赛百平台下联邦模式虚拟组织管理与信任评价机制研究

Federal Virtual Organization Management and Trustiness Evaluation in CyberInfrastructure

（申请清华大学工学硕士学位论文）

培 养 单 位 ： 自动化系

学　　科 ： 控制科学与工程

研 究 生 ： 王 震

指导教师 ： 曹 军 威 研 究 员

二〇一〇年五月

赛百平台下联邦模式虚拟组织管理与信任评价机制研究　　王　震

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：

清华大学拥有在著作权法规定范围内学位论文的使用权，其中包括：（1）已获学位的研究生必须按学校规定提交学位论文，学校可以采用影印、缩印或其他复制手段保存研究生上交的学位论文；（2）为教学和科研目的，学校可以将公开的学位论文作为资料在图书馆、资料室等场所供校内师生阅读，或在校园网上供校内师生浏览部分内容。

本人保证遵守上述规定。

（保密的论文在解密后遵守此规定）

作者签名：_____ 导师签名：_____

日　　期：_____ 日　　期：_____

摘　要

随着科学研究的逐渐深入，现代科学研究越来越依赖于实验仪器，计算机和网络交互，很多重大科研项目例如黑洞合并、气候变化及基因测序都需要不同领域的研究人员通力合作，依赖分布在世界各地的计算机，实验仪器和网络系统进行数据共享和模拟计算。因此为科学应用构建计算基础架构CyberInfrastructure(CI)以帮助科研人员动态创建虚拟组织，实时，按需的重新整合分布在不同地点和组织的各类资源就成了研究人员首要解决的问题，而这是传统网格技术所无法应对的。

本文以国际科研合作项目 LIGO 为实际应用背景，重点研究了如何在赛百平台中构建动态，可信任，可扩展的多级虚拟组织这一问题，并提出了基于委员会的成员评价方法（CMESM，Committee-based Member Evaluation and Selection Method）和联邦模式多级虚拟组织模型，分别解决在虚拟组织构建过程中出现的成员信任问题和可扩展性问题。CMESM 由代表和委员会两层组成。首先代表根据各自认知情况对申请成员进行单独评估，并递交委员会；然后委员会对各个代表意见进行有权重的综合计算，最终得出全面而准确的成员评价意见，为管理员提供参考。这种方法综合了虚拟组织内部成员意见，表现了虚拟组织对申请成员的个性化要求。仿真实验证明，CMESM 能很好的抵御共谋欺诈行为，并切实提高了虚拟组织内的服务质量。为了实现可扩展，安全的多级虚拟组织管理，本文参考现实社会中的组织结构，提出基于联邦模式的多级虚拟组织模型。这种模型核心思想在于保证自底向上的权限分布，即子虚拟组织对申请加入父虚拟组织的成员具有高度否决权。这样可以保证子虚拟组织在动态加入其它虚拟组织的过程中其内部安全性不会收到威胁。

最后本文也介绍了基于上述方法和模型开发的动态虚拟组织管理系统，并通过和 Globus Toolkit 工具协同工作，演示了一个赛百平台典型科学应用案例。此案例说明本文提出的方法和模型可以有效的帮助赛百用户构建动态，可信任和可扩展的虚拟组织，能比较好的应对了复杂的科学应用需求。

关键字：赛百平台　虚拟组织　动态管理　联邦模式　基于委员会的成员评价方法

Abstract

As the development of the information technology, modern scientific researches greatly rely on the digital experiment equipment, computers and network. Large number of scientific researches, such as collapsar modeling, dark matter, climate prediction and DNA sequencing, need collaboration of scientists from different areas and sharing of equipments, data and computers distributed in various locations and organizations. Establish a CyberInfrastructure (CI) for all the scientific researches to provide the researchers the trustable, scalable and dynamical resource re-aggreagation, and management service is a significant challenge, which the traditional Grid technology can not solve.

This paper, motivated by an international scientific collaboration, LIGO, focuses on how to establish trustable, scalable and hierachical Virtual Organization (VO) in CI environment. Authors propose a Committee-based Member Evaluation and Selection Method (CMESM) to guarantee the trustiness in resource sharing in VO and a Federal Hierachical VO Management Model (FHVOMM) to ensure the scalability and security of sub-VO in the VO collaboration. CMESM consists of a representative layer and a committee layer. Firstly, the representatives, usually the most experienced or the most important members in VO, judge the applicant individually according to their historical interactions and knowledge. Then the committee selects these individual judgments and gives a final conclusion according to the judement of the representative and the corresponding importance. The conclusion of the committee, which comprehensively represents the opinion of all the members in VO, can judge the applicant accurately, defend large number of collusion cheat behaviors and improve the quality of service in VO, proved by a serial of strict simulation experiments. Besides that, in order to ensure the security and scalability, authors propose the FHVOMM, inspired from the actual social organization architechture. In this model, the privilege is distributed from down to up, that is the sub-VO always has a higher power to deny an applicant of its father-VO. This would ensure the inner security of sub-VO when it dynamically joins in other VOs.

A CI middleware, Dynamical Virtual Organization Management System (DVOMS) is developed to facility the proposed CMESM and FHVOMM. This paper gives a detailed introduction of how to use the DVOMS and a case to show how DVOMS works with Globus Toolkit to finish a typical CI application, including setup a dynamical VO, join in the VO and finally

enable resource sharing among VO members. The case shows DVOMS can help users to establish a dynamical, trustable and scalable VO for a specific scientific research application efficiently.

Key Words: CyberInfrastructure Virtual Organization Dynamic VO Management Federal CMESM

目 录

第 1 章 引言 1
1.1 选题背景-赛百平台的提出及其意义 1
1.2 问题提出-虚拟组织管理和信任机制 4

第 2 章 国内外研究动态及相关技术 7
2.1 VOMS 7
2.2 GUMS 9
2.3 DAC，MAC 和 RBAC 10
2.4 PRERMIS 12
2.5 资源评价方法 12

第 3 章 基于委员会的成员评价方法 14
3.1 CMESM 结构 15
3.2 成员评价指标 16
3.3 代表层 18
3.3.1 推举代表 18
3.3.2 知识库 19
3.3.3 判定器 20
3.4 委员会层 23

第 4 章 联邦模式多级虚拟组织管理 26
4.1 联邦模式多级虚拟组织模型 26
4.2 联邦模式多级虚拟组织管理方法 29

第 5 章 CMESM 性能评价 32
5.1 仿真测试环境 33
5.2 CMESM 在不同参数下的性能表现 34
5.3 CMESM 在不同赛百环境中的性能表现 36
5.4 CMESM 对 Makespan 的提高 37
5.4.1 两种广泛使用的任务调度算法 38
5.4.2 实验设计 39
5.4.3 CMESM 在 Min-min 调度算法下的性能 39
5.4.4 CMESM 在 Max-min 调度算法下的性能 40
5.5 CMESM 性能总结 41

第 6 章 赛百平台虚拟组织管理中间件 42
6.1 软件架构图 44

6.2 认证中心及信息传输 47
6.2.1 CA 中心和签发证书 47
6.2.2 消息传输 48
6.3 内容解析 51
6.4 应用层：虚拟组织管理 54
6.5 显示层 56

第 7 章 DVOMS 使用及与 Globus 协同应用实例 59
7.1 DVOMS 使用 59
7.2 命令行模式下 DVOMS 与 Globus 的协同应用实例 66

第 8 章 总结 75

参考文献 79

致谢与声明 82

个人简历、在学期间发表的学术论文与研究成果 83

第 1 章 引言

1.1 选题背景-赛百平台的提出及其意义

为了验证爱因斯坦广义相对论中引力波的存在，2000 年初由美国国家科学基金（National Science Foundation）投资，加州理工（Caltech）和麻省理工学院（MIT）负责建设的激光干涉引力波天文台（Laser Interferometer Gravitational-wave Observatory，简称 LIGO）[1]正式投入了科学运行。LIGO 每天产生 1 万亿字节的数据，来自全世界四十多个科研单位，十几个不同科学领域，包括计算机，电子仪器，天体物理等，超过 500 名研究人员组成了 LIGO 科学合作组织（LIGO Scientific Collaboration，简称 LSC）[2]，共享分布在美国和欧洲的 10 多台高性能集群机的几千个 CPU 和上千万亿字节的存储能力，联合投入到 LIGO 的数据分析工作中。这仅仅是现代科学研究领域应用的一个例子。随着科学研究的逐渐深入，现代科学研究越来越依赖于实验仪器，计算机和网络交互，信息技术的支持需求也由于科学应用的特异性而多样化[3]。很多重大科研项目例如黑洞合并、暗物质结构、气候变化因素、蛋白质折叠以及基因测序都需要不同领域的研究人员通力合作，依赖分布在世界各地的计算机和实验仪器进行模拟计算。因此如何集成，管理和共享这些分布式的计算机，实验仪器和数据等资源就成了开发人员首要解决的问题。

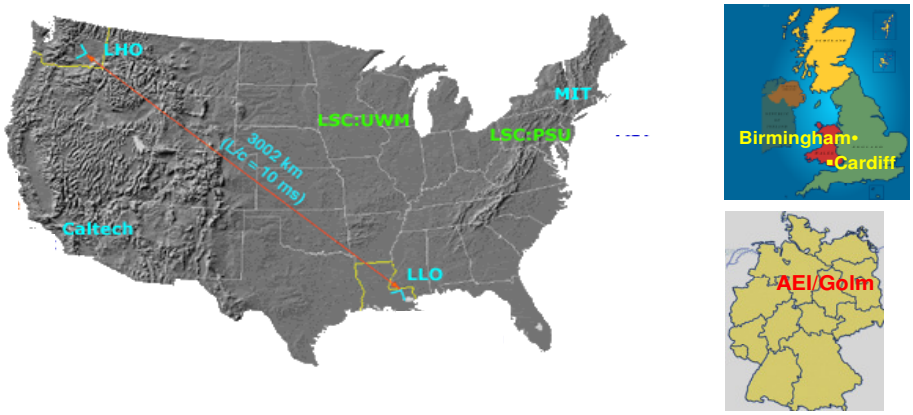


图 1.1 LIGO 示意图

作为传统解决方案，网格[4]技术被广泛应用于科学计算领域，为特定的大型科研应用提供专门的计算机，数据共享和管理平台。在过去十多年中，许多网格被建立起来以满足某些重大专项科研项目：地震仿真网络（Network for Earthquake Engineering Simulation，简称 NEES）[5]、开放科研网格（Open Science Grid，简称 OSG）[6]、国家生态观测网（National Ecological Observatory Network，简称 NEON）[7]、地球科学网（The Geosciences Network，简称 GEON）[8]、国家大气研究中心（National Center for Atmospheric Research，简称 NCAR）[9]、计算纳米技术网（Network for Computational Nanotechnology，简称 NCN）[10]、国家虚拟天文台（US National Virtual Observatory，简称 NVO）[11]和万亿次网格（TeraGrid）[12]等等。但和过去相比，当前的科学研究有着新的特点，频繁的跨学科交流合作，多样化的资源整合需求和信息支持整体解决方案，跨地理和跨组织研究活动等等。这种科学研究方式为信息技术研究人员提出了新的挑战：更加广泛的集成和共享资源；跨越地理和组织的限制，动态的，按需的集成资源；安全，可扩展的管理资源；而这是传统网格技术不能解决或者不擅长解决的。针对这种情况，在 2003 年，美国国家科学基金公布的专家咨询报告（一般称为 Atkins 报告）中提出了赛百平台（CI，Cyberinfrastructure）这个概念[3]，并且在 2006 年进一步成立了 Office of CI[13]专门负责 CI 理论方面的建设和具体计划订制[14]。

赛百平台，也就是计算机基础架构，作为一个新名词主要指基于分布式计算机、信息和通信技术的基础架构。赛百平台对于知识经济的重要性可以与基础架构对工业经济的支撑作用相比拟。在赛百环境中，资源所有人可以注册个人资源，并按照自己规定的方式共享这些资源；而用户则可以根据自己的需求搜集特定资源，并通过加入或者组织某一虚拟组织来满足特定应用的需求。赛百平台将是一个帮助和使能这些活动的环境，为用户提供注册，管理，搜索和评价等服务。在赛百平台中，资源提供者被统称为 RP（Resource Provider），而用户则被称为 User，他们统称是赛百平台的成员，都处于 CIMC（CI Management Center）的管理之下。成员可以动态方便的申请建立针对特定科学研究和应用的虚拟组织，通过虚拟组织整合其他赛百成员，包括特定领域的研究人员（即为 User）和某些网格或者分布式资源（即为 RP），实现资源动态，有效和有针对性的整合。和传统的网格相比，赛百环境有以下一些特点：

图 1.2 CI 与 Grid 的结构关系图

1. 赛百平台通过定制虚拟组织（VO，Virtual Organization）[15][16][17][18]来集成和管理资源。针对某一特定科学应用，赛百平台支持用户集成相关资源组织成为专属这一应用的虚拟组织。所有加入这一虚拟组织的成员均同意此组织协议，相互之间达成适当的信任关系，从而实现了资源的安全共享。而非此虚拟组织内部成员则一般没有权限访问和使用虚拟组织内部资源。因为虚拟组织的建立是受特定科研应用的需求驱动的，所以它的建立和取消都是动态的。当用户有需求的时候，用户可以建立虚拟组织，邀请合适的资源和成员加入到这个虚拟组织中来以便满足特定学科要求；当项目结束的时候又可以取消虚拟组织，释放出这些资源。每一个虚拟组织都相当于一个针对特定科学应用的网格，从这个角度来看，网格只是 CI 的特定应用。

2. 赛百平台成员来源更广泛，成员之间天然上是陌生的。在传统的网格环境中，由于网格应用有着强烈的专业性，其成员也都从事于相关领域，所以一般来说网格成员在现实世界中已经建立了相对信任和稳定的联系，网格技术仅仅通过 IT 技术使能了这种信任和联系。事实上也只有在现实世界中建立了信任关系之后，网格成员才通过加入网格来共享个人计算资源。而与之相比，赛百平台中的成员来自各个不同的科学领域和科研单位，成员更加复杂，很难事先就建立信任关系，成员之间在未发生交互行为是一般为陌生状态。为了保证资源的共享和使用能够安全可信，就需要赛百平台能够对平台中的成员的行为模式进行评价，以预防恶意行为并帮助成员之间建立可信安全的共享关系。

3. CI 是没有特定应用范围和学科领域的，它是针对所有学科的科学研究。目前已经构建好的网格无一不是专门用于特定学科领域，例如国家虚拟天文台

（NVO）用于天文台的数据共享和提供数据处理能力，国家大气研究中心（NCAR）则关注于天气预报和大气模拟。但随着科学研究的近一步发展，对于科学计算的需求也在快速增长。分别为每一学科都专门构建网格显然开销太大而且也不利于提高资源利用效率。事实上虽然网格内部实现了资源共享，但网格与网格之间仍然是有壁垒的。相比之下，CI 则打通了不同网格之间的壁垒，实现了网格之间的资源共享。从这个意义上来说，CI 其实就是各个网格的集合体，如图 1.2 所示。

1.2 问题提出-虚拟组织管理和信任机制

CI 作为一个新的理念，在满足科学研究的同时也为开发人员提出了挑战。如何将赛百平台上分布式的资源根据特定科研项目需求动态安全的集成为一个虚拟组织，实现虚拟组织内部的高度共享和虚拟组织之间便捷安全的合作成为亟待解决的问题。总的来说，挑战来自于以下三个方面。

1. 支持多级动态的虚拟组织管理。首先虚拟组织是受特定科研项目需求的驱动，其生命周期即为实际项目的生命周期，所以赛百平台需要能够支持虚拟组织的动态创建。现代大型的科研项目需要多方面的合作和支持，内部大多分为若干小组进行研究。与这种大型科研活动的这种模式相对应的就是赛百平台需要支持多级虚拟组织管理。比较典型的就是 LIGO[19]，LIGO 作为一个独立的科研合作组织，与 OSG，TeroGrid，MIT 等组织有密切合作，而在其内部，针对不同的分析方法又分为不同的小组进行相对独立的研究，例如 DASWG（Data Analysis Software Working Group）Group，Burst Analysis Group 和 Omega Pipeline Group 等。因此为了更好的支持当前大型科研项目，就必须保证赛百平台能够支持动态的支持多级虚拟组织管理，并在有隶属关系的虚拟组织之间建立适当的信任关系。动态性主要指虚拟组织的构建，虚拟组织之间的合作关系可能随着应用需求动态的发生改变，或建立新的虚拟组织关系或解除旧有的虚拟组织关系。适当的信任关系主要指在多级虚拟组织管理中要明确组织双方在对方合作组织中拥有的权限，对方合作组织中的成员能够满足本组织对于成员信任度方面要求。这两个方面在多级虚拟组织管理中是缺一不可的。

2. 信任度评价，针对不同的安全需求建立可信任共享关系。赛百平台和网格平台一个很大的不同就是成员不仅仅来自于某一专业，而是来自于各个不同

的科学领域和科研单位，成员来源更加复杂，相互之间天然是陌生关系。对于RP 来说，他们愿意让有良好使用信誉的 User 使用其计算资源，以防止潜在的破坏性使用情况的出现；对于 User 来说，他们希望选择可靠稳定的资源，以提高任务运行的稳定性和可控性。因此当在判断某一赛百平台成员（这个成员可能是 RP 也可能是 User）是否可以加入到某一虚拟组织当中的时候，就需要评价这个新成员是否满足了此虚拟组织在成员可靠性方面的要求。因此需要赛百平台能够为用户提供一个准确，可靠的信任度评价方法来评价特定 RP 和 User 的行为特点，帮助成员建立合适的信任关系。信任度评价方法需要能够反映虚拟组织对于成员可行程度个性化评价要求。 虚拟组织受不同的科研应用的驱动，而不同的科研应用对于计算资源的稳定性或者普通用户的可靠性的要求是不同的。在计算密集型科研应用，用户希望计算资源能够提供长期稳定的计算服务，而对于存储稳定性的要求就会差一些；资源提供方则希望成员提供的程序更加安全可靠，并行化程度高。在数据密集型科研应用中，用户希望资源提供方在存储方面有更高的稳定性，而对于计算稳定性则没有很高要求；资源提供方来说则希望用户程序尽可能减少小块数据频繁读写的情况以保证资源利用效率。

3. 反欺诈行为。赛百平台类似于 B2C 环境，是提供一个交互平台还帮助 RP 和 User 建立可信关系。和一般的 B2C 网站类似，成员评价方案也面临欺诈行为，也就是成员通过一些虚假交互行为欺骗信任度评价方法，使之做出不符合实际资源状况的结论。特别是共谋欺诈行为，当前广泛使用的信任度评价方法很难判别出来。因此为了维护赛百平台的正常秩序，也为了保证信任度评价方法的科学性和准确性，赛百平台下的信任度评价方法必须能够有效的抵御各种欺诈行为，特别是共谋欺诈行为。

针对以上三个问题，本文提出基于联邦模式的多级虚拟组织管理结构和基于委员会的信任度评价方法。联邦模式的多级虚拟组织管理拓扑结构在保证虚拟组织相对独立和自由的同时可以有效的解决虚拟组织之间进行合作所产生的信任度和权限设置问题。为了满足特定虚拟组织对成员信任度方面的个性化要求，本文提出基于委员会的成员评价方法，具体是每一个虚拟组织构建虚拟组织委员会，采用基于模糊数学的 K 近邻法模式识别方法帮助虚拟组织对特定的赛百成员进行准确和个性化的评价，帮助用户找到满足其信任要求的资源和赛百成员。通过仿真实验证明这种方法可以有效的抵御共谋欺诈行为，提高虚拟组织内部的服务质量。

本文结构安排如下：第二章将介绍相关的技术背景，包括传统的权限管理方法和虚拟组织管理方案，同时也对 EBay 和 Amazon 等网站的信任度评价方法做了介绍；第三章提出基于委员会的虚拟组织成员评价方法 Committed-based Member Evaluation and Selection Method (CMESM) ，CMESM 综合考虑虚拟组织内部成员的历史信息和偏好对申请人进行评价，帮助虚拟组织内部成员构建适当的信任关系；第四章节介绍基于联邦模式的虚拟组织结构及其特点，特别是介绍了如何在联邦模式下如何通过 CMESM 进行多级虚拟组织的管理；第五章构建仿真环境，通过设计仿真实验来检测基于委员会的成员评价方法的性能，从是否减少任务组 Makespan 和可否有效抵御共谋欺诈行为来说明其可靠性和有效性。第六章介绍了作者实际开发的赛百平台虚拟组织管理软件 DVOMS（Dynamic Virtual Organization Management System）的结构和实现；在第七章，演示了 DVOMS 的使用流程，并通过一个典型赛百平台应用案例来说明其是如何与 Globus Toolkit 配合使用，实现赛百平台成员从发现，到构建虚拟组织到最终共享资源的全部流程。最后在第八章总结本文并提出未来进一步研究计划。

第 2 章 国内外研究动态及相关技术

赛百技术是对网格技术的进一步发展和扩充，在电子商务和网格的实际应用中已经有访问控制，VO 管理及信誉评价的相关机制和技术。本章将分别介绍目前被广泛使用的几种技术和机制。

2.1 VOMS

VOMS[20][21][22]（Virtual Organization Membership Service），目前在网格环境中广泛应用于认证和访问控制。它是由欧洲数据网格[23](European DataGrid)和大西洋交通网格[24]（TransAtlantic Grid）合作开发，并且被很多其他的科学网格采纳应用，例如 LIGO (Laser Interferometer Gravitational-wave Observatory)，SBGrid[25] (Structural Biology Grid)，OSG (Open Science Grid), GUGrid[26] (Georgetown University Grid) 等等。开发 VOMS 的动机就是希望能够在 Grid 内部进一步细化组织，在 VO 层次上进行认证和权限管理而不仅仅在 Grid 层次上进行粗颗粒的管理和认证。这种精细的管理模式在一定程度上适用了当前科学研究过程中对资源的多样化需求。VOMS 是一个中心服务的架构，有一台中心服务器通过数据库来维护 VO 关系信息和成员信息。每名想加入这个 VO 的网格组织成员必须将自己的证书发送到 VOMS 服务器，通过审查后 VOMS 服务器将在原有证书的基础上将 VO 相关信息添加到证书扩展部分，同时重新签订证书，返回给用户。在 VO 中也可以建立 group，这个 group 实际上就是 sub-VO，不过 group 中没有相应的 VOMS 服务器，也不需要为这个 group 中的成员重新签发证书，这里的组织关系信息将直接通过 VOMS 服务器写在证书当中。在证书当中记载了每一名成员在 VO 中的角色和组织关系，而 VO 内部成员将通过来访者在 VO 下的这种角色和组织关系信息来开放相应的权限。在 VOMS 管理机制当中，在一个 VO 内部可以建立子一级的 VO，也就是 group，而对于 group 这个虚拟组织，其内部成员是不需要重新签发证书的，用户的虚拟组织关系通过被 VOMS 签发的证书来管理和认证。例如一名成员的组织关系和角色通过 VO-Group-Role 来定义。VOMS 主要是由四个部分组成，如图 2.1 所示，用户可户端和用户服务器运行在用户端，而管理员客户端和管理员服务器则运行在 VO 服务器端：

- 用户客户端：客户通过用户客户端和证书与服务器进行通信，通过认证后可以向 VO 服务器查询 VO 信息和成员信息，例如 VO 成员列表，角色分配情况，group 信息和资源性能信息等。
- 用户服务器：接受其他客户用户端和服务器客户端的查询请求,返回必要的本地用户信息。
- 管理员客户端: 管理员通过管理员客户端来管理 VO 组织关系，例如添加新的用户，签发证书，查询特定用户信息，创建 group 和分配角色等。
- 管理员服务器: 这个服务器主要就是数据服务器，管理 VO 数据库，相应用户客户端的查询请求。

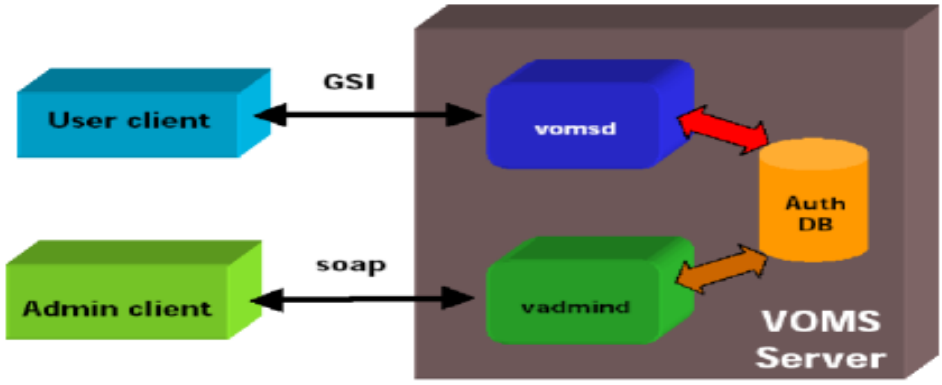


图 2.1 VOMS 结构图

作为 DataTAG 和 EDG 的合作项目，VOMS 采用 Globus tookit 工具包提供的 Globus Security Infrastructure（GSI）安全机制来签发证书和认证证书。用户通过“voms-proxy-init”来获得经过 VOMS 重新签发的 RFC 3281[27]数据格式证书。由于 GSI 被广泛应用于网格技术当中，所以通过 VOMS 签发的证书也可以和其他“非 VOMS”的 GateKeeper 通信。由于 VOMS 是基于 GT 工具包开发的组件，所以 VOMS 可以部署在任何通过 GT 组建的网格中。VOMS 的缺陷主要有两点：

一是 VOMS 并没有之间提供信任度评价机制，不过在证书中记载了用户的组织信息和 VO 角色，访问者可以通过证书上的这两种信息来决定赋予来访者何种权限，然后根据证书信息将远程用户映射到拥有合适权限的本地帐户。不过 VOMS 并没有提供这样的工具来自动的完成这个映射过程，而是通过另外一个

组建，Grid User Management System（GUMS）来完成。

二是 VOMS 提供的 VO 管理并不能够支持动态，复杂的 VO 构建和管理。VOMS 将 Membership 信息写到成员证书当中，成员之间通过提取证书信息来构建合适的信任关系。成员加入新的 VO 就需要向相应的 VOMS 请求签发新的证书。但在 CI 环境中，VO 的建立是动态的，对于用户来说频繁的修改证书不仅开销比较大而且也不够安全。同时 VOMS 仅仅支持 Grid-VO-Group 三层的虚拟组织结构，不能够满足 CI 环境下用户构建复杂 VO 的需求。

2.2 GUMS

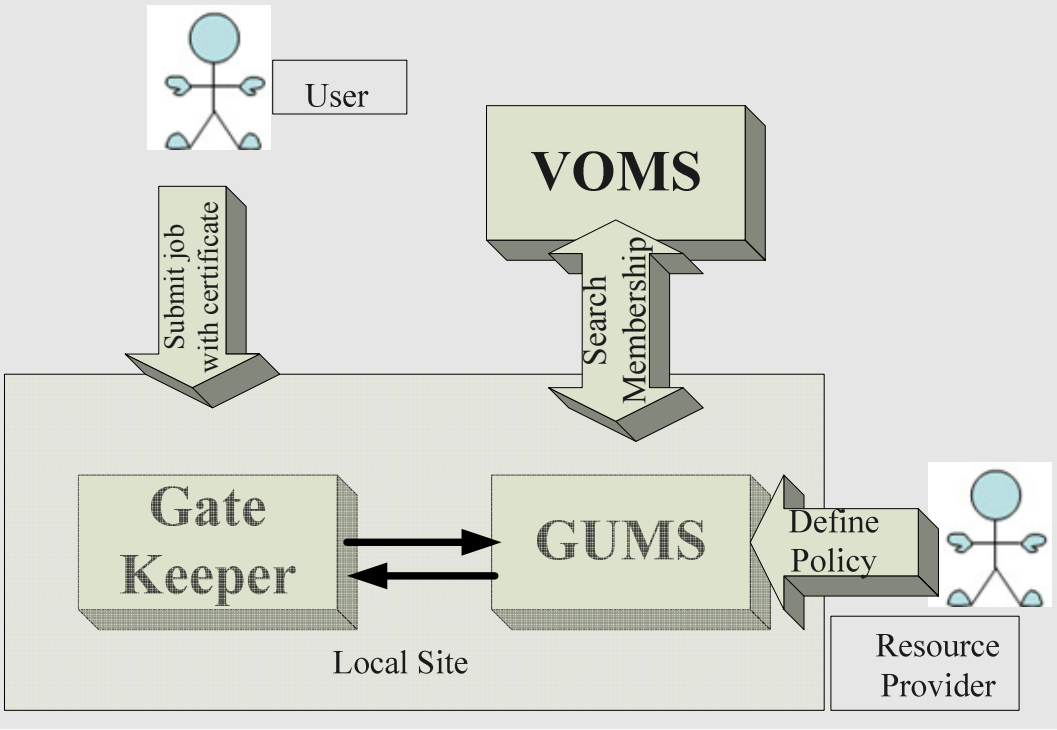


图 2.2 GUMS 账户映射流程及与 VOMS 的关系

Grid User Management System[28]（GUMS）是一个运行在网格用户端的组件，其主要功能是定制映射规则，根据被访问者的 VO 信息自动将被访问者映射为本地账户。在网格中，资源的共享是通过将远程用户映射为资源本地账户来实现的。User 或者 RP 有可能属于多个 VO，在这种情况下，GUMS 向 RP 提供合适

的接口和描述语言，帮助其定义映射规则；当接收到访问请求，GUMS 将自动根据之前定制的规则将来访者映射为本地账户。图 2.2 演示了 GUMS 完成账户映射功能流程以及和 VOMS 的关系。

GUMS 通过两种方式将网格证书映射为本地账户的：1）通过 map-file 静态的将网格证书映射为本地用户 2）根据映射规则和来访者身份动态的映射为本地用户。为了说明这两种方式都是如何实现映射的，我们假设有一个用户想将某项任务提交与远程的某台计算机。首先用户会将任务和证书一起提交与这台计算机。计算机的 Job Manager 接受任务和证书，提取证书信息，然后调用组件 Gatekeeper。Gatekeeper 根据设置或者通过查询 Map-file 或者通过搜索映射规则返回一个本地账户。Job Manager 从而可以在返回账户权限下运行这个任务。GUMS 的具体功能可以总结如下：

从 VOMS 或者 LDAP 获取来访者的身份信息和组织关系。

- 在本地维护一个手工添加用户的用户信息。
- 将多个组的多个用户映射为同一个本地账户。
- 将多个组的多个用户映射到一个本地账户池。
- 根据 NIS 或者 LDAP 提供的信息将多个组的多个用户映射为相应的本地账户。

GUMS 不提供认证，主要是向 Job Manager 提供合适的账户信息。从这个观点来看，GUMS 是一个规则决策组件（Policy Decision Point）而不是规则执行组件（Policy Enforcement Point）,所以它必须和其他组件配套使用。

2.3 DAC，MAC 和RBAC

Discretional Access Control(DAC)[29]，也就是自由访问控制，是 1970 年提出的访问控制方法，它主要是通过访问控制矩阵来实现对不同用户的访问控制。在自由访问控制机制中，任何成员都可以将不超过自己拥有权限范围的权限赋予其他成员，每个成员都会针对其他成员在本地赋予不同的权限，这些信息以访问控制矩阵的形式存储在本地。在访问控制矩阵当中，列表示来访者 User，行表示资源所有者 RP，矩阵元素表示对应列 User 在对应行 RP 上的权限。如果 User 希望访问资源 RP，则 RP 将会查询本地的访问控制矩阵，根据矩阵内相应元素的值来确定赋予来访者何种权限。但是由于自由访问控制是对等结构，任

何成员都可以在自己权限范围内赋予其他成员权限，这就有可能造成权限管理漏洞：例如 A 是禁止 B 访问，而 A，B 双方都信任第三方 C，于是 A 有可能通过 C 而拥有对 B 的访问权限。此外访问控制矩阵会随着加入成员的增加而快速增大规模，增加了管理复杂度。

Mandatory Access Control（MAC）[30][31]，也就是强制访问控制，是通过安全等级来实现对来访人员的控制。在强制访问控制机制中，任何成员都被系统管理员分配给一个安全标签，这个安全标签是不可被普通人员修改的。在这个安全标签中记录了对应成员在系统中的安全等级。资源提供者只容许超过一定安全等级的成员访问本地资源。强制访问控制主要用于稳定严密的组织当中，例如国防部门，政府部门等。但强制访问控制并不适合 CI 环境，这主要是因为 CI 环境是一个分布式，开放的环境，RP 拥有对资源的绝对控制权力，CI 平台对于环境中的资源并没有绝对控制权限，主要承担的是协调和调度功能。

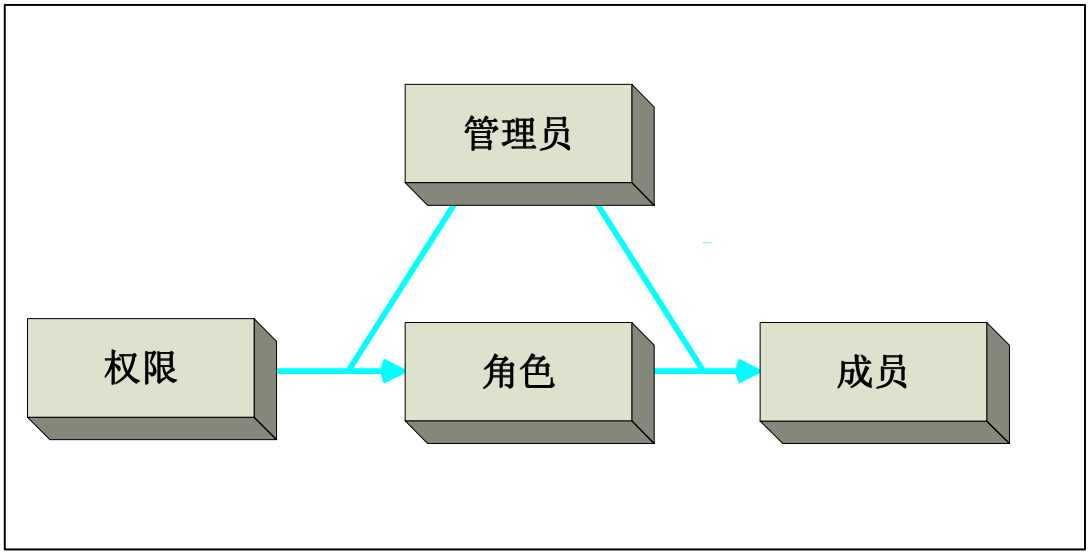


图 2.3 基于角色访问控制模型

基于角色的访问控制，Role-based Access Control（RBAC）[32][33]是由 National Institute of Standards and Technology（NIST）与 1990s 提出的一套访问控制方法，结构如图 2.3 所示。这套方法分离了具体成员与特定权限的联系，设立角色这个元素来指代特定权限。在 RBAC 机制中，有一位管理人员负责定制角色和分配角色。管理员根据实际需求，设定在系统中的角色，并且将合适的权限赋予这

个角色。当新的用户加入到系统当中或者原有用户需要更改角色，管理员就会根据情况将特定角色赋予申请人，被赋予角色的成员也就同时拥有了这个角色的所有权限。RBAC 的特点在于简化了访问控制和权限管理，而且能够满足复杂多变的应用需求，所以 RBAC 被广泛应用于各种组织内部的访问控制。不过 RBAC 控制方法同样只适合于集权制的组织当中，所有的资源都处于管理员的控制之下。同样不适合具有联邦组织结构的 CI 平台上。

2.4 PRERMIS

Privilege and Role Management Infrastructure Standards Validation （PREMIS）[34]是由 ISIS 开发的用于解决个人认证和本地权限管理的工具包。PREMIS 可以很方便的和当前通用的认证系统进行通讯并且加以补充。作为一款权限管理工具，PREMIS 除了提供权限管理以外还有主要有两个功能：一是为用户提供策略编辑功能用于创建本地策略；二是将远程用户根据策略和具体情况赋予合适的权限。PREMIS 管理两种策略：一种是认证策略用于定义如何赋予远程用户在本地合适得权限；二是委任策略用于定义如何将在同一小组内角色分配的权限委任给一个可信任的成员。所有的这些策略都是基于 XML 格式。同时 PREMIS 也提供了对本地权限的编辑功能，并且按照 X.509 格式。通过这些策略，PREMIS 可以提供一下服务：

- 当用户提出访问请求，PREMIS 可以通过来访者的身份和本地访问策略做出权限控制决策。
- 根据用户需求来编辑访问策略。
- 可以委托可信任的第三方来为同一个小组内的成员进行角色分配，不同角色将在本地拥有不同的权限。

PERMIS 运行在客户端，所以 PERMIS 的一个显著缺点就是无法为用户提供系统全局的信息，而且并不能够维护 VO 级的组织关系管理和权限管理。

2.5 资源评价方法

除了上面所述权限管理方法，还有一种是基于声誉的权限管理方法。这种方法被广泛的应用于各种成员天然陌生，交互频繁的大型虚拟社区，例如电子商务，P2P 应用等，而且也有了一些相应的解决方法[35][36][37]。

信誉在不同应用场合有着不同的定义，在[38]中，B 对 A 的信誉值定义为 A 在多大程度上相信 B 会按照 A 规定的方式来操作而且不会攻击 A。而在 CI 中，A 认为 B 的信誉值更是一般化为 A 愿意在多大程度上承担其赋予 B 一定权限后的风险。这个一般化的定义不论在分布式计算还是电子商务中都有普遍的意义。

仿照现实社会中个体声誉评价方式，在一般虚拟社区中，成员的声誉也是通过两种方式来进行计算的。假设在虚拟社区当中有三个成员 A，B，C，A 需要计算 B 相对于 A 的信誉度。

直接信誉度：直接信誉度是由 A 对 B 的观察值和 A 保有的与 B 交互的历史纪录来计算的。首先 A 会向 B 发送查询请求，获得对 B 的一般性的身份信息，然后结合 A 保有的对 B 的历史交互信息，通过一定的算法计算出 B 相对于 A 的直接信誉度。这个值是 A 对 B 的直接观察结果。直接信誉度是通过观察 A 与 B 的直接交互信息来进行估算，适用于成员较少，交互频繁的虚拟社区。但这种方法无法帮助成员建立安全可靠的初次交互，而且在成员较多的虚拟社区中希望进行交互的成员之间往往天然是陌生的，因此这种缺陷也变的更加明显。

推荐信誉度：这个信誉度是由第三方 C 提供的。有时候仅仅靠 A 本身的观察判断是不足以判定 B 的信誉，这就需要受到信任的第三方个体 C 来来帮助评判。这个评判意见就是推荐信誉度。C 相对于 A 的信誉度越高，则 C 的意见也就越加可信。这个值是第三方可信任个体对 B 的推荐值，也就是 A 对 B 的间接观测结果。这种方法被广泛的应用于各种集中式虚拟社区，例如 EBay，Amazon 和 taobao 等。在这些虚拟社区中，由平台运营中心作为可信任的第三方对社区中所有成员进行跟踪和维护，为希望建立信任关系的成员提供对方信誉度，帮助建立合适安全的信任关系。和这些虚拟社区相比，赛百平台是一个非集中（Decentralized），支持内部自治子虚拟社区的一个开放式平台。为了支持当前形式多变的科学计算应用，需要为用户提供个性化，满足特定应用需求的成员评价，相比较而已一般的推荐信誉度为成员提供的是一般化的信誉度，无法满足个性化成员评价需求。

第 3 章 基于委员会的成员评价方法

随着网络技术的进步和网络资源的不断丰富，网络世界中也慢慢形成了比较成熟的网络社区，同时也带来了现实社区中存在的问题，一个显著的挑战就是如何帮助虚拟社区中的实体建立适当的信任关系，而这种需求在赛百平台应用中变的尤为关键。研究人员在网格计算，网络虚拟社区等领域中借鉴了现实社会的社会组织形式，提出了虚拟组织 （Virtual Organization）的概念。虚拟组织是处于某种相同隶属域的实体的集合[16]，在科学计算当中，研究人员把遵守某一共享协议的实体集合及其组织协议称作虚拟组织。所有虚拟组织成员都遵守相同的共享协议，组织内部成员直接可以很方便的进行可信任的资源共享。和一般的访问控制方法，例如强制访问控制的等相比，基于虚拟组织的访问控制直观方便，更适合当前大规模的权限管理应用。但是随之而来的是如何确保虚拟组织成员信任度，当前虚拟组织结构僵化和构建繁琐等问题和挑战。一方面应用的需求使得资源共享成为一种需要和趋势，而另一方面网络社区的急剧扩张使得社区内的成员天然上是陌生的，个人经验的局限性使其不能够成为判定一个成员是否可信的依据。而虚拟组织的构建最关键的是评价待加入成员是否是可信任的。信任程度的不同表示了成员在多大程度上愿意承担接纳特定成员加入虚拟组织后带来的风险。不同的科学应用，User 或者 RP 对风险的定义和承受能力也是不同的，例如对于计算密集型的资源，其承受低效率程序带来潜在额外计算开销风险的能力就比较强而对频繁数据读取的承受能力就较差；保密类数据应用程序对数据泄密风险就特别看重而更愿意选择保密性好的资源。

因此动态，按需的构建针对特定应用的虚拟组织，其关键在于对赛百平台上的成员进行个性化，准确的评价以决定是否满足特定 VO 的信任需求。上文提到虚拟组织是由特定科研项目驱动的，有着很强的项目针对性。由于科研项目在资源和使用者方面有着个性化需求，所以这也就造成了不同虚拟组织在评价资源和成员是否可靠，可信任的时候有着明显的应用偏好。此外赛百平台上的成员来源广泛，数量众多，而且这些成员天然上是陌生的，需要赛百平台提供适当的评价体系来帮助成员之间建立信任关系。传统的网格技术和虚拟组织管理工具往往忽略了这两方面的需求，往往由网格成员或者虚拟组织的管理员人为的对资源进行甄别。这种甄别方法依赖

于管理员的个人经验，忽略了组织内部成员的需求，也不适用于大范围的资源共享社区。本章节参照现实社会中组织管理办法，提出基于虚拟组织的成员评价方法 CMESM（Committee-based Member Evaluation and Selection Method）[39]，为虚拟组织对特定成员进行有偏好的评价，帮助其构建可信任的虚拟组织。

3.1 CMESM结构

委员会作为一种组织管理结构被广泛的应用于各种实际组织当中。首先选举出能够代表普通成员意见且富有经验的成员作为代表，并由这些代表组成委员会；然后每一位代表根据个人经验对申请成员进行分析，向委员会提出自己的意见；最后委员会通过模糊决策算法综合各位代表的意见对申请成员进行最终评价。委员会模式和组织全部成员进行投票表决的模式相比，可以更快的响应事件，和集权式管理相比，又具有民主性，代表了大多数成员意见，最大程度的降低了由于个人原因而做出错误判断的可能性。实践证明委员会制度是目前比较可行，也能代表（除了全民公投模式）组织成员意见的一种组织管理模式。

赛百平台中的虚拟组织也同样面临如何快速和民主的响应组织事务这一问题，例如是否批准某一成员加入本组织，或者同意加入更高虚拟组织（参见第 4 章）等事务。因此本文将委员会组织管理制度加以适当的改造应用到赛百平台虚拟组织管理方案当中，其结构如图 3.1 所示.

CMESM 分为两层，代表层(Representative Layer)和委员会层(Committee Lay)。

代表层选自虚拟组织内部成员，代表由最活跃，最有经验的成员担当，或者由对虚拟组织信任度有特殊需要的成员担当。在默认情况下，将会选自组织成员中声誉值最高的成员作为代表，同时任何有特殊信任需求的成员可以向虚拟组织管理员提出申请，担当代表。代表层根据个人经验对虚拟组织申请成员（User，RP 或者 VO）做出独立判断，并将个人意见提交给虚拟组织的委员会层。

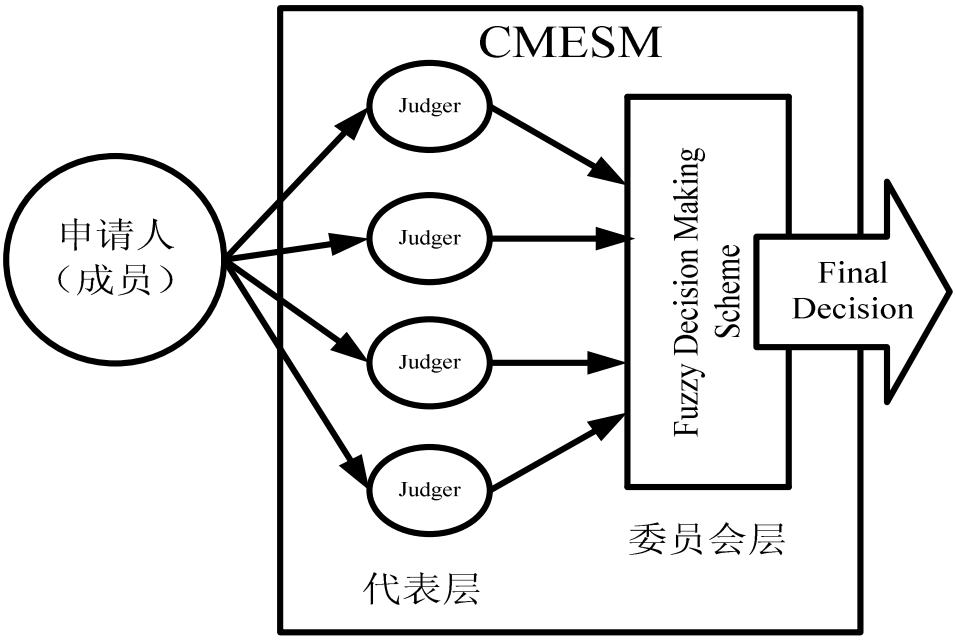


图 3.1 CMESM 结构

委员会层由所有的虚拟组织代表组成。委员会层接收代表判断意见，根据一定的表决制度和每一位代表在委员会不同的权重做出基于模糊数学的最终判断。

3.2 成员评价指标

赛百平台记录成员的历史交互数据，CMESM 中的代表通过申请成员的历史交互数据来判断申请成员在赛百平台的行为模式从而做出适当的安全性评判。理论上来说，赛百平台记录成员的数据越详细，做出的判断也就越准确。在开发应用过程中可以根据实际情况设定相应的成员评价指标来对特定成员进行描述，例如大部分的交易网站例如 Taobao，EBay 都将商户收到来自用户的正评价数量，负评价数量和中间状态评价数量作为判断和评价商户的基础数据。简单起见，本文采用类似的两类指标对特定赛百成员进行描述。首先给出一些定义.

估计执行时间（Predicted Execution Duration）：用户预测任务完成所需时间。任务执行时间估计问题在科学计算领域是一个比较基础的问题，有多种计算模

型可以通过估计任务复杂度和资源属性进行恰当的预测执行时间估计。

实际任务执行时间(Execution Duration)：任务实际完成时间，指某项任务由用户提交到最终接受结果实际花费时间，是一项实际观测值。

交互(Interaction)：本文将 User 提交任务给 RP，到 RP 完成任务并将任务结果返回给 User 这一完整流程定义为一次交互。

在科学计算中，不论 User 还是 RP 都希望某一项任务能够在给定估计执行时间内完成，这样有利于进行高效的资源调度和任务交互，特别是在工作流(Workflow)模式下，某一项任务的输出往往是下一项任务的输入之一，如果不能够在给定时间内完成，将会造成整个工作流的瓶颈，影响后续任务的完成。所以是否在估计执行时间内完成任务成为 User 和 RP 都关心的问题，也是估计某一成员是否可靠的重要指标，即 User 是否提供可靠高效的任务以及 RP 是否提供稳定的计算服务。当 User 和 RP 发生一次交互行为，可能出现以下情况：

如果是（估计执行时间>=实际执行时间）&（任务成功完成） （3-1）

那么一次高质量交互，User 及RP 高质量交互次数加一；

如果（估计执行时间<实际执行时间）\（任务成功完成）： （3-2）

那么一次底质量交互，User 及RP 底质量交互次数加一；

本文通过赛百成员高质量交互完成次数 *pes（Positive Evaluation Statistics）*及低质量交互完成次数 *nes（Negative Evaluation Statistics）*来描述某一赛百成员。当一次交互行为结束，赛百平台将根据其估计执行时间和实际执行时间进行判断，若为高质量交互，那么 User 和 RP 的 *pes* 各加 1；若为低质量交互，则 User 和 RP 的 *nes* 各加 1。这两个统计数字组成特征向量作为成员标签。对于赛百成员 j，其特征向量标记为：

$$x_j=(pes_j, nes_j). \quad (3\text{-}3)$$

此外，为了尽可能模拟人类判断，本文采用模糊数学来表示某一成员或者委员会对另外一个成员的判定。在虚拟组织管理中，最重要的判定是审核某一赛百成员是否满足判定人的信任需要。本文将这种判定结果划分为可以接受，拒绝和无法判断三种。在确定性数学当中，待审核成员只可能隶属于这三种中的一种而排斥其他两种判断，即非此即彼。但是在实际认知中，人类对某一事物的看法往往是模糊的，例如人们可能认为某一个商品比较好但略有瑕疵，或者

整体一般却没有重大缺陷等等。所以如果用确定单一的判断来表达这种模糊，有程度差异的看法，有大量有用的信息会被丢弃。在模糊数学中，对实体类型的评价通过隶属度来表达。隶属度从 0 到 1，0 表示这个实体完全不属于某一类，1 表示完全隶属某一类。假定某一件商品可能属于优质商品，可能属于伪劣商品，也可能属于一般质量的商品。在确定性数学当中，一件商品要么属于优质商品，要么属于伪劣商品，要么属于一般质量商品；而在模糊数学当中，这件商品可能 0.5 的属于优质商品，0.3 属于一般质量商品，0.2 属于伪劣商品。隶属度的不同体现了个人对这件商品不同的认可程度，能够表达更加丰富的信息。因此本文引入模糊逻辑概念，通过评价向量 $M_{i,j}=(r_{i,j}, d_{i,j}, u_{i,j})$来表示成员 i 对成员 j 的评价结论，$r_{i,j}+d_{i,j}+u_{i,j}=1$。其中 $r_{i,j}$ 表示成员 i 在多大程度上认为成员 j 为可靠成员，$u_{i,j}$ 表示成员 i 在多大程度上认为成员 j 为不可靠成员，$d_{i,j}$ 表示成员 i 在多大程度上认为成员 j 需要进一步判定。其中 1 表示完全认同，0 表示完全不认同。本文全文，不论是个人对成员的判断还是委员会对成员的判断，都通过评价向量 M 表示。

3.3 代表层

代表层是 CMESM 的基础，他们从各个方面反映其他成员对申请加入成员的信任要求。因此赛百平台需要提供合适的方法来选择合适的虚拟组织成员作为代表，以及帮助代表进行客观准确的判断。

3.3.1 推举代表

代表是 CMESM 的基础，一个虚拟组织一旦被建立，那么作为虚拟组织的创建人，同时也是虚拟组织管理员就自动成为这个虚拟组织的第一位代表，他代表了虚拟组织对申请人的最基本的要求。随着虚拟组织的扩大，虚拟组织的委员会会吸收更多的组织成员作为代表，帮助审核和评价新的申请加入成员。在实际组织当中，要么推举广泛了解组织成员情况及组织运行的成员作为代表，要么推举在某一方面有特殊专业才能的成员作为代表。本文根据赛百实际需求，提出两种代表推举方法

1. 推举最有经验的组织成员最为代表，即 pes_j-nes_j 值最高的成员。pes_j- nes_j 越高，说明成员 j 越活跃，与之发生交互行为的成员就越多，根据其交互经验

做出的判断越全面和可靠，也更有可能识别出恶意成员。

2. 推举有特殊可靠性需要的成员作为代表。在虚拟组织应用中，成员在组织中的重要性是不同的，而一般来说发挥作用越大，提供的资源越多的成员对安全性的需求越高。因此即使这些成员不是很有经验（也就是 pes_j-nes_j 值比较小），也需要着重考虑这些成员的安全性需求，把他们推举为代表，从而可以使得其意见得到充分的考虑。

以上是如何推举代表的方法论研究，但是在实际应用中，这两种因素往往需要综合考虑，例如一些计算中心同时属于多个虚拟组织，为大量的科学应用提供计算服务，作为重要虚拟组织成员，其对来访用户的可信任度和可靠性有着比较高的要求；同时由于完成了大量交互，也可以说他是最有经验的，综合考虑这两点而推举其作为代表。所以推举代表的过程有着比较强的主观性，不同的应用推举的具体方法有着很大的不同。一般来说越是活跃，信誉度越高的成员组成的代表其判断评价结果越可靠。不过为了测试其性能，本文在第 5 章的试验中采用随机推举虚拟组织成员作为代表的方法来测试 CMESM 的性能，其结果也更具有代表性。

3.3.2 知识库

赛百平台中，除了由平台维护的成员特征向量，每一位成员也需要维护个人的知识库。这个知识库记录了与之交互成员的基本信息和交互情况。图 3.2 显示了知识库的内容以及维护流程。

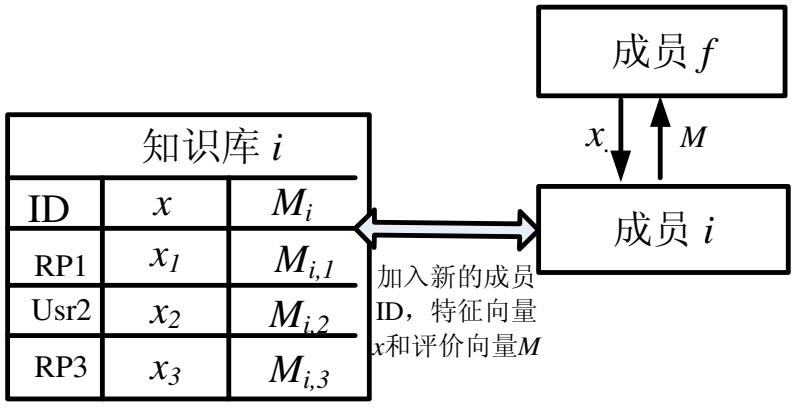


图 3.2 成员知识库

对于成员 i 来说，其知识库为 Learning Table i。每一行为一条记录，记录了曾与之发生交互行为的成员信息及交互情况。$x_j=(pes_j, nes_j)$为对方成员交互式的特征向量，由 M_i 是成员 i 对与之发生交互成员的判断，由三个元素 $r_{i,j}$, $d_{i,j}$, $u_{i,j}$ 组成。

知识库是随着用户交互行为的增加而增加和改变，当有新的成员与之发生交互，用户 i 首先读取新成员当前特征向量 x；在交互完成之后或者人为，或者通过一定的设置做出判断 M；最后将这些信息存入到个人的知识库当中。知识库将成为代表做出判定的主要数据依据。

3.3.3 判定器

如何根据待审核成员的统计数据对其进行准备的评价是很多虚拟社区都遇到的一个问题。在 Amazon，EBay 或者 Taobao 等著名 B2C 社区，系统向用户提供了商家成功交易次数和失败交易次数，以及由此计算出来的成功交易比例，用户通过这些数据进行人为的判断来评价商家是否可靠。但是这种判定有很强的主观性，而且容易被欺骗，例如采用共谋欺诈行为获得很高的成功交易次数从而欺骗用户。本文设计了基于模糊数学和模式识别算法的判定器，将代表的经验量化，从而在体现用户个性化认知的情况下客观，量化的对待审核成员进行评价。

常用的模式识别方法有 Support Vector Machine[40]，K-nearest Neighbor Classifier，Adaboost[41]和 Bayesian Classifier 等。SVM 多用在高维稀疏特征向量的识别中，例如生物信息学等。而 Adaboost 采用的是将多个弱分类组合起来以实现更加准确的分类结果，不过在训练过程中不仅需要对每一个分类器进行训练，同时还要动态调整每个分类器在最终结果中的权重，计算量相对其他算法来说比较大。Bayesian Classifier 在已知样本分布的情况下才会有较好表现，而在本文的应用中，样本数量和分布情况是无法预知的，也就很难采用 Bayesian Classifier 方法进行评价。模糊 K 近邻法（模糊 K-nearest Neighbor Classifier）是一种计算量比较小，适合用于低维特征向量分析的一种分类器，其核心思想在于从知识库中找到最相似待审核成员特征向量的已分类成员，认为这些特征相似的成员其行为模式也应该有一定的相同之处，并根据已分类成员的评价向量 M 来推算出待审核成员的评价向量。图 3.3 展示了模糊 K 近邻法的基本算法。

在赛百平台中，假定有成员 *1* 和成员 *2*，其特征向量分别为 x_1，x_2，两者相似程度用欧式距离表示，标记为$\|x_1 - x_2\|$：

$$\|x_1 - x_2\| = x_1 \bullet x_2 = \sqrt{(pes_1 - pes_2)^2 + (nes_1 - nes_2)^2} \qquad (3\text{-}4)$$

成员 *i* 为虚拟组织代表，其知识库为 $Table_i$，需要判定待审核成员 *f*。成员 *i* 的特征向量为 x_i，成员 *f* 的特征向量为 x_f。成员 *i* 在知识库中存储有 *N* 个已交互对象，其特征向量分别为$\{x_1, x_2, x_3 \ldots\ldots x_N\}$，评价向量分别为$\{M_{i,1}, M_{i,2}, M_{i,3} \ldots\ldots M_{i,N}\}$, $M_{i,j}$表示成员 *i* 对成员 *j* 的判定评价向量。

CMESM 采用基于模糊理论的 K 近邻判定器 FKNJ（Fuzzy K-nearest Neighbor Judger）作为代表层评价待审核成员的评价方法。FKNJ 通过 3 步计算出成员 *i* 对成员 *f* 的最终评价向量。

Step 1：从$\{x_1, x_2, x_3 \ldots\ldots x_n\}$中找出距离 x_f 最近的 *K* 个向量，如果知识库中样本数不够 *K* 个样本，则将知识库中所有样本作为 *K* 近邻样本参与第二步计算。

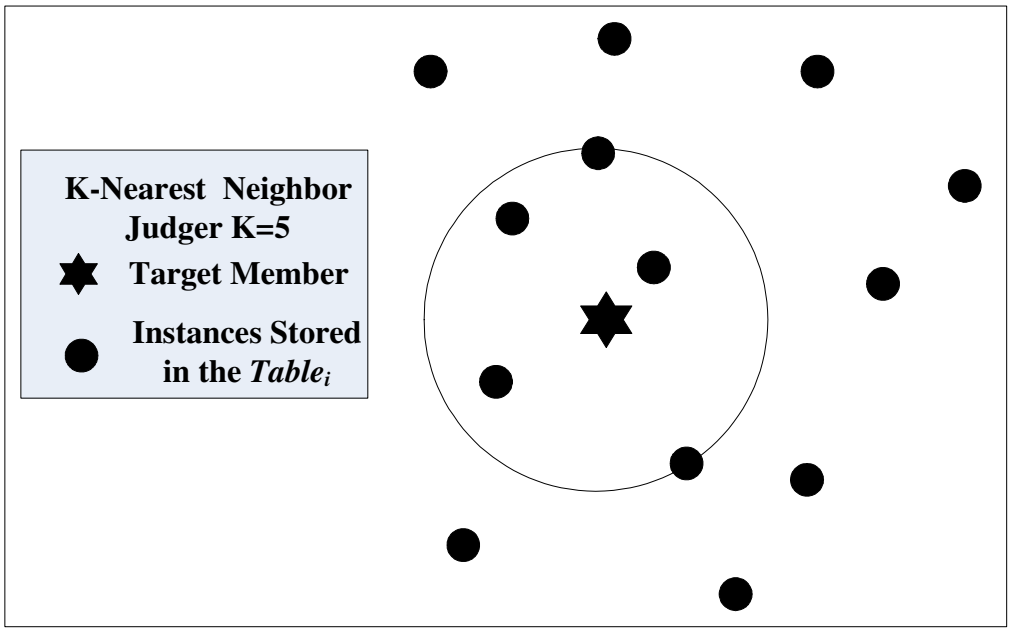


图 3.3 模糊 K 近邻法基本思路

如图 3.3 所示，FKNJ 从 $Table_i$ 中选出 *K* 个距离目标特征向量 x_j 最近的已分类特征向量，记为 $x_{j(k)}, k = 1,2,3 \ldots\ldots K$。若 *N*<*K*，则选择所有样本作为近邻样本参与下一步判定。

Step 2：假定选出的 K 近邻样本为 ，以这些样本与 x_f 的距离作为权重，分别计算评价向量中的不同元素。

$$r'_{i,f}=\frac{\sum_{k=1}^{K} r_{i,j(k)}/(\left\|x_i-x_{j(k)}\right\|)}{\sum_{k=1}^{K}\left\|x_i-x_{j(k)}\right\|} \tag{3-5}$$

$$u'_{i,f}=\frac{\sum_{k=1}^{K} u_{i,j(k)}/(\left\|x_i-x_{j(k)}\right\|)}{\sum_{k=1}^{K}\left\|x_i-x_{j(k)}\right\|} \tag{3-6}$$

$$d'_{i,f}=\frac{\sum_{k=1}^{K} d_{i,j(k)}/(\left\|x_i-x_{j(k)}\right\|)}{\sum_{k=1}^{K}\left\|x_i-x_{j(k)}\right\|} \tag{3-7}$$

从公式可以得出，已判定的样本距离目标特征向量越近，其判断评价向量对待审核成员的影响越大。特征向量的距离远近体现了两个成员之间特征向量的相近程度，所以距离越近，其行为模式也越相似。

Step 3：归一化。由于隶属度函数要求归一化，所以需要对 Step 2 计算出的结果进行归一化处理，即

$$r_{i,f}=r'_{i,f}/(r'_{i,f}+u'_{i,f}+d'_{i,f}) \tag{3-8}$$

$$u_{i,f}=u'_{i,f}/(r'_{i,f}+u'_{i,f}+d'_{i,f}) \tag{3-9}$$

$$d_{i,f}=d'_{i,f}/(r'_{i,f}+u'_{i,f}+d'_{i,f}) \tag{3-10}$$

成员 x_i 对待审核成员 x_f 进行判定后的评价向量为：$M=\{r_{i,f},\ u_{i,f},\ d_{i,f}\}$。

3.4 委员会层

当虚拟组织代表将各自判定待审核成员后的评价向量提交给委员会以后，委员会将通过一定的表决方法来综合考量各个代表的评价向量。在实际应用中，人们通常采用投票，或者有加权的投票方式来进行表决。本文在委员会层采用模糊决策方法[42]（Fuzzy Decision Making Scheme，FDMS）方法来总结代表的意见，做出合适的判定。FDMS 有以下两个优点：

1. 处理不确定，残缺和模糊的信息。由于引入了模糊数学的概念，在意见表达和处理过程中采用隶属度等带有程度描述的数字和方法，可以尽可能的利用不确定性和模糊信息。

2. 容错性强。由于代表都选择虚拟组织中的普通成员，或者是受到代表知识所限，或者是推举的代表为恶意成员，所以代表有可能提交错误的判定信息给委员会。传统的方法，例如投票制度很难避免这些错误信息对最终判断结果的影响。FDMS 可以通过权重设定限定代表在委员会发挥的作用，而且委员会的判定结果依赖于所有成员的综合考虑，所以可以有效的避免这种情况的出现。

假定委员会中有 N 个代表，待审核成员标记为 f，第 i^{th} 代表向委员会提交的判定隶属度函数为 $M_{i,f}$，则 N 个代表提交$\{M_{i,f} \mid i=1,2,3.....N\}$共 N 个评价向量。此 N 个评价向量组成隶属度矩阵 R：

$$R=\begin{bmatrix} M_{1,f} \\ M_{2,f} \\ \vdots \\ M_{N,f} \end{bmatrix}=\begin{bmatrix} r_{1,f} & d_{1,f} & u_{1,f} \\ r_{2,f} & d_{2,f} & u_{2,f} \\ \vdots & \vdots & \vdots \\ r_{N,f} & d_{N,f} & u_{N,f} \end{bmatrix} \tag{3-11}$$

为了描述代表在委员会中不同的重要性，本文引入权重向量 $A=(a_1,a_2,.....a_n)$。 a_i 为 0 到 1 实数，1 表示委员会完全考虑此代表意见，0 表示委员会完全不考虑此代表意见。

委员会做出的判定评价向量初步结果表示为 $B'=\{r',d',u'\}$，归一化后的结果表示为 B，通过 $M(\wedge \vee)$算计进行计算。

$$B^{'}=A\circ R=\{r',d',u'\} \tag{3-12}$$

具体计算方法为：

$$r' = \max_i(\min(a_i, r_{i,j})) \quad i = 1,2,3\ldots N \tag{3-13}$$

$$u' = \max_i(\min(a_i, u_{i,j})) \quad i = 1,2,3\ldots N \tag{3-14}$$

$$d' = \max_i(\min(a_i, d_{i,j})) \quad i = 1,2,3\ldots N \tag{3-15}$$

$\min(a_i, r_{i,j})$表明第 j 个代表能够向委员会提出待审核成员 f 隶属可靠成员的隶属度不能超过其权重 a_i，也就是说代表在委员会的呼声不能超过 a_i，这样在一定程度上消除了由于个别代表极端意见（例如被恶意待审核成员收买，有强烈个人偏见等）而导致委员会错误判定情况的出现，使得委员会在判定过程中更加稳定和抵御共谋欺诈行为。评价向量中的其他元素采用相同的算法，$\min(a_i, d_{i,j})$ 和 $\min(a_i, u_{i,j})$。

对 B'进行归一化，

$$r = \frac{r'}{r' + d' + u'} \tag{3-16}$$

$$d = \frac{d'}{r' + d' + u'} \tag{3-17}$$

$$u = \frac{u'}{r' + d' + u'} \tag{3-18}$$

$B=\{r,d,u\}$为委员会对待审核成员 f 做出的最终判定评价向量。$r+d+u=1$。其中 r 表示委员会在多大程度上认为成员 f 为可靠成员，u 表示委员会在多大程度上认为成员 f 为不可靠成员，d 表示委员会在多大程度上认为成员 f 需要进一步判定。

虚拟组织的委员会或者管理员可以设置一定的阈值$\{T \mid 0.5 \le T \le 1\}$来根据 B 来判断采用何种措施来应对待审核成员 f。如果 B 中某一项大于 T，则认为待审核成员隶属于相应类，例如设定 $T=0.6$，$B=\{0.7，0.1，0.2\}$，通过比较有 $T>r=0.7$，则可以认为待审核成员隶属于 r 类，即为可靠成员，可以批准加入。

权重向量在委员会评价中限定了代表意见上限，通过特定设置，限制了某些代表发表的相对极端的意见而突出重要代表的意见。例如在某一虚拟组织中委员会由代表 R_1 和 R_2 组成。R_1 为虚拟组织重要成员，需要充分考虑其意见，设定其权重为 0.9；同时另一位代表 R_2 重要性相对要小，设定其权重为 0.5。加入 R_1 判定待审成员的评价向量为 M_A={ $r=0.1, d=0, u=0.9$ }，B 判定此待审成员的评价向量为 M_B={ $r=0.9, d=0, u=0.1$ }。通过 $\min(a_i,u_{i,j})$，$\min(a_i,d_{i,j})$ 和 $\min(a_i,u_{i,j})$ 计算后，R_1 提交的判定评价向量变为 M'_A={ $r'=0.1, d'=0, u'=0.9$}，R_2 提交的判定评价向量变为 M'_B={ $r'=0.5, d'=0, u'=0.2$ }，其值最高的隶属度元素从 $r=0.9$ 变为 $r'=0.5$，限定了 R_2 在委员会中发挥的作用，而 R_1 具有更高的话语权。最后委员会给出的未归一化评价向量为 B'={$r'=0.5, d'=0, u'=0.9$}，归一化后为 B={$r=0.357, d=0, u=0.643$}。若假定阈值为 T=0.5，那么通过委员会判定，此待审核成员为不可靠成员，不能批准加入。但是如果仅仅根据 R_2 进行判定，此待审核成员为可靠成员，可以加入到虚拟组织当中。

第 4 章 联邦模式多级虚拟组织管理

虚拟组织的访问控制和一般的访问控制方法，例如强制访问控制的等相比，更适合当前大规模的权限管理应用，第 3 章介绍了如果快速准确的对陌生成员进行针对特定应用的评价，从而能够构建有适当信任度的虚拟组织，解决了动态性和信任度的问题。但随着跨学科研究和多层级学科研究，现代大型的科研项目需要多方面的合作和支持，内部大多分为若干小组进行研究。与这种大型科研活动的这种模式相对应的就是赛百成员的多级虚拟组织管理。例如 LIGO 目前就支持和管理近十个子科研小组，分别面向特定的子问题进行有针对的研究。多级的虚拟组织安全管理以支持科学应用已经成为当前科学计算应用领域突出的问题。主要挑战在于如何保证某一虚拟组织加入到更高一层虚拟组织后原虚拟组织内部成员的安全性不会受到威胁，权限不会被泄露。加入更高一层虚拟组织意味着更多的成员可以访问本地成员，如果没有安全有效的管理机制会造成某不被信任成员通过更高层虚拟组织访问本地成员，造成信任危机。针对这种情况本节介绍了一种基于联邦模式的虚拟组织管理方案，有效的改善现有虚拟组织管理方案的缺陷，更好的支持了多级虚拟组织管理和虚拟组织间合作等应用。

4.1 联邦模式多级虚拟组织模型

图 4.1 展示了一个典型赛百平台中出现的各种成员以及相关关系。在默认情况下，赛百平台中所有成员都隶属赛百资源池，成员之间并没有建立任何信任关系（也就是无法进行资源共享）。当有一个成员，可能是 User 或者 RP 由于某种科学计算应用需要查找和整合相关计算资源和研究人员共同研究和计算，一个针对性的虚拟组织将建立起来。VO 创建人（Virtual Organization Founder， VOF）通过搜索赛百资源池查找到相关 RP，User 和 VO，通过主动邀请方式邀请这些赛百成员加入到这个 VO 组织当中，同时其他成员也可以主动申请加入特定虚拟组织。当某一项科研应用完成，那么对应的虚拟组织也完成了其作用，会随着实际需求的结束而结束，进而把 VO 中的资源释放出来。

在科学计算应用中也会出现跨学科合作和划分多个子项目进行研究的

情况。一个典型的例子就是 LIGO （Laser Interferometer Gravitational-wave Observatory），首先它作为一项大型科研项目受到美国 OSG （Open Science Grid），TeraGrid 的大力支持，而仅仅 OSG 就支持了包括 LIGO 在内的超过 20 项大型科学应用项目；同时 LIGO 内部根据研究进展也成立了近十个子科研小组，例如 Burst Group，Omega Pipeline Group, Instrument Group，等。不论 LIGO 还是 LIGO 内部的 Sub-Group 都有着一定的生命周期，他们根据实际需求动态的创建和取消。图 4.1 显示了一个典型赛百平台虚拟组织结构图。从图中可以看到，VO2，VO3 同属于更高层的虚拟组织 VO1，同一个成员或者 VO 也可能隶属于不同的虚拟组织，例如 VO2 也隶属于 VO4。总的来说虚拟组织结果反映了应用中实际组织需求。

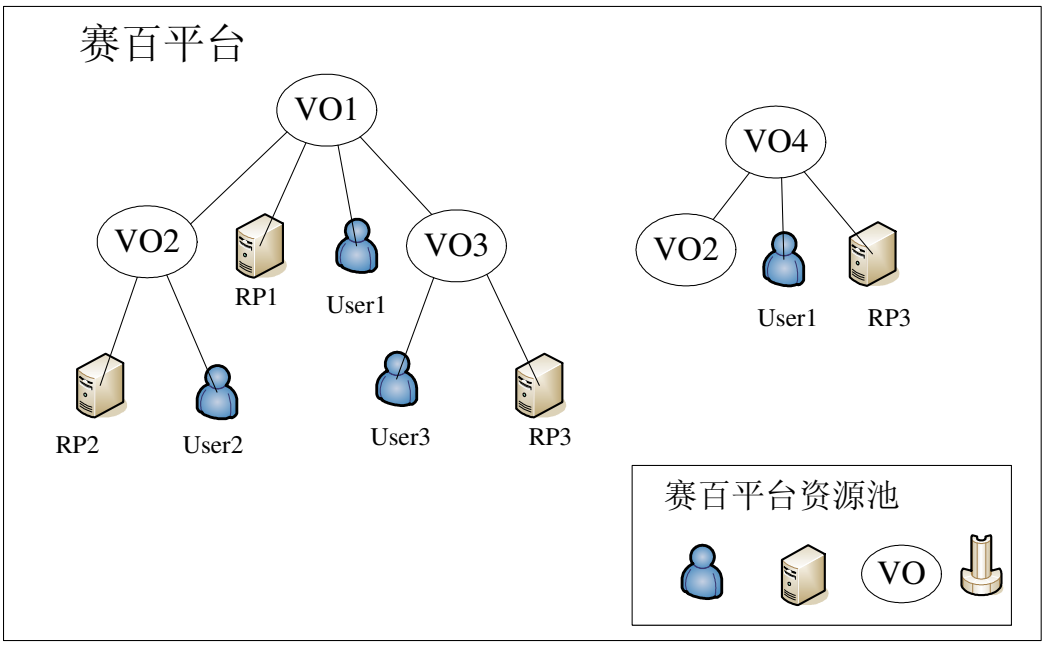


图 4.1 典型赛百平台成员及其组织关系

前文提过，多级虚拟组织面临的重要问题是如何解决在虚拟组织加入或者退出的过程中保持子虚拟组织成员的安全和信任需求。简而言之虚拟组织层组织关系的变更不能够影响到个体成员在信任度上的需求。以图 4.1 为例，假定 RP2 不信任 User1，即不希望其利用自己的资源，在 VO2 虚拟组织构建的时候通过 CMESM 可以保证 User1 不会加入到 VO2 中，但是如果 VO2 作为整体加入到 VO1 中，那么就有可能出现 User1 与 RP2 同

处于一个虚拟组织当中，其结果为 User1 可以通过 VO1 访问到 VO2 内 RP2，造成安全隐患。针对以上情况，本节提出基于联邦模式的多级虚拟组织构建和管理方法，以更好的满足实际科学计算应用需求。联邦模式（Federal Architecture）是一种现代多级组织结构形式。联邦是由若干个达成一定协议的子组织组成，组成联邦的各个子组织对其内部成员和资源有着更高权限的支配权，联邦作为协调方帮助各个子组织之间完成合作和沟通。当联邦发布的命令与子组织的命令发生冲突时，以子组织的命令为有效命令。在这种情况下，子组织在权限上相对独立，可以根据实际作为一个独立个体自由的进行组合或者加入到新的联邦。

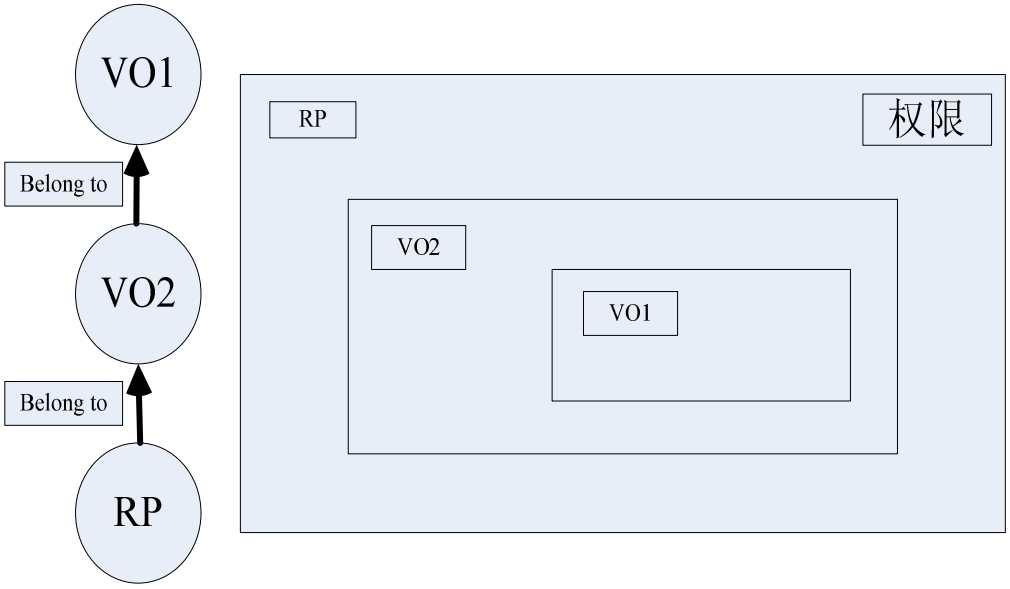


图 4.2 权限拓扑结构图

赛百平台联邦模式多级虚拟组织管理模式其主要特点为自底向上的权限分布，如图 4.2 所示。赛百平台和常规意义上的数据中心（Data Centre）和 Cloud Computing 不同，运行在平台中的计算资源始终隶属于不同的 RP，赛百平台仅仅提供资源共享使能环境，因此 RP 拥有对其资源的最高权限。同时对于多级 VO，底层 VO 对其内部资源拥有的权限也应该不低于上级 VO 成员对相应资源的权限。这是为了保证子虚拟组织内部权限的安全性和子虚拟组织在运行时的独立性。假定上级虚拟组织可对子虚拟组织成员

拥有高于其内部成员的权限，那么当上级虚拟组织以子虚拟组织的身份加入到更高虚拟组织当中，就会造成底层虚拟组织权限被无限好的扩大和泄露，不管是从管理上还是从安全性上都是不可取的。所以多级虚拟组织在应用上呈现出自底向上的权限分布情况，如何在赛百平台使能这种分布并提供简单安全的管理支持是本节要解决的一个问题。在图 4.2 中，假定 RP 隶属于 VO2，而 VO2 作为 VO1 的子组织，隶属于 VO1。在联邦模式下，RP 对其资源有着最大的权限和最高的话语权，而 VO2，VO1 的直属成员对 RP 资源的支配权限是依次递减，形成图 4.2 所示权限拓扑结构图。

4.2 联邦模式多级虚拟组织管理方法

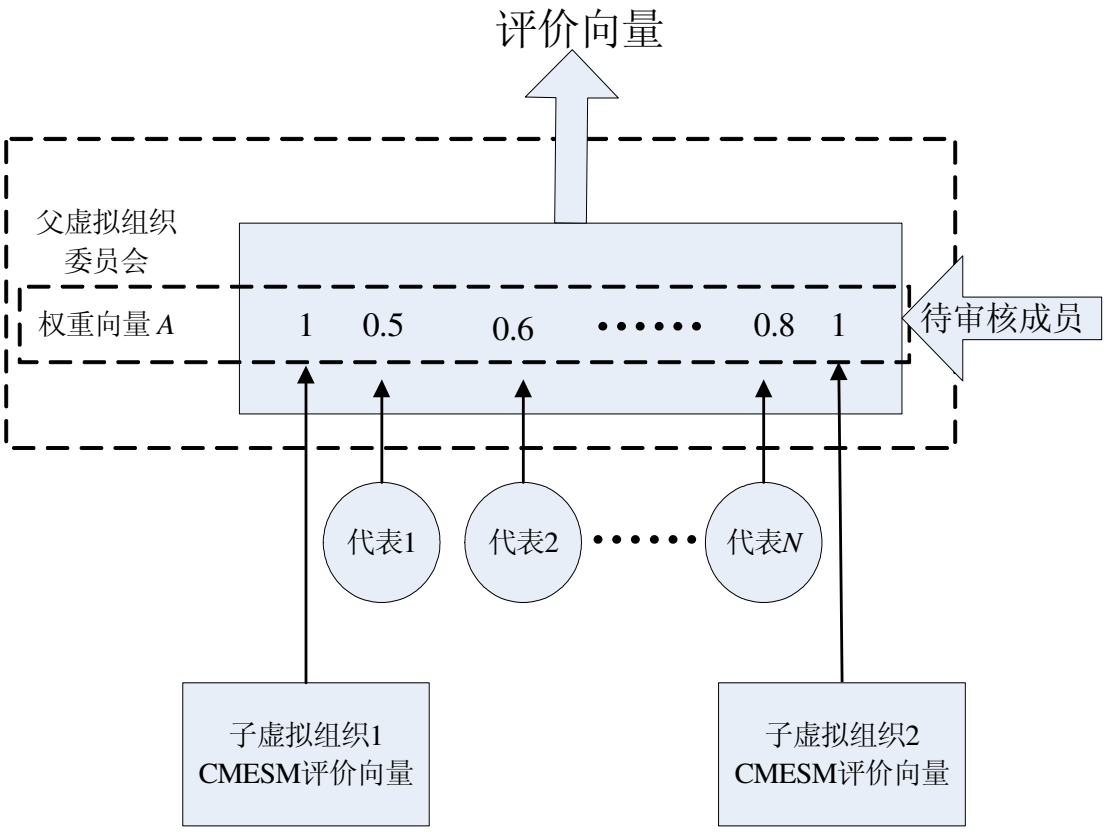


图 4.3 HCMESM 结构图

联邦模式多级虚拟组织模型其核心在于本地组织或者成员和父虚拟组织成员相比拥有本地资源的更大权限，从组织关系的维护上来讲，父虚拟

组织加入的成员如果希望共享子虚拟组织内部资源，必须通过子虚拟组织的认证和授权。根据虚拟组织的定义，其内部成员之间达成高度共享协议，加入虚拟组织即意味着可以自由访问虚拟组织内部所有资源，包括各个子虚拟组织资源。在这种情况下，父虚拟组织成员满足子虚拟组织信任度要求是其能够加入父虚拟组织的必要非充分条件。据此作者设计了多级委员会成员管理方法 HCMESM(Hierachical Committee-based Member Evaluation and Selection Method)，帮助赛百成员建立和管理安全的多级虚拟组织。

根据 CMESM 的定义，任何一个虚拟组织都会构建专有的委员会对内部成员进行信任度管理，参见第 3 章。假定虚拟组织 VO1 和 VO2 由于应用需求，共同加入了虚拟组织 VO3。那么 VO1 和 VO2 为子虚拟组织，VO3 为父虚拟组织，不论是子虚拟组织还是父虚拟组织各自维护其委员会。图 4.3 显示了 VO1 和 VO2 加入 VO3 以后，VO3 作为父虚拟组织其委员会的变化情况。

按照前面说明，VO1 委员会对成员的评价是其他成员可否访问 VO1 内部资源的必要条件，因此当 VO1 加入到 VO3 后，为了保证父虚拟组织成员能够直接访问 VO1 内部资源，就必须要求 VO3 直属成员在满足 VO3 委员会要求的基础上也要满足 VO1 的要求。因此 VO1 的委员会作为单个代表成为父虚拟组织委员会中的一员，向上一级提供评价向量。由于子虚拟组织在评判过程中拥有否决权，所以 VO3 委员会赋予 VO1 代表最大的权重。权重为 1 保证了 VO1 做出的评价向量可以完全的反映在 VO3 委员会做出的评价向量中来。例如假定 VO1 认为特定成员是不可信任成员，其评价向量为 $M_1=\{0,0,1\}$。在 M_1 的影响下，在 VO3 委员会做出的评价向量 M_3 的第一个元素即 $r_{i,j}$ 不会大于 0.5，即待审核成员隶属于可靠成员的隶属度不会超过 0.5。在这种情况下就避免了不满足 VO1 却满足 VO3 其他代表要求的待审核成员加入 VO3 进而访问 VO1 内成员的情况。同理，VO2 的委员会也会作为单个代表参与到 VO3 委员会的评价过程中来。

HCMESM 很好的实现了联邦模式的多级虚拟组织管理，确保了虚拟组织在与其他虚拟组织合作的过程中（加入更高级虚拟组织）不会发生权限方面的泄露，确保成员和子虚拟组织的独立性和安全性。此外 HCMESM 也具有很好的可扩展性，可以通过嵌套，满足任意层虚拟组织管理的需求。

最后由于 HCMESM 采用分层评价模式，由各级虚拟组织委员会独立做出评价，最高层虚拟组织的委员会仅仅承担汇总和计算功能，实现了负载均衡，避免出现单点负载过大情况的出现。

第 5 章 CMESM 性能评价

本文通过两个方面来衡量 CMESM 性能。

首先是 CMESM 的准确性和稳定性。CMESM 为赛百成员提供成员审核判定服务，所以它必须能够准确的判断出待审核成员的真正行为模式，拒绝和抵御欺诈行为特别是共谋欺诈行为（如何抵御共谋欺诈行为，例如平台刷等成为 EBay，Taobao 等虚拟社区面临的重大挑战）。此外 CMESM 也要能够适应多种不同的虚拟社区环境，并在不同参数下（例如 K 的选择）下性能稳定。本文 5.2 节和 5.3 节分别测试了在不同参数和不同仿真环境中 CMESM 的性能表现情况。

其次是 CMESM 的是否切实提高赛百平台成员的服务质量。虚拟组织对申请成员进行严格审核，防止恶意成员加入，其最终目标是实现了虚拟组织内部成员的高可靠性。这种可靠性提高了虚拟组织内部成员在信任方面的开销，提高了内部成员交互质量。为了说明这一点，本文通过测量通过不同方法建立的虚拟组织（分别由通过 CMESM 审核筛选的成员，由完美成员和由随机成员组成的虚拟组织）中执行 Workflow 所需 Makespan 来说明 CMESM 在提升虚拟组织服务质量方面的性能。

本章节中定义以下指标：

$$Positive\ Error(PE)=\text{错分为可靠成员的不可靠成员数理} \ / \ \text{不可靠成员总数量;} \tag{5-1}$$

$$Negative\ Error(NE)=\ \text{错分为不可靠成员的可靠成员数理} \ / \ \text{可靠成员总数量;} \tag{5-2}$$

$$Makespan:\ \text{将一组任务分配到一组资源上，所有任务完成所花费的总时长。} \tag{5-3}$$

PE 和 *NE* 代表了 CMESM 分别对不可靠用户和可靠用户进行审核时的错误率，*PE* 是把不可靠用户分为可靠用户的比例，*NE* 把可靠用户分成不可靠用户的比例。Makespan 通常用来描述调度算法的效率。在本文中我们用来证明通过 CMESM 筛选的资源具有更高的可靠性。

由于 CMESM 性能只有通过在大规模虚拟社区中应用才能体现出来。限于条件，本文采用设计仿真平台和实验方案来对 CMESM 性能进行测试。

5.1 仿真测试环境

本文通过建立仿真赛百平台环境来对 CMESM 进行性能测试。赛百平台是一个交互平台，成员通过建立和加入虚拟组织来构建适当的信任关系和进行交互。表 1 列出了在仿真平台中用到的参数设置。

1. 仿真平台首先对成员行为模式进行定义。若某一个成员能够提供完全可靠的服务，那么这个成员被定义为完美成员。对于 RP 来说，完美成员表示资源能够提供完全可靠的服务；对于 User 来说，完美成员表示其提供的任务是完全可靠和安全的。在本文中，任何一个成员，User 或者 RP 的行为模式通过 $PT=(pt_1, pt_2)$来定义，pt_1 表示成员与完美成员进行交互，交互为高质量交互的概率，pt_2 表示成员与完美成员进行交互，交互为低质量交互的概率，有 $pt_1+ pt_2=1$。

2. 在论文[43][44]中说明，一般交互平台中成员交互次数符合指数分布，即 Power-law Distribution。因此在本文中假定赛百平台成员交互次数也服从指数分布，其期望定义为期望交互次数。CMESM 进行审核之前赛百平台期望交互次数的多少对审核准确度有着明显的影响。

3. 在仿真平台中考虑到了骗子成员对 CMESM 的影响。骗子成员完成高质量交互的能力很低，从行为模式上来看就是 pt_1 比较低而 pt_2 比较高。骗子成员通过共谋的欺诈行为来骗取高质量交互完成次数的增长。例如骗子 User 向共谋的骗子 RP 提供虚假的任务，并声明任务很复杂需要大量时间才能完成，共谋的骗子 RP 在接收到虚假任务后不进行运算或者进行很少的运算而把结果返回给骗子 User，从而双方获得一次高质量交互次数的增加。这种共谋行为已经成为 EBay，Taobao 等网站在维护诚信方面的重大挑战。赛百平台本质上也是一种交互平台，所以需要 CMESM 能够很好的识别出有虚假高质量交互次数的骗子成员。在本文中，骗子成员的共谋欺诈行为通过欺诈频率来定义：1 次虚假高质量交互/每 10 次正常交互。

4. 此外还有针对 CMESM 的一些设定，包括阈值 T 的选择，委员会中代表数量和 K 近邻法中 k 的选择等。本文在下一章节将通过改变其中某一个值来说明 CMESM 参数对其性能的影响。

为了得到可靠和具有统计意义的结果，如果没有特殊说明，本文中所有仿真结果都是 10 次结果的平均值。

表 5.1 仿真实验设置

成员数量:	500
欺诈频率:	1 *pf* per 10 interactions
阈值 T:	0.5
可靠成员: pt_1	0.97-0.99
可靠成员: pt_2:	1- pt_1
不可靠成员: pt_1	0.9-0.92
不可靠成员: pt_2:	1- pt_1
工作流中任务数量:	100
委员会中代表数量:	5
k:	3
期望交互次数	100
不可靠成员比例:	30% (150 unreliable members)

5.2 CMESM在不同参数下的性能表现

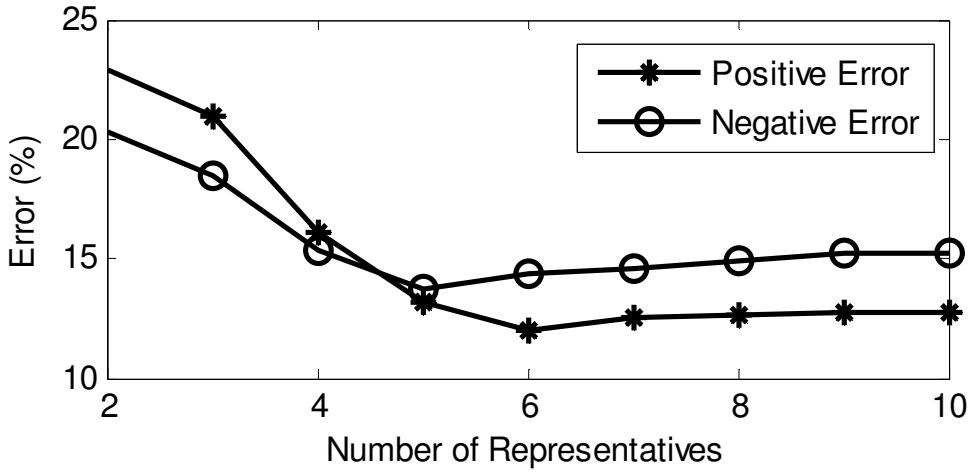


图 5.1 委员会代表数量对 CMESM 的影响

图 5.1 说明了委员会代表数量的不同对 CMESM 性能的影响。一般自然的想法是代表数量越多，其准确度也应该越高。加入所有虚拟组织成员都变为代表，那么就变成了全部成员共同参与投票的模式。从图中可以看出，在代表数量小于一定程度的时候，代表数量的增加可以显著改善 CMESM 审核的准确度。在本文实验中可以看出代表数量小于 6 时代表数

量增加对 CMESM 准确度的该晒最大；当代表数量超过 6 以后，代表数量的增加基本上不会对 CMESM 的性能产生明显影响吗，甚至还略有负面影响。这是因为当代表数量增加时，不重要或者不准确的代表的判定意见会降低重要代表或准确代表判定意见的权重，从而造成最后委员会的判定结果没有改善甚至略有恶化的结果。此外委员会数量增加也会造成计算量的增加。

代表数量与期望交互次数和虚拟组织规模有很大的关系，不同情况代表数量对 CMESM 性能改善情况也各不相同。

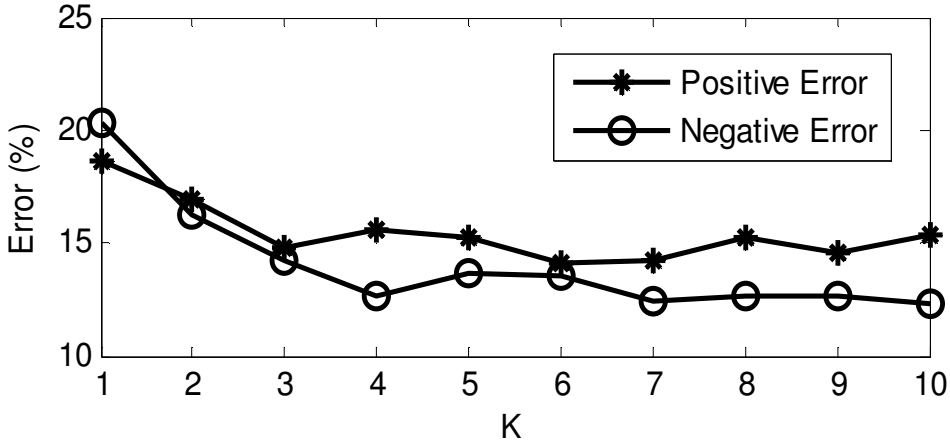


图 5.2 k 对 CMESM 的影响

图 5.2 显示了采用 K 近邻法作为判定方法，参数 *k* 对 CMESM 性能的影响。从图中可以看到，随着 k 的增加，CMESM 的准确度有较大的提高。在极端情况下，*k=1*，这是判定方法为最近邻法，从图中可以看到此时的 *PE* 和 *NE* 都比较大，均大于 18%。当 k 在 2 和 6 之间时，k 的增加明显提高了 CMESM 的性能。这是因为当 k 增加以后，代表会从知识库中选出更多的相似样本，增加的信息帮助代表做出更加准确的判定。此外由于是采用模糊 K 近邻法,所以 *k* 也可以选择偶数。和委员会中代表数量对 CMESM 性能影响情况类似，当 *k* 增加到一定数量后，*k* 的增加不能够对 CMESM 造成显著影响。本实验中当 *k* 大于 6 以后，*PE* 和 *NE* 基本上保持稳定不在发生改变。有两个原因造成了这种情况。1. 当 *k* 增加到一定数量后，新增加的近邻样本距离目标样本距离较远，不适合作为近邻样本提供参考意

见。2. 在采用 4.3.3 节 step3 公式计算代表最终判定过程中，如果近邻样本数量很多，每一个样本在最后判定中的权重减小，新增加的样本由于距离较远，影响力就更小。

5.3 CMESM在不同赛百环境中的性能表现

赛百平台中可能含有不同数量的骗子成员，其平台运作时间，及期望交互次数也有可能不同。本节测试 CMESM 两方面的性能：1. 赛百平台的成熟程度对 CMESM 性能的影响。一个交互平台越成熟，其内部成员直接交互越多，在仿真参数上的反映就是期望交互数量越高。一个不成熟，即没有经过很好运用的平台是很难分别出不同用户的差别，而好的方法对期望交互数量要求会比较低，而且能够很快的达到稳定值。2. 针对当前出现的共谋欺诈行为测试 CMESM 能否识别出共谋欺诈获得高评价的骗子成员。

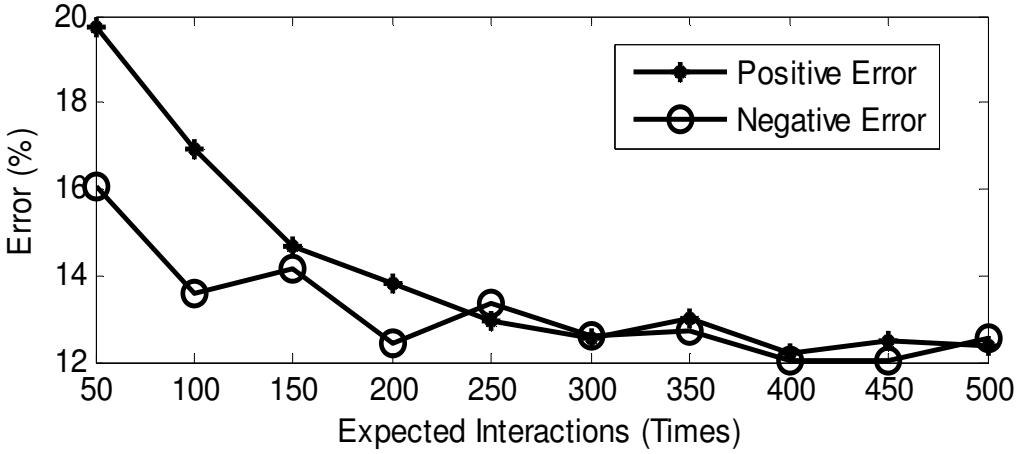


图 5.3 期望交互次数对 CMESM 的影响

图 5.3 显示了在不同期望交互次数下 PE 和 NE 的变化曲线图。在这个实验中，假定在赛百平台中有 30%的不可靠成员，即 30%的成员为不可靠成员。经过 CMESM 以后，超过 84%的不可靠成员被识别出来和摒弃掉。在虚拟组织中，只有 6.42%的成员为不可靠成员（200 期望交互次数），相比较在为审核情况下赛百平台中 30%不可靠成员，CMESM 大幅度的降

低了不可靠成员加入虚拟组织的危险。不过一直想对的是也有一定比例的正常可靠成员被错分为不可靠成员，在期望交互次数为 200 的情况下可以得知，有 14%的可靠成员被错分为不可靠成员，即 *PE=14%*。不过随着赛百平台不断成熟和使用，期望交互次数也会不断增加，与之相对应的 CMESM 准确度增加，*PE* 和 *NE* 随之降低。

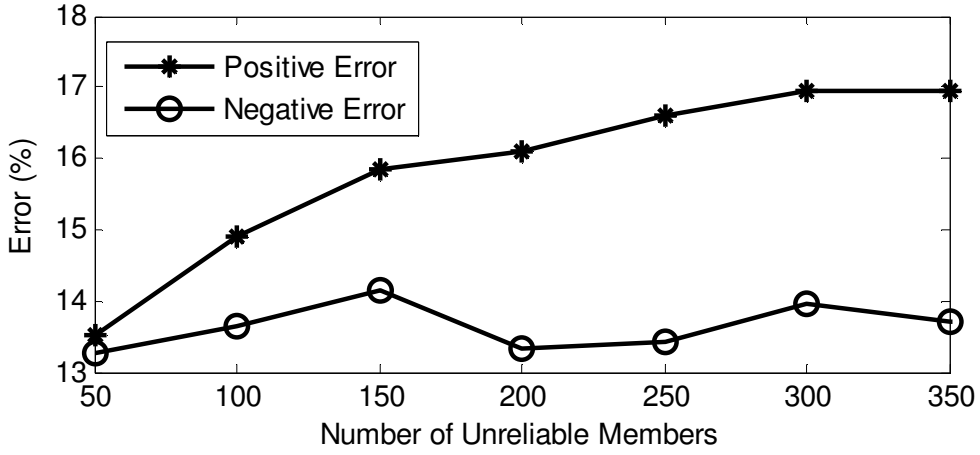


图 5.4 不可靠成员数量对 CMESM 的影响

图 5.4 显示了不可靠成员在所有赛百平台中所占比例对 CMESM 性能的影响。从图中可以看出，随着不可靠成员所占比例的增加，CMESM 的 *PE* 也随之增加，这是因为共谋欺诈行为获得了虚假的高质量交互数量，扰乱了 CMESM 对可靠成员的辨识。不过不可靠成员所占比例对 *PE* 的影响呈递减趋势，说明 CMESM 有比较高的鲁棒性。从图中也可以看出，不可靠成员数量对 NE 的影响很小，CMESM 基本上可以保证即使在不可靠成员数量很多的情况下仍然可以稳定，准确的识别出它们。NE 的最大值和最小值分别为 14.14%和 13.26%，方差为 0.306%，可以基本判定 NE 对不可靠成员占赛百平台成员比例是鲁棒的。

5.4 CMESM对Makespan的提高

本节测试通过 CMESM 审核在实际任务调度中发挥的作用。CMESM

通过代表和委员会对特定成员进行审核，筛选出可靠的成员使之加入到虚拟组织当中，而剔除那些有可能带来不稳定任务或者服务的成员。对于普通成员，特别是科学计算的用户来说，他们最关心的是通过 CMESM 可否使得任务能够在更短的时间内完成。本节采用两种不同的调度算法对任务组进行调度和仿真，比较通过不同方法筛选出的资源完成任务花费的 Makespan 差别。实验结果证明采用 CMESM 筛选机制可以有效的提高 Makespan，提高资源利用率。

5.4.1 两种广泛使用的任务调度算法

Min-Min 算法：

Min-min 优先对完成时间较小的任务，适用于调度完成时间差异较小的任务组。

假定有 N 个的任务组成了一个任务集合为 $T=\{t_1,t_2,t_3.....t_n\}$，可支配 M 台主机 $S=\{s_1,s_2,s_3....s_m\}$。若 T 不为空时循环执行：

Step1: 对集合 T 中每一个待调度任务 t_i 计算在所有 M 台主机上的完成所需时间（包括排队等候时间），得到一维预测执行时间向量 $D_i=$（$d_{i,1},d_{i,2},d_{i,3}.....d_{i,m}$）。

Step2: 从 D_i 中选出最小的元素, 此数据为任务 i 在 M 台主机上最短完成时间。

$$d_{i,j_i}=\min(d_{i,1},d_{i,2},d_{i,3}.....d_{i,m}) \qquad (5\text{-}4)$$

Step3：比较 d_{i,j_i},令 $d_{k,v}=\min_{i=1}^{N}(d_{i,j_i})$，则将第 k 个任务调度到第 v 台主机来完成。从 T 中删除任务 k，更新等待时间。

Max-Min 算法

Max-min[45][46]优先调度完成时间较长任务，适用于调度完成时间差异较大的任务组。

Max-min 与 Min-min 算法类似，采用相同的三个步骤来完成，在 Step3 改为，比较 d_{i,j_i}，$d_{k,v}=\max_{i=1}^{N}(d_{i,j_i})$ 令，则将第 k 个任务调度到第 v 台主机

来完成。更新等待时间，退出。

5.4.2 实验设计

在本次实验中采用三种不同的资源组来说明 CMESM 的性能。

完美组（Ideal Group）：由完美成员组成，完美组中所有成员能够成功，按期的完成任务。

随机组（Random Group）：从赛百平台中随机选择成员组成虚拟组织，按照表 1 描述的成员行为模式来确定是否完成高质量交互。

CMESM 组(CMESM based Group)：通过 CMESM 从赛百平台中筛选出成员组成虚拟组织。

为了通过数量说明 CMESM 对 Makespan 的改进，本文定义某一个组的 Makespan Ratio 为：

$$Makespan\ Ratio = \frac{对应组花费的\ Makespan}{完美组花费的\ Makespan} \quad (5\text{-}5)$$

这个值越高，说明对应组成员越可靠，提供的服务也越好。

本节实验中任务组中任务运行时间采用珀松分布，由于 Min-min 适合用于任务执行时间差异较小的任务组调度，所以在实验中采用期望为 10 的珀松分布作为任务执行时间分布情况；Max-min 适合用于任务执行时间差异较大的任务组调度，所以在实验中采用期望为 50 的珀松分布作为任务执行时间分布情况。为了测试在不同任务规模下的性能，本实验设定成员组委 100，任务数量分别为 200，400 和 600。

5.4.3 CMESM 在 Min-min 调度算法下的性能

从图 5.5 中可以看到三种不同成员组在 Min-min 调度算法下不同的 Makespan Ratio。实验采纳了表 1 的参数设置，所以 Random Group 中不可靠成员比例的期望为 30%。经过 CMESM 筛选的成员组中有不可靠成员为 8%，完美组中不含有不可靠成员。通过实验，CMESM 的 Makespan Ratio 平均值为 1.158，方差为 0.01, Random Group 的 Makespan Ratio 平均值为 1.324，方差为 0.05，大大高于 CMESM，这说明通过 CMESM 筛选后，虚拟组织可以提供更好，更可靠的服务。

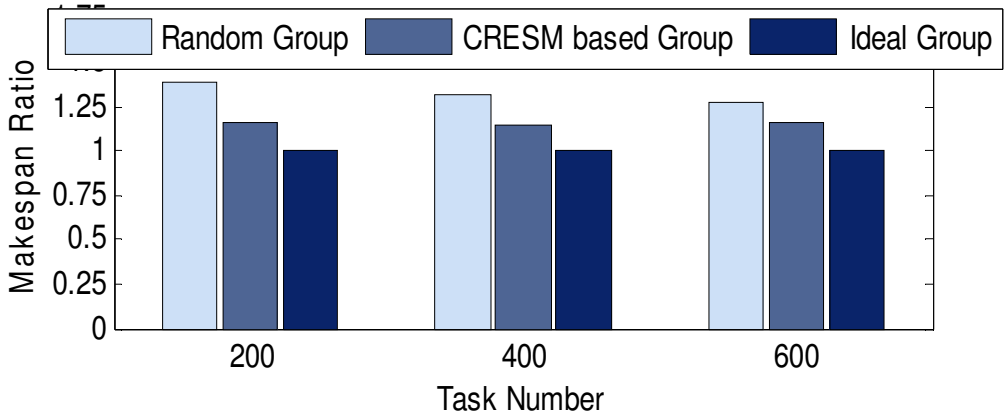


图 5.5 基于 Min-min 的 Makespan Ratio 直方图

5.4.4 CMESM 在 Max-min 调度算法下的性能

图 5.6 显示了采用了 Max-min 调度算法后各个组的 Makespan Ratio 情况。基于 CMESM 的虚拟组的 Makespan Ratio 平均值为 1.129，方差为 0.07；随机选择赛百平台成员组成的虚拟组的 Makespan Ratio 平均值为 1.212，方差为 0.1。从图中也可以看到，基于 CMESM 的虚拟组能够更快的完成任务，为应用提供更好的服务。

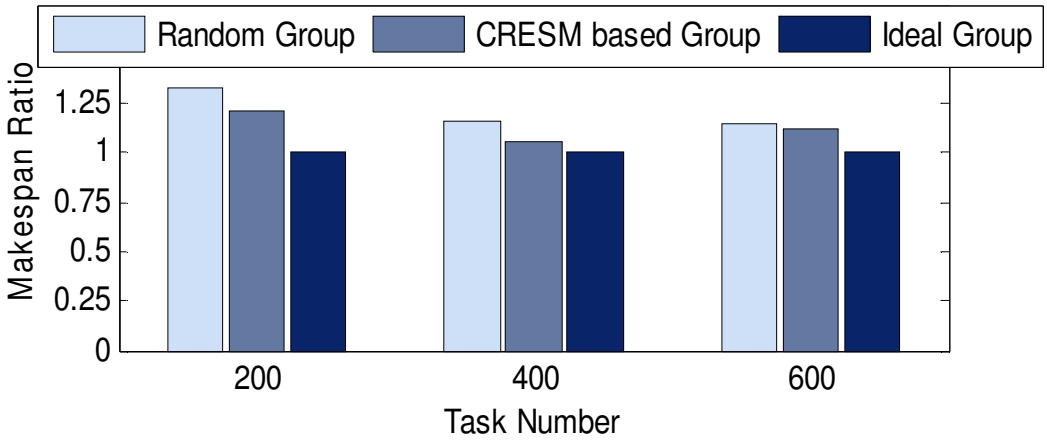


图 5.6 基于 Max-min 的 Makespan Ratio 直方图

5.5 CMESM性能总结

本节建立了一个仿真平台，通过基于现有的一些实验结果和随机理论，对成员交互行为进行了仿真和模拟，并在此基础上对 CMESM 进行了多方面的测试。5.2 节测试了 CMESM 不同参数对其性能的影响，实验表明，在一定范围内，增加代表数量和近邻数量都可以有效的提高 CMESM 的准确度，不过这种影响会随着数量增加到一定程度而迅速减小。5.3 节测试了 CMESM 在不同赛百平台下的运作情况，分别为赛百平台中含有不同比例不可靠成员，赛百平台的成熟度不同等情况。实验表明，即使在不可靠成员数量较多的情况下赛百平台仍然可以保证判定的稳定和准确，特别是在辨别出不可靠成员方面，*NE* 基本上保持稳定不会因为平台中不可靠成员数量的增加而增加。此外 CMESM 即使在赛百平台初步使用的时候也能提供比较好的服务，而且其性能可以随着赛百平台的进一步使用而显著提高。最后站在用户和实际应用的角度，测试了 CMESM 对服务质量的提高程度。5.4 节的实验说明在不同任务规模，采用不同算法的情况下，CMESM 均有效的提高了总任务完成时长（Makespan），提高了虚拟组织的服务质量。

第 6 章 赛百平台虚拟组织管理中间件

本文第 3 章提出了在虚拟组织管理中如何通过 CMESM 实现虚拟组织成员安全和建立信任关系；第 4 章讲述了联邦模式多级虚拟组织模型及其管理办法，将 CMESM 拓展为 HCMESM；在第 5 章对 CMESM 进行了仿真和性能测试。本节介绍作者根据以上方案开发的基于 SOA 结构的赛百平台中间件——动态虚拟组织管理系统（Dynamic Virtual Organization Management System），简称为 DVOMS。

DVOMS 采用 CMESM 方法，针对性的管理赛百成员的组织结果和权限。从功能上来讲是为赛百成员提供信息服务，包括虚拟组织的查询和管理，赛百成员信息维护和查询等。在整个赛百平台应用软件体系中，DVOMS 作为中间件为其他工具提供权限管理和成员查询相关的信息服务，配合各种共享使能工具来完成成员之间资源安全有效的共享。因此 DVOMS 告诉用户是否可以共享，而各种共享使能工具例如 Globus，Condor 和本实验室（清华大学信息技术研究院未来信息技术研究中心[]）正在开发的 ELOP 等软件则具体实现这种共享操作。总的来说，本文在开发 DVOMS 时将其定位于赛百中间件，相对独立但又可以配合其他现有的或者未来开发的各种使能工具共同完成复杂任务。

赛百平台下虚拟组织管理涉及到了三个逻辑参与实体：赛百管理中心 CIMC（CyberInfrastructure Management Centre），虚拟组织管理中心 VOMC（Virtual Organization Management Centre）和普通成员 Member，大多数虚拟组织管理操作和维护都要涉及到这三方的交互。图 6.1 显示了三方在操作中的逻辑关系。

CIMC 作为一个轻量级的服务中心，承担签发成员证书，管理用户信息，维护用户交互历史数据和对 VO 进行管理和提供必要信息搜索服务。面向 CI 成员，CIMC 为新加入 CI 的成员提供注册和签发证书的功能，此后对已加入的成员提供申请建立/取消 VO，搜索 CI 中的成员/VO 等信息维护和信息搜索功能，主要操作为数据库操作。此外 CIMC 也接受 CI 用户提交的交互报告，维护成员特征向量和成员知识库等成员个人信息。面向 VOMC，CIMC 除了提供和 CI 成员类似的信息搜索功能，也维护 VO

信息，例如 VO 功能，VO 管理员和构建 VO 目的等信息。和一般的服务器相比，CIMC 采用 Push 模式搜集信息，即由成员或者 VO 主动向 CIMC 报告，因此 CIMC 在系统运行过程中承担任务量较少，不会造成单点瓶颈问题，而且也加大了 VOMC 的独立性。

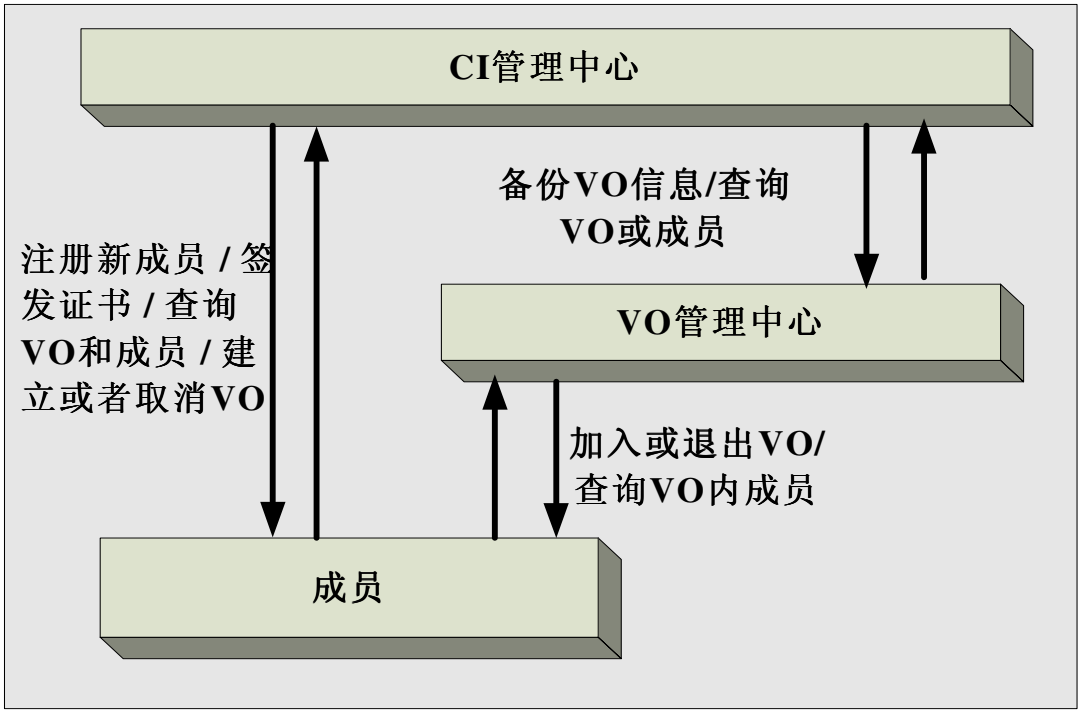


图 6.1 赛百平台，虚拟组织和成员逻辑关系

VOMC 负责维护具体 VO 内成员的组织关系并提供成员资格判定，成员搜索和维护委员会等功能。面向虚拟组织内部，虚拟组织管理员（往往就是虚拟组织的创建者）可以指定特定成员或者选定历史交互次数最多的成员由作为虚拟组织代表，多个代表组成了虚拟组织的委员会，对待申请成员进行审核判定。面向 CI 成员，VOMC 接受 CI 成员提交的加入或者退出等信息，对这些信息进行相应的处理。面向 CIMC，VOMC 主动报告 VO 信息，包括 VO 主要功能，构建 VO 目的和 VO 管理员等，此外也会在需要的时候向 CIMC 查询特定成员或者其他 VO 信息。

CI 成员是 CI 平台中一切活动的主体和 CI 服务对象。面向 CIMC，CI 成员在每次交互完成后主动提交交互报告以更新成员特征向量和知识库，

此外 CI 成员也可以向 CIMC 发出查询请求，对 CI 中成员和 VO 进行查询。作为本文相比其他类似工具的创新点，任何 CI 成员可以向 CIMC 提出构建个人 VO 的申请，CIMC 会帮助成员维护 VO 必要信息。面向 VOMC，CI 成员可以主动提出加入或者退出虚拟组织申请，发出查询 VO 内成员信息等请求。

图 6.1 展示的是 DVOMS 的逻辑图；在 6.1 节讲述了具体的软件实现架构图；6.2 节讲述了 DVOMS 采用的安全传输和认证方法；6.3 节演示了是如何通过 DVOMS 进行动态虚拟组织管理；6.4 节集中解释了如何解决软件依赖性问题，使得 DVOMS 具有较好的移植性；最后在 6.5 节演示了在 DVOMS 的帮助下，Globus 实现了更加灵活和安全的资源共享。

6.1 软件架构图

前文阐述了 DVOMS 在逻辑上的功能划分和各部分直接的交互，但是在实际开发中往往需要从技术难度，可移植性和可扩展性，以及模块化和用户使用角度等多方面进行考虑。图 6.2 展示了 DVOMS 层级结构，图 6.3 展示了 DVOMS 的软件架构。

DVOMS 大致可以分为四层和一个中心，四层结构松散耦合，从底向上分别为传输层，解析层，应用层和显示层；一个中心为认证中心（CA，Certificate Authentication）。

第一层为传输层，主要进行数据和命令的传输，为上一级提供可靠，安全的传输服务。本文将传输和接受服务封装为 SOA 结构，分为服务端和客户端。服务端监听特定端口并接受不定长度的二维字符串类型数据 vector <string>，并将之传输给相应的上级解析层；客户端通过服务端公布的接口定义文件(.wsdl)设计客户端程序，接受和传输待传输信息并等待返回信息，传给上级应用。传输协议采用轻量级 soap 协议，并通过 PKI 保证了传输的安全性。

第二层为解析层，接受底部传输层上传信息并通过预定义的配置文件对接收到的信息进行解析；接受上级应用层的信息，通过预定义的配置文件对待传输的信息进行编码，使之符合特定规范，保证接收方能够正确识别。由于预定义的配置文件在 DVOMS 使用过程中自由定义，所以保证了

DVOMS 很好的扩展性。

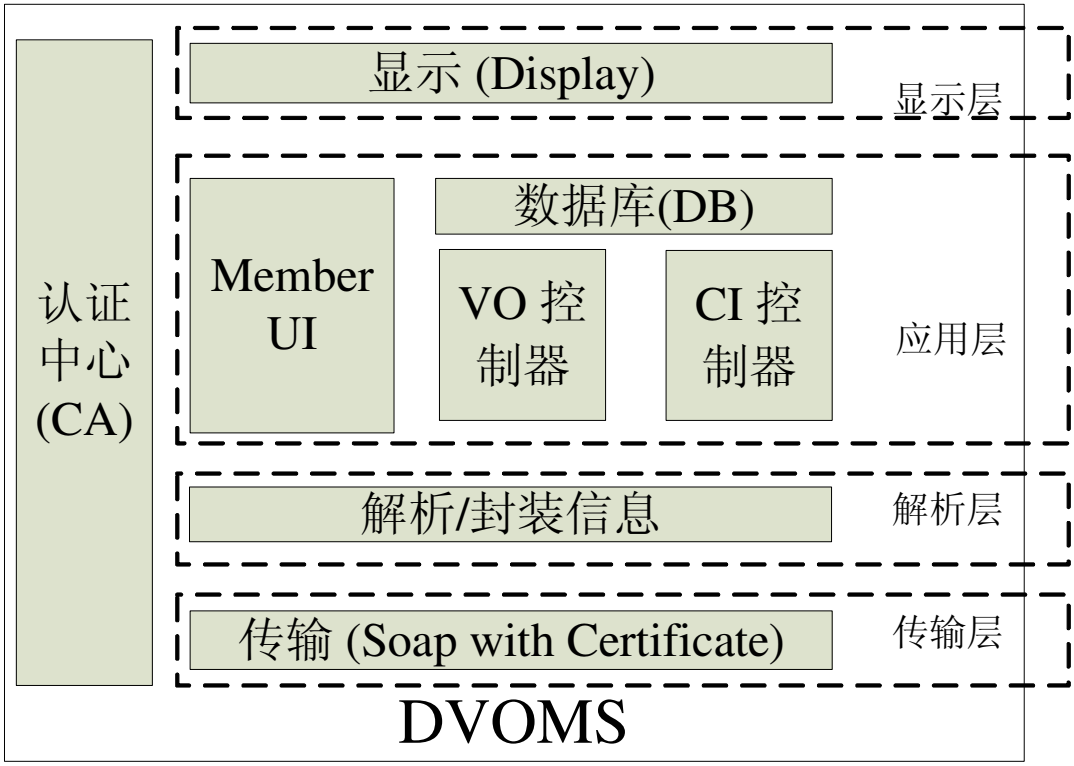


图 6.2 DVOMS 层级结构图

第三层为应用层，接受底层解析层解析好的消息，然后根据消息格式和内容调用合适的服务；在完成服务后，将服务结果和运行情况传输给解析层，解析层按照预定义的配置文件进行封装，最后交由传输层进行传输。从功能上分为 CI 成员 UI （User Interface），VO 控制器和 CI 控制器。普通成员端不需要维护数据库，由于采用的是 Push 模式，所以涉及成员的任何操作的发起人均为用户，所以仅仅需要为成员提供适当的用户界面。对于 VO 来说，VO 控制器接受其他两方（成员或者 CIMC）传输的信息并根据一定的策略进行适当的动作，例如添加新成员修改数据库，集合组织代表进行特定成员的判定等。VO 控制器通过操作 VO 数据库来维护 VO 的组织信息。CI 控制器接受底层出入的用户和 VO 请求，根据请求作出相应的操作，例如查询，修改 CI 数据库等。

第四层为显示层，主要功能是为普通用户，VO 管理员和 CI 管理员提

供显示和输入界面。作者开发中为用户提供了命令行模式和浏览器模式两种显示方法，用户可以根据个人偏好进行选择。

认证中心(Certificate Authentication)：认证中心为所有 CI 成员签发证书，通过 PKI(Public Key Infrastructure)保证任何一个成员不可以被假冒，其交互行为不可以被抵赖，传输的信息不可以被更改，是保证传输层信息安全，保证应用层能够正确识别成员表明身份的基础性服务。DVOMS 提供了完整的认证和签发证书功能，同时也兼容 X509 的由其他认证中心签发的证书。

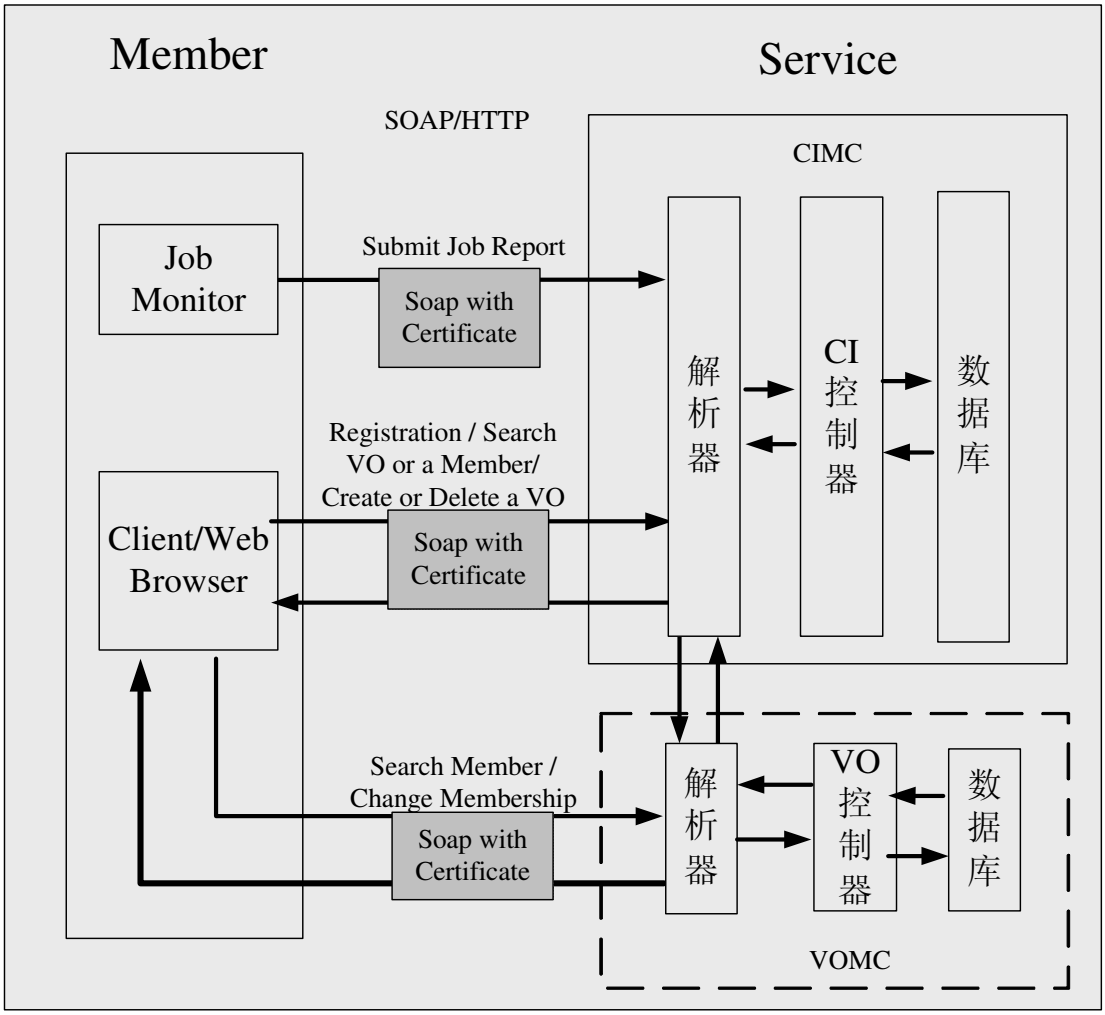


图 6.3 软件流程图

图 6.3 展示了系统架构图。实际架构和逻辑图 6.1 有一些不同。

首先是将虚拟组织的 VOMC 也托管到 CIMC 服务器上。虽然在逻辑上虚拟组织的管理是独立于赛百平台的管理，赛百平台仅仅作为信息备案和支持平台为虚拟组织提供必要的数据支持。但在实际开发中，考虑到以下因素：1. 在帮助普通用户动态建立专属虚拟组织的同时也尽量减少用户在维护虚拟组织方面的开销。2. 虚拟组织中心如果有个人进行管理就不能够保证实时的对虚拟组织进行管理，可能存在掉线，信息泄露或者丢失等问题。因此在实际实现中，将虚拟组织管理中心托管到了赛百服务器上进行管理和维护。

其次在成员端设计了 Job Monitor，对每次交互行为向 CIMC 提交报告。由于 DVOMS 仅仅提供认证和权限管理等信息维护服务，所以在实际应用中 DVOMS 还需要和其他工具，例如 Globus 或者 Condor 合作来为用户提供完整的资源共享服务。Job Monitor 的设计和应用与底层实现工具有着密切的关系，甚至在 Globus 和 Condor 工具包中本身就提供了这样的服务。因此在具体应用中，Job Monitor 可能会根据具体底层工具的不同而不同。

成员端：从图 6.3 中可以看到，在普通成员端运行的程序为 Client 或者 Web Browser，分别对应于客户端的命令行显示模式和浏览器显示模式。此外在成员端也运行有 Job Monitor 程序，这个程序通过接口调用底层任务调度工具，例如 Globus 或者 Condor，读取任务执行时间（Execution Duration），并将之报告给 CIMC。CIMC 将执行时间和预测执行时间比较之后更新成员特征向量，并存储与 CI 数据库当中。

VOMC：运行有 VO 控制器模块和数据库模块，具体功能参见 6.1 节第 4 段。

CIMC：运行有 CI 控制器和数据库，具体功能参见 6.1 节第 4 段。

6.2 认证中心及信息传输

在 DVOMS 中，安全认证和信息传输是应用层应用程序的基础。

6.2.1 CA 中心和签发证书

在 DVOMS 中采用认证中心签发证书的模式来表明成员身份和确保安全。以下展示了通过 4 步完成构建 CA 中心和签发证书的全部流程。

签发证书流程：

Step 1：构建 CA 中心。在 CA 中心输入 CA 中心的 C(Country), OU(Organization Unit), U(Unit)和 CN(Common Name)等信息，生成 CA 证书 cacert.pem 及其私钥 cakey.pem。其中 cakey.pem 为保密文件，由 CA 管理员进行管理，需要通过密码来访问

Step 2: 成员在客户端输入成员 C(Country), OU(Organization Unit), U(Unit)和 CN(Common Name)等信息，生成待签署证书 usercert_req.pem 文件和私钥文件 userkey.pem。

Step 3：成员将在本地生成的待签署证书 usercert_req.pem 文件通过邮箱等其他手段发送给 CA 中心，CA 中心通过 ccacert.pem 及 cakey.pem 对 usercert_req.pem 进行签署，签署后的成员证书为 usercert.pem。

Step 4：CA 中心将签署后的证书 usercert.pem 返回给成员，证书签发完成。

证书认证基于 PKI 中定义的认证流程，在这里就不在赘述。

DVOMS 定位于相对独立，配合其他使能赛百平台使能工具的赛百中间件，因此在默认情况下 DVOMS 提供了完整的建立 CA 中心和签发证书的全部工具和环境。DVOMS 的 CA 中心和认证系统基于 openssl 库，采用了被广泛使用的 X509 证书格式，任何已有的基于 X509 格式的证书系统均可以被识别和使用。这是考虑到大多数现有赛百工具或者网格工具都附带有安全认证系统，均基于 Openssl 工具和 X509 格式，DVOMS 可以在不改动原有证书系统的情况下部署到已有赛百环境和网格环境中去。

6.2.2 消息传输

在网络编程中，消息传输的方式有多种多样，常用的为直接建立 socket 来完成，如果考虑到安全性，可以采用 Secure Socket。基于以下两点本文决定采用 SOA 架构来设计消息传输的客户端和服务器端，采用加密

soap 协议进行传输。

1. SOA 架构为松耦合架构，对运行平台的依赖性也很小，这就保证了开发的消息传输模块有很好的扩展性和可移植性,例如支持跨操作系统传输，可以支持嵌入式系统等；而且采用 SOA 架构可以对客户端和服务器端进行解耦，单独开发，有利于未来程序的升级。

2. 通常 SOA 架构下的基于 soap 协议的消息是没有加密和认证的，这就有可能造成消息的泄露。为了弥补这一个确定，可以在 soap 信道建立之前先进行基于证书的认证，认证通过后对待传消息进行加密，确保消息的安全传输。因此本文通过已有的 SSL（Secure Socket Layer）进行 soap 消息传输，保证在 soap 信道建立之初进行认证，建立以后进行加密通信。Gsoap 工具包已经提供了相关接口，其函数如下, 具体参数含义可以参考[http://www.cs.fsu.edu/~engelen/soapdoc2.pdf]。

```
soap_ssl_server_context(&soap,SOAP_SSL_REQUIRE_CLIENT_AUTHENTICATION,
keyfile,password, cert, CA_Path, NULL (DH file, if NULL use RSA),NULL, NULL)

soap_ssl_client_context(&soap,SOAP_SSL_REQUIRE_SERVER_AUTHENTICATION|
SOAP_SSL_SKIP_HOST_CHECK, keyfile, password, cert, NULL, NULL)
```

DVOMS 构建了客户端和服务端程序来完成加密 soap 传输。

服务端开发

hdl.h 对服务器接口进行了如下定义。数据类 elop_soap 作为传输的基

```
class elop_soap
{
{public:
        struct soap *soap;
        std::vector < std::string > vec_err;
        std::vector < std::string > vec_string;
        elop_soap();
        ~elop_soap();
};
int ns__hdl(elop_soap *input_data, elop_soap *res_data);
```

本类，主要包括三项内容，soap 结构体，用来存储 soap 句柄；vec_string 为要传输的信息主体，如果作为输入则为命令和相关参数，如果作为输出则为执行结果信息；vec_err 为错误信息，一般用在输出，表示命令执行情况，例如执行错误，命令无法解析等。

hdl.h 为服务端头文件,服务端可以通过 Gsoap 工具包中 soapcpp2 命令，将定义接口的头文件转变为描述文件 WSDL（Web Services Description Language）文件[]，并通过 UDDI（Universal Description Discovery and Integration）进行发布。在 gsoap 工具包中命令如下

```
soapcpp2 hdl.h
```

生成 hdl.wsdl 文件，其内容如下所示：

```
<message name="hdl">
 <part name="input-data" type="ns:elop-soap"/>
</message>

<message name="elop-soap">
 <part name="vec-err" type="xsd:string"/>
 <part name="vec-string" type="xsd:string"/>
</message>

.....

<portType name="hdlPortType">
 <operation name="hdl">
  <documentation>Service definition of function ns__hdl</documentation>
  <input message="tns:hdl"/>
  <output message="tns:elop-soap"/>
 </operation>
</portType>
```

在 hdl.wsdl 文件中可以到，<message name="elop-soap">..... </message> 项对 hdl.h 中的 elop_soap 类做了描述，在 <output message="tns:elop-soap"/> ， <input message="tns:hdl"/> 和 <part name="input-data" type = "ns:elop-soap"/>中定义了 hdl 服务的输入和输出参数格式。

客户端

客户端可以直接从服务端获取头文件，也可以通过 UDDI 公布的服务描述文件 hdl.wsdl 中获得服务端接口定义。在采用第二种方法的情况下，

可以通过 Gsaop 工具包中的 wsdl2h 命令来自动将 WSDL 文件转化为头文件：转化后生成的头文件其主要内容如下：

```
struct ns1__hdlResponse
{
    char*                                   vec_err;
    char*                                   vec_string;
};
int ns1__hdl(
    ns1__elop_soap                          input_data,     ///< Request
parameter
    struct ns1__hdlResponse             *       ///< Response struct
parameter
);
```

从头文件中可以看到，同样定义了特定数据结构体 hdlResponse，其中 vec_err 和 vec_string 定义为字符指针，而不是前面 hdl.h 所显示的 vector<string>类型。这是因为在 Gsoap 开发中，在数据接收方会建立虚拟内存空间，而这个内存空间及其地址的管理是客户端和服务端共享的。在逻辑上数据发送端是通过 Gsoap 虚拟内存管理器将待发送消息写入特定虚拟内存空间，并将此地址告知数据接收端。数据接收端则通过此地址在虚拟内存空间中找到待接收消息。vector <string>实际传输的就是数据起始地址 char *. 因此客户端定义的参数为 char *类型。

6.3 内容解析

6.2 节基于 SSL 传输和 SOA 的架构设计保证了客户端和服务器端的松散耦合，传输的安全性。但是由于 DVOMS 涉及到了大量的数据库操作，特别是考虑到动态复杂的权限管理，所以不论是客户端还是服务器端都需要根据命令进行复杂的操作。在这个过程中就需要对传输层接受到的信息进行解析和评价。由于本文对 DVOMS 的定位为赛百中间件，提供用户权限的信息管理，为了真正让用户能够利用分布式资源，还需要和其他组件例如 Globus，Condor 等工具的支持，因此保证 DVOMS 良好的可扩展性和其他组件的松耦合性就成了关键，特别是在远程命令解析方面。在本文采用了域名和关键字配合使用来对接受到的信息进行命令解析。

首先构建解析表，解析表的结构由域和关键字组成，其基本结构如下：

```
[域名1]
关键字个数=2
关键字1=
关键字2=
关键字1_default=Default1
关键字2_default=Default2
[域名2]
关键字个数=1
关键字1=
关键字1_default=Default3
```

域名定义了域内关键字的应用范围，其作用范围为自域名往下至解析表结束或者下一个域开始为止。一个解析表中可以出现多个域名，一个域内可以有多个关键字。

解析过程：解析层从传输层接收到不定长字符串向量。这种不定长字符串向量为 vector<string>类型，可以看做存储字符串的链表，向量中每一个元素为一个字符串，其长度不定；此外向量长度也不定长，可以根据需要动态添加，可以参考 6.2.2 节对传输层接口的定义。由于首字符串往往说明了当前操作的主要类型和目的，所以首先解读首字符串，确定域。假定接收到的不定长字符串首字符为“域名 1”，那么就采用域名 1 中预定义好的解析表。然后按照关键字出现的顺序依次将向量中的元素赋值给特定关键字，如果向量中某一个元素为空，那么就将对应的关键字默认值赋予对应关键字。解析层在解析过程中对传输层无特定含义的字符串向量根据预定义赋予特定的实际含义，向上提交给应用层应用程序使用。

封装过程：解析层从应用层程序（可能是多个）接受消息，并根据应用程序的不同添加不同的域名，并按照解析表中关键字顺序对消息进行封装。主要功能为：1. 根据特定应用程序添加域名，应用程序的不同其采用的域名和关键字含义也有所不同；2. 根据关键字顺序和数量对待传输消息进行检查。

采用解析表对待传输消息进行解析和封装解除了数据与程序的耦合问题，同时有着很好的可扩展性。

在扩展性方面，由于解析表在实际应用中可以根据需求动态添加和修

改，因此当现有应用程序发生改变或者需要为应用层添加新的应用程序的时候，可以根据需求修改解析表中特定域的域名和关键字或者添加新的域和域名，实现应用层良好的可扩展性。

实现应用程序和数据的松散耦和：在不同程序间的数据通信中，特别是数据库操作密集型服务，接口格式和数据表格的改变往往牵一发而动全身，其数据格式紧紧耦合与特定的数据格式。应用中常常需要对某一个数据库表进行改动，例如增减属性，修改属性名称等，这种改动对于已有应用程序功能的实现并没有影响，但是由于在程序一般都硬性的将带访问数据表的属性名写入程序，这就需要修改应用程序源码，而在很多时候是不可行或者是效率很低的。这种程序与数据格式的紧耦合造成了程序僵化，修改困难等问题。而解析表的使用使得数据格式和应用程序之间不在直接发生关系，实现了松散的耦合。DVOMS 就采用了通过解析表说明数据库特定表格的方式，实现了数据库表结构与应用程序的解耦。

```
[DB_Member]
size=6
item0=user
item1=VO
item2=status
item3=Membership
item4=Reputation
item5=SuccIntr
item6=FailIntr

item0_default=Default0
item1_default=Default1
item2_default=Default2
item3_default=Default3
item4_default=Default4
item5_default=Default5
item6_default=Default6
```

上面的解析表存储在 CI 服务器中，说明 CI 成员表在数据库中的各项属性。在这个解析表中 [DB_Member]说明这个域为数据库成员表域，size=6 说明在成员表中一共定义了 6 项属性，第一项属性 item0=user 说明成员表中记录的第一项属性为 user 属性，表中其他属性分别为 item1=VO，item2=status 等，此外在域中还定义了默认值。应用程序通过读取域

[DB_Member]关键字来获知当前成员表中属性数量及属性名称。因此当数据库表发生改变，用户或者开发人员可以直接修改解析表而尽可能避免对应用程序的修改。

6.4 应用层：虚拟组织管理

应用层对解析层传输的格式化的消息进行判断，调用相关操作进行操作，并返回执行结果信息。考虑到可扩展性，应用层在 VOMC 和 CIMC 采用了 Daemon+可执行程序的模式，客户端由于不需要数据库，操作较少，所以主要运行 UI 来与 CIMC 和 VOMC 进行交互。

Daemon 模式

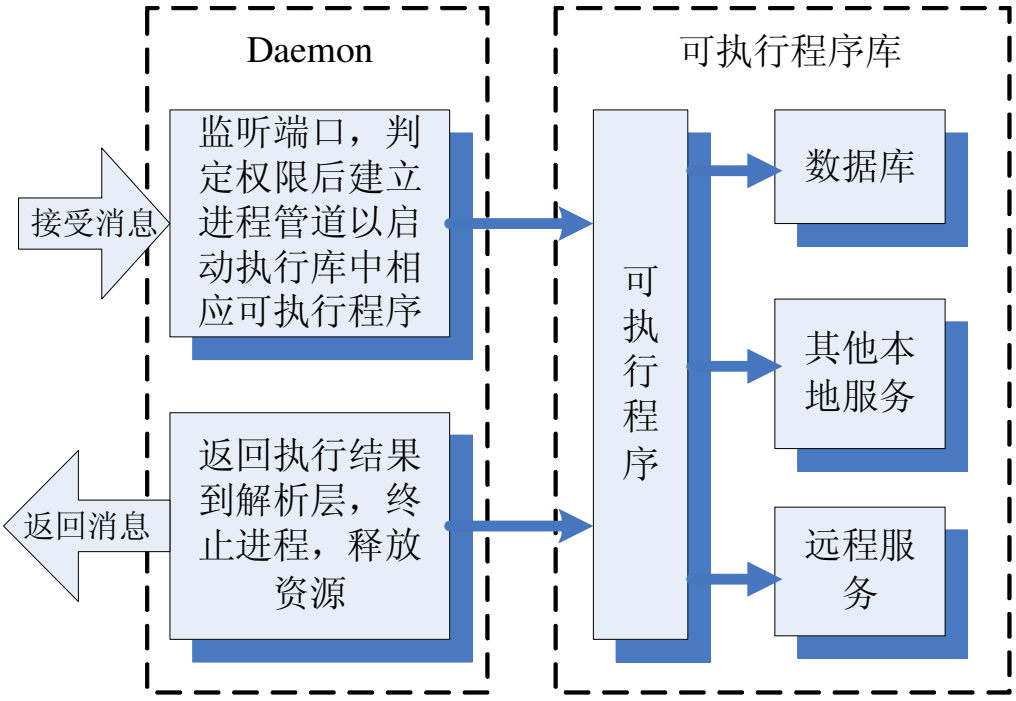


图 6.4 Daemon 模型

Daemon 为守护程序，分为两个主要功能

1. 监听从解析层上传消息，判定权限，如果远程用户拥有此权限则搜索本地可执行库中对应程序，并通过管道将消息作为参数注入此可执行程序。

2. 监听管道，读取可执行程序返回结果和错误信息，并将此信息传输给解析层。

多进程管理方法：

Daemon 作为服务端需要同时处理多个客户端请求，所以如何进行并行任务处理和进程管理是设计 Deamon 时需要解决的重要问题。在 DVOMS 中主要采用两个步骤来进行处理。

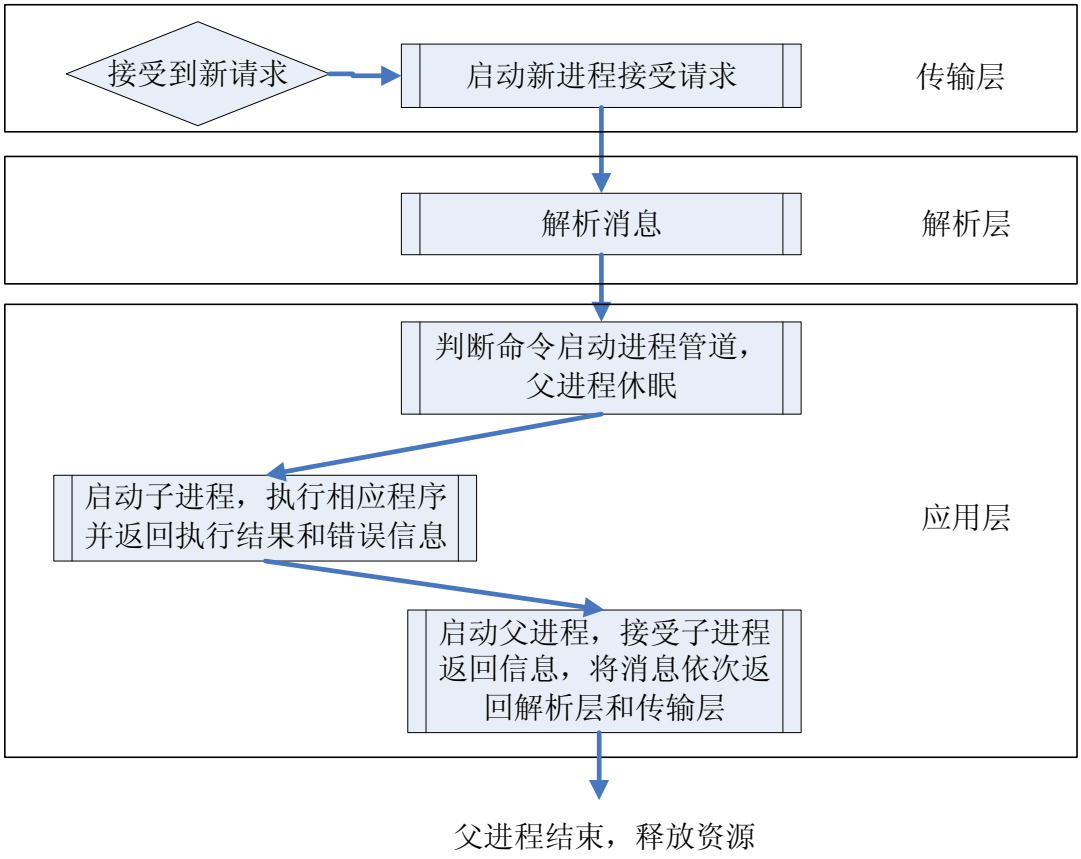


图 6.5 DVOMS 多进程管理

1. 传输层监听来自客户端 soap 消息，当有新的请求到来，则启动新进程负责处理和传输此消息。这个进程在实际运行中始终贯穿传输层，解析层和应用层。

2. 当上文提到的进程到达应用层后，应用层根据解析后的消息内容，启动管道进行处理。其模式为用子进程替代父进程，效果为父进程暂时终端，启动新进程并等待此进程运行结果，子进程结束后将执行结果和错误

信息返回父进程，父进程启动。如图 6.5 所示。

6.5 显示层

DVOMS 为用户了提供两种界面，分别为命令行模式界面和图形界面，分别对应不同的用户偏好和应用场合。命令行模式下响应快速，操作自由，安装方便而且消耗资源很小。图形界面比较直观，更容易操作。下面分别采用两种模式演示几种典型的虚拟组织操作。

图形界面模式：

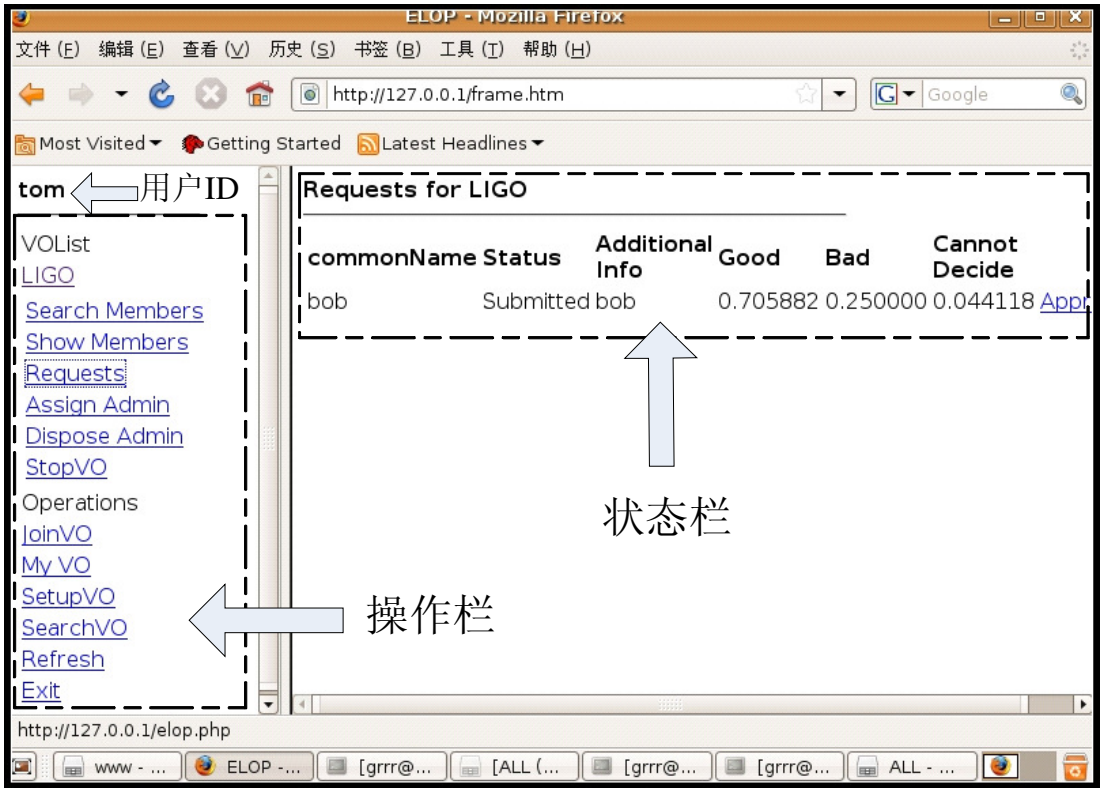


图 6.6 图形操作界面—VO 管理员操作界面

图形界面采用基于浏览器模式以方便移植和安装。图形界面从功能上分为两个部分，如图 6.6。左边为操作栏，右边为状态栏。点击左边操作栏进行需要的操作，右边状态栏会显示操作结果和需要输入的参数等。

命令行模式

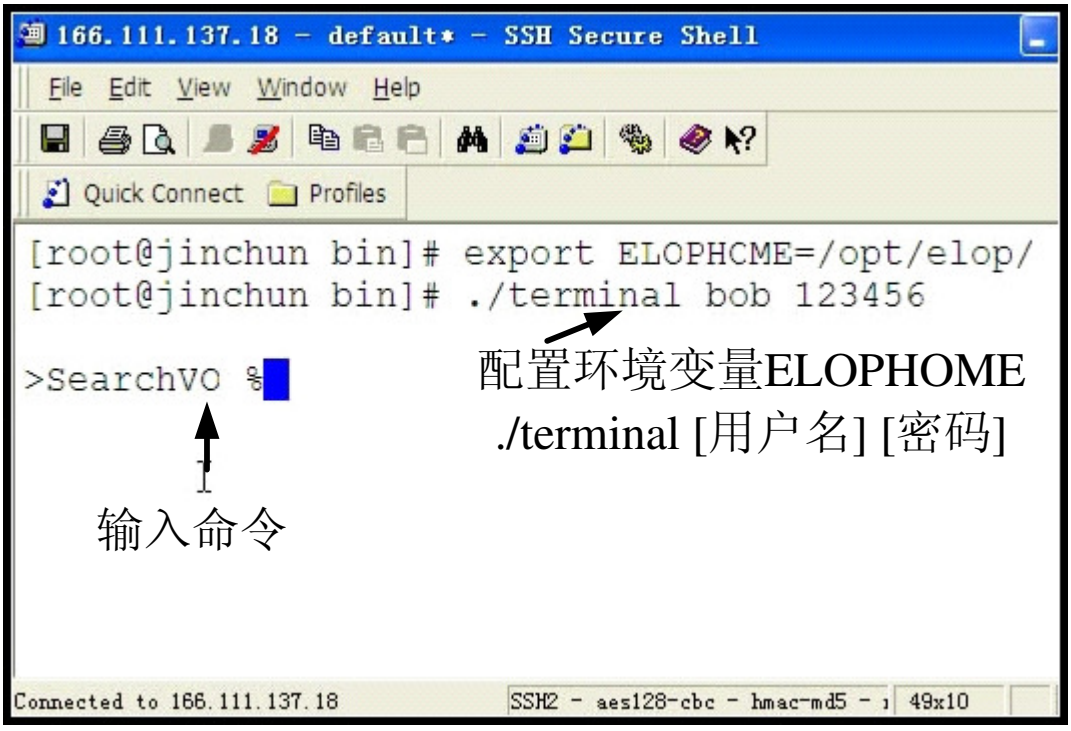


图 6.7 DVOMS 命令行输入模式

命令行模式下，通过./terminal 启动命令行控制台，此控制台不断读取用户输入的指令，通过传输层和解析层与远程服务器进行交互。同时虚拟组织管理员或者赛百平台管理员也可以通过此控制台查看和管理服务器及数据库。表 6.1 显示了在命令行模式下用户常用命令。

表6.1 常用命令：

命名	功能
Approve $VONAME $MEMBER	获得 VO 列表
Exit	显示$VONAME 成员列表
ForceStopVO $VONAME	加入一个 VO $VONAME
GetVOList	创建一个 VO
JoinVO $VONAME	查找 VO
Kick $VONAME $MEMBER	列出自己加入/申请加入了哪些 VO
MyRequest	查找$VONAME 中的成员，查找的部分为证书 subject，$Condition 为 SQL 查找条件。%为通配符

QuitVO $VONAME	退出$VONAME 这个 VO
SearchMember $VONAME $Condition	把$VONAME 中的$MEMBER 成员踢出 VO
SearchVO $Condition	批准$VONAME 中$MEMBER 成员的加入申请
Set +(-)Admin $VONAME $MEMBER	查看所有对$VONAME 的加入申请
SetupVO	将$VONAME 中的$MEMBER 成员提升+Admin/取消-Admin 管理员权限
ShowMember $VONAME	终止$VONAME 这个 VO
ShowRequest $VONAME	退出 terminal
StopVO $VONAME	强制终止$VONAME 这个 VO（CI 管理员命令）

第 7 章 DVOMS 使用及与 Globus 协同应用实例

DVOMS 有两种操作模式，作为赛百平台中间件也可以配合其他工具使用。7.1 节描述了 DVOMS 的图形界面使用方法，7.2 节演示了在命令行模式下 DVOMS 与 Globus 协同工作的一个实例，不但可以说明 DVOMS 命令行模式的使用方法，更从实用角度说明 DVOMS 在科学计算中的应用。

7.1 DVOMS使用

在虚拟组织管理中，比较典型的组织管理应用可以分为 1. 查找虚拟组织；2. 建立虚拟组织；3. 加入/退出虚拟组织；4. CMESM 计算推荐值（评价向量）帮助虚拟组织管理员审核申请人；5. 踢出不安全的虚拟组织成员；6. 在实际应用结束后终止虚拟组织。

（一）查询 VO

图 7.1 至 7.4 显示了成员 bob 向赛百服务器查询当前平台运行 VO 信息。有两种搜索方式，一种是查询所有虚拟组织，如图 7.1 和 7.2 所示；一种为条件查询，支持通配符“%”查询，如 7.3 和 7.4 所示。在 DVOMS 中采用%为通配符，可以代表任何长度字符。%VO 表示查询当前赛百平台中名称以“VO”结尾的所有虚拟组织。

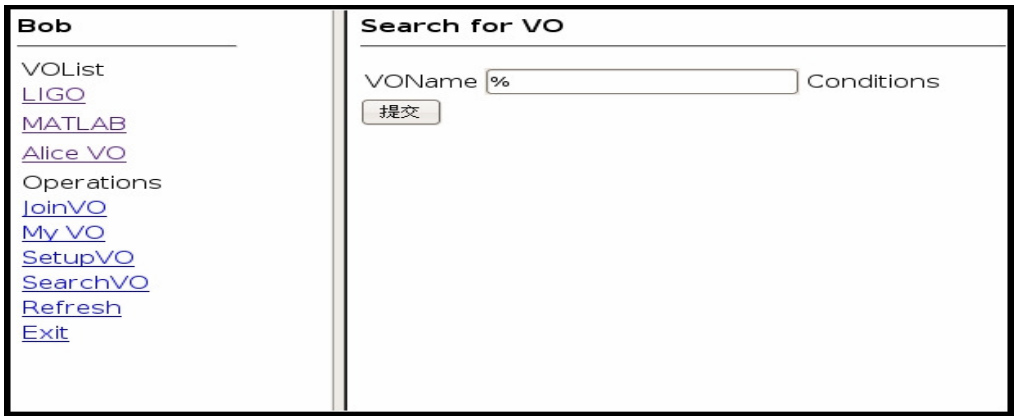


图 7.1 查找所有虚拟组织

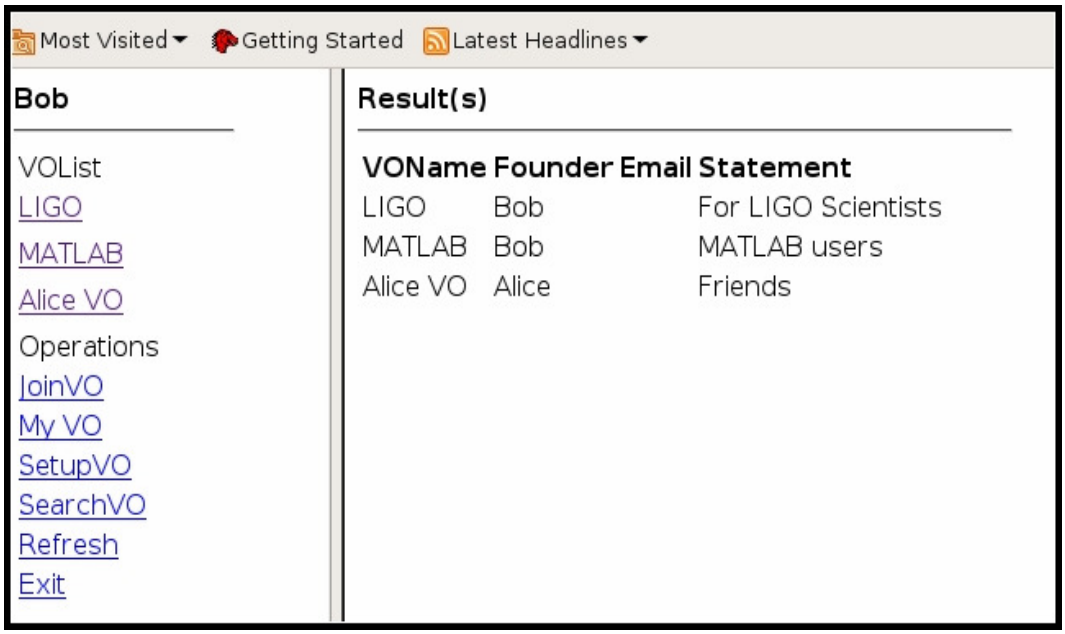


图 7.2 返回当前赛百平台所有虚拟组织信息

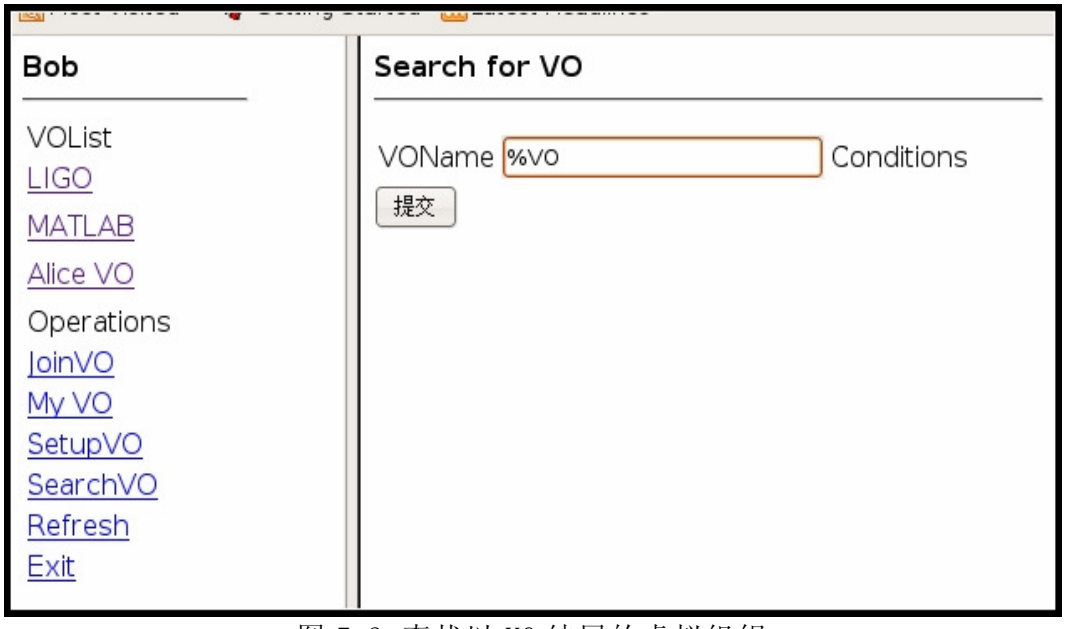
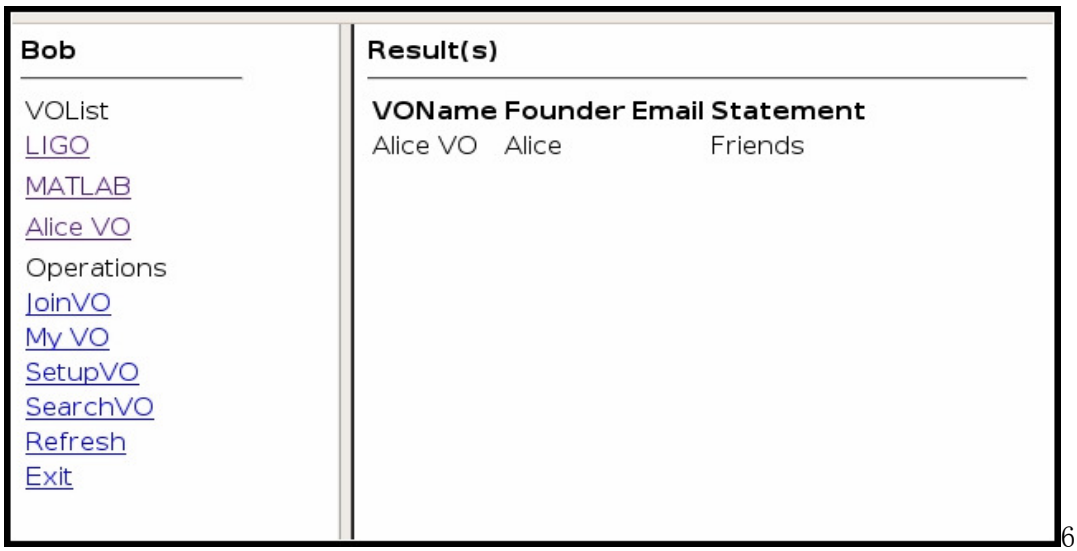


图 7.3 查找以 VO 结尾的虚拟组织



6.

图 7.4 返回 AliceVO 信息

- **建立虚拟组织**

作为示例，假定赛百平台中有 tom，alicce 和 bob 三个成员，其中 tom 构建了名为 LIGO 的虚拟组织。图 7.5 显示了 tom 如何构建 LIGO 虚拟组织，图 7.6 通过查找证明 LIGO 虚拟组织构建成功。

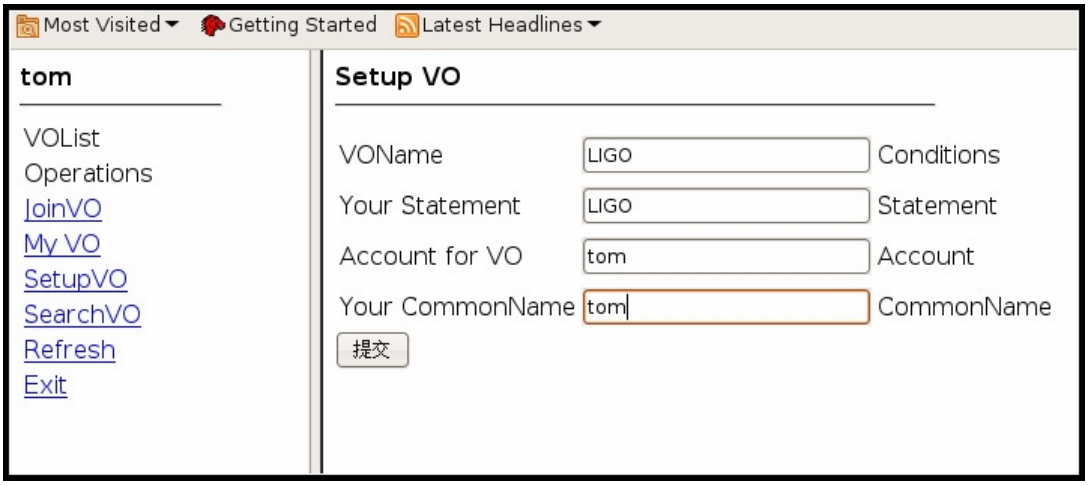


图 7.5 tom 构建 LIGO 虚拟组织图

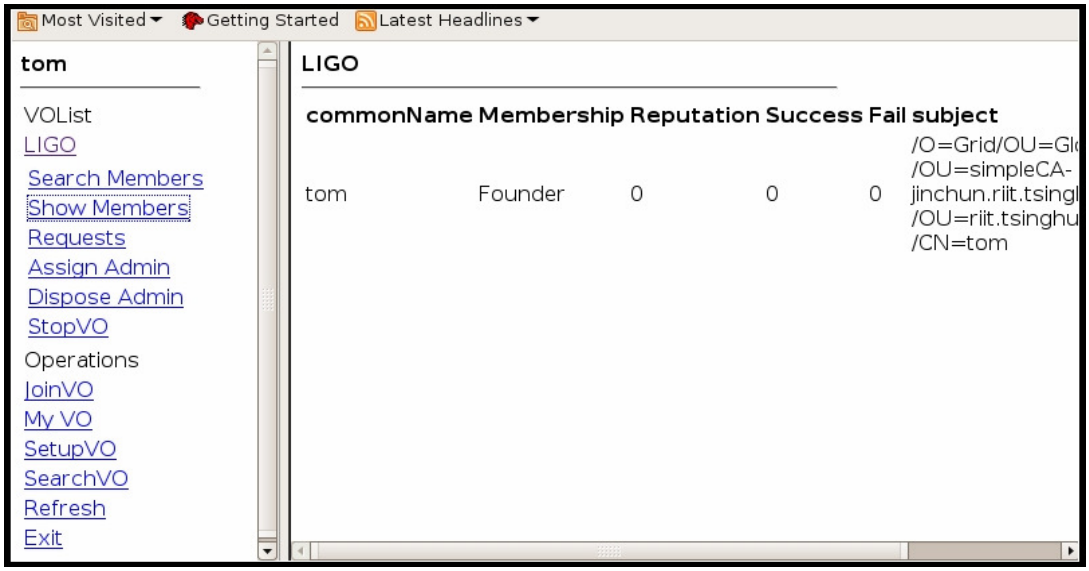


图 7.6 LIGO 虚拟组织创建成功

- **加入虚拟组织**

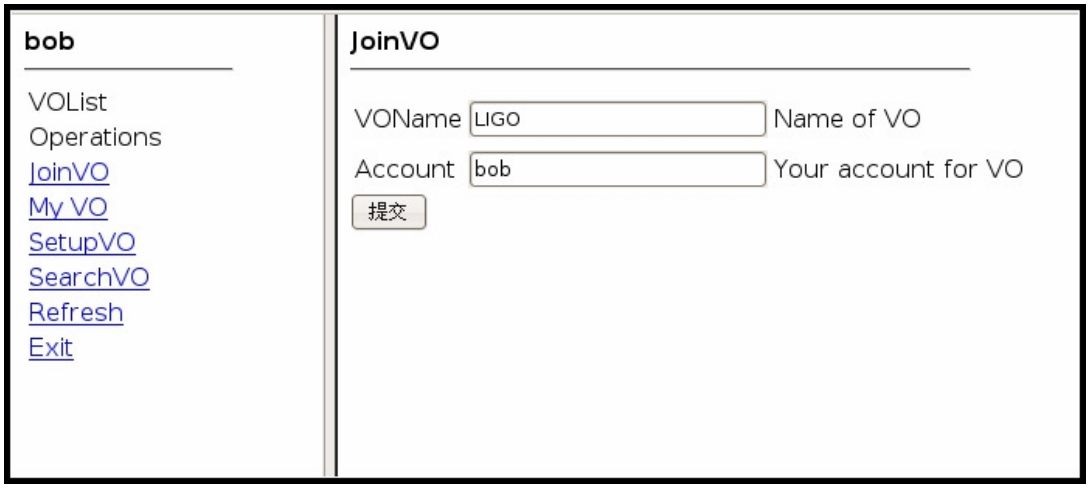


图 7.7 bob 申请加入 LIGO 虚拟组织

图 7.7 至 7.9 显示了申请人 bob 查找到感兴趣的虚拟组织 LIGO 后是如何最终加入此虚拟组织的。bob 向 LIGO 提交申请后，在其个人虚拟组织管理状态中显式 LIGO 为提交状态。VOMC 的数据库维护 bob 申请信息，

并在 LIGO 管理员 tom 再次登陆后将此信息通知给他。LIGO 管理员 tom 的状态栏中主动显示了 bob 的申请。CMESM 通过计算得出针对 bob 的评价向量最为参考值以供 tom 参考，如图 7.8 所示。tom 最终决定批准 bob 加入到本虚拟组织当中，DVOMS 会将这一消息在 LIGO 组织内进行广播。此时 tom 再次查看 My VO 状态，会发现当前 LIGO 成员为两名。

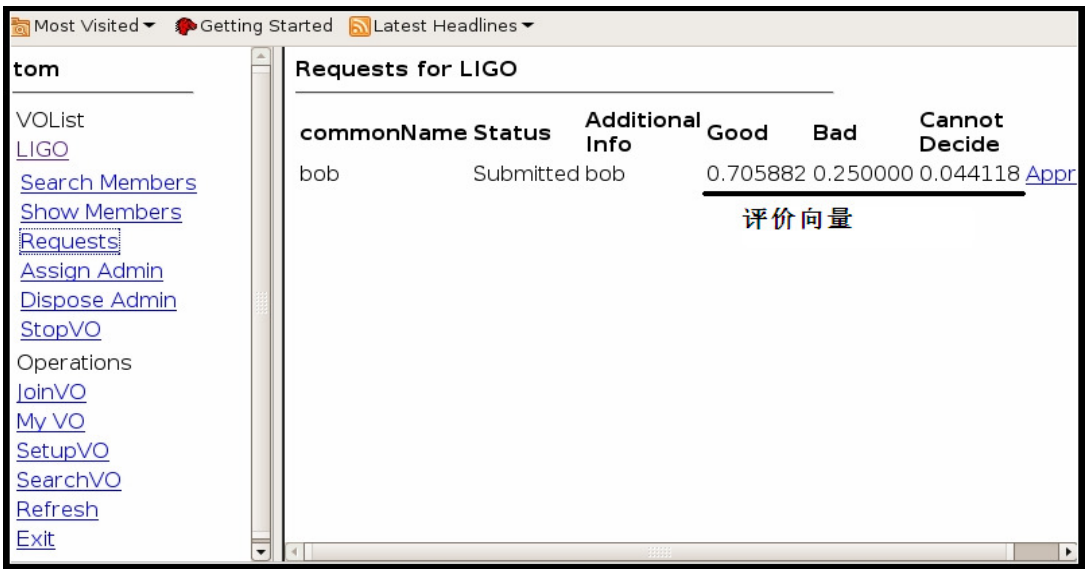


图 7.8 tom 对 bob 进行审核

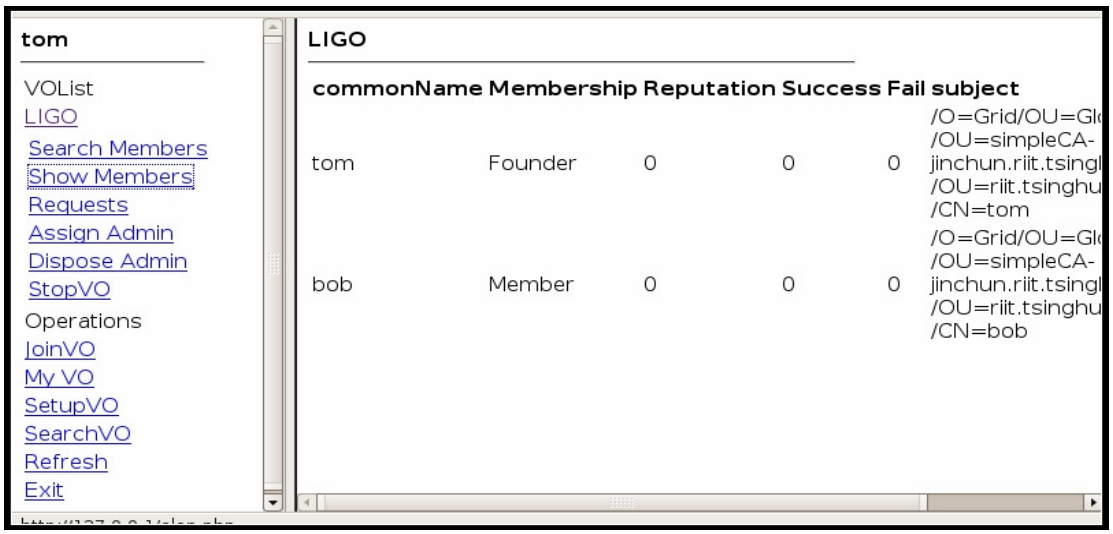


图 7.9 批准 bob 加入后 LIGO 中有两名成员

- **修改虚拟组织关系**

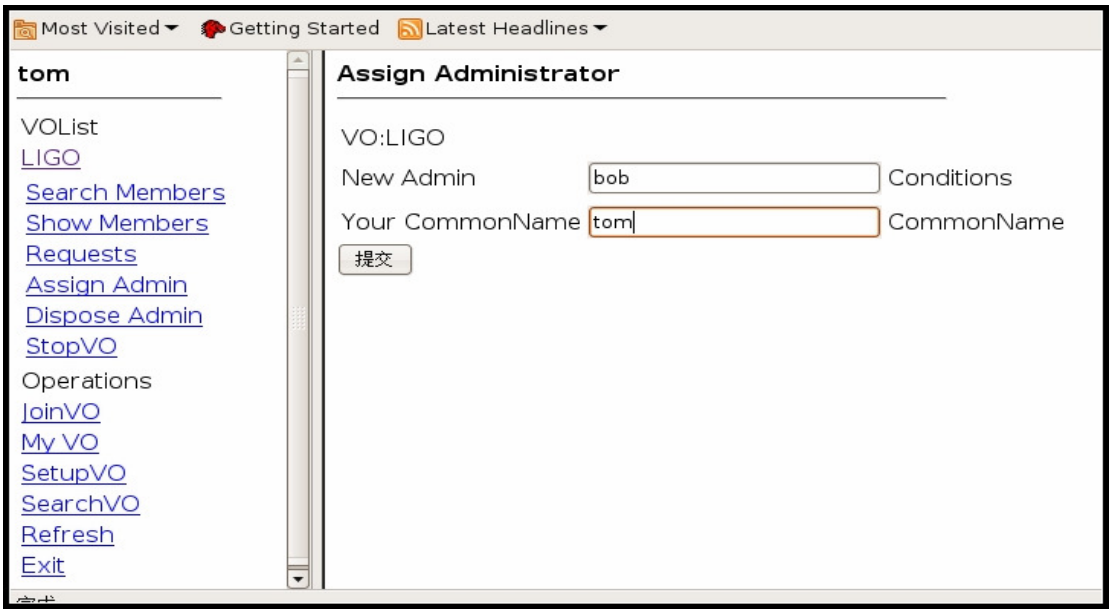


图 7.10 tom 提升 bob 为管理员

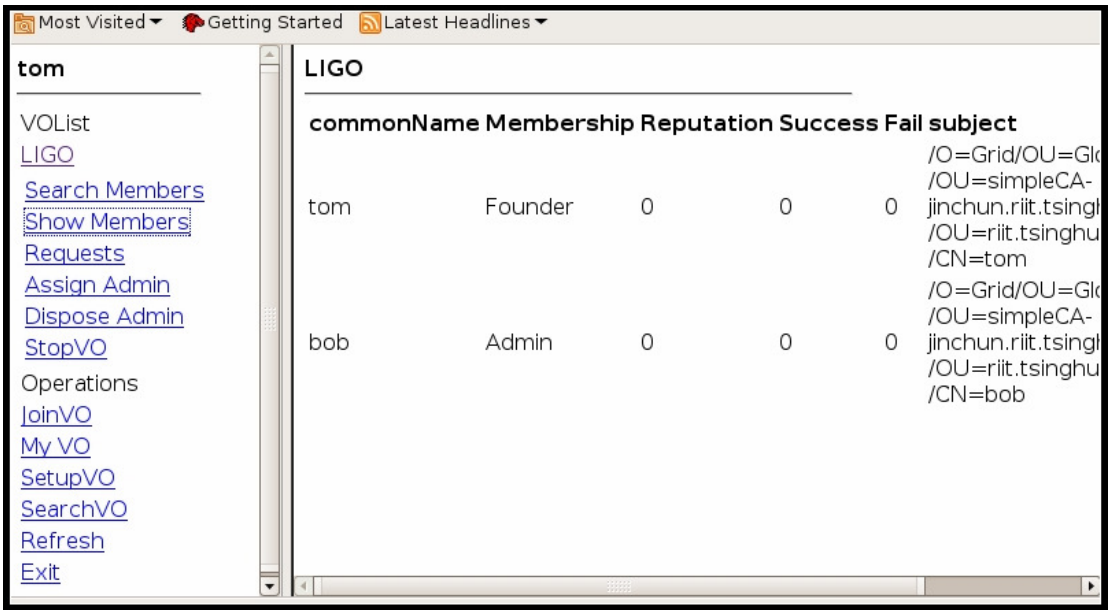


图 7.11 成员在 LIGO 中的角色

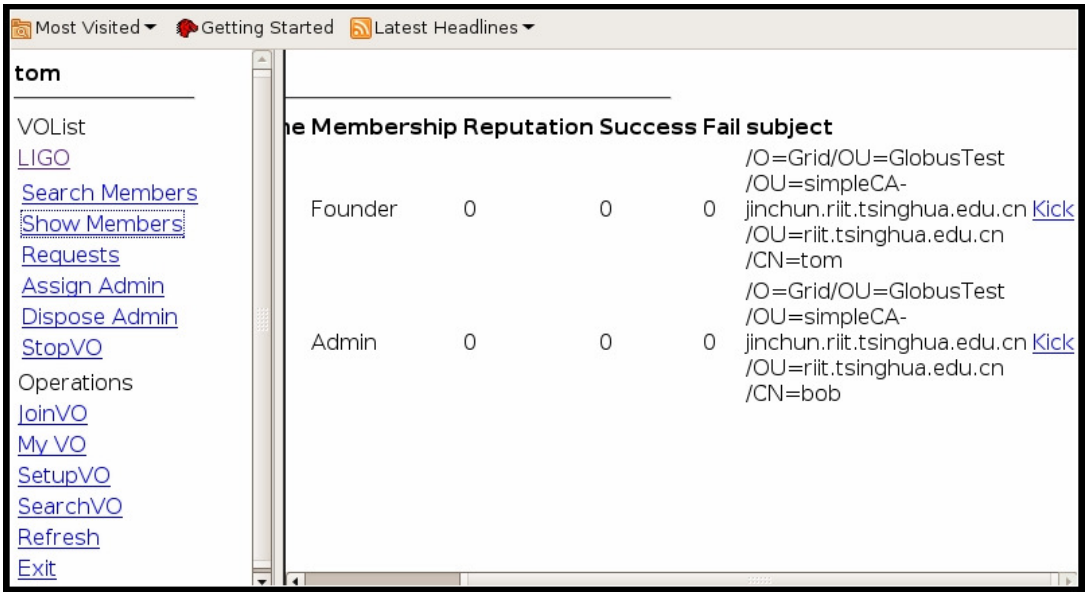


图 7.12 从 LIGO 中踢出 bob

修改虚拟组织关系主要包括提升/降低某一成员权限，终止成员的虚拟组织关系（即将某个成员提出虚拟组织）。图 7.10 和 7.11 显示了 tom 将 bob 提升为虚拟组织的管理员，使其拥有批准加入或者踢出成员的权限。这是考虑到当有频繁虚拟组织变动情况下，一方面虚拟组织的创建人 tom 需要承担很大的工作量；另一方面对成员来说也很难有及时的响应。因此为了分担工作和提高响应速度，虚拟组织创建人可以将部分虚拟组织管理权限赋予特定用户，即赋予 admin 的角色和权限。

此外，如果在虚拟组织运转过程中发现某些成员的行为不符合虚拟组织的预设目标和期望或者对其他成员造成了危害，虚拟组织管理员也有权限将此成员清理出虚拟组织。图 7.12 显示了 tom 将 bob 踢出的情况。

- **终止 VO**

当 LIGO 项目结束时，LIGO 虚拟组织中的所有成员和资源都应该释放出来，以供其他虚拟组织使用。在这种情况下就需要结束 LIGO 虚拟组织。在应用中，只有 LIGO 创建者拥有取消虚拟组织的权限。图 7.13 显示了 tom 取消 LIGO 虚拟组织的操作。

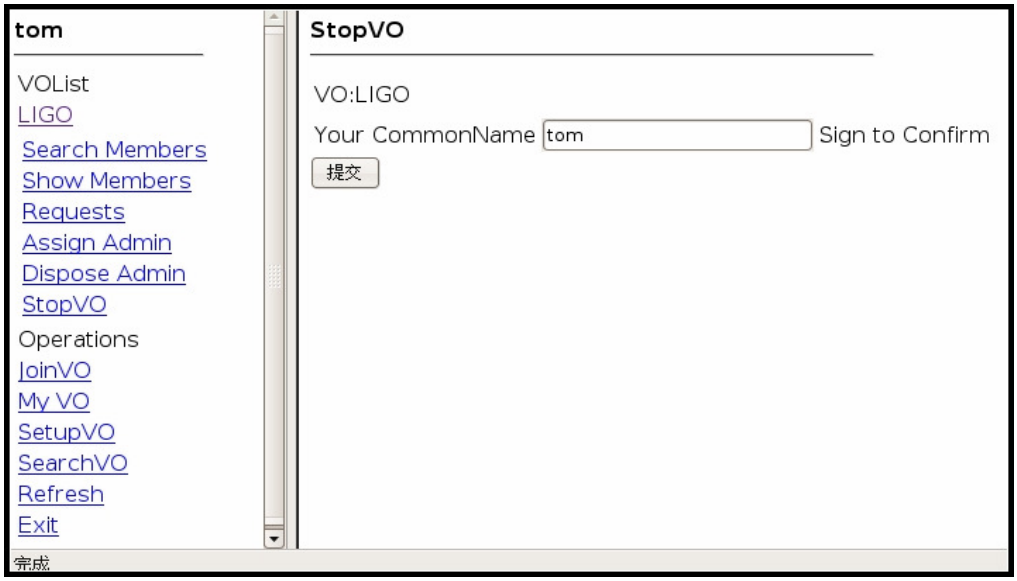


图 7.13 终止 LIGO 虚拟组织

7.2 命令行模式下DVOMS与Globus的协同应用实例

DVOMS 为赛百平台中间件，主要为用户提供权限管理信息。因此在实际应用过程中需要第三方分布式共享使能软件来实现共享，这也弥补了当前第三方分布式共享软件，例如 Globus 和 Condor 缺乏灵活有效的大规模信任机制的缺憾。

Globus 是由美国 USC 大学开发的开源网格使能软件。就像本文在第 1 章所讲到，赛百平台是众多网格的集成，打破了网格之间的屏障，实现不同网格资源的重新调配和安全访问。一般来说，网格成员在加入网格之前已经达成了共享协议和共识，各类网格使能工具例如 Condor，Globus Toolkit 在技术上将这种现实世界达成的共享协议在网络环境中加以实现。换句话说，Globus Toolkit 完成了资源共享使能，即如何共享的问题，但没有解决是否应该共享的问题。相比较而已，DVOMS 解决了是否应该共享的问题却并没有实现如何共享。因此 DVOMS 与 Globus Toolkit 的结合可以为用户提供完整的赛百应用服务流程。

Globus Toolkit 采用证书认证的方式来证明网格中各成员的身份，通过 grid-mapfile 或者第三方中间件 GUMS 来帮助用户指定来访成员在本地的权限管理，核心思路在于将远程来访者映射到本地的特定账户，使之拥有

本地账户相应的权限。grid-mapfile 通过文本文件显式的说明映射关系，GUMS 通过动态调整 grid-mapfile 内容，帮助用户制定比较复杂的映射测量。本例中采用调整 grid-mapfile 的模式来完成赛百平台权限管理。下面是一个典型 grid-mapfile 例子。

```
    /O=Grid/OU=GlobusTest/OU=simpleCA-jinchun.riit.tsinghua.edu.c
OU=riit.tsinghua.edu.cn/CN=bob    alice
    /O=Grid/OU=GlobusTest/OU=simpleCA-jinchun.riit.tsinghua.edu.c
OU=riit.tsinghua.edu.cn/CN=tom    john
```

grid-mapfile 中每一行为一条完整映射关系，它的前半部分为从远程用户证书中提取的 subject 内容，完整说明了此用户信息。其中 CN 是 Common Name 的缩写，为此远程用户名称。后半部分为此远程账户在本地的映射账户。第一行表示如果来访者是 bob，那么就将本地的 alice 账户赋予来访者，远程用户可以以本地 alice 账户访问本地资源。

在 Globus toolkit 使用过程中，grid-mapfile 是网格成员实现资源共享的关键步骤。不过传统网格环境中，grid-mapfile 的维护是由人工来完成。例如当网格环境中有新成员加入或者有成员退出，那么网格管理员会通过邮件，电话等方式通知所有网格成员，告知其需要添加或者删除 grid-mapfile 中的记录。此外，grid-mapfile 的修改需要用户拥有计算机的 root 权限，而在实际使用中，大型计算机，电子天文台等资源中心的用户拥有的仅仅是普通用户权限，只有特定人员，例如设备管理员才有权限进行 grid-mapfile 的修改和维护，而这会造成相当大的不方便。因此在网格规模较大，组织成员变化频繁的情况下网格成员对 grid-mapfile 的维护往往是不能及时响应的。如果没有及时添加新的成员，那么就会造成新成员不能够快速的共享资源，造成时间和资源的浪费；如果没有及时删除旧成员，那么就会造成非网格成员还可以继续访问资源的情况，造成安全隐患。此外，由于大型计算中心或者数据中心（例如电子天文天，电子图书馆或者传感器等）一般作为基础设施支撑众多不同学科的科学应用，因此他们往往加入了不同的网格。在这种情况下，依靠人工来维护 grid-mapfile 就变得拖沓，低效而且十分的不安全。

针对这样的问题，在实际应用中可以通过 DVOMS+Globus Toolkit 作为资源共享环境的整体解决方案，以支持复杂，动态，安全的多项网格组织管理，也就是赛百平台中虚拟组织的管理。

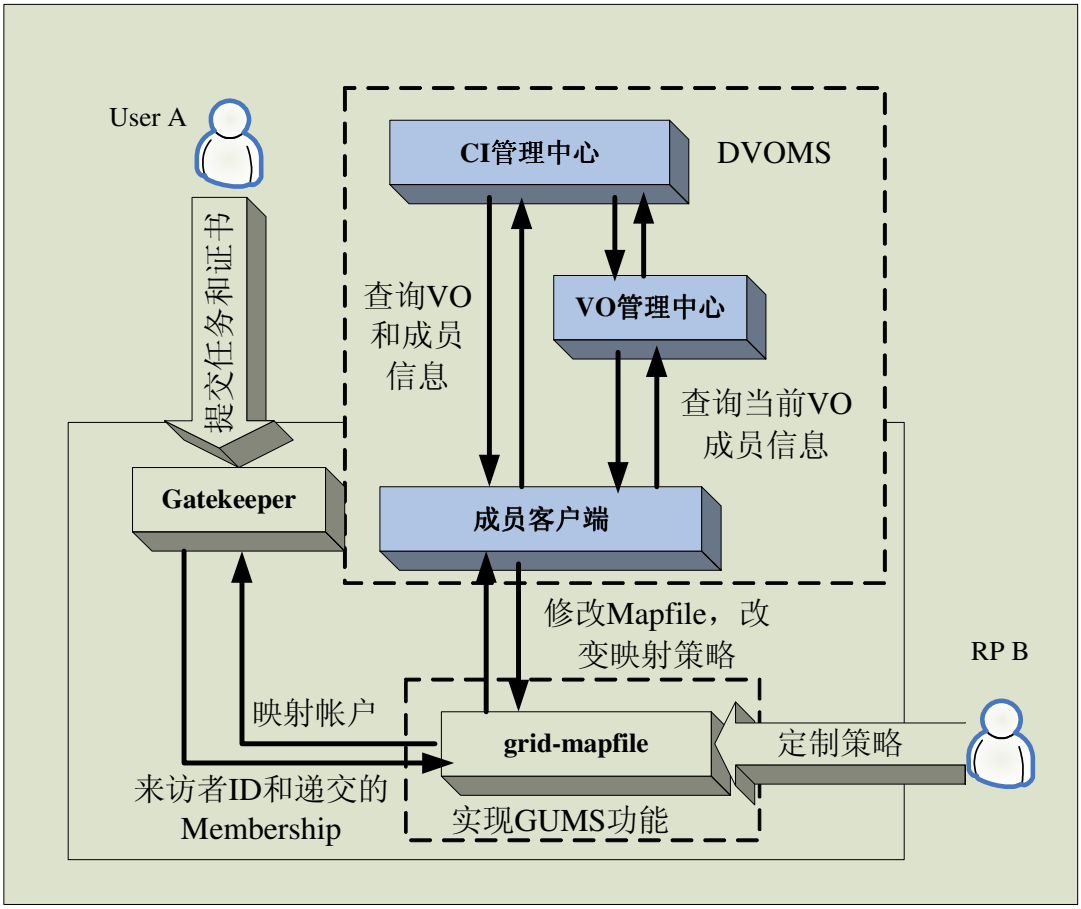


图 7.6 DVOMS 和 Globus Toolkit 协同应用结构图

图 7.6 显示了 DVOMS 和 Globus Toolkit 协同工作的流程。当 User A 向 RP B 提交任务和本人证书，本地运行的 Globus 会通过 Gatekeeper 验证证书是否合法和正确，然后提取来访者 subject 信息向 grid-mapfile 进行查询，根据映射关系获得本地账户。DVOMS 负责对 grid-mapfile 进行维护，当 B 所在虚拟组织成员发生改变，例如有新成员加入或者退出，DVOMS 动态的更新 grid-mapfile 映射信息，确保 B 的资源被安全使用。同样当 B 作为用户需要访问其他成员的时候，例如 A，在 A 处也需要完成类似流程。

在本例中，实验环境为三台安装了 Globus toolkit4.0.5 版本的计算机，

分别充当 VO 创建者，成员 A，成员 B。此外还有一台计算机作为赛百服务器，其具体信息如表 7.1 所示。

表7. 1 DVOMS+Globus Toolkit 中成员配置情况

域名（Host Name）	IP	证书 CN（Common Name）	本地账户(User Name)	角色
yushu. riit.tsinghua.edu.cn	166.111.137.17	CA	无	CI 服务器
jinchun. riit.tsinghua.edu.cn	166.111.137.18	bob	bob	VO 成员
hanqiu. riit.tsinghua.edu.cn	166.111.137.19	tom	tom	VO 创建人
ligo. riit.tsinghua.edu.cn	166.111.137.170	alice	alice	VO 成员

全部实验室在清华大学信息技术研究院完成，研究院域名为 riit.tsinghua.edu.cn，为每一台计算机分别分配子域名，如表 7.1 域名项所示。为了简单起见，在下文中仅仅用子域名代表相应计算机，例如 jinchun.riit.tsinghua.edu.cn 计算机用 jinchun 代表。第二项为相应 IP 地址。Common Name 项填写的是成员证书上的 Common Name，代表用户采用何种 CN 访问远程资源；本地账户为远程用户访问本地资源时，grid-mapfile 映射在本地的账户 ID。例如在 jinchun 上有 ID 为 bob 的用户，其证书上 Common Name 也为 bob，当然也可以自定义证书 CN 为 john,smith 等。jinchun 的 bob 用户就可以通过证书访问其他计算机。在实验中为了简单起见，在申请用户证书的时候填写的 Common Name 与本地账户一致。表 7.1 中最后一项为角色，在实验中一共有三种角色：赛百服务器，虚拟组织创建人和普通虚拟组织成员。CI 服务器对赛百平台数据库和 VO 数据库进行管理和维护，不参与具体的资源共享活动，所以其 Common Name 为 CA（Certificate Authentication），没有本地账户。jinchun 和 ligo 为普通成员，hanqiu 为虚拟组织创建人。

实验过程中为了更好的实时对比各台计算机运行和响应情况，通过 ssh

远程控制这四台计算机，并将其结果一并显示出来。

- 启动客户端和赛百服务器

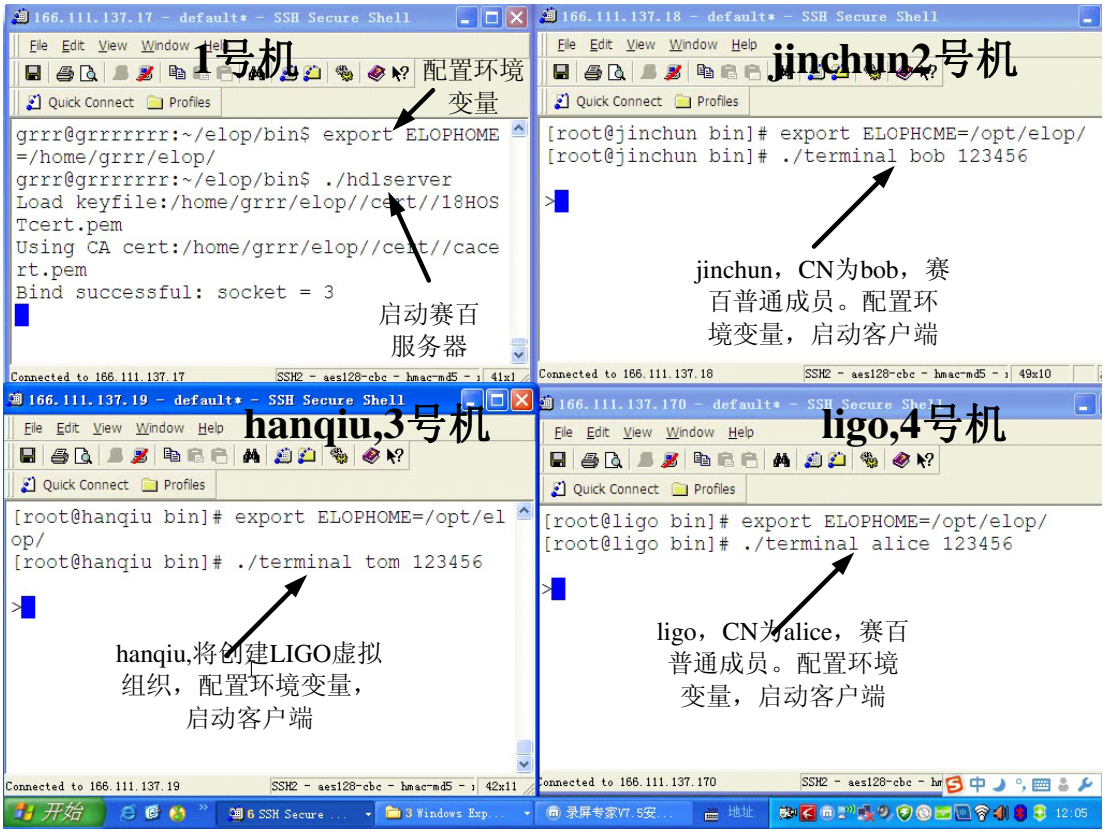


图 7.7 启动赛百服务器和命令行客户端

图 7.7 显示了如何启动赛百服务器和命令行客户端的情况。1 号机为赛百平台服务器，负责启动和维护服务器及数据库；2 号机和 4 号机的主机名分别为 jinchun 和 ligo，为普通赛百成员；3 号机主机名为 hanqiu，将会创建虚拟组织。从图中可以看到，不论是启动客户端还是服务器端，都首先需要设置环境变量 ELOPHOME=ELOP 代码存放地点。在客户端启动过程中，需要输入 Common Name 和密码。

- 创建 LIGO 虚拟组织，并且加入此组织

图 7.8 显示了创建和查询虚拟组织的过程。3 号机以 tom 的名义向赛百平台提出构建 LIGO 虚拟组织的申请。如图所示，tom 向赛百服务器提交虚拟组织名称，个人 Common Name 和申明等信息。其中申明主要是向其

他赛百成员说明此虚拟组织的主要功能或者目的是什么。提交申请并通过赛百服务器完成构建虚拟组织流程后，2 号机进行了当前虚拟组织查询，发现当前赛百平台中有名称为 LIGO 的虚拟组织。

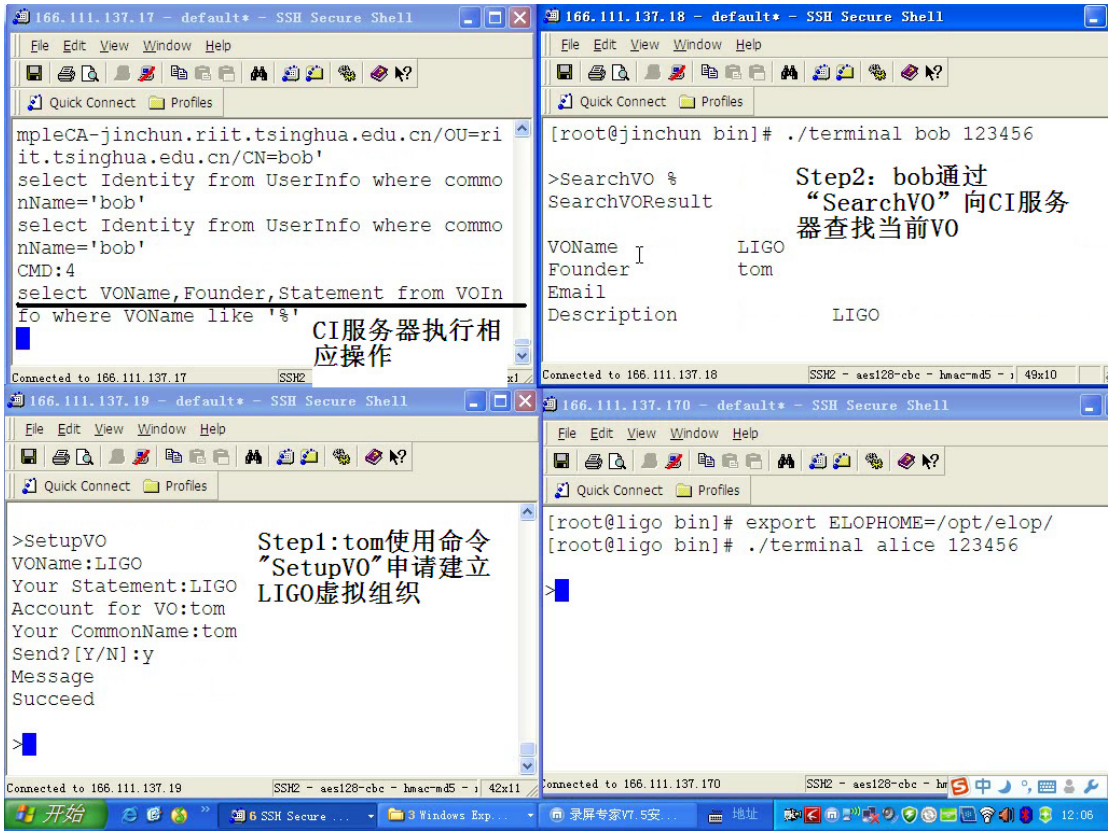


图 7.8 创建并且查询虚拟组织

在图 7.9 中，bob 和 alice 分别输入命令 JoinVO，向 tom 提出加入 LIGO 虚拟组织申请。tom 的客户端会提示 tom 当前有新的成员申请。此时 tom 输入命令 ShowRequest LIGO，可以看到客户端显示所有待审核成员的申请以及通过 CMESM 计算出来的评价向量。例如 bob 的评价向量为{Good=0.689189，Bad=0.121622，Cannot Decide=0.189189}。tom 参考了评价向量后最终决定批准 bob 加入 LIGO 虚拟组织。命令为 Approve [VO 名] [申请人]。 同理 tom 也批准了 alice 加入 LIGO。

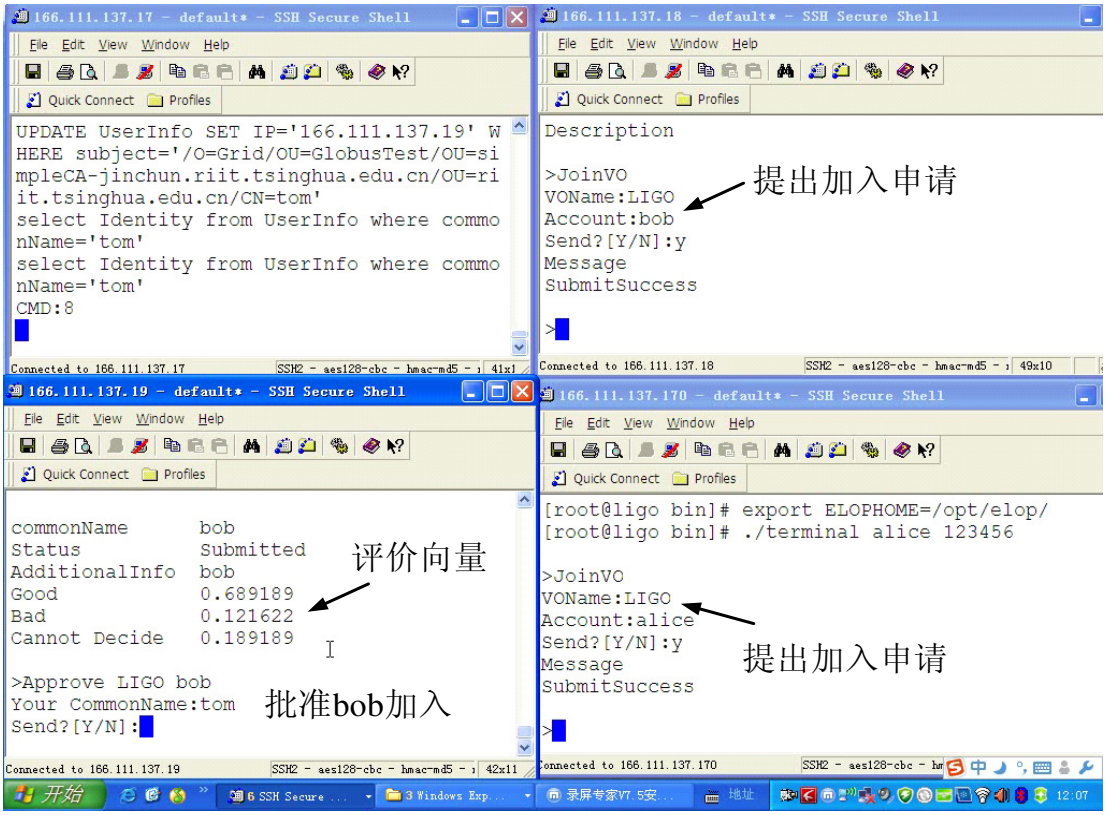


图 7.9 申请加入虚拟组织并批准

- 通过 Globus Toolkit 调用远程计算机上可执行程序

alice 计算机的的/bin/date 为系统可执行，其主要功能为获取本地系统时间，任何 ligo 主机上的账户均可以访问此程序。图 7.10 在未加入虚拟组织的情况下直接调用 globus-job-run 函数的执行结果。程序报错，原因为没有通过远程计算机的安全认证。成员通过命令行客户端加入虚拟组织以后，DVOMS 会动态修改 Globus Toolkit 的 grid-mapfile 文件，在虚拟组织的所有成员客户端添加最新成员信息。图 7.11 显示了加入虚拟组织以后的执行情况。从图中可以看到，bob 成功调用了 alice 计算机上的/bin/date 程序，并获得其执行结果。实际科学计算中可能是更加复杂的应用和深度共享，这些都可以由 Globus Toolkit 工具包来提供。

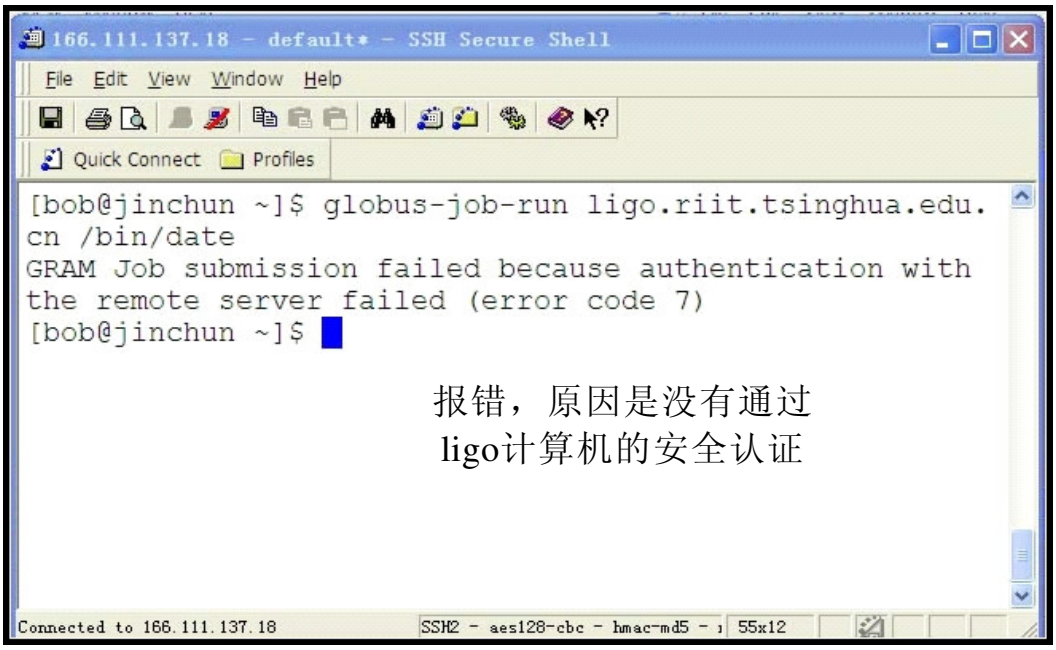


图 7.10 bob 未加入 LIGO 虚拟组织前调用 alice 本地程序

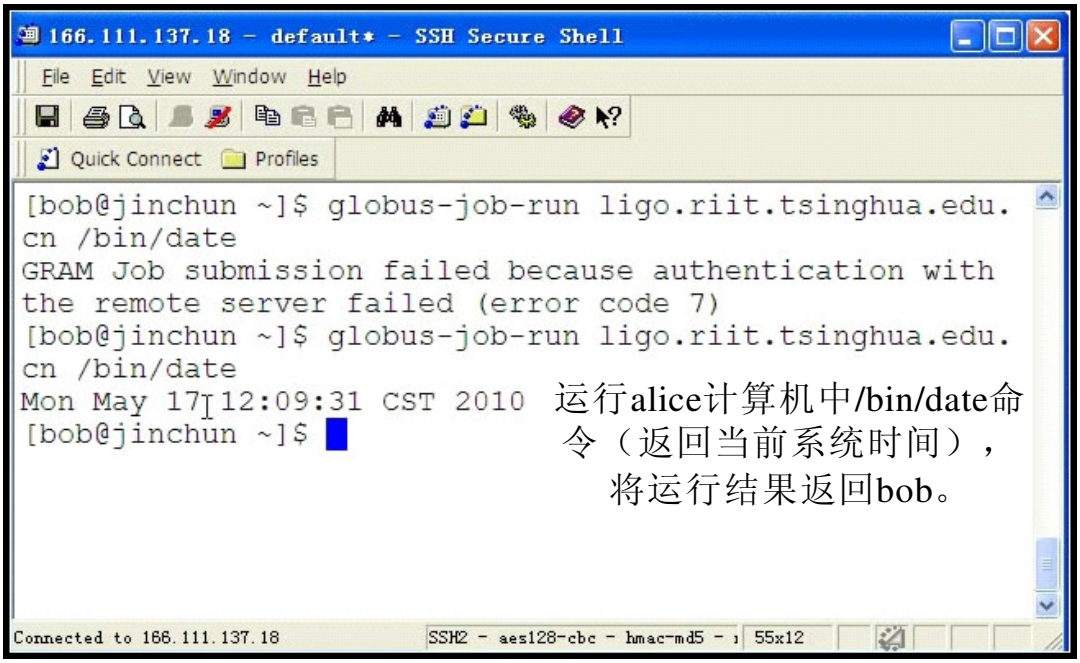


图 7.11 bob 加入虚拟组织后调用 alice 本地程序

- 取消虚拟组织

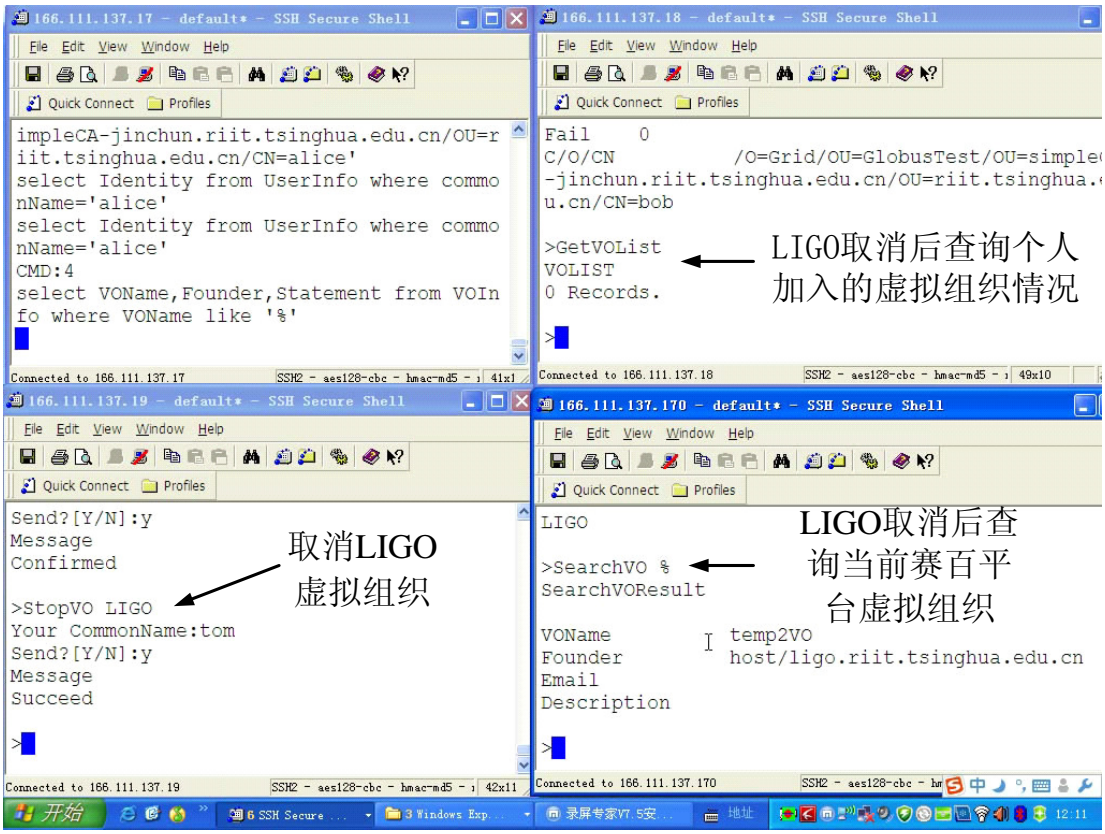


图 7.11 取消 LIGO 虚拟组织

虚拟组织取消只能由虚拟组织的创建者来完成。在本例中 tom 向赛百服务器提出取消 LIGO 虚拟组织申请，批准后赛百服务器会向所有 LIGO 成员进行广播，各个成员客户端根据内容删除所有相关 grid-mapfile 记录，确保原成员不再能够够访问本地资源。

本节演示了 DVOMS 在不同用户界面下的使用方法和使用效果。DVOMS 通过和网格中间件 Globus Toolkit 的配合使用，弥补了传统网格技术中存在的组织关系僵化，不便于管理等问题，为用户提供了动态，弹性和易操作的权限管理和资源共享整体解决方案。

第 8 章 总结

电力电网系统和以公路，铁路和航空为一体的交通物流系统是上个世纪人类最伟大的发明创造之一，他们是工业经济活动的基础支撑，所以也被称为基础架构。进入 21 世纪以后，随着科学研究的逐渐深入，现代科学研究越来越依赖于实验仪器，计算机和网络交互，很多重大科研项目例如黑洞合并、气候变化及基因测序都需要不同领域的研究人员通力合作，依赖分布在世界各地的计算机，实验仪器和网络系统进行数据共享和模拟计算。因此为科学应用构建计算基础架构 CyberInfrastructure(CI)以帮助科研人员动态创建虚拟组织，实时，按需的重新整合分布在不同地点和组织的各类资源就成了研究人员首要解决的问题。和传统网格环境相比，赛百平台容纳的资源不论是数量还是类型都更多，因此赛百成员（RP 或者 User）一般来说天然上是陌生的；此外由于是作为基础支撑平台，赛百平台支持的科研项目更加广泛，其资源组织和共享方式更加多样化和动态。如何帮助成员准确的评价其他成员以帮助其与陌生的目标成员建立适当，安全的信任关系和如何进行多级虚拟组织管理成为赛百平台面临的重要挑战。现有技术，例如 VOMS，GUMS 等都是网格技术框架下帮助用户尽可能实现细粒度的权限管理。从应用平台来说是针对某个网格这样小规模，成员单一的共享环境；从功能上来讲并没有帮助用户对陌生来访者进行信任度评价，无法应对成员天然陌生的情况，也没有针对多级虚拟组织提供管理工具。

针对这种情况，本文以国际科研合作项目 LIGO 为实际应用背景，重点研究了如何在赛百平台中构建动态，可信任，可扩展的多级虚拟组织这一问题，并提出了基于委员会的成员评价方法(CMESM)和联邦模式多级虚拟组织模型，分别解决在虚拟组织构建过程中出现的成员信任问题和可扩展性问题，并在此基础上开发了动态虚拟组织管理系统的赛百平台中间件。 CMESM 由代表和委员会两层组成，综合了虚拟组织内部成员意见，表现了虚拟组织对申请成员的个性化要求。第五章的仿真实验证明，CMESM 能很好的抵御共谋欺诈行为，并切实提高了虚拟组织内的服务质量。为了实现可扩展，安全的多级虚拟组织管理，本文参考现实社会中的组织结构，提出基于联邦模式的多级虚拟组织模

型，保证子虚拟组织在动态加入其它虚拟组织的过程中其内部安全性不会收到威胁。

其次针对 DVOMS 赛百平台中间件的应用特定，本文对其进行了详细的设计和开发，其四层结构设计，传输层，解析层，应用层及显示层和 SOA 架构设计保证了 DVOMS 具有很好的可移植性和可扩展性，确保未来便于进行二次开发和功能扩充（本文第 6 章）。最后作者详尽的演示了 DVOMS 的使用方法，并结合传统网格技术 Globus Toolkit，演示了赛百平台的一个实际使用案例，包括建立虚拟组织，加入虚拟组织和调用远程可执行文件等操作。

本文工作的创新点有：

1. 提出 CMESM 评价算法。CMESM 是委员会制度和模糊逻辑模式识别算法的结合，其特点在于即使在共谋欺诈成员数量较多的情况下赛百平台仍然可以保证评价的准确性和稳定性，这是一般信誉评价方法所不能提供的；其次 CMESM 即使在赛百平台初步使用的时候也能提供比较好的服务，而且其性能可以随着赛百平台的进一步使用而显著提高；最后站在用户和实际应用的角度，经过 CMESM 对资源进行筛选后，用户总任务完成时长（Makespan）有了显著的改善，提高了虚拟组织的服务质量。

2. 提出基于联邦模式的多级虚拟组织管理框架。联邦模式的多级虚拟组织管理，确保了虚拟组织在与其他虚拟组织合作的过程中（加入更高级虚拟组织）不会发生权限方面的泄露，确保成员和子虚拟组织的独立性和安全性。此外这种模式也具有很好的可扩展性，可以通过嵌套，满足任意层虚拟组织管理的需求。最后由于联邦模式的多级虚拟组织管理框架采用分层评价模式(HCMESM)，由各级虚拟组织委员会独立做出评价，最高层虚拟组织的委员会仅仅承担汇总和计算功能，实现了负载均衡，避免出现单点负载过大情况的出现。

3. 良好移植性的软件设计。在 DVOMS 的设计中采用了 SOA 架构和分层设计方法。SOA 架构保证了客户端和服务端的松散耦合，具有很好的可移植性。分层设计方法将程序按照功能和流程分为传输层，解析层，应用层和表现层，各层通过一定的协议进行交互，这样有利于 DVOMS 未来功能扩展和二次开发。此外在 SOA 架构和分层设计情况下，DVOMS 也可以很好的嵌入现有工具，通过配合使用完成更加复杂的功能，例如本文在

7.2 节演示的 DVOMS 与 Globus Toolkit 协同使用完成一个典型赛百平台应用。表 8.1 是 DVOMS 和现有的 VOMS 工具包的比较。

表 8.1 VOMS与DVOMS比较

功能	VOMS	DVOMS
组织关系	修改证书，通过证书链来表明组织关系	通过 VO 管理中心进行管理
信誉机制	没有	支持
动态性	稳定的 VO 组织	可以动态的建立/取消 VO 和修改 VO 关系
可移植性	与 GT 紧密结合	可以与 GT 通讯，也可以单独使用
VO 级的合作	不支持	支持

未来工作方向

1. CMESM 算法改进：CMESM 在抵御共谋欺诈行为方面表现良好，不过其准确率还有待提高。在实际应用中，即使极少量的欺诈成员也有可能对虚拟组织内部成员的安全造成很大的威胁。因此选用合适的统计参数对赛百成员历史交互行为进行描述，根据评价反馈动态调整虚拟组织委员会各个代表的权重将有助于 CMESM 性能的提高。

2. 联邦模式多级虚拟组织管理过程中还存在着评价流程长（待审核成员需要所有子虚拟组织委员会进行评审），子虚拟组织访问更高层虚拟组织过程中存在安全隐患等问题。理论上一个虚拟组织对申请成员的评价完全依赖于委员会中各个代表的权重和其知识库。下一步希望能够找出一种通行的算法，通过比较父虚拟组织和子虚拟组织的委员会来判定各个虚拟组织对成员的偏好和要求，训练出满足所有委员会，能动态调整的评价器，作为父虚拟组织的评价方法。这样可以大大减少子虚拟组织委员会和父虚拟组织委员会在评价申请成员中的通讯开销，可以大大提高效率和响应时间。

3. DVOMS 中间件还存在着安装繁琐，软件依赖性强和功能较为单一

等问题，特别是在架构 CI 服务器过程中，需要安装多项相关软件包和进行数据库配置。这些工作对于非计算机相关专业的用户来说困难较大。因此，进一步完善 DVOMS 功能，简化安装流程是 DVOMS 下一步开发目标。

参考文献

[1] LIGO – Laser Interferometer Gravitational-wave Observatory. http://www.ligo.caltech.edu.

[2] LSC – LIGO Scientific Collaboration. http://www.ligo.org.

[3] Atkins D. E. et al. Revolutionizing Science and Engineering through Cyberinfrastructure. National Science Foundation Blue – Ribbon Advisory Panel on Cyberinfrastructure, January 2003.

[4] Foster I. and Kesselman C., The Grid: Blueprint for a New Computing Infrastructure, Morgan Kaufmann Publishers, 1998.

[5] NEES – Network for Earthquake Engineering Simulation. http://www.nees.org.

[6] OSG – Open Science Grid. http://www.opensciencegrid.org.

[7] NEON – National Ecological Observatory Network. http://www.neoninc.org.

[8] GEON – The Geosciences Network. http://www.geongrid.org.

[9] NCAR – National Center for Atmospheric Research. http://www.ncar.ucar.edu.

[10] NCN – Network for Computational Nanotechnology. http://www.ncn.purdue.edu.

[11] NVO – US National Virtual Observatory. http://www.us-vo.org.

[12] TeraGrid. http://www.teragrid.org.

[13] OCI – Office of Cyberinfrastructure. http://www.nsf.gov/oci.

[14] NSF Cyberinfrastructure Council. NSF's Cyberinfrastructure Vision for 21st Century Discovery, Version 7.1. National Science Foundation, July 2006.

[15] Katzy, B.R., Design and implementation of virtual organizations, System Sciences, 1998., Proceedings of the Thirty-First Hawaii International Conference, 142 - 151 vol.4,1998.

[16] Niinimaki, M., White, J., de Cerff, W.S., Hahkala, J., Niemi, T., Pitkanen, M., Using virtual organizations membership system with EDG's grid security and database access, Database and Expert Systems Applications, Proceedings. on 15th International Workshop, 517 – 522, 2004.

[17] YuanPu Shao, Lee, M.K.O., ShaoYi Liao, Virtual organizations: the key dimensions, Research Challenges, 2000, Research Challenges, 2000. Proceedings. on Academia/Industry Working Conference, 3 – 8, April, 2000.

[18] Sinnott, R.O., Chadwick, D.W., Doherty, T., Martin, D., Stell, A., Stewart, G., Su, L. Watt, J., Advanced Security for Virtual Organizations: The Pros and Cons of

Centralized vs Decentralized Security Models, Cluster Computing and the Grid, 2008, 8th IEEE International Symposium, 106 – 113, May 2008.

[19] LIGO- Laser Interferometer Gravitational-wave Observatory, http://www.ligo.org.

[20] http://vdt.cs.wisc.edu/VOMS-documentation.html.

[21] http://hep-project-grid-scg.web.cern.ch/hep-project-grid-scg/voms.html.

[22] Dongguk Univ. Grid Information Retrieval Management System for Dynamically Reconfigurable Virtual Organization, Grid and Cooperative Computing, 2006. GCC 2006. Fifth International Conference, Oct. 2006.

[23] EDG – European DataGrid http://eu-datagrid.web.cern.ch/.

[24] DataTAG – Data TransAtlantic Grid http://datatag.web.cern.ch/datatag/.

[25] SBGrid – Structural Biology Grid http://www.sbgrid.org/.

[26] GUGrid – Georgetown University Grid http://gugrid.arc.georgetown.edu/.

[27] Farrell S., Housley R., An Internet Attribute Certificate Profile for Authorization, http://www.rfc-editor.org/rfc/rfc3281.txt, April 2002.

[28] GUMS-Grid User Management Service, https://www.racf.bnl.gov/Facility/GUMS/1.2/index.html.

[29] Sandhu, R.S. Samarati, P. Access control: Principles and Practice. Communications Magazine, IEEE. Volume: 32 , Issue: 9 On page(s): 40 – 48, 1994.

[30] Thomas, T.; A mandatory access control mechanism for the Unix file system, Aerospace Computer Security Applications Conference, 1988.

[31] Yixin Jiang; Chuang Lin; Hao Yin; Zhangxi Tan, Security analysis of mandatory access control model, Systems, Man and Cybernetics, 2004 IEEE International Conference, Volume 6, 10-13 Oct. 2004.

[32] Andreas Schaad, Jonathan Moffett, Jeremy Jacob. The role-based access control system of a European bank : a case study and discussion, Proceedings of the sixth ACM symposium on Access control models and technologies, May. 2001.

[33] Ninghui Li; JiWon Byun; Bertino, E, A Critique of the ANSI Standard on Role-Based Access Control, Security & Privacy, IEEE, Volume 5, Issue 6, 2007.

[34] Privilege and Role Management Infrastructure Standards Validation: http://www.permis.org

[35] Alexandria, Virginia, Trust Management for Trusted Computing Platforms in Web Services, Conference on Computer and Communications Security, Proceedings of the 2007 ACM workshop, Scalable trusted computing, Nov. 2007 - Nov. 2007.

[36] Guangwei Zhang, Jianchu Kang, Rui He, Towards a Trust Model with Uncertainty for e-Commerce Systems, Proceedings of the 2005 IEEE International Conference on e-Business Engineering (ICEBE'05).

[37] Alireza Pourshahid, Thomas Tran, Modeling Trust in E-Commerce: An Approach Based on User Requirements.

[38] Tao Sun，Mieso K. Denko A Distributed Trust Management Scheme in the Pervasive Computing Environment，Electrical and Computer Engineering, 2007. CCECE 2007. Canadian Conference on 22-26 April 2007 Page(s):1219 – 1222.

[39] Zhen. Wang, Junwei. Cao, Committee-based evaluation and selection of Grid resources for QoS improvement, 10th IEEE/ACM International Conference on Grid Computing, 2009, pp. 138 – 144, 2009.

[40] Chih-Chung Chang, Chih-Jen Lin (2001). LIBSVM - A Library for Support Vector Machines. URL http://www.csie.ntu.edu.tw/~cjlin/libsvm/.

[41] Yoav Freund, Robert E. Schapire: Experiments with a new boosting algorithm. In: Thirteenth International Conference on Machine Learning, San Francisco, 148-156, 1996.

[42] Baldo N., ZorzH M., Cognitive Network Access Using Fuzzy Decision Making, IEEE International Conference on Communications, ICC, pp.6504-6510, 2007.

[43] Faloutsos M., Faloutsos P., Faloutsos C., On Power-Law Relationship of the Internet Technology, Proc. ACM SIGCOMM' 99, pp. 251-262, 1999.

[44] Siganos G., Faloutsos M., Faloutsos P., Faloutsos C., Power laws and the AS-level Internet Topology, IEEE/ACM Transactions on Networking (TON), 2003.

[45] Braun T. D., Siegel H. J., Beck N., Bölöni L. L., Maheswaran M., A. I. Reuther, J. P. Robertson, M. D. Theys, B. Yao, D. Hensgen, R. F. Freund, A Comparison of Eleven Static Heuristics for Mapping a Class of Independent Tasks onto Heterogeneous Distributed Computing Systems, Journal of Parallel and Distributed Computing, Vol.61(6), pp.810-837, 2001.

[46] Maheswaran M., Ali S., Siegel H. J., Hensgen D., and Freund R., Dynamic Mapping of a Class of Independent Tasks onto Heterogeneous Computing Systems, Proc. 8th Heterogeneous Computing Workshop (HCW '99), pp. 30-44, 1999.

致　谢

衷心感谢导师曹军威研究员对本人的精心指导。在论文研究和软件开发阶段，曹军威研究员高屋建瓴，以广博的领域知识和敏锐的前沿把握对作者进行了详实的指导，帮助作者把握正确的研究方法，引导作者开阔思路，克服研究和开发中的困难。曹军威研究员踏实，热忱的工作作风将使作者一生受益。

感谢林筱同学在软件实现过程中做出的努力，感谢实验室其他成员在学术上和生活上对作者的帮助，并与作者一起度过了难忘充实的求学时光。

感谢实验室全体老师和同窗们学的热情帮助和支持！

本课题承蒙国家自然科学基金资助，特此致谢。

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：_____

个人简历、在学期间发表的学术论文与研究成果

个人简历

1985 年 12 月 27 日出生于陕西省子洲县。

2003 年 7 月考入清华大学自动化系控制科学与工程专业，2007 年 7 月本科毕业并获得工学学士学位。

2007 年 9 月免试进入清华大学自动化系攻读控制科学与工程硕士至今。

发表的学术论文

[1] Zhen. Wang, Junwei. Cao, Committee-based evaluation and selection of Grid resources for QoS improvement, 10th IEEE/ACM International Conference on Grid Computing, 2009, pp. 138 – 144, 2009. (EI 检索，Accession number: 20101112765832, Number of references: 19).

[2] Zhen. Wang, Junwei. Cao, Trust Cooperation Negotiation Platform based on Federal Mechanism in Cyberinfrastructure Environment, on <Cyberinfrastructure Technologies and Applications>, Vol.11, Nova Science Publishers, Inc. Commack, NY, USA, 2009.

[3] Zhen Wang, Junwei Cao, Xiaoge Wang, Lindy Blackburn, Erik Katsavounidis, SVM Multivariate Veto, Gravitational Wave Data Analysis Workshop-14, Poster, Jan. 2010.

研究项目

2008.1 至 2010 年 4 月，参与赛百平台软件 ELOP 的开发设计，主要负责虚拟组织层软件的开发设计。

2009.6 至今，参与美国麻省理工学院 LIGO 国际合作项目，并发表 Poster 一篇。