**摘要:** 本文讨论了某公司发货系统的分布式数据库集成解决方案。该公司由于业务的发展，要在另三个城市设立货仓进行发货。为此，需要增加原先的MIS系统实现这一功能。公司委任我作为项目经理完成系统的设计和开发的工作。我经过分析，使用了Sybase的分布式数据库技术。我设计的这个系统是采用典型的C/S结构，但客户端连接服务器的网络采用电话线拨号，速度有限，传统Windows界面的客户端应用程序相应速度比较慢。于是我采用了优化数据库结构的方法，把数据分两部分存放，基础数据存放于客户机上，销售资料主要采用键码存放于服务器上，应用程序再现数据时从服务器取键码，到客户机取对应的解释。由于键码的数据量少，网络传输便快。在构建这个公布式数据库系统的过程中，我着重研究并解决了数据同步和事务协调的问题，到得了良好的应用效果。

2009年3月，由于公司业务的发展，要求在其它三个城市设立货仓，处理发货业务。公司本部运行着一套用Sybase数据库的MIS系统可以实现发货，该系统用的是C/S结构。由于客户端连接服务器的网络采用电话拨号，所以直接把客户端软件直接安装在外地访问本部数据库，速度很慢。于是，公司成立了一个项目，专门解决这个问题。在这个项目中，我担任项目经理。经过对现有系统的分析，我们决定利用Sybase提供的技术，采用分布式数据库集成的方法来改造目前的系统使之能适应新的需要。项目分三个阶段进行，一是进行需求分析，确定要增加的功能。二是进行系统设计，改变后数据分布如何，系统架构如何。最后是实现和测试，上线。整个项目历时从分析到实现历时三个月，最后于2009年6月份系统成功上线。   
在分析阶段时我发现由于客户端地域的分散，遍及三个省境内，连接服务器数据库的网络采用电话拨号方式，速度有限，在使用客户端应用程序时感觉界面速度很慢。我经过分析，认识到许多操作都要从服务器中取数据，速度慢就慢在数据访问上。服务器是没有瓶颈的，问题出在网络速度上。出于成本和业务量方面的考虑，公司不会用专线连接，只能是电话拨号。这时只能改变目前软件的实现方法，来适应这种低速网络的使用模式。 经和项目组的人员一起探讨，结合关系数据库的知识，我认识到，应用程序的每一次数据库操作，都要访问多个相联的表，其中，有销售订单表和物料基础数据表/客户资料表/货仓的基础数据等。销售订单表中存放着出销售的订单编号，成品编号等，数据量少。而基础数据表就则放着成品的相关信息，有大量的数据。如果考虑把销售订单放在服务器，基础数据放在客户端，当应用程序中访问数据时，总是从服务器上存取销售订单，从客户端提取成品/订单的详细信息。由于订单的数据量少，便减少了网络上传递的数据量，从而提高了界面的响应速度。   
把数据分散存放只是工作的第一步，接下来要考虑应用程序怎样访问这种分布式数据。开发应用时，如 果每一功能都针对两个数据库进行，就带来了很多麻烦。所以，我通过研究Sybase的分布式数据库技术，决定采用CIS（组件集成服务）部件，来合并两个数据库成一个统一的分布式数据库。应用程序只要连接一个数据库，就可以透明统一访问到两个数据库中的数据。 该技术具体实施方法是：在客户端数据库中建立一个对服务器数据库的远程访问服务名，包含访问地址，登录用户名，登录密码等关键的连接信息；前且对服务器中销售订单建立一个本地代理表。结构和服务器中远程表完全一样，它是访问服务器中会员资料的中转和代理。客户端应用程序访问本地代理销售资料表时，实际上是通过预先定义的远程访问服务名中包含的连接信息到服务器中对应的实际销售资料表中访问数据。这种访问对于客户端完全透明，感觉不到是从物理上独立的两个服务器中存服数据。所以，这种数据库结构是典型的分布式数据库。   
部署这种分布式数据库不是难事，只要在客户端和服务器上安装12.0版本以上的数据库服务器，在客户端服务器上建立远程服务名和代理表即可。由于Sybase数据库的安装支持脚本方式，在客户端应用程序的标准安装过程中，嵌入Sybase数据库的安装和配置脚本，就自动化地完成了所有工作。   
在实际使用该分布式数据库系统的过程中，遇到了几个问题，第一，数据同步。客户端基础数据不是绝对静态的，也有变化，因此在服务器要设置一个统一的基准，称为主点数据。客户端总是要复制使用，称为复制点数据。如何及时感知到服务器端主点数据的变化，有效率地复制到客户端，是个难题。Sybase针对这种应用场合，提供了复制服务器技术，但为了避免过于复杂，我们采用实际应用程序来管理同步。当服务器端主点数据有了更改时，保存一个相应的标识和时间戳，客户端应用在登录服务器时，检查这些标识，一检测到了数据有更新，就首先下载，然后再进入系统正常使用。这种方法实现起来，增加了额外的开发量，且不能判别绕过应用程序对数据的直接修改，但是，是最简单和有效的方法。 第二个问题是事务协调问题。物理上独立的两个数据库，在协同操作时，如果服务器正好停机或者网络故障，完整的一个事务没能完成，就会“事务崩溃”。虽然Sybase CIS内嵌了两阶段提交技术，能够自动恢复。但是应用程序在这种情况下，敏感性不够，操作界面会无端凝固，影响了使用的方便性。我针对PB对劲于连接的判断和感知，用了一个小小编程技巧，使应用程序能够及时感知到数据库连接故障，及时停止和恢复事务，使操作界面表现友好灵活。   
在具体的应用中,我们在三个城市安装了增强的客户端应用程序,同时安装了Sybase数据库。初始化时，把基础数据放从公司本部的数据库导入客户端的数据库中。用户在外地进行发货时，先拨号上网，然后启动客户端程序。在登录过程中，客户端程序会检查服务器上的标识和时间戳检查这些主数据是否有更新，如果有就先下载，下载完成后再进入系统正常使用。在服务器更新的数据比较多的情况下，下载的时间会比较长，这时如果遇到急需发货，则会影响到货物不能及时发出去。为了解决这个问题，我设置了在每天凌晨的某个时刻自动登录和启动客户端程序，在下载更新数据完成后自动关闭。这样可以把一部份数据更新的内容放在非工作时间里完成，减少了发货登录的时间。用户登录后开始进行发货操作。输入销售订单，通过本地代理表系统自动到服务器获取该销售订单的数据，如发货的数量，客户编号等。而一些基础数据则可以直接从本地的数据库中得到，如销售产品的描述，客户的地址/电话/传真，发货的库位等。完成出货动作后，会自动更新服务器的库存。而这一更新通过提交事务在后台进行，不影响前台的操作。所以，对用户来讲，能够进行正常的操作，不会因为速度慢而进行不下去。   
在当今的信息社会里，互联网带来了相互连通的方便，而且知识爆炸，数据的分布式访问是个必然的趋势。目前新起的XML技术，提供了各种平台数据库之间的一个公共数据访问标准，可以用来构建更加灵活，适应性更强的分布数据库技术。将XML用在分布式数据库中，将是未来的一个趋势。