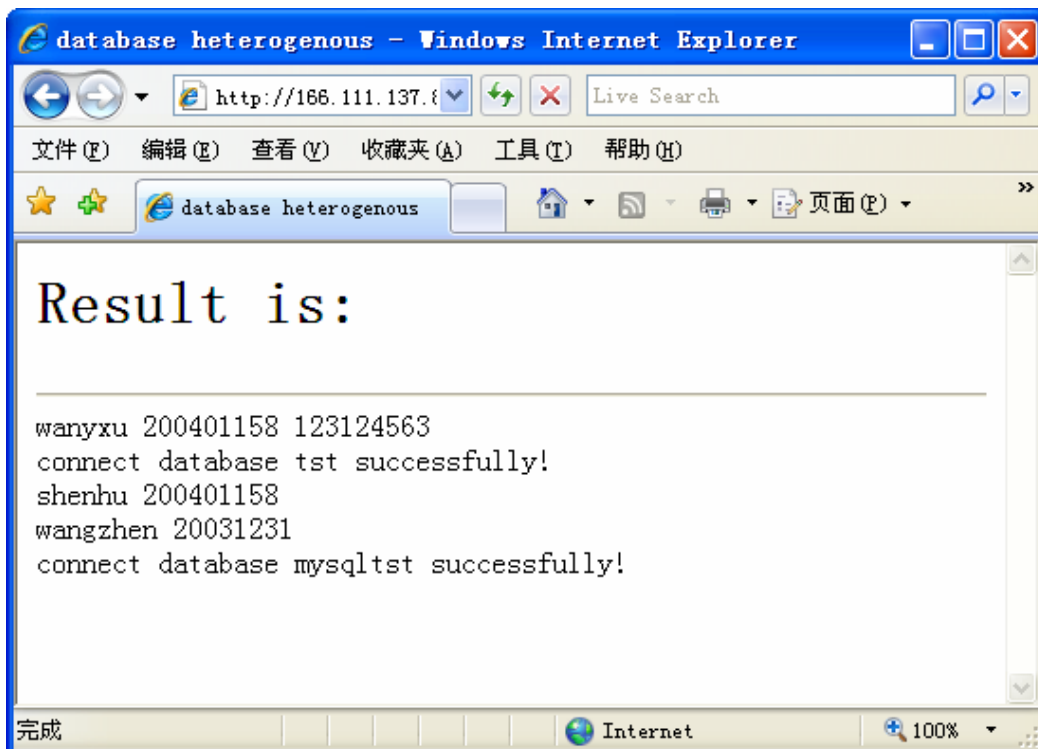


图4.10 所有的数据信息

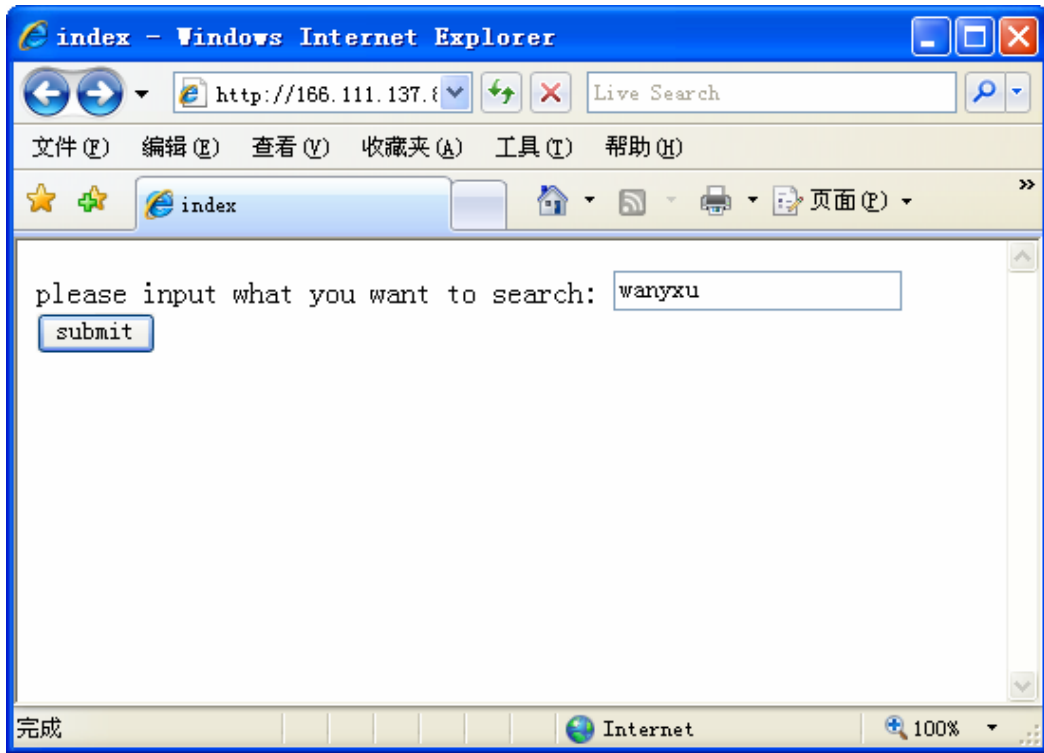


图4.11 针对指定信息查询

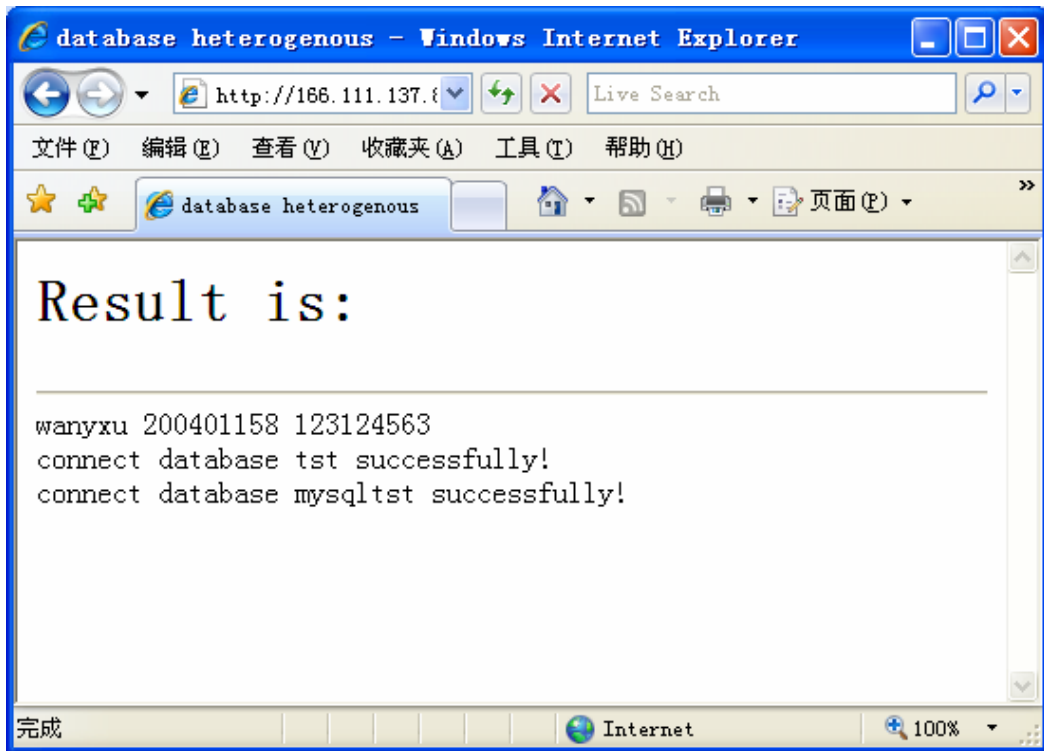


图4.12 指定信息查询结果

第5章 系统评估及未来工作展望

5.1 系统性能评估

总体来说，目前搭建的信息集成系屏蔽了底层数据源的异构性、实现了数据网络的松散耦合性并完成了一些具体的应用功能，解决了系统的框架搭建问题。系统总体构架过程较为方便，安装及配置过程也不是很复杂。

在实验过程中主要呈现出来的问题有两个，首先是查询时间的问题，目前的 Web 服务器采用的是同步的查询模式，而 Web 服务器访问应用服务器资源时首先要经过 Globus 的代理认证才可以将执行文档传递到应用服务器，而在非局域网的环境下 Globus 的认证时间还是比较长的，首先说明一下 Globus 的认证是针对在域名的认证，而不是 IP 地址。在实验中我们开始将 Web 服务器及应用服务器的 DNS 设置为 166.111.8.28，结果认证时间非常长，用户提交申请之后需要等待 30s 左右才能返回结果，而之后我们把 DNS 设置为 166.111.131.46，即 FIT 本地的 DNS 后，整个查询时间缩短到了 2s 左右，可见在广域网环境下时间消耗还是一个比较大的问题。这个问题目前还没有想到比较好的解决办法，除非采用异步的查询方式，但异步查询如何实现信息的实时性又是一个问题。

另一个问题是 ODBC 驱动程序及 Globus 的配置安装过程没有一个可视化的界面，当然也包括 Globus 数字证书及代理的申请过程。这样就需要管理员去配置所有的 Web 服务器、应用服务器及底层数据源，在实验环境下这一点可以做到，但在实际工作中却很难实现，所以，如果要将系统投入实用，最好有一个可视化的配置界面。

5.2 未来工作展望

本项目属于农业信息集成课题的一部分，我们与农科院都承担了本项目的一部分工作。农科院主要承担的是文本的搜索，也就是利用网络爬虫程序将网络上所有可发现的与农业相关的信息资源抽取出来放置在数据库中，而这些数据库实际相当于我们底层的数据资源。目前我们虽然实现了异构数据的集成，但在应用

服务上所提供的功能还非常少，下面可以考虑的是在 **Web** 服务层加入更多的应用服务功能，并且考虑将我们与农科院两部分的工作整合在一起。同时，上面提到的关于安装配置过程的可视化界面也是一个需要解决的问题。

插图索引

图 2.1	分布式数据库系统模型结构	7
图 2.2	Globus 框架图	9
图 2.3	OGSA-DAI 组成结构	10
图 2.4	典型 OGSA-DAI 交互过程	12
图 3.1	农业信息集成系统框架图	16
图 3.2	实际系统模型	17
图 4.1	ODBC 应用系统体系结构图	19
图 4.2	tsql 测试图	22
图 4.3	测试 ODBC 驱动程序是否正常工作	23
图 4.4	ODBC 管理程序查询示例	23
图 4.5	ODBC 管理程序工作流程	24
图 4.6	用户证书及代理签发过程	26
图 4.7	系统实现框架图	28
图 4.8	农业信息集成系统工作流程图	29
图 4.9	输入界面	31
图 4.10	所有的数据信息	31
图 4.11	针对指定信息查询	32
图 4.12	指定信息查询结果	32

表格索引

表 4.1 系统配置表.....	21
------------------	----

参考文献

- [1] 夏靖波、刘颖、汪胜荣.《网格原理与开发》西安：西安电子科技大学出版社，2006年，5~15
- [2] 何峰，“网格环境下异构数据库整合研究”：[硕士学位论文].北京：清华大学计算机科学与技术系，2007年5月
- [3] 毛国君.《高级数据库原理与技术》.北京：人民邮电出版社，2004年8月. 8~20
- [4] 王珊、萨师焯.《数据库系统原理及概论》.北京：高等教育出版社，2006年5月. 353~357
- [5] Globus 官方网站 <http://www.globus.org>
- [6] 都志辉 陈渝 刘鹏.《网格计算》.北京：清华大学出版社，2002年，62~70
- [7] OGSA-DAI 项目官方网站 <http://www.ogdadai.org>
- [8] OGSA-DAI 项目工作组，“OGSA-DAI Status and Benchmarks” .2005年
- [9] OGSA-DAI 项目工作组，“The Design and Implementation of Grid Database Services in OGSA-DAI” .2005年
- [10] 王珊、萨师焯.《数据库系统原理及概论》.北京：高等教育出版社，2006年5月. 249~254
- [11] 王珊、萨师焯.《数据库系统原理及概论》.北京：高等教育出版社，2006年5月. 249~254
- [12] Atul Kahate.《密码学与网络安全》(，邱仲潘).北京：清华大学出版社，2005年9月. 35~43

致 谢

在本学期毕设项目开展期间，指导老师曹军威研究员给予了我最耐性的指导和帮助，为我指明了课题并提出了具体的要求，在项目进行过程中多次就进展情况对我进行指导，使得我的毕业设计最终能顺利开展下来，首先感谢曹老师对我的帮助。

实验室的几位师兄与同学我毕设期间也给予了我很多的支持，特别感谢张文师兄，是他给我提供了利用 ODBC 集成底层数据资源的思路并耐心解决了我很多程序中遇到的问题；感谢王震师兄，在毕设期间他为我提供了往年的资料并且在中期评估及最后的毕业论文中给予了我许多指导；感谢李俊伟、吴充、沈虎同学在毕设期间与我的讨论。

声 明

本人郑重声明：所呈交的学位论文，是本人在导师指导下，独立进行研究工作所取得的成果。尽我所知，除文中已经注明引用的内容外，本学位论文的研究成果不包含任何他人享有著作权的内容。对本论文所涉及的研究工作做出贡献的其他个人和集体，均已在文中以明确方式标明。

签 名：_____ 日 期：_____

附录 A 外文资料调研报告

Heterogeneous database integration

The phrase “Great Global Grid” is not a new concept to most computer engineers. A definition by Ian Foster is “Coordinated resource sharing and problem solving in dynamic, multi-institutional virtual organizations”. As the primary aim of grid is to found a new infrastructure which can using multiple distributed resources to cooperatively work on a single application. It will provide large computing and data-storage ability. Both commercial and scientific world has shown a lot of interests in this domain recently. In this article we mainly focus on one aspect of the applications---heterogeneous data integration. In the first part we’ ll talk about what is data grid and data integration. Then we’ ll see three main software and their framework. And the last part is the backdrop of this study.

According to the different services provided by grid. We can divide the grid into two categories. One is computing grid, which aim at making use of distribute computing resource and offer better problem solving ability. The other is data gird, which provide access to distribute storage sources. If we think about the two services in a sort, we can easily make a conclusion that data grid is the foundation of computing grid. As a matter of fact the concept of data gird is brought out when the scientists wanted to transfer the large amount of data produced by scientific observations. However these days the data grid has found more than that use, especially in the commercial world. Take an example of a large supermarket, lots of sale records are been produced and stored in her distributing shops all around world. And the sale manager want to know the exact data of every shop, however those shops are not so organized that they use different system and different database to record data. So he need to study about heterogeneous data integration.

The purpose to establish heterogeneous data integration system is to provide a virtual database which users can get the data he needs and don't have to contact with the real database one by one. We call it data grid transparencies. From down to top it means:

1. Access data without knowing the type of storage, which stands for "storage resource transparency". It's the bottom layer, aiming at combine different types of physical database into a logical set of sources and offering standard interface for accessing.

2. Access data without knowing the location---- "Storage location transparency".

3. Find data without knowing the identifier---- "data identifier transparency". There are four types of data identifiers: Unique name, descriptive name, collective name and physical name.

4. Retrieve data using your preferred API.

5. Provide transformations for any data collection.

At present there are three software which provide data grid management: SRB (Storage Source Broker), Avaki Grid, and OGSA-DAI. SRB program was started in 1995 founded by DARPA (Defense Advanced Research Projects Agency, US). It has features to support collaborative management of distributed data including: controlled sharing, publication, replication, transfer, attribute based organization, data discovery, and preservation of distributed data. In the year 2001, Nirvana Storage got the commercial copyright of SRB and from then on released some edition. Avaki is an enterprise application integration software product designed for Enterprise Information Integration System.

OGSA-DAI is a project conceived by UK Database Task Force and is working closely with the Open Grid Forum DAIS-WG. OGSA stands for "Open Grid Source Architecture" while DAI means "Data Access Integration". The main aim of the program is for scientific use "To contribute to a future in which scientists move away from technical issues such as handling data location ... instead focus on application specific data analysis and processing". This middleware bases on Globus and has provided control

and access to relational and XML database management system currently. The framework, however, has been designed to allow other data sources such as file systems to be accessed through the same interfaces.

Figure 1 shows the framework of OGSA-DAI:

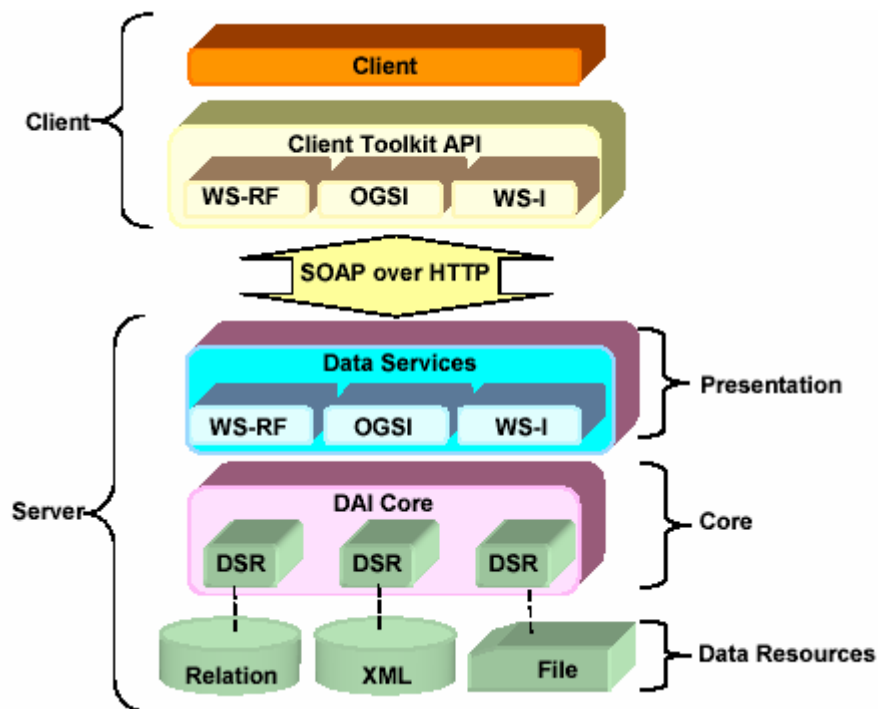


Figure 1 Schematic representation of OGSA-DAI

1. The presentation layer accepts message from clients and decide which Data Service Resource (DSR) in the Core Layer is the proper one to submit.

2. The core layer consists of a set of DSR. Each DSR implements the core DAI functionality which includes overseeing the coordination of the activities for a specific Data Resource. A DSR may also expose additional capabilities such as data transport-related operations and can also cache data for retrieval by third-parties.

3. The client toolkit (CTk) API has been refactored to abstract away the differences between the different messaging infrastructures.

Figure 2 shows a typical OGSA-DAI interaction.

4. Data Access and Integration Service Group Registry (DAISGR) is a service allowing other services to publish metadata about any data resources they represent and the capabilities they expose. A client can

thus use a DAISGR to identify, by querying its registered metadata, a resource provider that best satisfies its needs.

Figure 2 below shows a typical OGSA-DAI interaction.

Grid Data Service Factory (GDSF) acts as a persistent access point to a data resource and contains additional related metadata that may not be available at a DAISGR. A GDSF creates GDSs to access and manipulate data resources.

Grid Data Service (GDS) acts as a transient access point to a data resource. It is through a GDS that a client interacts with a data resource.

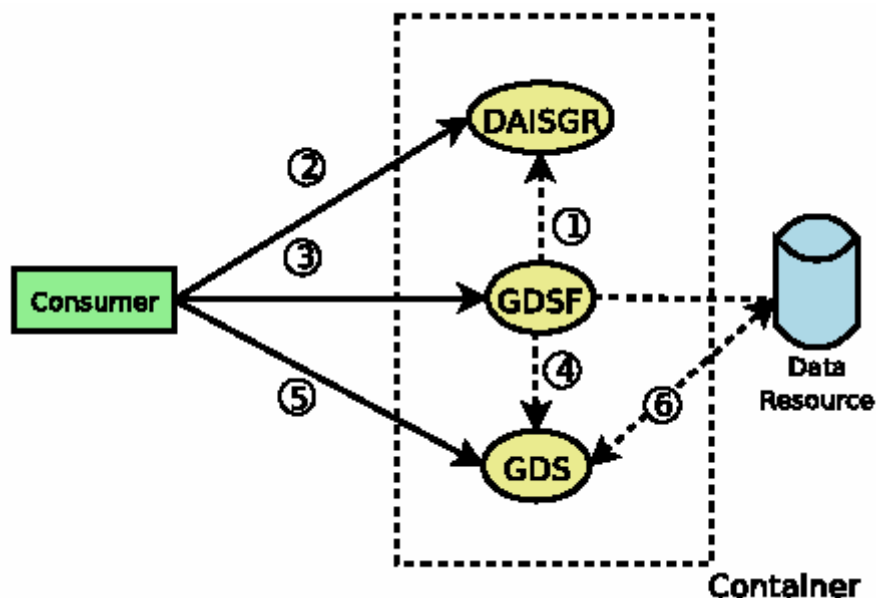


Figure2 a typical OGSA-DAI interaction

1) a GDSF may register its service handle with a DAISGR, along with sufficient metadata and capability information to allow service/resource discovery to take place.

2) Clients obtain information about available resources (represented by GDSFs) by querying a DAISGR. They can then ask for detailed information, e.g. the schema of the resource, at a particular GDSF of interest.

3) A GDSF, in effect, acts as a persistent Grid-enabled wrapper for a data resource but does not provide direct access to that data resource.

Access to a data resource requires the creation of a GDS through the GDSF' s Factory portType as specified in OGSI.

4) GDSs are transient services created at the request of clients who wish to access a data resource. Data resource access is done through a single document-based operation provided by the GDS.

5) A client submits a perform document to the GDS which contains the sequence of activities to be executed on that data resource or the resulting data.

6) The activities that can be executed by the GDS are defined when a GDSF is configured. The inner workings of a GDS are examined in more detail in the next section.

Our program based on the study and research on agriculture data grid of China. One goal of the project is providing a middleware for agriculture data integration and data digging. The main aim of the program is establishing a Semantic Grid for agriculture data searching, sharing, discovering and studying. For the phase of the project just completed, it' ll be a standard interface and management founded for agriculture scientific research.

References

- [1] OGSA-DAI Status and Benchmarks. OGSA-DAI project group.
- [2] The Design and Implementation of Grid Database Services in OGSA-DAI. OGSA-DAI project group.
- [5] OGSA-DAI project official website. <http://www.ogsadai.org/>
- [6] Grid Data Management Systems & Services. VLDB Tutorial Berlin, 2003.
- [7] Heterogeneous Database Integration in Grid Environment. He Feng, tsinghua, 2007.
- [8] Globus alliance official website. <http://www.globus.org/>
