

C#实现分布式数据库查询 PDF转换可能丢失图片或格式，建议阅读原文

https://www.100test.com/kao_ti2020/459/2021_2022_C_23_E5_AE_9E_E7_8E_B0_E5_88_c97_459820.htm

随着传统的数据库、计算机网络和数字通信技术的飞速发展，以数据分布存储和分布处理为主要特征的分布式数据库系统的研究和开发越来越受到人们的关注。但由于其开发较为复杂，在一定程度上制约了它的发展。基于此，本文提出了在.Net环境下使用一种新的开发语言C#结合ADO.Net数据访问模型来开发分布式数据库系统，大大简化了开发过程。

1 分布式数据库系统就其本质而言，分布式数据库系统的数据在逻辑上是统一的，而在物理上却是分散的。与集中式数据库相比它有如下主要优点：

- 解决组织机构分散而数据需要相互联系的问题。
- 均衡负载。负载在各处理机间分担，可避免临界瓶颈。
- 可靠性高。数据分布在不同场地，且存有多个副本，即使个别场地发生故障，不致引起整个系统的瘫痪。
- 可扩充性好。当需要增加新的相对自主的组织单位时，可在对当前机构影响最小的情况下进行扩充。

分布式数据库系统虽然有诸多优点，但它同时也带来了许多新问题。如：数据一致性问题、数据远程传递的实现、通信开销的降低等，这使得分布式数据库系统的开发变得较为复杂。幸运的是，微软的.Net开发环境为我们提供了C#开发语言和ADO.Net数据访问模型，结合两者来开发分布式数据库系统能够大大简化开发工作。

2 远程处理框架和ADO.Net 开发分布式数据库系统需要解决的两个重要问题是：各场地间的数据通信以及对数据库的操作及管理。使用C#结合ADO.Net能够高效、可靠地解决这两方面的问题

。具体表现为，在C#中通过使用.Net远程处理框架能够方便地解决数据、命令远程传递问题；C#通过ADO.Net对数据库进行操作，使分布式数据库系统中对数据库的各种操作变得高效、可靠，同时易于解决数据一致性问题。

2.1 .Net远程处理框架

实现数据和命令的远程传递有三种方式。第一种是使用报文或消息的方式，把要传送的数据转化为流格式，再通过套接字编程用报文的形式发送到远程主机。此种方法麻烦，不易实现。第二种是使用Web Service，即各远程主机提供一个数据库查询服务的Web Service。这种方式只能对单个场地进行查询，无法实现多场地的联合查询。第三种是使用.Net远程处理框架（.Net Remoting Framework）技术，它将远程调用的技术细节隐藏起来，服务程序只需通过简单的设置就可以把本地对象变成为远程提供服务的远程对象，客户端可以像访问本地对象一样透明地访问远程对象，所有的消息、报文等都交给.Net Remoting对象处理，大大简化了开发。

远程处理的一般过程如图1所示：首先，服务器端创建一个服务器类的实例，远程处理系统创建一个表示该类的代理对象，并向客户端对象返回一个对该代理的引用。当客户端调用方法时，远程处理基础结构连接检查类型信息，并通过信道将该调用发送到服务器进程。侦听信道获得该请求并将其转发给服务器远程处理系统，服务器远程处理系统查找（或在必要时创建）并调用被请求的对象。然后，此过程将反向进行，服务器远程处理系统将响应捆绑成消息并由服务器信道发送到客户端信道。最后，客户端远程处理系统通过代理将调用的结果返回给客户端对象。

2.2 ADO.Net

ADO.Net以Xml为核心，是.Net数据库应用程序的解决方案。它使用离

线数据结构，数据源中的数据被缓存到数据集（DataSet）对象中，用户无须锁定数据源，数据以Xml格式保存。 2.2.1 ADO.Net管理数据一致性 在分布式数据库系统中，很可能出现多个用户同时访问和修改数据的情况，因此，对于分布式数据库系统，数据一致性是不可或缺的。ADO.Net通过使用乐观一致性方案来控制数据一致性（实际上DataSet对象被设计成支持使用乐观一致性控制机制），即数据行只有在数据库中真正被更新时才会被锁定，而在悲观一致性方案中，数据行在从被提取出来到在数据库中更新这段时间内一直被锁定。因此，使用ADO.Net能够在更少的时间响应数量巨大的用户。另外，在分布式数据库系统中，还会经常遇到当用户修改自从提取出来以来已经被修改的行时，违反一致性原则。对此问题ADO.Net也作了很好地解决，即使用DataSet对象为每一条修改过的记录维护两个版本：原始版本和更新版本，在更新的记录被写回数据库之前，先要把数据集中记录的原始版本与数据库中的当前版本进行比较，如果两个版本匹配，就在数据库中更新记录；否则，就会出现违反一致性原则的错误。 100Test 下载频道开通，各类考试题目直接下载。详细请访问 www.100test.com