

一种面向天文网络协同观测的代理服务器^①

A Broker for Astronomical Collaboration Observation

毕树人¹ 南 凯¹ 阎保平¹ 马永征¹ 卢晓猛² 杨宏伟¹ (1.中国科学院 计算机网络信息中心
北京 100190; 2.中国科学院 国家天文台 北京 100012)

摘 要: 天文观测是天文学领域的最基本、最重要的科研活动。天文学网络虚拟实验室致力于通过网络集成多台隶属于不同观测基地的天文望远镜资源, 为天文科学家提供基于互联网的远程协同观测环境。代理服务器是天文学网络虚拟实验室的核心模块之一, 是实现望远镜资源集成与网络协同观测的基础。

关键词: 天文学网络虚拟实验室 代理服务器 协同天文观测 控制权转移 天文照片共享

1 引言

随着计算机和网络技术的发展, 天文观测者现在可以使用天文台提供的客户端软件, 远程地控制该天文台的天文望远镜, 从而足不出户就可进行天文观测^[1]。这种方式为天文观测带来了便利, 但如果某个天文观测者想同时使用多个观测基地的望远镜, 或者多个天文观测者想同时使用一个天文望远镜协同地观测, 是比较困难的。

天文学网络虚拟实验室是国家科技基础平台的一个示范项目, 致力于集成若干天文观测基地的观测资源, 为天文观测者提供远程的协同观测环境。目前, 天文学网络虚拟实验室已集成了国家天文台兴隆观测基地、上海天文台佘山观测基地、云南天文台丽江观测基地和国家天文台怀柔太阳观测基地的多台天文望远镜。通过天文学网络虚拟实验室, 天文观测者可以使用 Web 客户端, 远程地同时控制不同观测基地的多个天文望远镜, 从不同地理位置的天文望远镜观测同一个天文现象; 另一方面, 不同地区的天文观测者也可以同时登录到同一台望远镜, 其中一个用户控制望远镜、其他用户可以看到望远镜的状态和接收拍摄的天文照片。望远镜的控制权也可以从一个用户转移到另一个用户, 从而协同地进行天文观测。这种方式大

大提高了天文观测的准确性和望远镜的使用效率, 开辟了一种新的天文观测工作模式。

代理服务器是天文学网络虚拟实验室的核心模块之一。它连接了天文观测基地服务器和 Web 客户端, 解析、处理并转发它们之间的应用层协议数据^[2], 并管理每台天文望远镜的用户组群。因此, 代理服务器是实现观测资源集成与协同观测的基础, 也是整个天文学网络虚拟实验室性能的关键。本文介绍该代理服务器的功能、设计和实现方法, 以及在天文观测中的应用。

2 天文学网络虚拟实验室概述

天文学网络虚拟实验室的网络拓扑结构可分为四层: (1)客户端, (2)代理服务器, (3)天文台服务器, (4)望远镜。如图 1 所示。

每个天文台拥有所属的服务器, 其管理、控制和连接了该天文台的天文望远镜。为了实现观测资源的集成, 代理服务器连接天文台服务器, 并以客户端的身份登录到每个天文望远镜上。当天文观测者打开 Web 形式的客户端, 客户端自动连接代理服务器, 显示所有望远镜的列表。之后, 天文观测者就可以同时登录到多个不同地理位置的天文望远镜上进行天文观

^① 基金项目: 国家科技基础条件平台“天文学网络虚拟实验室示范”项目(2005DKA64100); 国家高技术研究发展计划(863)(2006AA01A120); 国家自然科学基金(F020303); 中国科学院网络信息中心(CNIC)主任基金(CNIC_ZR_09001)

收稿时间: 2009-03-25

测，而不必关心这些望远镜属于那个观测基地。

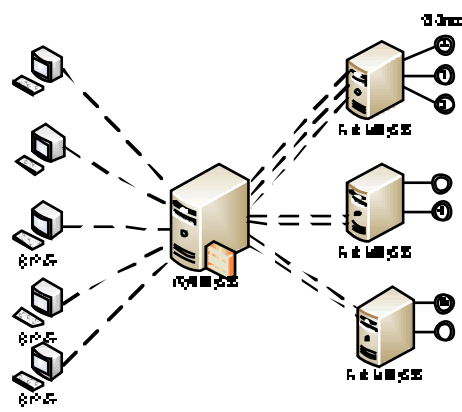


图 1 天文学网络虚拟实验室网络拓扑结构

天文台网络虚拟实验室允许多个用户同时登录到同一台望远镜进行协同观测，用户权限分以下三类：

(1) 控制用户

在同一时间，只有一个用户拥有望远镜的控制权。控制用户可以调节望远镜指向、命令望远镜拍摄照片、接收天文照片和查看望远镜的状态。控制用户需事先向天文台申请望远镜的使用权限。

(2) 共享用户

控制用户可以将普通用户升级为共享用户。共享用户能接收该望远镜拍摄的天文照片和查看望远镜状态。

(3) 普通用户

刚刚登录到望远镜上的非控制用户，默认权限都为普通用户。普通用户只能查看望远镜的状态。

3 代理服务器概述

如图 1 所示，代理服务器是整个网络拓扑结构的枢纽与核心。它连接了所有的客户端和所有的天文台服务器，解析、处理并转发天文台服务器与客户端之间、客户端与客户端之间的应用层协议数据。

对客户端而言，代理服务器是一个统一的网络观测服务器。只要登录到代理服务器，就可以访问所有后端的多个天文台的望远镜，而不必关心这些望远镜属于那个观测基地。

与普通的代理服务器相比，天文学网络虚拟实验室的代理服务器需要满足在天文观测领域的一些特殊要求。

首先，在天文观测中，天文观测者常常根据目前

摄到的天文照片，决定望远镜下一步的动作。天文照片必须是无损压缩传输，通常比较大(2M 以上)，这要求代理服务器必须在较短的时间内把较大的图片传输给天文观测者，通常要求在 10 秒种之内将图片传输给观测者。

其次，天文台要求每个望远镜在同一时间只能有一个用户登录并使用。为了实现多个用户同时登录到一个天文望远镜上进行协同观测，代理服务器需要在其内部将每个望远镜的用户扩展为用户群，并对用户群进行管理与维护。

最后，代理服务器需要与客户端共同实现协同观测和照片共享的功能。

4 代理服务器设计

4.1 总体结构

天文学网络虚拟实验室的代理服务器的主要功能可以分为两大部分：(1)转发应用层协议消息的网络处理功能。(2)管理每个望远镜的用户群、控制权转移和天文照片共享的逻辑业务功能。因此，该代理服务器采用分层结构设计。如图 2 所示：



图 2 代理服务器分层结构图

网络处理层与物理网络交互，负责应用层协议数据的接收与发送。该层将网络操作封装在其内部，向上层的逻辑业务层提供几个接口函数。使用这些接口函数，逻辑业务层就可以方便地发送和接收应用层数据包，而不必关心繁琐的 socket 操作。这样就为逻辑业务层更好的处理实现内部功能提供了便利。接收和发送应用层数据是代理服务器最频繁的操作，因此该层的性能直接影响到代理服务器的整体性能。

逻辑业务层是代理服务器的主体部分，负责处理应用层消息，实现望远镜的用户管理、控制权转移和天文照片共享等功能。

4.2 网络处理层

网络处理层的目的是实现一个高效、扩展性好的网络框架，为上层模块提供简单的网络访问接口。

只所以称之为框架,是因为该层将所有与网络有关的操作、整个程序的控制流都隐藏在其内部。上层的模块只需知道在框架的哪个函数内将接收到应用层消息、在哪个函数内获取 **socket** 函数返回的错误值、调用哪个函数发送消息,至于框架内部如何运作完全不需考虑。因此,网络框架它就像是整个程序的骨架和引擎。

网络处理层的设计中涉及到以下一些关键的技术:

(1) 异步 I/O

在 **Windows** 平台下,**socket** 的工作方式有两种:同步 I/O 与异步 I/O。

同步 I/O 是传统的 **socket** 工作方式。当一个 **socket** 函数被调用后,直到相关的操作执行完成,该函数才会返回。其优点是编写网络程序简单。但是,一旦某个 **socket** 函数的操作无法完成,会使得函数所在的线程阻塞在该函数上,进而使得服务器效率低下。因此,同步 **socket** 工作方式不适合大型网络服务程序。

异步 I/O 是很高效的 **socket** 工作方式,当一个 **socket** 函数被调用后,不论相应的操作是否完成,该函数都立即返回。这样,当一个网络 I/O 操作执行时,线程可以做其他事情,增加了程序的并发度,而且不会因为某些操作未完成而使整个线程阻塞在某个 **socket** 函数上,从而大大提高了服务器程序的工作效率。

(2) 网络 I/O 模型

网络 I/O 模型是编写网络应用程序中另一个比较重要的技术,其主要功能就是通知应用程序异步 **socket** 函数何时完成。异步 **socket** 通常都要与某个网络 I/O 模型结合起来才能使用。网络 I/O 模型由操作系统以应用程序接口(API)的形式提供。在 **Windows** 平台下有下面五种网络 I/O 模型: **Blocking**, **WSAAsyncSelect**, **WSAEventSelect**, **Overlapped** 和 **Completion Port**。

对于这五种 I/O 模型的性能比较,有如下实验^[3]:用它们分别实现五个字符串 **echo** 服务器。然后,若干个客户端同时向一台 **echo** 服务器建立 12000 个连接,不断的向其发送字符串。观测这些 **echo** 服务器建立的连接数、网络吞吐量和线程数。性能数据如图 3 所示:

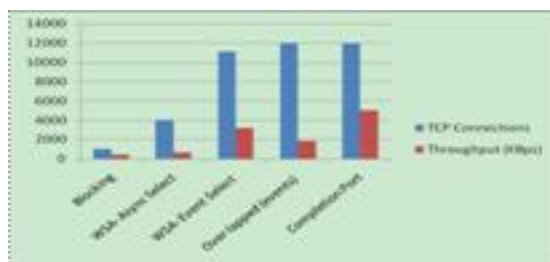


图 3 五种 windows I/O 模型接收连接数与吞吐量

如上图所示,使用 **Completion Port** 网络 I/O 模型实现的 **echo** 服务器,建立的连接最多(为 12000 个连接请求全部建立了连接),吞吐量也最大。而实现这些性能仅使用了 2 个线程。因此,天文学网络虚拟实验室的代理服务器采用 **Completion Port** 网络 I/O 模型实现。

(3) 线程池

在 **Completion Port** 网络 I/O 模型中,通常会预先创建一个线程池,再将若干 **socket** 与一个 I/O 完成端口内核对象绑定。当某个 **socket** 上的 I/O 操作完成时,操作系统就会通知线程池中的一个空闲线程处理这个 I/O 操作。处理完这个 I/O 操作之后,这个线程就再次回到线程池,等待下一次操作系统的通知。

异步 **socket**、**Completion Port** 网络 I/O 模型与线程池三种技术的结合,为天文学网络虚拟实验室的代理服务器在 **Windows** 平台上高效地运行、以及较好的用户数量扩展打下了良好的基础。

4.3 逻辑业务层

逻辑业务层实现望远镜的用户管理、天文照片共享和控制权转移等功能,使用面向对象思想设计。

为了方便对望远镜和客户端的管理,代理服务器在内部定义了两个类: **Telescope** 和 **Client**。**Telescope** 类是对每个天文望远镜的抽象, **Client** 类是对每个逻辑客户端的抽象。每当代理服务器连接到一个天文望远镜上时,就为该望远镜创建一个 **Telescope** 类的对象,用以保存该望远镜的相关信息,并对其进行管理。类似地,每当一个客户端登录到代理服务器上,就为其创建一个 **Client** 类的对象,用来保存该客户端的相关信息,并对其进行管理。**Telescope** 类和 **Client** 类的类图如图 4 所示:

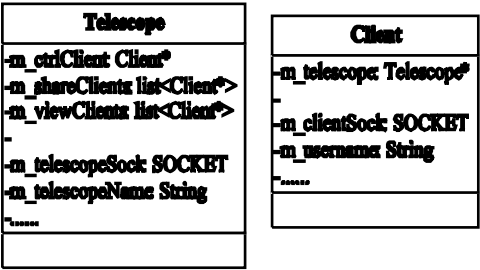


图 4 Telescope 和 Client 类图

(1) 用户管理

用户可用客户端同时登录多台天文望远镜。为简化实现，将每个客户端又分成若干个逻辑客户端，每个逻辑客户端只登录到一个望远镜。Client 类是每个逻辑客户端的抽象，本文后面提到的用户等同于逻辑客户端。

为了实现多个用户同时登录到一台望远镜，Telescope 类为每个望远镜维护了三个用户群：一个控制用户(m_ctrlClient)、多个共享用户(m_shareClients)和多个普通用户(m_viewClients)。它们保存了登录到该望远镜上的用户 Client 对象地址，这样就将使用望远镜的用户数从原来的一个扩展为多个。Client 类为每个用户保存了其所登录的那个天文望远镜的 Telescope 对象地址。通过这种方式，每个 Telescope 对象知道所有登录到其上的用户，而每个 Client 对象也知道其所登录的望远镜。从而维护了望远镜与用户群的对应关系，实现了用户管理。

(2) 天文照片共享

当用户登录某个望远镜时，如果事先申请了该望远镜在这个时间段的观测权限，则代理服务器将其设置为控制用户，否则将为普通用户。控制用户的 Client 对象指针保存在望远镜的 Telescope 对象的 m_ctrlClient 成员变量中，普通用户的 Client 对象指针保存在 m_viewClients。

如果控制用户想把一些普通用户升级为共享用户，他只需在客户端上的普通用户列表中选中那些用户，然后点击“设置共享用户”按钮。代理服务器就会将那些用户的 Client 对象地址从 m_viewClients 列表中转移到 m_shareClients 列表中。

之后当代理服务器收到该望远镜传来的天文照片数据时，会同时分发给控制用户和所有的共享用户。从而实现了天文照片的共享。

(3) 控制权转移

控制用户可以把望远镜的控制权转移给一个共享用户。其只需在客户端上的共享用户列表中选中一个共享用户，然后点击“控制权转移”按钮。代理服务器就会将原来的控制用户的 Client 对象指针添加到 m_shareClients 列表中，并将被选中的那个共享用户的 Client 对象指针赋值给 m_ctrlClient，并从 m_shareClients 列表中删除。这样就实现了该望远镜控制权的转移。

因此，通过对望远镜和用户抽象成类，并在类中维护他们的对应关系，有效地扩展了每台望远镜的用户数，并方便地实现了用户管理、天文照片共享和控制权转移等功能。

5 实现与性能评价

根据上述设计，代理服务器使用 C++ 语言、Windows Sockets 2 函数、Completion Port 网络 I/O 模型、线程池和面向对象技术实现，运行在 Windows 平台上。

转发天文照片的时间是代理服务器最重要的性能参数，该时间受到网速和运行代理服务器的计算机性能影响。在实际测试中代理服务器运行在一台 CPU 为奔腾 4 2.0GHz、内存为 512M 的计算机上，带宽为 10M 的普通网络。随着测试用户数的递增，转发 2.5M 天文照片的时间如图 5 所示：

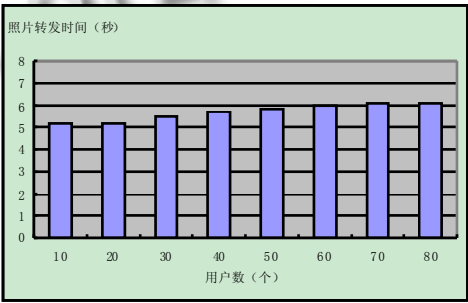


图 5 代理服务器转发天文照片性能测试

当用户数为 10 时，代理服务器转发一个 2.5M 的天文照片用时 5.2 秒；当用户数为 80 时，转发时间为 6.1 秒。其中一半左右的延迟是由于代理服务器等待天文台服务器传输照片数据而引起的。因此，该代理服务器较好地满足了 10 秒内完成天文照片转发的要求。

(下转第 125 页)

6 总结与展望

本文介绍了代理服务器在天文观测中的一个应用和它的设计实现方法。该代理服务器集成了若干天文台的观测资源,并与客户端共同完成了天文协同观测,为天文学网络虚拟实验室奠定了重要基础,大大提高了天文观测的准确性和望远镜的使用效率,开辟了一种新的天文观测工作模式。该代理服务器仍然需要在功能性和高效性方面进一步改进,以便支持更多的用户,同时加入更多的协同天文观测方式。届时将更有效的支持协同科研并为天文观测领域带来更多切实效益。

参考文献

- 1 李绯,王维国,罗泽,阎保平.集成网格服务的中国虚拟天文台 workflow 系统研究与设计.微电子学与计算机, 2006,(S1):105-108.
- 2 李令坤.基于网格服务的虚拟天文台数据处理与封装技术研究[硕士学位论文].北京:中国科学院计算机网络信息中心, 2007.
- 3 Jones A, Ohlund J. Network Programming for Microsoft Windows. 2nd ed. Washington:Microsoft Press, 2002.192-198.