



目 录

目 录.....	2
第 4 章 Tuxedo 的基本概念.....	3
4.1 域 Domain.....	3
4.1.1 域的概念和范围.....	3
4.1.2 为什么要使用域.....	3
4.2 逻辑机器 Machine.....	3
4.2.1 Machine 的概念和范围.....	3
4.2.2 为什么使用 Machine.....	3
4.3 服务器组 Group.....	3
4.3.1 组的概念和范围.....	4
4.3.2 为什么要使用组.....	4
4.4 服务进程 Server 和服务 Service.....	4
4.4.1 什么是 Server 和 Service.....	4
4.4.2 Server 和 Service 的关系.....	4
4.4.3 服务进程中的主要函数.....	4
4.5 Tuxedo 通信方式综述.....	4
4.5.1 请求/ 应答式通信.....	4
4.5.1.1 同步方式 tpcall().....	5
4.5.1.2 异步方式 tpacall().....	6
4.5.1.3 嵌套调用.....	7
4.5.1.4 转发调用 tpforward().....	8
4.5.2 会话方式 tpsend()/tprecv().....	9
4.5.3 通知广播 tpnotify()/tpbroadcast().....	11
4.5.4 事件代理 tppost()/tpsubscribe().....	13
4.5.5 队列存储 tpenqueue()/tpdequeue().....	14
4.6 Tuxedo 多机部署.....	16
4.6.1 Tuxedo 集群.....	16
4.6.2 多套 Tuxedo 应用之间的通信.....	16
4.7 Tuxedo 远程客户端.....	16
4.7.1 什么是远程客户端.....	16
4.7.2 WSL/WSH 配置与工作机理.....	17
4.7.3 Java 远程客户端接入 Jolt.....	18

第 4 章 Tuxedo 的基本概念

Tuxedo 应用系统的部署，可以划分为以下几个层次：域 (Domain)，逻辑机器 (Machine)，服务器组 (Group)，服务进程 (Server)，服务 (Service)。

Tuxedo 的配置文件，称为 UBBCONFIG 或 ubb，正是对这些部署信息的定义。运行前，要把 UBBCONFIG 装载成二进制文件，称为 TUXCONFIG。

4.1 域 Domain

这里从域 (Domain) 开始，分层细述各级概念及其含义。

4.1.1 域的概念和范围

Domain 概念：一个 Tuxedo 应用系统就是一个 Domain。

域范围：Tuxedo 的域特性把客户/服务器模型扩展到多个应用系统。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。域同时也可能是一组运行在另一个非 Tuxedo 环境中的应用程序。Tuxedo 和 WebLogic 应用系统的互操作就是利用域的概念来实现的。

4.1.2 为什么要使用域

为了有效实现与其他系统的互连，Tuxedo 提出了 DOMAIN (域) 的概念，将由多台服务器共同组成的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。不同的 Tuxedo 应用域中的服务程序可以互相访问对方的服务，并且当一个交易同时执行多个应用域中的服务 (即对于分布式事务处理) 时，能够确保交易的完整性。同时，Tuxedo 系统可以指定哪些服务是可供外部应用域访问的，并可为这些服务设置访问控制表等安全认证手段，提高整个系统的安全性。

Tuxedo 应用系统：一个 Tuxedo 应用系统是由在一个 TUXCONFIG 文件中定义的资源及其客户端的总称，它只能有一个 TUXCONFIG 文件，一个 Tuxedo 应用系统能够通过域网关与别的 Tuxedo 应用系统或其他的应用中间件系统实现互操作。

4.2 逻辑机器 Machine

在域下面的一个概念就是逻辑机器 Machine。

4.2.1 Machine 的概念和范围

Tuxedo 对分布在多台机器上的同一个应用采用集中管理，因此 Machine 是在 Tuxedo 配置文件 UBBCONFIG 中第二个需要配置的段也是必须配置的段。它主要涵盖了一台物理服务器的基本信息，如主机名、Tuxedo 安装路径、应用程序路径、Tuxedo 配置文件路径等。在一个 UBBCONFIG 配置文件中可以配置多个 Machine，即 MP 模式。

4.2.2 为什么使用 Machine

Machine 节点中定义了 Tuxedo 应用系统的基本信息，如相关文件的存放位置以及本机可提供的客户端最大访问量、可提供的最多服务数等基本信息。

4.3 服务器组 Group

在逻辑机器 Machine 下面的一个概念就是服务器组 Group。

4.3.1 组的概念和范围

Group 只是一个逻辑概念，并不真实存在，它下面是一个个的 SERVER，每个 SERVER 也就是一个进程，同时每个 Group 又属于一个 Machine。

4.3.2 为什么要使用组

Group 在逻辑上起到一个承上启下的作用。一个应用可以开启多个进程，每个进程可以属于不同的组，组又可以指定到不同的 Machine。因此可以通过增加物理机器 Machine 和组 Group 以及 Server 来增加系统的处理能力实现高负载。组同样也是消息路由以及服务迁移的单位。在 XA 事务环境中，组还与资源管理器相关联。

4.4 服务进程 Server 和服务 Service

在服务器组 Group 下面的概念是 Server 和 Service。

4.4.1 什么是 Server 和 Service

Tuxedo Server 就是应用系统的一个进程，提供了对数据库或其他集中式资源的访问。

Tuxedo Service 从业务逻辑上来说就是对外提供的一种服务，是运行在 Server 上的一个函数，可命名的，客户端在调用时只需要知道 Service 的名字即可，不用知道 Service 运行在哪台机器的哪个 Server 上。

4.4.2 Server 和 Service 的关系

Server 是服务器上的一个进程，可以理解为一个程序，程序里有很多函数而这些函数就是 Service。Service 需要注册到 Server 里才能被客户端调用。

4.4.3 服务进程中的主要函数

Tuxedo 服务程序启动后，它总是保持运行状态，直到接收到一个 shutdown 命令为止。一个典型的 Tuxedo 服务程序在 shutdown 或 reboot 之前都在执行着数千个服务。部分服务进程中的函数如下：

- 1、在 Tuxedo 服务程序启动时，执行 tpsvrinit() 函数，可以在里面打开一些如数据库之类的资源供以后使用；
- 2、在 Tuxedo 服务程序终止时，执行 tpsvrdown() 函数，可以在里面关闭 tpsvrinit() 中打开的资源；
- 3、Tuxedo 服务程序以服务的形式来响应客户程序的请求，客户程序不是直接指定调用的服务程序的，而是调用服务，客户程序不知道处理它请求的服务程序的位置；
- 4、服务程序调用 tpreturn() 函数来结束服务请求，并返回一个缓冲区，必要时，将它传给客户程序