

**摘要:** 随着计算机网络的迅速发展和客户需求的变化,对系统的多机通信、软件复用及扩展升级提出了更高的要求。而 DCOM 以其版本化功能、二进制标准及 RPC 技术等优点为解决这些问题提供了全面的保证。本文讨论了 DCOM 技术的基本原理和主要特点,并提供了 DCOM 技术在水利监控系统中的一个应用实例。

**关键词:** COM DCOM 分布式监控系统 RPC 多线程

## 1 引言

在进行水库大坝分布式监控系统设计开发时,常常需要能够支持多个用户共享统一的数据库,并能允许在多个客户端操作现场设备。这就涉及到多台计算机间的通信,这些计算机采用以太网技术连接,传统的通信方法是采用套接字(socket)方法直接针对网络协议编程,用这种方法开发应用软件比较复杂,而且系统难以扩展升级以及与其他系统集成,不能适应现代水库管理的实际需求,如何及时适应用户需要,在保证可靠性的前提下实现系统扩展升级和集成已经成为一个用户很关注的问题。为了解决这一问题,我们采用当今最流行的组件技术COM/DCOM来实现处于不同计算机上的程序间的通信与代码复用,DCOM实际上是一项协议,它能让软件以一种安全、高效的方式通过网络进行直接通信,由DCOM来处理网络通信的底层细节问题,从而使我们能集中精力解决客户最关心的问题。

# 基于 COM/DCOM 技术的分布式监控系统的实现

赵明宇 徐立中 (南京河海大学计算机及信息工程学院 210098)

## 2 COM/DCOM

COM(component object model 组件对象模型)是微软公司提出的一种二进制通信规范,用于软件组件间跨越多个进程、计算机和操作系统进行互操作,它是一种开放式的跨平台的客户服务器开发技术,DCOM (Distributed COM,即分布式组件)扩展了组件对象模型技术,使其能够支持局域网、广域网甚至 internet 上不同计算机的对象间的通信,使用 DCOM 可以使应用程序在位置上实现分布性,从而满足客户和使用的需要。

COM 定义了组件服务器和客户之间相互作

用的方式,它使得组件服务器和客户无需中介软件就能相互联系,通过 COM 服务器提供的接口客户进程直接调用组件服务器中的方法,接口是集合在同一个名称下的相关方法的集合,组件间的通信是基于接口的,由于在操作系统中各进程之间是相互屏蔽的,当一个客户进程和另一个进程中的组件通信时它不能直接调用该进程,而需要遵循操作系统对进程间通信所做的约定,COM 提供了一种透明的方式来实现这种通信,即 COM 截取从客户进程来的调用并将其传送到另一进程中的组件,值得注意的是 COM 只是在建立客户和服务器之间的连接时才运行,一旦连接建立它就退出了<sup>[1]</sup>。

当客户进程和组件位于不同的机器时,DCOM 仅仅是用网络协议代替了本地进程间的通信,客户和组件都不知道连接它们之间的距离增大了,DCOM 建立了一种代理占位机制来实现不同机器上客户与组件间的通信,客户通过接口调用来访问代理服务器,然后代理服务器再与 RPC (远程过程调用)通信,RPC 使用底层网络协议(如 TCP/IP 等)来实现远程通信,在服务器端 RPC 与存根程序对话,然后存根程序再通过接口调用访问服务器<sup>[2]</sup>,DCOM 的网络体系结构如图 1。



图1 DCOM 网络结构图

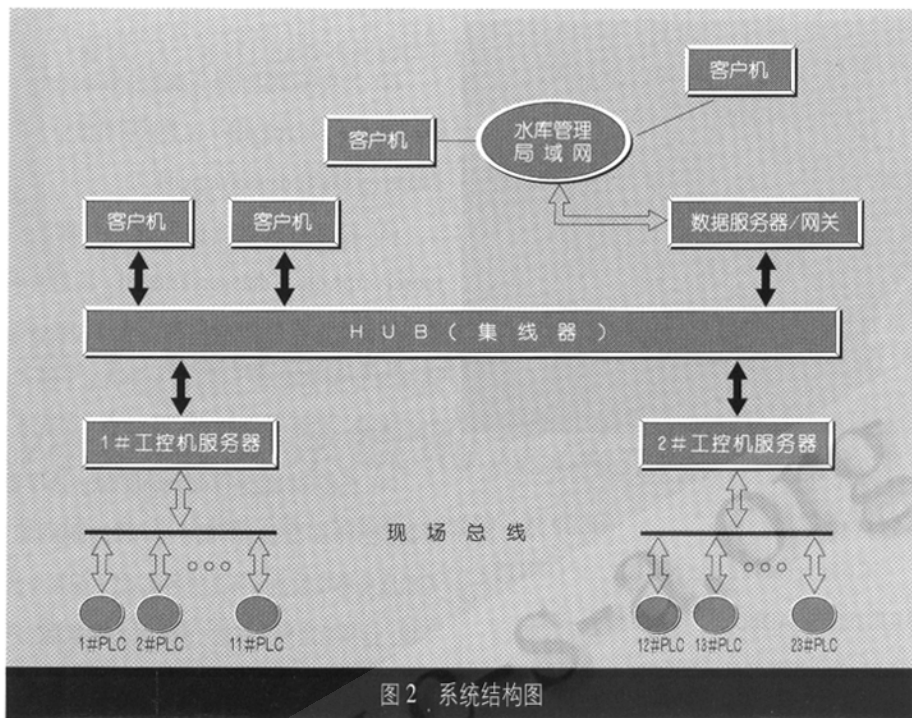


图2 系统结构图

另外 DCOM 运行库向客户和组件提供了面向对象的服务, 并且使用 RPC 和安全机制产生符合 DCOM 通信协议标准的网络包, 以此来达到客户和组件交互的目的。

### 3 大坝枢纽分布式监控系统组成与设计

#### 3.1 系统组成

本监控系统的组成是: 两台工控机服务器, 一台数据服务器, 多台客户机以及 23 个 PLC (可编程逻辑控制器)。系统结构如图 2 所示。图中两台工控机服务器用于监控现场的 PLC, 它们的功能包括获取受控设备(闸门)的实时数据并把数据存储到数据服务器上的数据库, 并具有向 PLC 发送控制命令对受控设备进行远程操作的功能。数据服务器上除了装有一个数据库系统之外, 同时还兼具网关的功能, 负责工控机与水库管理局域网中的客户间数据的转发。各个客户机负责从数据服务器获取数据以显示大坝的现场运行情况, 同时它们也具有向 PLC 发送控制命令实现远程操作的能力。

#### 3.2 设计方案

为实现以上功能, 软件设计方案如下: 工控

机服务器中的 DCOM 组件包含从 PLC 获取闸门的实时数据并存储到数据服务器上的数据库的功能和向 PLC 发送控制命令的功能。其中获取闸门数据并存储到数据库的功能仅由工控机上的 DCOM 客户调用, 该客户一直处于运行中以确保设备所有时段的数据都能被保存, 以便历史查询。

数据服务器端组件一个功能是负责从数据库获取数据以便在客户机上显示现场状态, 它通过 ODBC 接口与 SQLServer 数据库通信。当客户机调用该组件后, 客户机直接和数据库通信; 因为数据库组件能够把自己复制到客户机上, 并将自己与数据库相连接, 而此时客户机实际上已经不再和服务器的数据库组件相连了, 而是与该组件的一个本地副本连接着。组件另一个功能是根据当前设备数据进行分析计算, 为客户操作提供专家方案。

客户端应用程序调用工控机服务器和数据服务器上的组件, 为用户提供友好的人机操作界面, 实现设备操作的远程控制和现场状态的图形显示功能。

## 4 系统实现

在本系统中, 组件服务器端(工控机服务器与数据服务器)的操作系统为 window2000Server, 数据库系统是 SQLServer7.0, 组件客户端操作系统为 win2000; 采用的开发工具是 VC6.0, 其中组件开发采用 VC6.0 提供的 ATL 开发, 组件客户采用 MFC 开发。

### 4.1 组件服务器

我们以工控机服务器上的 DCOM 组件为例, 在这个组件中封装了一个 com 类 Cipc, 它提供了一个用户接口 lipc, 通过这个接口为 com 客户提供了两个方法: 一个方法是给 PLC 发送控制命令: sendorder(); 第二个方法是向数据库服务器中写入受控设备的实时数据: writetosqlserver()。

服务器端的编程步骤如下:

(1) 通过 ATL 建立一个 COM 组件项目 ipcserver。在项目中添加一个 ATL 对象。

在对象的属性栏内选择线程模型为“Free”, Free 是 MTA(多线程套间)的另一种称呼, 一个程序只能有一个 MTA, 在一个线程以 CoInitializeEx(NULL, COINIT\_MULTITHREADED) 初始化 COM 后, 当其他线程也以 CoInitializeEx(NULL, COINIT\_MULTITHREADED) 初始化 COM 后就会进入同一个 MTA, 也就是说在一个 MTA 中可以有多线程, 里面的线程可以自由使用组件支持的接口指针。

(2) 为这个 ATL 对象添加接口 lipc。

(3) 为接口 lipc 添加两个方法函数 sendorder() 和 writetosqlserver()。

(4) 为这两个方法函数编写具体的代码。在 ipc.cpp 文件中定义了两个方法函数 sendorder() 和 writetosqlserver()。

```
STDMETHODIMP Cipc:: writetosqlserver()
{ HANDLE hCom
Char * buf;
// 串口初始化
hCom=CreateFile("com2",GENERIC-
```

```
WRITEIGENERIC-READ,0,NULL,
OPEN-EXISTING,0,NULL);
.....
// 读串口, 获取运行设备数据
ReadFile(hCom,buf,length,&length,&m-
osRead);
.....
// 采用 ODBC 连接 SqlServer 数据库中的表
devedata 并添加新记录
m-pSet = &GetDocument()->m-dataSet;
.....
m-pSet->AddNew();
return S-OK;
}
```

#### 4.2 组件客户实现

DCOM 组件客户的代码编写与进程内组件客户的编程是相同的。客户端的编程步骤如下:

- (1) 采用 MFC 创建一个客户项目。
- (2) 在菜单栏中增加三个菜单项, 一个是发送命令菜单项, 另一个是显示数据菜单项, 第三个是专家系统菜单项。
- (3) 为这三个菜单项建立三个消息映射函数。
- (4) 具体编写这三个消息映射函数的函数体, 以其中发送命令函数为例。

```
void CClientView::OnLinkSendorder()
{ Iipc *pipc;
HRESULT hr;
CString order;
```

// 声明线程安全模式为多线程同时初始化  
COM 函数库

```
CoInitializeEx(NULL, COINIT-
MULTITHREADED);
```

// 建立组件实例, 并返回指向 COM 组件所支持的接口的指针: 其中接口指针的获取是通过 COM 函数库授权调用 Iunknown 接口的 QueryInterface () 函数获得的。

```
hr = CoCreateInstance(CLSID_ipc, NULL,
CLSCTX_SERVER,
```

```
IID_Iipc, (void **) &pipc);
```

// 根据 PLC 命令格式给 order 赋值

// 调用组件接口通过的方法函数。

```
hr = pipc->sendorder(order);
```

```
pipc->Release();
```

```
CoUninitialize();// 释放被引进的 COM  
函数库
```

```
}
```

#### 4.3 建立代理、存根程序 (以工控机组件为例)

只需要在“project ->setting”菜单中的“custom build”栏内的 commands 窗口中添加下面两条指令:

```
nmake ipcserverps.mk 和 regsvr32  
ipcserverps.dll
```

然后再进行编译即可得到代理、存根程序 ipcserverps.dll。

#### 4.4 系统安装及 DCOM 的配置

- (1) 首先在工控机服务器和数据服务器上为

客户开设一个用户帐号并设置密码和访问权限。

(2) 其次在工控机服务器上注册 ipcserver.exe 和 ipcserverps.dll; 在数据服务器上注册 dataserver.exe 和 dataserverps.dll。

(3) 在客户机端注册代理、存根程序 ipcserverps.dll 和 dataserverps.dll; 同时由于类型库包含在组件程序中, 所以我们在服务器端和客户端还需要注册 ipcserver.exe 和 dataserver.exe 程序。

(4) 在客户机端运行 dcomcnfg.exe。选择要配置的组件名, 配置运行组件服务器的位置即工控机服务器或数据服务器的名称; 然后配置身份标识, 选择指定用户按钮, 输入服务器端为客户开设的用户帐号及其密码; 然后在各客户机安装应用程序。

## 5 结束语

DCOM 作为一种优秀的组件技术可以有效地解决许多应用实践中的问题。在本例中, 我们通过采用 DCOM 技术的 RPC 通信机制和多线程支持解决了监控系统中的多机通信问题, 而通过其良好的软件复用和版本化功能, 就可以方便地实现系统升级以及本监控系统与其他管理系统的集成。■

#### 参考文献

- 1 David S.Platt 著, 信达工作室译《COM 精髓》人民邮电出版社, 2001.1。
- 2 Robert J.Oberg 著, 刘谦、苏建平译《深入学习: com + 高级编程》, 电子工业出版社, 2001.1。
- 3 Armstrong T.Patton R 著, 董梁、丁杰、李长业等译,《ATL 开发指南》, 电子工业出版社, 2001.11。