

数据库中间件的结构分析

李冕 张佐 吴秋峰

清华大学自动化系 100084

摘要: 在企业网络应用日益发展的今天, 实用的数据库技术越来越和企业网络相联系, 将数据库与web技术相结合的数据库中间件技术应运而生。相比已往的连接技术, 中间件思想和方式具有相当大的优势和先进性。本文对一种实用的数据库中间件产品进行了分析, 建立一般的数据库中间件模型, 并从模型的角度将它和已往的CGI技术作了比较。相比而言, 数据库中间件结构简单、功能完善, 使数据库应用效率大大提高。

关键字: 数据库、数据库中间件、CGI、模型

1 企业网络和 Intranet 下的数据库应用

随着 Intranet 和企业管理信息系统 (MIS) 的发展, 数据库与计算机网络之间的联系和应用越来越紧密, 随着应用的深入和趋于复杂, 对数据库访问时的功能和性能要求就越高^[1]。基于 Web 应用的数据库技术日趋先进, 简单的三层结构已经不能满足网络发展的需求, 数据库中间件技术的出现是对传统的三层结构的有力扩充。本文提出一种将数据库中间件应用在一般中、小企业网络中的方法, 作者认为其结构简单、功能完善, 使数据库检索效率大大提高, 从模型角度而言这种三层结构对比已往的结构具有较大优势。

2 Web 数据库应用中的中间件技术

本文从基本结构的角度讨论了一种以 DB(database) 中间件形式在 Intranet 中应用的新方法。在这种结构中, 系统通过对数据库连接和访问机制的管理, 改善网络上多用户访问数据库的性能, 优化网络传输。

2.1 目前, 实现 Intranet 上对数据库访问的接口大致可以分为以下 3 种^[2]:

2.1.1 通用网关接口 CGI(common gateway interface)

CGI 最大的优势就在于, 几乎所有的 Web 服务器都支持 CGI 程序。但是, CGI 接口也存在着一些不足, 其中最主要的就是性能问题, 特别是在需要多个数据库连接的多用户应用程序中尤其明显。CGI 程序不能被多个客户请求共享, 当一个新的请求到达时, 即使 CGI 程序有一个实例在运行, 也必须再启动一个新的实例。并发

请求越多, 服务器上创建的并发进程也越多。为每个请求创建一个应用进程会消耗较多时间且需要大量内存, 同时限制了应用程序自身可用的资源, 最终导致性能降低, 增加等待时间。在本文中, 作者将通过分别对 CGI 方式和中间件方式建模而比较出两者的不同。

2.1.2 专用 API(application programming interface)

针对 CGI 程序的上述不足之处, 为了提高 Web 服务器与数据库服务器之间的通信效率和性能, 各大 Web 服务器厂商和数据库厂商纷纷推出各自的专用 API。

与 CGI 程序是作为独立的进程运行不同, 专用 API 都是以动态链接库的形式存在的, 其效率和性能较之 CGI 程序有大幅度的提高。但是采用这种类型的 API 也存在一些缺陷, 例如, 各种 API 相互之间不兼容; 它们只能在特定的服务器和操作系统上运行; 由于采用了动态链接库的形式, 一旦代码质量较差就比较容易造成服务器系统的崩溃; 并且进行程序设计时会更复杂。

2.1.3 Java 数据库互连 JDBC(Java database connectivity)

为了克服 CGI 程序访问数据库效率低和专用 API 可移植性差的缺陷, 新近出现了 JDBC。JDBC 是用来执行 SQL(structured query language) 语句的 Java 应用程序接口。它由一组用 Java 程序设计语言写成的类和接口组成, 易于向任何关系数据库发送 SQL 语句, 支持对多种数据库的访问。运用 JDBC API, 只需写出单独一个程序就能够发送 JDBC API 给相应的数据库。Java 语言的中性结构和 JDBC 的开放性使得程序员只需写一次程序就能让它到处运行。JDBC 是一个低层次的 API, 它是更高层次 API 的

基础。开发人员可以在JDBC的基础上,设计出用户更容易理解、便于使用的更高层次的接口和工具。

2.2 数据库中间件的设计思想

数据库中间件的基本结构如图1所示。其中数据库中间件 UniWeb Server, 是前端的客户与后端的数据库之间进行通信的桥梁。当客户向 UniWeb Server 发出对某个数据库的 SQL 请求时, UniWeb Server 搜索当前可用的与该数据库的连接(UniTcl Server), 通过 UniTcl Server 将 SQL 请求转发给对应的数据库服务器。数据库服务器执行 SQL 语句后, 将结果通过 UniTcl Server 返回给 UniWeb Server, 再由它返回给客户。整个数据库中间件的体系结构采用的是三层(Three-tier)客户机/服务器模型。中间件与各个客户的数据通信采用流套接字(Stream Socket)机制实现, 并且利用了多线程的优点, 使得中间件与各个客户的套接字通信能够并发地进行。中间件与后端数据库的接口使用了 UniTcl, 并且根据 Internet/Intranet 上用户对数据库访问方式的特殊性, 采用多客户共享同一个 UniWeb Server 的机制, 提高了数据库访问的效率。

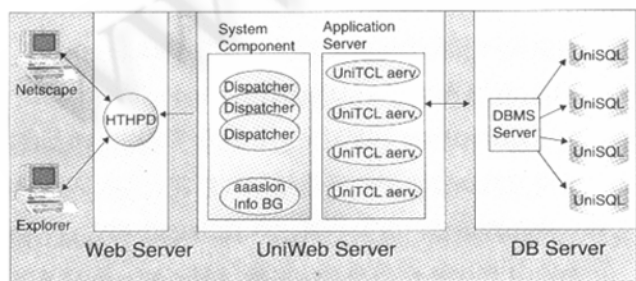


图1 数据库中间件的基本结构

2.3 数据库中间件 UniWeb Server 的实现

通过把 Web 服务和高效的数据库管理系统的数据服务结合在一起, 可以组成大型的多媒体数据库管理系统。UniWeb 是一种能够把 DBMS 和 WWW 结合在一起的中间件系统, 它能够使 DBMS 达到最优化, 并且为将 Tcl 嵌入 HTML 文档的 Web 应用提供开发环境来实现中间件的功能。特点:

- (1) 提供嵌入 HTML 文本的 UniTcl 的 Script 描述语言, 不用进行 CGI 编程, 提高 Script 描述语言处理能力和编程效率, 减少维护消耗;
- (2) 具有系统状态管理功能, 优化用户查询进程;
- (3) 支持并优化数据库事物处理功能, 能够防止突发事件的发生对数据库性能的影响;
- (4) 多个事务处理(transaction)可在一次连接(session)

中完成, 一次连接可以控制多个 HTML 文档中的事务处理和连接, 具有 OLTP 的功能。

3 中间件与 CGI 的比较

3.1 CGI 的应用进程及模型分析

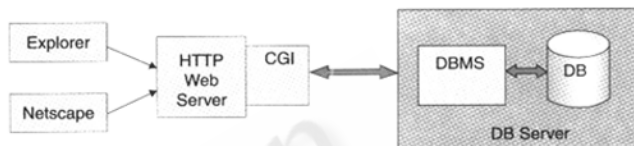


图2 CGI的基本结构

通常的 CGI 的应用对于数据库应用而言是一个单一的排队问题, 其结构如图 2。从排队模型的角度看, 是一个单一服务器可同时完成多个服务的排队问题, 模型如图 3。

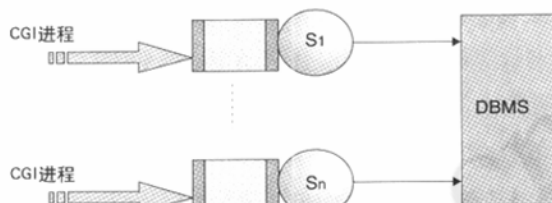


图3 CGI排队模型

在 CGI 的应用方式中, 应用往往会受到数据库用户数的限制: 通常的 DBMS 服务器都定义了一个用户数 n , 通过这个用户数限制了同时连接到 DB 服务器的进程数量。当与 DB 连接的用户数量达到这个上限时, 新的用户连接请求就会失败。同时, 系统灵活性比较低, 无法根据用户的访问量及时调整系统参数。在 Intranet 应用中, 网络上会出现大量用户同时访问一个数据库服务器的情况, 并且每个用户在一段时间内会保持这种服务连接, 大量耗费了系统资源, 从而限制了可用的连接数。但是, 当用户在进行客户端的处理时, 与 DB 的连接通常是空闲的, 如果能够充分利用这些时间, 使得这些连接能够为其他用户提供服务, 那么, 就能大大提高同时服务的用户数。如果在用户不使用这些数据库连接时就断开 DB 连接, 虽然可以提高 DB 服务器的多用户服务能力, 但是建立 DB 连接通常都是相当费时的操作(可能包括建立某种通信连接、启动服务进程、权限检查等等), 这样会降低数据库服务器的服务响应速度, 在实际使用时效率较低。

3.2 中间件的应用进程及模型

本文使用的中间件服务器是 Korea computer communication 公司提供的第三方中间件产品 UniWeb Server。

UniWeb 提供分布式的环境可以使 Web Server 和 database Server 运行在不同的机器上。图 4 显示了这种情况，它与单机运行是同样的。这时数据库和应用文件应运行在 UTS Server 上。

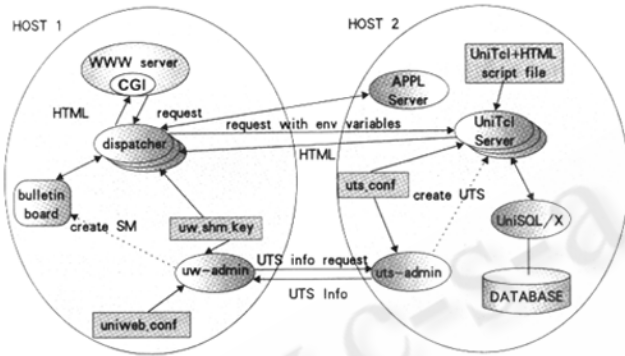


图 4 UniWeb 结构

Components that consists of UniWeb

Web server side	UTS server side (DB side)
Uw-admin	uts-admin
Dispatcher	uts (UniTcl Server)
Uniweb.conf	uts.conf
Uw.shm.key	getsize
	uniweb.key

表 1 UniWeb 中间件内容

UniWeb 包括以下部分：

- (1) uw_admin: uw_admin 通过 uts_admin 管理 UniTcl Server (UTS)。
- (2) uts_admin: uts_admin 用 UTS 的信息来响应 uw_admin。一个调度程序 (dispatcher) 在 Web 服务器和 UTS 之间传送数据。当客户端请求时，调度程序把请求送给 UniTcl Server，然后把从 UniTcl Server 得到的结果返回给 Web 服务器。
- (3) 公告板 (Bulletin board): 用于控制 UniTcl Server 的状态和数目并选择空闲的 UniTcl Server。
- (4) UniTcl Server: UniTcl Server 阅读客户端的请求，当其中包含一个数据库操作时，它与数据库连接并把结果送给调度程序 (dispatcher)。由于 UniTcl Server 作为一个

后台通讯进程与数据库保持持续的连接状态，所以它能够对数据库查询 (query) 进行优化，具有对象缓存功能。

UniWeb 的服务器进程说明如下：

- (1) 一个调度程序从 Web 服务器接收客户端请求。
- (2) 调度程序把客户端请求传送给根据公告板指定的 UniTcl Server。
- (3) UniTcl Server 解释请求的脚本程序，和数据库连接，把结果送回调度程序。
- (4) 调度程序把结果送回 Web 服务器。

我们不难从以上描述中得到中间件的排队模型如图 5 所示，相对于 CGI 模型而言，它增加了一层中间服务器，形成一个两层多服务器的串行结构，并且中间服务器的数目可根据系统的状态进行调整。图中的 S0 是调度程序，S1 到 Sn 是中间件服务器中的 UniTcl Server 的数目。在每个 UniTcl Server 中有一个缓存队列，记录相应的请求，其可容纳的请求数也可以根据系统的应用状况作相应的调整，并且每个 UniTcl Server 与 DBMS 的连接是建立好的，不需要在建立连接方面耗费系统资源。同时，由于具有对象缓存功能，它在接受一个请求后，不被这一请求独占资源，还可以接受另外请求，就节省了请求等待时间。

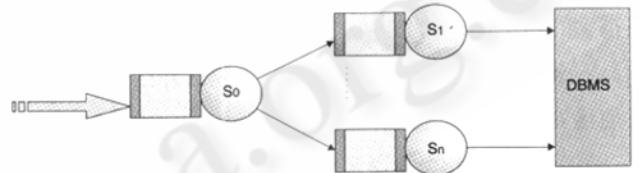


图 5 数据库中间件模型

这种中间件的设计思想就是使用独立的 DB 连接管理进程 (UniWeb) 提供连接管理的服务，客户不是直接与 DB 服务器建立连接，而是通过 UniWeb 建立连接，各种 DB 操作也是通过 UniWeb 完成的。

概括起来，该数据库中间件完成了下列功能：

- (1) 有效改善 Intranet 上多用户对数据库的访问性能；
- (2) 减少网上信息的传输；
- (3) 优化了网络传输性能；
- (4) 既支持类似 Web 中的无状态和一次连接，也支持应用有状态和持续性的连接；
- (5) 为 Intranet 应用中与数据库的连接提供一条功能强大而又灵活高效的途径。

(下转第 11 页)

4 小结

可以看出,基于以上分析和研究,基于数据库中间件方式的数据库应用系统较之已往的应用方式具有很大的优势。从模型角度考察,中间件的模型在增加了一层 UniTcl Server 服务器后,节省了进程与数据库服务器建立连接的资源和时间,显著的提高了系统效率,其结构简单、灵活,可根据系统的运行状态作出相应的调整,具有很高的灵活性和极高的效率。同时大大提高了并发用户数,适合在网络环境下的应用。相比于传统的 CGI 模型,从它的模型本质而言就具有根本的不同和优势。从作者本人对这两种模型的仿真比较中,如果对于在 Intranet 环境下,同时可能的并发用户数较大的情况下,中间件模型的效率要比传统模型的高出 10 ~ 100 倍左右。■

参考文献

- 1 马松,盛浩林. Intranet 上数据库中间件原型的研究与构造. 软件学报, No.1, 1999。
- 2 宋晓梁,刘东生,许满武. 中间件及其在三层客户机/服务器模型中的应用. 计算机应用, Vol.19,NO.7 Jul,1999
- 3 Alexis D. Gutzman, SELECTING THE APPROPRIATE MIDDLEWARE FOR YOUR WEB-TO-DATABASE APPLICATION, ACM SIGUCCS XXV 1997。
- 4 张伟华. 中间件在远程调用技术的研究及在 TreeSVR 中的应用. 上海铁道大学学报, 第 20 卷第 10 期, 1999 年 10 月
- 5 王辉,施小英. 中间件服务及其集成框架. 计算机工程与应用, 1998 年 9 月