

网格 成长

Adaptive Grid

主办：华中科技大学网络俱乐部

1

2005
总第二期

网格界的豪门盛宴 —— GCC2004专题报道

Anysee: 一种基于对等网格的在线流媒体直播系统

Services@Anywhere简介和jUDDI介绍

中国教育科研网公共支撑平台简介

信息服务网格项目组简介

E-Learning



Ian Foster 博士简介

作为“网格之父”的Ian Foster博士为网格的发展和应用做出了巨大的贡献, 这里对他作简单的介绍



CGSP数据管理模块简介

为满足ChinaGrid中的大量数据密集型计算, CGSP数据管理模块提供了独特的功能



网格技术在地球信息科学中的应用

为满足每个国家对空间信息资源的巨大需求, 网格技术为空间信息获取和处理提供了新的技术途径

卷首语

2004年，振奋人心的一年！在这一年，中国人让世界震惊了……奥运赛场上，第一次，炎黄子孙摘下了100米跨栏的金牌，捧走了网球女双的桂冠；信息产业界，第一次，华夏后代收购了蓝色巨人的传统基业，把联想播向全世界；文艺界，第一次，一股秦风卷走了山姆大叔的票房第一……

这一切都让人如此的振奋，我们关注的网格也丝毫不示弱的迎来了属于中国网格的元年！十月份武汉的一场豪门盛宴，聚集了来自全球各地的网格专家，在这里，中国网格的成长让世界吃了一惊！正如“网格之父”Ian Foster教授评价中国教育科研网格公共支撑平台时所说的“*That's an amazing work!*”（“这是一件令人震惊的工作”）——中国人的网格也走到了世界的前沿，开始引领新科技的风骚！

《网格成长》成员组的每位编辑怀着无比激动和澎湃的心情，记下了这一年中国网格的成长中所经历激动人心的一幕幕精彩与大家分享，和大家一起来感受这份成长的喜悦与振奋！希望通过我们的杂志可以吸引更多的人来建设中国人自己的网格，在网格世界里面筑起属于我们的万里长城！

当然，在中国网格飞速成长的一年中，由于我们的能力所限，无法记述发生所有的一切，我们也希望更多的中国网格的建设者们能关注我们的杂志，把你们的工作内容和进度告诉我们，希望通过这个窗口，可以见证中国网格成长的每一个足迹。

编者 于瑜园
2004年12月30日

共争朝夕，
共绘中国教育
科研网络蓝图
熠熠明天！

肖农 (国防科大)	郑伟民 (清华大学)	海松 中山大学
李昭东 (上海交大)	李晓明 (北京大学)	
陈小武 (北京航空航天大学)	董小庄 (西安交大)	张凌 华南理工大学
罗尧舟 (东南大学)	王兴伟 (东北大学)	刘宏 中科院
高晓旭 (山东大学)	董宇斌 (华南理工大学)	

《网格成长》

Adaptive Grid

2005年1月1日 第1期 总第2期

编委会主任：金海

主编：齐力

执行主编：贾勇

编辑：黄晓菲 童心
杨思睿 黄莉

美术编辑：李方杰

顾问：刘伟 叶启伟 王福臣

荣誉顾问：

中国教育科研网格专家组成员

郑纬民(清华大学)

李晓明(北京大学)

陈小武(北京航空航天大学)

张凌(华南理工大学)

李明禄(上海交大)

董守斌(华南理工大学)

罗军舟(东南大学)

董小社(西安交通大学)

王兴伟(东北大学)

肖 依(国防科大)

孟祥旭(山东大学)

曾海标(中山大学)

目录

人物与报道 >>>

5 网格界的豪门盛宴——GCC2004 专题报道

7 Ian Foster 博士简介

9 Ian Foster：最近网格界的变化

技术专栏 >>>

14 中国教育科研网格公共支撑平台简介

18 CGSP 数据管理模块简介

22 E-Learning

24 Services@Anywhere 简介和 jUDDI 介绍

27 Anysee：一种基于对等网络的在线流媒体直播系统

32 信息服务网格项目组简介

33 ImageGrid：图像处理网格原型及典型应用研究

网格应用 >>>

37 Butterfly.net

39 网格技术在地球信息科学中的应用

42 GridRM



征稿

网格俱乐部所创办的《网格-成长》杂志是国内第一份由高校学生创办，着力于普及网格基础知识的杂志。杂志涵盖的领域包括：网格应用开发案例的分析和讨论，开发经验的交流，网格应用前景，P2P 计算以及网格相关技术技巧等方面的文章。我们热诚的期待您的投稿。

您也可以通过访问我们的官方网站：

<http://grid.hust.edu.cn/gridclub>

来获取俱乐部最新信息和下载《网格-成长》杂志的最新版本。我们热忱欢迎您的建议和批评！

联系方式：grid.club@gmail.com

联系电话：027-87547047 87543529

联系人：齐力 贾勇

恭贺新禧

《网格·成长》全体工作人员

祝大家在 2005 年里万事如意，心想事成！



由华中科技大学集群与网格计算湖北省重点实验室承办，中国计算机学会、IEEE 北京分会协办的“第三届网格与协同计算国际学术会议（The 3rd International Conference on Grid and Cooperative Computing, GCC 2004）”于2004年10月21日至24日在华中科技大学成功举行。有来自20多个国家和地区的400多名代表与会。就P2P计算、网格中间件、网络安全、网格应用、资源管理和调度、信息网格和知识网格、网格服务等进行了深入和广泛的讨论。网格和协同计算国际学术会议（GCC）自从2002年成功举办第一届以来，迅速扩大影响，它已成为国际网格计算领域研究、开发、应用人员和企业就网格计算技术现状和发展趋势进行交流的重要舞台。

网格是构筑在互联网上的一种新兴技术。网格的目标是将各种硬资源、软资源，如网络、计算机、数据库、传感器、设备、知识等融为一体，实现互联网上所有资源的全面连通、全面共享，为广大用户提供更多的资源、功能和交互性。网格计算最初主要被各

大学和实验室用在需要进行高性能计算的项目之中。正是网格聚合资源的巨大能力，许多国家和地区，如英国、荷兰和美国等都已制定并实施了网格计划。随着网格计算技术的进一步成熟，应用前景更为广阔。企业预感到了网格所潜在的巨大市场，也纷纷加入了网格研发的队伍。IBM、Microsoft、SUN、HP等都纷纷提出面向企业的网格解决方案。除开IT公司以外，其它行业领先公司，如辉瑞制药公司、波音公司等大公司，也已开始进行有关网格计算的研究和应用。网格不但将成为生物工程、气候模拟、能源探索、航空航天、数字地球等重要科技领域的重要工具，而且网格技术也将提升电子商务到一个新的台阶，它将有效地整合互联网上的资源，为电子商务提供内容、计算、存储和交易等服务。

正是由于网格与协同计算在解决广域范围内分布、异构的各种资源的共享与协同工作，这一困扰几乎所有大型应用领域的难题的巨大潜力，中国科学院、中国计算机学会以及IEEE北京分会于2002年

发起并共同主办了第一届“网格与协同计算国际会议”。自从 2002 年成功举办第一届网格和协同计算国际学术会议以来,影响迅速扩大。到 2003 年上海交通大学举办第二届网格与协同计算国际学术会议时,会议规模达到了新高。由于华中科技大学集群和网格计算湖北省重点实验室在网格研究领域的良好学术声誉,第三届网格与协同计算国际学术会议由华中科技大学集群和网格计算湖北省重点实验室承办。

GCC 2004 的大会指导委员会主席由中科院计算所的李国杰院士担任。教育部副部长赵沁平教授担任大会名誉教授。大会主席由国防科技大学的卢锡诚院士和美国加利福尼亚大学圣迭戈分校的 Andrew A. Chien 教授担任。大会程序委员会主席由教育部“ChinaGrid”计划专家组组长、华中科技大学计算机学院副院长金海教授和美国乔治亚州立大学 YiPan 教授担任。

大会程序委员会主席金海教授和 YiPan 教授进行了大量艰苦细致和卓有成效的组织工作,各项工作筹备得井井有条,取得了显著的成果。第三届网格与协同计算国际学术会议受到了国内外学术界和工业界的广泛重视和支持。本次会议在规模上和学术上都更上一层楼,与 GCC 2004 同时举行的还有 5 个 Workshop: International Workshop on Storage Grid and Technologies (SGT 2004); Information Security and Survivability for Grid (GISS 2004); International Workshop on Agents, and Autonomic Computing, and Grid Enabled Virtual Organizations (AAC-GEVO 2004); International Workshop on Visualization and Visual Steering (VVS'2004); International Workshop on Information Grid and Knowledge Grid (IGKG's 2004)。会议(包括 Workshop)收到了来自美国、英国、德国、加拿大、瑞士、爱尔兰、澳大利亚、印度、日本、韩国、新加坡、波兰、香港和台湾地区等 20 多个国家和地区约 600 篇投稿。受教育部“十五”211 工程公共服务体系建设项目和国家技术研究发展计划(863 计划)项目支持的中国教育科研网格 ChinaGrid 计划,在这次会议的展厅里得到了展示了最近研究进展。中国教育科研网格支撑平台由华中科技大学、清华大

学、上海交通大学、北京航空航天大学联合开发,它基于 Web 服务的参考架构,达到国际先进水平。该平台利用中国教育科研网和高校的大量计算资源和信息资源,实现资源的有效共享,消除信息孤岛,提供有效的服务器,形成高水平、低成本的计算服务平台。GCC 2004 在新的高度上促进中国在网格与协同计算相关领域的研究进展,促进国内外研究人员,工业界和学术界的交流。

该会议在更高层次上把工业界领先的企业和学术界领先人物汇集在一起,交流对科技发展的观点,共同探讨本领域需要解决的迫切问题。会议邀请到了众多国内外著名学者和专家的参加。会议邀请到的国外著名学者包括:美国 Argonne 国家实验室资深科学家、国际网格计算杂志两主编之一、美国网格计算项目的领导人 Ian Foster 教授,美国田纳西大学负责测试和公布世界超级计算机 500 强排行榜的科学家、美国工程院院士 Jack Dongarra 教授,制定全球网格标准的国际组织全球网格论坛(GGF)主席、美国 Argonne 国家实验室资深科学家 Charlie Catlett,英国政府有史以来投资强度最大的科学研究计划英国 e-Science 计划首席科学家、英国南安普顿大学 Tony Hey 教授, Entropia 公司创始人、美国加利福尼亚大学圣迭戈分校计算机科学与工程系科学系首席教授 Andrew A. Chien, 美国乔治亚州立大学 Yi Pan 教授等。会议邀请到的国内著名专家包括:中国教育科研网格计划 ChinaGrid 专家组组长、华中科技大学计算机学院副院长金海教授,科技部 863 计划网格专项专家组成员、中科院计算技术研究所副所长徐志伟研究员,科技部 863 计划网格专项专家组组长、西安交通大学钱德沛教授;清华大学郑纬民教授、北京大学李晓明教授,中国教育科研网格计划 ChinaGrid 专家组所有专家,科技部 863 计划网格专项专家组所有专家等。还有许多来自国内外著名 IT 公司的高层领导和资深科学家。这些著名学者和专家的与会使得该会议成为 2004 年度学术界的一次盛会。

GCC 2004 会议日程安排非常饱满。整个会议期间安排了 11 场主题演讲,演讲人中既包括学术界著名学者:“网格之父”Ian Foster 教授、Jack Dongarra

教授、Charlie Catlett 和 Tony Hey 教授,也包括工业界领先企业的著名领袖人物或代表,如 HP 的 Greg Astfalk、Microsoft 公司的 Greg Rankich、Intel 公司的 Karl Solchenbach、曙光公司的代表孙凝晖教授、IBM 的 Moon (James) Kim、Dell 的 Eric D'Angelo、太阳微系统公司的 Andrew Lim。主题演讲的主题涵盖了网格标准,网格对社会的影响,高性能计算的发展情况,国外网格研究进展以及企业针对网格提出的解决方案。其中包括 Ian Foster 的 The Grid: Beyond the Hype, Jack Dongarra 的 High Performance Computing Clusters and Grids Oh My!, Tony Hey 的 e-Science and Web Service Grids_An Overview of UK e-Science 等等精彩的演讲。10月22日会议组织了一次别开生面的讨论,与会的多位主题演讲人在钱德沛教授的主持下就“目前网格应用的主要障碍是什么”进行了深入讨论,每位讨论参与人员就自身的经验和体会发表

了自己的看法,其他会议代表也纷纷向这些讨论参与人员提出了自己的疑问和理解。整个讨论气氛非常热烈。与主题演讲和论坛穿插进行了 37 次分会供作者和参会者进行讨论,这些分会主题包括 P2P 计算、网格中间件、网络安全、网格应用、资源管理和调度、信息网格和知识网格、网格服务等。总之,本届大会学术气氛浓厚,研讨的问题有相当的深度,主题报告具有很高的学术权威性。并且此次国际学术会议得到了 IT 世界的著名公司惠普、Intel、曙光、IBM、DELL、Sun、Microsoft、浪潮等的大力支持。

GCC 2004 为海内外网格研究者提供了一种良好的交流与合作平台:向世界展示中国网格研究的进展;供他们共同探讨网格与协同计算技术的最新发展。通过召开本次国际学术会议,使得国内学术界能与世界最新科技发展进行交流和竞技,为提升我国在网格计算相关领域的地位起到了很好的促进作用。

Ian Foster 博士简介



Ian Foster 博士于新西兰 Canterbury 大学(位于 Christchurch)获得了计算机科学学士学位,伦敦皇家学院计算机科学博士学位。

Foster 现在在美国阿岗国家实验室数学和计算

机科学部副主任和高级科学家,芝加哥大学计算机科学教授。Foster 的研究兴趣是大规模分布式计算中创新技术和应用。作为网格计算领域中国际著名科学家和领军人物,他已经出版了 5 本书,在并行计算、分布式系统和高级应用方面发表了超过 200 篇论文和技术报告。是圣迭戈超级计算机中心的高级成员和战略顾问以及全球网格论坛指导组成员。在国际网格计算领域享有盛誉。

关于网格相继发表了

The Grid: Computing With Bounds (Scientific American, April, 2003)

通过在全球范围连接数字处理机,存储系统和软件,网格技术能将个人性质或共同性质的计算转换为一种通用的计算。

The Grid: A New Infrastructure for 21st Century Science (Physics Today, February, 2002)

这篇 6 页的论文提供了网格计算的介绍,评论网格技术的当前状态,并且解释了网格技术如何应用于解决科学问题。

The Physiology of the Grid: An Open Grid Services

Architecture for Distributed Systems Integration(草稿 2002). 这篇 30 页的论文介绍开放式网格服务架构, 网格和 Web 服务技术的综合, 并且解释了这种架构是如何使得网格技术应用于电子商务和电子科学。

The Anatomy of the Grid: Enabling Scalable Virtual Organizations (IJSA, 2001)

这篇 30 页的论文根据对虚拟组织的支持对“网格问题”进行了定义, 并且定义了网格架构的主要原理, 同时介绍了 Globus Toolkit (Globus 开发工具包)

Computational Grids, Chapter 2 of The Grid: Blueprint for a New Computing Infrastructure (1999)

这篇早期的论文对网格计算进行了定义。

主要项目

Globus: 这个项目为从事高性能分布式计算提供了一个统一的框架。这其中包括安全, 资源管理, 交互协议, 数据管理机制和其它课题的研究, 这些都是基于大量的资源, 特别是 DOE Office of Science MICS (包括它的 SciDAC 的项目), NSF PACI 项目, NASA IPG, IBM, 和微软, 以及早期的 DARPA。

GriPhyN (网格物理网络) 和 PPDG (粒子物理数据网格) 是在 NSF ITR 和 DOE SciDAC 项目的支持成立的, 分别计划实现 21 世纪第一个用于数据集中科学的 Petabyte 数量级的计算环境。

VDGL (国际虚拟数据网络实验室) 正在建立一个数据网络的基础。

Earth System Grid: 这个项目在 DOE SciDAC 项目的支持下成立的, 正在研究对环境数据的协同和分布式分析的技术。

GRIDS Center: NSF Middleware Initiative 的一部分, 侧重于网格中间件的整合, 发布和支持。

NEESgrid: 一个深入研究地震试验和模拟的分布式虚拟实验室。NEESgrid 将扩大在地震研究中估计逼近的实验方法, 这其中包括: 对可以预测房屋反应的多种模式的发展, 大量的初始材料, 或是在大量负载的特别结构性的部分。

GrADS (网格应用程序开发软件) 这个项目致力于关于异质动态环境的应用程序开发的基本问题。

NCSA Alliance 是国际计算科学联盟的成员, 致力于开发计算科学的进一步基础。

Information Power Grid Globus 技术已经为 NASA 建立跟好的分布式计算环境。

ASCI FLASH 这个项目的目标是解决如中子星, 白矮星的致密星的表面和白矮星内部的高热原子核反应的闪光的长期没有解决的问题。

获奖情况

Fellow of the American Association for the Advancement of Science, 2003

R&D Magazine's Innovator of the Year, 2003

Illinois Innovation Award, 2003

InfoWorld Innovator, 2003, 2003

MIT Technology Review, one of "Ten Technologies that Will Change the World", 2003

University of Chicago Distinguished Performance Award, 2003

Silicon.com Top 50 Agenda Setter, 2003

Member, World Technology Network, 2003

Federal Laboratory Consortium Technology Transfer Award, 2002

Lovelace Medal, 2002

R&D Magazine's Most Promising New Technology Award (best of the R&D100), 2002

Fellow of the British Computer Society, 2002

Gordon Bell Award, 2001

Global Information Infrastructure "Next Generation" Award, 1997

Best Paper Award, 1995 Supercomputing Conference

British Computer Society Award for Technical Innovation, 1997

Ian Foster : 最近网 格界的变化

记者 : Mark Baker

译者 : lhztot@yht.org, ancry@yht.org

Globus Toolkit 作为开源的软件对广域的分布式计算提供了一个统一的框架,它由以下几个组件构成:安全、资源管理、通讯、数据管理等。在 1999 年出版的《the Grid (网格)》一书中,他和 Kesslman 首次提到了构成网格的几个主要要素。从此之后, Globus Toolkit 已经成为众多网格软件基础件实现中的先锋,网格真正达到可用性的一个关键组件。

从第一版开始, Globus Toolkit 经历了几次大的变化,但其中最大的一个变化就是转向基于服务的 (service-based) 实现。在 2003 年 7 月 Globus Toolkit3.0 的发布中,它就实现了 OGISI 标准 (译者注:OGISI 是一个网格服务的规范)。但是令网格界惊奇的是,在 2004 年 1 月 20 日的 GlobusWorld 会议上, Foster 和他的同事提出把 OGISI 的概念向网络服务资源框架 (WSRF, Web Service Resource Framework) 演化。

在 Foster 提出了 WSRF 的声明之后, IEEE 分布式系统的在线编辑 Mark Baker 联系了他,探讨了近来的变化及其对网格界的影响。

问题 1: 基于 OGISI 的网格服务与以前的网格建构 (Grid Architecture) 有很大的跳跃。把网格搭建在 WSRF 之上,是另一个具有重要意义的转变。现在的网格服务与 Web 服务有趋同的趋势,除此之外,什么人,什么事情有这么大的影响力,以至于要采用 WSRF?

WSRF 框架实际上重新分解了 OGISI 的概念,从而更好地符合 Web Services 标准。为什么需要对网格计算的标准和实现一再修改?按照 Foster 的解释,尽管 Web Services 的厂商 (vendors) 认识到了 OGISI

思想的重要性,但是他们不打算采纳 OGISI 1.0 发布时的定义。这样就威胁影响到了对 OGISI 规范的网格基础设施的广泛支持。而获得广泛的基础设施的支持是最初制定 OGISI 的主要目的。所以,在这种情况下,有必要重新组织 OGISI。而 Globus 联盟不想重新设计 OGISI (他们更乐于开发高层的服务,而不是纠缠于基础设施),所以 Globus 的结论是,对 OGISI 做一些重新分解如果能够促进网格服务和 Web 服务的聚合的话,那么这样就是正确的、值得的和有意义的。于是, Globus 与 Web 服务架构师组成了一个小组来研究这个问题,成果就是 WSRF。

尽管从 OGISI 到 WSRF 的演化主要是句法上的,但是也代表了一些有益的进展。把 OGISI 的功能分割成 6 个独立的规范,简化了别人的采用过程,而使用了 WS-Addressing 是一个进步,弱化对 XML Schema 和 WSDL2.0 的过分使用将会更加便于使用目前已有的工具。但是这些好处足以抵消对网格基础设施规范的修订带来的破坏么?站在纯技术的角度来看是很有争议的。但是事实上,我们已经获得了 Web 服务社区对网格基础设施强大支持,这对于网格社区来说是一个很大的成就 - - 这意味着我们将会在核心的 Web 服务产品里看到对 WSRF 的支持,这对于网格社区来说更是一个美妙的消息。但我们不应过分强调这个变化的影响范围:尽管这里有需要一些从 OGISI 到 WSRF 的转换工作,但是这个工作量看起来并不是非常大,而且我们正试图减轻这个过渡工作。

问题 2: 显然, WSRF 的计划和设计工作在一段时间前就已开始了,网格社区的很多人都很关心这一进展为什么要保密,而且有多少参与者,直到 2004 年 1 月的 Globus World 上宣布 WSRF,这个计划一直没有被透露。为什么要秘密进行?以前这些进程都是公开的,怎么现在好象和一些商业界的做法一样呢?

WSRF 的研究工作始于 2003 年夏末,在收到网络服务界对 OGISI 的反馈之后。幸运的是,我们在这个工作中成功得到了高级 Web 服务架构师的参与。不幸的是,他们希望与我们的合作方式在封闭的情况

下进行。幸运的，通过大量的艰苦工作，我们很快就结束了这个过程，迅速地提出这个规范供大家评论。我并不会说这个过程是理想的。然而，我们应该从这个过程的结果来评价它。在几个月的闭门讨论之后，我们提出了一系列的规范供大家探讨。这些规范已经被 OGSi 的作者们接受，许多网络服务界的人也接受了它们。我能想象更好的过程，我也能想象更坏的结果。

问题 3 :制定 WSRF 至少的一大动因就是为了使得网格服务更合 Web 服务社区的胃口。而这样做的结果就是，网格服务的概念好像完全消失了。这样说对么？所以为什么我们两年前需要网格服务，而现在不需要呢？

不，网格服务的概念没有消失。OGSI 和 WSRF 所致力于解决的本质问题——也就是从一个网格的观点来看，Web 服务所缺少的关键功能就是能够创建、定址、检查、发现和管理有状态的资源。在 OGSi 里面，这些有状态的资源被称作网格服务；而在 WSRF 里，他们被称为 WS-Resources。一个网格服务拥有一个身份 id、服务数据 (Service Data) 生命期管理机制；一个 WS-Resource 拥有一个名称、资源属性和生命期管理机制。术语变了，但需要、概念和机制并没有变。OGSI 和 WSRF 都定义了这些确实很重要的规范所需的机制，这些规范都是用来定义构建上层的 OGSA 的。OGSA 工作将会继续向前，这些变化对 OGSA 的影响很小。

问题 4 :OGSi 第一次提出时，其基本特征已经被相当秘密地制定了，Globus 项目已经沿着 OGSi 走了很长的时间了。网格界中大部分人都原谅了你，投入了很大的精力来理解 OGSi，修订了他们的项目计划。他们还会再次这样做么？

你开始说的不对：OGSi 不是在秘密中制定的。从 2000 年起，我们就发现网格社区越来越多的人愿意采用 Web 服务作为网格的实现技术。与此相对应

的，Globus 联盟和 IBM 组成了一个小组在 2002 年 2 月提出了一个 OGSi 规范的草案。随后，在 GGF 的一个工作组内经过一年半的全面细致的检查审阅后，于 2003 年 6 月才形成了最终的 OGSi 规范。我认为这是一个极好的例子，Globus 联盟看到了大家新的要求，而预先研发，并制定了开放的标准。

WSRF 的初始动机也是为着相同的目标，并且沿着同样的进程进行的。网格的成功需要采用已经被广泛地采纳应用的标准化了的基础设施来创建、定址、检查、发现和管理有状态的资源。OGSi 定义了需要的机制。WSRF 稍微改编了其中的陈述来便于在商业的 Web 服务工具里的应用。它是前进路上的一次颠簸，但不是非常严重，而获得的好处是巨大的。

问题 5 :许多项目都向 OGSi 靠拢了一段时间了，有些已经在开发网格服务，这些工作是浪费么？另外一些坚持使用 Web 服务，希望一旦一个广泛部署的 OGSi 基础件到位时，转向网格服务是没有痛苦的、很快乐的一件事情。似乎后者等待 WSRF 是更明智的，但是对许多项目来说，他们的生命期太短了。而且还有很多坚持用 GT2，现在好象可以直接用 GT4，而不必理会 GT3 了。一些人正在考虑是否使用 Globus，而使用 Globus 的替代品。你怎么能说服牢骚满腹的大家，与转向所经历痛苦相比，长期的利益会更大呢？我们是不是也遭遇了阻碍网格发展的一个激烈反应呢？就象 AI 的冬天一样，现在也是网格的冬天了呢？

e-science 和 e-business 仍在持续增长：例如最近 Oracle、IBM、Sun 还有其它一些公司最近都宣布支持它们；美国的 Cyberinfrastructure 项目和新的 EU 项目也在进行中。同样地，人们也认识到了中间件扮演了一个极重的角色：保证了安全性、可靠性、互操作的资源共享与管理。因此，我不关心会有激烈的反对反应：网格已经到来了，不管我们在构建它时经历多大的痛苦，我们也要接受它。

我不想低估你谈到的“剧变”，但是我想，大家经常把它的想得过于严重了。当 Globus 联盟最初在 GT3.0 里包括对 OGSi 支持的时候，我们承诺支持

GT 的 pre-Web 服务部件,许多用户继续使用它们。确实我们继续推荐他们用于实际的产品级服务,因为目前基于 Web 服务 (WS-based) 的部件还没有达到产品级的水平。WSRF 是从 OGSi 的一个渐增的变化,我们在将来一段时间内将继续支持 OGSi。

分布式系统的开发者必须仔细地审视可用的技术,然后决定那一种最适合需要。Globus Toolkit, OGSi 和 WSRF 都不是适用于所有情况的。但是,我们也不要不加考虑就忙下结论“我的问题是不同的或者很简单,所以我还是开发自己的中间件把”。分布式计算在多个层面上都是很复杂的,从安全性、可靠性到可信度管理;互操作性总是具有挑战性的。在标准框架下工作有很多优势,能解决一些很紧迫的问题:一次登录,多次使用;远程布署;计算管理、数据移动等。采用 GT 的用户能得到这方面的帮助,还能在一个开发者和用户所组成的全球范围的社区内讨论。

问题 6:你对提议的 WSRF 标准获得认同的时间表是什么?你认为 GGF 作为一个标准组织存在可行么?

WSRF 已经被列入了 GGF 的 OGSi 工作组的讨论日程,将会对其进行大量的讨论。这些规范很有可能很快被提交到 OASIS (Organization for the Advancement of Structured Information Standards),为了保证网格的需求能够得到满足,OASIS 与 GGF 启动了一个正式的联络进程。这么做的原因很简单:WSRF 不是一组纯粹的网格标准,而是一组与网格非常相关的 WS 标准,WS 标准通常情况下是由 Oasis 或者 W3C 处理。提交之后,什么时间能完成这个标准的制定还是未知的,不过 GGF 的 OGSi 小组付出了巨大努力之后,这个进程会更快进行。

有人认为把 WSRF 提交给 Oasis 说明 GGF 并不是一个独立的标准化组织。我不这么想。GGF 已经提出了许多正在应用的网格标准,比如 GridFTP 和 GSI。其它的一些规范还在制定中,比如 DAIS。很自然地,网格也建立在其它组织(如 IETF、W3C 和

Oasis) 的标准之上。一些 GGF 的参与者也参与了其他的这样的组织,例如,OGSi 的作者作为了 W3C 的 WSDL 工作组的特邀专家。GGF 的 CMM 和 Oasis 的 WSDM WG 也存在交集。我们期望看到更多的这种情况,这些进展都能使 GGF 不仅仅作为网格标准的制定者,而且也是网格要求能集成进其它标准化组织的工作中去的推进者。

问题 7:OGSi 规范除了 GlobusToolkit 实现之外,还有一些其他的实现。可是一旦 Globus 弃船逃跑,转向了 WSRF,OGSi 还会拥有这么多追随者么?

由于我刚才提到的原因,Globus 联盟致力于把 Globus Toolkit 升级成 WSRF。据我所知,其它的 OGSi 实现,如 pyGlobus、OGSi.NET、OGSi::Lite 和 Unicore 也有同样的趋势,就如 OGSA 相关规范的主要设计者,象 WSDM (管理)、DAI (数据访问和集成) 和 WS-Agreement 所做的一样。因此,虽然没有理由不继续在 OGSi 框架下工作(我们将继续支持 GT 3.x 的基于 WS 组件一段时间,具体依赖于我们的人力和用户的需要),但应该准备明年向演进到 WSRF。

问题 8:许多科学家仍然使用库风格的 API,并不理解 Web 服务或者网格服务。而对 GT2 那样的风格库包还会继续支持么?

当前各种社区正在努力学习通过创建服务而不是共享程序或者数据来提供汇聚专门技能知识,我完全期望科学计算的问题解决环境(PSE, problem-solving environment)越来越转向面向服务。例如,这个概念就是国际虚拟天文台启动计划(International Virtual Observatory Initiative)的核心思想。稍小规模美国的 Fusion Collaboratory 项目通过远程访问仿真代码,就免除了远程用户需要下载和安装复杂的应用的麻烦。更一般的,网络服务是一个建立大规模分布式系统的强有力的框架,通过一个严格的接口,可以不受约束地进行开发。然而,许多社区和用户仍然需要特殊的功能(例如,访问远处的计算机或数据)这是通

过 GT2 风格库创建和管理远程的计算、传递数据等实现的。这就是 GT3 和 GT4 提供这些功能的客户端 API 的原因。只要有类似的重要的需要, 这些 API 就会继续被支持。确实, 我们希望 GT 以及基于 GT 的工具中这样的 API 会更多, 这样, 更多的高层次的、面向应用的网格工具就会被开发出来。

问题 9: OGSi 规范的一个缺点就是对于某些焦点问题说得很少, 例如如何支持两个独立的但是兼容的实现进行互操作。例如, GT3 中的安全模型就是 GSI 和 WS-Security 的一个非标准的混合体。WS-Addressing 应该会有助于避免重复解析句柄的难题。Globus 联盟是否也采取了措施来避免重蹈覆辙, 例如互操作性?

OGSi 规范集中于网格服务的 WSDL 接口。互操作涉及到了许多其他的问题, 如安全和可靠的通信。我们与其他 OGSi/WSRF 的实现研究人员保持紧密的联络, 希望能够尽快进行互操作测试。这个工作应该能有助于构画出 OGSi/WSRF 实现的互操作性的轮廓。

最终地, 互操作需要一个广泛的 WS 和网格标准的标准化。只有在这些标准都到位了并且满足了网格需要的时候, 才能完全达到采用 WS 作为网格的基础的价值。因此, 我们就很有必要参与到更广泛的 Web 服务社区和象 WS_I 论坛那样的组织中去才能确保我们的符合 WSRF 标准的服务具有完全的互操作性, 不管是由网格提供的还是由 Web 服务中间件提供的。

问题 10: 怎样从 WSRF 网格上构建信息服务? 有指定的机制(如监视和发现服务)么? 或者说, 会用现有的注册技术么?

OGSi/WSRF 中令人振奋的一个新特征就是信息服务机制的高度集成。任何网格服务(在 WSRF 中叫 WSRF 资源)都可以声明服务数据(WSRF 中的资源属性), 这些服务数据都可以用标准的机制发现、取出、放入。WSRF 的 WS-Notification 规范使

我们迈了重要的一步, 更广泛地使用基于消息的中间件建立可以探知的接口, 就可以提供更健壮的、内容更丰富的信息服务。打个比方, 一个程序员开发了一个文件传输服务, 它只需要定义一些服务数据(资源属性), 这个服务就对别人可见, 并可以被监视。在此基础上, 就有可能定义一个更大范围的信息组件, 来发现、监视、得到它们, 对出现的错误也可以捕获到, 等等。Globus 联盟正在开发这些组件的雏形, 包括一个存档服务和注册服务, 状态提供者可以更快地更新状态信息。第一版将出现在 GT3.2, 更完善的将在 GT4.0。

问题 11: 在 WSRF 的可靠实现可用之前, 你对开发者有什么建议? 我们什么时候能够看到其他语言的 WSRF 组件, 而不仅仅是 JAVA?

先谈一下 Globus Toolkit。GT3 包括了 GT 的 pre-WS 组件的可靠实现和基于 WS 组件的早期实现。即将发布的 GT3.2, 对基于 WS 组件做了重大改进, 在继续支持 pre-WS 组件的同时, 更强调了网络服务的可用性、健壮性和性能。GT4.0, 可能会在 2004 年第三季度发布, 将引进对 WSRF 的支持, 同时会继续改进 GT 的基于 WS 组件的可用性、健壮性和性能, 支持 pre-WS 组件。

其它的 OGSi 实现也计划向 WSRF 演进, 不过我并不知道具体的时间表。Stephen Pickles 最近在 OGSi WG 的邮件列表上宣布他已经几乎完成了 OGSi::Lite 向 WSRF 的转变。

借助这个场合, 我提出几点建议:

如果你用 pre-WS 工作得很好, 就象很多产品化的网格一样, 那么继续。它们仍将在 GT3 和 GT4 中得到支持。只要还有需要, 我们会继续支持 pre-WS 组件, 我们有人力支持到 2005 年。

如果你正在用 GT3.0 的基于网络服务的 GT 组件, 为了得到更好的性能, 在 GT3.2 发布时, 转向它。进一步的, 如果 WSRF 对你有意义, 转向它。我们会继续支持 GT3.x 组件 我们有人力支持到 2005 年。

如果你刚刚开始熟悉网络服务和网格, 先用

GT3.2, 然后到明年的某个时间转向 GT4.x。

问题 12: 最后一个问题来自英国, 我知道英国的 e-Science 项目从 2002 年就开始了, 而它的基础设施里面许多都是与 Globus 技术紧密相关的。事后看来, 英国的 e-Science 项目是不是开始的过早?

不是的。英国的 e-Science 项目启动的时机非常好, 使得英国在开发网格上起到了一个清楚的作用。例如, 如果没有英国的 e-Science 项目的资助, OGSA-DAI 的工作就会启动的更晚或者根本就不能起动, 英国就不会在数据服务起先导作用。Globus 联盟也不会把 Edinburgh 大学加入进来。

对于这个问题, 我想澄清一个与之相关的误解。e-science 方面的工作就是关于创建 21 世纪的科学所需要的环境。很自然地, 类似于建筑, 我们更愿意直接搬入一个已建好的房子, 而一些用户好像就是希望 Globus 是一个这样的房子。但是, 正如它的名字所说, Globus Toolkit 仅仅是一系列的工具, 并不是一个完整的解决方案。对于熟练的开发者来说, 只要适当地运用这些工具来解决认证、发现、远程访问和其他的分布式计算的挑战, 就能开发出很好的成果。但是成功并不是仅仅在有了科学家和 Globus Toolkit 之后, 就自然而然地完成了。它需要在科学家和网格技术的合作, 和 / 或可用的高层面向特定应用的解决工具, 就象 GriPhyN/PPDG 的虚拟数据工具包, Access 网格和一些 portal 工具。

所以从整体上来讲, 网格的发展需要创造性和理解。随着经验的积累, 我们也在不断深化我们的认识, 知道怎么更好地建立一个健壮的、可靠的全球的网格。作为合作的很好的例子, 我们看一下今年就要完成的地震模拟网格所组成的网络 (Network for Earthquake Engineering Simulation Grid)。Neesgrid 用 GT3 的 OGSi 软件建立了一个成熟的、分布式的实验控制和协作系统。其中的一些工作需要把 OGSi 软件转向 GT4, 不过 Neesgrid 设计者认为变化的代价与现在已经布署在各个 Neesgrid 节点上的集成的协作工具取得的巨大进展和对大家的培训相比, 仍然很

小。去年 7 月, Neesgrid 完成了一个多站点结构的地震试验, 试验设备在 Colorado 和 Illinois, 模拟系统在 Illinois, 数据文件和人力参与达 12 个站点之多。如果没有 Neesgrid, 这种新的科学就不会完成。

我举的第二个例子是美国的 Grid3 系统。从 2003 年 11 月开始, 已经用 GT 的 pre-WS 组件, 在美国和韩国的 27 个网格子节点上持续运行了 500 到 2000 个数据分析任务, 涵盖一系列物理、生物和计算机科学项目。最终用户和 GriPhyN/PPDG 虚拟数据工具箱 (Virtual Data Toolkit) 的开发者之间展开合作, 使用 GT 机制来协同在多个站点上的数据密集的工作流的调度, 目前这个计划已经取得了很大的应用成果。此外, 对于创建和运行一个可操作的网格需要什么, 还积累了许多经验。

Mark Baker 是英国 Portsmouth 大学计算机学院分布式系统的高级讲师, Westminster 大学 Cavendish 计算机学院的计算机科学访问教授。联系方式:

mark.baker@computer.org 

缩略语

CMM: Common Management Model
 DAIS: Data Acquisition for Industrial Systems
 GGF: Global Grid Forum
 GSI: Grid Security Infrastructure
 IETF: Internet Engineering Task Force
 MDS: Monitoring and Discovery Service
 Neesgrid: Network for Earthquake Engineering Simulation Grid
 Oasis: Organization for the Advancement of Structured Information Standards
 OGSA: Open Grid Services Architecture
 OGSi: Open Grid Service Infrastructure
 W3C: World Wide Web Consortium
 WG: Working group
 WS: Web services
 WSDM: Web Services Distributed Management
 WSRF: Web Services Resource Framework



1. CGSP 整体结构

ChinaGrid 公共支撑平台 (ChinaGrid Support Platform, 简称 CGSP) 是为 ChinaGrid 的建设和发展而研制的网格核心中间件。CGSP 基于当前 CERNET 网络及将来 CERNET 高速传输网, 提供了一套完整的网格服务支撑平台。它对教育和科研系统中的各种资源进行整合, 屏蔽网格资源的异构性和动态性, 为各种科学计算与工程研究提供高性能的、高可靠的、安全方便的透明网格服务, 形成一套面向 CERNET 的公共网格服务体系。CGSP 提供 ChinaGrid 的服务门户, 同时也为各种网格应用的开发提供一套完整的开发环境。整个 CGSP 平台由多个功能模块构成, 如图 1 所示。

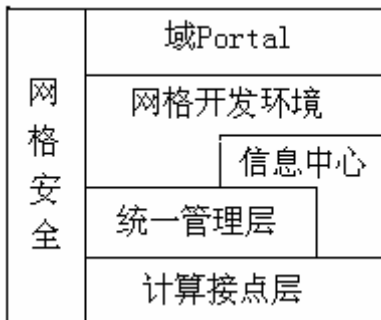


图 1. CGSP 逻辑结构图

CGSP 共划分为如下六个功能模块：

- 网格门户 (Portal) : 作为 ChinaGrid 的网格服务展现方式, 网格门户是最终用户使用网格的入口。通过网格门户, 用户可以提交作业、监视作业运行、管理和传输数据、查询网格资源信息, 同时网格门户还具有用户管理、网格资源使用记账等功能。
- 网格开发环境 : 其主要功能是提供资源网格化封装的工具包和网格构建管理工具包, 以及提供面向网格环境的编程模型, 用于网格作业的开发。
- 信息中心 : 其主要功能是负责网格环境中各类资源的管理, 实现一个全局的资源视图, 提供网格信息服务, 使得最终用户透明访问网格环境上的计算节点、应用程序、以及各类仪器设备等。
- 统一管理层 : 其主要功能是为网格环境中各类作业操作提供基础支持, 具体分为如下四个子模块。
 - ◆ 服务容器 : 其主要功能是在计算节点上, 提供一个网格服务安装、部署、运行、监控的环境, 并为监视计算节点的资源实时状态提供必要的支持。
 - ◆ 数据管理 : 主要功能是负责网格环境中各类存储资源和文件数据的管理, 实现一个全局的文件数据视图, 使得用户透明访问网格环境上的各种数据文件。
 - ◆ 作业管理 : 主要功能是在信息服务和数据管

- ◆理的基础上,为最终用户执行计算任务提供一个虚拟的计算平台,实现对数据和资源的透明访问和分布资源之间的协同工作。
- ◆域管理:主要功能是负责本域的用户管理、日志、记账、以及与其他域之间的交互,使得域管理员可以方便地对本域的用户,服务和资源,域间交互策略等进行管理。
- 网络安全:所提供的功能包括用户身份的认证、资源和服务的授权、以及用户身份到资源授权的映射等。
- 计算节点层:真正提供网格服务的节点,可以是集群等大型计算设备,也是普通微机和工作站等。

CGSP 是一组互相配合的软件组件,支持 ChinaGrid 网络应用的开发、调试、部署、运行管理以及系统监控等各个环节。基于 CGSP,整个 ChinaGrid 网络系统可以构成一个分层树型结构,如图 2 所示。其中,域是指可以作为一个独立的网格对外提供服务的系统,一个域可以是一个专业网格,也可以是一个区域网格;每一个域,不管是全局域、域、还是子域,都有自己的 CGSP Portal、开发层、管理层和信息层等;各个层次的域具有相同的逻辑结构和功能模块。全局域、域、子域用来表明它们之间的父子关系,一个域的用户可以方便地使用以该域为根的子树中所有域的资源,而整个 ChinaGrid 就是由这样的一些域作为节点组成的一颗树。各个域都从认证中心获取用户证书,不同的域可以有不同的认证中心,也可以用同一个认证中心。现阶段,为了减低复杂性,整个 ChinaGrid 采用同一个全局认证中心。

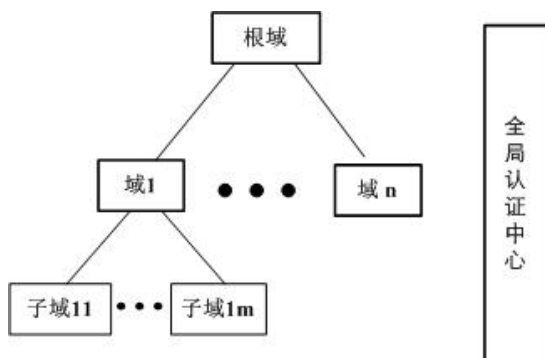


图 2. 基于 CGSP 的 ChinaGrid 树型结构

树型结构中所有域节点的内部逻辑结构都相同,即如图 1 所示,都有自己的域 Portal,网格开发环境,信息中心,统一管理,计算节点层,以及网格安全等 6 个部分组成。不同模块层次间的交互关系可通过他们之间的是否有交界线来表现:安全模块与其他所有模块都相关;网格开发环境与信息层、统一管理存在交互;信息中心与统一管理、节点层有交互关系。

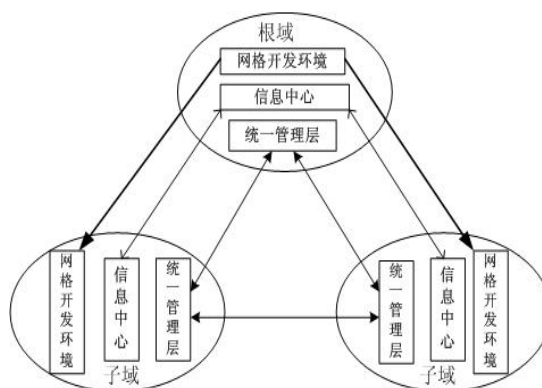


图 3. 基于 CGSP 的 ChinaGrid 域间关系

不同域之间可以互相访问和使用资源,但是他们的交互只发生在网格开发环境、统一管理层和信息中心这三层,其他模块的域间交互都通过上述三层来进行。图 3 表示了树型结构中的一父节点和两个子节点的 CGSP 功能模块之间的调用关系。不同域的统一管理层之间可以互相调用;父域的网格开发环境可以调用子域的网格开发环境;信息中心的调用只存在父节点和子节点间,子节点与子节点之间不存在调用关系,它们之间的交互通过父节点完成。

CGSP 采用上述结构的主要目标在于:

- 适应 ChinaGrid 区域性强的特点,使区域用户尽量使用区域资源;
- 满足 ChinaGrid 的扩展性要求,使其规模逐步扩大到 100 所高校;
- 保证 ChinaGrid 的完整性和统一性,实现全局监控和统一入口;
- 满足目前应用网格的需求,构建应用网格方便、快捷,用户使用简单。

2. CGSP 模块划分与工作流程

2.1 CGSP 软件模块及其形态

CGSP 是一套相互协作的软件组件,和 Microsoft Office 一样,它包含若干个可独立运行的软件系统(用于支持开发过程、运行过程、系统安装过程和系统管理的各个环节)以及一组面向网格应用开发者的应用编程接口(API)和编程模式。根据这些软件系统和 API 面对的不同网格应用阶段,我们将这些软件系统划分为运行态和开发态。

● 运行态软件模块

包括 Portal、作业管理、服务容器、信息服务、域管理器、数据管理、安全管理。这些软件模块经过合理部署后可以协同工作,实现 ChinaGrid 范围的资源共享与协同问题求解。

● 开发态软件模块

包括 CGSP 开发环境提供的各种工具,例如资源包装工具、作业定义工具、Portal 开发工具、并行编程模型及其运行环境(又称专业网格开发工具)、可视化管理/监控程序以及面向远程安装配置管理的 CGSP 安装工具。

图 4 从功能关联的角度描述了上述两类软件模块组织成的 CGSP 模块结构图。图中标识了各软件子系统,即 Portal 子系统、网格开发环境、信息服务、域管理器、作业管理器、服务容器、信息管理系统、数据管理系统等。其中,白色的部分是 CGSP 1.0 版本(2004 年 12 月正式发布)完成的功能模块,灰色的部分表示将在后续工作中进一步关注的问题。下面,我们简单介绍这些主要软件模块的功能定位与主要形态。

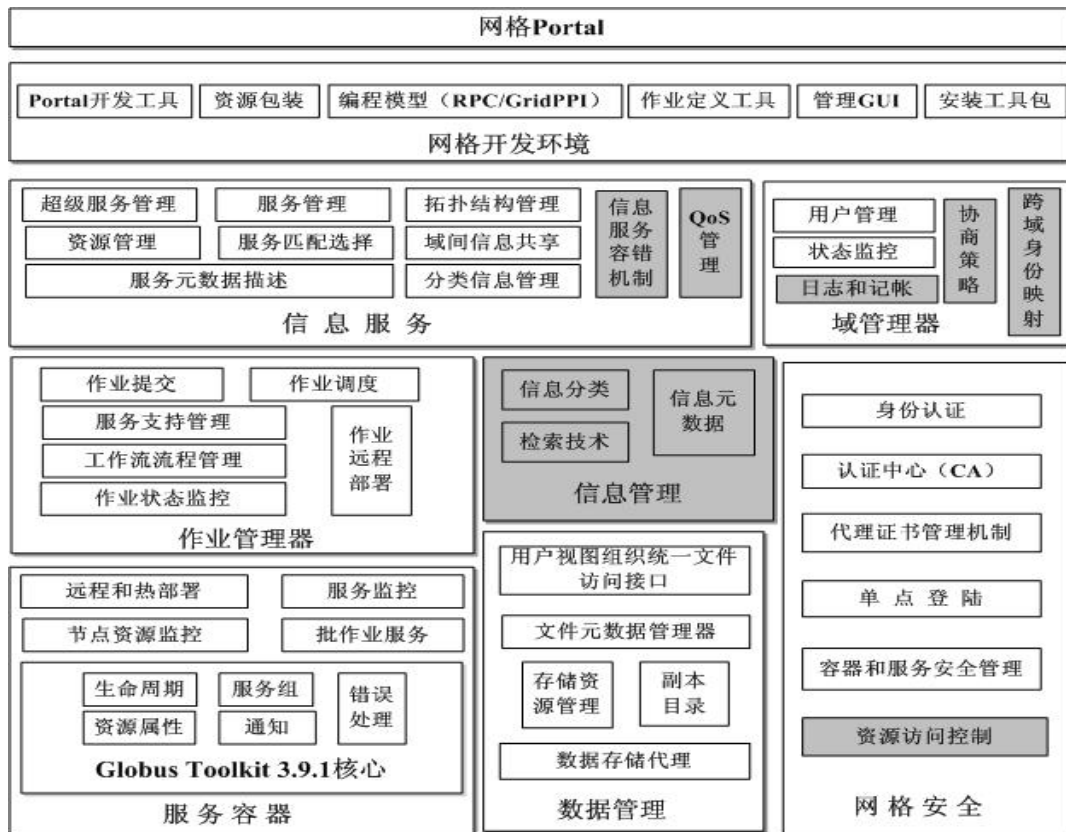


图 4. CGSP 详细模块结构图

2.2 CGSP 软件模块部署方案与典型流程

图 5 是 CGSP 各软件模块的部署方案示意。

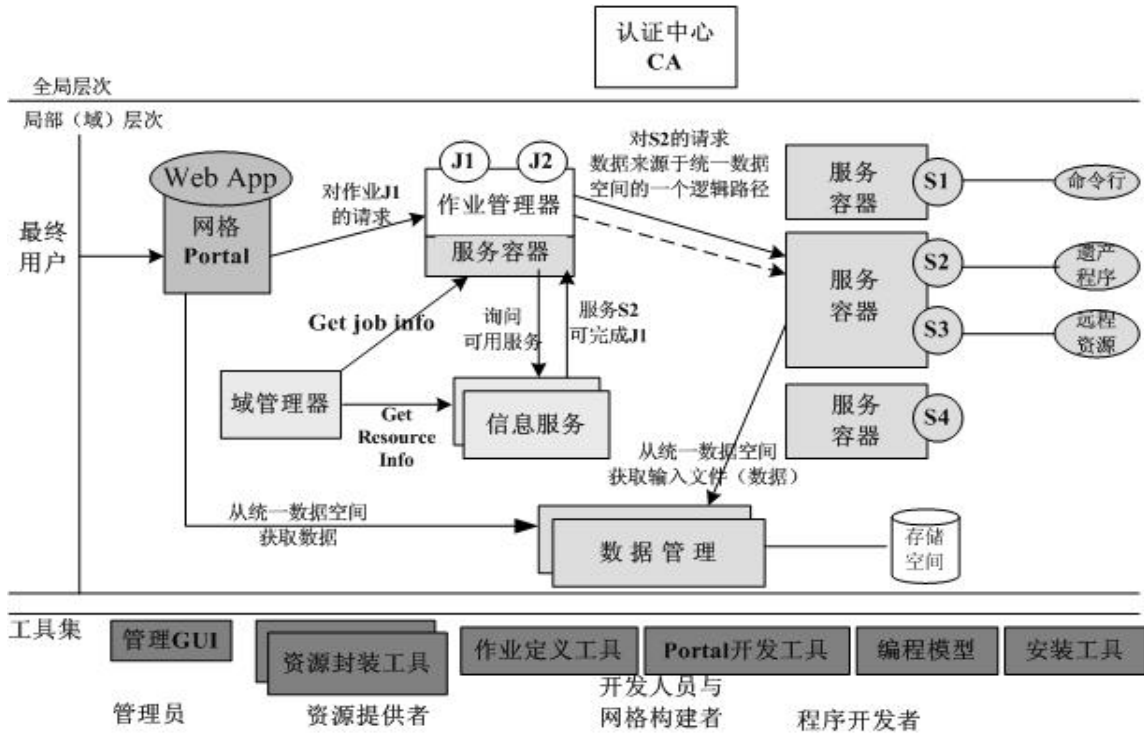


图 5. CGSP 软件模块部署方案

每个网格应用可以划分为三个部分：资源（包括计算资源、存储资源、软件与遗留系统资源、数据资源等）、作业（也称业务流程）和人机交互。在围绕多个网格作业部署 CGSP 的时候，首先，需要在待集成的资源附近部署 CGSP 的网格服务容器，并通过开发工具中提供的资源封装工具针对不同资源进行封装，封装后的服务应当被部署进服务容器（如图 5 中的 S1、S2、S3 等分别封装命令行形式的软件、遗产程序以及远程资源等）。针对存储空间这类特殊的资源，我们通过数据管理系统对其进行封装和抽象，提供逻辑的存储地址空间。其次，我们需要通过作业定义工具将某个领域的典型业务流程建模为用 JDL 描述的作业并部署进作业管理器。由于作业管理器本质上也需要接受以 SOAP 协议传输的作业请求，因此作业管理器需要同服务容器集成部署，由服务容器实现基本的网格

服务处理，由作业管理器完成作业请求处理、服务发现与选取、作业向服务的分配等。第三，每个域内需要部署信息服务，它从服务容器收集服务信息，并向作业管理器等其他软件模块提供支持。第四，可以根据每类专业需求部署一个或若干个网格 Portal，Portal 设计工具可以协助用户生成满足用户交互需求的 Web 应用并部署进 Portal。此外，每一域内还需要部署域管理器，完成域管理功能和用户管理、跨域身份映射的需求。

应用部署后，处理过程如下：

- 1) 用户首先需要注册并获取由 ChinaGrid 统一的 CA 所签发的证书以表示自己的身份。然后用户通过一定的安全协议（如 HTTPS）登录网格 Portal，通过 Portal 中部署的 Web 应用完成同网格系统的人机交互。



模块设计

1. 模块描述

在 ChinaGrid 中存在着大量数据密集型应用, 这些数据密集型计算任务的特点是数据量大, 对数据的传输性能有比较高的要求, 需要可靠、高效的传输机制; 同时, 在通常情况下, 每一类应用有一些固定的存储资源为该应用提供存储服务, 因此要求在进行存储资源分配的时候, 对特定的应用应该能够将数据分配到一组特定的存储资源上, 并且每个应用应该能够很方便地创建一个属于自己的、符合本应用需求的数据逻辑域, 从而使应用的性能达到最高; 另外应用并不关心自己的数据到底存在于系统中的哪个存储资源上, 只需要指定一个逻辑空间中的文件路径, 就能够很方便的取得数据。

根据以上的应用需求, ChinaGrid 数据管理提供以下的功能:

1. 提供高效可靠的数据传输机制: 通过使用多个副本进行并行传输使得对大文件的传输性能得到提高; 通过使用断点续传来保证数据传输的可靠性, 这里断点续传包括两种续传机制, 一种是由于网络的原因, 导致一个传输链路的暂时阻塞, 可以等到网络恢复以后重新恢复传输链路, 继续上一次的传输; 另一种断点续传机制则



(接上页)

1) 交互的过程将产生一个(或一组)对特定类型(如 J1)的作业的请求。Portal 根据用户的交互操作产生请求并提交给作业管理器。
2) 作业管理器根据本地存储的 J1 的作业处理规则(以 JDL 形式描述), 询问信息服务有哪些可用的服务可以帮助完成作业 J1。
3) 基于同服务容器之间进行“Push/Pull”模式获取的信息, 信息服务维护资源视图, 据此回答作业管理器提交的查询, 以一定的排序原则返回一个可用服务列表。在返回服务列表时, 原则上信息服务应尽量返

回本地的可用服务, 如果在本域中找不到满足条件的服务, 则请求其他域的信息服务获得更多可用服务。

4) 作业管理器选取服务 S2, 并向 S2 发送服务请求。

5) 服务容器获得请求后, 如果需要访问数据, 还需要调用数据管理系统提供的访问接口, 以统一数据空间的逻辑路径获取数据, 执行服务, 必要时将处理结果写回统一数据空间的另一个逻辑路径。

6) 作业管理器在作业执行过程中不断收集服务执行状态, 以便满足系统对作业监控的需求。AIG 是在某一存储资源不能提供服务后, 传输链路自动

转到传输文件所在副本的另外一台存储资源上继续进行传输,通过文件分片保证大文件的传输效率。

2. 建立在数据域之上的数据逻辑域:数据域是按照地理位置,将存储资源划分到不同的数据域中,由不同的数据域管理器来进行管理,从而使得数据管理不会因为存储资源的数目增加而导致性能的下降,达到数据管理的可扩展性,并且能够使每个数据域能够进行自治管理,例如每个数据域能够采用不同的策略进行存储资源的分配。同时,实际应用要求每个数据域之间不能是没有联系的,他们之间应该能够进行协作,共同满足某一个应用的存储需求。数据逻辑域这个概念就是为了解决这一问题而提出来的。所谓数据逻辑域,指的是在数据域之上根据某一个特定的应用建立的一个特殊的数据逻辑域,一个数据逻辑域可以包含多个数据域中的存储资源。在这个数据逻辑域中执行的数据任务,只能在创建该域的时候指定的一系列存储资源之上,这样可以保证任务执行的时候,数据不会存放到该数据逻辑域外,从而满足应用在存储性能上的需求。数据逻辑域可以只在一个数据域之内,也可以建立在多个数据域之上。一个数据逻辑域有一个公共空间,每个加入到该数据逻辑域的用户都能在一定的权限控制下访问该公共空间,同时每个用户在该数据逻辑域中又存在着自己的数据空间,用户对该数据空间有完全的访问权限。ChinaGrid 提供创建、修改、删除数据逻辑域的功能,这些功能由应用管理员来执行,从而使该数据逻辑域将数据存放在一些特定的存储资源上,并且有特定的用户群。

3. 实现透明的文件访问机制:对于应用来说,它并不关心数据具体存放在哪个存储资源之上,它所需要知道的只是文件在用户空间中的访问路径,然后按照系统所提供的文件访问机制就可以获得文件了。这种透明的文件访问机制要求系统能够实现不同数据名字空间的映射和转换。对于一个文件来说,它在系统中有一个全局唯一的标识;同时可以有不同的用户有权使用该文件,它可以存在于不同的用户空间之中,在不同的用户空间中,该文件的用户逻辑文件名是不一样的,也就是说,不同的用户可以分别对该文件命名,并且在用户空间中的路径也是不尽相同的;

最后该文件可能存在着多个副本,每个副本存在一个存储资源之上,因此对于该文件来说就有不同的物理访问路径,通过该路径并采用系统的传输协议就可以获得数据了。在这里实际上是形成了三个文件名字空间:用户逻辑文件空间、全局文件空间和副本的物理文件空间,ChinaGrid 要提供在这三个名字空间的映射和转换的功能。一个用户访问自己用户空间的数据,提供的是用户逻辑文件名,系统需要将该逻辑文件名转化为全局唯一的文件标识,然后通过全局唯一标识,为用户选择一个最好的副本,将文件的物理传输文件名返回给用户,使得用户可以通过系统所提供的数据访问形式来获取数据。

4. 提供存储资源的管理功能:存储资源是数据管理中实际存放数据的地方,存储资源的性能直接决定了ChinaGrid 数据管理的性能。存储资源管理主要提供存储资源的运行状态信息监控,这样就可以按照一定的策略为用户分配存储资源,使得数据能够存放在离用户最近的地方,以保证传输性能;同时存储资源管理提供存储资源的错误检测功能:存储资源定期向数据域管理器汇报其运行的状态信息,而数据域管理器则定期检查每一个存储资源运行状态信息的更新时间,如果超过一定的时间,该信息没有被更新,则认为该存储资源状态为不可用,以后的资源分配将不会分配到该存储资源上;若该存储资源在一段时间恢复运行,则该存储资源状态为可用。

软件结构

图 5-1 是数据管理的功能模块图,数据管理是由数据逻辑域代理(Agent),存储资源,数据域管理器,数据逻辑域管理器,统一访问入口(Entry),以及数据客户端组成。数据管理各部分的功能描述如下:

1、**存储资源**:这里的存储资源并不是一个简单的硬盘或者磁盘阵列,而是指安装了由数据管理提供的文件访问模块和信息汇报模块的存储系统。它提供存储空间,负责完成客户端与存储系统之间的数据传输以及不同存储资源之间的数据迁移,同时存储资源提供者负责向数据域管理器注册可用存储资源,以及向数

域管理器定期汇报存储资源的状态信息,如剩余存储空间、CPU 使用率、内存使用率、网络状况等信息,数据域管理器根据存储资源提供者的汇报情况作为存储资源分配的依据,同时存储资源提供者负责处理文件删除任务。

2、**数据域管理器**:数据域管理器负责存储资源的分配,维护数据域的可用存储资源列表,记录每个可用存储资源的可用空间大小,机器性能信息,这些信息可以被数据逻辑域代理获取,返回给用户来判断选择那些存储资源创建一个数据逻辑域,同时,物理资源管理器负责将一个数据逻辑域的存储请求调度到该数据逻辑域对应的存储资源列表中的某个资源中去,如果该数据逻辑域是跨数据域的,那么可以通过先由一个数据域的资源管理器进行资源分配,如果一个数据域不能完全满足的话,将剩余的资源请求转给下一个数据域资源管理器来完成,其中这些本数据域内以及外域分配的信息由发起请求的数据逻辑域管理器负责记录。

3、**数据逻辑域管理器**:数据逻辑域管理器负责维护用户列表,以及每个用户的数据逻辑域列表信息;数据逻辑域还负责组织用户数据空间,用户的文件存在于默认数据逻辑域以及加入的数据逻辑域中,数据逻辑域管理器管理为用户组织这些文件,使得用户能够浏览自己所有的文件。此外,数据逻辑域管理器还负责文件的生命周期管理,每个文件在创建之后都有一个默认的生命周期,用户在创建文件之后可以对其进行更改,当某一个文件的生存时间超过了其生命周期时,数据逻辑域管理器负责将该文件的元数据以及物理数据删除。

4、**数据统一访问入口**:数据统一访问入口组织用户的各个数据逻辑域的视图,即通过查询用户的数据逻辑域列表索引,判断用户的资源请求是与那个数据逻辑域交互,将用户的资源请求交给特定的逻辑域处理。

5、**数据逻辑域代理**:数据逻辑域代理可以获取物理资源信息,管理员可以通过数据逻辑域代理来创建、删除和修改数据逻辑域。

6、**数据客户端**:数据客户端主要提供文件和目录的

上传下载操作,以及浏览目录、更改文件或者目录的属性、创建目录、删除文件及目录等功能,还提供用户空间内部的数据拷贝及移动等功能。

7、**数据移动模块**:数据移动模块用来在不同的存储资源中转移临时数据,是独立于数据管理系统的一个功能模块。数据管理系统不需要记录临时数据的元数据信息,不必对这些数据进行管理。数据移动模块的功能是将数据从一个服务器复制到另外一个服务器(比如在远程部署的时候,需要将服务的 gar 包从一个服务器转移到另外一个服务器),这个过程由服务器之间使用数据移动模块自动完成,不需要与数据管理系统交互。



图 1 数据管理功能模块图

数据管理的使用方式分为两种,一种是管理员使用数据管理,另一种是普通用户使用数据管理。管理员使用数据管理是为了创建和删除数据逻辑域,并为数据逻辑域添加和删除用户和资源;普通用户对数据管理的使用则是对其用户空间中的数据进行操作。下面分别描述两类用户使用数据管理功能的流程。

普通用户访问数据管理使用流程,如图 5-2 所示:

- 1、用户通过 Portal 或者数据管理客户端提交数据请求;
- 2、Portal 或者数据管理客户端将用户请求发至数据访问统一入口;
- 3、数据访问统一入口根据用户请求向数据逻辑域管理器提交具体的数据请求以获得请求数据的元数据信息;
- 4、数据逻辑域管理器请求数据域管理器,请求该文

e-Learning

集群与网格计算湖北省重点实验室 e-learning 组

摘要: 随着信息技术的飞速发展,人们在获取知识的基本方式上发生了翻天覆地的变化。大量的 e-Learning 的系统被设计出来帮助人们从这项技术中获得极大的便利。最近兴起的点对点 (P2P) 技术和网格计算在未来社会将起到关键性的作用。这篇文章就是评估点对点技术和网格技术在 e-Learning 系统中的潜在作用,并介绍一种新型的基于 e-Learning 的 P2P 系统,简称为 APPLE。借鉴了典型的 P2P 网络 Gnutella 的成功经验,APPLE 将提供一种在线直播的服务来共享教育资源。我们也建立一个基于 WSRF.NET 的虚拟课堂服务,它是一个在网格上的所有用户都能访问的一个网格服务。

1. 介绍

在最近几年 e-Learning 已经成为一个热门话题。很多 e-Learning 的平台和系统被开发并商业化。但是这些 e-Learning 系统都是基于 web 方式的。基于 web 方式的课程及其他教育应用在本质上无非就是静态文本页面的网络。这样的模式就是一个信息传递的模式,它主要集中在信息的内容上,它的目标为感兴趣的同学产生更多的静态页面。学习被认为是通过网络在老师和学生之间传递多媒体的信息的一个过程。这种模式再早期的 e-Learning 系统中应用得非常广泛,这并不是说明这种模式很有效果,只是因为使用基本的网络工具就很容易的可以实现,并且不需要传统的参与者。

上述的 e-Learning 模式的不足之处不仅阻碍了它在大众中广泛的采用,而且也不能达到所期望的教学效果。这种模式是一种简单的,单调的教育,不能体现教育的多样性和规模化。人们需要一种 e-Learning 环境中革新的模式,以支持在系统和用户,共享,甚

至在教学材料的再利用方面都有更高水平的交互。

建立一个协作式的 e-Learning 系统是一个基本的想法。知识和信息不仅来源于传授者,还来源于其他有相同兴趣的同学。当一个问题解释被提交给所有相关的同学时,他们能对提问者提供更多建设性的回答,而且他们可以彼此交换信息。越来越多的人试图将点对点计算和 e-Learning 结合起来用于商业,理论和个人。网格技术的发展也提供了 e-Learning 的一个广阔的天地。在网格计算的帮助下,e-Learning 能够从不同的网格服务和计算和存储资源中获利。在 e-Learning 环境中使用网格技术,人们已经开始了一些预备性的前期工作。

现在更多的工作是使用这两种技术的其中之一来设计 e-Learning 系统。但是 P2P 和网格的结合是非常重要的。在 e-Learning 环境中很多不同的应用如消息传递,文件数据传输,在线视频流的传输需要很高的效率。P2P 技术是解决这种问题的一个很好的选择。在一个 e-Learning 的环境中,我们也需要一些稳定和强有力的服务提供诸如教育资源的存储,索引,和信息服务。在这些问题上,使用网格技术可以得到满意的解决。如果我们将这个问题深入到一个哲学层面上,一些要与过程是个人中心的,一些是协调 P2P 和网格特点的团体中心。所以,P2P 和网格技术的结合能够处理 e-Learning 的各种情况,能够使 e-Learning 环境更有力,更高效,更灵活,更多样性。

我们用 Microsoft .NET 开发的这个 e-Learning 系统名字叫做 APPLE。我们使用了 P2P 和网格技术建立了 APPLE。我们在 Gnutella 网络协议上作了修改使它适应我们的系统。我们的平台支持在 internet 上直播课堂。在地理上不同地方的学生不仅可以在他们的桌面上看直播,而且可以和其他学生交流,就好像他们在同一个教室一样。老师可以让同学回答问题,学生的视频图像可以被在这个虚拟课堂上的所有参与者看到。这样我们的直播是基于 P2P 的网络能够比传统的客户-服务器模式的系统提供更多的用户访问。

当用户在同一个网段时,我们使用组播的方式,

这样可以大大减少主机和网络带宽。我们在节点连接它的父亲节点时，我们采用 IP 相近的原则，这样可以使用户得到高效的视频。

WSRF.NET 在 APPLE 系统中也有使用。我们开发了一个虚拟的课堂服务并通过 WSRF.NET 发布。这种方式使我们的 P2P 平台可以和网格连接和整合起来，所以获得了很多方便，一方面我们的基于 P2P 的 e-Learning 环境可以利用网格大量的计算和存储资源来扩展服务的容量；另一方面，网格可以使公众的注意力提供一个可应用的广泛接收和大量网格计算机的应用程序。这篇文章的余下部分组织如下：在第二部分，我们将展示这个体系的体系结构。在第三部分，我们讨论系统的设计。第四部分，我们描述一个实验环境。第五部分，我们介绍 e-Learning 系统相关工作的状态。最后，我们在第六部分得出一些结论，并对我们的未来研究工作做一个展望。

2. 系统的体系结构

一个 e-Learning 的系统由两种重要的应用程序组成部分。第一种是需要一个稳定，可靠的服务器作为硬件平台来使服务在其上运行。另一种应用通常需要很多的精力去注意数据的传输上面，因为使用的用

户量很大。对于第一种服务，网格技术是一种很好的解决办法来提供一个可靠的平台，因为在网格系统中的服务器通常不变的，除了特殊的原因不会被关掉。对于第二种应用，对于服务器来说，可能要求造价很高，因为数据的传输是基于传统的客户机/服务器模式的。这样，P2P 技术对这种问题就是一种很好的解决方案。

为了能够在 e-Learning 系统中很好的利用这两种技术，我们采用了如图 1 的体系结构，它从底层到应用层被分为三个模块：资源层，服务层，网格应用层。

2.1 资源层

在资源层中有两个部分：资源域和资源的虚拟组织。资源域包括物理资源（如 PC，服务器，存储器等等）；逻辑资源（数据库，软件工具等等）通过虚拟的方式组织起来。根据不同资源的特征，这些资源被分成两个部分。一种是稳定的资源，如专用服务器以及相关的软件，这些都是在随时随地都是可以利用的。另一类是可变资源，如客户机和相关的资源，这些都是可以动态的加入和离开 e-Learning 系统的。

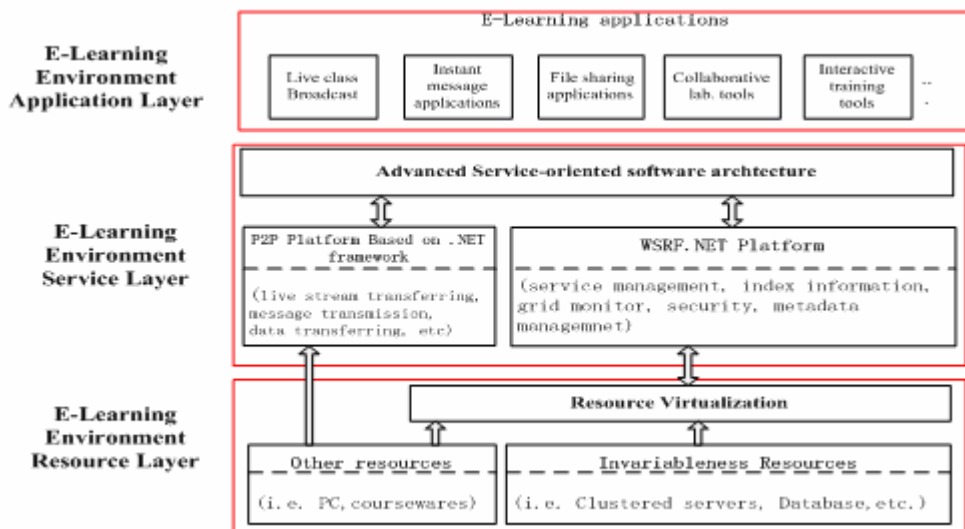


图 1 基于 WSRF.NET 的 e-Learning 环境的体系结构

资源服务环境提供了在一个单个管理域中支持

本地设备的一组资源，如一个 J2EE 应用服务器，

Microsoft.NET 系统, 和其他 Linux 集群系统。

2.2 服务层

服务层是建立在资源层之上的。一系列的管理工具和协议规范在这里被定义来实现在资源层中所有资源的共享, 整合和协作。同时, 对资源和服务的透明和一致的访问接口是必须提供以便支持应用层的应用程序。服务层由三个基本的组件构成: WSRF.NET 组件, .NET 框架, 基于 P2P 的平台, 和比较高级的服务导向软件结构。

第一, Internet 技术的最近的发展标志着应用网络技术和通过 web 服务的服务导向的 Internet 使资源共享技术发展的趋势。Web 服务资源框架(WSRF)是一组相关的 Web 服务规范来定义 Web 服务资源的表现形式的方法。这些规范允许开发者声明和实现一个 Web 服务和一个或多个有状态资源的联系。

WSRF.NET 是一个基于 Microsoft.NET 框架, 独立的完全适应 WSRF 规范的开发环境。它提供了一

个容器框架可以在 .NET 环境下实现适应 WSRF 的网格计算。它提供了对基于属性的开发的工具和支持, 通过把它变成元数据来把一个服务变成网格服务。WSRF.NET 也包含了类库来实现一些服务和客户端需要的基本功能。

第二, 在线直播流, 消息, 文件数据需要高效的传输。现有的 P2P 系统, 如 Gnutella 和 Kazza, 提供了一个解决在 e-Learning 环境下这样问题的 P2P 计算技术的有力证据。这项技术提供了一个高效的传输方法。这样, 我们在服务层设计了一个 P2P 平台的模块。

最后, 一个服务导向的结构实现了一个相应的原形框架来支持应用的处理和分析。在我们的工作中, 由 WSRF.NET 平台提供这部分的功能。在基于 .NET 框架的点对点平台中, 我们处理网格平台和应用层中的问题, 定位网络的调度, 管理多种应用程序的问题。在这个层面里, 我们研究和设计了在整个平台中在 WSRF.NET 之上的一系列应用, 数据分发的方法, 任务的分解和相关的核心工作。AG



1. 产品介绍

Service Anywhere 平台是服务于手机用户和各种商业服务提供商的综合的服务平台, 它使得手机用户可以

集群与网格计算湖北省重点实验室移动项目组通过在 Service Anywhere 平台上的注册和服务定制获得随时随地的各种服务, Service Anywhere 平台向手机用户提供的服务是基于 GIS 和用户希望的最优化服务, 集天时, 地利, 人和于一身。同时向各种商业服务商提供服务资源的注册和服务信息的发布。

这相当于为商业服务商开拓了全新的市场。通过 Service Anywhere 平台，服务提供商的各种服务可以根据地域漫布空间。

2. 产品的应用领域

我们提供面向移动服务商和面向移动服务消费者的两种服务

面向移动服务商：

我们将向各种服务商提供服务资源的注册和服务信息的发布，结合其具体的地理位置将服务资源和地理信息结合起来，我们同时向服务商提供两种方式的信息发布，一种是提供服务商服务资源整合注册，这种方式以非托管的形式提供资源的查找，定位，从而通过我们的平台将服务的消费者和服务的提供者联系起来。另一种是服务商服务简单服务的发布，这种方式是以托管的形式提供服务的使用，具体是服务商通过我们的平台发布自己的一些服务，然后有我们的平台对这些服务进行统一的调度和管理。

面向移动服务消费者：

我们向移动服务消费者提供了服务的个性化的定制，和服务资源的查找，以及基于地理信息系统的

服务资源的向导，从最大程度上满足服务消费者的需求。

我们的平台实际上是提供了一个从服务商到移动服务消费者的双向的桥梁，我们不仅将各种服务商的服务信息提供给消费者，同时我们将消费者的各种服务请求告诉的相应的服务提供商。

3. 产品的创作思路

本平台的创作思路来源于今年来移动服务市场的空前增长的需求，以及各个服务提供商在随之产生的又一个宣传媒介上进行宣传的需求，本平台的创作的思路是，我们提供一个共用的平台，服务商可以在上面发布他们所提供的服务，这样我们的平台可以通过电信网络将其服务宣传出去，而同时我们为手机用户提供了一种基于位置的查询和享受服务，这样我们的平台就在服务商和手机用户之间架设了一座桥梁，将这两者联系起来，之前也有一些类似的产品可以实现将服务提供商的信息提供给手机用户，但是他们提供的只是一种静态的信息，手机用户不能之间得到相关的服务，例如：饭店的订餐，火车订票等，而我们的平台将这些服务全部通过服务提供商的注册展现给手机用户了。

4. 对实际问题的场景描述



图 1 ServiceAnywhere 场景描述

5. 体系结构设计和 UDDI 注册中心的使用

本系统是一个服务平台，它向两类用户提供服务，一类是服务提供商，我们的平台提供的服务是服务提供商可以将他们的服务发布到我们的平台，以供第二类用户也就是手机用户来查询和享受服务。同时我们的服务是和服务所在的地理位置相结合的。

我们需要一个为服务提供商提供注册他们的详细服务信息的模块，自然而然就想到了使用 UDDI 注册中心，当然我们这是个小型的实现，并不能达到真正的 UDDI 那样实现联合分布的查询，只是一个单独的孤立的专用的注册中心。

为了实现这个注册中心，我们选择了使用开源项目——jUDDI 的开发包来进行开发。

但首先要先介绍一下，什么是 UDDI 注册中心。

6. UDDI 标准介绍

UDDI 是 **Universal Description, Discovery and Integration** 的缩写，目的是为电子商务建立标准；UDDI 是一套基于 Web 的、分布式的、为 Web Service 提供的、信息注册中心的实现标准规范，同时也包含一组使企业能将自身提供的 Web Service 注册，以使别的企业能够发现的访问协议的实现标准。通过 UDDI，企业可以注册、发布，以及寻找相应的 Web Services。

(1) UDDI 的信息类型

UDDI 注册中心使用的核心信息模型是由 XML Schema 定义的。UDDI XML Schema 定义了四种主要的信息类型，它们是技术人员在需要使用合作伙伴所提供的 Web 服务时必须了解的技术信息。它们是：商业实体信息 (businessEntity 结构)、服务信息 (businessService 结构)、绑定信息 (bindingTemplate 结构) 和技术规范信息 (tModel 结构)。在 UDDI 的数据

模型中，tModel 是一个很特殊的数据项。tModel 描述了一切技术信息，tModel 的全体组成了 UDDI 中的所有技术注册信息。

◆ 商业实体信息：businessEntity 元素

在商业领域内，合作伙伴和潜在的合作伙伴都期望能准确地定位到商业实体所能提供的服务或产品的相关信息，并把这些信息作为了解你们企业的开始。而在技术领域，技术人员、程序员或应用程序都期望能知道他们需要集成的商业实体的名称和一些关键性的标识，以及该商业实体是属于那个具体工业分类之类的分类信息，以及联络方法 (包括 Email、电话、URL) 等。支持对 UDDI 商业注册的商业信息发布和发现的核心 XML 元素都包含在 “businessEntity” 结构中。这个结构是商业实体专属信息集的最高层的数据容器，位于整个信息结构的最上层。

◆ 商业服务信息：businessService 元素

businessService 结构将一系列有关商业流程或分类目录的 Web 服务的描述组合到一起。businessService 和下面要提到的 bindingTemplate 一起构成了 “绿页” 信息。其中，一个可能的商业流程的例子是一组相关的 Web 服务信息，包括采购服务、运输服务和其它的高层商业流程。这些服务都将是提供这些商业流程服务的商业实体所需要注册的 Web 服务。

这些 businessService 的信息集合可以再次加以分类，使 Web 应用服务的描述可以按不同的行业、产品、服务类型或是地域划分来进行。分类的方法的机制与 businessEntity 是类似的。

◆ 技术绑定信息：bindingTemplate 元素

对于每一个 businessService，存在一个或多个 Web 服务的技术描述 bindingTemplate。这些技术描述包括应用程序连接远程 Web 服务并与其通讯所必须的信息。这些信息包括 Web 应用服务的地址、应用服务宿主和调用服务前必须调用的附加应用服务等。另外，通过附加的特性还可以实现一些复杂的路由选择，诸如负载均衡等。

◆ 元技术信息：tModel 元素

调用一个服务所需要的信息是在 bindingTemplate 的结构中定义的,这在前面一节中已经阐述了。不过一般来说,仅知道 Web 服务所在的地址是不够的。例如,如果我知道我的合作伙伴提供一个 Web 服务来让我下订单,同时也知道这个服务的 URL,不过如果不知道一些具体的信息,如订单的具体格式,应该使用的协议,需要采用的安全机制,调用返回的响应格式等,那样的话,通过 Web 服务将两个系统集成起来仍然是非常困难的。

当一个程序或是程序员需要调用某个特定的 Web 服务时,必须根据应用要求得到了足够充分的调用规范等相关信息,以使调用被正确地执行。因此,每一个 bindingTemplate 元素都包含一个特殊的元素,该元素包含了一个列表,列表的每个子元素分别是一个调用规范的引用。这些引用作为一个标识符的杂凑集合,组成了类似指纹的技术标识,用来查找、识别实现了给定行为或编程接口的 Web 服务。

(2) UDDI 的 API

程序员 API

Uddi 规范包括了 web 服务的接口定义,使得能通过编程实现对 UDDI 注册中心的信息访问。程序员的 API 规范文档详细定义了程序员 API。API 分为两类:查询 API 和发布 API。

查询 API 包含两类调用,使程序能快速地定位候选商业实体、Web 服务及其调用规范,然后在最初调用获得的初始信息的基础上,获得进一步的相关信息的细节。这类以 find_xx 命名的 API 提供了多种搜索标准,从而能对注册中心中的数据进行搜索。发布 API 包括四个 save_xx 函数和四个 delete_xx 函数,每个对应于一个 UDDI 主要结构(businessEntity, binsinessService, bindingTemplate, tModel)。

jUDDI 介绍

jUDDI 是一个开源的 UDDI 服务器软件,采用 java 语言编写,jUDDI 的主页是:

<http://ws.apache.org/juddi/>。目前 jUDDI 的最新版本是

2004.9.2 号发布的 0.9rc3。

jUDDI 的特点:

- 1 开源:你可以查看它的代码。这在我们通过 jsp 构建注册中心的时候,尤其重要。
- 2 平台独立:你可以部署在 win 或者 Linux 下,我们是部署在 RH 7.3 上。
- 3 支持 JDK1.3.1 以及后继版本;
- 4 实现了 UDDI 2.0 版本;
- 5 后台的数据库没有特定的要求,只要支持 ANSI 标准的 SQL 语言操作的关系型数据库系统即可。我们采用的是 IBM 的 DB2 V8.1 for linux 版本。
- 6 可以部署在任何一个 java 应用程序服务器上,只有这个服务器支持 Servlet2.3 标注。主要有 Jakarta Tomcat, JonAS, WebSphere, WebLogic, Borland Enterprise Service, JRun 等。我们采用的是 IBM 的 WebSphere Application Server for linux 5.0
- 7 jUDDI 注册支持集群式的分布配置;
- 8 可以很方便的和现有的认证系统相结合。做到对注册 web 服务的用户的认证。

7. jUDDI 的安装和配置

下面讲讲我们的安装和配置过程,使大家对 jUDDI 的安装过程有个了解。

1. 准备条件:

安装 WebSphere Application Server 和 DB2 数据库系统。在安装 DB2 时,

我们建议一个名为 db2 的实例,并且访问这个实例的帐号为 db2inst1,密码设置为 ibm。

2. 安装配置数据库:

由于 UDDI 将用户注册的 web services 的信息都保存在数据库中,因此我们需要对数据库进行配置,包括安装驱动程序以及建立相应的表。在下载 juddi-0.9rc3-src 的代码中,有一个名为 sql 的目录,该目录保存的是建立相应的数据库表的 sql 语句。由于我们使用的后台数据库是 DB2,因此我们使用 DB2 目录下的 create_database.sql 文件和

insert_publishers.sql 两个文件来建立 jUDDI 需要的数据表格。操作如下：

- ◆ a, 运行一个 db2 的实例,该实例访问的帐号为 db2inst1,密码为 ibm;
- ◆ b, 在该实例中建立一个 uddi 的数据库 db2 => CREATE DATABASE uddi

并且连接该数据库：db2 => connect to uddi

- ◆ c, 然后退出 DB2 命令行: db2=>quit
- ◆ d, 运行 sql 脚本, 建立所需要的表格和数据 #db2 -vtf juddi-0.9rc3-src/sql/db2/create_database.sql

#db2 -vtf juddi-0.9rc3-src/sql/db2/insert_publish.sql
另外由于 jUDDI 需要以 JDBC 方式访问数据库, 因此需要安装相应的驱动程序。将访问 DB2 的 JDBC 驱动程序拷贝到 WebSphere 下的 lib 目录。

3. 编译 jUDDI:

解压我们下载的 juddi-0.9rc3 的压缩包, 需要我们对它进行编译, 生成所需要的 juddi.jar 和 juddi-proxy.jar 文件。不过在编译 jUDDI 之前, 需要我们安装一些软件和 jar 包。下面给出 jUDDI 需要的软件包以及下载地址。

- jakarta-ant from <http://jakarta.apache.org/ant/>
- forrest from <http://xml.apache.org/forrest/>
- axis from <http://ws.apache.org/axis/>
- commons-logging from <http://jakarta.apache.org/commons/logging.html>
- servlet from <http://jakarta.apache.org/tomcat/>
- jdbc-2.0-stdext from <http://jakarta.apache.org/tomcat/>
- log4j from <http://jakarta.apache.org/log4j>

在源代码目录下新建一个 lib 目录, 将这些需要的 jar 包拷贝 lib/目录。接下来在进入源代码的根目录, 然后命令行模式下输入命令 ant jar 进行编译。编译完成之后, 会多出一个 build 目录。该目录下放着我们需要的 juddi.jar 和 juddi-proxy.jar。

4. 安装 jUDDI 和配置

将编译好的 juddi.jar 和 juddi-proxy.jar 拷贝到源代码下的 webapp/WEB-INF/lib 目录。然后将 webapp 部署为 WebSphere 的一个应用程序。接下来我们需

要根据特定的环境来修改

juddi.properties 文件, 包括一些初始化。首先, 对于 juddi.properties 文件里的 jUDDI-managed ConnectionPool properties 的属性进行配置。我们的配置如下:

```
juddi.useConnectionPool=true
juddi.jdbcDriver=com.ibm.db2.jcc.DB2Driver
juddi.jdbcURL=jdbc:db2://127.0.0.1:50000/UDDI
juddi.jdbcUser=db2inst1
juddi.jdbcPassword=ibm
juddi.jdbcMaxActive=10
juddi.jdbcMaxIdle=1
```

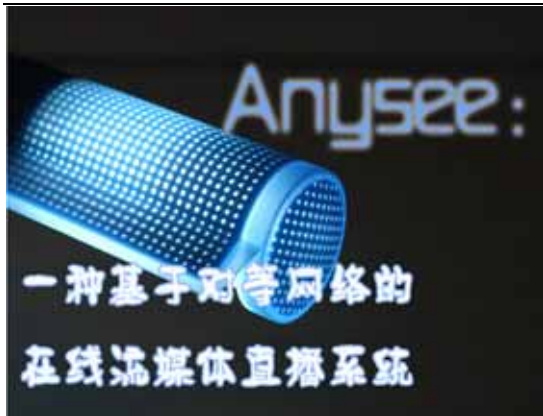
jUDDI 根据这里配置的账号和密码去访问我们的 DB2 数据库数据。另外根据你选择的认证的情况, 可能需要进行一些相关的配置。我们的认证体系是 juddi.auth

org.apache.juddi.auth.DefaultAuthenticator, 因此我们通过 juddi.users 来指定包含用户的 xml 文件的位置, 在我们的系统中, juddi.users= /opt/WebSphere/AppServer/installedApps/lcgrb/juddi.e ar/wholenew.war/WEB-INF/ juddi-users.xml.

juddi-users.xml 文件结构非常简单, 如下:

```
<?xml version="1.0" encoding="UTF-8"?>
<juddi-users>
  <user userid="yeqiwei" password="yeqiwei" />
  <user userid="jdoe" password="ibm" />
  <user userid="juddi" password="password" />
  <user userid="ibm" password="ibm" />
</juddi-users>
```

指定用户名和密码即可, 这个用户名和密码是登陆 uddi 注册中心来注册和发布 web 服务所需要的。接下来我们可以通过查看 jUDDI 自带的测试页面来查看我们的 jUDDI 是否安装成功了。通过在浏览器中打开 jUDDI 自带的 happyjuddi.jsp 来查看是否安装成功。



集群与网格计算湖北省重点实验室 p2p 项目组

近年来,随着 Peer-to-Peer (P2P) 理念的普及,越来越多的 P2P 应用软件问世,他们的出现给整个互联网的发展带来了极大的影响;在诸如文件共享、流媒体服务等领域中。资源流动比原来更加的方便和快速。“Peer”英文意为“同等的人,伙伴”,所谓 P2P,就是“端对端”或者“点对点”,P2P 的实质即代表了信息和服务在一个个人或设备与另一个个人或对等设备间的流动。任何一台 PC 机可以不通过任何的中央节点而直接与需要通信的节点进行交互。它最大的优势在于克服了传统客户机服务器模式中中央服务器的带宽和性能限制,扩大了整个系统的可扩展性,体现了一种“我为人人,人人为我”的理念,使网络重新回到了 Internet 的本质,即让共享与自由的精神充满网络世界。通过 P2P 网络中的各个节点的合作,可以有效的利用分散的计算和存储资源来完成特定的任务和服务,时下最为热点的应用体现为文件共享和流媒体服务两种。

AnySee 就是这样一个基于 Peer-to-Peer 模式的视频直播系统软件。它通过自身 P2P 的体系结构很好的解决了当前教育网内网络电视服务器带宽难以服务众多用户的问题,使更多的用户可以观看和发布网络电视频道,为广大网友提供一种观看网络电视的全新且方便的途径。通过 Anysee,在任何一台普通的 PC 机上都可以发布任何视频和音频信息比如直播电视,电影,个人 DV 作品,mp3 音乐等,并且也能够收看或收听已发布的任意一个节目源。AnySee 采

用了应用层上的多播技术,以 P2P 系统构架,结合局部的调度策略、完善的容错机制、缓冲区的高效管理机制,保证了整个系统的服务质量,最为重要的是,不会出现因为收看同一个节目源人数过多而无法满足服务的情况,即不存在服务的上限。

下面,我们首先介绍一下该系统的总体框架图;然后介绍一下关键技术;最后介绍一下她的应用情况。

1. Anysee 系统架构

Anysee 的系统结构如下图所示。

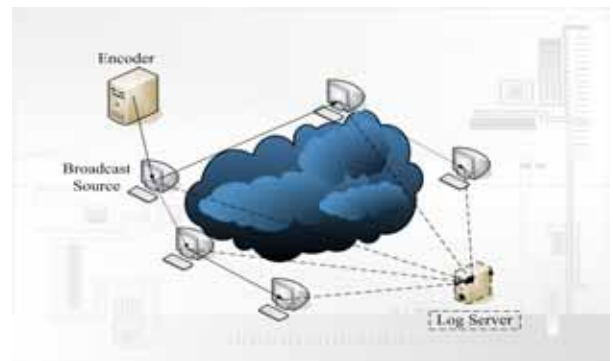


图 1 Anysee 系统结构图

上图为 Anysee 整体结构图。大体上包括 3 个部分:

第一部分:Channel Server(频道服务器):作为节目的提供者,又包括 Encoder(编码器), Broadcast Source(广播源)两部分;

第二部分:Log Server(节目列表服务器):以 WEB 的形式列出正在播放的节目;

第三部分:Anysee Client(客户端软件):接受播放数据。所有参与到网络中的节点运行该客户端;既充当服务接受者,也充当服务转发服务者,充分体现了对等计算的精神。

首先,编码器对实时节目进行编码。编码器采用普遍使用的 windows media player,编码后的文件格式为 ASF。编码后的数据流实时传送给 Broadcast Source Server。

Broadcast Source Server 建立节目的时候向 Log Server 注册,将自己的节目信息传送给 Log Server。

所有的 Anysee Client 自动的从 Log Server 上取得实时的节目列表。选择节目后,连接到提供该节目的服务器上。Broadcast Source Server 将所获得的数据留提供给跟自己直接连结的 Anysee Client。

Anysee 的网络成树状结构。当有结点加入的时候,服务结点根据一定的调度规则将请求与自己连接的结点或者是同意连接,或者是重新将其定位到其他结点上。因此,一个 Anysee 网络上的结点既是消费者又是服务者。他在收看节目的同时又为其他人提供了数据服务。这种特殊的结构降低了服务器的负荷,提高了整个网络的可接入性。

2. 关键技术

AnySee 的关键技术主要体现在其广播端 (Broadcast) 技术。

AnySee 软件分为广播端和客户端,广播端用来将输入的视频或音频信息直播,作为整个 AnySee 网络的唯一数据来源。目前的 AnySee 的广播端 (Broadcast) 直播来源方式有两种: asf、wma、wmv、mp3、avi、nsv 格式的媒体文件;由 Windows media encoder、Shoutcast server 软件输出的实时视频流和音频流。另外,AnySee 的广播端也有其独特的技术特点。

关键技术之一就是基于地域的近播调度策略。由于流媒体的服务质量需要足够而且稳定的带宽做为保障,所以怎样保证相连节点之间的最好的带宽条件是流媒体服务软件的一个挑战。AnySee 的广播端使用的是一种基于地域调度的近播调度策略,具体是将物理位置一致的节点尽量调度到一起。因为同一个地域的网络连接,不需要太多的路由器中转,基本上可以视作带宽情况良好。比如如果 A 点在武汉, B 点也在武汉,那么尽量将 AB 两点调度成直接相连的状态,这样 AB 直接的带宽条件就保证了。基于这种策略的考虑,AnySee 的广播端为不同的地域留出相应数目的接口,如华北地区、华中地区、华南地区等等,每一个新加入的节点都将被调度到不同的地域接口中,接着再根据一种 landmark 机制,尽可能的细分调度到各个省、市、区、学校、校区。所以在 AnySee

的邻居里面经常可以看到服务者和消费者的 IP 地址与自己的很相近。

关键技术之二就是容错机制。AnySee 的基本结构是一棵应用层上的多播树,其中的某一个节点正常或异常离开都会造成对下面相连节点的影响。怎样将这种影响降到最低,是 AnySee 的另外一个挑战。目前,AnySee 使用的是一种有效的容错机制,每个节点记录了两层的节点信息,即其直接服务者及其服务者的上层节点信息,这样当其直接服务者失效的时候(无论是因为正常离开还是异常离开),都会找到新的服务者。而在这个过程中,由于缓冲区的有效管理,在客户端的观看或者收听不会有任何影响,与此同时,该节点所服务的所有节点也因此不会收到任何的影响。这就是为什么在使用 AnySee 的时候,AnySee 的邻居信息中显示直接服务者的地址变化,但是接收到的图像和声音依然很流畅的原因。这一点 AnySee 的用户应该会有比较直观的认识。

关键技术之三就是缓冲区的管理。Buffer 的管理是非常必要的。主要是有三个原因,即:保证媒体服务的软实时特征,减少网络的抖动以及保证 p2p 系统的高动态性不会影响系统总体性能。

AnySee 的 Buffer 管理策略可以概括如下:每个 peer 都拥有一个 buffer,以时间间隔为单位进行管理; Buffer 缓存最近固定时间长度内的媒体数据; Buffer 占用的空间大小随时间呈动态变化;头部元数据包和媒体数据包分别进行管理。

关键技术之四就是信息的广播。为了加强广播端的广播通告功能,比如告知所有的用户该节目的开始时刻和结束时刻等等。AnySee 的广播端使用了一种广播包的数据包,它将被传送到每个正在使用 AnySee 的用户,显示广播端要广播的信息。

3. 应用情况

1、2004 年 8 月雅典奥运会期间

AnySee 客户端提供主页下载和 www.5qzone.net 网站 BT 下载,并在武汉白云黄鹤 BBS 站发布信息。5qzone 官方种子下载量达到 9 千以上,加上 AnySee 主页和其他网友发布的种子,下载量远超过 1 万。图

2与图3展示了在线并发节点分布情况和加入节点分布情况。

到奥运会结束时, AnySee 联合华中科技大学校内门户媒体网站, 如华中大在线网站、醉晚网站等, 提供了3、4个稳定节目源供整个教育网用户收看, 软件也得到了第一次广泛的测试。期间近20日长期提供服务, 同时在线人数峰值达到4至5千, 使用AnySee收看在线直播累计人数达到数万, 基本满足了以华中科技大学为主的广大教育网用户在电脑上收看奥运节目的需要。AnySee提供的服务也得到大部分用户的好评, 不少用户在AnySee网页上给予开发小组高度的赞扬、真挚的鼓励和殷切的期望, 也反馈了不少使用时的问题和意见。白云黄鹤BBS站的奥运版、华中大在线版、网络资源版、研究生版等版面都有关于AnySee的热烈讨论, 不少网友作为使用者和受益者主动为AnySee作宣传甚至解答其他网友的问题, 让开发小组十分感动, 也增加了相当的信心。

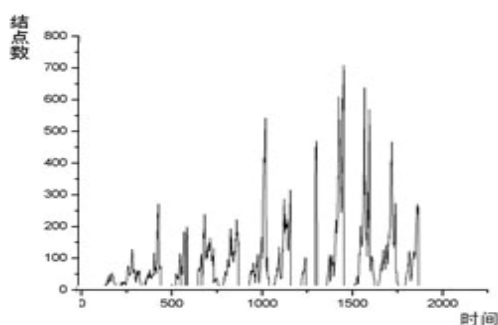


图2 一个多播树中结点数

2、GCC会议

2004国际网络与协同计算大会(GCC)于2004年10月24日至27日在华中科技大学顺利举行, 参与会议的有Ian Foster, Charlie Carlett及包括我校金海教授等在内国内外网络方面的专家。Anysee同步提供每日Keynote Speaker演讲的直播, 并提供10月20日至22日在武汉举行的2004国际网络与并行计算会议(NPC)录播, 为学术同好及时了解会议和国际动向提供了一定的帮助。

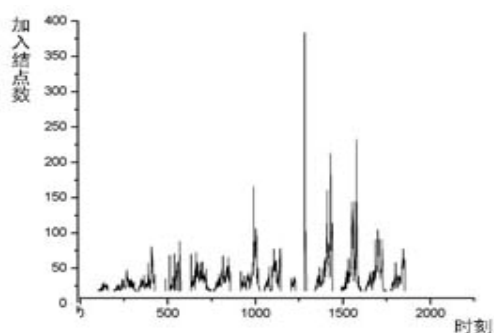


图3 一个多播树加入结点数

3、与华中大在线合作

作为华中科技大学的知名网站, 华中大在线网站(www.hustonline.net)对AnySee对视频在线直播的作用很感兴趣, 在和开发小组接触后双方愉快地展开了合作。现在华中大在线完全取消了原有地网页链接在线直播地模式, 整合了AnySee软件广播端为AnySee用户提供了两个稳定的在线直播节目源。考虑到AnySee的初衷和在校大学生的喜好, 其中一个固定播放中央电视台体育频道节目, 方便了不少希望能在电脑上观看体育比赛的学生用户。如每周的欧洲足球联赛、NBA球赛等, 都可以很方便的在寝室观看, 不用像原来一样辛苦地跑到校外观看电视。

此外, 经过改进, 该系统还在于武汉市的电子政务网络中为武汉市896家社区提供了稳定的多媒体服务。

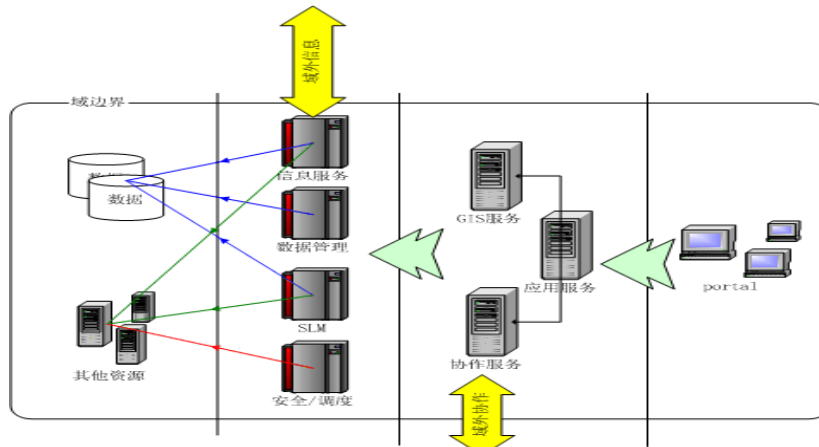
在流媒体服务领域, 本研究小组已有了多年技术积累, 形成了以集群视频服务器、Anysee网络在线直播系统、HoneyComb网络点播系统为代表的一系列原型系统。其中集群视频服务器还在浪潮公司北京方案解决中心进行了实测, 获得了好评。同时, 本小组还在流媒体与分布式领域发表了包括著名国际期刊杂志、著名国际会议在内的三十余篇高档次学术论文。我们也欢迎广大热衷于P2P流媒体服务的研究者参与进来一起努力, 共同做好流媒体方面的研究; 更欢迎有实力的公司与我们合作, 一起为把科技成果转化为市场产品而努力。AG

信息服务网格项目组简介

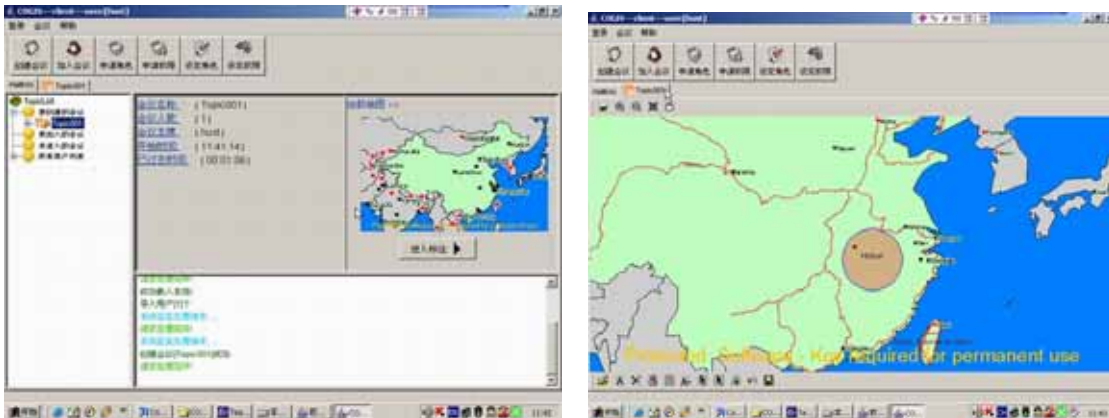
信息服务网格旨在研究发展以服务为中心、能满足实时、容错、安全等用户需求的网格平台及应用服务。通过结合 Web services 和网格理论,参考国际上提出的开放式网格服务体系,开发以服务为中心的网格平台。旨在广域网范围内提供资源共享,协作处理能力,并能满足实时、容错、安全等用户服务质量需求。随着信息化、网络化要求的提高,人们需要在地理位置分布的情况下协同完成同一个任务,而不同组织通过广域网在异地实时、协同工作成了一个迫切的需求。

本项目组承担的研发任务主要的应用背景有:

附图:



以下是该软件部分截图:





图像处理具有数据量大、处理过程复杂、计算密集等特点,许多应用都需要高性能计算资源,甚至并行计算环境的支持。广域范围内大量闲置的高性能并行计算机、集群和 workstation 可以提供这方面的资源和服务。但图像处理研究人员大多不是计算机专业出身,他们对计算机、分布式和并行等技术缺乏了解,导致他们对使用高性能计算机存在一定的困难,同时地理上的分散也制约了图像处理应用和高性能计算的整合。网格技术能较好的解决这些矛盾。



图 1 ImageGrid 图像处理网格平台

ImageGrid 图像处理网格平台(如图 1 所示)运用网格聚集计算能力的特点将所有资源组织起来,用以解决图像处理过程中遇到的各种高性能计算问题。ImageGrid 利用网格平台屏蔽网格资源分布和异构特性,利用应用中间件屏蔽内部复杂的实现细节,利用 Web Portal 面向用户提供友好、方便、易用的操作环

境,大大降低了研究人员使用网格资源的难度。同时,还面向图像应用提供一系列工具集和基础服务,采用服务组件重组技术聚合网格环境中庞大的计算资源解决大规模图像处理问题,利用 workflow 引擎分散和简化应用逻辑,减少用户的中间介入,提高服务可重用性和服务性能,利用集群的并行计算环境和网格多点协同处理机制高效解决各类图像处理应用,提高服务协同工作的效率;提供应用服务级的全局资源视图方便用户随时对整个网格进行信息查询与监视,等等。

1. ImageGrid 总体结构

图像网格原型系统在网格基础平台上实现图像处理应用的网格化,系统总体结构设计如图 2 所示,分为网格资源、网格基础平台、图像网格应用中间件和图像网格交互界面。



图 2 图像处理网格总体结构

网格资源是图像网格系统的硬件集成,这些资源在逻辑上是孤立的,只有将其封装成网格服务,通过

网格基础平台实现广域计算资源的有效共享。网格基础平台屏蔽网格资源的分布和异构特性,向图像网格应用中间件提供网格计算的核心服务。图像网格应用中间件提供图像处理应用支持,如图像处理集成开发环境、并行任务协同处理、应用信息监测和远程图像显示等。图像网格 Portal 面向普通用户提供基于 Web 的应用服务,用户可以方便的在图像网格 Portal 上完成一系列注册、处理、查询和管理等工作。

2. ImageGrid 关键技术

图像处理编程环境

图像处理编程环境面向广大网格用户提供友好的图像处理需求提交与问题解决环境,通过组件重组完成复杂多样、甚至单个服务组件无法满足的应用需求。根据服务组件功能和接口的不同将所有网格服务分类,每一类都被封装成“虚拟服务”,对外提供统一的标准接口。每个服务组件与具体的网格资源和服务句柄对应,称为“物理服务”。虚拟服务屏蔽了服务底层实现细节,降低了用户使用网格的难度,而物理服务在通用服务请求代理进行服务调用时用到。信息服务模块搜集所有服务组件信息,贯穿整个图像处理运行过程,如图 3 所示。

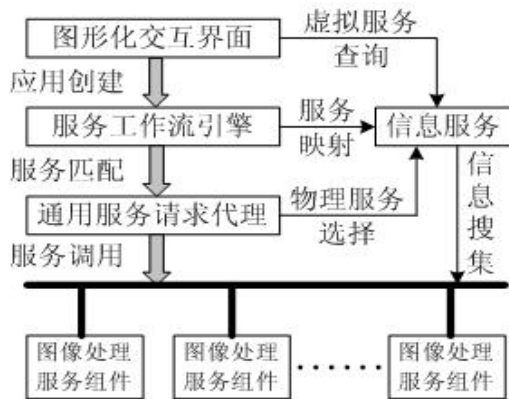


图 3 图像处理编程环境运行图

在该编程环境中,用户不需要关注纷繁的物理细节与编程技巧,只需通过信息服务查询平台上的虚拟服务,简单配置服务之间的上下文关系,构造出逻辑的图像处理应用(如图 4 所示)。用户交互界面根据生成的逻辑应用将需求转换成平台识别的工作流语

言,驱动服务工作流引擎执行应用。服务工作流引擎解析工作流语言,根据服务之间的映射关系和服务上下文做出判断,进行服务的查找和匹配,确定具体的物理服务,然后通过服务请求代理进行服务调用完成图像处理应用任务。

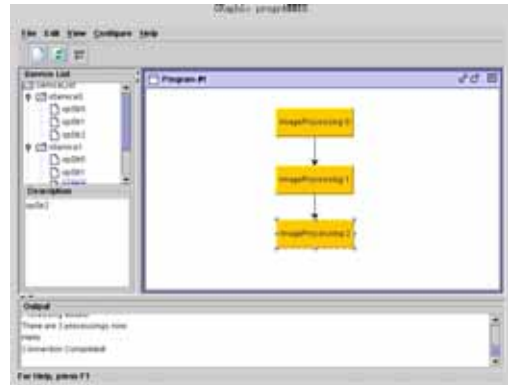


图 4 应用编程环境交互界面

通用服务请求代理

用户请求平台上的某项服务时,通常需要启动调用该网格服务的客户端程序来处理执行。一般说来,服务不同,接口也不同,相应的客户端程序也会有所不同。当有新的服务加入网格平台时,需要编写相应的使用程序提供给用户。服务一旦增多,工作将十分繁琐。通用服务请求代理的设计能为网格服务提供者自动完成这些工作细节,提供一个通用的接口给远程客户端,该接口能访问网格范围内指定的网格服务句柄,并且能够将执行结果返回,大大方便了服务提供者,同时也保证了程序的稳定性、可靠性和性能。

通用服务请求代理采用服务器/客户端模式,其中服务器是部署各种网格服务的网格结点,客户端是用户访问网格的 Web Portal 服务器。整个处理流程为:来自网格用户的服务请求,首先由 Portal 服务器端的调度模块解析出具体的服务类型和参数,交给相应的服务请求客户端,客户端选择合适的网格服务并生成服务请求提交给代理服务器,代理服务器检验请求的有效性和合法性,验证通过后选择已有的或产生新的网格服务实例句柄返回给客户端。客户端收到结果后,远程访问该服务实例,并执行该服务,得到预期的结果。

远程可视化

图像网格利用高性能计算资源完成用户自身无法胜任的图像处理应用,这些应用处理的结果通常是一些高清晰的 2D 图片或渲染后的 3D 图像,如何将处理结果以最快的速度 and 最好的效果呈现给用户是需要考虑的问题之一。然而网络传输带宽与大量传输数据之间的矛盾成了问题解决的瓶颈所在,用户要想通过网络远程查看高质量的图像处理结果比本地可视要困难的多。ImageGrid 研制了 3D 远程可视化工具,利用 IBR (Image Based Rendering) 技术不直接传送拥有庞大数据量的三维模型,而是根据用户请求实时的将服务器处理结果以图片的形式传回客户端,具有传输数据量小、客户端无需安装任何插件、用户交互性好等特点。

ImageGrid 远程可视化工具利用图片在客户端进行新视图的重构,图 5 给出了系统的整体结构图,分为服务器和客户端两部分。服务器端的参考帧生成模块根据客户端信息完成需相应视图的生成;在用户第一次提交请求时,从 3D 模型数据库中读取所需 3D 模型数据,完成最初的绘制工作;之后根据接收的用户信息生成用户视图,压缩后发送给客户端。

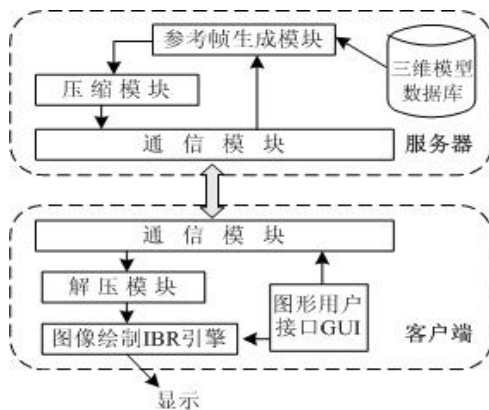


图 5 基于 IBR 的远程可视化结构

客户端的图像绘制 IBR 引擎完成服务器端生成视图的重构工作。GUI 将接收的用户请求同时发送给 IBR 绘制引擎和服务器参考帧生成模块。一方面 IBR 引擎根据原有 Cache 中的视图重构出相应的“近似视图”;另一方面,服务器参考帧生成模块根据用户参数生成“精确视图”,通过网络发送给客户端,替换掉

原有的“近似视图”。例如,当用户只需要对模型或场景进行粗略了解而快速浏览时,“近似视图”完全能够满足用户的视觉要求;而当用户低速浏览或需要对某场景或视图进行细致观察而停下来时,用“精确视图”替代“近似视图”,这样用户不仅觉察不到图像在网络上的传输,而且也提高了显示图像的质量。

3. ImageGrid 典型应用

遥感图像处理

遥感技术已广泛应用于民用和军事技术领域。遥感图像处理是指利用计算机对遥感图像及其资料进行各种快捷、准确、客观的技术处理,如卫星遥感图像预处理、图像变换、滤波增强(图 6 所示)、边缘检测、自动配准等。遥感图像处理具有数据量大且操作复杂的特点,它对计算资源的要求和并行处理的需求十分迫切;同时,大多数遥感图像处理都是线性的,需要人工介入,这样不仅操作麻烦,而且用户的干预也是影响整体处理速度的主要瓶颈。



图 6 图像增强效果示意图

ImageGrid 图像处理网格利用网格中的高性能计算资源作为遥感处理的并行计算环境,基于自行开发的图像处理编程环境,利用服务组件和工作流相结合的技术将其转换成工作流进行处理,既方便了用户,减少了人工干预,又提高了工作效率。

数字化虚拟人

数字化虚拟人是指利用大型计算机把人体形态学、物理学和生物学等信息进行综合处理而实现的数

数字化虚拟人体,可代替真实人体进行实验研究。数字化虚拟人应用的基础在于获取高精度的切片数据,随之而来的是数据量大、层次关系多、数据实体复杂、以及海量数据处理等问题。以我国数字化虚拟人女号为例,切片间距 0.2mm,全身共有 8556 个切片,总数据量达到 149.7GB。若计及随后重构处理所产生的新的数据,整体规模还要成倍增长。如果采用传统方式来处理这些数据,处理速度和处理规模将使该应用受到极大的束缚。为了提高数据精度和工作效率,必须借助网格强大的存储能力和计算能力来解决数字化虚拟人的存储管理、三维重建和远程显示及操作等问题,在分布、异构的环境下实现协同工作。

数字化虚拟人体模型的建立大体可以分为数据获取和存储,数据三维重建及可视化几个步骤。在获得图像数据之后,ImageGrid 利用根据体数据提取物体等值面的 MC(Marching Cube)算法重建三维立体模型,并利用可视化工具对其进行远程显示,可任意查看冠状截面,如图 7 所示。因虚拟人体模型数据量十分巨大,三维重建任务需要利用多个网格结点协同处理、共同完成,以减少执行时间。

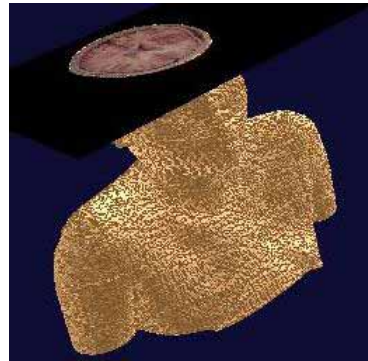


图 7 基于轮廓数据的重建结果及冠状截面显示

医学图象诊断

医学图象诊断应用利用图像网格计算平台,创建一个大型的医学数字化“相册”(图 8),给患者、医生和医院提供更快更准确和简便的方式,以获得更加可靠和精确的各种医学图片(如:X 光片、CT 片、MR 片、B 超片等等),这一应用的实现将改变疾病的诊断方式,实现医疗资源共享。AG

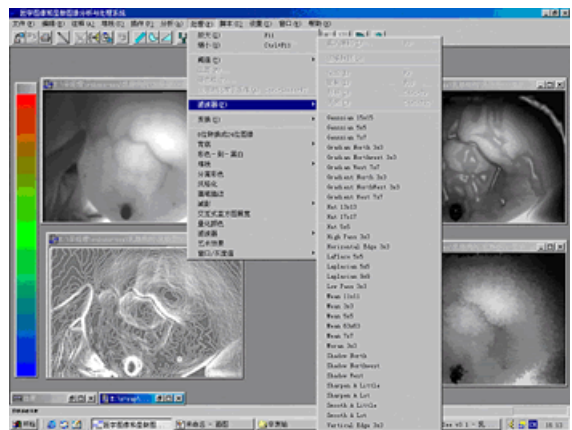
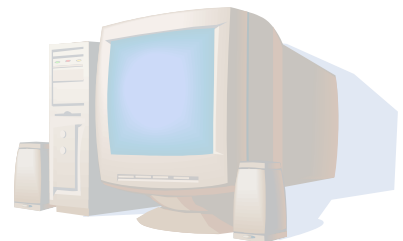


图 8 医学诊断图像库



BUTTERFLY.NET



<http://www.butterfly.net>

Butterfly.net,是一个以 IBM 公司为技术背景的项目,它利用网格技术来运行在线游戏,用来处理在线游戏背后日益增长的挑战,它拥有成千上万的并发的游戏玩家,并需要成百个服务器。Butterfly 使用 Globus 的开发工具,它可以多个服务器协同工作作为一个虚拟组织,在虚拟主机上迅速的切换处理任务。游戏程序可以迅速存取额外的服务器空间和其他的资源,所以它们吸引了更多的玩家。IBM 为 Butterfly 项目提供了服务器,物理存储和相关的软件,利用网格技术使得网络计算更

可靠,更有效。多玩家的 PC 游戏发布商们使用 Butterfly 的软件来产生在线的游戏组件,然后根据它们的使用情况来为这个服务付费。但是这项支持在线游戏的服务早在几年前的 Sony 公司的游戏站上就已经很完善了。

Glympse 是在建立的一个伟大世界。玩家们将发现他们自己正在和一个不断展开的故事情节交互,也正在建立他们的冒险。他们建立城市,经济,和政治结构。他们将开发这个世界,磨练他们的技术,铸造他们的联



盟，还可以宣布他们的战争。Sojourn 经过艰苦的努力设计了游戏软件可以使玩家们在这个虚拟的世界里做任何他们想做的事情，而胜于开发者希望玩家要做的事情，所有这些将在总体上是一个适应游戏玩家兴趣的英雄般的故事情节。在 2004 年的晚些时候，Glympse 将在 PC 上发布他的版本。

Glympse 为什么要选择在网络上呢？

在 Butterfly.net 上运行大半年后，Sojourn CTO Chris Miller 有如下的评论：

在网络带宽方面：“我们非常感谢 Butterfly.net 网络的方式，可以保证良好的网络带宽，减少网络层的噪音。不仅网络使它的客户-服务器通信管道具有良好的利用率，它也使用 UDP 格式提高了包的效率，也使得无连接的协议变得可用。”

在“碎片”方面：“Glympse 的世界是无限的。网络使我们通过提供一种容易和较直观的方式来‘划分整个世界’来维持我们需要的这个世界。Glympse 将使这种分割时没有碎片的，这意味着我们所有的玩家在一个世界为基础进行游戏。这就转变成对完全的网络和服务器的可靠性的一种需要，也需要一种方法是对我们的世界的改变不能是在离线方式下做的。Butterfly 网络和 Butterfly.net 的技术参与者使这些成为现实。”

在“Python 语言”方面：“开始我们担心我们整个游戏开发小组还要学习一种全新的编程语言。我们现在可以放心 Python 使多么的容易学习，作为一种开放源代码的语言，它有一个巨大的支持基础。而且，这种支持是我们的 Butterfly 的开发团队的员工更加有效率”。

开发团队介绍：Sojourn

这是一个来自不同背景：包括网络程序设计，图形程序设计，物理仿真，动画，图像设计，广告设计，游戏设计，视频制作等等 35 人的团队，他们都来自 San Diego, California。

Butterfly 网格：一个分布式的在线游戏平台

Butterfly 网格是一个完善的网格软件解决方案，包括了所有的需要开发，测试，操作，管理和支持时时，嵌入式在线游戏的组件。多种组件的共同工作在物理上分布的异构系统来分派在线游戏的处理，包括：最终可靠性，无限的可测性，高可能的性能，以及计算资源的更有效的利用。

Butterfly 网格在配置可用性上特别的设计满足了游戏开发者，发布商和服务提供商的不同需求：生产，市场，操作和持久性的在线游戏，通过彻底的减少生产和提供在线游戏的成本，同时也提高在线游戏的经历，Butterfly.net 的技术使所有在线游戏工业链上的参与者，包括游戏玩家本身受益。

因为 Butterfly.net 通过提供可以使在线游戏市场达到潜在的顶峰的技术来承诺使所有游戏工业参与者和消费者受益，我们将聚焦在产生一个可以使玩家有最好的体验的平台，玩家所需要的东西：

根本的可靠性：所有在网格上的游戏必须总是可用的，总是可以访问的。

无限制的升级：没有玩家不得不要在一个大厅中等待一个服务器。没有玩家要在一个碎片上远离他们的朋友。

绝对的安全性：客户端的攻击，服务器的脆弱，攻击性的连接和其他侵扰会扰乱乐趣。

工业标准：玩家需要适当的游戏软件包在一个单一的记号上有多属性，完整的语音通信和多平台的支持。

较高的性能：比起落后的挫败，玩家更加抱怨的是不厉害的网络新手。

最大的效率：一个有效率的平台意味着一个成本有效的结构。对于工业，这意味着跟多的游戏玩家和跟多的利益。对于玩家，正意味着少花钱而获得更多的乐趣。



网络技术在地球信息科学中的应用

<http://www.enet.com.cn/>

与空间位置有关的空间信息在信息总量中约占80%，每个国家的社会发展和经济持续增长必然对空间信息资源有空前巨大的需求，不断提高空间信息获取与处理技术的能力才能满足人类的需求。网络技术为空间信息获取与处理提供了新的技术途径，在地球信息科学领域具有非常广阔的应用前景。

1. 空间信息网络概述

空间信息网络 (Spatial Information Grid, SIG) 是一种汇集和共享地理上分布的海量空间信息资源，对其进行一体化组织与处理，从而具有按需服务能力的、强大的空间数据管理和信息处理能力的空间信息基础设施。空间信息网络是一个创新性的体系框架，它为空间信息用户对空间数据进行信息获取、共享、访问、分析和处理等各种需求提供了实用可行的解决思路和实施方案。空间信息网络是一个分布的网络化环境，连接空间数据资源、计算资源、存储资源、处理工具和软件、以及用户，能够协同组合各种空间信息资源，完成空间信息的应用与服务。在这个环境中，用户可以提出多种数据和处理的请求，系统能够联合地理上分布的数据、计算、网络和处理软件等各种资源，协同完成多个用户的请求，确保来自任何空间信息源的空间信息 (any resource) 经过处理能在任何时候 (anytime) 发送并服务于在任何地点 (anywhere) 任何有需求而且有相应权

限的最终用户 (anyone)。

空间信息网络以一种新的结构、方法和技术来管理、访问、分析、整合分布的空间数据，充分利用空间信息系统的各种资源提供服务，实现空间信息的有效共享与互操作，提供空间信息的联机分析处理与服务。空间信息网络提供了一体化的空间信息获取、处理与应用服务的基本技术框架以及智能化的空间信息处理平台和基本应用环境。空间信息网络通过对空间信息源的数据、信息和知识进行有效的描述、组织、管理、处理、交换。

在空间信息网络中，各种空间信息资源被统一管理和使用，空间信息处理是分布式协作的和智能化的，用户可以通过空间信息网络门户透明地使用整个网络上的各种资源。空间信息网络的最终目标是把 Internet 上的空间信息服务站点连接起来，实现服务点播 (Service On Demand) 和一步到位的服务 (One Click Is Enough)。发展空间信息网络将从空间信息应用与服务的技术体系和基础高度角度推动我国空间信息资源的共享与应用，满足日益增长的多层次、多样化空间信息应用需求，这对提升我国空间信息基础技术和应用的水平，促进国民经济和社会发展，具有十分重要的意义。

2. 空间信息网格的研究内容

空间信息网格要求能够对从 TB 到 PB 量级的海量数据进行高效的、实时的分析处理；能够实现应用层面

的互连互通和各种异构资源(如高性能计算机、海量存储系统、GIS 软件系统)共享,从而提高空间资源利用率;空间信息网格不仅可以用于构造新的先进空间信息系统,也可以用于集成现有空间信息系统,从而提供延续性、继承性,保护用户投资;大规模的空间信息应用与服务地域跨度大,涉及多个异地工作部门,需要提供远程访问数据与服务、一站式、无障碍服务;应用系统的业务需求不断变化,系统运行管理策略不断变化,使用模式不断变化,IT 产品技术不断升级,因此需要空间信息网格具有适应动态变化的能力。

为了搭建实用的空间信息网格,应该加强空间信息网格的理论、技术及其应用方面的研究工作。

(1)空间信息网格理论模型研究

空间信息网格的理论模型研究包括空间信息网格的体系结构、技术框架、应用模式等方面的研究,主要回答什么是空间信息网格?空间信息网格由哪些部分组成以及它们之间的相互关系是什么样的?空间信息网格都有哪些技术?这些技术之间的相互关系是什么样的?怎样应用空间信息网格?有哪些可取的商业模式?等等。

(2)空间信息网格关键技术研究

空间信息网格关键技术包括超大规模空间数据的存储与管理、多源空间信息资源共享与融合技术、空间数据元数据及其服务技术、空间信息网格服务技术(包括网格资源信息服务、网格性能信息服务、网格服务信息服务等)、空间信息系统之间的协同工作问题、空间信息网格环境下的地理空间数据可视化等。

(3)空间信息网格具体应用研究

空间信息网格在信息地质中的应用,例如空间信息网格在地质环境仿真、地震预报、油藏模拟、矿产勘探等领域中的应用;空间信息网格在数字城市中的应用,例如数字城市空间信息服务集成,特别是城市突发事件应急响应,如火灾、地震、洪水、台风等;空间信息网格在电子政务中的应用,例如空间信息网格在辅助政府决策方面的应用,包括城市规划、重大工程选址等。空间信息网格在数字流域中的应用,例如利用空间信息网格进行整个流域的数字模拟,包括洪水演进、流域生态、坝位选址等。此外,还有许许多多空间信息网格在各行

业的具体应用例子,此处不再一一列举。

3. 空间信息网格的框架体系

由美国 Globus 工程提出的网格结构体系采用由网格纤维层、网格服务层、网格应用工具层和网格应用层构成的四层结构。网格纤维层提供资源相关、站点相关的基本功能,便于高层分布式网格服务的实现;网格服务层实现资源无关和应用无关的功能,网格服务的实现涉及到地域和机构的分布;网格应用工具层提供更为专业化的服务和组件用于不同类型的应用;网格应用层由用户开发的应用系统组成,网格用户可以使用其他层次的接口和服务完成网格应用的开发。

美国 Argonne 国家实验室、芝加哥大学、南加州大学以及 IBM 公司共同倡议的开放网格服务体系结构(Open Grid Services Architecture, OGSA)采用纤维层、联络层、资源层、协作层和应用层的五层结构。对于各种计算设备、存储设备和网络设备分别开发相应的远程调用控制网格服务。这类网格服务数目众多,每个网格服务的实现细节也各不相同,OGSA 把这类网格服务纳入纤维层。在庞大的纤维层之上是联络层,联络层里的网格服务分成查询服务、通讯服务和安全控制三类。联络层之上是资源层,无论是纤维层还是资源层里的网格服务只涉及单个资源,协作层中有负责协调多个资源调度的网格服务,应用层则向各个领域用户提供软件开发和运行平台。

空间信息网格是在计算力服务、宽带传输和超大规模数据存储等网格支撑环境基础上建立的一个多层次的空间应用服务体系。根据现有的网格体系结构,结合空间信息应用服务领域的固有特性,笔者认为空间信息网格主要由资源层、服务层和应用层 3 个层次组成:

- (1) **资源层** 这些资源包括已建和在建的各类空间数据库(包括基础地理数据库、地物波谱库和遥感信息模型库等)、各种信息处理设备(包括超级计算机、PC、PDA 等)、各种存储设备(大型磁盘阵列等)、各种空间信息获取仪器(包括航空/航天遥感器、地面遥测设备等),它们通过 Internet 或各种无线通信设施实现物理连接。
- (2) **服务层** 提供一个空间信息一体化管理与处理平台,

为综合使用各类资源提供数据存储、组织管理、分发、检索和处理等服务,主要包括遥感信息定量处理软件、大型地理信息系统、空间信息搜索引擎、空间数据和信息整合与组织管理、空间信息智能处理、空间信息在线分析处理和各种服务规范/协议等。

(3) 应用层:提供一个面向应用领域的空间信息集成应用环境,即在上述空间信息网格服务层的基础上,面向各个具体应用领域,对空间信息的使用模式和使用特点,提供空间信息使用政策和协议、应用软件工具、应用开发平台等,建立空间数据处理与信息服务集成环境。

4. 国家 863 计划空间信息网格项目

"十五"国家 863 计划信息获取与处理技术主题的战略目标是面向我国信息资源设施建设的重大需求,发展高分辨率多维空间信息获取技术,研制高分辨率机载光学、微波对地观测数据获取系统,开发轻小型星载高空分辨率多光谱成像仪,掌握自主的先进小卫星对地观测系统技术。系统性地开展对地观测数据定量化、智能化处理技术研究,突破大型地理信息系统、空间信息网络化共享中的关键技术,通过重大应用示范,构建我国的空间信息网格,推动我国空间信息资源的产业化。

"十五"国家 863 计划信息获取与处理技术主题现设空间信息网格总体技术、高分辨力空间信息获取技术、空间信息处理技术、空间信息应用与产业促进以及空间信息获取与处理前沿技术 5 个专题。在课题布置上,该主题以空间信息网格为技术框架,以应用为龙头,强化空间信息获取、处理和应用的有机衔接,形成链式布局,从整体上实现技术的创新跨越和可持续发展。2002 年,与空间信息网格有关的国家 863 计划课题包括空间信息网格框架体系和关键支撑技术(2002AA-13-01-01)和基于空间信息网格框架的城市空间信息应用服务系统(2002AA-13-04-07)。

课题一:空间信息网格框架体系和关键支撑技术

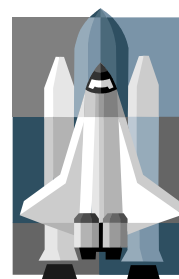
主要研究空间信息网格的体系结构及其应用环境基础框架,实现海量空间信息在线分析处理和服务。基

于空间信息标准和协议,建立空间信息网格核心技术验证平台,提供空间信息处理和大型地理信息系统的基本测试环境。结合资源环境等领域空间信息与服务体系的建设,提供关键技术支持。研究内容包括空间信息网格体系结构;空间信息网格的资源描述、组织与管理技术;空间信息网格中的在线分析处理和服务技术;空间信息处理和大型地理信息系统测试环境;空间信息网格核心技术验证平台和示范应用。

课题二:基于空间信息网格框架的城市空间信息应用服务系统

在城市空间信息基础设施的基础上,选择重点示范城市,根据城市建设、管理与产业发展需求,基于空间信息网格框架,建立城市空间信息资源汇集与共享的基础平台,开发面向政府决策和社会化服务、具有基本按需服务能力的运行系统,促进城市空间信息应用与产业化发展。研究内容包括城市空间信息共享应用的政策与协议;基于空间信息网格框架的城市空间信息获取、处理与应用服务系统的总体设计,并建立相应的软硬件平台;城市多源空间信息获取、共享、传输等关键技术以及基于空间信息网格框架的城市空间信息应用服务系统开发。

AG



GridRM

A Resource Monitoring Architecture for the Grid

1. 前言

GridRM 是一个开源的可扩展的资源管理系统，基于全球网格论坛的网格监控体系结构(GMA)，GridRM 不是同应用程序打交道，而是用于监控应用程序可能利用的资源。GridRM 的整体目标是组织资源的数据，按照客户个人的请求的形式来发送数据。

我们将按照下面的几个方面来进行讲解：

- 设计 GridRM 的动机
- GridRM 的目标
- GridRM 体系结构
 - - 全局层
 - - 本地层
 - - 可扩展的网关架构
- 目前现状
- 将来的工作

2. 动机

电子商务的应用：网格资源类型和可用性的理解是必需的

网格信息服务能提供网格现状的“健康”信息

这些服务可以用于发现和描述网格资源

对于资源信息的缺乏将会妨碍资源的时序安排、分配和使用

重要的网格信息和服务错误将会被报告，并用一种非常适合的方式处理错误

多种成果：

- Heartbeat Monitor (Globus), NetLogger (LBNL), Network Weather Service (Univ. Calif), AutoPilot (Pablo

Toolkit, Univ. Illinois), Remos (CMU), JAMM (LBNL), ...

注意概念：

-网格资源不是应用程序，例如：

- 计算机资源（服务器，集群）
- 专业器材（无线天文望远镜，传感器）
- 网络资源（通信节点，网络设备）

3. 目标

普通资源的控制

通过网格站点可以对当前和近期状况进行有效的控制

客户透明地使用信息

通过多方面报告机制的执行来保护客户

保证没有单独的失效点

信息陈述：

-可以非常简单地使用基于网页的接口

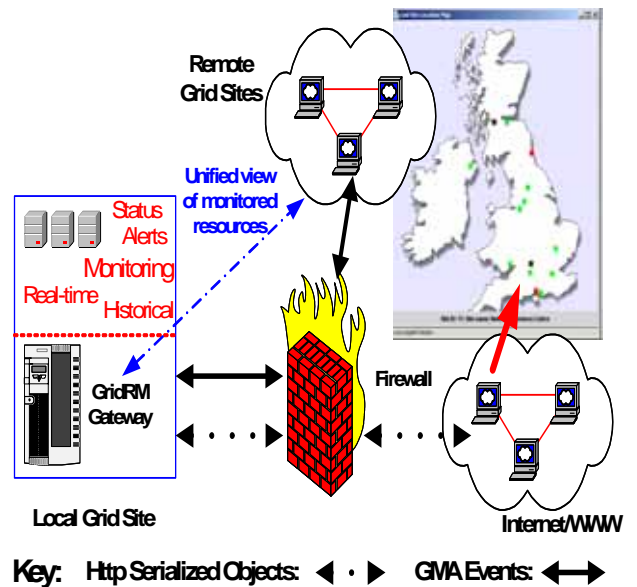
-调度机制可以利用监控数据

安全性

客户可以连接到任何的网关进而浏览资源信息

在单独的网格站点中包含了网关的层次

资源的访问控制：分散的，最少的网关



1) GridRM 全局层

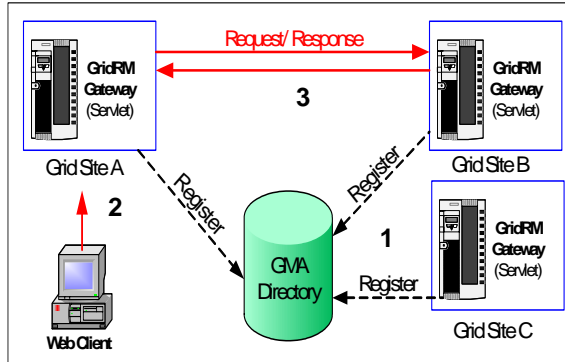
GridRM 网关

通过 GMA 联系在一起

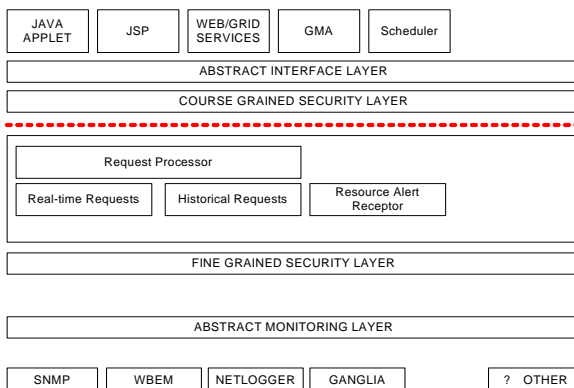
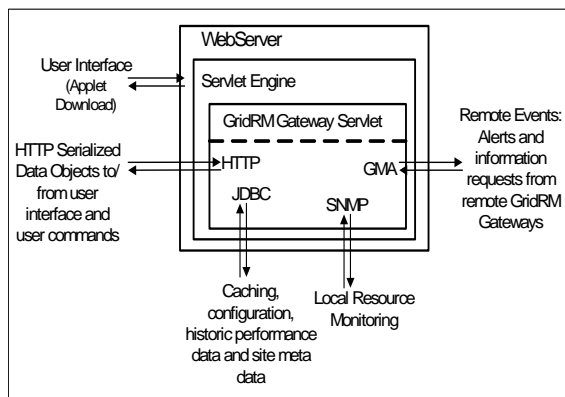
在 GMA 上注册为资源的使用者或者是提供者

如果网关报错，那么客户就进行下一个尝试，

GMA 则重新配置。



2) GridRM 本地层



5. 摘要

网格环境中的开源、普通的、分布式的资源监控系统

基于 GMA 的全球层和本地层

全球层：网关在以可升级和安全的方式交换信息，相互合作提供一致的网格性能信息

本地层：包含在网格站点的内部，监控本地站点的资源

隐藏复杂性（例如，基于网页的图形接口）

客户可以连接到任何的 GridRM 网关进行浏览和查找资源

网关提供错误警报，提供错误的处理。例如，用户转移到不同的网关上获得接口

事件的协调一致性：只有当需要的时候才在网关之间传送数据。

【更多信息】

GridRM :

<http://dsg.port.ac.uk/projects/research/grid/grid-monitoring/>

R-GMA, European Data Grid :

<http://hepunix.rl.ac.uk/edg/wp3>



网格俱乐部

成長

Adaptive Grid



湖北武汉市洪山区珞瑜路1037号南一楼451号 (430074)

电话: 027-87543529