

BEA TUXEDO 在移动计费系统中的应用

郭 磊 吴春旭 (中国科学技术大学信息管理与决策科学系 合肥 230026)

摘要:当前大型数据库应用系统大多采用三层体系结构,实现三层结构系统的关键技术就是中间件技术。本文先对 BEA Tuxedo 中间件做了较为详细的介绍,然后详细描述了 Tuxedo 在安徽移动计费系统中的应用。

关键词:BEA TUXEDO 中间件 三层体系结构 移动计费系统

1 前言

当前大型数据库应用系统大多采用三层体系结构,实现三层结构系统的关键技术就是中间件技术^[1]。Tuxedo 是美国 BEA 公司的一种事务处理(交易)中间件产品^[2],在金融、电信、邮政、证券、政府、航空等部门的大规模事务处理领域中有广泛的应用。它提供了一个开放的环境,支持各种各样的客户、数据库、网络、遗留系统和通讯方式,能够在 C/S 环境下整合各种异构平台,通过联机事务处理,保证交易的完整性和数据的一致性。

目前电信行业中的应用系统大多基于 Tuxedo 中间件开发,例如计费系统,帐务系统,信用管理系统,客户服务系统等。其中,计费系统是电信运营商进行市场竞争的核心支撑系统,地位非常重要。它即是分布式应用系统,也是联机处理系统,它需要大量并发进程,处理并涉及到操作系统、编程语言、数据通讯、数据库系统、系统管理及应用软件,相当复杂,所以需要使用 BEA Tuxedo 构建三层体系结构^[3]来简化它。

本文接下来先对 BEA Tuxedo 做较为详细的阐述,然后着重介绍 Tuxedo 在安徽移动计费系统中的应用。

2 BEA TUXEDO

2.1 Tuxedo 概述

Tuxedo 是 1984 年在当时属于 AT&T 的贝尔实验室开发完成的,但由于分布式处理当时并没有在商业应用上获得像今天一样的成功, Tuxedo 在很长一段时期里只是实验室产品,后来被 Novell 收购。在经过 Novell 并不成功的商业推广之后,1995 年被现在的 BEA 公司收购。尽管中间件的概念很早就已经产生,但中间件技术的广泛运用却是在最近 10 年之中。

BEA Tuxedo 中间件是分布式事物处理中间件,也称交易中间件,是用于分布计算的中间件基础结构,它使开放式应用系统具有高可缩放性、高灵活性和高可维护性。它不仅具有分布式交易处理和应用间报文通信的功能,而且具有一系列极其完善的服务,可帮助企业建立和运行应用系统,使开发人员能够建立跨越多个平台、数据库和操作系统的应用程序。这样,可以灵活选配操作平台以充分适应应用环境。

2.2 Tuxedo 特点

BEA Tuxedo 具有以下特点^[2]:

(1) 支持全局事务管理,支持 X/Open 规范,支持 TCP / IP 协议,支持全局的两阶段提交。

(2) 支持多种软硬件平台,支持包括 Unix、Windows NT、AS/400 和大型机专用系统在内的 70 多个硬件平台和操作系统。

(3) 分布式应用管理,支持异构环境下的分布应用,如同一应用中有不同的数据库,多个数据源。

(4) 进程管理。系统根据需要(客户端请求量)自动增加或减少每个 Service 的进程数(1 个 Service 可启动 n 个相同的进程)。某个 Service 忙,则相应地增加(启动)其进程数;某个 Service 闲,则相应地减少(关闭)其进程数,从而减少该 Service 占用的系统资源(如:内存、数据库连接数等),使整个系统的 Service 最优化运行。

(5) 优先级管理。可将 Service 根据优先级的不同赋权值,系统根据优先级权值将客户请求(Service)排队管理。

(6) 丰富的通信方式。有同步调用、异步调用、管道通信、会话、广播、通知、队列、发布订阅等通信方式,能很好地满足应用开发的要求。

BEA Tuxedo 采用三层结构的 C/S 模式组件软件模型。该结构利用交易中间件将应用的业务逻辑、表示逻辑和数据分为三个不同的处理层：客户为第一表示逻辑层，实现用户交互和数据表示；第二层的逻辑业务层为应用的主体，负责业务的逻辑处理；第三层数据层负责对数据库的检索、读取和更新等操作。交易中间件作为构造三层结构应用系统的基础平台，提供了两个主要功能：负责客户机和服务器间的联接和通讯；提供一个三层结构应用开发和运行的平台^[4]。由此可见，BEA Tuxedo 交易中间件提供了一个基础的框架去帮助你建立、运行和管理一个三层 C/S 模式的应用，大大缩短了应用开发的时间，提高了应用开发的成功率。

3 BEA TUXEDO 在移动计费系统中的实现

3.1 移动计费系统的三层体系结构

安徽移动计费系统采用的就是三层体系结构，三层体系结构中主系统与外围系统均采用中间件 Tuxedo 通讯，后台服务系统为 Tuxedo 的 Server 端，查询系统、报表系统等子系统为 Tuxedo 的 Client 端，客户端程序主要使用三种方式：Delphi 客户端、Web、营业终端，图 1 为移动计费系统的三层体系结构图。

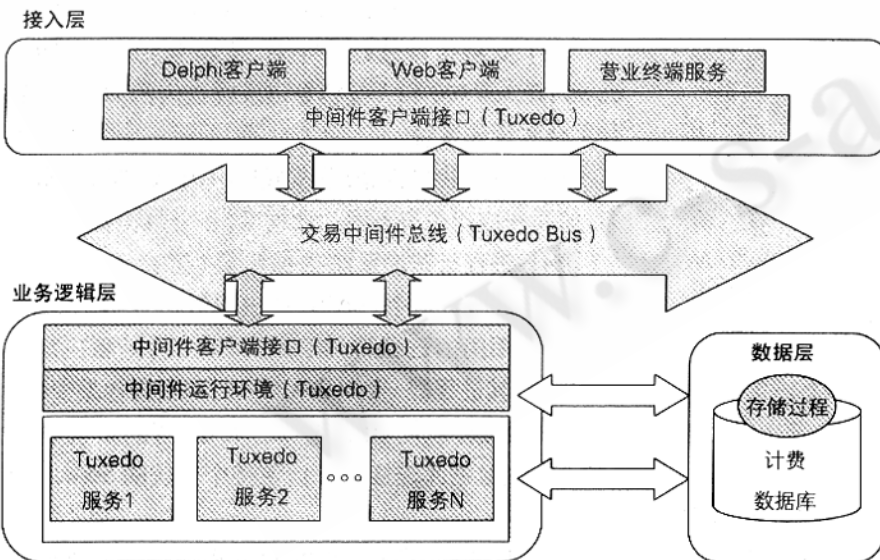


图 1 移动计费系统的三层体系图

三层体系中采用 Tuxedo 提供的基于服务的编程方式，Server 端编写服务，Client 端调用 Server 提供的

服务。在服务器端，服务器收到请求信息后，进行业务逻辑处理并访问数据库，然后将处理结果返回给客户。客户机收到 Tuxedo 服务返回的相应结果，从结果缓冲区中将数据提取出来，通过客户端程序进行处理，数据接口格式都采用规定格式的字符串。客户端与服务端的通信是由 Tuxedo 的 API 函数实现的，客户端由函数 Tpinitt() 与服务器建立连接，由函数 Tpcall() 申请 Service 服务，再由相关的 Tuxedo 函数对数据解包。三层体系中客户端与数据库服务端的通信是由 Tuxedo 来完成的。客户端所有的请求都发送给后台的 Tuxedo 服务，请求处理及返回由 Tuxedo 负责，而数据库服务器只接收 Tuxedo 服务发来的请求。由此可见，Tuxedo 服务是处于客户端与应用服务端的中间。一切的负载均衡、控制都由 Tuxedo 来完成。

3.2 具体实现

本文以缴费模块中的押金查询服务为例，说明系统服务的具体实现。押金查询是通过输入：ACCTID 账户标识，来输出结果集：col1: NAME 账本科目，col2: AMOUNT 押金金额。服务器端使用 UNIX 操作系统，Oracle8i 数据库，代码使用 pro * c 编写，代码片断如下：

.....

```
#include "atmi.h" /*
Tuxedo ATMI 函数库的头文件
*/
.....
void Deposit ( TPSVCINFO *
rqst)
/* TPSVCINFO 是 Tuxedo 定义
的 tpsvcinfo 结构，里面包含有
连接描述符、账户标识 ID、指向
接收缓冲区的指针等信息 */
{ .....
/* 从缓冲区中取出账户标
识 */
GetValue ( " ACCTID", sAcctID,
sizeof ( sAcctID)); /* 从缓
冲区中取出账户标识 */
if ( GetAreaDBLink ( sArea,
sDbName, v_ErrorInfo) != SUCCESS) /* 查询账户
所在数据库 */
```

```

{
    userlog (" RetCode = % s" , " F00001" );
    userlog (" ErrorInfo = % s" , v_ErrorInfo);
    return;
}
sArea[0] = sAcctID[0]; /* 由账户标识截取营
业区 */
sprintf ( sSql," SELECT a. BALNAME, b. BALANCE
FROM STATIC. T_SP_BALITEM a, info. T_CI_USERBOOK
_% s b WHERE a. BALITEMTYPE = '0' AND a. BALITEM =
b. BALITEM AND b. USERID = '% s'" , sArea, sAcctID);
/* 动态 SQL 语句 */
EXEC SQL AT :sDbName DECLARE S STATEMENT;
EXEC SQL PREPARE S FROM :sSql;
EXEC SQL DECLARE cur CURSOR FOR S;
EXEC SQL OPEN cur;
.....
/* 生成两列( 账本科目, 余额)的结果集 */
RsCreate(2);
RsSetColNameList(" 账本科目, 余额");
do
{
    EXEC SQL FETCH cur INTO :sName, :iAmount;
    RsSetCol(1, sName);
    sprintf( tmp, "% d" , iAmount);
    RsSetCol(2, tmp);
}
RsSaveRow();
EXEC SQL CLOSE cur;
.....
Tpreturn( TPSUCCESS,0, rqt - > data,0,0); /*
将 rqt - > data 作为响应缓冲区返回 */
}

```

在配置时,针对服务的不同特点,需要不同的配置要求(包括优先级、Service 进程数量等)。例如:公用服务由于被其它各具体业务调用,所以使用机会多,因此要求有较高优先级,Service 进程数量要求较多。对于公共查询服务函数可设置成会话方式的服务,该方式用于客户端和服务端需要进行大量、多次数据传输并保持连接的情况。

3.3 系统优点

采用 BEA Tuxedo 设计三层结构实现移动计费系统,具有以下优点:

(1) 减少工作量。开发工作量大量减少,缩短投放市场的时间,并且维护工作量大量减少;

(2) 提高了系统效率。中间件维护与数据库常连接,减少与数据库的连接/断开次数,分担数据库的业务处理工作,提高数据库服务性能;

(3) 跨平台支持,对程序移植非常方便;

(4) 使用方便。客户端无须安装数据库 Client 端,只需简单的设置服务器的地址 Tuxedo 安装目录即可,开发时也只需按 Tuxedo 提供的几个函数,即可取得 Server 端的响应内容;

(5) 提高了网络的安全性和数据的安全性,客户端不能直接访问服务器的数据库。

4 总结

安徽移动计费系统是一个结构复杂、业务繁多、数据量庞大的大系统,它包括多个子系统,其各个子系统之间是相互制约而紧密联系的整体。因此,在实现过程中,必须充分注意各个子系统之间的协调,系统运行的可靠性、安全性、快速性是需要首要考虑的问题。采用三层体系结构模式,利用交易中间件 BEA Tuxedo 实现联机交易处理系统,使得程序开发相对简便,大大缩短了开发周期,提高了系统效率,很好地实现了系统的功能,保证了系统的可靠性、实时性和可维护性,同时,具有良好的可扩展性和可移植性,这样移动计费业务的拓展也可以方便地在系统中实现。

参考文献

- 1 Wolfgang Emmerich. Software engineering and middleware: a roadmap. Proceedings of the Conference on The Future of Software Engineering. May 2000.
- 2 BEA Systems, Inc. The BEA Tuxedo Datasheet. www.bea.com/products/tuxedo/index.shtml.
- 3 马雪英,基于 BEA 中间件设计三层结构 MIS 系统,计算机应用与软件,2004.6。
- 4 徐春金, TUXEDO 中间件开发与配置,北京 中国电力出版社,2003(7)。