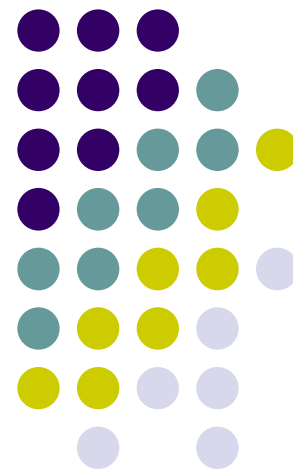
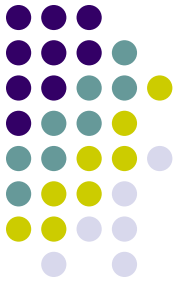


第一章 Web应用概述





主要内容

- Web应用的出现
- Internet的几个关键属性
- 可扩展和高性能的Web应用
- Web应用的体系结构
- Web应用开发技术介绍



1、Web应用的出现

现在，Internet的出现，已经使World Wide Web(Web)成为了最重要的使用最广泛的信息共享和交互的平台，每天都有数以千万计的人们通过Web来获取到他想要的信息，也同时为其他人提供信息。Web应用使得提供者和使用者都能够极大的享受到Web应用提供的方便和快捷。

同时，Web应用要求合理、稳定快速的运行速度，而且对于不断增长的用户和请求需求还应该有较强的而渐增的容许性，这都是Web应用性能的重要方面。



桌面应用程序示例



用户



Windows 媒体播放器

媒体播放器播放 MP3 文件

用户选择一个 MP3 文件



桌面应用程序示例



用户

用户发送一个请求

播放器发送一个响应

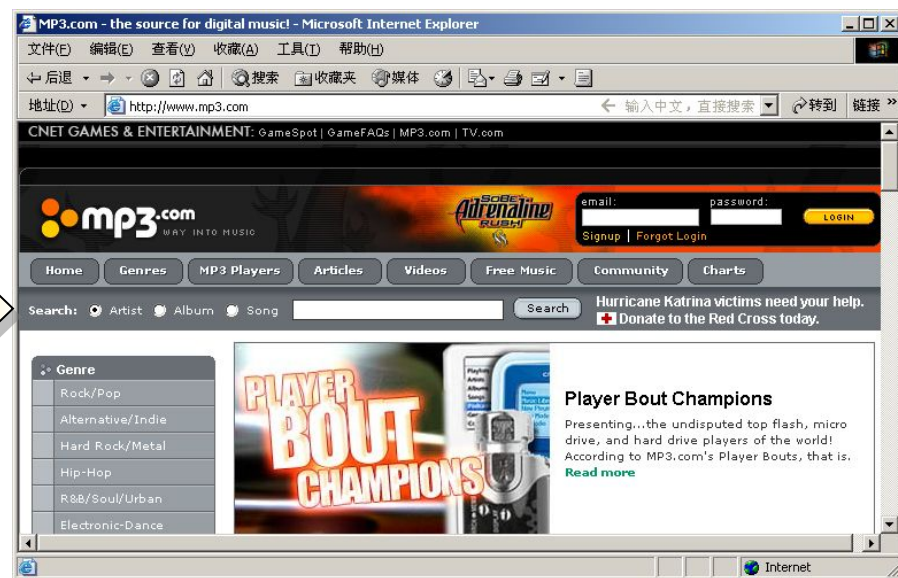


Windows 媒体播放器

媒体播放器是桌面应用程序的一个示例

响应 = 媒体播放器播放选定的文件

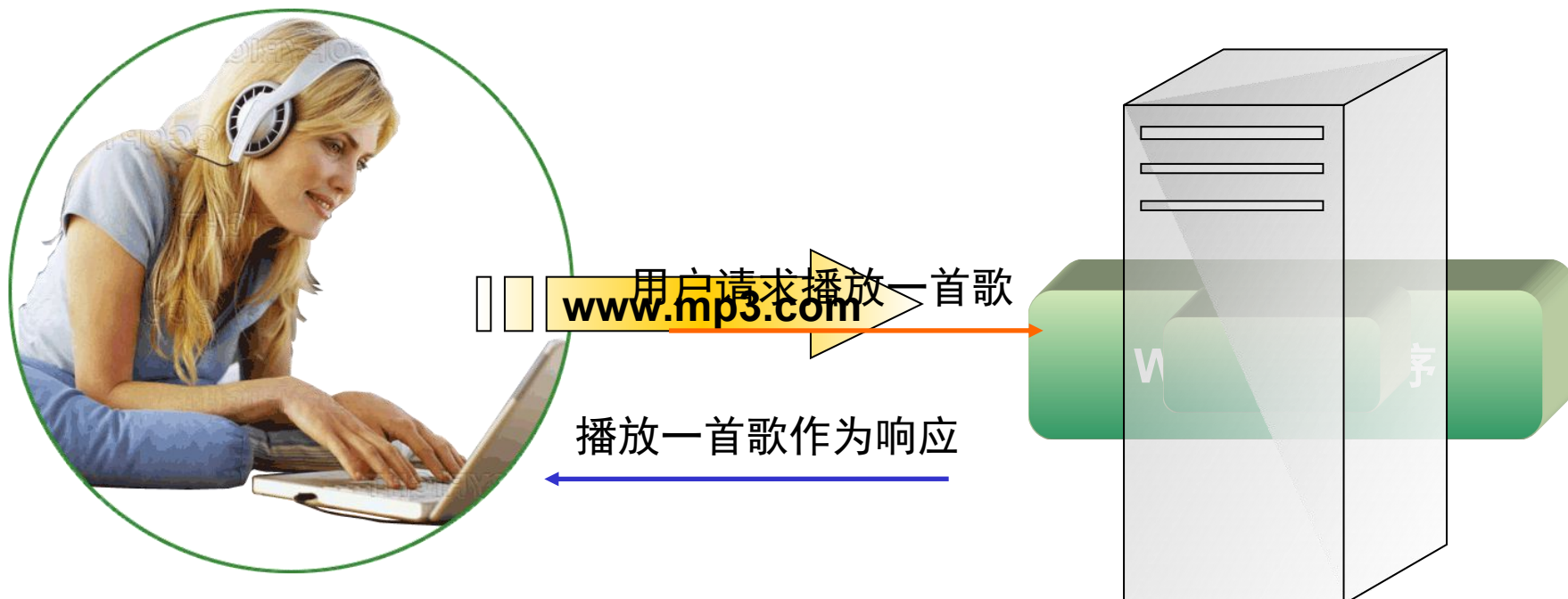
Web应用程序示例



位于 Web 服务器上的网站



Web应用程序示例



Web 应用程序的优点：

- 访问 Web 应用程序更容易
- 维护和部署成本低

Web 应用程序在 Web 服务器上运行



Web应用定义：

是一种通过Internet技术加以连接的客户/服务器软件，可以传输其处理的数据。通过“Internet技术”，指的是在信息的使用者和提供者之间，组成相应网络基础架构的硬件和软件的集合。Web应用可以通过专门的客户端软件来访问，也可能利用一个或多个Web页面访问，这些页面基于某种特定的用途。



注意，我们这里讲的是Web应用，它不只是Web网站，它是包括了服务器端以及网络和服务端的综合。

而web网站则是由大量的页面组成，每个页面通常由一个文件或多个文件组成。组成一个网站的大量文件相互之间通过特定的方式进行链接，并存在一个系统来管理这些文件，管理这些文件的系统通常称为应用服务器。



web应用运行原理：

上网的一般过程如下：

- (1) 打开浏览器。**
- (2) 输入某个网址。**
- (3) 经过一段时间的等待。**
- (4) 浏览器显示要访问的信息。**

然后可以在网页继续进行其他操作，可能的操作如下：

- ◆ 在网页上点击超级链接访问我们希望访问的内容，等待浏览器中内容的再次更新。**
- ◆ 在网页上输入一些信息，然后点击按钮，等待浏览器中内容的再次更新。**



不管是在地址栏输入地址，还是点击超级链接或者点击按钮，都需要等待浏览器中内容的更新。等待浏览器内容更新的过程，实际上是浏览器访问Web应用的过程。这个过程如下：

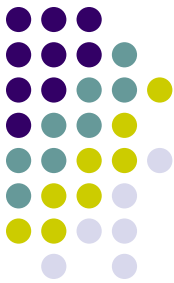
- **浏览器根据我们输入的地址找到相应的服务器，不同的网站对应不同的服务器。这个服务器可以接收浏览器发送的请求，通常称为Web服务器。**
- **Web服务器把这个请求交给相应的文件管理器，这个文件管理器也是一个服务器，通常称为应用服务器。**
- **应用服务器接收到请求之后，查找相应的文件，加载并执行这个文件。执行的结果通常是HTML文档。**
- **应用服务器执行完相应的文件之后，把执行的结果返回给Web服务器，Web服务器再把这个结果返回给浏览器。**
- **浏览器解析HTML文档，然后把解析后的网页显示给用户。**



Web应用示例

- 例如：当用户从浏览器的地址栏输入http://www.baidu.com时，便显示出百度搜索页面，当在搜索页面中输入“Java Web应用”并单击“百度一下”时，就会显示搜索到的资源列表。





问题：

- (1) 百度的首页和我们要搜索的资源是存放在哪里的呢？**
- (2) 计算机又是如何将其显示在浏览器中的呢？**

✧解答：

(1) 百度的首页和我们要搜索的资源是存放在百度网的Web服务器上面。其部分内容或全部内容是不确定的。只有当用户请求 Web 服务器中的某个页面时，才确定该页的最终内容。因为页面内容基于用户的操作，随请求的不同而变化，所以这种页面称为动态页面，反之则为静态页面。

(2) 当用户在地址栏中输入百度的首页地址 (URL , 即 “统一资源定位符”) 时，浏览器会向百度网的服务器发送HTTP请求，这个请求使用HTTP协议，其中包括请求的主机名、HTTP版本号等信息。服务器在收到请求信息后，将根据用户请求定位资源，通过网络发送到客户端的浏览器。



Web应用程序工作原理

1. 静态页工作原理

静态网页也称为普通网页，是相对于动态网页而言的，并不是指网页中的元素都是静止不动的，而是指在通过浏览器进行浏览时，Web服务器中不再发生动态改变（没有表单处理程序或其他应用程序的执行），因此网页不是即时生成的。

浏览器“阅读”静态网页的执行过程较为简单，如图1-4所示。

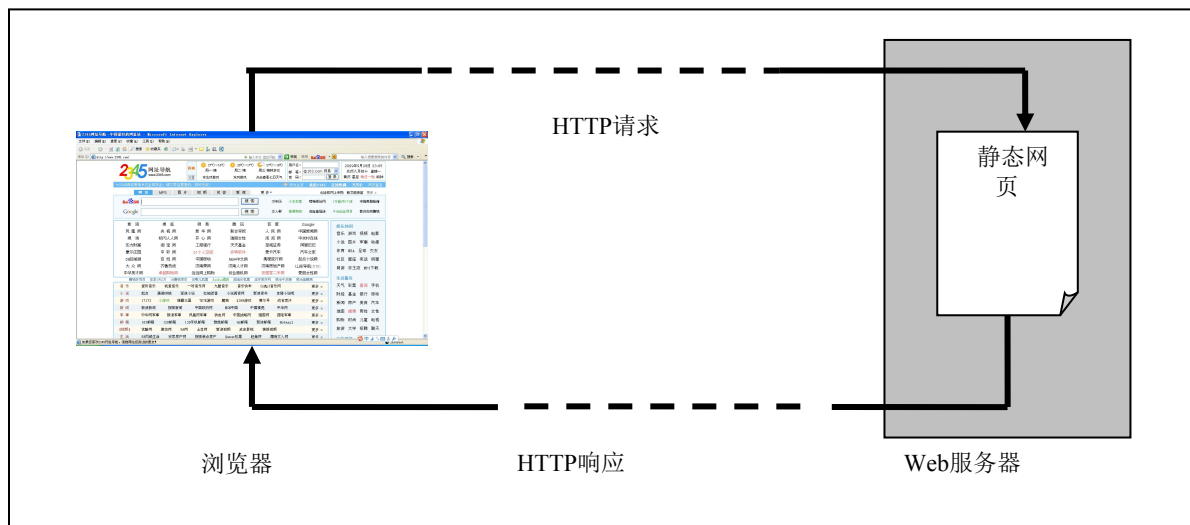
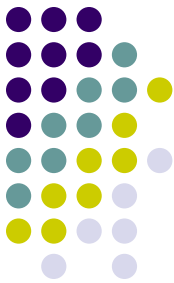


图1-4 静态网页工作原理



工作原理：

- 当 Web 站点接收到浏览器端发送来的一个 HTTP 请求后，服务器检查在服务器上是否存在客户端所请求的文件，如果该文件存在，则用 HTTP 响应（HTTP Response）的形式把客户端所请求的文件送回给客户端的浏览器。当客户端浏览器接收到了 HTTP 响应后，开始解读HTML标签，然后将其转换，将结果显示出来。



2. 动态页工作原理

动态网页是指网页中除了静态网页中的元素外，还包括一些应用程序，这些应用程序使浏览器与Web服务器之间发生交互，而且应用程序的执行有时需要应用程序服务器支持才能够完成。

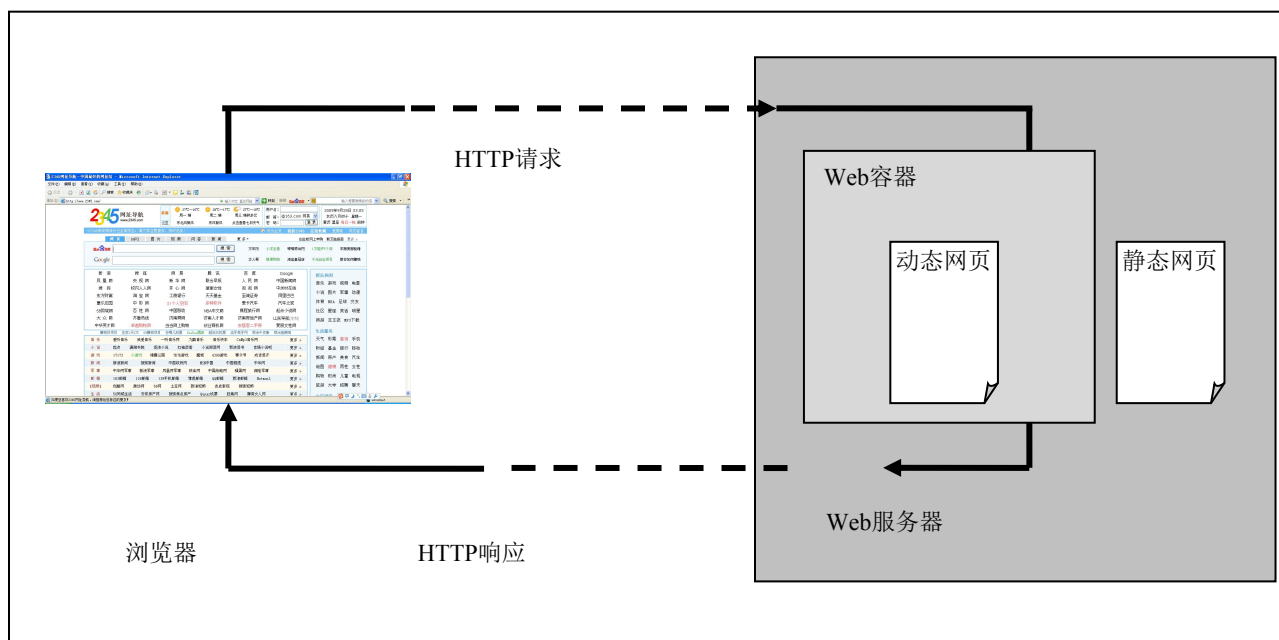
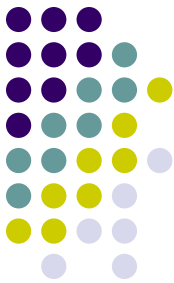


图1-5 动态网页工作原理



工作原理：

当客户端浏览器向网络中的Web服务器发出请求时，浏览器会将用户输入的信息一起发送到Web服务器，Web服务器接收请求信号后，将该网页发送至应用程序服务器，应用程序服务器检查该网页，执行其中的应用程序，在执行应用程序的过程中，可能会查询数据库，查询完数据库，应用程序服务器会将查询到的数据插入网页中，此时动态网页变为静态网页，Web服务器将完成的静态网页传给浏览器，浏览器接到Web服务器送来的信号后开始解读HTML标签并将其转换，有时还执行脚本程序，然后将结果显示出来。



2、Internet的几个关键属性

- 广泛性

现在越来越多的人正在使用Internet这种新的媒体。

- 交互性

与其它媒体不同，Internet上信息的发布并不在提供者的控制范围之内，实际上，使用者通过查询来请求信息，而这需要客户与服务器之间完成一系列的交互

- 动态性

Web应用所表示的信息取决于与用户或会话相关的数据。随着用户的操作，大量统计和历史会话属性都会使得应用的响应情况有所不同。



- **连续的开放性**

很多Web应用是需要全天候的不间断运行。

- **集成性**

使用者在请求信息时，提供者会查找本地和远程的多个资源，这就需要对其响应中的多条信息加以集成。

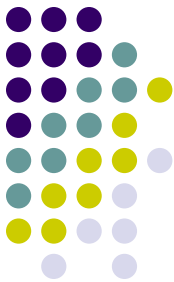
- **缺乏完全控制性**

对于信息的提供者，它是无法保证其信息发送的服务质量问题。



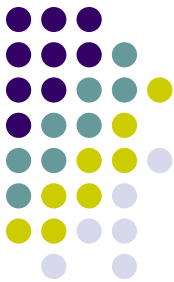
3、可扩展和高性能的Web应用

由于Web应用是架构在Internet上的，所以Internet这个媒体的基本属性，对于构建的Web应用的影响也是至关重要的。我们只有更深入的理解Internet媒体的基本属性，我们才能够更好的构建具有良好扩展性和高性能并且正常合理运行的Web应用。



■ 性能

- ↗ 当您访问一个Web网站的时候，您点击一个链接，它的响应时间如何呢？我们会发现有的网站速度快，有的网站速度慢，从单个用户的角度看，性能可以仅仅描述为应用的执行速度。
- ↗ 虽然各种因素都可以衡量Web应用的性能，但是应该强调只有端到端的性能指标才能真正衡量其性能，端到端的性能意味着需要使面向用户的操作更为迅速。我们可以通过各种方式来改善它



■ 可扩展性

- ✧ 可扩展性的描述为：随着用户和请求数量的不断增长，Web应用仍可用，并保持稳定的反应速度。也可以定义为一个针对持续增长的用户负载而进行的弹性度量。
- ✧ 一般的，我们通过3种技术来提高可扩展性：第一增加资源（更大型的机器，更多存贮空间以及内存等），第二改进软件，第三增加资源同时改进软件，从长远看，第三种是最终的办法



3、Web应用的体系结构

- Web应用术语
- Web应用需求
- Web应用的体系结构



Web应用术语

■ Web事务

- ✧ 是一个对应于某个逻辑应用操作的请求-应答会话。也就是客户所发出的请求会启动服务器端的应用逻辑，而它最终将返回给客户一个应答，而且应用逻辑的操作可能是一个很复杂的操作，可能包含了数据库级的事务。



■ Web请求

- ✦ 那么一般的Web请求都是无需应用逻辑的、孤立的请求，这个要和Web事务加以区别，一般的Web请求都是静态的访问某个Web网页，而事务则是动态的，涉及到更多的服务器端资源。



■ 会话

- ✧ 所谓会话就是指客户在一段时间内对应用的使用，会话由一个或多个在前面定义的事务所组成。就像事务对应为一系列的应用操作，会话也对应为一系列的事务。
- ✧ 但是与事务不同的是，会话中的事务之间可以不存在逻辑关联。一般而言，会话是从用户登陆到应用或者打开与应用的连接为开始，到用户显式地退出或者会话生命周期到时结束。



Web应用需求

每个应用都要有相应需求来指定它应支持的功能。那么Web应用的需求为：

1、业务逻辑

业务逻辑需求是任何一个Web应用中最为重要的一部分，这些需求指定了那些业务过程需要由应用（以某种方式）进行处理。首先，业务逻辑是一组提供服务所需的操作。最基本的情况下，此逻辑仅仅是一组功能集合，每个功能均由一系列有明确用途的步骤组成。例如转账业务



其次，业务逻辑应该是动态的，可以定制的。我们可以不必重建整个应用即可以对业务逻辑进行替换或修改。实际上，对于许多应用来说，动态业务逻辑也确实是一个需求，例如新闻过滤要有相应的过滤条件，员工管理则需要必要的合格规则，带有可定制的业务逻辑意味着逻辑本身（或者参数）也需要存储到数据库中



2、数据管理

如果我们把业务逻辑比喻成为Web应用的管道，那么与应用有关的数据则是管道中流淌的水。所以业务逻辑和应用数据是相辅相成的。

数据管理必须处理可靠、公平、安全而有效的数据，那么数据库技术就是用于满足这些需求的最主要的机制。



3、接口（表示）

用户可以通过Web浏览器、普通电话、个人数字助理（PDA）等设备来访问应用。一般来讲所有应用功能都可以通过Web浏览器访问，当然有些应用的功能也可以利用其他的接口来访问。当然我们一般的情况下还是以浏览器作为Web借口的重点。



4、Web应用的体系结构

抽象的Web应用体系结构

客户 <--> 接口（表示） <--> 业务逻辑 <--> 数据管理

传统的胖客户或者瘦客户要么是把业务逻辑放到了客户端这边要么就是把业务逻辑和数据管理系统放到了服务器端，还有一些混合设计的把业务逻辑的一部分放在客户端，另外一部分放在服务器端。



Web应用的多层体系结构

1、客户：

一为人类客户，它通常有一台带有操作系统的主机并能够访问网络，一般情况它使用Web浏览器来访问Web应用，也可能使用其它的访问接口，浏览器使用HTTP协议通信

二为自动的基于软件的客户，可能运行在一个功能更为强大的主机上，可能使用HTTP协议或者使用低级或者专用的协议来通信。



2、网络：

客户和服务端之间的网络通常被称为Internet，而Internet是由分布在世界各地的许多主机和子网组成的。它使用TCP/IP协议作为通信协议，IP是网际协议，而TCP是传输控制协议，它们一起构成一个Internet上的面向连接的协议，提供可靠的信息传输服务保证。基于Web应用的环境它又可以被我们逻辑的划分为三部分：



(1) 客户端网络元素

靠近客户端的网络部分，绝大多数的客户都是通过一个Internet服务提供商(ISP即Internet Service Provider)来访问Internet。

利用浏览器缓存或者其他缓存将远程Web页面数据进行本地存储，我们称为Web缓存代理缓存，它可以是硬件或者软件，它对频繁请求的Web页面进行缓存，ISP就不需要重复地为客户获取相同的页面。



(2) 服务器端网络元素

除了和客户端一样，需要有一个ISP，另外还有一个重要的部分，就是本地的负载均衡器，负责把可能到来的连接分发给相应的服务器资源。

另外，服务器端可以建立反向代理缓存，对于频繁访问对象，可以提供对其快速访问来减少对服务器的请求。



(3) 介于客户端和服务端端的网络元素

网络中第三个部分就是介于客户端和服务器的大量的路由器和交换机。一般来讲中间的网络通信，是通过Internet的主干网络服务提供商(Network Service Provider)来进行转发的。



3、服务器

服务器端应用体系结构是最复杂的，但是我们可以一般的把它划分为三个层次，我们按照请求经过部分划分为：请求处理器、应用服务器和数据库。

请求处理器的一个例子就是Web服务器，一方面它处理客户端发来的Web请求，另一方面作为请求路由器它可以确定请求的类型，并把它分派给合适的处理程序。



4、基于层的划分

在客户请求得到处理之前所经过的区域，我们把它称为层。
每一层都与一个或者多个逻辑相关联：表示、业务或者数据访问等。多层划分的优点在于：

- (1)提高模块化和组件的重用性**
- (2)更好的分布式处理和任务并行化**
- (3)更有效的服务复制**



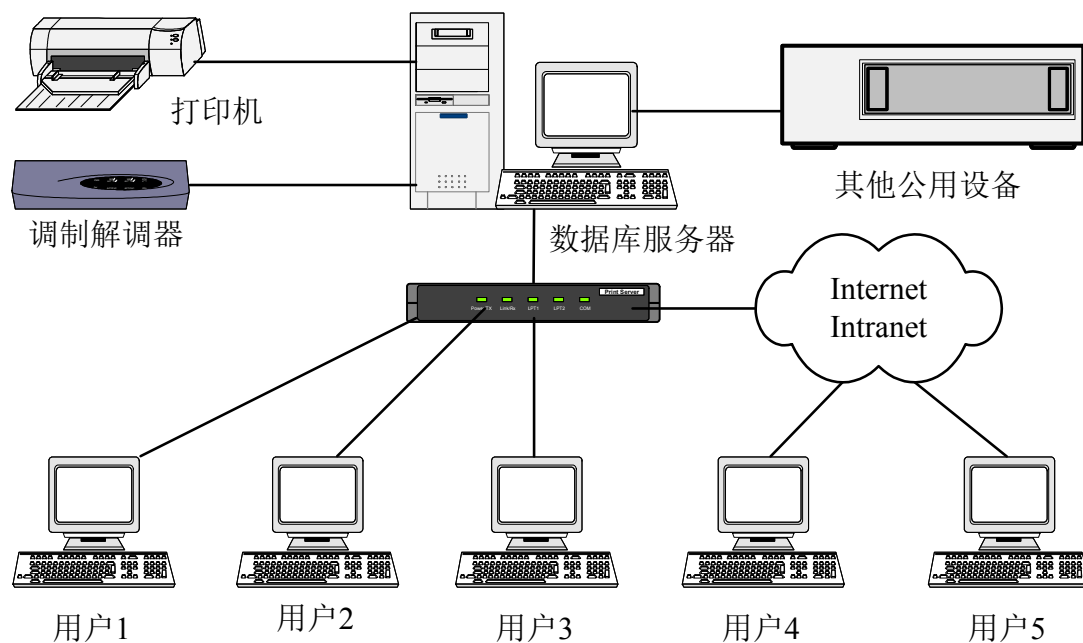
5、Web应用开发技术介绍

- 应用开发模式
- B/S模式常用开发技术体系
- 企业应用架构
- Java企业开发技术体系
- 常用Web应用开发环境介绍

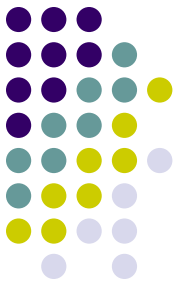
应用开发模式



■ C/S和B/S开发模式

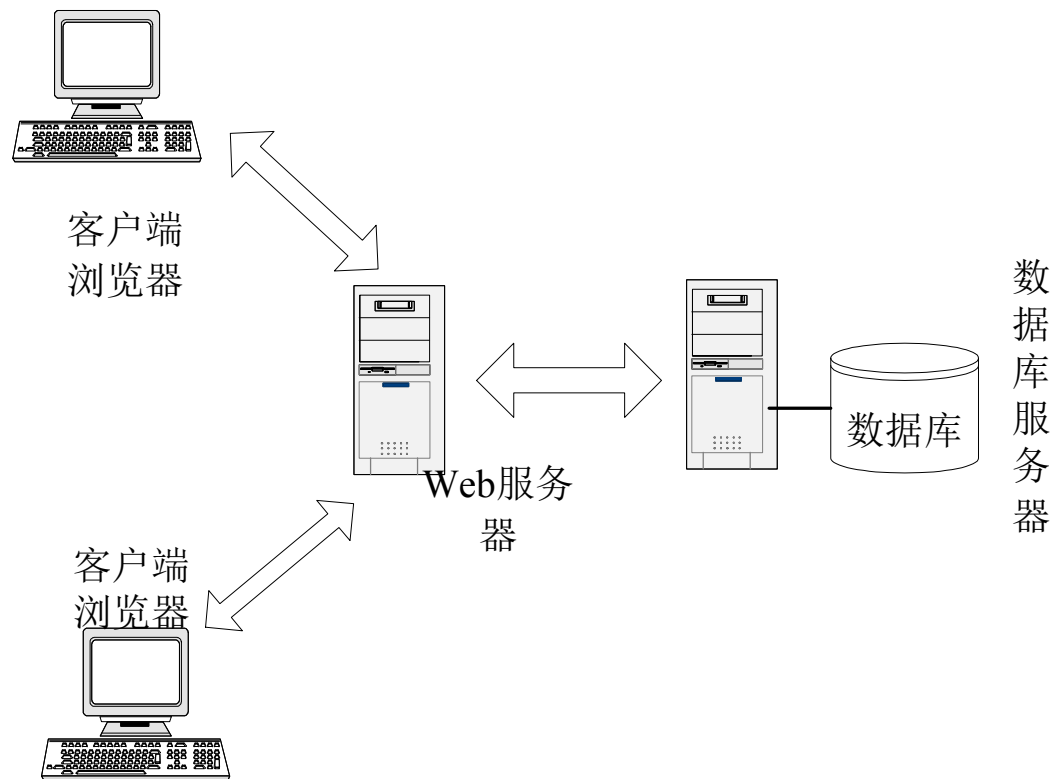


C/S模式体系结构

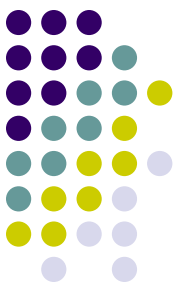


■ C/S模式特点：

- ✧ 采用这种模式编写的软件分为客户端和服务端两部分，需要分别在客户机和服务器上安装。
- ✧ 这种模式在用户数据录入方面很有优势，也降低了系统的通信开销，但是也有一定的不足之处。例如，当客户端软件需要升级的时候，所有的客户端软件都必须进行升级安装或者重新安装，这就给软件的使用者或系统管理员带来很大的不便。同时，由于不同的客户可能使用不同版本的客户端，在设计和升级服务器端软件的时候不得不考虑软件版本的兼容性，这对程序员来说是件很痛苦的事。

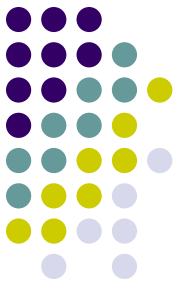


B/S模式体系结构



■ B/S模式特点:

- ✧ 这种模式可以说是C/S的变体或者说是改进，这种模式围绕着Web服务器来进行，Web服务器是安装了Web服务软件的计算机，它能接受客户端发出的HTTP Web请求。如在浏览器地址栏中输入 `http://www.baidu.com` 即会向Web服务器发出请求，Web服务器处理请求后，产生HTML脚本发回客户端，而客户端就不需要像C/S那样专门编写一个程序，而是变为IE浏览器。接收服务器返回的HTML，然后将该HTML显示出来，提供人机交互界面。
- ✧ B/S开发中的重点就是编写Web程序，目前大部分流行的Web程序，都采用HTML加服务器端嵌入式脚本的方式组织。



■ B/S模式的优点在于：

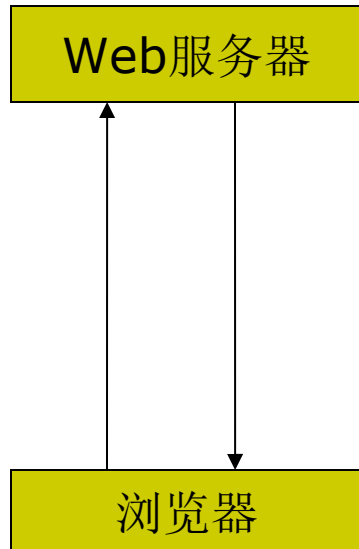
- ✧ 客户端基于统一的Web浏览器，减少了投资，解决了系统维护升级的问题；
- ✧ 灵活性和可扩展性：系统可根据规模的不断扩大，在不影响用户日常工作的前提下，对Web服务器和数据库服务器等设备进行扩展；
- ✧ 简易性：操作直观、简单，培训方便，对使用人员的计算机操作水平要求不高；
- ✧ 实施成本低：充分利用现有的办公网络，避免了网络重复建设。
- ✧ 目前流行着多种B/S开发语言，如ASP，ASP.NET，PHP，JSP等。本书将重点介绍JSP。



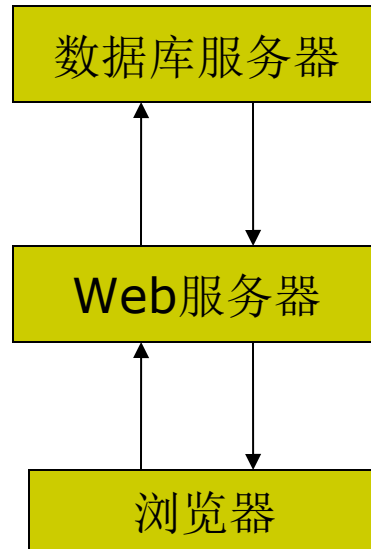
应用开发模式

随着Internet的迅速发展，越来越多的企业正在利用Internet技术构建自己的企业信息系统，并使用浏览器/Web服务器(B/S)开发模式构建其web应用。

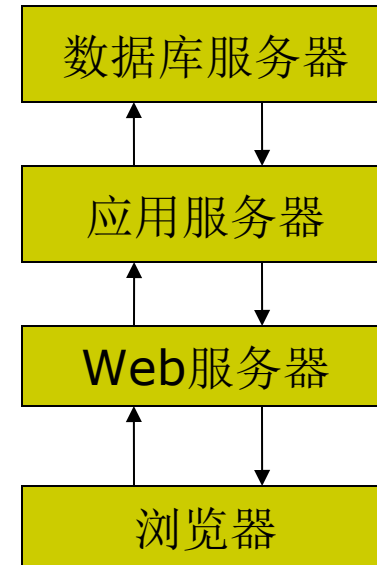
这里，Web服务器是Internet上基于超文本传输协议(http)的服务器，而浏览器又称为Web浏览器，使客户端用来访问Web服务器的通用软件。一般B/S有三种结构。



(a) 两层B/S结构



(b) 三层C/S结构



(c) 四层C/S结构

B/S计算模式



B/S模式常用开发技术体系

■ 目前主要采用两种服务器端语言：

- ◆ JSP (Java Server Pages)
- ◆ ASP (Active Server Pages)

■ 两种语言构成两大开发体系：

- ◆ JSP+Oracle 体系
- ◆ ASP+SQL Server 体系



■ B/S结构的编程语言

- ◆ 浏览器端编程语言
- ◆ 服务器端编程语言

■ 浏览器端

- ◆ HTML (Hypertext Markup Language , 超文本标记语言)
- ◆ CSS (Cascading Style Sheets , 层叠样式表单)
- ◆ JavaScript语言
- ◆ VBScript语言



■ 服务器端

■ 目前主要是3P技术

✧ ASP、JSP和PHP

- ◆ **ASP**是微软公司推出的，在这三种语言中是用得最为广泛的一种。
- ◆ **JSP**是**SUN**公司推出的是**J2EE (Java 2 Enterprise Edition , Java2 企业版)**十三种核心技术中最重要的一种
- ◆ **PHP** 在 **1999** 年的下半年和 **2000** 年用得非常广泛，因为 **Linux+PHP+MySQL**（一种小型数据库管理系统）构成全免费的而且非常稳定的应用平台



企业应用架构

在构建企业级应用的时候，通常需要大量的代码，而且这些代码一般分布在不同的计算机上，划分代码运行在不同计算机上的理论就是多层设计理论。

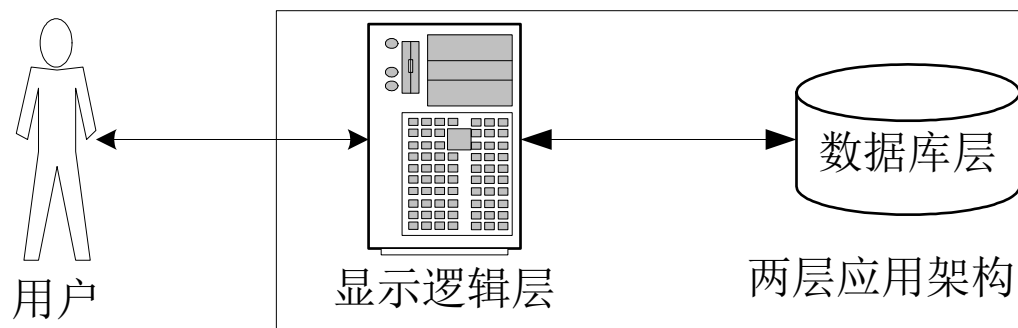
企业级应用系统通常分成：

- ✧ **两层**
- ✧ **三层**
- ✧ **N层架构**



■ 两层架构

传统的两层应用包括用户接口和后台程序，后台程序通常是一个数据库，用户接口直接同数据库进行对话。实现上，通常使用JSP、ASP或者VB等技术编写这类软件，结构如图：

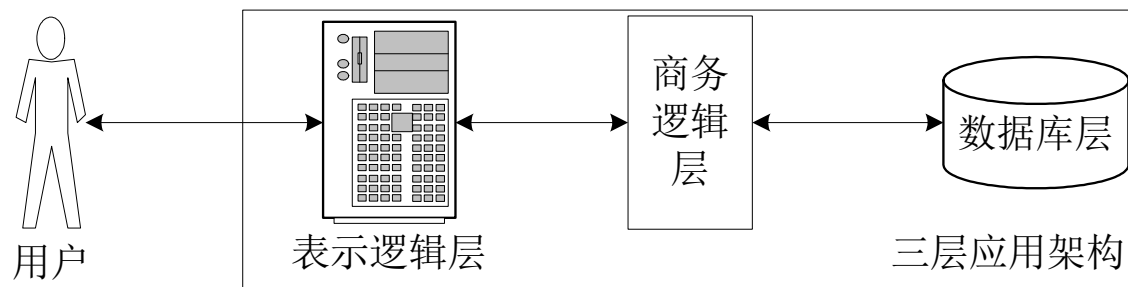


两层应用架构显示逻辑层一般由HTML、JSP、ASP实现，通过JSP和ASP直接和数据库相连



■ 三层架构

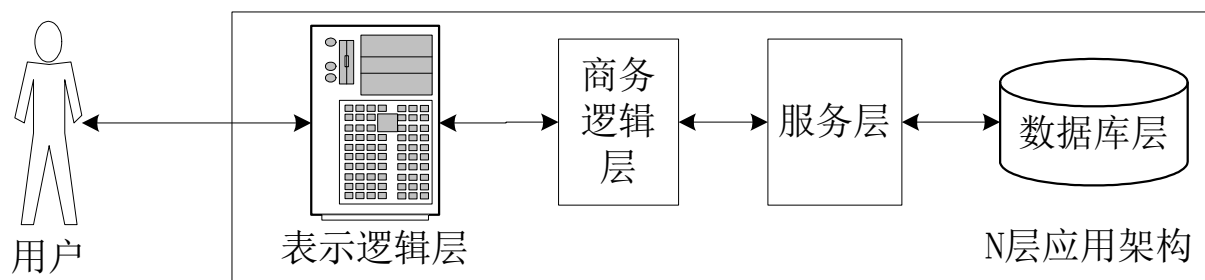
在两层应用中，应用程序直接同数据库进行对话。三层结构在用户接口代码和数据库中间加入了一个附加的逻辑层，通常这个层叫做“商务逻辑层”，如图：





■ N层架构

如果某个应用超过三个独立的代码层，那么这个应用叫做N应用，就不再叫四层或者五层等名称，而是统称为N层。如图：





■ 开发架构的比较：

（1）两层架构的优点是开发过程比较简单，利用服务器端的程序直接访问数据库，部署起来比较方便。缺点是程序代码维护起来比较困难，程序执行的效率比较低，用户容量比较少。



（2）三层架构基本解决了两层架构的缺点，将显示部分和逻辑流程控制分开，利用服务器应用程序实现显示部分，利用商务逻辑层实现程序的流程控制，分层使维护变的方便一些，而且执行效率也会有所提高，但是相对部署起来就比较困难一些。

（3）根据实际的需要，会进一步细化每一层，或者添加一些层，就形成了N层架构，和三层架构一样，组件化的设计使维护相对容易，但是部署相对困难。



Java企业开发技术体系

- J2EE , Java2平台企业版 (Java 2 Platform Enterprise Edition) , 是 [Sun](#) 公司为企业级应用推出的标准平台。Java平台共分为三个主要版本Java EE、[Java SE](#)和[Java ME](#)。
- [Sun](#) 公司在 [1998年](#)发表 [JDK1.2](#) 版本的时候 , 使用了新名称 Java 2 Platform , 即 “Java2 平台” , 修改后的 [JDK](#) 称为 Java 2 Platform Software Developing Kit , 即 [J2SDK](#) 。并分为标准版 (Standard Edition , [J2SE](#)) , 企业版 (Enterprise Edition , J2EE) , 微型版 (MicroEdition , [J2ME](#)) 。J2EE便由此诞生。
- 2005年6月 , JavaOne大会召开 , SUN公司公开Java SE 6。此时 , Java的各种版本已经更名以取消其中的数字 “2” : J2EE更名为Java EE, [J2SE](#)更名为[Java SE](#) , [J2ME](#)更名为[Java ME](#)。
- 随着[Java](#)技术的发展 , J2EE平台得到了迅速的发展 , 成为[Java](#)语言中最活跃的体系之一。现如今 , J2EE不仅仅是指一种标准平台 ([Platform](#)) , 它更多的表达着一种[软件架构](#)和设计思想。

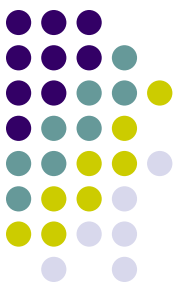


■ **JavaEE是一系列技术标准所组成的平台，包括：**

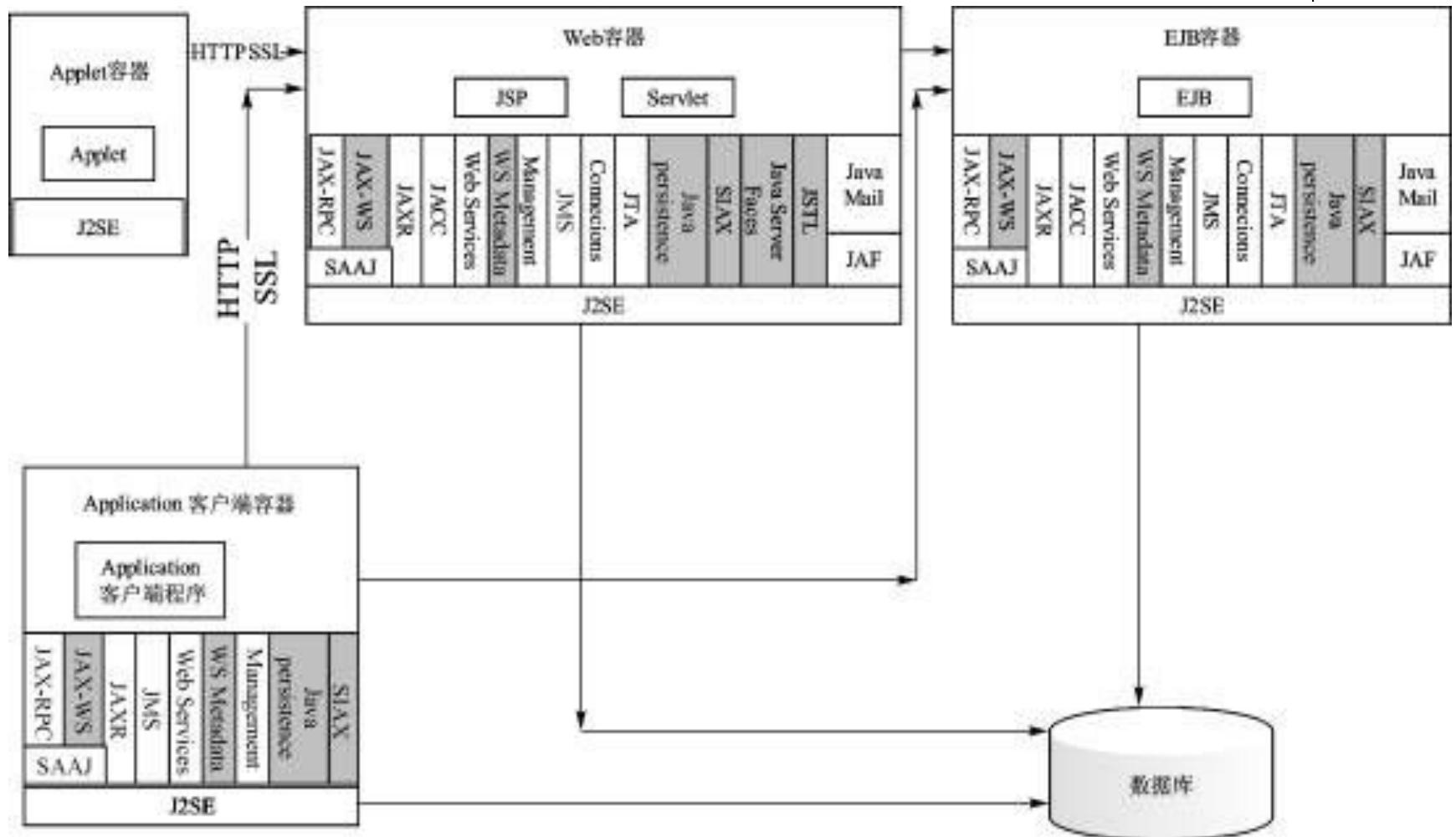
- ✧ **Applet** - Java Applet
- ✧ **EJB** - 企业级**JavaBean** (Enterprise Java Beans)
- ✧ **JAAS** - Java Authentication and Authorization Service
- ✧ **JACC** - J2EE Authorization Contract for Containers
- ✧ **JAF** - Java Beans Activation Framework
- ✧ **JAX-RPC** - Java API for XML-Based Remote Procedure Calls
- ✧ **JAX-WS** - Java API for XML Web Services
- ✧ **JAXM** - Java API for XML Messaging
- ✧ **JAXP** - Java XML解析API (Java API for XML Processing)
- ✧ **JAXR** - Java API for XML Registries



- ✧ [JCA](#) - J2EE连接器架构 (J2EE Connector Architecture)
- ✧ [JDBC](#) - Java数据库联接 (Java Database Connectivity)
- ✧ [JMS](#) - Java消息服务 (Java Message Service)
- ✧ [JMX](#) - Java Management
- ✧ [JNDI](#) - Java名字与目录接口 (Java Naming and Directory Interface)
- ✧ [JSF](#) - Java Server Faces
- ✧ [JSP](#) - Java服务器页面 (Java Server Pages)
- ✧ [JSTL](#) - Java服务器页面标准标签库 (Java Server Pages Standard Tag Library)
- ✧ [JTA](#) - Java事务API (Java Transaction API)
- ✧ [JavaMail](#)
- ✧ [Servlet](#) - Java Servlet [API](#)
- ✧ [StAX](#) - Streaming APIs for XML Parsers
- ✧ [WS](#) - Web Services



JavaEE总体框架图





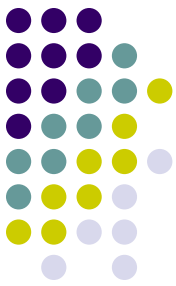
■ JavaEE N层架构的特点

JavaEE带来的好处是双向的，对于开发应用的公司和使用它的客户，优点主要在于4个方面。

1、简化了体系和开发。相对于传统的应用开发，JavaEE提供了简单的、基于组件的开发模型。这个模型提供了WORA（Write Once, Run Anywhere）的可移植性，只要符合JavaEE规范的服务器，就可以运行应用。同时，基于组件的开发模型也大大提高了生产力。

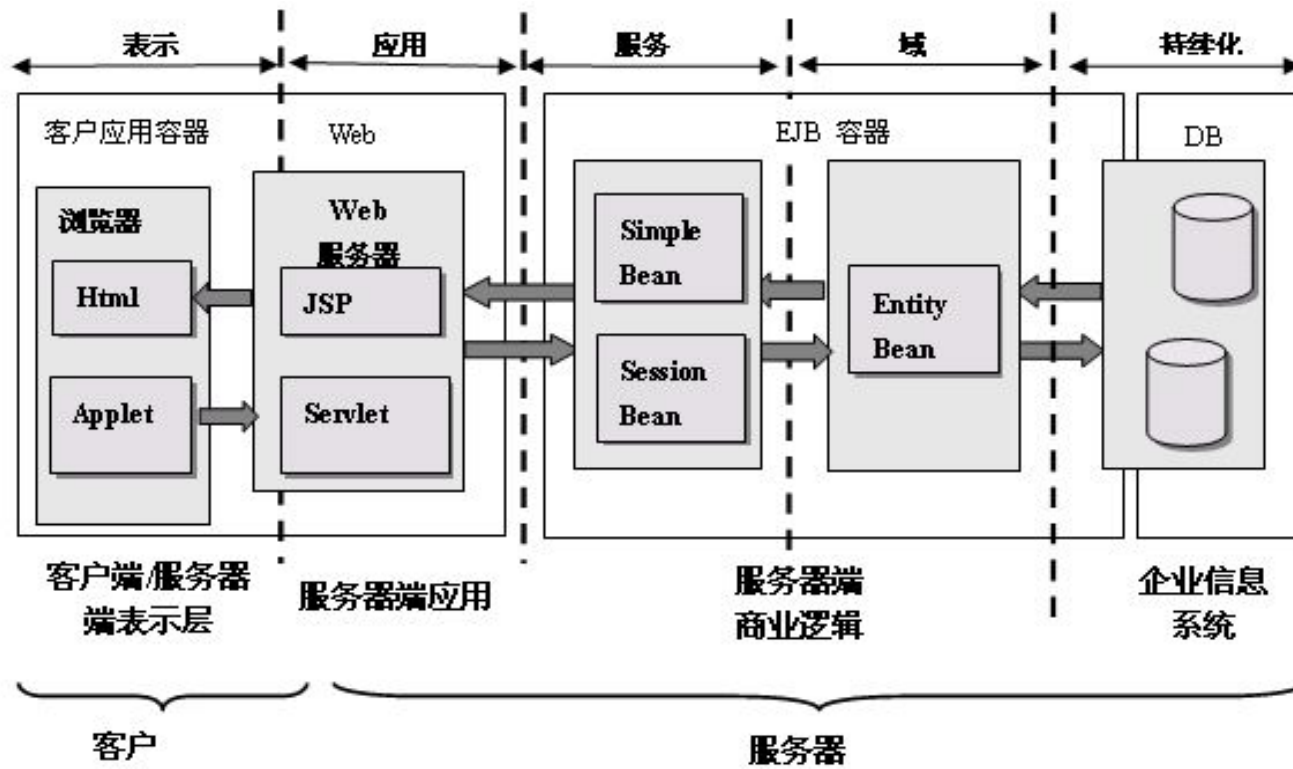


- 2、集成现有的企业信息系统。JavaEE平台，与JavaSE平台一样，提供访问现有的企业信息系统的一些工业标准API：
- 3、提供可扩展性。当前JavaEE容器向组件提供事务支持、数据库连接、生命周期管理和影响性能的有关特性，从而提供可扩展性（scalability）。
- 4、安全机制。JavaEE支持丰富的跨越整个体系的一致安全措施。



■ JavaEE的N层体系结构

理想的JavaEE体系包括六个层：表示、应用、服务、域、连通性和持续化。这些层物理上横跨客户机和服务器，而他们逻辑上划分为Web容器、EJB容器和数据库，如图：





常用Web应用开发环境介绍

- 在开发过程中，基于JSP项目通常涉及到
 - ✧ Web服务器
 - ✧ 数据库服务器
 - ✧ 集成开发环境的选择



Web服务器

- WebLogic是BEA公司的产品，是目前应用最广泛的Web服务器，支持J2EE规范，而且不断的完善以适应新的开发要求，启动界面如图





Web服务器

- 另一个常用的Web服务器是IBM公司的WebSphere，支持J2EE规范，启动界面如图





Web服务器

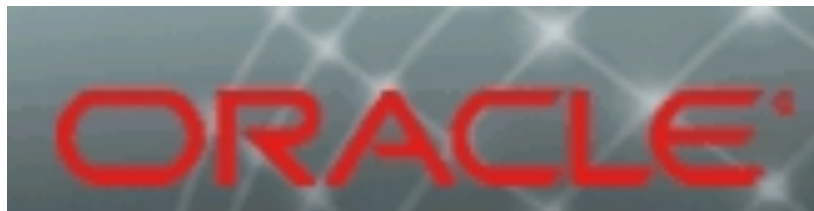
- 在小型的应用系统或者有特殊需要的系统中，可以使用一个免费的Web服务器：Tomcat，该服务器支持全部JSP以及Servlet规范，但是目前还不支持EJB，启动界面如图





数据库服务器

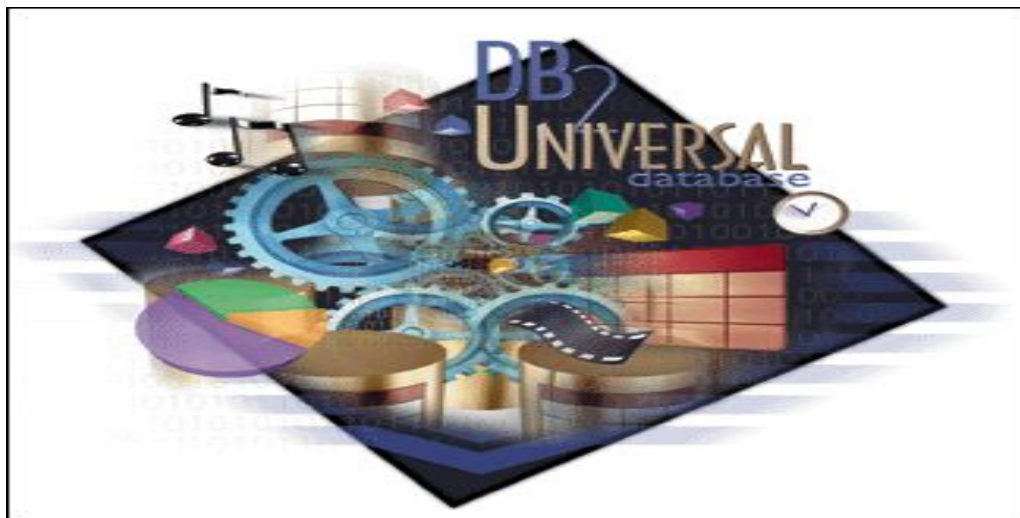
- 在Java相关的开发领域中，常用的是3种数据库：Oracle、DB2和MySQL。有时候也使用微软公司的SQL Server数据库服务器
- Oracle数据库是Oracle公司的产品，也是历史最悠久的数据库，世界前50强的企业中，80%使用Oracle数据库作为公司的数据库服务器。服务器的图标如图





数据库服务器

- IBM 公司的 DB2 也是一种常用的数据库，和 IBM 的 WebSphere 配合构成完整的企业级解决方案，数据库服务器的启动界面如图





数据库服务器

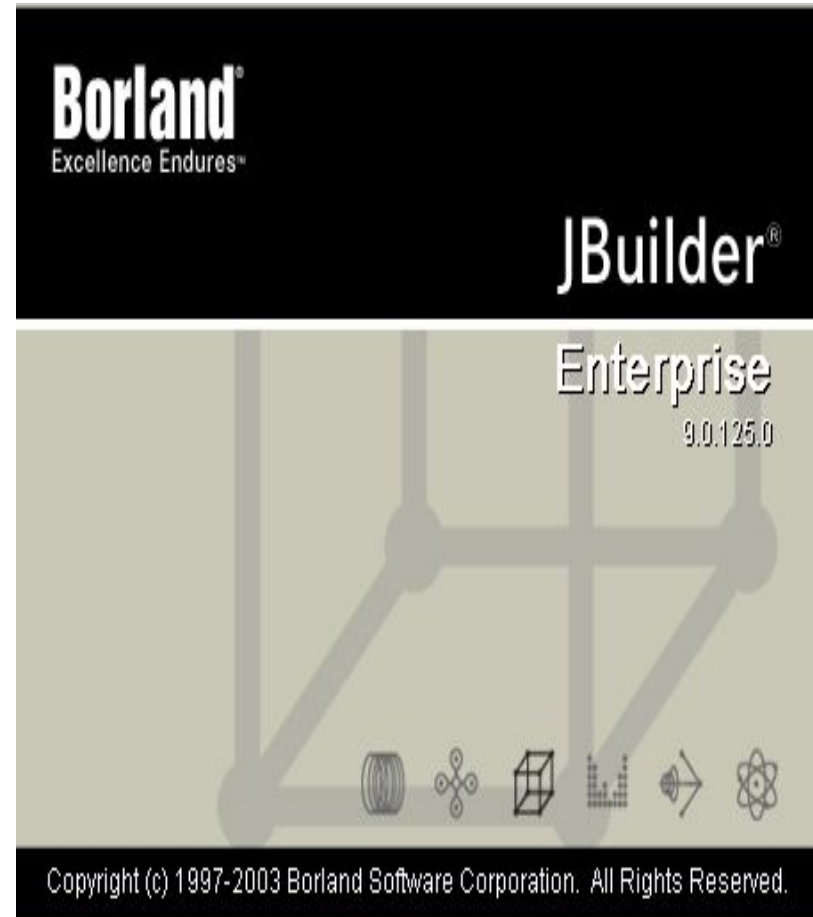
- 目前商业数据库服务器市场三足鼎立：Oracle、DB2和微软公司的SQL Server，在实际应用中，有一个常用的免费的小型数据库服务器：MySQL，启动界面如图





集成开发环境

- Web服务器和数据库服务器构成了发布平台，需要选择合适的开发工具以提高编码的效率，目前有两种常用的企业级集成开发环境，JBuilder和WebSphere Studio。
- JBuilder是Borland公司的产品，支持几乎全部的J2EE组件，是目前最为常用的开发环境，启动界面如图



集成开发环境



- IBM公司的WebSphere Studio，注意和WebSphere的区别，WebSphere是Web服务器，只能用作发布服务器，WebSphere Studio是一套集成开发环境，同时集成Web服务器功能，启动界面如图

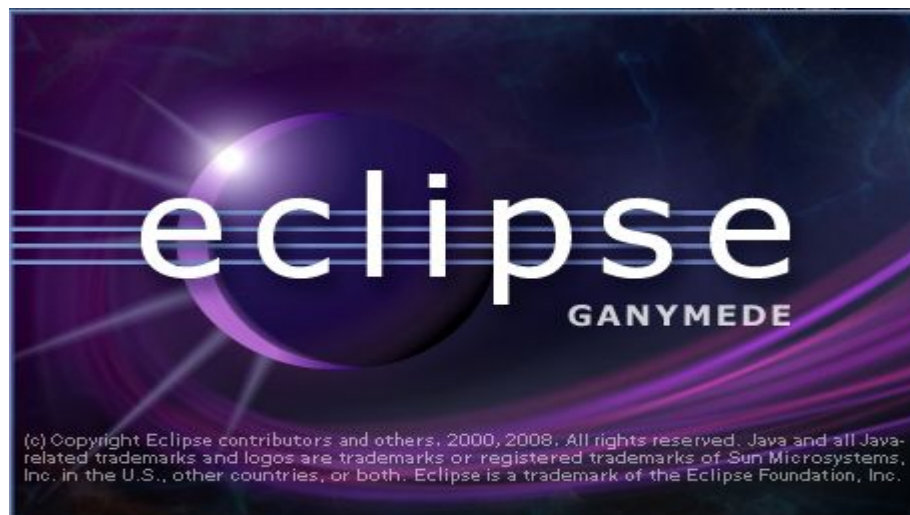


集成开发环境



■ Eclipse/MyEclipse

- ✧ Eclipse IBM主导推出的是一个开放源代码的、基于Java的可扩展开发平台。Eclipse提供对多重平台及不同语言的支持，其采用插件机制，是一个真正可扩展、可配置的IDE。
- ✧ MyEclipse 本质上为Eclipse开发JavaEE的插件集合。其企业级开发平台（MyEclipse Enterprise Workbench）是功能强大的商用JavaEE集成开发环境。





集成开发环境

■ NetBeans

- ✧ 由Sun公司（2009年被甲骨文收购）在2000年推出的开源的Java集成开发环境，可以方便的在windows,Mac,Linux和Solaris中运行。

