

目 录

目 录.....	2
第 3 章 WebLogic 的基本概念.....	3
3.1 域 Domain.....	3
3.1.1 域的概念、范围和限制.....	3
3.1.2 为什么要使用域.....	4
3.2 服务器.....	5
3.2.1 管理服务器.....	5
3.2.2 受管服务器.....	6
3.3 计算机 Machine.....	6
3.3.1 Machine 的概念.....	6
3.3.2 为什么要使用 Machine.....	7
3.4 集群.....	7
3.4.1 什么是集群.....	7
3.4.2 主要功能.....	8
3.4.3 基本集群架构.....	8
3.4.4 基本集群架构的优缺点.....	8
3.4.5 多层集群架构.....	8
3.4.6 何时使用多层集群架构.....	9
3.4.7 多层集群架构的优缺点.....	9
3.4.8 代理服务器.....	10
3.4.9 基本集群代理架构.....	10
3.4.10 多层集群代理架构.....	10
3.4.11 生产模式与开发模式.....	11

第 3 章 WebLogic 的基本概念

这一章我们会讲述 WebLogic 涉及到的一些基本概念，对它们有个宏观而概括的理解，才能比较好的体会后续的各个细节，因为可以在一个大的框架下来审视。

3.1 域 Domain

3.1.1 域的概念、范围和限制

“域(Domain)”就是逻辑上相关的一组 WebLogic Server 资源，可以作为一个单元进行管理，它是 WebLogic 一个相对完整的独立范围。

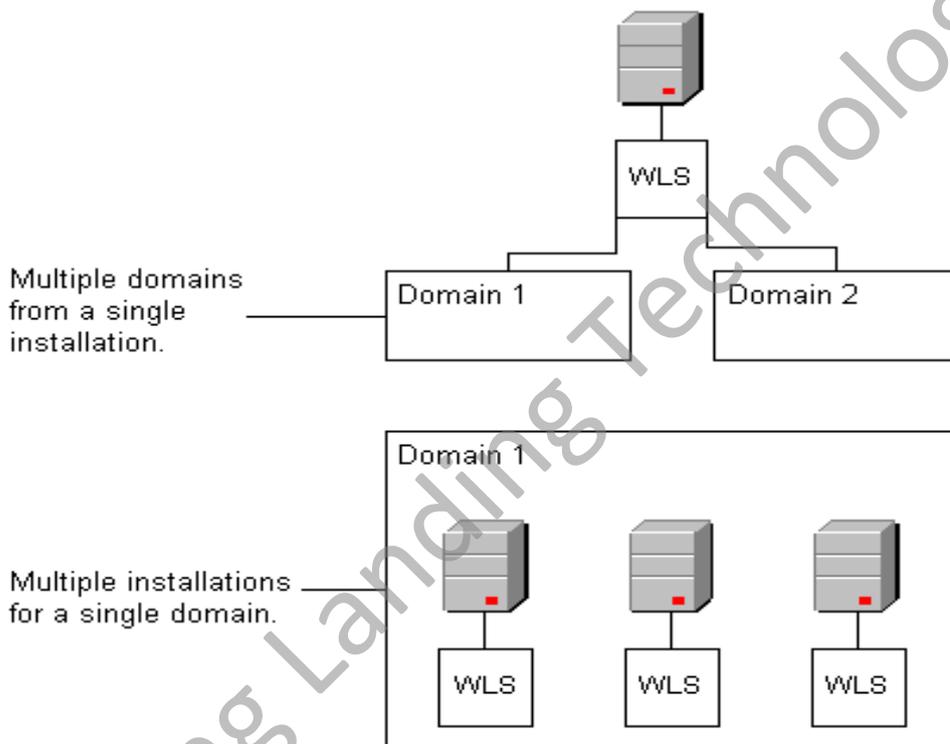


图 3-1

其大致可以分为以下几种类型：

1. 含有受管服务器的域：

简单的生产环境可以只有一个域，该域包含承载应用程序的多个受管服务器和执行管理操作的一个管理服务器。

在该配置中，应用程序和资源部署在单个受管服务器上；同样，访问应用程序的客户端也连接至一个受管服务器。

2. 独立服务器域：

在开发或者是测试环境中，您可能会在生产域中的服务器之外，部署一个独立的 应用程序和服务器。可以部署一个简单的域，其中包含一个服务器实例来充当管理服务器并承载您所开发的应用程序。

随 WebLogic Server 一起安装的 examples 域便是独立服务器域的一个示例。

可以根据以下这些条件组织域：

- 应用程序的逻辑划分：例如一个程序专用于提供用户最终功能，而另一个专用于后端的财务管理。
- 物理位置：针对不同的地域或业务分支机构建立域；
- 规模大小：以小型单元组织域，有助于更高效的管理域，这些域可能由不同的人员进行管理。从概念上讲，一个域就是一个管理点；

另外，你的域配置，请注意下列限制：

- 每个域要求其履行自己的管理服务器管理活动，当您使用管理控制台来执行管理和监测任务，你可以来回切换域之间，但在这样做的话，你要连接到不同的管理服务器。
- 在集群中的所有 Managed Servers 必须位于同一个域中，你不能分割多个域集群。
- 在一个域中的所有 Managed Servers 必须运行的 Oracle WebLogic Server 软件同一版本。

3.1.2 为什么要使用域

WebLogic Server 域是一种管理功能，WebLogic Server 编程不涉及域。所有与域相关的信息都在配置文件中，只有管理员需要考虑“域”。

“域”是一种管理功能具有以下特点：

- 对应用程序是透明的；
- 可以根据技术或者是业务需要对其进行配置和管理，即使是在部署应用程序之后或在生产使用中。

WebLogic Server “域”可用于区分：

- 开发应用程序、测试应用程序和生产应用程序；
- 管理和运营职责；
- 组织或业务划分；

综合来讲，域的多个优势大致有：

一个企业可能有多种不同的应用程序，他们在地理上可能是分散，也可能被组织到不同的职责领域中，因此可能有多个独立的域。

每个域都是一个单独的管理单元。可以根据地理因素(位于同一位置的所有计算机)对其进行组织。也可以根据企业中的部门进行划分(财务、制造、运输)对其进行组织。

最后，企业可能要求其多个域中的应用程序能够进行互操作。通常可能的做法是将一个域扩展以包容整个企业；但扩展的域会因计算机和服务的数量众多而变的无法管理。因为一个域必须作为一个管理单元来管理，所以配置将会快速变的非常庞大，管理难度要远大于开发和实现应用程序。

要保持域相对紧凑，易于管理，必须有一种方法将应用程序划分到多个域，并且使一个域中的应用程序仍然可以访问其它域中的服务，WebLogic Server 域功能可以实现域之间的通信。

3.2 服务器

“服务器”是一个在 JVM 中执行的 weblogic.server 实例，WebLogic Server 只是在执行一个 weblogic.server 类的 java 虚拟机。

weblogic.server 类是包含有 main() 方法，用于启动 WebLogic 系统。由于服务器在单个 Java 虚拟机中执行，它占用专用的 RAM 容量，该容量是在系统引导时确定的。一个服务器仅可以与一台计算机相关联，但一台计算机可以与多台服务器相关联。

3.2.1 管理服务器

管理服务器的定义：是域的中央控制点，存储域的配置信息和日志，并运行 WebLogic 管理控制台。

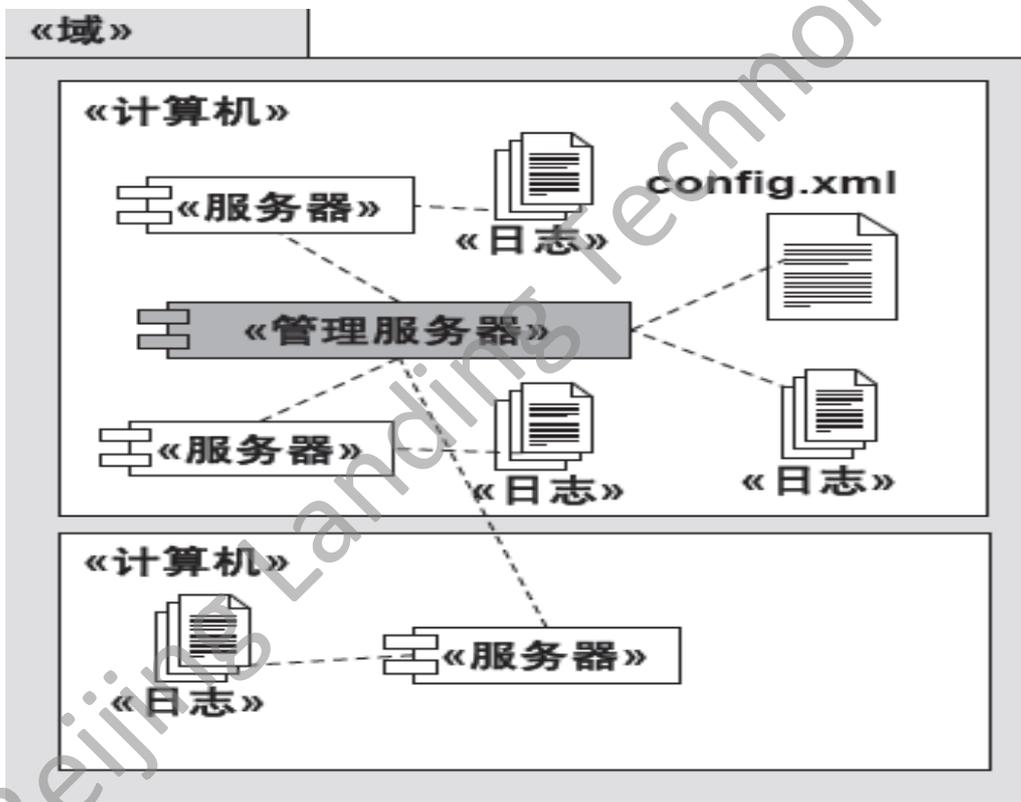


图 3-2

管理服务器是用作配置整个域的中央控制实体。

可以通过以下方式调用管理服务器的服务：

- WebLogic Server 管理控制台：管理控制台是用来配置域的基于浏览器的图形界面 (GUI)；
- WebLogic Server 应用程序编程接口 (API)：您可以使用 WebLogic Server 提供的 API 编写 Java 类来修改配置特性；

- WebLogic Server 命令行实用工具(weblogic.Admin)创建脚本实现自动化的域管理;
- SNMP: 可以实现简单的网络管理协议来监视 WebLogic Server 域。

要修改域配置, 域管理服务器必须正在运行。管理服务器是通过写 config.xml 文件来维护域的配置信息。

3.2.2 受管服务器

受管服务器的定义: 域中任何不属于管理服务器的服务器, 与管理服务器联系以获得配置信息, 在生产环境中运行业务程序。

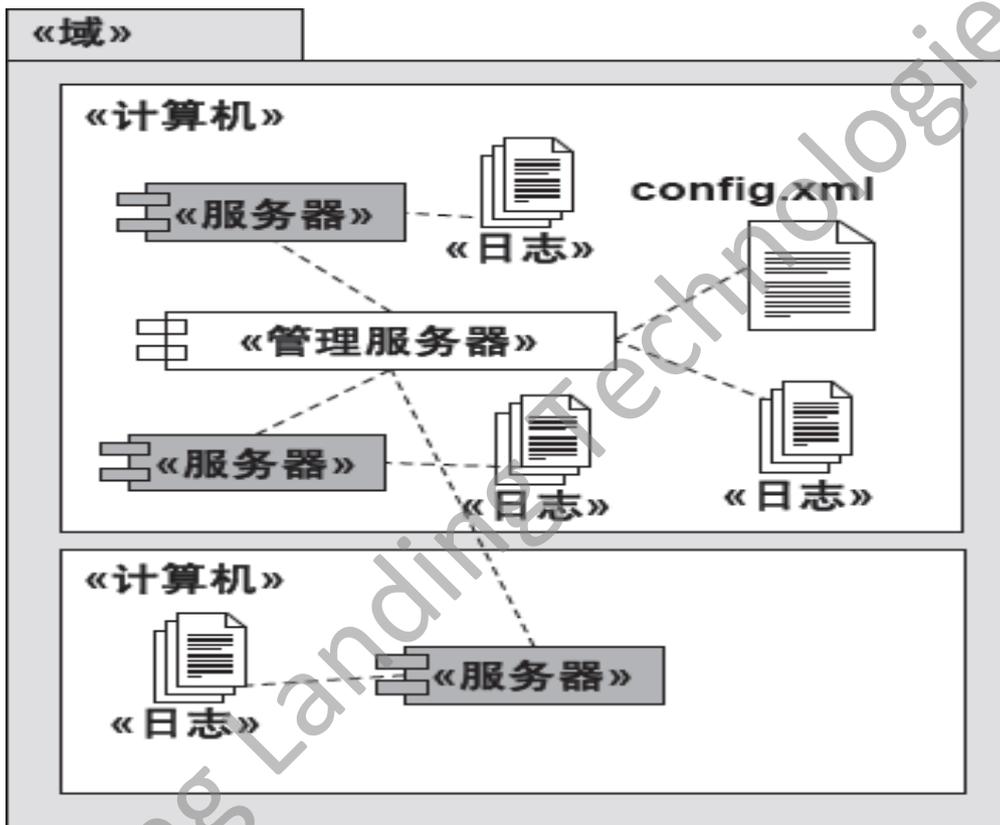


图 3-3

“受管服务器”是一个 WebLogic Server 实例, 它从管理服务器中检索域配置数据。域中可以有多个受管服务器, 但是只有一个管理服务器。

通常, 您在创建并启动作为受管服务器的服务器实例在生产环境中运行业务应用程序。在该标准场景中, 作为管理服务器启动的服务器实例不会运行业务应用程序, 它仅管理域中的资源。

为提高可靠性和性能, 您可以在多台计算机上安装 WebLogic Server 软件, 并在不同的 WebLogic Server 主机上运行您创建的服务器。

3.3 计算机 Machine

计算机 Machine 顾名思义, 一般是和真实的物理机器相对应, 如果说服务器 Server 是一个逻辑概念的话, 与其相对, 计算机 Machine 则是一个物理概念。

3.3.1 Machine 的概念

Machine 的定义：是承载一个或多个 WebLogic Server 的计算机，运行受支持的操作系统平台并且可以承载多个 WebLogic Server 实例。

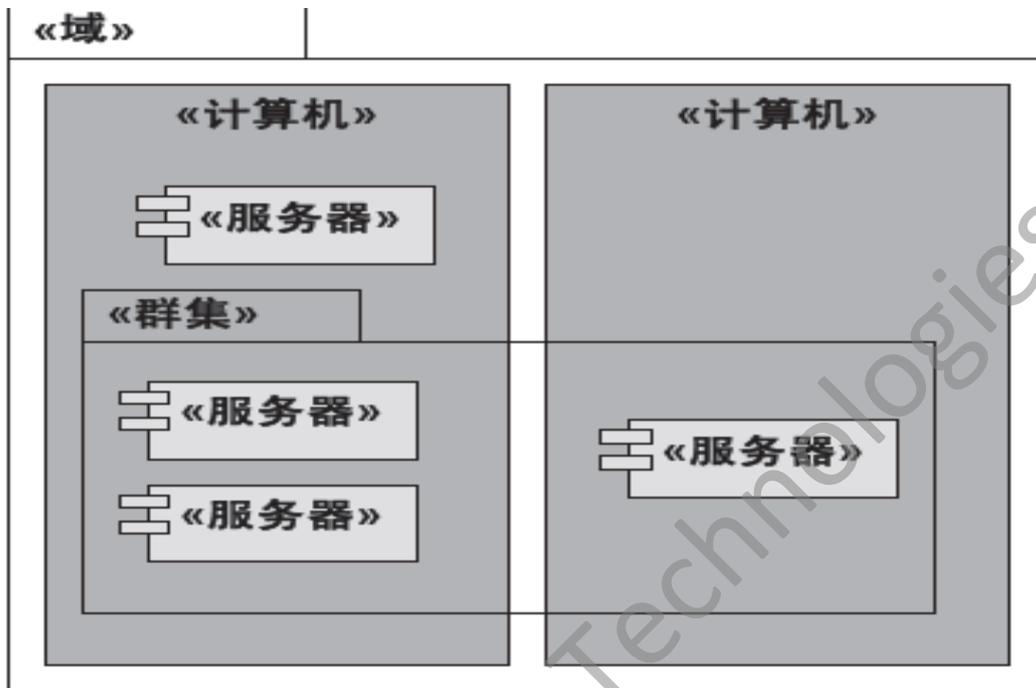


图 3-4

“计算机”表示承载一个或多个 WebLogic Server 实例的物理计算机。

WebLogic Server 使用所配置的计算机名来确定将任务(如 HTTP 会话复制)委托给集群中的哪个服务器是最佳的。管理服务器使用此计算机定义与节点管理器应用程序来启动远程 WebLogic Server 实例。

3.3.2 为什么要使用 Machine

在 WebLogic 体系中引入 Machine，带来如下便利：

- 可以对应到服务器所在的物理硬件；
- 可以用来远程管理和监控；
- 用于加强 fail over 管理。

3.4 集群

集群，即大家通常所说的 Cluster，是在计算机企业级应用领域普遍被关注和采用的一种技术，有软件的，有硬件的，其核心思想是通过一组并行的实例，来实现实时的负载均衡和故障备份恢复。

3.4.1 什么是集群

WebLogic 集群就是一组相互协作的 WebLogic Server 实例。将 WebLogic Server 集群在一起，有两个主要优点：

- 可伸缩性：是为应用程序提供更多容量的能力，在这里，是通过添加更多的服务器，而无需进行重大的架构更改实现的；
- 高可用性：确保当(集群中的)某个服务器出现故障时，有其它服务器接管其工作，从而不会对客户端造成影响。

3.4.2 主要功能

WebLogic 集群的主要功能是：

- 应用程序故障转移：应用程序中正在执行任务的某个对象不可用时，另一个对象将接管并完成其工作；
- 站点故障转移：一个站点的所有服务器和应用程序出现故障时，这些服务和应当某个服务器出现故障时，其上的固定服务可迁移到集群中的另一个服务器；
- 负载均衡：多个服务器中均匀的分配任务和通信。

3.4.3 基本集群架构

基本集群架构将静态 HTTP、呈现逻辑、业务逻辑和对象组合到一个集群中：

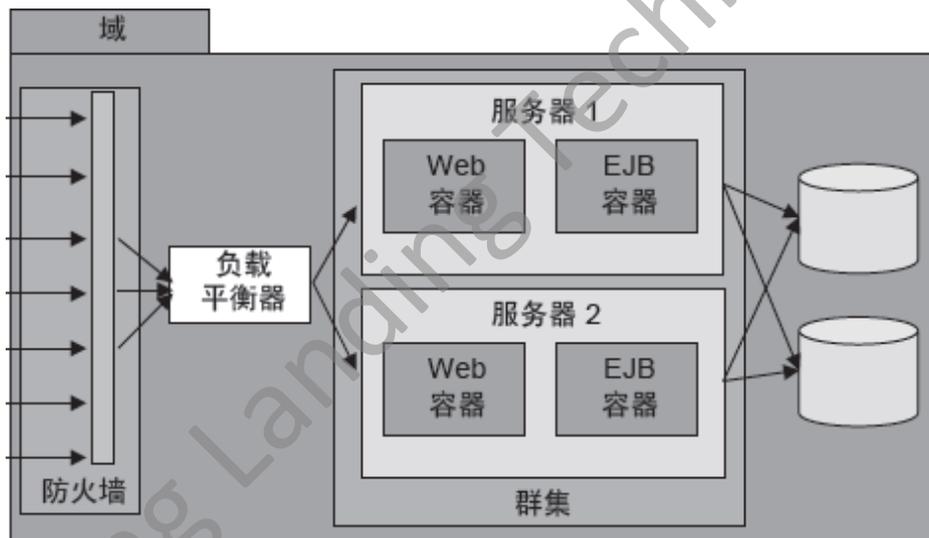


图 3-5

3.4.4 基本集群架构的优缺点

基本集群架构有以下优点：

- 易于管理；
- 灵活的负载均衡；
- 可靠的安全性。

基本集群架构的缺点：

- 无法对 EJB 方法调用进行负载均衡；
- 跨层平衡负载可能会出现不平衡。

3.4.5 多层集群架构

Web 层和提供服务的业务逻辑可分别放到两个集群中：

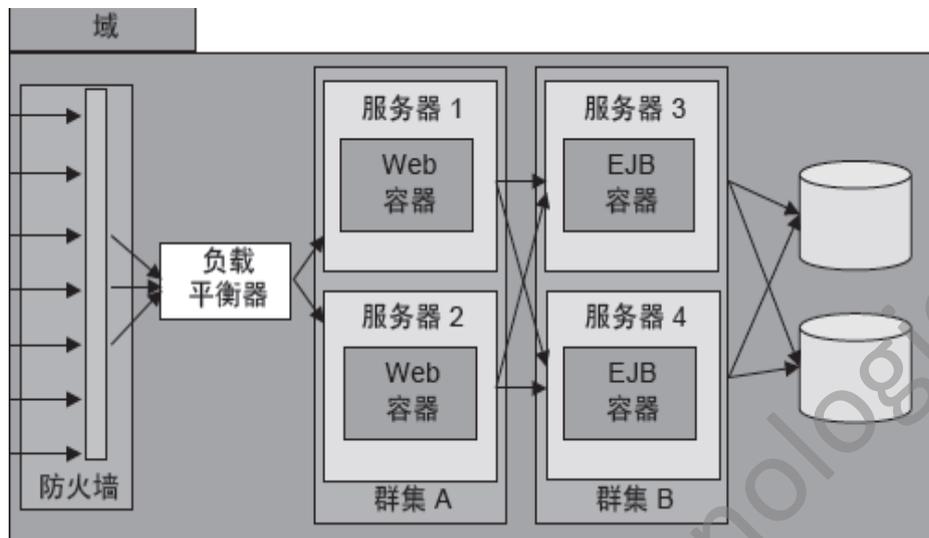


图 3-6

建议多层架构使用两个单独的 WebLogic Server 集群：一个提供静态 HTTP 内容和集群 Servlet，另一提供集群 EJB。

3.4.6 何时使用多层集群架构

建议对有以下要求的 Web 应用程序使用多层集群：

- 需要对集群 EJB 的方法调用进行负载均衡；
- 需要在提供 HTTP 内容和提供集群对象的服务器之间进行灵活的负载均衡；
- 需要更高的可用性（减少单点故障数）；
- 更灵活的安全性规划。

3.4.7 多层集群架构的优缺点

多层集群架构具有以下优点：

- 负载均衡得到改进
- 可对 EJB 方法进行负载均衡：通过分别在不同的集群上承载 Servlet 和 EJB，Servlet 中对 EJB 的方法调用可以在多个服务器间进行负载均衡；
- 更高的可用性：通过使用更多 WebLogic Server 实例，多层架构的故障点要比基本集群故障点少；
- 安全方案得到改进：将呈现层和对象层放到不同的集群上，可以使用只在 DMZ 中放置 Servlet/JSP 集群的防火墙策略。通过拒绝来自不可信客户端的直接访问，承载集群对象的服务器可以得到进一步的保护。

多层集群架构具有以下缺点：

- 呈现层频繁调用业务逻辑时会造成瓶颈；
- 许可证成本增加；
- 增加了防火墙配置的复杂性。

3.4.8 代理服务器

代理服务器用于为集群提供负载平衡和故障转移，此外，此类服务器还具有以下特点：

- 是客户端与集群交互的第一级；
- 使集群看上去就像是一个服务器

代理服务器可以基于软件也可以基于硬件：

基于软件的代理服务器可以是 WebLogic 提供的第三方 Web 服务器插件和 Proxy Servlet 或第三方应用程序。

基于硬件的代理服务器通常是物理负载平衡器(如 Local Director 或 F5 Networks Big IP)。

3.4.9 基本集群代理架构

除静态内容由非集群 HTTP 服务器承载以外，其余都与基本集群架构类似：

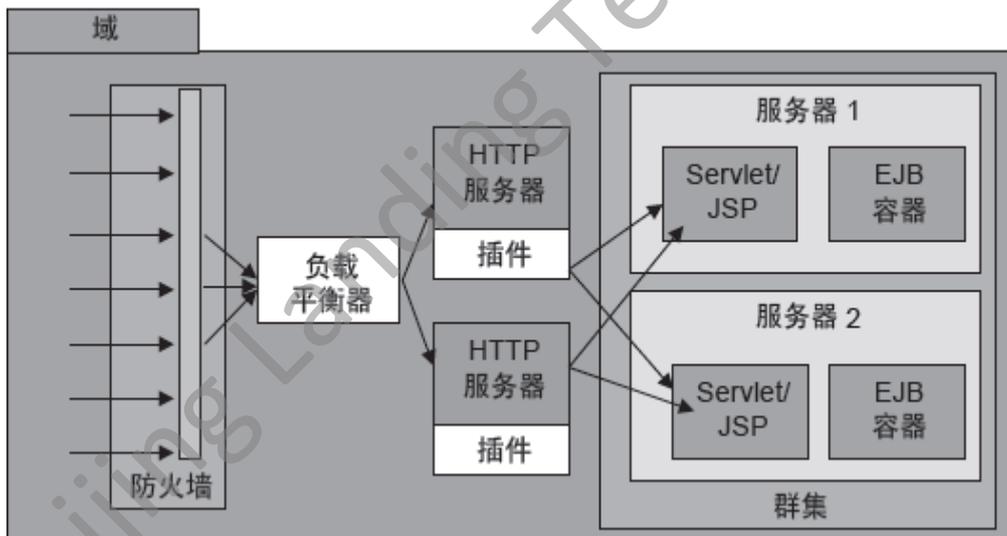


图 3-7

带有外部 web 服务器的基本集群架构包含两个物理硬件和软件层。

该架构利用硬件和软件层来专门执行提供应用程序 web 层的任务，该物理 web 层由一台或多台相同配置的计算机组成，这些计算机承载下面一种应用程序组合：

- 带有 HttpClusterServlet 的 WebLogic Server；
- 带有 WebLogic Server Apache 插件的 Apache；
- 带有 WebLogic Server NSAPI 代理插件的 Netscape Enterprise Server；

- 带有 WebLogic Server Microsoft IIS 插件的 Microsoft Internet Information Server。

3.4.10 多层集群代理架构

除静态内容由非集群 HTTP 服务器承载以外，其余都与多层集群架构类似：

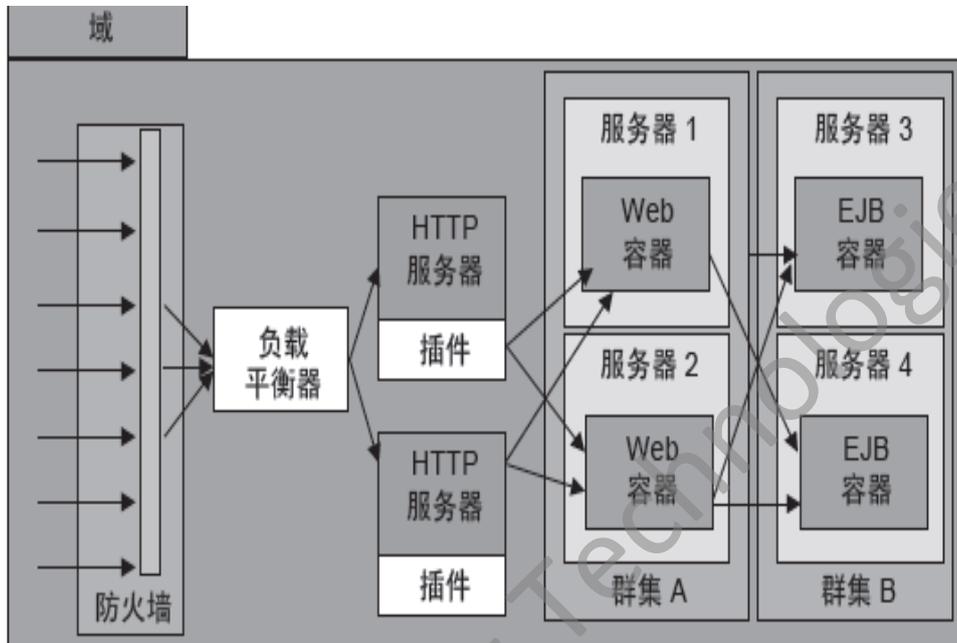


图 3-8

使用独立的 Web 服务器和代理插件有以下优势：

- 可以使用现有硬件
- 如果已有向客户端提供静态 HTTP 内容的 web 应用程序架构，则可以轻松地将现有 Web 服务器与一个或多个 WebLogic Server 集集群成，以提供动态 HTTP 和集群对象。
- 可以使用熟悉的防火墙策略：在 Web 应用程序前端使用 Web 服务器代理，可以使用熟悉的防火墙策略来定义 DMZ。通常，当不允许直接连接到架构中的其余 WebLogic Server 集群时，您还可以继续在 DMZ 中放置 Web 服务器，如上图 3-8 所示。

但是该方案也存在一些缺点：

- 管理成本增加；
- 负载均衡方案受到限制。

3.4.11 生产模式与开发模式

- 开发模式，该模式启用自动部署
- 生产模式，该模式关闭自动部署

WebLogic Server 支持三种部署方法：自动部署，控制台部署，命令行部署。后两者顾名思义，都有专门的登录界面和命令行工具来操作，后续章节会陆续讲到。这里重点对于自动部署做一个解释。

自动部署是一个标志，它使得配置或定位工作不需要任何管理员或开发人员的操作。当其处于启用状态时，管理服务器将监视指定的目录以确定是否有新建的或更新的应用程序，WebLogic Server 将自动部署该目录中的所有文件或目录。

WebLogic Server 配置为定期监视该目录以确定新建的或更新的文件。

默认情况下，自动部署目录为域的“\autodeploy”子目录。在 autodeploy 目录中找到的任何新文件都将被自动部署。applications 目录中编辑或修改过的归档文件将会重新部署。

如果应用程序处于展开格式，则创建一个名为 REDEPLOY 的空文件，将其放置在 WEB-INF 目录中，然后当应用程序需要重新部署时，请修改该文件以更新其时间戳。服务器将定期检查对 REDPLOY 文件时间戳的更新。

最后，当前存在于 applications 目录中，但后来被删除的所有文件都将自动从应用服务器中取消部署。

Beijing Landing Technologies