

一个基于 C/S 模式的证券交易系统

汤仰止 (湖南省工商银行稽核处 410011)

1. 引言

为适应广大投资者的需求,湖南省工行自行开发了一套基于 Client/Server 模式的证券交易系统。该系统是在 UNIX 操作系统下,基于 INFORMIX 数据库开发的广域网方式的证券交易系统。与其他证券交易系统相比,本系统具有网络功能强、投资省、安全性好等特点。

2. 总体结构

本系统基于银行的机构特点而设计。由于各银行下属机构多、地域分散,若各分支机构在上海、深圳自行设立交易席位,其席位费、通信费、人员费将会相当昂贵,交易成本将大幅度提高。这对于经济欠发达、交易量不大的地区将难以承受。这将遏制证券交易的发展,而且影响银行开发新型业务。基于这种情况我们利用 INFORMIX SE 4.0 数据库设计开发了适合银行系统的证券交易系统。本系统利用 Client/Server 模式在各级的交易系统上装上相应的 Client 或 Server 部分及其应用程序。系统拓扑结构如图 1。

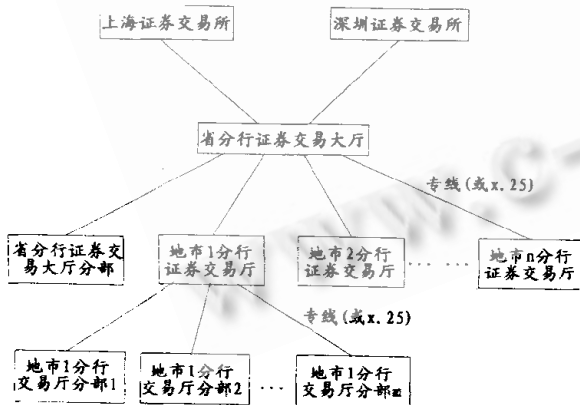


图 1

省分行证券交易厅的系统中,除安装基本应用系统外还需安装 Server 部分,交易厅分部则需要安装 Client

部分,地市分行证券交易厅若有分部则在地市分行证券交易厅的系统中需要安装 Client 和 Server 两部分。通过 SPX/IPX 将 UNIXWARE 服务器与工作站相连接,或者通过 TCP/IP 将 Novell Netware 与 UNIX 相连接,使通过卫星接收的证券行情及成交回报直接进入各网点的应用中。

三、实现方式

系统应用程序主要使用 INFORMIX 4GL 编写,使开发周期大大缩短。而对于 4GL 中的通信部分则使用 C 语言编制,然后利用 4GL 提供的工具 cflglo 生成新的 flglo 程序,使得用 C 语言编制函数能有机的嵌入 4GL 中。通信程序与应用程序用流文件方式交换信息,其中规定:Client 方的应用程序为 CA,Client 方的通信程序为 CC;Server 方的应用程序为 SA,Server 方的通讯程序为 SC。Client 方有流文件 comcltr 和 comcltw,Server 方有流文件 comsrvr 和 comsrwv;其关系可由图 2 说明;CA 与 CC 以及 SA 与 SC 之间的协调工作是靠两个消息队列来完成的。而 SC 是一个精灵进程,系统启动时即有此进程守候,当任何客户机与之联结时本精灵进程将自动复制自身与之对应,并为之服务,服务结束后本进程将自动消失。通信协议使用 TCP/IP,每一个交易信息均由应用程序首先进行打包处理,再交给 TCP/IP 进行通信打包处理。考虑到我行所有前台业务系统均使用 INFORMIX 数据库设计,为考虑今后联网的需要,我们设计了一个通用的打包格式;打包方式为:包头 + 通信内容。

·包头为 15 个字节定长格式,其内容如下:

内容	变量类型	备注
job-run	long	工作流水号
job-tyt	short	工作种类
sub-tyt	short	工作子类
clt-adr	short	客户机地址
srv-adr	short	服务器地址
buf-len	short	信息长度

其中:

·流水号:为整个 Client 方通信的流水号,为今后核查而设立;

·工作种类:为了使本系统可以为银行其他业务服务而设立。现规定如下:

- 1. 证券业务
- 2. 会计对公业务
- 3. 储蓄业务

·工作子类:在每一工作种类中均可以有许多的不同的业务小类,本系统规定:

- 1. 委托买卖
- 2. 撤消委托
- 3. 托管
- 4. 转托管

·客户机地址及服务器地址:本系统各下级单位统一的编码,其真正的 IP 地址与本编码的对照表由系统统一管理。

·信息长度:为本次通信的真正信息内容的长度(不含包头)。

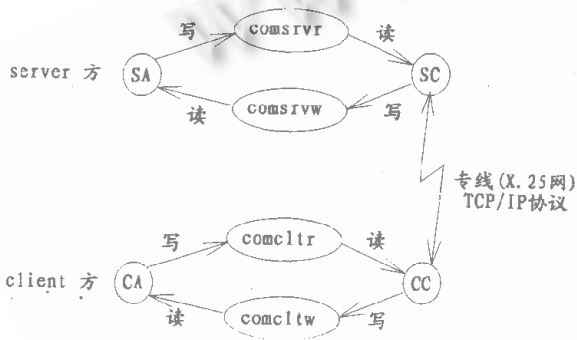


图 2

每一次通信写入流文件的格式如下:

标志位	内容
标志位	内容
.....	

说明:

·流文件以 80 个字符定长,信息内容若超过 80 个字节则通过续行再加 80 个字符。

·标志位为 1 个字节其中:

- 1. 新写入的第一行
- 2. 续行
- 3. 已被相应程序读出的第一行

与 Client/Server 有关的 C 语言函数有:

```
msgcltid() /* Client 方打开消息队列获取消息队列 id 号 */
msgsndctl() /* 通知 Client 通信程序,从 comcltr.st 读取委托 */
msgrcvctl() /* 监听 Client 通信程序发送消息,从 comcltw.st 读取委托结果 */
clientwrite() /* 委托内容送上级(写 comcltr.st) */
clientread() /* 委托结果接收(读 comcltw.st) */
msgsrvid() /* Server 方打开消息队列获取消息队列 id 号 */
msgsndsrv() /* 通知 Server 通信程序,从 comsrvr.st 读取服务结果 */
msgrcvsrv() /* 监听 Server 通信程序发送消息,从 comsrvw.st 读取委托 */
serverwrite() /* 服务结果送下级(写 comsrvr.st) */
```

```
serverread() /* 委托接收(读 comsrvw.st) */
```

应用程序 CA 的流程为:

- (1) 打开消息队列,并初始化。/* msgcltid() */
- (2) 从数据库中获取委托信息。
- (3) 将委托信息写入 comcltr.st 文件中。/* clientwrite() */
- (4) 通过消息队列通知程序 CC 读取委托信息。/* msgsndctl() */
- (5) 监听 Client 通信程序发送消息,从 comcltw.st 读取委托结果。/* msgrcvctl() */
- (6) 从文件 comcltw.st 中读取委托结果。/* clientread() */

(7) 检查委托返回结果并将委托结果写入数据库。

(8) 转第二步循环执行。

应用程序 SA 的流程为:

- (1) 打开消息队列,并初始化。/* msgsrvid() */
- (2) 监听 SC 发送的消息队列信息。/* msgrcvsrv() */
- (3) 从 comsrvw.st 接收委托信息。/* severread() */
- (4) 检查委托内容,若出错,返回出错代码,转第六步。
- (5) 若正确则根据委托内容为之服务并返回结果。
- (6) 服务结果或出错代码写入 comsrvr.st 文件。/*



serverwrite() */

(7)通过消息队列通知程序 SC 读取服务结果信息。/* msgsndsrv() */

(8)转第二步循环执行。

在开发过程中的几个问题和解决方法如下:

(1)重复交易处理:由于通信故障或系统突然崩溃造成服务结果没有正确到达 Client 方,而导致 Client 方发送重复交易(Duplicate Transaction),若不对这种情况进行管理则会发生买空卖空的事件。为了解决这个问题,我们通过以 Client 方的流水号和地址为关键字建立 Server 方的流水帐,并以 Client 方的流水号和地址建立唯一索引,当写入流水帐出现关键字不唯一错误时,则进行重复交易处理,并将第一次处理的结果发送给 Client

方。

(2)数据更动的复原处理:由于通信故障或者 Server 方系统故障,导致 Client 方无法获得处理结果,系统将进行超时处理(Time out)。即通过 INFORMIX 数据库提供的事务处理功能,将已更动的数据库进行复原处理(Rollback),以保证数据库的一致性。

4. 结束语

本系统开发完成后已在湖南省工行系统内全面推广,运行情况良好,经济效益十分明显,为湖南省工行的新型业务拓展方面作出了一定的成绩。

(来稿时间:1997年5月)