**北 京 邮 电 大 学**

**本科毕业设计（论文）开题报告**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 学院 | 计算机学院 | 专业 | | 网络工程 | | 班级 | 12 |
| 学生姓名 | 金元浩 | 学号 | | 2014211471 | | 班内序号 | 18 |
| 指导教师姓名 | 马跃 | 所在单位 | | 北京邮电大学 | | 职称 | 教授 |
| 设计（论文）题目 | （中文）公司网站的设计与实现 | | | | | | |
| （英文）Design and implementation of website for a company | | | | | | |
| 毕业设计（论文）开题报告内容：（主要包含选题的背景和意义；研究的基本内容和拟解决的主要问题；研究方法及措施；研究工作的步骤与进度；主要参考文献等项目）  **选题的背景、目的和意义：**  在网络技术发展迅速的今天，作为网络技术流行载体的网站在过去的几年中井喷式发展。各式各样的网站如个人网站，实用网站，企业网站等都出现在了庞大的网络世界中，作为新时代获取信息的主要方式，网页技术已经受到全世界无数网民的青睐，当然也包括数目庞大的开发者们。  从1993年英国物理学家提姆(Tim Berners-Lee) 在欧洲粒子物理研究所(CERN) 时发明的第一个只能显示简单字符的网站，到 1998年Table,css与javaScript技术加入网页技术的大家庭中，再到如今java Spring技术，微软.net技术，Python Django技术将网站建设体系化便捷化，网站技术发展迅速且具有革命性，时时刻刻影响着每一个信息化社会中的人们。  本课题即依托于丰富的网络技术，网页开发技术，通过实现现实生活中常见的公司网站的搭建过程，探讨各种网页开发技术的优点与不足，深入探索网页开发体系的当前结构与未来发展。  **对流行的网站技术的调研：**  网页技术主要包括前端，页面服务端，持久化层以及部署阶段。经过多年的发展与演进，各个技术模块均发展出了许多分支且各有优劣。在进行网站搭建的实践之前，我们对各个层面的各个技术进行了重点的调研：   1. 前端技术：   前端技术主要是对网站视图（即用户所见，所得）的设计与交互。基本技术是html（网页的基本模块设计），CSS（网页版式设计），Javascript（网页交互的实现）。这三种技术是网页前端的基础技术，但应用起来有些繁杂，所以有许多的外部库提供给开发网页的工作者使用。下面是我们对几种流行的外部库的调查与理解  **BootStrap**：来自 Twitter，是目前最受欢迎的前端框架。Bootstrap 是基于 HTML、CSS、JAVASCRIPT 的。其主要优点是①用户友好的语法结构，非常容易上手。②对移动设备支持良好，其响应式CSS可以自适应台式机，平板电脑和手机。③它包含了功能强大的内置组件，易于定制。④是开源的，方便定制我们的需求。在实际应用时主要负责网页版式等设计工作【1】  **Jquery**:一个JavaScript函数库，是一个轻量级的"写的少，做的多"的JavaScript库。可以非常方便的进行HTML元素选取与操作，CSS操作，HTML事件函数，JavaScript特效和动画，HTML DOM遍历和修改，AJAX操作以及其它丰富的插件库。在实际应用时主要负责网页的交互，是最流行的前后端交流方式。【2】  **Node.js**：可以被理解为运行在服务端的JavaScript，基于Chrome JavaScript运行时建立的一个平台，基于Google V8引擎。主要的特点是非阻塞和事件驱动，摆脱了传统前后端需要长时间轮询来实现实时通信并且解决了请求事务的阻塞危险。是一种具有革新意义的前后端连接器，主要用于实现实时交互。【3】  **Vue.js**：是一套用于构件用户界面的渐进式框架，优势在于轻量级及可以实现渐进增量。需要依赖Node.js。   1. 页面服务端框架：主要的框架是基于JAVA的Spring框架和基于Python的Django框架及Flask框架   Java Spring Web开发框架：主要基于三种技术   1. DI（依赖注入）/AOP（切面编程）：是一个以DI和AOP为核心Java Web一站式的集成（粘合）框架。DI和AOP能够让代码更加简单，具有良好的松耦合特性和可测试性，极大地简化开发。 2. MVC框架：字面解释即模型+视图+控制器模式。其最大作用是将开发的关注点分离，使开发者的代码复用性及组织性更好，使开发网页时摆脱繁琐的配置解决通用问题的过程，极大缩短了开发流程 3. ORM（Object Relation Mapping，对象关系映射）：将JAVA中的对象与数据库中的关系联系起来。是一种以面向对象的方式来进行数据库操作的技术。使开发者从繁重的“JAVA中编写SQL语句执行”的工作中解脱出来【4】   Python Django Web开发框架：  Django采用了MTV设计模式，即模型+模板+视图设计模式。封装了复杂的控制器，而使用调用方法的方式来执行。Django的主要目标是使得开发复杂的、数据库驱动的网站变得简单。Django注重组件的重用性和“可插拔性”，敏捷开发和DRY法则（Don't Repeat Yourself）。在Django中Python被普遍使用，甚至包括配置文件和数据模型。【5】   1. 持久化层：   即使用的数据库及其维护方案。当今比较流行的数据库有MSSQL SERVER,ORACLE,MySql以及轻量级的Sqlite。  **MSSQL SERVER**：是一个关系数据库管理系统，它最初是由Microsoft， Sybase和Ashton-Tate三家公司共同开发的版本。它的优点在于其图形化的用户界面及简单的编程接口还有提供数据仓库功能。但最大的缺点是其开放性差而且只支持C/S模式【6】  **ORACLE**：是目前世界上流行的关系数据库管理系统，系统可移植性好、使用方便、功能强，适用于各类大、中、小、微机环境。它是一种高效率、可靠性好的 适应高吞吐量的数据库解决方案。【7】总体来说Oracle是适用于大型项目的支持大并发，大访问量的最佳数据库。  **Mysql**：是开源的适合中小型项目的数据库，相比Oracle的高昂价格，免费的Mysql是许多项目搭建持久层的最佳选择。   1. 部署阶段：   部署所使用的HTTP Server的主流服务器包括：Apache HTTP Server，Apache Tomcat和Nginx。为了研究哪一种HTTP Server更加适合本项目，我们将其进行了比较与探讨。  HTTP Server的主要功能：   1. 监听端口 2. 接收（accept）socket连接 3. 解析HTTP请求 4. 使用通用或专用协议对请求进行分发 5. 接收分发的请求产生的运行结果 6. 将结果格式化成HTTP Response并写到socket里面 7. 关闭连接或者Keep-Alive   我们即基于这几类功能探究HTTP Server：   1. 连接处理方式   Apache和Nginx的一个巨大的差距是他们实际处理连接和请求的方式。在不同的网络流量条件下，这产生了两个服务器非常不同的表现。  Apache  Apcahe提供了多种多进程工作模式（Apache称这些为MPMs, multi-processing modules）来处理请求，基本上，这样可以让管理员很容易地变换服务器处理连接的方式。这些模式如下：  mpm\_prefork：这种模式为每一个请求创建一个处理它的进程。每个子进程同一时间只处理一个请求。如果请求数量小于进程数量，这种模式运行的非常快。不过当需要处理请求超出进程数量时，性能下降非常严重，所以对于很多应用场景这种模式不是一个好的选择。每一个进程都会耗损相当的内存，所以这种工作模式很难优化。当处理请求的内部组件不支持线程工作模式的情况下，这种模式可能仍然是一个好的选择。比如，PHP就不是线程安全的，所以这种工作模式被推荐为与mod\_php一起运行的唯一安全模式。  mpm\_worker: 这种模式创建的进程可以管理自己线程。每一个线程可以单独处理一个连接。因为与进程相比系统可以创建更非常多的线程，也就意味着这种模式比prefork模式能处理更多的连接。一个新的连接会被马上处理，而不需要等待一个空闲的进程来处理。  mpm\_event: 这种模式和worker模式很像，不过它优化了keep-alive请求的处理方式。当使用worker模式的时候，一个连接（connection）会持有一个线程直到这个连接失效，不管这个连接上是否有请求过来。在event模式下，有专门的线程用来处理和保持连接，然后将请求转发给其他线程处理。这种方式可以让系统逃脱处理大量keep-alive请求的泥潭，使请求处理器得到更快的执行。这个模式在Apache2.4版本中已被标记为稳定（stable）状态。  就像你看到的，Apache提供了一个灵活的结构可以配置不同的连接和请求处理算法。这些工作模式代表了服务器的功能演进以及伴随着互联网格局变化而来的对大并发处理的需求的增长。  Nginx  Nginx在Apache之后来到这个舞台，与生俱来就意识到需要面对大并发问。利用这方面的认识，Ngeix彻底地由内到外地使用了非同期，非阻塞和事件驱动的连接处理算法。  Nginx生成工作进程，每一个工作进程可以处理上千个连接。工作进程通过实现一个快速的循环算法不断的获取并处理事件。从连接中分离出实际的请求处理工作，让每一个工作进程只有当新的事件发生是才与一个连接产生联系。  所有被工作进程处理的连接被放在事件循环（event loop）里面，在这个循环里，事件被非同期处理，使得处理变成一个非阻塞的过程。当连接关闭，连接从这个循环中删除。  这种连接方式使得Nginx可以利用有限的计算资源处理不可思议的大量请求。因为服务器是单线程的并且并不为处理一个新的连接生成一个单独的进程，所以内存和CPU的消耗趋于保持相对一致，甚至是在大并发的情况下也如此。   1. 静态内容VS动态内容   在实际应用方面，Apache和Nginx之间最常见的比较是但请求到来时它们各自处理静态内容和动态内容的方式。  Apache  Apache使用传统的基于文件的方式处理静态内容的请求。它的性能主要取决于它是被设定成哪工作模式（上面提到的）。  Apache也可以处理动态内容，它通过嵌入一种处理语言到运行实例，在服务器内部就可以处理动态内容，而不需要依赖外部组建。它通过使用可动态加载的模块来开启处理动态内容的进程。  Apache的可在服务器内部处理动态内容的能力意味着配置动态处理进程也比较简单。不需要和一个附加的软件交互并且当内容处理的需求发生变化时模块也很容易替换。  Nginx  Ngnix本身没有任何的动态处理能力。如果想执行PHP代码或者为请求生成动态内容，Nginx必须将请求传递给一个外部的处理器并等待渲染好的内容（通常是HTML文档）然后再把它转发给客户端。  对于服务器管理员来说，这个意味着你必须配置Nginx和外部处理器之间的交互，这种交互必须基于一种Nginx能理解的协议（http, FastCGI, SCGI, uWSGI, memcache)。这可能会使事情变得稍微复杂，特别是试图占用允许连接的最大值时，因为已有了一个额外的连接用来转发请求到处理器。  不管怎么说，这个模式同时也有一些优点。因为动态解释器不是嵌入到Nginx的工作进程中的，所以它的开销只限于处理动态内容，而静态内容请求会被直接处理，Nginx只在需要的时候连接程序解释器。Apache也可以这样工作，不过这样的话就失去了我们在前面说的优点。   1. 分布式配置VS集中配置   对于服务器管理员来说，两个服务器表现出来的最明显的不同是是否允许文件夹级别的配置。  Apache  apache提供一个选项，允许对每个目录设置附加的配置。这个功能基于监测和实时翻译内容所在文件夹上一个隐藏文件中的指令来完成。这个文件就是大家所熟知的.htacess。  因为.htacess文件就存在于请求内容所在文件夹。当处理一个请求的时候，Apache检查每一个文件的路径，查找.htacess文件，执行里面的命令，这使服务器的分散配置成为可能。这个功能经常用于重写URL，控制访问甚至是缓存策略。  虽然上面说的例子可以在Apache的主配置文件中设置。但是.htacess有一些重要的优势。首先，Apache在每次请求来的时候解释指令，所以.htacess的配置会立即生效，而无需重启服务器。其次它允许无权限的用户来控制他们自己的web内容的某些方面，而无需修改Apache主配置文件。  这给某些web软件，如内容管理系统（cms）提供了简单方式来配资它们自己，而无需访问主配置文件。这也同样被用于共享主机提供商使他们在保持控制主要配置文件的同时让客户对他们自己的特定目录有控制权限。  Nginx  nginx不会即时解释.htacess文件，也不提供在主配置文件之外的任何支持目录级别配置的技术。这可能相对于apache来说不够灵活，但它却有自己的优势。  最被大家熟知的相对于基于.htacess机制实现目录级别配置的系统优势就是提高了性能。例如一个典型的Apache配置可能允许配置.htaccess在任何目录，这样的话每一个请来访问任何一个目录下的资源的时候，服务器都会检查这个目录以及它所有的父目录的.htaccess文件，如果在这个过程中有一个或者多个.htaccess文件被找到，他们必须被读取并解释执行。Nginx不允许目录重写技术，每一个请求过来Nginx只查找和读取一个文件（假设文件可以在约定目录结构下找到），所以Nginx处理请求更快。  另一个优势是关于安全的。分散的目录级别的配置也同时把安全配置web服务的责任分散到了每一个用户（web应用管理员）头上，他们可能无法胜任这个任务。确保服务器管理员控制整个web服务器可预防一些把控制权转交给他人所造成的安全隐患。  如果这些观点与你产生共鸣，你应该时刻考虑是否可以关闭解释执行.htaccess文件。   1. 文件VS基于URI的解释执行   web服务器是如何解释执行一个请求以及如何查找到与请求所匹配的系统资源的？这是另一个这两个服务器的不同之处。  Apache  Apache提供一种可以把请求转换成文件系统上的物理资源或者一个更抽象的URI的能力。总体上，以前的Aapache使用 <Directory>或者<Files>块配置，同时使用<Location>块配置更抽象的资源。  因为Apache完全是为web服务器所设计。默认情况下一个请求通常被解释成一个文件资源请求。它在查找请求路径中去掉域名和端口号后相对于document root的路径下的真实文件。默认情况下，文件结构会被以文档树的形式展示。  当请求没有匹配到文件资源时，Apache提供很多可选项处理这种情况。比如，一个alias命令可以让请求关联到另一个位置。用<Location>块可以使用URI代替文件系统工作。当然还可以使用正则表达式，从而使基于文件系统的资源查找更加灵活。  虽然Apache同时具有操作底层文件系统和网站空间的能力，但是它非常依赖于文件系统，包括使用.htacces文件实现目录级配置这，都可以被看作它的设计哲学。Apache docs警告用户当请求可以映射到底层文件系统时，最好不要使用基于URI的配置限制访问。  Nginx  Nginx被当做web服务器和代理服务器而创造。考虑这两种角色所需要的架构，它主要基于URI工作。只在需要的时候在将请求映射到文件系统。这可以被看作是构造和解释执行Nginx配置的方式。Nginx不提供任何关于文件系统目录的配置，取而代之的是解析URI本身。  举个例子，Nginx主要的配置块是server和location块。server配置块用来解释请求的域名部分，同时location配置块负责匹配URI中域名和端口号后面的的部分。从这个观点看，请求被解释成一个URI请求，并不是一个映射到文件系统上的文件。  对于静态文件，所有的请求最终会映射到文件系统上的一个文件。首先，Nginx找到用于处理这个请求的server配置块和location配置块，然后组合document root目录和URI，再根据指定配置做任何必要的调整。  这可能看起来很简单，但是解析请求成URI而不是文件系统路径使得Nginx能跟简单地以web、mail和代理服务器方式工作，通过简单的配置就以对应不同模式的request请求。Nginx不会检查文件系统直到准备好提供被请求的内容，这解释了为什么它没有实现一种类似.htaccess的配置方式。   1. 模块   Nginx和Apache都支持通过模块扩展服务器，但是他们的工作方式非常的不同。  Apache  Apache的模块系统允许你在服务器的运行期间动态的加载或者卸载模块。Apache的核心始终存在，而模块可以打开或关闭，以添加或者删除挂载到server的主程序部分的附加功能。Apache通过这一个功能完成很多任务。由于Apache平台非常成熟，有很多用于扩展Apache的模块存在。他们可以用于改变Apache的一部分核心功能，例如mod\_php就可以嵌入PHP处理器到每一个运行的工作进程中。  模块不仅仅只限于处理动态内容，它们还被用于URL重写，认证客户端，硬件化服务器，日志，缓存，压缩，代理，速率控制和编码等等。动态模块可以很容容易的扩展核心功能并且并不需要太多额外的工作。  Nginx  Nginx也实现了一个模块系统，但是它与Apache的实现方式很不同。在Nginx中，模块不是动态加载的，它们必须被编译到Nginx的核心程序中。  对于很多用户来说，这可能让他们感觉Nginx不够灵活。尤其对那些只会安装发布版本而不熟悉通过编译来管理维护自己软件的用户来说，这确实是一个问题。不过发布版本总是倾向于包含使用最广泛的模块，如果你需要一个非标准的模块，你必须要自己编译你的Nginx服务器软件。  编译Nginx的模块还是非常有用的，他们允许指出哪些功能你不想放在服务器中，而哪些功能你需要使用。很多用户也认为这样更加安全，因为那些未经挑选的组件不会挂载到服务器中。不管怎样，如果你的服务器已经配置好了，它很可能是妥协后的产物。  Nginx模块的功能很多都和Apache模块相似。例如，Nginx模块可以提供代理服务，压缩，速率控制，日志，重写，地理位置，认证，编码，流媒体和邮件功能。【8】  **对我国相关政策法规的调研：**  **第二条** 在中华人民共和国境内直接从亚太互联网信息中心等具有IP地址管理权的国际机构获得IP地址的单位和具有分配IP地址供其他单位或者个人使用的单位，适用本办法。  **第五条 中华人民共和国信息产业部（以下简称“信息产业部”）对基础电信业务经营者、公益性互联网络单位和中国互联网络信息中心的IP地址备案实施监督管理。**  **各省、自治区、直辖市通信管理局（以下简称“省通信管理局”）对本行政区域内其他各级IP地址分配机构的IP地址备案活动实施监督管理。**  **第六条 信息产业部统一建设并管理全国的互联网IP地址数据库，制定和调整IP地址分配机构需报备的IP地址信息；各省通信管理局通过使用全国互联网IP地址数据库管理本行政区域内各级IP地址分配机构报备的IP地址信息。**  **第七条 各级IP地址分配机构应当通过信息产业部指定的网站，按照IP地址备案的要求以电子形式报备IP地址信息。**  **第九条 各级IP地址分配机构应自取得IP地址之日起二十个工作日内完成IP地址信息的第一次报备。**  **第十条 各级IP地址分配机构申请和分配使用的IP地址信息发生变化的，IP地址分配机构应自变化之日起五个工作日内通过信息产业部指定的网站，按照IP地址备案的要求以电子形式提交变更后的IP地址信息。**  **各级IP地址分配机构的联系人或联系方式发生变更的，应自变更之日起十个工作日内报备变更后的信息。**  **附 录：**  **需报备的IP地址信息**  **一、备案单位基本情况，包括备案单位名称、备案单位地址、备案单位性质、电信业务经营许可证编号、联系人姓名、联系人电话、联系人电子邮件等。**  **二、备案单位的IP地址来源信息，包括IP地址来源机构名称、IP地址总量、各IP地址段起止地址码等。**  **三、备案单位的IP地址分配使用信息，包括：**  **（一）本单位自用的IP地址信息，包括IP地址总量、各IP地址段起止IP地址码、IP地址使用方式、网关IP地址、网关所在地址；**  **（二）尚未分配的IP地址信息，包括IP地址总量、各IP地址段起止地址码；**  **（三）向其他用户分配的IP地址信息，包括所分配的用户基本信息（包括用户名称、单位类别、单位所属行业、单位详细地址、联系人姓名、联系人电话、联系人电子邮件）、所分配的IP地址总量、各IP地址段起止地址码、网关IP地址、网关所在地址、IP地址使用方式。**  **四、自带IP地址的互联网接入用户信息，包括用户基本信息（含用户名称、单位类别、单位所属行业、单位详细地址、联系人姓名、联系人电话、联系人电子邮件）、自带IP地址总量、IP地址段起止地址码、自带IP地址的来源、网关IP地址、网关所在地址、IP地址使用方式等。【9】**  **本工程所采用的项目体系：**  根据前期的对网站开发技术的调研与比较，以及本项目的具体需求特点。本项目决定采用以下开发方案：  前端：html, css, javascript为基础，开发框架使用node.js+Vue.js响应式框架。  页面服务端：为快速开发及方便维护，使用python+Django框架进行开发  持久化层：公司网站的数据库规模属于小型，故使用与Django框架连接良好的Mysql数据库InnoDB（关系型数据库）模式。  部署环境：服务器端应当满足易于维护且系统紧凑的特点，故本项目采用linux Ubuntu系统，HTTP SERVER使用NginX服务器，通过Xshell 5进行远程连接维护。 | | | | | | | |
| 参考文献：  【1】 [www.bootcss.com/](http://www.bootcss.com/) Bootstrap中文网  【2】 <http://api.jqueryui.com/>  【3】 <http://nodejs.cn/api/documentation.html>  【4】 <https://baike.baidu.com/item/ORM>  【5】 <https://docs.djangoproject.com/en/2.0/intro/>  【6】 https://docs.microsoft.com/zh-cn/sql/sql-server/sql-server-technical-documentation  【7】 https://baike.baidu.com/item/Oracle数据库  【8】 <https://www.linkedin.com/pulse/apache-vs-nginx-practical-considerations-pablo-g-cisneros>  【9】**互联网IP地址备案管理办法（节选）：信息产业部网站 2005年2月22日** | | | | | | | |
| 指导教师签字 |  | | 日期 | | 年 月 日 | | |