

# C/S 模式下分布式多任务协同应用的实现

柳鑫 (中南财经大学信息系 430064)

## 一、概述

客户/服务器模式下的分布式多任务协同应用与客户/服务器下的其他应用(例如 MIS)有所不同,它是在分布式计算环境下关注多任务的运行,因此它的主要目标在于解决多个任务间的协同。

### 1. 分布式系统

一个分布式计算系统包括软件程序以及在通过通信网络连接起来的各个独立的计算机之间分布的数据资源。分布式应用的体系结构允许各个独立的用户通过连网的计算机共享软件程序和数据资源,通常这些用户在地址上相隔遥远。

### 2. 群件

群件,简单地说就是支持协作工作的计算机软件。在过去的几年中,群件原型和商业产品已经在很多领域被开发出来,例如组写作(一群作者同时书写和编辑同一个文档),电子会议(集体参与讨论问题和作出决定)。

### 3. 协同

“协同”这个词在不同的情况下有不同的意思。广义上讲,它是指使多个独立的工作结合并相互协调以完成一个大的任务。

### 4. 代理者

上面提到的代理者是 Networks 结合进一个活动的对象,称为 agent(代理者)。Networks 是一个基于面向对象的消息传递技术的工具。Networks 结合进了一个活动的对象叫 agent(代理者)。这些代理者提供了独特的、结构化的部件,它们可以(1)调节应用程序和通信内

核的交互作用;(2)执行一些指定给应用程序的综合性或分布式控制的任务。从第(2)点来讲,代理者扮演了类似于 RPC 中 stub 的中间性角色,见图 1。

## 二、分布式多任务应用中的协同需求

### 1. 应用程序的性质

不同的群件应用程序有不同的协同需求并且可以用几种方法分类。

可以从群件的主要功能来为协同需求分类:通信、合作和组决策。如在线交谈或 workflow 管理等这类面向通信的应用程序中,其核心功能是提供一个通道,通过它消息能被正确地传递。通信的协同需求包括正确处理消息路由、活动次序和访问控制。

象联合绘制图表或写一份文件这种合作工作,各面向通信的任务不同,它是一组人员为完成一个任务而在一个明确的对象上协同工作。在这种情况下的协同需求包括版本控制,一致性检查和一致性控制,外加前面提到的消息路由、活动次序和访问控制。

### 2. 工作环境

工作环境的两个方面影响了群件的协同:(1)它是单任务还是多任务环境;(2)工作环境是否允许个人的控制。在单任务环境中,例如电子会议系统或 workflow 系统,协同机制仅对指定的应用程序需求要考虑,换句话说,协同被限定在一个特定的应用程序的内部。

在一个多任务环境中,系统必须支持不同应用程序的多重需求。协同不仅在单个应用程序中需要,而且在多个应用程序之间也需要。

工作环境是否允许个人控制指的是什么范围的用户能定制他们的工作环境。一个定制的环境应该给用户很多选择项,从选择窗口颜色到选择计算平台。一个工作环境允许几种工作平台一起工作时需要更为复杂的协同机制。

### 3. 代理者交互

因为协同要使多个活动协调一致,所以代理者交互也影响了协同的需求。在多任务组工作环境下,每个参与者能加入几个组中,每个组也有很多参与者。因此,代理者交互包含五个不同类型:(1)个人对个人;(2)个

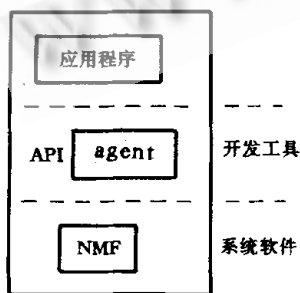


图 1

人对组;(3)个人对所有人;(4)组对组;(5)组对所有人。这里,组指的是为了同样的任务(如画一幅画)而一起工作的参与者,而所有人指的是在群件环境上注册的所有用户。

代理者交互的第三个方面涉及参与者和应用程序的关系。协同机制必须解决以下问题:(1)一个参与者能自由地加入或退出吗?(2)在参与者的数量上是否有上限?(3)一个参与者能被强迫退出吗?(4)在系统的使用上有时间或其他限制吗?

#### 4. 目标控制

目标是参与者在工作的对象。例如,在组写作中,一个被参与者编辑的文档即是一个目标。和目标相关的协同问题包括目标管理,用户权限管理,一致性控制,修改的传播等。

#### 5. 应用程序工具

不同的应用程序可以使用不同的工具并有不同的协同机制。例如,在一个电子会议系统中有时需要锁定一个数据记录以便参与者编辑它。但当使用投票工具时这一点不需要,因为用户不需要访问同一个文件或文档。下面的三个关系涉及应用程序工具的协同需求:

- (1)对参与者来说,谁能使用哪一个工具?
- (2)对工作环境来说,在一定的环境下什么工具适用于应用程序?工具如何来适应不同的工作环境?
- (3)对目标来说,对什么目标应使用什么工具?

#### 6. 系统维护

系统维护指的是一个系统的操作需要,包括数据恢复、数据备份等。在一组参与者交互作用时,数据的正确性和一致性的维护也是很重要的。

### 三、分布式多任务协同应用中的三层结构

对于分布式计算的传统的 C/S 体系结构包括两层:一个客户层和一个服务器层。系统的逻辑功能中,针对用户端的应用,如用户界面单元,定位到客户端,而涉及用户交互或综合的、多重的应用程序被定位到服务器。

这种两层 C/S 体系结构在多任务群组环境中是不适合的,因为在这种情况下的协同作用不仅涉及单个应用程序之中,而且必须考虑在多个应用程序相互之间的作用。因此,当设计这样一个系统时,我们必须把针对某个特别应用程序的协同功能从被所有程序共享的功能中区分开来。本文提出一个三层体系结构,它把传统的 C/S 体系结构扩展为三层:一个群件服务器、应用程序服务器和客户端,见图 2:

#### 1. 群件服务器

群件服务器的主要目的是为所有群件应用程序提供必要的协同功能。它包含了基本的通信功能如消息路由和顺序,还有用户管理功能如增加和删除用户。对于消息路由,群件服务器追踪哪个应用程序正在运行以及每个应用程序服务器和它的客户的位置,以便消息能被传递到正确的目的地。对于用户管理,群件服务器为所有应用程序维护着用户的描述和权限。

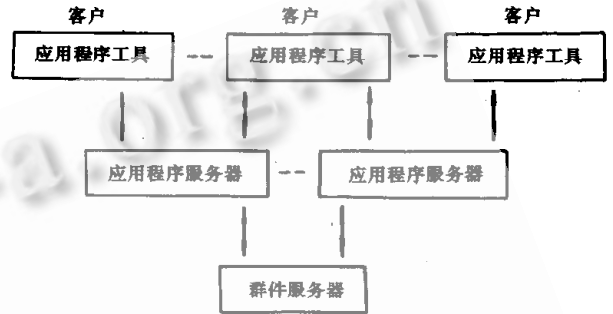


图 2

#### 2. 应用程序服务器

一个应用程序服务器是一个系统,它关注和某个特定应用程序的执行有关的协同和其他需求。它建立于群体服务器之上。一个应用程序服务器的主要职责包括:

- (1)控制对应用程序工具的访问;
- (2)执行版本控制;
- (3)检查一致性以确保改变被及时记录和传播;
- (4)在实时应用程序中控制一致性和冲突的解决。

#### 3. 客户

在应用程序服务器上面的一层是客户。它主要是提供用户界面和应用程序工具以使多种活动能执行。例如,组绘画需要绘画工具如画笔、刷子和擦子等。所有这些都在客户层提供。电子会议系统的客户端功能有投票和组决策工具等。

### 四、一个原型——DEEDS

本节提出一个分布式多任务协同应用的原型 - DEEDS (Distributed Environment For Electronic Decision Support)。它是一个分布式多任务环境下的系统,支持在线交谈、组绘画和电子会议三个群件任务。它显示了三层体系结构的可行性。系统在 MS - Windows 环境下实现,用 Borland c++ 和 Winsoket 库作为开发工具。既然群件应用程序可被分类为通信、合作和组决策,所选

的三个应用程序作为每一类的典型:在线交谈作为通信类,组绘画作为合作类,电子会议作为组决策类。图3说明了 DEEDS 的开发环境。

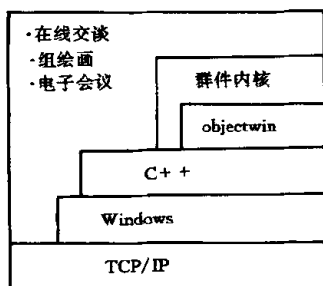


图 3

图3中, DEEDS用TCP/IP作网络协议。选择TCP/IP是因为两个原因:(1)它支持局域网通信和广域网通信;(2)它是一个事实上的工业标准,没有特定的限制。在TCP/IP上面,是MS- Windows环境,选择它主要是因为它很流行并支持多任务。

### 1. 应用程序的功能

DEEDS的应用程序功能属于三个子系统,在线交谈、组绘画和电子会议,见图4。

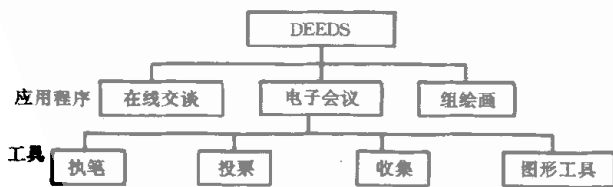


图 4

在线交谈是一个在线通信工具,它在各个用户间彼此交换信息。它为会议的参加者或组的设计者提供通信通道。

组绘画是流行的画笔程序的基于群组的版本。它包括通常的绘画功能,台画笔、刷子、擦子、字母表、画线、画形状(圆、方形)等。但是除了个人用具外,组绘画允许一个组的用户共同绘画。

电子会议由意见产生,意见评价和结果的提出等工具组成。其中,执笔功能作为意见产生工具,通过它一组参与者输入他们的意见。每个参与者不仅能看到他

自己的意见,而且能看到其他组成员输入的意见。

一旦足够的意见产生了,会议主席能停止执笔功能并请求意见评价和作出决策。DEEDS的意见评价工具包括注释、投票表决、收集等。主席可以从中选择一个工具。例如,主席激活了投票工具,选票会产生,并通过电子邮件发送给参与者。参与者对他们支持的挑选出的意见投票表决,结果被回送到会议主席,然后以图形或表格方式显示出来。

### 2. 任务的协同

DEEDS中的任务协同在三层结构中都存在。群件服务器承担了大部分应用程序间的协同功能。通过定义应用程序协议来确认每一个应用程序以便群件服务器通过确认特殊的消息来认知应用程序。

应用程序内的协同是由应用程序服务器承担。因为不同的应用程序有不同的协同需求,所以应用程序服务器的复杂性不同。例如,交谈提供的工具仅是为了收到和发送文本,所以交谈服务器的主要功能使消息正确的排序。

组绘画中的协同更复杂一些。例如,多个用户也许想在同一个区域工作,这会引发冲突。一个用户也许想删除另一个用户生成的对象或增加一个对象在上面。换句话说,此时应用程序服务器必须管理空间分配,对象归属和用户权限。

组绘画在其应用程序服务器中用一个基于工作空间的方法来处理协同。简单的方法是:应用程序服务器给每一个注册到系统的用户分配一个空的工作空间。用户在自己的工作空间有充分的权力,包括读、写、删除和修改。用户可以查看其他参与者的工作空间,但如果没有这个工作空间拥有者的授权,他不能写、删除和修改此空间中的对象。这种方法降低了协同的复杂性。

当需要看到联合工作的结果时,一个用户能把所有用户的工作自由地结合起来。工作空间可被重叠起来以显示结合的成果。

电子会议中的协同在这三个应用程序中是最复杂的。有时人为的中断是必须的,例如,当电子会议从一个阶段移到另一个阶段,会议主席必须通知所有参与者以确信他们都跟上了进度。当意见已经收集得足够多时就要申请投票表决,此时主席必须停止再产生意见并开始投票。这些阶段地转换活动必须人为地控制。

(注:原文篇幅较长,本刊做了删略。)

(来稿时间:1997年2月)