第34卷 第23期 计 算 机 工 程 2008年12月
Vol.34 No.23 Computer Engineering December 2008
·软件技术与数据库· 文章编号：1000—3428(2008)23—0079—03 文献标识码：A 中图分类号：TP391

NGN业务系统中异构数据库中间件的设计和实现

宋述燕，尹建新，王助娟
(中南民族大学电子信息工程学院，武汉 430074)

摘 要：针对下一代网络(NGN)业务开发过程中业务提供商需要和电信运营商的各种异构数据库进行通信的情况，设计和实现一种基于异构的数据库中间件。采用分层结构和多线程的方式，并对数据访问请求进行统一调度排队，从而有效地提高数据访问的安全性、灵活性和可扩展性。目前该中间件已经大量地应用于在线的NGN业务系统中。

关键词：中间件；调度；人工业务；自动业务

Design and Implementation of Heterogeneous Database Middleware on NGN Service System

SONG Shu-yan, YIN Jian-xin, WANG Zhu-juan
(College of Electronic Information Engineering, South-Central University for Nationalities, Wuhan 430074)

【Abstract】 In the development process of Next Generation Network(NGN) service, service provider needs to communicate with various heterogeneous database belonging to telecom operators, so it is required imminently to design and implement a heterogeneous database middleware. It adopts layer-framework and multithreading method, queues and schedules data access request uniformly, therefore it improves the security, flexibility and expansibility on data access, and now this middleware is extensively applied to on-line NGN service system.

【Key words】 middleware; scheduling; manual service; auto service

目前，国内各大电信运营商的数据大都应用商业的大型数据库来存储和管理，如MS sql server, Sybase, Oracle, informix等。其客户端访问接口各不相同，因此，当NGN业务提供商开发的业务需要访问运营商的数据时，需要和不同的数据库打交道，而第三方提供的一些数据库访问中间件还要经过二次开发才能满足要求。为了节省成本，同时提高灵活性和安全性。本文设计和开发了一套异构的数据库中间件系统[1]。

1 异构数据库中间件的设计与实现

1.1 系统结构

在NGN业务中，具体业务一般不能直接去访问运营商的数据库，因为成百上千的客户端同时访问，很可能导致数据库反应超时，甚至崩溃。而数据库中间件可以提供安全访问的机制，所有来自客户端的请求都要在这里统一调度排队，通过数据库中间件向运营商数据库发送请求，然后把运营商数据库的应答返回给客户端。数据库中间件可以控制并发请求的数量，可以根据数据并发访问的数量作出实时的调整，保证数据库的负载均衡。

通常把 NGN 业务系统中，业务通过数据库中间件访问各种异构数据库，并且获取数据的过程划分为 4 层，如图 1 所示。具体描述如下：

(1)业务层

业务层即具体的 NGN 业务，根据是否需要人工参与，分为自动业务和人工业务。

(2)接口层

接口层指支持各种业务和数据库中间件之间通信的接口，业务可以直接和数据库中间件通信，比如自动语音业务(通过 TCP/IP 协议)，这样可以不需要接口层。其他如人工业务，需要通过 Web 网页方式，访问数据库，需要开发一些接口如 javabean 控件来和数据库中间件通信。

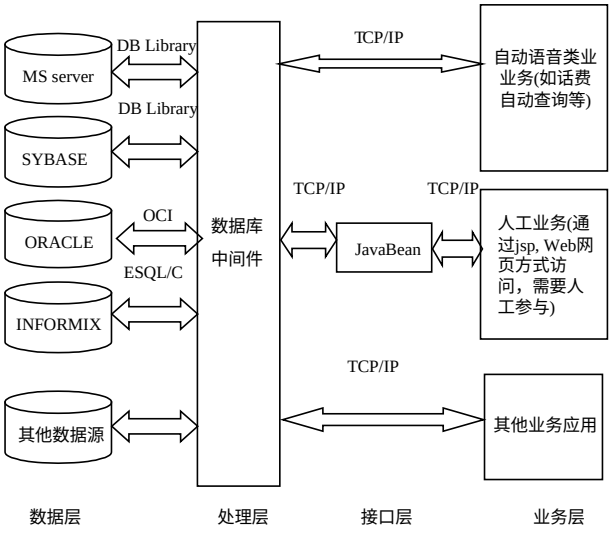

图 1 数据库中间件系统结构图

(3)处理层

处理层即数据库中间件，主要完成各种业务的数据访问和获取数据的请求，对数据库请求进行调度排队，控制并发数据连接数量，并获取数据分发给业务。这样就对业务层屏

作者简介：宋述燕(1976－)，女，讲师、硕士，主研方向：通信与计算机；尹建新，副教授；王助娟，硕士

收稿日期：2008-03-22 **E-mail：**song_shuyan@163.com

蔽了数据源的细节，因而在开发业务时可以不必考虑数据库具体结构带来的影响。

(4)数据层

数据层即存储各种业务数据的数据库，数据库中间件需要支持当前多种大型主流数据库，且使用各种主流数据库提供的客户端接口与之通信，这样使得业务访问数据的效率更高，如使用MS SQL提供的DB LIBRARY访问MS SQL数据库，使用SYBASE提供的DB LIBRARY访问SYBASE数据库，使用ORACLE提供的OCI访问ORACLE数据库[2]，使用INFORMIX提供的ESQL/C访问INFORMIX数据库等。

1.2 模块设计

数据库中间件主要由 6 个模块组成：SQL 预处理模块，请求调度模块，请求处理模块，结果分发模块，连接监控模块，分级日志模块。其中，自动语音类业务的数据请求进入程序的主线程以后，交给 SQL 预处理模块，读取相关参数后，进行有关 SQL 语句的组装，然后进入请求调度模块；人工业务的数据请求则直接进入请求调度模块。在请求调度模块里，主要对数据请求进行调度排队，控制数据访问的并发数量，然后交给请求处理模块进行处理。处理模块调用数据库访问接口，得到处理结果后发给结果分发模块，然后由结果分发模块将数据分发给对应的业务。数据库中间件中各个模块间的通信采用线程间自定义消息方式。其模块消息流程关系如图 2 所示。

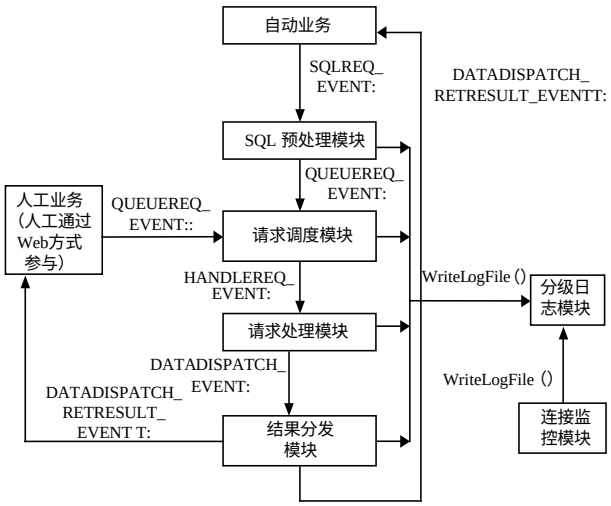

图 2　数据库中间消息流程

(1)SQL 预处理模块

负责接收并预处理自动语音业务的请求。数据库中间件主线程转发自动语音业务的数据请求，并根据配置文件中的配置将请求转换为具体的 SQL 语句，然后交给请求调度模块进行处理。

(2)请求调度模块

主要完成调度分发各种业务应用传来的数据请求，根据当前的数据处理状态对请求进行排队，然后分发给请求处理模块进行处理。

数据库中间件对调度排队的主要处理方式如下：

1)定时检查请求队列(每隔 2 s)，对超时的请求进行清除队列，并向客户端返回请求超时错误。

2)当新的请求加入到数据请求队列时，请求调度模块首先检查请求队列是否满，如未满，则插入队列尾部，进行排队，如队列满，则清除第一个请求，并向客户端返回请求重发的消息。

3)请求处理线程处理结束后，向请求调度模块取新的处理请求时，先检查该请求是否超时，如超时，则不再处理，并向客户端返回系统超时错误，然后取下一个新的请求进行处理。

4)当数据库连接失败时，超时时长的设置会自动减少，每次减 1 s，直到最小 2 s，当数据库正常连接后，超时时长再恢复默认值。

(3)请求处理模块

负责接收并处理请求调度模块转发的数据请求 SQL 语句，调用数据库的客户端接口对数据库进行操作，并将结果写具体格式化，最后将处理的结果交给结果分发模块。

(4)结果分发模块

主要完成接收请求处理模块发来的数据请求结果，并将请求结果分发给各种业务和应用。

(5)连接监控模块

主要完成监控各个数据库连接线程的状态，如果某个数据库连接因为数据库错误，或者异常或网络异常而断开，连接监控模块则一直重新连接该数据库，直至重连成功。

(6)分级日志模块

负责系统运行时日志的输出，可以直接写入日志文件，也可以动态通过窗口输出，可以根据需要进行日志的分级输出，共分为以下 3 级：

1)RUN_LEVE：表示运行时级别，输出信息很少。主要在中间件正式上线时使用。

2)TRACE_LEVE：表示跟踪业务处理过程，输出信息较多，主要在跟踪业务流程时使用。

3)DEBUG_LEVEL：调试级别，输出详细的处理过程，输出信息最多，主要在程序调试时使用。

2　对外业务接口规范

这里对数据库的业务接口规范进行统一的定义，以后只要各种业务和应用符合这个统一的规范，就可以利用数据库中间件和具体的数据源进行数据交换。这里业务接口主要是指各种业务和数据库中间件通信时，所必需遵循的对外业务接口规范，根据业务不同分为 2 类：自动语音业务的接口规范和人工业务接口规范。

(1)自动语音业务接口规范

自动语音业务接口规范主要定义了数据库中间件和自动语音业务之间通信的消息结构。分为 2 大类消息：1)数据请求消息=标准消息头+请求消息头(体)；2)数据应答消息=标准消息头+数据应答消息头+数据应答消息体。

具体结构如下：

1)结构 MessageHead，为自动语音业务和数据库中间件之间通信的标准消息头，具体结构如表 1 所示。

表 1　标准消息头结构

序号	名称	数据类型	说明
1	MessageFlg	WORD16	消息 ID
2	VersionNo	BYTE8	版本号
3	MessageNo	WORD16	消息序号
4	SensionNo	WORD16	会话 ID

2)结构 VoiceServiceSqlReq，为数据库中间件接收到的自动语音业务发来的请求消息头(体)。为方便起见，将请求消息头和请求消息体合二为一。具体结构定义如表 2 所示。

表 2　请求消息头(体)结构

序号	名称	数据类型	说明
1	ServiceID	WORD16	业务 ID
2	DbId	BYTE8	数据源(库)类型：数据库类型 0:MS SQL 1:ORACLE 2:SYBASE 3:INFORMIX 4:其他扩展
3	SqlType	WORD16	0--select，1—update，2 —删除
4	OperSerialID	WORD32	会话 ID，需要回传
5	Sequence	WORD32	操作流水号
6	SqlNum	BYTE8	sql 字符串个数，每串最大长度为 100 B
7	RetRecordNum	WORD16	返回数据包的最大记录数
…	…	…	…

3)结构 VoiceServiceSqlAck，为数据库中间件返回给自动语音业务的数据应答消息体。由于各个业务所需要数据的不一致，因此数据应答消息体需要设计为一变长的消息体，而且与具体业务紧密相关。

(2)人工业务接口规范

人工业务主要通过网页的方式同数据库交互，如 jsp, asp 等，主要有以下 4 种消息：数据查询请求消息，数据查询应答消息，数据更新请求消息，数据更新应答消息。其对应的消息结构如下：

1)结构 SelectStruct，为数据库中间件接收到的人工业务传来的数据查询请求消息体。

结构体定义如表 3 所示。

表 3　数据查询请求消息体结构

序号	名称	数据类型	说明
1	ClientAddress	CLIENTADDRESS	Client 地址
2	DbId	BYTE8	数据源(库)类型：数据库类型 0:MS SQL 1:ORACLE 2:SYBASE 3:INFORMIX 4:其他
3	RetPattern	BYTE8	返回方式
4	IsRetFieldName	BYTE8	是否需要返回字段名
5	OperSerialID	WORD32	本次操作序列号
6	SQLString [MAX_SQLSTRING_LEN]	CHAR	SQL 语句
…	…	…	…

2)结构 SelectAckStruct，为数据库中间件接收到的人工业务传来的数据查询应答消息体，定义如表 4 所示。数据更新请求消息体和数据更新应答消息体也都有标准的结构定义，具体结构限于篇幅这里不再一一说明。只要遵循数据库中间件以上外部业务接口规范，就可以根据需要来创建和扩充符合实际需要的业务层和接口层。

表 4　数据查询应答消息体结构

序号	名称	数据类型	说明
1	ClientID	WORD16	Client 标识
2	OperSerialID	WORD32	本次操作序列号
3	ResultID	BYTE8	返回操作结果 //0--成功，1--失败
4	RetRecordNumber	UINT16	返回实际记录数
5	PackFlag	UINT16	0 整包、1 最后一包、2 空包
6	FieldNumber	BYTE8	返回包的字段数
…	…	…	…

3　通信机制

通过图 1 可以看出，数据库中间件主要和以下 3 个实体进行通信：(1)自动业务：对自动业务的接口主要是通过 TCP/IP 来完成，直接建立 TCP/IP 连接，发送以上定义的标准接口消息来完成和数据库的交互。(2)人工业务：人工业务主要通过 JavaBean 控件和数据库中间件进行数据交互，JavaBean 控件提供 Java 同 C 的数据传输通道(即业务和数据库中间件之间的数据传输通道)。其原理是建立 TCP/IP 连接的机制和数据库中间件的通信采用 TCP/IP 协议。(3)数据库：对数据库的通信主要是通过如 MS SQL Server 的 DB Library、Sybase 的 DB Library 和 Oracle 的 OCI 等各个数据库服务器提供的客户端工具开发的 DLL 来实现，针对不同的数据库访问接口，开发不同的动态连接库，如 SYB.DLL, ORA.DLL 等来封装对数据库的直接访问，在数据库中间件程序启动时，通过读取配置文件来动态加载相应的数据库驱动。

4　结束语

此中间件支持大多数的主流商业数据库，对基于C/S架构(自动业务)和B/S(人工业务)架构的系统都有接口，而且还可以通过进一步的扩展，支持通过其他中间件如Tuxedo来间接访问数据库[3]。由于数据请求来源的透明性，数据请求响应的及时性，并且还有排队调度功能，且扩展灵活，容易维护。因此，目前该数据库中间件已经在许多NGN业务系统中得到了应用，并全天候运行，其系统的稳定性、安全性、灵活性得到了有力的验证。

参考文献

[1] 王卫民, 苏德富. Web 服务数据库访问中间件的分析和设计[J]. 计算机工程, 2005, 31(16): 102-103.

[2] 郑阿奇. Oracle 实用教程[M]. 北京: 电子工业出版社, 2004.

[3] 王秀敏, 袁晓红, 赵丽娜. 中间件及其在电信计费信息系统中应用[J]. 辽宁工程技术大学学报, 2005, 24(2): 220-223.

（上接第 58 页）

规则仍然由开发企业应用程序的 IT 人员与业务人员通过沟通完成，而不是直接由熟悉规则的业务人员直接编写与管理，这是本设计尚待解决的问题。

参考文献

[1] Johnson R. J2EE Development Without EJB[M]. 北京: 电子工业出版社, 2005.

[2] 孙勇强 基于EJB的业务规则引擎的设计和实现[J]. 计算机工程, 2005, 31(20): 220-222.

[3] Kamoski M. Business Rules Engine Overview[EB/OL]. (2006-08-20). ftp:///Files/Carlwave/BusinessRulesEngineOverview.rar.

[4] Ross R G. 业务规则方法原理[M]. 韩　柯, 译. 北京: 机械工业出版社, 2004.