

多机协同建立并发系统模型实现原理

任爱华 牛锦中 孙自安 林仕鼎
北京航空航天大学 计算机系 100083

摘要

本文介绍在并发系统开发环境中采用多机协同方式建立应用系统模型的实现机制,该机制是目前我们完成的 OOPN-IDE 集成开发环境的通信控制子模块,该模块采用分布式技术完成开发环境中多机协同机制。

1 引言

OOPN 集成开发环境 OOPN-IDE (Object Oriented Petri nets Integrated Development Environment),是以面向对象 Petri 网为基础,支持并发软件建模的一种工具,包括 OOPN 模型的建立、仿真、管理和并发软件的生成。

利用 OOPN 模型描述比较复杂的系统时,所生成的模型有可能十分庞大,若由一个人来创建,将会花费大量的时间或者几乎不可能,所以复杂系统的 OOPN 模型必须由多个人协作共同完成,这就需要本环境支持多机协同机制,以便允许多个用户在多台机器上同时协作创建同一个 OOPN 模型。另外,若共享已经创建的 OOPN 模型,简便的作法是在本地环境中直接打开远程机器上的 OOPN 模型,多机协同机制的分布式支持环境会提供更方便的手段。

2 技术支持

随着微处理和网络通信技术的迅猛发展,分布式网络计算逐渐代替以单机为中心的计算模式,分布式系统已成为计算科学的一个重要研究内容。相对传统集中式主机或其它单机系统,分布式系统的主要优点有:1、具有良好的性能,价格比。2、更合适于具有相互协作性质的应用。3、具有更高的可靠性。4、具有良好的可扩展性。因此,本环境将采用分布式技术实现多机协同机制。

在建造分布式系统时,RPC(远程过程调用)是一个关键的部分。它的目标就是使用远程机器上的过程看起来就象调用本地过程一样。RPC 的实现并不象其概念一样简单,各种系统中 RPC 的设计实现不尽相同。这一事实是造成各种分布式环境的可靠性,透明性,和灵活性等方面差别的一个重要原因。本系统是用 Java 语言开发的,为了访问远程对象,采用了 Java 语言的 TCP/IP 协议应用接口。

3 OOPN-IDE 的体系结构与多机协同机制

OOPN-IDE 的体系结构如图 1 所示。

在上述技术的支持下,本系统以 Client/Server 方式完成整体结构设计,它支持同一软件的分布式协同建模和管理,由网络通讯模块完成 Client 与 Server 的通讯。Client 端完成与用户的交互,Server 端负责存储和管理用户建立的 OOPN 类及相应实现类,并对 Client 希望得

到的各种信息提供服务，对 Client 反馈的信息进行应答和控制，以及对多 Client 的协同建模操作进行必要的一致性检查，以实现多个 Client 的协同。

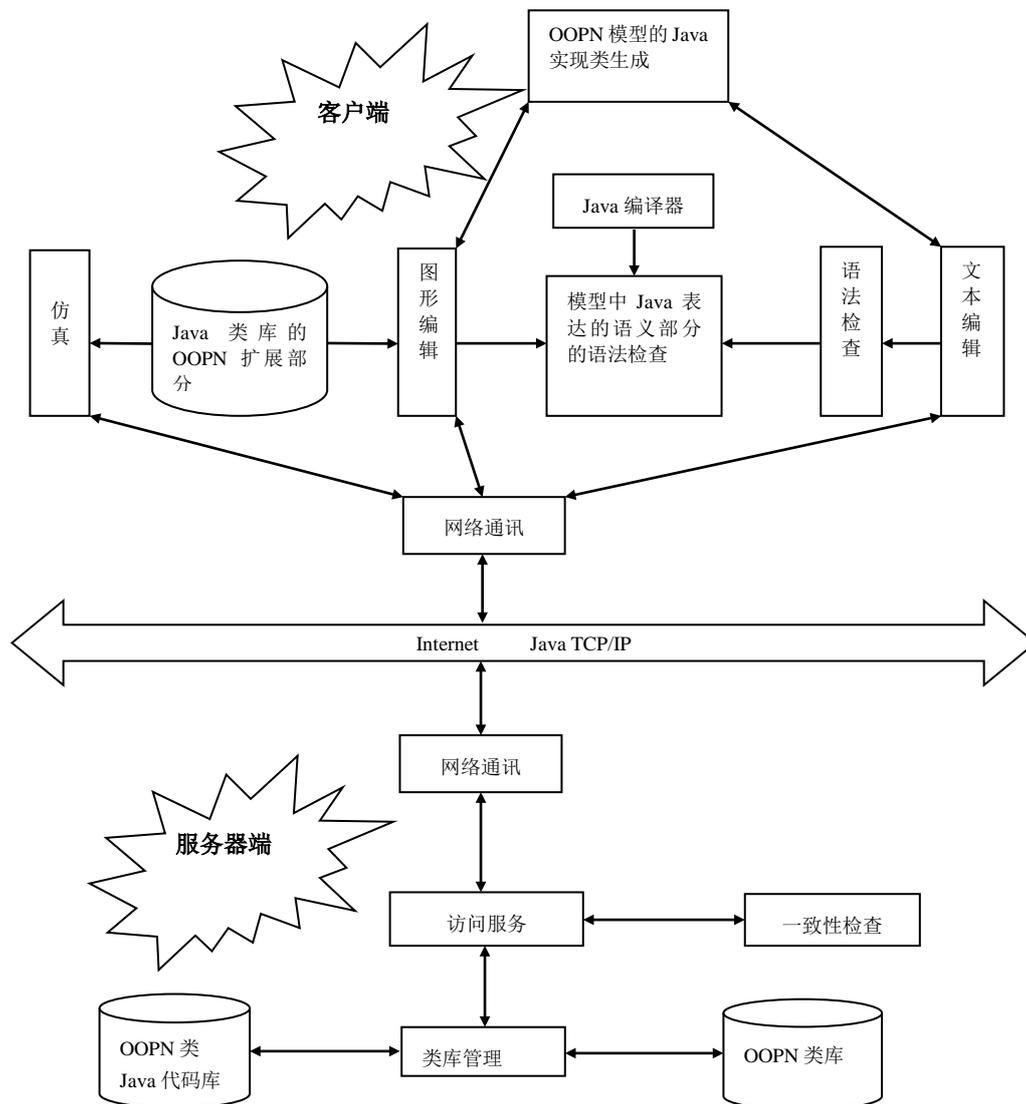


图 1 并发软件开发 OOPN 支持工具

4 多机协同功能模块及工作流程

多机协同工作是由多个功能模块来完成的，其中客户部分包括四个功能模块：连接模块负责申请与服务器建立连接；指令转换模块负责将客户的原始指令转换成可以进行机间传送的数据结构或相反；数据发送模块负责将转换后的数据发送给服务器；接收模块负责接收来自服务器的返回信息及指令。

服务器包括以下功能模块：连接模块负责接收客户的连接申请，帮助客户建立连接；接收模块接收来自客户的指令；发送模块将指令及反馈信息发送给用户；指令分析模块对接受

到的指令根据以前的指令执行情况分析其可否执行；指令执行模块对已经确定执行的指令执行，修改主机的数据结构。

这些功能模块的工作流程如图 2 所示，其工作流程如下：

- 1、客户端的连接模块向服务器发出连接请求，使客户机与服务器建立连接。
- 2、客户端的指令转换模块将要发送的原始指令转换成可以进行机间传送的数据结构，并将转换后的结果传送给数据发送模块。
- 3、客户端的数据发送模块将转换后的指令发送给服务器。
- 4、服务器的接受模块接收到客户机发来的指令后，将其传递到指令分析模块。
- 5、服务器的指令分析模块分析收到的指令，若指令可以执行，则将其交给指令执行模块执行，并将成功信息发送给原客户机，将指令发送给所有其它编辑该文件的客户机；若不能执行，则将出错信息发送给原客户机。
- 6、客户端的接受模块接受来自服务器的反馈信息或指令。
- 7、若客户端接受到的是指令数据，则接受模块将其传递给指令转换模块，以便转换为客户机能够识别的指令。

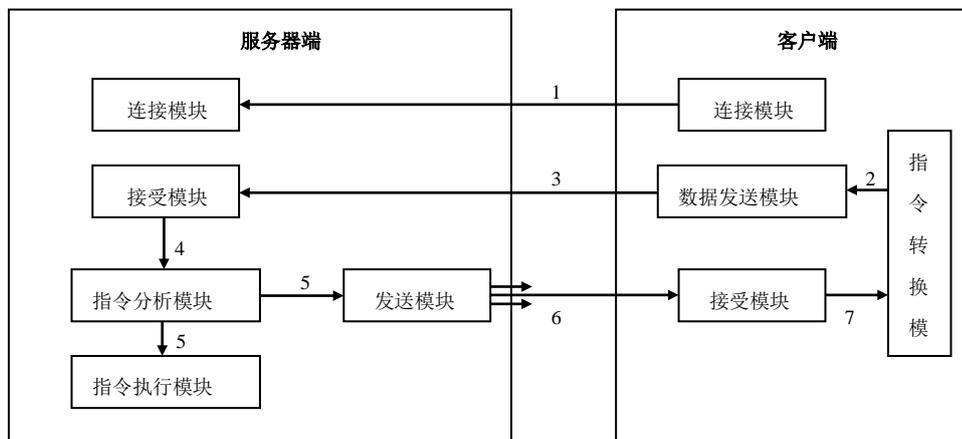


图 2 多机协同工作流程

5 实现多机协同的线程及其数据结构

在多机协同中主要依靠线程机制完成并行工作。网络通信实现中主要利用了 JDK 中的 Java.net 程序包提供的类和方法。

5.1 服务器端的服务线程

1、server 线程

监听网络连接端口，提供连接服务，创建 ClientServer 线程

2、ClientServer 线程

对应于一个客户的连接，接收该客户的指令，并根据指令做相应的工作：为该客户分配一个唯一的端口，并告知客户，然后对此端口进行监听，随时准备接收和处理客户发来的指令。

数据结构:

- 1) Hashtable fileList: 记录该客户打开的文件。
- 2) Socket 和一些流: 用于网络通信。

对外接口:

- 1) send(Hashtable cmd): 将参数 cmd 传送给客户, 该方法主要由 FileServer 调用。
- 2) shutdown(): 若服务器关闭, 则由 Server 调用该方法。

3、FileServer 线程

对一个已打开的共享图形指令文件进行编辑。接收从 ClientServer 传来的对图形结构进行修改的指令, 做相应处理, 并保证数据修改的一致性。一致性的保证通过两个渠道来完成: 一是把几个端口同时发来的指令按一种简单算法排好队依次处理, 每次一条指令地向各端口广播通知其变更。二是利用互斥锁来锁定共享数据结构, 以保证绘图时的某些复合动作能正确完成。

数据结构:

- 1) Hashtable clientList: 记录连接到这个文件的所有客户。
- 2) Hashtable cmd, ClientServer client: 是 ClientServer 线程传进来的参数, cmd 为指令, client 为这个线程的引用。
- 3) Monitor monitor: 本线程调用 monitor 的 suspend 与 resume 方法, 将自己阻塞或唤醒。
- 4) DocPane pane, OPNClassShow show: 打开的文件的数据结构。

对外接口:

- 1) beDestoried(): 当本文件服务线程陷入死循环时, 由 ClientServer 线程调用。
- 2) shutdown(): 当服务器要关闭的时候, 调用该方法关闭该文件。

5.2 客户端的服务线程

1、Client 线程

主要负责客户的连接与断开连接, 平时处于阻塞状态, 由服务器控制其唤醒。

- 1) 先阻塞自己。当被唤醒时, 分析接收到的命令。
- 2) 分析接收到的指令:
 - I. LOCAL_CONNECT, 与本地的 Server 创建连接。
 - II. REMOTE_CONNECT, 根据用户输入的 IP 或域名与远端的 Server 创建连接。
 - III. DISCONNECT, 断开与 Server 的连接。
- 3) 连接建立的过程:
 - I. 首先在 Server 的端口处创建连接, 读取 Server 传来的新端口号, 断开这个连接。
 - II. 在新的端口处与 Server 创建连接, 创建完成后, 取得输入输出流。
 - III. 创建 Import, Export, Receive 等线程, 并初始化这些线程的输入输出流。Import, Export 和 Change 的输出流为这个 Socket 的输出流。Receive 的输入流为这个 Socket 的输入流。使 Import, Export, Receive 等线程开始

运行。

- 4) 断开连接的过程:
 - I. 关闭所有在服务器上打开的文件。
 - II. 断开连接。

2、Import 线程

主要负责用户打开服务器上的文件这一操作:

- 1) 先阻塞自己, 当被唤醒时, 准备打开文件。
- 2) 打开文件的操作:
 - I. 首先向服务器发送 Import 命令, 准备接收文件列表。
 - II. 接收到文件列表后, 由用户选择一个要打开的文件。
 - III. 向服务器发送 openfile 命令, 文件名在 filename 中, 等待服务器的返回。
 - IV. 接收到服务器返回的 OPNClassShow 类, 根据这个类生成 DocPane。
- 3) 操作完成后, 阻塞自己。

3、Export 线程

主要负责保存、关闭文件操作:

- 1) 先阻塞自己, 当被唤醒时, 判断指令类型。
- 2) 分析指令类型, 并执行相应操作:
 - I. 若为 Export, 则文件名可由 DocPane 的 getTitle() 得到。
 - II. 若为 ExportAs, 则由用户输入文件名。
 - III. 将指令类型和文件名向服务器发送, 等待服务器的回答。
 - IV. 若服务器返回为 saveok, 则文件正确保存; 若为 saveerror, 则文件保存出错; 若服务器在 30 秒内没有返回, 则认为出错; 显示提示信息。
- 3) 关闭文件的操作, 若本客户为该文件的最后一个客户, 文件服务线程会询问是否保存, 用户回答后, 关闭该文件窗口。

4、Change 线程

负责用户对文件的修改, 每次的改动都会生成一个线程来执行:

- 1) 根据用户的各种操作的指令名和部件对象的名字生成关键字, 在 Receive 的 change_list 中登记, 值为本 Change 线程的引用。
- 2) 将用户要添加、移动、删除的部件对象和指令名传递给服务器, 等待返回。
- 3) 变量 ok 原本为 null。在 30 秒内, 如果 Recieve 线程接收到服务器的返回, 则 Receive 线程会将 ok 置为 “ok”。每 1 秒判断一次 ok 的值, 若不为 “ok” 则转向睡眠。过了 30 秒, 则认为出错, 操作失败, 给用户提示信息。

5、Receive 线程

主要负责接收服务器的返回, 并根据指令类型, 传递给其它线程处理或自行处理:

- 1) 若是接收到与文件打开有关的指令, 则传递给 Import 线程解决。
- 2) 若是接收到与文件存储和关闭有关的指令, 则传递给 Export 线程解决。
- 3) 其余的指令为文件被修改后, 服务器的返回信息。
 - I. 先判断该指令是否是由本用户发出的, 根据接收到的指令类型和部件名形成关键字在 change_list 中查找, 若该关键字存在一个值, 则取出该关键字对应的 Change 线程, 将其中的变量 ok 置为 “ok”。
 - II. 根据得到的指令执行相应的操作。

6 总结

作为多机协同并发程序建模工具，提供了广播方式的进程通信模型，并且保证了多机协同工作时数据的一致性。主要局限是通信抽象层次比较低，采用 Client/Server 模式。

OOPN 并发程序开发集成工具已完成的子工具有：面向对象 Petri 网建模工具（图形方式或文本方式）、并发程序执行工具（系统运作）、死锁检测工具、以及系统调试工具。

目前 OOPN 工具包已完成以多机协同方式建立 Petri 网模型，进一步的工作是完成多机同时进行同一 Petri 网的执行。

7 参考文献

1. 杨文龙 等，“基于 PETRI 网并发软件开发方法及其支持工具的研究(论文集)”
科技文献出版社，1993 年 5 月，ISBN 7-5023-2034-2/TP
2. 任爱华 等，“基于 PETRI 网的并发系统死锁检测方法”
科技文献出版社，1993 年 5 月，pp66-81
3. 任爱华 牛锦中，“面向对象 PETRI 网建模方法”
第四届面向对象语言与系统技术国际会议暨首届中国面向对象技术及应用学术会议论文集(TOOLS Asia'97&OOT China'97)，1997.9.22 pp26-30，
ISBN 7-80003-409-7/TP•16
4. 任爱华 等，“一种基于面向对象 PETRI 网的并发程序建模方法”
北京航空航天大学学报，1998 年，vol.24 ， No.4， pp491-494， ISSN 1001-5965
5. 廖卫东，陈梅，“JAVA 程序设计指南”，机械工业出版社，96.10

The principles of implementing concurrent system modeling in the multiprocessor coordinated

Aihua Ren, Jinzhong Niu, Zian Sun, Shiding Lin

Dept. of Computer Science, Bijing University of Aeronautics and Astrunautics, 100083

Abstract

The implement mechanism of concurrent system developing environment in multiprocessor coordinated is introduced in this paper. This mechanism is the communication module of OOPN-IDE integrated developing environment. The module design adopted the distributing technique for achieving multiprocessor coordinated developing environment. OOPN-IDE stands for Object Oriented Petri nets Integrated Development Environment. This environment is developed by us.