



# 基于Windows DNA 的Com+ 实现分布式地理信息系统

陈 澎 张 锦 刘国栋 (太原理工大学 030024)

|     |  |
|-----|--|
| 摘 要 | 如何建立高效, 稳定, 可扩展和安全的分布式地理信息系统是企业级应用的核心问题。本文在分析 COM+ 的特性和 Windows DNA 体系结构的基础上, 提出了一种实现分布式地理信息系统的实现框架总体论述。 |
| 关键词 | 地理信息系统 Windows 分布式互联网应用程序体系 组件对象模型 COM+ MTS  |

## 1 引言

随着地理信息系统的深入应用, 如何建立稳定, 高效, 易于维护和扩展的分布式地理信息系统以是人们关注的焦点问题之一。Windows 2000, 特别是COM+技术的出现, 为实现这一目标提供了很有效的技术基础。为此, 我们按照Windows DNA (Windows Distributed Internet Application Architecture)体系结构的思想, 主要结合 COM+ 技术等来构造分布式地理信息系统的实现。

### 1.1 Windows DNA

Windows DNA 是 MIRCOSOFT 公司提供的一种多层分布式应用程序模型。它可以分为表示层、业务逻辑层、数据层三层。而这些层是抽象的概念, 并不一定是物理结构, 每一层也可能含有多个物理结构。COM+ 表示 Windows DNA 中的业务逻辑层。它在整个多层分布式应用程序模型中具有核心地位。

### 1.2 COM+

COM+ 的概念产生于 1997 年, 其最初的目的是使基于COM的组件

开发、配置、服务变得更加容易。随着 Windows 2000 的发布, COM+ 已经成为一个相对稳定的结构了。基于COM+的基本结构, 我们可以认为COM+ 是COM、DCOM、MTS的集成, 它是基于COM基本技术, 是COM技术的新的发展和进步, 将应用层次由系统层提升到企业应用层。需要注意的是: COM+并不是简单的合并COM/DCOM/MTS技术, 而是一方面继承了许多特性, 同时另一方面又重新调整和完善了COM /DCOM的许多不足之处, 如优化了组件管理环境, 对装载均衡和对象的集中控制的支持, 简化了事件模型。在此基础上, 同时新增加了一些新的服务: 如内存数据库、队列服务、异步分发等(见图1)。

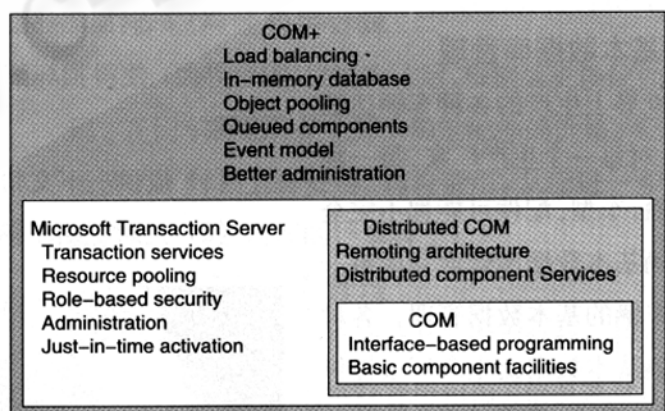


图 1 COM+ 组成结构

## 2 实现框架

### 2.1 硬件结构和配置

从系统的硬件结构来说, 可以分为表示层、业务逻辑层、数据层三层。其中表示层分为两种形式, 一种为基于浏览器的客户, 另一种为基于应用程序的客户, 业务逻辑层则由较为复杂的硬件结构组成, 目的是提高系统的可靠

性和稳定性。其中主要由两台集群路由器和二至八台的应用集群作为访问服务器,数据层可采用RAID,来提高系统的可靠性。

各层的软件配置是不同的,表示层中的浏览器端应有IE或NETSCAPE,应用程序端则有支持DCOM运行的基本环境和MSMQ组件,MSMQ INDEPENDNET CLIENT即可。业务逻辑层的配置相对高一些,每台集群路由器在Windows 2000 Advanced Server下配置常规路由和防火墙软件以外,还必须装入MICROSOFT CLUSTER SERVER。同时,需装入COM+装载平衡服务将配置成装载平衡路由器,并为定义相应的应用集群。访问服务器也是在Windows 2000 ADVANCED SERVER下主要配置COM+ EXPLORE、MSMQ组件、MSMQ ACTIVE DIRECTION SEVER、COM+ Event System、INTERNET INFORMATION SERVER服务等,数据层应有MICROSOFT SQL SERVER等。

## 2.2 软件结构和实现框架

### 2.2.1 表示层

一般而言,分布式的地理信息系统设计出发点是:系统通过一定的物理和逻辑结构,通过一定的软件和硬件结构,通过各个计算元素,能够稳定、高效、快速、准确、安全的为客户提供地理信息服务。根据客户端和服务器的计算特点不同,客户端常是些小量的、独立的、局部的数据计算,而服务器端则为大量的、相关的、全局的计算。为此,基于WEB的GIS客户我们采用ASP申请和本地脚本语言的辅助计算。APPLICATION的客户则利用DCOM和MSMQ机制进行通信,本地只是实现一些相应COM+事件和其他的少量计算。

客户的操作过程可分为三个阶段:登陆和身份验证、浏览信息、取消身份和离线等。基于WBE的GIS客户端不需要编码。而基于APPLICATION的客户端则需要构造用户界面和DCOM的建立和使用,实现COM+回调事件等。特别应该引起注意的是在创建DCOM对象时,不能使用DCOMCNFG.EXE去配置指定服务器名字,因为通过这种方式的设置,不能将此信息再传给其他的服务器,而不能实现动态负载平衡,我们必须在程序中指定服务器的名字。可以用CoCreateInstanceEx函数指定服务器名字而建立DCOM对象,同时由于该函数提供的多接口指针的功能,便于提高程序和网络通信效率。

### 2.2.2 业务逻辑层

而最为复杂的是业务逻辑层程序设计,以我们构造的软件框架为例,它必须实现不同客户端的响应和处理,以及集群功能,消息队列的处理功能,动态负载平衡,数据层的访问等。所以必须对它进行整体的认识,全局的规划,系统的设计。为了便于理解我们将业务逻辑层程序设计从概念上分为客户响应、系统底层协调、数据层访问三个部分。

(1) 客户响应部分。主要针对基于WEB的客户和APPLICATION的客户请求作出的响应。我们知道ASP(ACTIVE SERVER PAGE)可以直接在脚本语言中使用是由ASP.dll提供的APPLICATION,SESSION,ASPEROR,OBJECTCONTEXT,REQUEST,RESPONSE,SERVER等七个内嵌COM对象和几个组件。但对于一个分布的地理信息系统来说,利用专业的COM组件和COM+应用程序更有效,同时简化了ASP的开发过程。为此,我们须同时结合利用ASP

脚本和COM组件,作为COM+应用程序的组成,将COM+应用程序既可进行事物性处理,也可进行ASP处理。而ASP也可通过设置支持事务性处理。

如: <%language=Javascript transaction=required%> 这样便可提高系统的健壮性和扩展性。

(2) 系统底层协调部分。本层主要实现集群功能,消息队列和动态负载平衡等功能,是分布式地理信息系统的重要组成部分。利用这些机制,可以增强系统的可靠性,提高系统的稳定性。

① 集群功能:使用集群功能之前,必须首先定义MSCS集群资源,可被注册的资源有:应用程序,磁盘驱动器,IP地址和其他的服务器属性,而每个集群资源都登记了它所依赖的其他资源,当依赖的资源发生故障时,集群资源将从迁移到另一个节点上,从而实现故障屏蔽。被注册的集群资源以组的形式进行管理,每个组可以由一个或多个资源。而系统自定义了一个组被称为虚拟服务器,它包含IP地址和网络名资源,是实现故障屏蔽的基础。MICROSOFT CLUSTER SERVER是实现集群功能的系统软件,它包括几个组件:集群服务(必须运行在每一个节点上)、管理程序、可感知集群的程序、不可感知集群的程序、集群数据库、资源动态连接库、资源监视器。其中集群数据库是用来存储集群的配置信息,状态和属性数据,是实现迁移的数据基础。

实现分布式地理信息系统,需使用集群API。集群API分为三类:集群对象管理API,资源动态连接库API,集群管理扩展API。集群对象管理API可管理的对象分为组、节点、资源、资源类型、网络、网络接

口六种。一般通过使用控制码函数和控制码来管理这些资源,因为这样可使集群服务向资源动态连接库通知事件。资源动态连接库是通过两组接口与资源监视器和集群服务进行通信的。一个接口为入口函数,它是在资源动态连接库中实现的,可处理来自MSCS的请求。入口函数共有十个,而资源回调函数有LOGEVENT, SETRESOURCE-STATUS等。我们必须实现相应的入口函数和回调函数,才能使集群资源有效地屏蔽故障。在我们设计的分布式地理信息系统中,将路由软件和负载均衡服务定义为集群资源,旨在提高系统的寻址稳定性和均衡负载的稳定性。为此,将这两种资源定义到一个组中,以便于集中管理。通过重载入口函数和回调函数,实现了资源有效的迁移和故障的屏蔽。

② 动态负载平衡:为了在分布式地理信息系统中实现动态负载平衡,需要首先定义一个应用簇,然后配置动态负载均衡路由器。在我们设计的系统中,定义了由四台服务器组成的应用簇和两台集群路由组成的动态负载均衡路由器。当客户需创建COM+ OBJECT时,则SCM根据负载平衡引擎,按照默认或指定的算法来确定负载最小的服务器,从而实现动态负载。负载平衡引擎利用默认的算法是根据响应时间来确定负载的大小,但对于地理信息系统而言,并不能准确地反映出实际的负载量。我们提出了基于地理信息系统的负载判断算法:根据COM+ OBJECT的计算量(主要为图形、图象、数据分析等),内存消耗量,使用频度为判断依据的算法。为此,我们事先定义了原子的计算单位,作为COM+ OBJECT计算量的可比单位。内存消耗量是根据

一般情况下,组件的消耗最大内存量或对象池的消耗量为基础。而使用频度是根据SCM(SERVICE CONTROL MANAGER)截获的创建请求为依据进行计算的。而这三个判断依据的权值是根据系统的实时状态和性能反馈信息在一定的时间间隔内动态变化的,所以它更能有效的负载平衡。

### 2.2.3 数据层

对于WINDOWS的DNA体系中的数据层,MICROSOFT采用UDA(UNIVERSAL DATA ACCESS)技术,做为企业级应用提供了关系数据和非关系数据接口标准。UDA使用OLEDB对象模型,它是底层的数据访问模型,指定了事务和行集的定义。OLE DB是通过一系列COM接口来提供各种数据系统的访问操作和组件服务。而ADO(ACTIVE DATA OBJECT)是基于OLE DB面向业务逻辑层的简化模型,可以通过它访问OLEDB的大部分功能,是OLEDB应用者。但对于一个分布GIS来说,由于数据的复杂性,数据的存储形式是多样的,不可能将全部数据都以数据库的形式存在,还可能为文件系统,目录服务,电子表格等形式,所以必须实现OLE DB PROVIDER,而不能只使用ADO。

在OLE DB对象模型中定义了,枚举器(enumerator),数据源(data source),对话(session),命令(command),行集合(rowset),索引(index),事务处理(transaction),错误(error)等对象。我们利用这些对象才能有效的访问和编辑数据源。对于一个分布式的地理信息系统而言,存在着各种信息形式和各种数据管理方式,但大体上可以把它们分为三类:

(1) 地理空间信息和属性信息。即为传统意义上的地理信息,是地理

信息系统的核心内容。而存储形式和方式在实际上也是多种多样的,常用是数据库(几何图形库、地图符号库、空间分析算法库等)、文件系统(自定义文件、标准文件、混合文件等)、数据库和文件系统的混合等形式。

(2) 地理信息分析模板信息。多以数据库的形式存储,便于扩展与修正。

(3) 用户和管理信息。包括用户的个人资料,密码,使用记录,系统LOG文件,访问统计文件等。这些信息既需以数据库的形式存储,也需结合文件系统。从中我们可以分析出,数据库和文件系统是地理信息系统现在常用的存储形式,所以我们须利用OLEDB的DATA PROVIDER来实现数据库的PROVIDER和文件的PROVIDER。

## 3 结束语

本文通过阐述COM+的技术特点,按照WINDOWS DNA论述了分布式地理信息系统的实现框架,分别从硬件和软件的角度进行了分析。特别是对表示层,业务逻辑层,数据层的实现做出了较为详细的描述。同时提出了符合开发地理信息系统的方式,为开发稳定、高效、可扩展的分布式地理信息系统提供一种有效、实用的途径。■

### 参考文献

- 1 Guy Eddon, Henry Eddon. Inside COM+ Base Services, Microsoft Press, 1999.
- 2 Mikey Williams 著,前导工作室译, Windows 2000 编程技术内幕,北京机械工业出版社, 1999。
- 3 潘爱民著, COM 原理与应用,清华大学出版社, 1999。