

基于网格的 CSCL 资源共享平台的研究

刘波 翁佩纯

(华南师范大学计算机学院 广州 510631)

摘要 资源共享一直是协作学习研究的一个热点。本文针对当前协作学习系统在资源共享及管理方面存在的问题,运用网格计算技术搭建一个平台,实现资源的有效共享及统一管理,消除资源孤岛,优化资源调度。提出了系统的结构模型,讨论了其访问模型,并实现了网格管理器的四种网格服务。

关键词 协作学习, CSCL, 网格, 资源共享

Research of Grid-based CSCL Resource Sharing Platform

LIU Bo WENG Pei-Chun

(School of Computer, South China Normal University, Guangzhou 510631)

Abstract Resource sharing has always been an intensive research issue in collaborative learning. Aiming at the problems about resource sharing and management in current collaborative learning system, a platform based on grid is designed to provide effective resource share and uniform management, eliminate resource island and optimize resource schedule. This paper designs the system structure, discusses its access model, and implements the four grid services provided by the grid manager.

Keywords Collaborative learning, CSCL, Grid, Resource sharing

1 计算机支持的协作学习(CSCL)

1.1 CSCL 概述

计算机支持的协作学习 CSCL(Computer-Supported Collaborative Learning)是由计算机支持的协同工作 CSCW(Computer-Supported Collaborative Work)与协作学习(Collaborative Learning)的理论及教育方法结合而成的一种学习方法。它利用计算机技术(尤其是多媒体和网络技术)来辅助和支持协作学习。CSCL 不但能够充分发挥协作学习的优势,更能够实现传统教室环境下无法开展的协作学习方式。身处不同地域的多个学习者,利用计算机网络与多媒体等相关技术,针对同一学习内容彼此交互与合作,以达到对教学内容的深刻理解与掌握^[1]。

1.2 在资源共享方面存在的问题

20 世纪 90 年代以来,计算机支持的协作学习系统的研究与开发受到极大重视,出现了许多极具特色的研究与应用成果,并在教学实践中得到很好的应用。资源共享是 CSCL 研究的永恒主题。通过分析基于 Web 环境协作学习系统开发现状^[2],发现该领域尚存在下列问题。

(1) 缺乏有效的资源管理机制,系统信息量大,但利用率不高。

目前这些系统提供了大量的网上资源,却没有一个有效的资源管理机制,不能充分利用与共享资源,严重妨碍了 CSCL 系统的进一步发展。在现有教学系统中,需要查询资料的学生在搜索引擎返回的大量信息中找出可能满足需要的信息,并通过进一步阅读信息内容来判断是否需要该资料。另外,市场上各种各样的教育资源库可谓是“资源丰富”,但其中有效资源内容匮乏,没有参考价值的垃圾资源很多。因此,

在 CSCL 系统中资源管理就显得非常重要。

(2) 异构系统带来资源孤岛问题

随着协作学习和网络教育的迅猛发展,网络上出现了越来越多的教学系统。这些系统都积累了各种资源库和媒体课件库,但却是在不同的时间、由不同的机构、在不同体系结构的计算机上开发的,互不相通。学习者针对不同的教学系统需要以不同的身份注册、登录,不能统一使用纷繁复杂的教学内容和资源,导致网络上的教育信息资源不能得到有效共享,而形成许多“资源孤岛”。

(3) 服务器瓶颈问题

教学系统大都包含实时可交互的视频、语音、白板、文档等,以实现实时互动式教学,其中信息的上传与下载所占比重较大,以致于客户端常常受到服务器端存储和访问瓶颈的困扰。

2 网格计算技术

网格技术^[3]是近年来国际上兴起的一种重要信息技术,被誉为继互联网和 Web 之后的第三次信息技术浪潮。它利用现有互联网的架构,把地理上广泛分布的各种资源,包括计算资源、存储资源、带宽资源、软件资源、数据资源、信息资源、知识资源等整合成一个逻辑整体,实现各种资源的全面共享,消除信息孤岛和资源孤岛。简单地说,传统因特网实现了计算机硬件的连通;Web 实现了网页的连通;而网格试图实现互联网上所有资源的全面连通。网格“在动态变化的多个虚拟机构间共享资源和协同解决问题”。

2.1 网格资源及其特性

网格的资源可以包括计算机、数据库、仪器设备、信息服务等极其广泛的内容,从超大型仪器设备到一个简单的课件

资源都可以成为网格共享的目标。作为一种广域的大规模分布式环境,网格资源具有以下特性。

(1) 分布性。网格环境中的资源是分布广泛的。从局域网到广域网、从有线环境到无线环境,网格资源分布在地理位置互不相同的多个地方。网格资源虽然是分布的,但它们却是可以共享的,即网格上的任何资源都可以提供给网格上的任何使用者。共享是网格的目的,没有共享便没有网格。因此,解决分布资源的共享问题,是网格的核心内容。

(2) 异构性。网格资源是异构多样的。在网格环境中可以有不同体系结构的计算机系统和类别不同的资源。网格资源从硬件体系结构、操作系统到应用软件平台等不尽相同,因此网格系统必须能够解决这些不同结构、不同类别资源之间的通信和互操作问题。

(3) 动态性。网格资源本身是动态变化的,其可用性、性能、访问等策略等是根据应用需求和资源的实际使用情况等因素不断变化的。在虚拟组织中,资源的部署也具有动态性,即资源是可以随时加入和撤销的。网格资源的动态变化特点要求网格资源管理能够及时采取措施,实现任务的自动迁移,做到对高层用户透明或尽量减少用户的损失。

(4) 自治性。网格资源没有也不可能有一个集中的并且完全的控制中心。不同的资源由不同的组织机构所拥有,并有各自的负载管理、访问控制等策略。网格资源管理应该允许资源拥有对资源有自主管理的能力。

网格资源的这些特性给网格资源管理带来了实现上的困难和不可控性。开放网格服务体系结构(OGSA, Open Grid Service Architecture)在一定程度上解决了部分难题。OGSA 已成为网格资源访问接口的事实上的标准。这种标准化接口所带来的优势是显而易见的,除了平台无关性外,接口与实现的分离给用户的使用和资源的管理提供了方便。

2.2 开放网格服务体系

在开放网格服务体系结构(OGSA)中,网格资源以服务形式提供标准化的接口。OGSA 作为一种新的计算模式,提供了在动态的、可扩充的虚拟组织中平等地使用和共享资源的机制。它建立在网格和 Web 服务的概念和技术之上,定义了一个统一的对外服务语义——网格服务。在 OGSA 中,一切都封装成网格服务,网格就是网格服务的集合。网格服务是增强的 Web 服务,开放网格服务基础架构 OGSI 利用 WSDL 和相关的开放规范定义了创建、命名和发现临时服务实例的标准机制,提供服务实例的定位透明性和多协议绑定,并支持与本地平台设施的整合;还定义了包括生命周期、改变管理和通知等一系列机制,这些机制特别适合对协作学习中各种服务和教学资源的统一管理和调度。

2.3 应用网格计算技术解决 CSCL 资源共享和管理问题

网格技术的目标是实现网络环境上的高性能资源共享和协同工作,消除信息孤岛和资源孤岛。将网格计算技术应用到 CSCL 资源共享和管理中,具有如下几个方面的优势。

2.3.1 统一的资源管理

网格打破了传统的强加在资源上的限制,将网络上大量分布与异构的资源进行统一管理,为使用者提供一种前所未有的“高级服务”,这主要体现在^[4]:

(1) 资源的网格化,即将资源从特定的地理位置的束缚中解放出来,使得该资源可以通过网格输送到任何角落,达到网格资源彻底与地理位置无关的目的;

(2) 网格资源的协调,即对任何网格资源,在一定的规则约束和管理下,都可以实现相互协作,打破不同资源之间在广泛共享与协作方面的障碍;

(3) 网格资源的融合,可打破原来在资源能力和资源类型方面的限制。因此,网格系统提供的资源,是增强和放大后的可以动态任意组合的资源。

2.3.2 实现资源共享,消除资源孤岛

网格可以将分布在各个异构系统中的计算机、存储设备、图书馆数字资料、数字博物馆、论文、各类多媒体课件和教学视频等海量信息资源集成起来,建立一个 CSCL 网格。教师和学生可以充分利用这些分布式的、异构的资源,并将资源有效地整合,充分地进行资源共享和协作,为自己的教学和学习服务。教师和学生通过单一入口访问系统,便可透明地访问他们所需要的各种资源,不需要关心资源的物理位置、所在的系统以及如何访问等问题。

2.3.3 优化资源调度,解决系统访问瓶颈问题

网格的资源是分布广泛的,利用网格计算技术进行资源调度,为每个资源请求动态选择最优的网格资源,使网格中的各个节点负载均衡,从而解决了系统访问瓶颈的问题。当用户上传课件资源或教学视频时,系统可以根据资源的大小、服务器节点的可用存储空间以及当前的负载情况来决定将资源存储到哪个服务器的节点上。当用户进行流媒体实时学习时,系统可根据动态负载情况为其选择最优的一个服务器。

3 基于网格的 CSCL 资源共享平台的设计

3.1 系统结构模型

本文综合当前协作学习系统的资源共享和管理的需求,采用 OGSA 网格体系结构,建立网格平台,提出了基于网格的 CSCL 资源共享平台的体系结构模型,如图 1 所示。

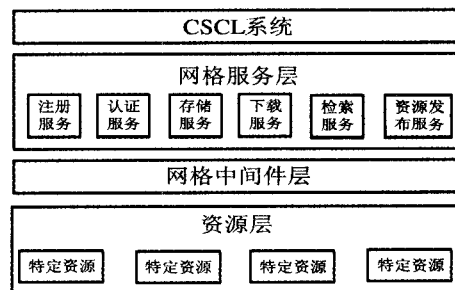


图 1 系统结构模型

该体系结构模型自底向上分为 4 层。最底层就是各种可以用来共享的网格资源,它可以包括极其广泛的内容,从超大型仪器设备到一个简单的视频课件资源都可以成为共享的目标。为了实现该资源的共享,需要提供相应的支持手段,使得其它地方的使用者或者是应用程序可以访问它,甚至是操纵它。

网格中间件层是该体系结构模型的核心。它是指一系列的工具和协议软件,其功能是屏蔽网格资源层中各种资源的分布、异构特性,支持资源的统一管理、分布调度和安全控制,向上层提供透明、一致的使用接口。

网格服务层位于中间件层的上方,主要提供网格环境下的资源共享和管理服务。针对资源的分布性、存取的普适性以及完成大协作任务的多资源动态分配问题,需要在统一的调度框架下协调资源的使用,以优化系统整体的性能。服务

层主要提供的服务包括:注册服务、认证服务、存储服务、下载服务、资源发布服务以及检索服务等。

最上层是 CSCL 系统层,也是网格的应用层。CSCL 系统层是为最终的用户提供的,它提供用来访问网格资源与服务的 Web 接口。对于用户来说,不需要知道网格的具体实现与细节,只需要为他们协作学习提供最方便、最容易理解以及最快捷的支持就行了。因此,CSCL 系统层为用户提供了统一的交互界面。

3.2 网格节点分类

在基于网格的 CSCL 资源共享平台系统中,网格系统是它的核心。在该网格系统中,我们把计算机分为几种角色,网格服务就是通过这些角色来交换信息的。

- 网格管理器 —— 负责从网格客户机接收最初请求,用来控制请求的安全性和身份验证,分发任务,并对所存储的资源进行搜索,对任务进行调度,接受各个异构的教学系统的资源发布请求。

- 网格节点 —— 这里网格节点包括两种节点:一种是系统本身具备的节点,另一种是各个异构的教学系统的节点。在我们的实验中,把网格节点设定为一些存放学习资源的服务器。网格节点负责接收请求来存储资源,发送资源,并支持对资源进行处理。

- 网格客户机 —— 在存储和下载服务以及执行搜索操作时将请求发送到网格管理器。

3.3 系统的访问模型

在网格平台中,我们采用直接模型进行数据的访问和传输,网格管理器负责管理,而各个节点负责实际传输视频数据。教学资源的实际传输可以从存储文件的网格节点直接发起。使用直接访问模型最主要的原因是,这样可以提高可以达到的带宽。通过直接访问这些内容,而不是通过网格管理器来进行访问,就可以确保网格管理器不会由于传输信息而超载——尤其是在对多个网格客户机的需求提供服务时^[5]。

如果使用单一的分发点,即网格客户机直接与网格管理器传输数据,网格管理器上可能会由于在网格客户机与存储这些信息的网格节点之间传输信息而饱和,这样就会出现系统的访问瓶颈。

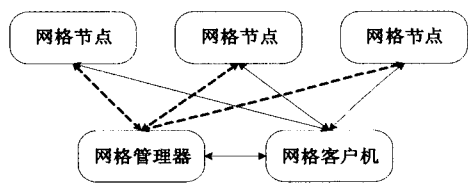


图2 系统的访问模型

4 系统的关键技术

4.1 异构教学系统资源的整合

各个异构的教学系统的网格节点,通过 WSDL 中的接口定义以及 WS-Notification 标准将自身系统描述成对外提供开放接口的教育网格服务。WS-Notification 标准可以由网格节点用来发布所发生的事件(例如网格服务的启动和关闭),网格管理器可以订阅这些通告信息,这样它就可以知道各个网格节点的状态和可用信息了。通过从多个节点上订阅信息,网格管理器就应该可以查看并监视这些状态信息来确定各个节点上提供的服务以及其负载和可用性。

4.2 网格节点状态

正如前面介绍的一样,网格管理器需要具有检查网格节点状态的能力。这可以通过查询节点上存储的教学资源、磁盘空间、相关负载和请求级别等信息来实现,以确保不会产生单个节点过载的情况。

因此,网格平台上的各个网格节点需要能够报告自己的状态和可用性,通过接受各个 Web 服务请求来提供某个特定节点的状态信息。

一个简单的例子是节点的可用性。如果某个网格节点由于某个客户机前面的请求而忙于传送或处理资源,那么它就不应该再被分配任务。这个请求应该由网格管理器调度和排队,从而确保各个网格节点不会超载。

WSRF 提供了一种标准的方法来维护有关 Web 服务的数据和状态信息,并保存状态和多个 Web 服务请求之间的其他永久性数据。Web 服务的属性是在一个 XML 文档中定义的。下面的代码列出了网格节点的部分属性。

```
<xs:element name="status" type="..." />
<xs:element name="cpuload" type="..." />
<xs:element name="diskload" type="..." />
<xs:element name="NodeResourceProperties">
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="tns:status" />
      <xs:element ref="tns:cpuload" />
      <xs:element ref="tns:diskload" />
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

4.3 网格管理器提供的服务

Web 服务描述语言(Web Services Description Language, WSDL)有两种用途:对于一个 Web 服务提供者来说,WSDL 提供了用于定义服务应该接收的消息及其格式的存根和信息;对客户机而言,WSDL 文件可以帮助定义如何将请求提交给服务、请求的格式、消息的内容以及参数及其响应值。

网格管理器提供的部分服务如下:

- Certificate —— 对网格客户机所提供的证书进行验证,并返回一个安全令牌,网格客户机在提交请求时需要使用这个安全令牌。

- SubmitResource —— 接收资源数据,选择存储数据的网格节点,并提供网格客户机将资源数据提交到节点上所需要的安全信息。

- SearchResource —— 使用所提供的条件来搜索资源数据库,并返回当前存储资源的节点的列表以及最适合用户访问的网格节点。

- ConformResource —— 接受各个异构的教学系统的网格节点,将自身系统描述成对外提供开放接口的教育网格服务。

下面的代码定义了主要网格管理器和客户机操作的部分元素,它们之间通过 Web 服务进行的通信是基于消息的交换进行的。消息(包括请求和回复)具有特定的名称、格式以及构成消息内容的系列参数(数据)。由于篇幅有限,以下列出的是 WSDL 文件定义的部分内容。

```
<types>
<complexType name="SubmitResourceResStruct">
  <element name="success" type="xsd:boolean"/>
</complexType>
<complexType name="SubmitResourceReqStruct">
  <element name="token" type="xsd:string"/>
  <element name="resourceName" type="xsd:string"/>
  <element name="resourceData" type="xsd:string"/>
</complexType>
<complexType name="SearchResourceResStruct">
  <element name="success" type="xsd:boolean"/>
  <element name="nodeName" type="xsd:string"/>
</complexType>
```

```
<element name="resourceSize" type="xsd:int"/>
</complexType>
<complexType name="SearchResourceReqStruct">
  <element name="token" type="xsd:string"/>
  <element name="resourceName" type="xsd:string"/>
</complexType>
</types>
```

4.4 网络管理器的连接

下面通过在 WSDL 文件中创建的 Web 服务存根定义到网络管理器的连接:

```
CSCLGridAppPortTypeStub stub = new CSCLGridAppPortTypeStub( null, "http://localhost: 8080/axis2/ services/CSCLGridApp")。
```

结束语 本文提出了利用网格技术来解决协作学习系统中的资源共享和管理问题,搭建基于网格的协作学习系统平台。将网格计算技术和 CSCL 有效结合,能摆脱各类学习系统的运行环境、开发语言等技术方面的束缚,建立稳定、安全的交流平台,真正达到协作学习教学资源高效管理和充分利用的目的。

尽管我们的工作还是很初步的,但通过这次实验,证明了该方案是可行的、正确的,为今后工作打下了良好的基础。进一步的工作包括将该平台由局域网推广到 Internet,并在全国范围内建立一个基于网格的协作学习平台。

参考文献

- 1 黄荣怀. CSCL 的理论与方法[J]. 电化教育研究,1999(6): 25~30
- 2 赵建华,李克东. Web 环境下协作学习系统开发的现状及趋势[J]. 电化教育研究,2004(2): 28~34
- 3 都志辉,陈渝,刘鹏. 网格计算[M]. 北京:清华大学出版社,2002
- 4 都志辉. 网格计算:宏伟蓝图?海市蜃楼? [EB/OL]. http://industry. cidnet. com/art/9/20030123/37462_1. html,2006-7-10
- 5 Brown M. Build agrid using Web services standards[EB/OL]. http://www-128. ibm. com/developerworks/edu /gr-dw-gr-mov-iel-i. html, 2006-7-15

(上接第 98 页)

4.5 水印恢复算法

水印信息的提取过程几乎是嵌入的逆过程,只是在检测到水印位后,为避免因为内容在使用过程中被更新,而造成计算出的元组二维坐标出错,在提取水印时,采用绝大多数表决法 MVM(Majority Voting Method),以得到原始的水印信息。

水印的提取过程如图 2 所示。

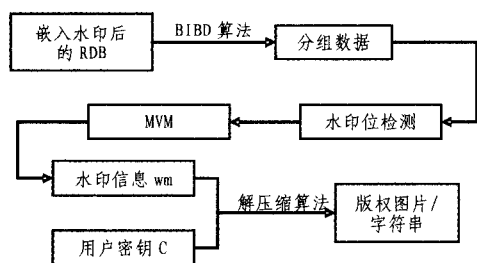


图 2 水印的提取过程

由表 1 和表 2 不难看出,对于子集修改或增加攻击时,攻击的比例越大,对水印的破坏程度就越大,就越不容易提取出正确的水印信息,因而准确恢复的比例就越低。

因为本文提出的嵌入水印算法是根据元组的 MSA 得到的 ID 值以及该元组对应的主键进行的分组排序,与元组本身的顺序无关,所以对于元组重新排序并不会影响水印信息的提取。

总结 本文首次将组合数学中区组设计原理引入数据库水印,并提出了压缩数据库水印技术。在嵌入含有版权信息水印的同时,最小化对原有数据库的修改。通过用户自定义的方式,可以针对不同的数据库中字段的重要性做相应的调整,更加灵活地嵌入水印信息,并且使水印的嵌入过程具有更强的隐蔽性。虽然利用本算法要求原有数据库中至少有 2 个以上的字段是可以进行少量修改的,但在对于大规模的数据库而言,这也是可以接受的。

参考文献

- 1 牛夏牧,赵亮,黄文军,张慧. 利用数字水印技术实现数据库的版权保护. 电子学报,2003(31):2050~2053
- 2 张勇,赵东宁,李德毅. 关系数据库数字水印技术. 计算机工程与应用,2003(25):193~195
- 3 朱勤,于守健,乐嘉锦. 数据库水印研究与进展. 计算机工程与应用,2006(29):198~201
- 4 Agrawal R, Kiernan J. Watermarking Relational Databases. In: Proceedings of the 28th VLDB Conference, Hong Kong, China, 2002. 155~166
- 5 张勇,赵东宁,李德毅. 水印关系数据库. 解放军理工大学学报(自然科学版),2003,4(5):1~4
- 6 Brualdi R A. Introductory Combinatorics, 3E. 机械工业出版社,2006
- 7 Rosen K H. Discrete Mathematics and Its Applications, 4E. 机械工业出版社,2002
- 8 孙松,孙淑玲,邵佩英. 多级安全数据库中标记生成规则的一致性. 计算机工程与应用,2001(16):123~127
- 9 Atallah M S R, Prabhakar S. Rights protection for relational data. In: Proceedings of the 2003 ACM SIGMOD International Conference on Management of Data San Diego, California, ACM SIGMOD, 2003. 98~109

5 仿真实验

实验中使用到的关系数据库共有记录 7600 条。其中可做一定修改的字段为 2 个,即 $k=2$ 。MSA 字段为地区分店名称。嵌入的水印信息为“DongHua Authority@2007”,原始长度为 149,此时,系统提示最佳压缩率 $C=234$,就按此时的 C 进行压缩,得到 $q=97$ 。实验时,测得数据库中共有 3 组元组集合可以嵌入水印信息。

对嵌入水印信息后的数据库进行 10 次攻击,随机修改、增加数据库中不同的字段值,而后对水印进行鲁棒性测试,仿真结果如下。

表 1 子集修改攻击

修改比例	68%	66%	64%	62%	≤60%
准确恢复	5 次	7 次	8 次	10 次	10 次

表 2 子集增加攻击

修改比例	110%	104%	100%	96%	≤95%
准确恢复	5 次	7 次	9 次	10 次	10 次