联动北方
LANDING

企业级 IT 运维宝典系列

联动北方 出品

目　录

第 7 章 Tuxedo 应用的部署模式

Tuxedo 使用 Domain 来组织应用程序。通常情况下，一个 应用程序由一个 Domain 组成，这种组织称为“单域”模式(Single-Domain Model)。在某些情况下，一个复杂的应用程序可能由多个 Domain 组成，这种组织模式称为“多域”模式(Multi-Domain Model)。

在“单域”模式下，一个应用程序既可以部署在一台物理主机上，又可以部署在多台使用高速局域网连接在一起的主机上。如果应用程序只部署在一台物理主机上，这种模式称为“单机”模式(Single Host Model,SHM)；如果应用部署在多台主机上，这种部署模式称为“多机”模式(Multi-Processor，MP)。

在“多域”模式下，一个应用程序由若干个逻辑上相对独立的 Domain 组成。这些 Domain 通过网关进程连接在一起，其中一个 Domain 既可以把自己实现的服务发布给其他 Domain，也可以从其他 Domain 中导入服务。多域中的每个域既可以使用单机部署，也可以使用多机部署模式。

单机模式、多机模式和多域模式是 Tuxedo 应用程序的 3 种组织模式。

7.1 单机 SHM 模式

单机模式是用的最多，也是最简单的一种组织模式。在这种模式下，所有业务处理进程，Tuxedo 系统进程和管理进程都部署在同一台物理主机上，这台主机既担负着域的管理任务，又担负着业务的处理任务。单机模式的网络拓扑结构比较简单，ubb 配置文件也很简单。

如下所示：

```
*RESOURCES
IPCKEY          123456
DOMAINID        simpapp
MASTER          SITE1
MAXACCESSERS    200
MAXSERVERS      120
MAXSERVICES     350
MODEL           SHM
LDBAL           N

*MACHINES
DEFAULT:
        TUXDIR="/home/tuxedo" #相关目录需要更改为您自己的
        APPDIR="/home/landingbj/simpapp"
        TUXCONFIG="/home/landingbj/tuxconfig"
        ULOGPFX="/home/landingbj/log"
        MAXWSCLIENTS=100
landingbj  LMID=SITE1
#landingbj 是主机名，UNIX 可通过 uname -n 获得，LIMD 表示主机逻辑 ID

*GROUPS
GROUP1  LMID=SITE1  GRPNO=1  OPENINFO=NONE
GROUP2  LMID=SITE1  GRPNO=2  OPENINFO=NONE
```

```
*SERVERS
DEFAULT:    RESTART=Y MAXGEN=10     GRACE=3600
simpserv    SRVID=1    SRVGRP=GROUP1 MIN=5 MAX=10
WSL         SRVID=10   SRVGRP=GROUP2
            CLOPT="-A  -t -- -n //192.168.0.168:7110
                   -m 10  -M 20  -x 10  -c 1024"

*SERVICES
TOUPPER    PRIO=50 LOAD=50
```

示例 7-1

这里再次强调一下，UBBCONFIG 文件具有如下关键字段需要注意（详细内容请参考第 6 章的解释和含义说明）：

*RESOURCE

定义了 Tuxedo 应用程序中使用到的域级参数，核心参数如下：

IPCKEY：Tuxedo 公告板（BB，Bulletin Board）的唯一 IPC 标识符；

MASTER：管理服务器，也称为主节点（Master）的逻辑名字；

MAXACCESSERS：最大的域访问量；

MAXSERVERS：一个域中可以配置的最大服务进程数；

MAXSERVICES：一个域中可以发布的最大服务数；

MODEL：指定单机域还是多机域；SHM=单机，MP=多机；

LDBAL：是否启用负载均衡；Y=启用负载均衡。

*MACHINES

指定域中包含的计算机。在 SHM 模式中，该节点只能包含一台计算机。在该节点中，必须设置 TUXCONFIG，并且它的值必须与 TUXCONFIG 环境变量中的值相同。

*GROUPS

Oracle Tuxedo 允许根据业务需要把后台服务分成组，如，账务服务可以定义在“FINAN_GRP”组中，人力资源服务可以定义在“HR_GRP”中。一个组中的所有服务必须部署在*MACHINE 段中定义的一台计算机上。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。

*SERVERS

定义域中部署的服务进程。它必须指定一个服务组。在上面的例子中配置了一个 simpserv 服务进程，属于 GROUP1 组。其中 DEFAULT 关键字用于设置一些通用的参数，这些参数对 DEFAULT 以下所有服务进程都有效。本例中设置了 RESTART、MAXGEN 和 GRACE 3 个参数，表示 simpserv 和 WSL 两个进程都是可重启的，即当 Tuxedo 系统发现它们处于 DEAD 状态时，就会尝试着重新启动它们。

SHM 应用的特点是，单机模式支持进程级的伸缩性和容错。

按照上面的这个配置，Tuxedo 在启动 simpapp 时，会启动 5 个 simpserv 实例。当压力增大时，用户可以启动更多的 simpserv 实例，但最多不超过 10 个；当压力减小时，用户可以终止 simpserv 进程实例，但最少要保留 5 个。

在系统运行过程中，如果某些 simpserv 实例异常终止了，Tuxedo 系统会把它们隔离起来，然后把客户请求派发给处于活动状态的实例去处理。

如果只一个进程只启动了一个实例，那么它就不具备进程级容错功能。比如，本例中的 WSL 进程，它只启动了一个实例，监听着 landingbj 主机的 7110 端口，如果这个进程死了，在它被 Tuxedo 系统重新启动之前，所有远程客户端将不能再连接到 Tuxedo 服务器上来做交易。

单机模式不支持主机级容错：如果 landingbj 主机因硬件错误不能再运行，那么整个 simpapp 将不能再对外提供服务。因此，大多数单机应用通常借助硬件 HA 方式来提供主机级容错功能。

7.2 多机 MP 模式

在多机模式下，一个 Tuxedo 应用程序需要部署在多台物理主机上，这些主机通过高速局域网连接在一起，并在 Tuxedo 系统的协调下，共同完成特定的任务。

MP 应用的构成：MP 的 MASTER 节点上面要运行 DBBL、BBL、BRIDGE、tlisten（以备 MASTER 迁移时使用） 4 个系统进程。MP 的其它节点上面要运行 tlisten、BRIDGE、BBL 3 个系统进程。此外，每个节点上都运行着若干个本地客户进程（Clients）和应用服务进程（Servers）。每个节点上的 BBL 都维护着一个本地公告板（Bulletin Board，BB），Master 节点上的 DBBL 负责协调使所有公告板的数据保持一致。

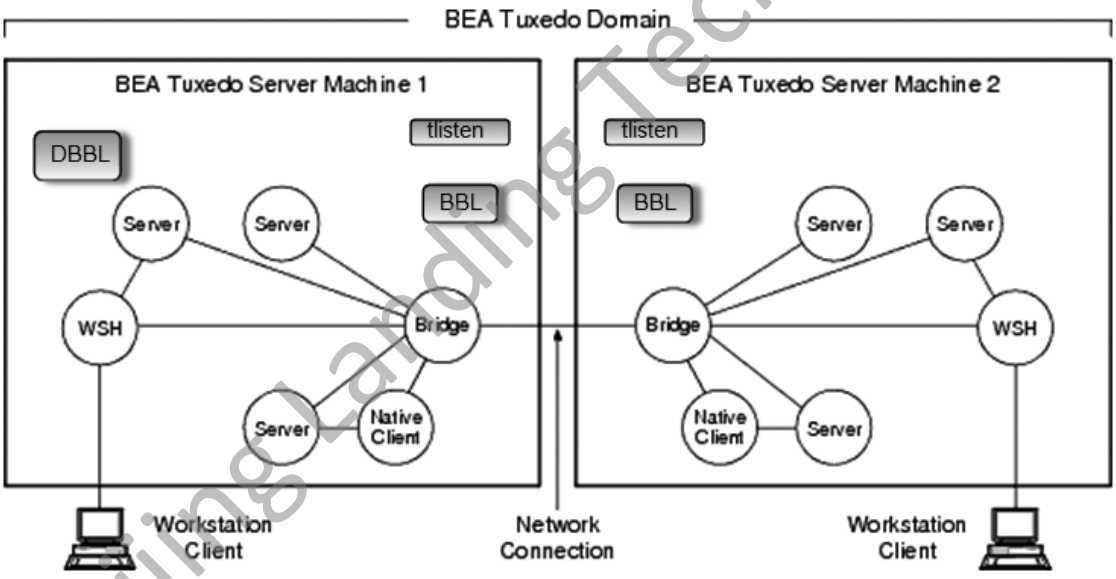

图 7-1

各个节点之间通过桥进程（BRIDGE）连接在一起，所有节点之间的消息交换都通过 BREDGE 进程来完成。在每一个节点的公告板中既可以找到本地服务的调用入口，也可以找到远程服务的调用入口。当客户机要调用一个本地服务时，它可以直接从公告板中找到提供这个服务的服务器的 IPC 请求队列，向这个队列发送请求；当客户机要个调用一个远程服务时，它从公告板中找到本地 BRIDGE 进程的消息队列，然后把请求消息放到里面，BRIDGE 会把请求消息路由到适当的远程服务器的请求队列，远程服务器处理完请求后，再将响应结果通过 BRIDGE 传回给发出个请求的客户机。这一切对客户机是透明的，也就是说客户机无需知道哪一个服务是本地的，哪一个服务是远程的。

Master 节点维护着整个应用程序配置文件的主拷贝（TUXCONFIG-original），在执行 tmboot 启动应用程序时，DBBL 会与成员节点上的 BBL 进行通信，并把配置文件复制到各个成员节点上，BBL 再根据配置文件（TUXCONFIG-copy）更新它们维护着的公告板。

从管理控制台（Administrative Console）上发出的对成员节点的管理指令，也是通过 BRIDGE 进程转发到远程的，远程节点的 BBL 进程收到管理指令后，修改公告板，完成管理任务。

tllisten 是一个运行在 Non-Master 节点上的后台网络监听进程，它接收来自 Master 节点的管理指令，然后启动本地的 BSBRIDGE 进程。因此在执行 tmboot 启动应用程序之前，所有节点上必须先启动 tlisten 进程。事实上，在启动应用程序的过程中，Master 节点上的 tlisten 并没有发挥任何作用，因此没有必要启动它。尽管如此，还是推荐吧 Master 节点上的 tlisten 也启动起来，以防将来因为系统故障而把 Master 节点降级为 Non-Master，把原来的 Non-Master 节点提升为 Master。

下表对 DBBL、BBL、BRIDGE、tlisten 4 个进程的用途做一个总结：

进程	说明
DBBL	Distinguished Bulletin Board Liaison(DBBL)，作用是记录所有 BBL 的状态，并负责与它们联络，保持各个主机上的公告板的数据同步
BBL	Bulletin Board Liaison(BBL)，作用是维护本地公告板，记录所有服务器和服务的状态，定期对 Tuxedo 系统作健康检查，与 DBBL 通信保持公告板上的数据的同步
BRIDGE	BRIDGE 进程被称为桥进程，它的作用是负责各个节点间的数据通信。这个进程在 tmboot 时会自动启动，不需要在 UBBCONFIG 文件中对它进行配置
tlisten	tlisten 是一个在后台独立运行的网络监听进程，它的作用是从管理控制台、公告板或命令行接收命令，然后启动 BBL、BRIDGE 等管理进程

表 7-1

MP 应用的启动流程相对比较复杂，如下图 7-2 展示了一个两节点 MP 应用程序的启动过程：

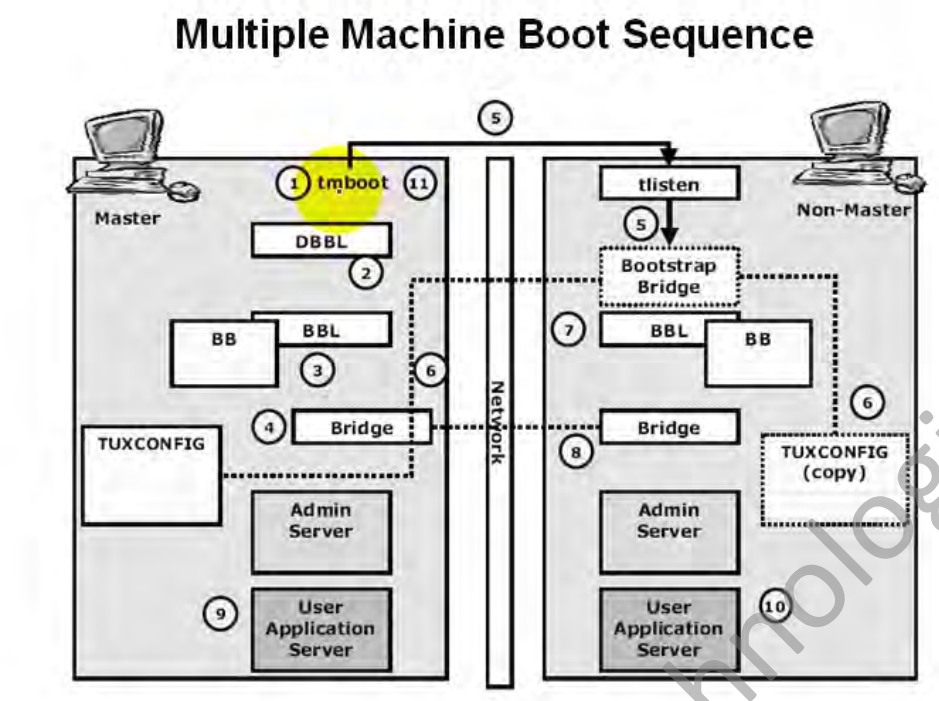

图 7-2

具体启动步骤如下：

1、在 Master 机器上执行 tmboot 启动服务(在非 Master 机器上已启动 tlisten 进程)；

2、在 Master 机器上首先启动 DBBL 进程；

3、启动 BBL 管理进程；

4、启动 Bridge 进程，Bridge 进程与非 Master 机器建立连接，此时非 Master 机器上所有服务都还没有启动（除了 tlisten 进程外）；

5、tmboot 连接到非 Master 机器上的 tlisten 进程，tlisten 进程启动 BSBRIDGE(bootstrap Bridge)进程；

6、BSBRIDGE 进程连接到 Master 机器上的 Bridge 进程，并向 Master 索求 TUXCONFIG 配置文件，TUXCONFIG 配置文件传递到非 Master 机器上；

7、启动远端机器的 BBL 创建 BB；

8、启动远端机器的 Bridge 进程，同时 BSBRIDGE 进程终止，此时两台机器的 Bridge 进程建立了连接，此后两台机器之间的通信主要通过 Bridge 进程；

9、启动 Master 机器上的其他进程；

10、启动非 Master 机器上的其他进程。

MP 应用的配置也相对稍微复杂一些，如下所示为一个简单的 MP 配置:

```
*RESOURCES
IPCKEY                  123456
```

```
#SECURITY              APP_PW
MASTER                 SITE1 ,SITE2
MAXACCESSERS           450
MAXSERVERS             350
MAXSERVICES            350
MODEL                  MP
OPTIONS                LAN
ENCRYPTION_REQUIRED    N
LDBAL                  Y
BLOCKTIME              300

*MACHINES
vmLinux      LMID=SITE1
             APPDIR="/home/landingbj/simpapp"
             ULOGPFX="/home/landingbj/log/ULOG"
             TUXCONFIG="/home/landingbj/simpapp/tuxconfig"
             TUXDIR="/home/tuxedo"
             MAXWSCLIENTS=200

vmWin        LMID=SITE2
             APPDIR="C:\landingbj\bin"
             TUXCONFIG="C:\landingbj\bin\tuxconfig"
             TUXDIR="C:\bea\tuxedo"
             UID=0
             GID=0
             MAXWSCLIENTS=200

*GROUPS
GMSTR          LMID=SITE1        GRPNO=60
GBKUP          LMID=SITE2        GRPNO=100

*NETWORK
SITE1          NADDR="//192.168.0.168:6001"
               NLSADDR="//192.168.0.168:6002"
SITE2          NADDR="//192.168.0.169:6001"
               NLSADDR="//192.168.0.169:6002"

*SERVERS
DEFAULT:
               CLOPT="-A" SYSTEM_ACCESS=FASTPATH
WSL            SRVGRP=GMSTR       SRVID=100 RESTART=Y MAXGEN=5
               CLOPT="-A -- -n //192.168.0.169:12345 -m 3"
simpserv       SRVGRP=GMSTR       SRVID=1
simpserv       SRVGRP=GBKUP       SRVID=10
```

示例 7-2

MP 配置文件并不复杂，在编写的时候要注意以下几点：

(1) 在*RESOURCES 段，MODEL 必须为 MP，表示当前是一个 MP 应用；

(2) OPTIONS 必须配置为 LAN，表示当前应用跨越网络，部署在多台服务器上；同时也可以加上 MIGRATE 选项，表示可迁移；

(3) MASTER 参数中可以加入一个备用 Master 节点；

(4) 把 LDBAL 参数设置为 Y，表示激活负载均衡算法；

(5) 在*MACHINES 段中，必须加入参与计算的每一个节点的配置信息；

(6) 同时可以使用 NETLOAD 为每个节点指定网络负载因子，这个参数表示把一个请求从本地转发到远程去处理要付出的代价，取值越大表示有更多的请求会在本地处理。

此外还有两个与 MP 模式相关的重要参数，它们是 BBLQUERY 和 DBBLWAIT。

(7) BLQUERY 用于设置 DBBL 定期检查 BBL 状态的时间间隔，默认值为每隔 300 秒检查一次。如果 BBL 没有回应，则 DBBL 就会认为它所管理的 Tuxedo 系统不再是 MP 的一部分，并且将它隔离出去；

(8) DBBLWAIT 是 DBBL 在给某个 BBL 发送消息后，等待它作出回应的最长时间，默认为最多等待 20 秒，如果 BBL 没有回应，DBBL 就认为它们已经死了；

MP 应用的特点是，容错能力强与负载均衡。

MP 应用程序具有容错能力强、可靠性高的特点。首先，Tuxedo 运行管理机制提供了对软件错误的自动监控和修复功能。每一个管理进程（如 DBBL、BBL、BBRIDGE 等），以及处理业务逻辑的服务进程都被 Tuxedo 系统自动监控起来，当某个进程出现错误时，Tuxedo 都会自动尝试着去修复它，而不需要人工干预。

在 MP 应用程序中，DBBL 会定期向各个节点的 BBL 进程发送询问消息，BBL 收到询问后，会向 DBBL 发送类似于“我还活着”的回应消息，这种机制被称为“心跳监测”。如果 DBBL 在特定的时间内得不到 BBL 的回应，就会认为 BBL 已经死了，并试着去重新启动它。如果该节点上的 BBL 不能被启动，DBBL 就会把该节点标记为“Partitioned”的状态，然后自动隔离起来。

Master 节点上的 BBL 反过来也会监控 DBBL 的状态，必要时也会尝试着去启动它。如果 DBBL 不能被启动，就需要管理员手工地将 DBBL 迁移到备用节点上。每个节点上的业务进程都可以配置可重启属性。在例行健康检查时，当 BBL 检查到某个进程死了，也会试着去重新启动它。

其次，MP 应用程序支持主机级容错。即便是构成 MP 的某几个节点死了，应用程序一样能够不间断地对外提供服务。MP 模式的主机级容错特性有别于硬件 HA 容错。在硬件 HA 模式下，只有当主节点发生故障时，备用节点才会接替它的工作，否则备用节点始终处于空闲状态。在 MP 模式下，所有的节点都参与计算，某些节点发生故障时，会被自动隔离出去。

第三，MP 应用程序的远程客户机访问可以通过设置 WSNADDR 环境变量来提高容错能力。假设 1 个 MP 应用由 3 个节点组成，它们的地址分别为 IP1、IP2 和 IP3，每个节点上都运行着 WSL 进程，端口分别为 PORT1、PORT2 和 PORT3，则客户端可以通过下面的方式来设置 WSNADDR，以提高容错能力：

```
SET WSNADDR=//IP1:PORT1,//IP2:PORT2,//IP3:PORT3
```

示例 7-3

在这种情况下，客户机会依次尝试着使用这 3 个地址来连接服务器，如果第一个连不通就会尝试第二个，第二个连不通就会尝试第三个，如果都连不通，tpinit()才会报错。

另外在 MP 模式下，负载均衡功能默认是打开的，即 LDBAL 取值为 Y。这样，所有节点都会均匀分摊客户机请求造成的负载。然而，节点之间并不会相互通知彼此的负载信息，即每个节点并不知道其他节点负载的准确值，只能作一个大致的估计。因此，不同的远程节点对同一个节点的负载估计完全可能是不一样的。

对于同一个服务请求，假设在本地处理的负载为 LOAD，转发到远程处理要付出的额外代价为 NETLOAD，则每当本地处理了（1+NETLOAD/LOAD）个请求后，就会把一个请求转给远程处理。假设 LOAD=50，NETLOAD=100，则每当本地处理 3 个请求之后，就会把 1 个请求转给远程处理。

可以在*MACHINES 段中为每个节点指定 NETLOAD，也可以通过 TMNETLOAD 环境变量来指定，取值越大，就会有更多的请求在本地处理。

在远程客户端，按照下面的方式设置 WSNADDR 环境变量也可以达到 3 台服务器之间负载均衡的目的。

```
SET WSNADDR=(//IP1:PORT1^|//IP2:PORT2^|//IP3:PORT3)      #Windows
WSNADDR=(//IP1:PORT1|//IP2:PORT2|//IP3:PORT3)            #UNIX
```

示例 7-4

每一次客户机执行到 tpinit()调用时，都会从 3 个服务器地址中任选一个来建立连接。

客户机可以和任何节点建立连接，如果该节点没有提供客户机请求的服务，Tuxedo 就会把请求转发到其他提供了该服务的节点上去处理，如果多个节点都提供了同一个服务，那么 Tuxedo 就会把请求送到负载最小的那个节点去处理。

7.3 多域模式

企业的 IT 部门通常会按功能、地理位置和机密等级来把企业的 IT 系统规划成若干个子系统，这些子系统在管理上是完全独立的，它们既可以独立的工作，又可以通过网络连接在一起，实现协同工作和应用集成，形成一个统一的信息平台，为不断变化和增长的企业业务提供服务。随着时间的推移，各类子系统会越来越多，如何把这些子系统很好地整合起来，是企业 IT 部门必须应对的挑战。

Tuxedo 引入了域的概念。域是 Tuxedo 分布式应用程序的一种组织方式和管理单元，一个域就是一个独立的应用系统，它既可以由单个节点构成（SHM 模式），又可以由多个节点构成（MP 模式），它的配置通过一个 TUXCONFIG 文件来描述。

在企业计算环境中，一个 Tuxedo 域可以使用特定网关去连接其他域，实现业务集成。一个 Tuxedo 域可以使用 TDOMAIN 网关去连接另一个 Tuxedo 域，也可以使用 TDOMAIN/WTC 去连接一个 WebLogic 域。如果企业计算环境中存在 TOPEND 域或主机应用，Tuxedo 域也可以使用 TEDG 或 TMA 网关去连接它们。

一个域可以把其它域的服务导入到本地，也可以把本地服务发布给其他域。对于客户端而言，本地服务和远程服务是完全透明的，即本地客户端可以像调用本地服务一样去调用一个远程服务，反之亦然。域和域之间支持安全上下文传递，可以为跨域的服务调用设置安全验证和访问控制。域和域之间还支持事务上下文的传递，多个跨域的操作可以纳入到同一个全局事务中来进行事务控制。在 Tuxedo 和 WebLogic 之间，还可以跨域实现单点登录(Single Sign On,SSO)。

在 MP 模式下，所有节点必须位于同一个局域网中，而在多域模式下，连接两个域之间的链路可以是不可靠的广域网。如果要跨越网关传递大量数据，可以在网关上配置压缩特性；如果要通过网关传递重要数据，可以在网关上配置加密特性。

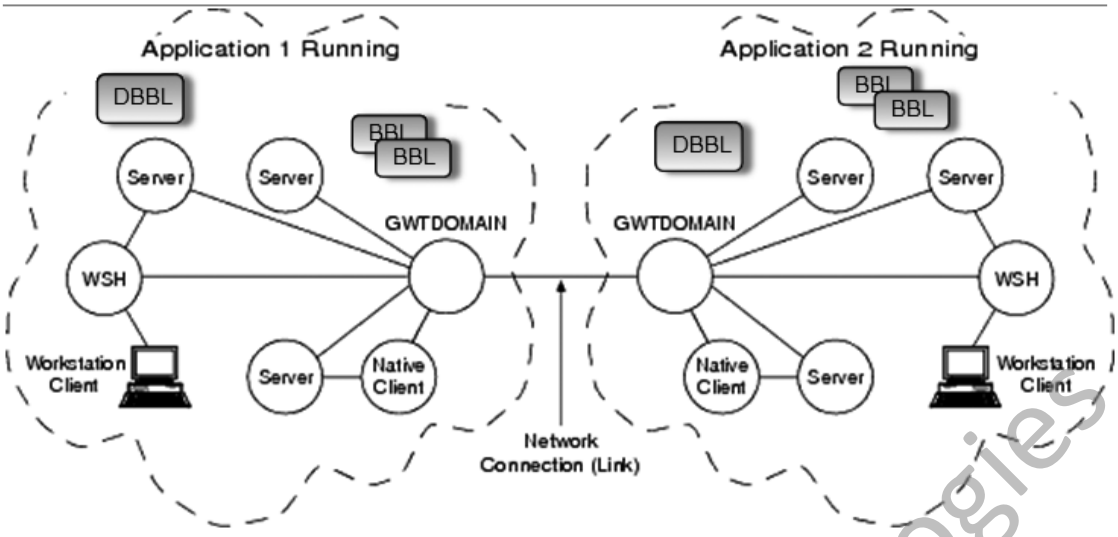

图 7-3

Tuxedo 提供了 3 类域网关来连接不同的系统，它们是 TDOMAIN、TOPEND 和 TMA。根据使用协议和目的的不同，可以把 TMA 网关分为 TCP、SNA、OSI TP 3 类。

TDOMAIN 网关是基于 TCP/IP 设计的，它的进程名是 GWTDOMAIN，常用于连接其他 Tuxedo 域和 WebLogic 域。在连接 WebLogic 时，WebLogic 一端使用的是 WTC。TOPEND 网关也是基于 TCP/IP 网络设计，进程名是 GWTOPEND，它为 Tuxedo 系统连接 TOPEND 系统提供支持，从 9.0 开始，Tuxedo 将不再支持这个网关。TMA for Mainframe TCP 网关进程名是 GWIDOMAIN，它为 Tuxedo 系统连接 IBM OS/390 的 CICS 和 IMS 系统提供支持。TMA for Mainframe SNA 网关的进程名是 GWSNAX，它为 Tuxedo 系统连接 IBM OS/400、OS/390 CICS 和 IMS、VSE/CICS 等使用 SNA 网络的系统提供支持。TMA for Mainframe OSI TP 网关进程名是 GWOSITP，它为 Tuxedo 系统连接其他 OSI TP 系统（如 IBM CICS、Ecina、Tong/Easy 等）提供支持。

连接不同类型的远程域需要使用不同的网关（如连接 WebLogic 使用 GWTDOMAIN，连接 TOP END 域使用 GWTOPEND），但对于同一类型的多个远程域来说，既可以使用一个网关也可以使用多个网关。如果 1 个本地 Tuxedo 域要和 3 个远程 Tuxedo 域建立连接，那么有两种方案可以选。第一种方案是在本地域中使用一个网关去连接三个远程域，第二种方案是在本地域使用 3 个网关分别去连接 3 个远程域。第一种方案的优点是网关数量少，易于管理和监控，缺点是网关压力大，没有办法针对每一个远程域设置个性化的连接策略。第二种方案的优点是可以设置个性化连接策略，网关压力小，多网关之间可以设置 Failover，缺点是网关进程多，不易于管理。

每个域都有一个域管理进程 DMAMD,它管理着域的配置文件 BDMCONFIG 和网关组。每个组有一个网关管理进程 GWADM 和一个网关进程，GWADM 管理着网关。网关进程负责域之间的通信，它可以把远程域的服务导入到本地，并在 BB 中发布它们，使本地客户端可以调用它们。

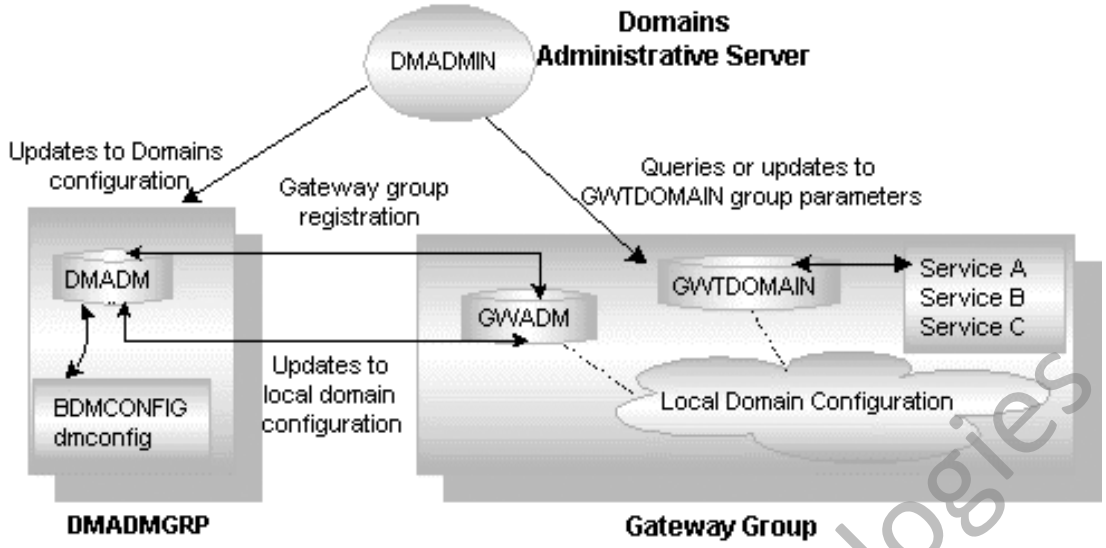

图 7-4

下表 7-2 详细的描述了这些进程的作用：

进程	说明
DMADM	域管理进程，管理着配置文件 BDMCONFIG 和网关组，它发布了一个同名的注册服务，每个网关组的管理进程 GWADM 在初始化时，都要调用这个服务来注册网关
GWADM	运行时域网关管理服务器进程，它从 DMADM 服务器上获取信息，并定期告诉它“我还活着！”
GWTDOMAIN	TDOMAIN 的网关进程，负责导入远程域的服务，并在本地域的 BB 中发布它们。它是本地域对远程域的访问点，负责域之间的可靠消息传递、数据加密、压缩和安全认证等

表 7-2

如果一个域有 n 个网关组，那么域管理进程、网关管理进程和网关进程的数量总和为 2n+1 个。多域应用中的每个域可以独立启动，在什么时候和远程域建立连接，将由域网关上配置的连接策略决定。

关于多域应用基本配置，Tuxedo 使用一个单独的配置文件 DMCONFIG 来保存域的配置信息。DMCONFIG 和 UBBCONFIG 文件非常类似，也有 ASCII 和二进制两个版本。ASCII 版由管理员编写，在启动之前，必须使用 dmloadcf 把它转换成二进制的形式，并通过 BDMCONFIG 环境变量指向它。

DMCONFIG 文件由若干个段组成，下面就按顺序对它们关键点提示：

*DM_RESOURCES

这个段定义全局的域配置信息，目前只有一个 VERSION 参数。管理员可以用这个参数来标记域配置文件版本，取值是一个字符串，除此之外没有别的意义。这个段不是必须的，可以不定义。

```
    VERSION=Landingbj_V1
```

示例 7-5

*DM_LOCAL

这个段的别名是*DM_LOCAL_DOMAINS，用于定义本地域网关访问点。每个域网关对应着一个访问点，如果本地域有多个网关，则必须在这里定义多个访问点。

下面的例子定义了一个本地域网关访问点 DOM1：

```
    DOM1   GWGRP=LGWGRP    TYPE=TDOMAIN     DOMAINID=“DOM1”
```

示例 7-6

在这个段中，GWGRP、TYPE、DOMAINID 参数是必须的，其他参数都是可选的。GWGRP 指定了网关进程所属的组名 LGWGRP，它必须在 ubb 文件中定义过。DOMAINID 的别名是 ACCESSPOINTID，它用于定义网关标识，在配置安全特性和 Failover 时，会用到这个标识。TYPE 用于定义网关的类型，可取值为 TDOMAIN、TOPEND、SNAX、OSITP 和 OSITPX。

*DM_REMOTE

这个段的别名是*DM_REMOTE_DOMAINS，用于定义远程域网关访问点。

下面的例子定义了一个远程域网关访问点 RMDOMAIN1：

```
    RMDOMAIN     TYPE=TDOMAIN      DOMAINID=“RMDOMAIN1”
```

示例 7-7

这个段中，DOMAINID 和 TYPE 参数是必须的，其他参数是可选的。

*DM_TDOMAIN

这个段为本地网关和远程网关指定监听地址和端口。远程网关通过本地网关的监听地址来建立连接和发送请求，本地网关通过连接到远程网关的监听地址上去发送服务请求。

下面的例子定义了两个网关访问点的监听地址:

```
    DOM1         NWADDR=“//192.168.0.168:2507”
    RMDOMAIN1    NWADDR=“//192.168.0.169:3186”
```

示例 7-8

在这个段中，只有 NWADDR 是必需的，其他的可选参数还有：NWDEVICE（指定使用的网络设备）、CMPLIMIT（指定压缩阀值）、TCPKEEPALIVE（指定是否保持 TCP 连接）。

*DM_ACCESS_CONTROL

这个段用于定义访问控制列表（ACL），以控制哪些远程域可以请求本地的服务。

下面的例子定义了一个名为 MY_ACL 的访问控制列表：

```
    MY_ACL       ACLIST=RMDOMAIN1,RMDOMAIN2
```

示例 7-10

*DM_EXPORT

这个段的别名是*DM_LOCAL_SERVICES，用于定义本地域导出的服务，这些服务可被远程域导入。在默认情况下，所有的本地域的服务都是可以导出的，这些服务继承了在 TUXCONFIG 中定义的属性，如 LOAD、PRIO、AUTOTRAN、ROUTING、BUFTYPE 和 TRANTIME。

下面的例子是为本地域导出的服务 TOLOWER 定义了 ACL 和服务别名：

```
    TOLOWER     RNAME=LOWERCASE   ACL=MY_ACL
```

示例 7-11

这个例子说明，只有来自 RMDOMAIN1 和 RMDOMAIN2 网关的请求才能调用 TOLOWER 服务，来自其他网关的请求会被拒绝。RNAME 为本地服务提供了重命名功能，当远程域导入 TOLOWER 时，服务名不再是 TOLOWER，而是 LOWERCASE。

这个段用于定义本地服务的参数有 RNAME、LDOM、ACL 和 CONV。RNAME 用于指定导出服务的别名。LDOM 的别名是 LACCESSPOINT，用于指定导出网关。ACL 用于指定一个 *DM_ACCESS_CONTROL 中定义的 ACL 名，CONV 可取值 Y 和 N，用于指定导出的服务是否为会话类型。

*DM_IMPORT

这个段的别名是*DM_REMOTE_SERVICES，用于定义从远程域中导入的服务。下面的配置用于从远程域导入 TOUPPER 服务。LDOM 的别名是 LACCESSPOINT，它指定了本地网关访问点是 LAPP。RDOM 的别名是 RACCESSPOINT，它指定了 TOUPPER 的远程网关访问点是 UAPP。当本地调用 TOUPPER 时，服务请求将由 LAPP 传递给 UAPP

```
    TOUPPER     LDOM=DOM1      RDOM=RMDOMAIN1       RNAME=UPPERCASE
```

示例 7-12

这个段中用于定义远程服务的参数还有 CONV、LOAD、RNAME、ROUTING 等。与 *DM_EXPORT 段的同名参数相同， RENAME 用于指定导入服务的别名，CONV 用于指定服务是否为会话类型，另外，LOAD 用于指定负载因子，ROUTING 用于指定路由规则。

*DM_ROUTING

这个段用于定义多个域之间的数据依赖路由规则，这些规则可以被* DM_REMOTE_SERVICES 段中的服务引用。

下面的例子定义了一个路由规则 ON_ACCOUNT，这个规则被 ACCT_BAL 服务引用：

```
    *DM_REMOTE_SERVICES
       ACCT_BAL    LDOM=DOM1   ROUTING=ON_ACCOUNT
    *DM_ROUTING
       ON_ACCOUNT BUFTYPE=“FML32”  FIELD=ACCOUNT_ID
       RANGES=“0-9999：RMDOMAIN1，*：RMDOMAIN2”
```

示例 7-13

ACCT_BAL 是一个导入服务，RMDOMAIN1 和 RMDOMAIN2 两个域都提供了这个服务，它在本地的网关访问点是 DOM1，请求缓冲区类型是 FML32。路由规则 ON_ACCOUNT 的作用是，如果请求缓冲区中的 ACCOUNT_ID 字段取值在 0～9999，之间，则本地域将通过 DOM1 把请求转发给 RMDOMAIN1 域，否则将通过 DOM1 转发给 RMDOMAIN2 域。

*DM_ROUTING 与 UBBCONFIG 文件中的*ROUTING 的用法基本是一样的，差别仅在于 UBBCONFIG 中*ROUTING 的 RANGES 参数中，取值范围对应的是组名，而在 DMCONFIG 中 *DM_ROUTING 中取值范围对应的是远程网关访问点。