

Tuxedo

中间件开发与配置

徐春金 编著

- 国内资深专家精心编写
- 对 Tuxedo 的各个主题进行全面深入的介绍
- 填补国内计算机图书的一项空白
- 本书既适合初学者学习，也可作为有一定使用经验的开发管理人员的参考书



中国电力出版社
www.infopower.com.cn

Tuxedo

中间件开发与配置

徐春金 编著

中国电力出版社

内 容 提 要

在电信、金融等行业的大型计算机应用系统中，中间件的使用日益普及，中间件已与操作系统、数据库并列为三大基础软件。BEA Tuxedo 作为最优秀的交易中间件，在很多行业中的使用非常广泛，本书结合作者多年使用 Tuxedo 的经验，通过大量例子详细介绍 Tuxedo 的设计、开发、配置、管理等方面的内容。使读者可以快速、全面的掌握 Tuxedo。本书的主要内容如下：Tuxedo 的安装和配置、Tuxedo 应用系统的管理维护、Tuxedo 的缓冲区介绍、Tuxedo 的服务端编程、Tuxedo 的客户端编程、Tuxedo 的通讯方式、Tuxedo 的数据库编程、Tuxedo 的可靠消息队列/Q、Tuxedo 与其他系统的互连和 Tuxedo 的性能调整。

本书既适合初学者学习，也可作为有一定使用经验的开发管理人员的参考书。

图书在版编目（CIP）数据

Tuxedo 中间件开发与配置 / 徐春金编著. —北京：中国电力出版社，2003

ISBN 7-5083-1315-1

I.T... II.徐... III.事务计算程序, Tuxedo IV.TP319

中国版本图书馆 CIP 数据核字（2002）第 088903 号

中国电力出版社出版、发行

（北京三里河路 6 号 100044 <http://www.infopower.com.cn>）

汇鑫印务有限公司印刷

各地新华书店经售

*

2003 年 7 月第一版 2003 年 7 月北京第一次印刷

787 毫米×1092 毫米 16 开本 14.5 印张 326 千字

定价 25.00 元

版 权 所 有 翻 印 必 究

（本书如有印装质量问题，我社发行部负责退换）

前 言

计算机技术日新月异，从大型机、小型机、个人计算机、客户/服务器到互联网。软件的开发技术也在不断推陈出新，从 C/S 结构到 B/S 结构；从两层到三层；从 CORBA、COM、到 J2EE、.NET，现在基于中间件的开发应用系统已成为主流。中间件的使用日益普及，中间件已与操作系统、数据库并列为三大基础软件。作者结合多年使用 Tuxedo 的经验，通过大量例子详细介绍了 Tuxedo 的设计、开发、配置和管理等方面的内容。使读者可以快速、全面的掌握 Tuxedo 并更好的使用它。本书既适合初学者学习，也可作为有一定使用经验的开发管理人员的参考书。

Tuxedo 的安装介质可以从 <http://commerce.bea.com/index.jsp> 下载，联机文档可以从 <http://e-docs.bea.com> 下载。

本书中的源代码如果读者需要的话，可以发 E-mail 到 xcjing@yeah.net。关于本书，如果你有任何问题，也可以通过 xcjing@yeah.net 与作者联系。

最后，衷心感谢中国电力出版社，特别要感谢责任编辑温振宁，是他们专业、负责、辛勤的工作，使本书得以顺利出版。最后一并感谢为本书的出版提供过帮助的人。

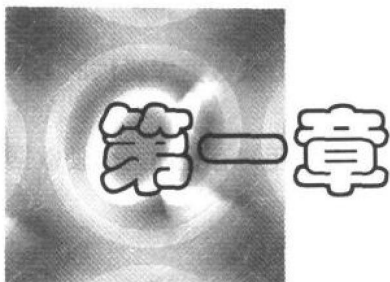
作 者

目 录

前 言

第一章 中间件技术介绍	1
1.1 两层结构与三层结构	1
1.2 中间件技术简介	3
第二章 Tuxedo 简介	8
2.1 Tuxedo 的核心系统组成部分	8
2.2 最新版本 Tuxedo 8.0 的新功能	14
2.3 Tuxedo 的安装	15
2.4 一个简单的例子	17
第三章 Tuxedo 应用系统的配置	22
3.1 Tuxedo 应用系统的配置	22
3.2 多机 (MP) 模式的配置	29
第四章 Tuxedo 应用系统的管理	34
4.1 Tuxedo 提供的管理手段	34
4.2 Tuxedo 应用系统的启动和关闭	35
4.3 tadmin 的使用	38
4.4 图形化管理工具的使用	42
4.5 动态改变 Tuxedo 应用系统的配置	44
4.6 Tuxedo 的文件系统	47
第五章 Tuxedo 的缓冲区	49
5.1 Tuxedo 的缓冲区介绍	49
5.2 VIEW (VIEW32) 缓冲区	50
5.3 FML (FML32) 缓冲区	53
5.4 XML 缓冲区	56
第六章 Tuxedo 的服务端编程	61

第七章 Tuxedo 的客户端编程	67
7.1 Tuxedo 的客户端简介	67
7.2 客户端编程常用的 ATMI.....	70
7.3 采用 C 语言编写 Tuxedo 的客户端程序.....	73
7.4 采用 VC 编写 Tuxedo 客户端程序.....	74
7.5 采用 PB 编写客户端程序	75
第八章 Tuxedo 的通信方式	98
8.1 同步请求/回答方式	98
8.2 异步调用及例子	104
8.3 会话通信方式及例子	107
8.4 消息通信方式及例子	114
8.5 发布/订阅通信方式	120
第九章 Tuxedo 数据库编程	129
9.1 Tuxedo 如何处理分布式事务	129
9.2 ORACLE 数据库 XA 的配置.....	135
9.3 INFORMIX 数据库 XA 的配置.....	141
9.3 数据库服务端的编程技巧	147
9.4 事务编程应注意的一些问题	156
第十章 Tuxedo 的可靠消息队列 / Q	163
10.1 /Q 介绍	163
10.2 /Q 的管理	166
10.3 /Q 的编程	171
10.4 /Q 一个简单的例子	174
10.5 /Q 与事务	178
第十一章 Tuxedo 与其他系统的互连	192
11.1 Tuxedo 的/DOMAIN 技术介绍.....	192
11.2 Tuxedo 通过 Elink 产品来与大机实现互连.....	203
11.3 Java 客户端通过 Jolt 调用 Tuxedo 服务.....	205
11.4 Tuxedo 与 Weblogic 互连	207
第十二章 Tuxedo 的性能调整	215
12.1 Tuxedo 应用系统对 IPC 资源的要求	215
12.2 Tuxedo 应用系统的性能优化方法	219



中间件技术介绍

1.1 两层结构与三层结构

长期以来，我们一直使用着“客户端/服务器”的两层结构，这种两层结构曾让无数人 为之兴奋和惊叹，即客户端提供用户界面、处理业务逻辑，数据库服务器接受客户端 SQL 语句并对数据库进行查询和更新等操作，然后将操作结果返回给客户端，如图 1-1 所示。

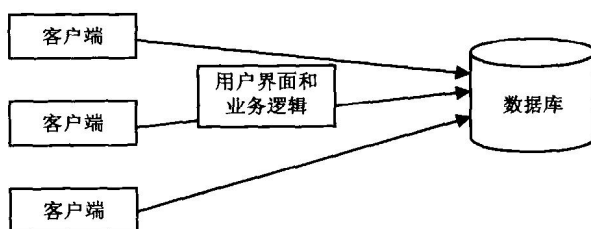


图 1-1

在一个比较简单的计算机应用系统中，采用两层体系结构的确给人们带来了相当的灵活性。但随着计算机应用水平的飞速发展、企业信息化水平的不断深入、企业客户的不断增加，以及新业务的不断出现，越来越多的用户对计算机应用系统提出了更高的要求：

- (1) 要能够同时支持成千上万乃至更多用户的并发服务请求。
- (2) 由单一的局域网向跨多个网络协议的广域网扩展。
- (3) 不仅要支持一般的信息管理，而且还要支持关键业务的联机交易处理。
- (4) 从支持单一的系统平台和数据源转向支持异构的多系统平台和多数据源。

面对用户的新需求，两层结构的应用模式由于采用客户机与服务器直接连接的方式形成了其固有的一些缺陷：

(1) 难以维护。客户端/服务器 (Client/Server) 结构用户界面、业务逻辑和数据逻辑相互交错，通常在第一次部署的时候比较容易，但难于升级或改进，而且经常基于某种专有的协议（通常是某种数据库协议）。它使得重用业务逻辑和界面逻辑变得非常困难。

(2) 难以扩展。随着系统的升级，系统复杂程度大大增加，难以扩展，另外，它是一个

封闭的系统，很难与其他的应用系统实现互操作。

(3) 安全性差。客户端程序可以直接访问数据库，可通过编程语言或数据库提供的工具直接对数据库进行操作，不安全。

(4) 性能不好。客户端直接与数据库建立连接，当有大量的并发用户存在时，会使数据库不堪重负，性能迅速下降，甚至死机。

三层结构

为解决传统两层模式与应用需求日益突出的矛盾，以交易中间件为基础框架的三层应用模式应运而生，三层结构以中间层管理大量的客户端并为其连接、集成多种异构的服务器平台，通过有效的组织和管理，在极为广泛的范围内将客户机与服务器进行高效组合。同时，中间件开创的以负载平衡、动态伸缩等功能为代表的管理模式，已被广泛证实为建立关键业务应用系统的最佳环境，使在两层模式下不可能实现的应用成为可能，并为应用提供了充分的扩展余地。这种模式的成功应用已为许多国际大型企业在应用开发和部署方面节省了大量的时间和金钱，由此促使越来越多的系统开发商和用户采用三层结构模式来开发和实施其应用。

三层客户机/服务器模式的核心概念是利用中间件将应用的用户界面、业务逻辑和数据逻辑分为三个不同的处理层，如图 1-2 所示。

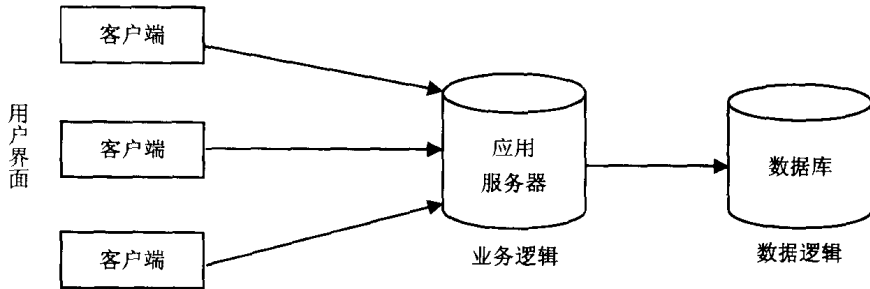


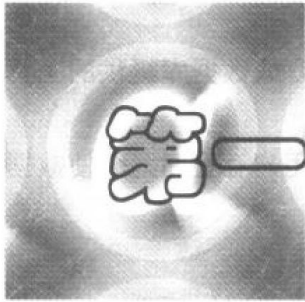
图 1-2

(1) 表示层（用户界面）：它的主要功能是实现用户交互和数据表示，为以后的处理收集数据，向第二层的业务逻辑请求调用核心服务处理，并显示处理结果。这一层通常采用 VB, PB, DELPHI 等语言编写，或采用浏览器实现。

(2) 中间层（业务逻辑）：实现整个应用系统核心业务逻辑，通常把业务逻辑划分成一个个独立的模块，用中间件提供的 API 结合数据库提供的编程接口实现。客户端通过调用这些模块实现相应的业务操作。

(3) 数据层（数据逻辑）：数据库负责管理整个应用系统的数据资源，完成数据操作。中间层上应用程序在处理客户端的请求时，通常要存取数据库。

随着市场竞争的日益加剧和企业电子信息化建设的不断深入，高度灵活、能快速部署新服务和新应用的三层结构应用系统将成为企业信息化的必由之路。采用以中间件为基础的三



第一章

中间件技术介绍

1.1 两层结构与三层结构

长期以来，我们一直使用着“客户端/服务器”的两层结构，这种两层结构曾让无数人为之兴奋和惊叹，即客户端提供用户界面、处理业务逻辑，数据库服务器接受客户端 SQL 语句并对数据库进行查询和更新等操作，然后将操作结果返回给客户端，如图 1-1 所示。

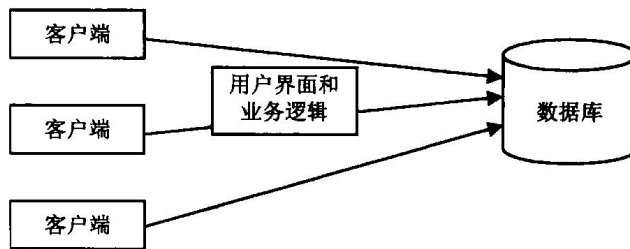


图 1-1

在一个比较简单的计算机应用系统中，采用两层体系结构的确给人们带来了相当的灵活性。但随着计算机应用水平的飞速发展、企业信息化水平的不断深入、企业客户的不断增加，以及新业务的不断出现，越来越多的用户对计算机应用系统提出了更高的要求：

- (1) 要能够同时支持成千上万乃至更多用户的并发服务请求。
- (2) 由单一的局域网向跨多个网络协议的广域网扩展。
- (3) 不仅要支持一般的信息管理，而且还要支持关键业务的联机交易处理。
- (4) 从支持单一的系统平台和数据源转向支持异构的多系统平台和多数据源。

面对用户的新需求，两层结构的应用模式由于采用客户机与服务器直接连接的方式形成了其固有的一些缺陷：

(1) 难以维护。客户端/服务器 (Client/Server) 结构用户界面、业务逻辑和数据逻辑相互交错，通常在第一次部署的时候比较容易，但难于升级或改进，而且经常基于某种专有的协议（通常是某种数据库协议）。它使得重用业务逻辑和界面逻辑变得非常困难。

(2) 难以扩展。随着系统的升级，系统复杂程度大大增加，难以扩展，另外，它是一个

客户机服务器的操作系统之上，管理计算资源和网络通信。

IDC 对中间件的定义表明中间件是一类软件，而非一种软件；中间件不仅仅实现互连，还要实现应用之间的互操作；中间件是基于分布式处理的软件，最突出的特点是其网络通信功能。

最早具有中间件技术思想及功能的软件是 IBM 的 CICS，但由于 CICS 不是分布式环境的产物，因此，人们一般把 Tuxedo 作为第一个严格意义上的中间件产品。Tuxedo 是 1984 年在当时属于 AT&T 公司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 公司 1995 年成立后收购 Tuxedo 才成为一个真正的中间件厂商，IBM 的中间件 MQSeries 也是 20 世纪 90 年代的产品，其他许多中间件产品也都是最近几年才成熟起来。国内在中间件领域的起步阶段正是整个世界范围内中间件的初创时期。东方通科技早在 1992 年就开始中间件的研究与开发，1993 年推出第一个产品 TongLink/Q。可以说，在中间件领域国内的起步时间并不比国外晚多少。

中间件的作用

具体地说，中间件屏蔽了底层操作系统的复杂性，使程序开发人员面对一个简单而统一的开发环境，减少程序设计的复杂性，将注意力集中在自己的业务上，不必再为程序在不同系统软件上的移植而重复工作，从而大大减少了技术上的负担。

中间件带给应用系统的，不只是开发的简便、开发周期的缩短，同时也减少了系统的维护、运行和管理的工作量，还减少了计算机总体费用的投入。Standish 的调查报告显示，由于采用了中间件技术，应用系统的总建设费用可以减少 50% 左右。在网络经济大发展、电子商务大发展的今天，从中间件获得利益的不只是 IT 厂商，IT 用户同样是赢家，并且是更有把握的赢家。

其次，中间件作为新层次的基础软件，其重要作用是将不同时期、在不同操作系统上开发的应用软件集成起来，彼此像一个天衣无缝的整体协调工作，这是操作系统、数据库管理系统本身做不了的。中间件的这一作用，使得在技术不断发展之后，我们以往在应用软件上的劳动成果仍然物有所用，节约了大量的人力、财力投入。

中间件的分类

中间件技术已经日渐成熟，并且出现了不同层次、不同类型的中间件产品。大致可分为以下几类：

1. 消息中间件 (Message Orient middleware, MOM)

将数据从一个应用程序发送到另一个应用程序，这就是消息中间件的主要功能。它要负

责建立网络通信的通道，进行数据的可靠传送，保证数据不重发，不丢失。消息中间件的一个重要作用是可以实现跨平台操作，为不同操作系统上的应用软件集成提供数据传送服务。它适用于进行非实时的数据交换，如银行间结算数据的传送。主要的产品有：IBM MQSeries、BEA MessageQ、BEA Tuxedo /Q、MicroSoft MSMQ、东方通科技 TongLink/Q。

2. 交易中间件 (Transaction Processing, TP)

交易中间件也和消息中间件一样具有跨平台，跨网络的能力，但它的主要功能是管理分布于不同计算机上的数据的一致性，协调数据库处理分布式事务，保障整个系统的性能和可靠性。交易中间件所遵循的主要标准是 X/Open DTP 模型。它适用于联机交易处理系统，如银行的 ATM 系统，电信的计费营收系统。主要产品有：BEA Tuxedo、IBM CICS、东方通科技 TongEasy。

3. 对象中间件 (Object Momitor)

对象中间件也叫 Object TP Momitor，它一般也具有交易中间件的功能，但它是按面向对象的模式来组织体系结构的，在线的电子交易很适合采用这种中间件类型。因为这种类型的应用会被频繁地修改，面向对象的体系结构可以保持足够的弹性来应付这种改动。提到面向对象的中间件，就不得不提到对象请求代理 (Object Request Brokers, ORBs)，现在有 3 种对象请求代理体系结构，CORBA、EJB 和 COM+，ORBs 是一组协议或者标准，现在的对象中间件都是按照上面三种体系结构的某一种来构造的，如：Borland VisiBroker、MicroSoft Transaction Server、IONA orbix、IBM componentbroker、东方通科技 TongBroker。

4. 应用服务器 (Application Server)

应用服务器主要用来构造基于 WEB 的应用，是企业实施电子商务的基础平台。它一般是基于 J2EE 体系结构。让网络应用的开发、部署、管理变得更加容易，使开发人员专注于业务逻辑。主要的产品有 BEA weblogic、IBM webspere、Borland AppServer 等，还有一些开放源代码的 J2EE 应用服务器，如 JBOSS 等。

5. 企业级应用集成 (Enterprise Application Intergration, EAI)

一个大型企业内部往往有很多的计算机应用系统，EAI 可用于对这些系统进行有效的整合，使它们之间能够互相访问，实现互操作。EAI 所提供的上层开发工具或许是 EAI 和其他中间件最大的区别，它允许用户自定义商业逻辑和自动使数据对象符合这些规则。EAI 的典型用户是那些巨型企业的大量应用系统的整合，主要的产品有 BEA Elink、BEA WLI 等。

6. 安全中间件

近几年，随着互联网的发展，信息安全越来越受到普遍关注，安全中间件也应运而生。安全中间件是以公钥基础设施 (PKI) 为核心的、建立在一系列相关国际安全标准之上的一个开放式应用开发平台，向上为应用系统提供开发接口，向下提供统一的密码算法接口及各种 IC 卡、安全芯片等设备的驱动接口。主要产品有：ENTRUST entrus、东方通科技 TongSec 等。

上面对中间件的分类描述并不是一个很严格的定义，只是一个大致的划分。中间件是一个快速发展的技术，市场越来越大。除了上面提到的大型厂商提供的通用产品，还有很多小公司专注于专有的领域，为客户提供面向某一领域，更加符合客户需求的中间件产品。如：

清华北美的 THMT、JAVATS、青牛公司的 CT-USE 等。

使用中间件带来的好处

世界著名的咨询机构 Standish Group 在一份研究报告中归纳了中间件的十大优越性：

(1) 缩短投放市场所需时间：时间因素绝对是所有项目的首要问题。自行建立软件基础结构耗时长，使用现成的基础结构软件则可以将软件开发时间缩短 25%~50%。如果应用系统每月可带来 100 万美元的利润或节省 100 万美元的开销，那么软件开发时间缩短的每一个月就相当于在银行存入 100 万美元。

(2) 节省应用开发费用：Standish 组织调查 100 个关键任务应用系统，对其编码中的业务/应用部分的代码量和基础结构部分的代码量进行评估，结果表明：只有少于 30%的代码与应用/业务有关，而其余部分均归属于基础结构，如果使用现成的基础结构，费用可节省 25%~60%。对于一个 200 万美元的项目而言，这意味着将节省 50 万~120 万美元。

(3) 减少系统运行开销：一个不采用商用中间件产品部署的系统，其初期购买及运行费用将加倍。许多大企业由于采用中间件产品而在硬件及软件方面节省了大量的投资。一个 200 万美元的项目因此将只需花费 100 万，而其中还包括了中间件的投资。

(4) 降低失败率：Standish 市场组织对项目失败的定义为项目被取消或没有完成预期的预算、交付使用时间以及业务要求等目标。调查表明自行开发中间件的项目失败率高达 90%以上，可见这种做法是十分危险的。其结果可能是 100%推翻重来，以至于 1000%超出预算。

(5) 提高投资效率：采用中间件产品既能保护现有投资，又能提高投资效率。通过使用中间件产品，用户可以建立专有系统以外的应用程序，不但扩展了主机应用，而且还能将主机应用与整体系统实现无缝连接。许多企业发现其在两层客户机/服务器结构下建立的新的应用系统并不能在 Internet 上运行，而已被淘汰的应用程序则更适合 Internet。采用中间件技术可以恢复被 Internet 淘汰的应用程序的生命，该费用将大大低于应用程序重新开发的费用。这笔费用通常会在数十万美元到数亿美元之间。

(6) 简化应用集成：使用中间件产品，现有应用程序、新开发应用程序以及所有其他购买软件均能实现无缝集成。从而能够从开发、投放市场时间两方面节约数百万美元的开支。

(7) 降低软件维护费用：自行开发基础结构成本很高，维护时则更会变本加厉。对于自行开发的基础结构，其年维护费可达开发费用的 15%~25%，而应用程序的维护费则达到开发费用的 10%~20%。以一个 200 万美元的项目为例，其中 120 万用于基础结构建立，其年维护费为 18 万~28 万美元，而购买现成的中间件仅需项目总成本的 15%~20%，依购买规模和供应商的不同还有可能大大低于该价格。

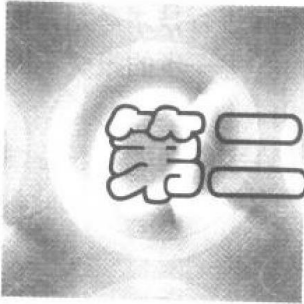
(8) 高质量：在自行建立中间件的应用系统中，每次将新的应用组件加入系统时，相应的新的中间件模块被加入到当前的中间件之上。在一个实际的应用系统中，Standish 集团发现它使用了 17000 个应用接口。而商用中间件产品则具有清晰的接口层次，从而大大降低新系统及原有系统的维护成本。此外，由于商用中间件支持数百万的交易吞吐量，其质量远远高于用户自行开发的中间件产品。

(9) 保证技术革新：除了需对自行建立的中间件进行维护，还需对其进行技术革新，而这似乎不太现实。而从第三方购买的中间件产品则会随着其所属公司对其进一步的投资不断得到增强。采用具有层次接口设计的中间件产品，将能节省时间和费用。

(10) 增强应用程序吸引力：由于中间件提供了一个灵活的平台，许多新功能、新特性均可以在应用系统中得以建立。

中间件的市场前景

据 IDC 预测：到 2002 年全球中间件市场销售额将达到 80 亿美元，建立在中间件基础上的通用组件销售额也会增加至 70 亿美元。在中国，1998 年中间件市场总值达 12.34 亿美元，1999 年为 17 亿美元，到 2004 年将达到 90.3 亿美元，年增长率为 39.7%，远远高于软件业的平均增长水平。



第二章

Tuxedo 简介

Tuxedo 是 BEA 公司的交易中间件产品，1984 年由贝尔实验室开发成功，1992 年易主 Novell 公司，1996 年由 BEA 公司收购，经过十多年的不断更新和完善，Tuxedo 已经发展成为交易中间件领域事实上的标准。

2.1 Tuxedo 的核心系统组成部分

BEA Tuxedo 是由服务器端的事务管理器、可靠队列服务、应用域以及客户端的工作站等几个核心部分组成的，如图 2-1 所示。下面对这几部分进行介绍。

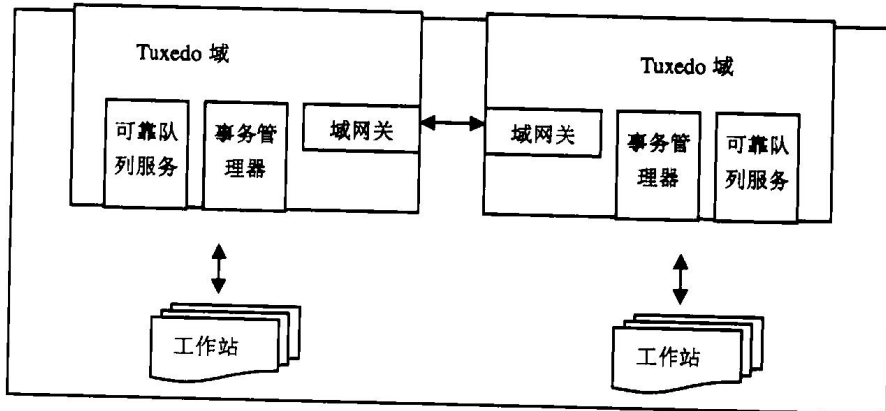


图 2-1

1. 事务管理器/T

运行于服务器端的事务管理器是 Tuxedo 体系结构的中心，它是每个 Tuxedo 服务器的核心，提供重要的分布式应用服务，包括：名字服务、数据路由、负载平衡、配置管理、分布式事务管理和安全性管理。它包含 Tuxedo 的核心数据结构公告板 BB (Bulletin Board)，BB 中包括服务名、路由信息、请求服务的队列和负载等基本信息，Tuxedo/T 负责访问和维护 BB 中的信息，并利用这些信息实现其各项功能。

2. 工作站/WS

工作站把 Tuxedo ATMI API (ATMI, 应用事务管理接口，是支持应用开发的一组函

数)扩展到客户端应用程序中,它适用于 DOS、Windows 3.1/95/98/2000/XP、Windows NT、IBM OS/2、Macintosh OS 和所有商用版的 UNIX。有了工作站,开发者用 ATMI 客户端的功能可透明访问在 Tuxedo 分布式环境中任何服务器上的服务。有了工作站,用户可在常见的编程环境中如: Visual Basic、PowerBuilder、DELPHI 和 VC 等编写 Tuxedo 的客户端程序。

3. 可靠队列服务/Q

Tuxedo 提供了一个简单的可靠的队列机制,保证应用系统提交的请求和数据可在网络故障或目的服务器瘫痪的情况下,也能递交到目的服务器。应用程序能将服务请求入队和出队,并可以设定系统,使队列中的请求自动地转发给 Tuxedo 的服务进程,并取回处理结果。这种可靠队列作为一种资源管理器,可以和其他资源管理器(如数据库)协作,完成全局事务处理。

/Q 的常见用法是用于实现数据的可靠传送,把数据从一台机器中可靠地传送到另一台机器中。如:在电信计费业务中,可以用/Q 把采集到的计费数据发送到计费中心进行处理。在银行中,不同的银行间可用/Q 传送结算数据。数据传送可以是在客户端与服务端之间,或服务端与服务端之间。/Q 还可以用于实现批处理或 workflow 等业务。

4. 域/Domain

扩大范围、由很多台(几百甚至上千)服务器共同组成的一个应用系统按功能或结构划分为不同的域,每个域独立地完成域内的操作,域间操作由域网关完成,从而提高每个域和整个系统的运行效率。

Tuxedo 的域特性把客户/服务器模型扩展到多个独立自治的应用系统。一个域既可以是一组 Tuxedo 的应用程序(若干相关的应用服务和配置环境的组合),也可以是一组运行在另一个非 Tuxedo 环境中的应用程序。Tuxedo 和其他中间件的互操作也是利用域网关的概念来实现的。

Tuxedo 对域的划分和管理类似于网络应用中划分子网的概念——将复杂的网络应用按功能或地域等因素划分为不同的子网,子网间以路由器连接,不同的网络协议通过网关透明地转换,从而提高网络效率并加强整个网络的可管理性。这一应用模式已被广泛证明是处理大范围、复杂应用的成功经验和事实上的标准,同时也是 Tuxedo 在多年大规模企业用户的实际应用中总结和开发的行之有效的中间件功能。而其他绝大多数中间件厂商尚无力涉足域的划分与管理,从而很难为超大规模的应用提供强有力的支持。

在英国教育与就业部的就业服务项目中,在英格兰、苏格兰和威尔士三岛广泛分布的 1100 多个办公机构要负责 2,450,000 名失业人员的福利管理和再就业,其应用网络中有 1,100 多台服务器和 21,000 多台 PC,同时为超过 30,000 名工作人员提供服务,系统每天要处理 7,000,000 笔交易,其中 95% 的交易必须在 3 秒钟内完成,该系统号称世界最大的客户机/服务器应用项目,同时也是整个欧洲最大的广域网应用项目。面对如此庞大的应用和极高的系统时间响应要求,其他中间件厂商束手无策,而正是由于 BEA 具备了丰富的超大规模应用领域的成功经验和领先技术, Tuxedo 轻松地击败竞争对手,被选为该系统的中间件运行平台,并取得了令用户极为满意的实际效果。

Tuxedo 的主要功能介绍

Tuxedo 具备了所有交易中间件产品所应具备的功能，作为构造三层结构应用系统的基础平台，Tuxedo 的主要功能可概括为五大平台：

- 应用开发平台：提供 API，供开发人员调用，用于开发整个应用系统。
- 应用部署平台：通过配置文件，对应用系统进行部署。
- 应用运行平台：提供应用系统运行的环境。
- 应用管理平台：对整个应用系统进行管理和监控。
- 应用互联平台：可以和其他的应用系统互联，实现互操作。

下面对这五大平台分别进行介绍。

应用开发平台

1. 支持的开发工具

(1) 在服务器端 Tuxedo 支持 C、COBOL 语言。

(2) 在客户端 Tuxedo 支持几乎所有的编程语言和开发工具，只要这种语言或工具支持动态链接库 DLL 或支持 C、COBOL 的语言调用。其客户端通过 DLL 可以和 Visual C++、Visual Basic、Power Builder、SQL Windows、Delphi、Develop/2000 以及其他 4GL 和 CASE 工具互连。另外，通过 BEA Jolt，用户还可用 JAVA 语言编写客户端程序。

(3) BEA 公司提供 BEA Builder 工具，帮助开发人员在图形化的开发工具上完成编程。提供 DES 工具，帮助开发人员在字符界面上完成编程，使系统的开发效率极大提高。

(4) Tuxedo 还得到其他第三方开发管理工具厂商的支持。

Tuxedo 提供的编程 API 总共只有几十个函数，常用的只有十几个，开发人员可以很快掌握这些 API，进行 Tuxedo 服务器端和客户端的开发工作。

2. Tuxedo 支持的平台

Tuxedo 是一个非常开放的平台，支持包括 UNIX、Windows NT、Linux、AS400、大机操作系统等 70 多种平台，并且在所有平台上的 API 都是一致的，平台间的数据表示的差异由 Tuxedo 自动屏蔽，这也是 Tuxedo 的独到之处，极大地拓宽了用户对平台的选择范围。

构筑在 Tuxedo 系统之上的应用独立于计算机硬件、操作系统和数据库，将应用从一种开放平台移植到另一种开放平台，只需作应用程序的重新编译和极少的 SQL 语句的调整（不同数据库产品其 SQL 语法可能稍有不同），应用系统就能顺畅完成平台转移。

为支持异构系统互联，Tuxedo 允许用户在配置文件中设置机器类型，Tuxedo 支持自动码集、位数及字节顺序的转换，Tuxedo 屏蔽不同平台间的数据表示，不需要编程人员精通各种平台的数据表示差异，从而“自动”完成异构系统的互联。

Tuxedo 支持几乎所有的数据库，包括 ORACLE、SYBASE、INGRES、DB2 等 UNIX 上的大型数据库和 NT 上的 SQL Server，并且还支持 C-ISAM 文件系统。

应用部署平台

在 Tuxedo 中，应用的开发与部署是分开的，在开发时不用关心应用的部署问题。一个应用系统的配置信息都保存在一个文件中，通过修改该文件进行系统的部署。可根据系统的需要决定是把一个服务部署到一台服务器上还是多台服务器上，根据系统的负载情况决定该启动多少个服务进程等等。可以很容易地把在一种操作系统上开发的应用部署到另一个系统上，如把在 Windows NT 平台上开发的应用系统部署到 UNIX 上，应用的程序几乎不用做任何的改变，只要重新编译一下就可以了。

应用运行平台

Tuxedo 为应用系统提供运行环境，在运行时可根据系统的配置提供故障恢复、负载均衡、自动伸缩、安全认证和数据加密等功能，为 Tuxedo 应用系统提供一个可靠的运行环境。下面对一些主要的功能进行介绍。

1. 处理分布式事务

当 Tuxedo 应用系统只有局部事务，没有全局事务时，Tuxedo 直接利用数据库的事务处理功能（用或不用 XA），这样可提高系统性能。

当 Tuxedo 应用系统涉及跨多个数据库（同构或异构）的全局事务时，Tuxedo 利用 XA 标准和两阶段提交协议协调各个数据库的同步。Tuxedo 还可保证可靠性队列/Q 的入、出队操作和数据库的更新操作的同步。在各种情况下，Tuxedo 能够确保异构的（或者同构的）数据库以及其他资源管理器之间的完整性和一致，并且还能在出现节点故障、网络故障或全局资源死锁时利用内部的交易日志来协调全局事务的恢复。

2. 服务和节点的故障恢复

Tuxedo 可以对系统组件进行持续监视，查看应用系统、交易、网络及硬件是否出现故障。一旦出现故障，它支持备份服务器组继续运行，并透明地将故障组件从系统中排除，管理必要的恢复程序，重新为消息和交易选择路由；保证交易平台的高可用性。

3. 请求的重定向

Tuxedo 可以利用一些参数，如超时，使资源管理器可以在不中断进程的情况下将出现故障情况的节点上的服务器及服务移植到其他结点上重新运行。如果一个节点上的某些服务器出现故障，Tuxedo 还可自动地将客户的请求重新定向到可以完成相同服务的服务器上，使客户感觉不到变化。

4. 动态伸缩

在一个生产系统中，一般是不允许停机的，Tuxedo 应用系统可以在不停机的前提下，动态修改系统的配置，包括：

- (1) 增加新的 Tuxedo 系统应用服务器、数据库和服务进程组、服务进程和服务。
- (2) 修改现有的服务程序。