

# 西安邮电大学

## 毕业设计（论文）

题目： 基于商城系统的高可用负载均衡架构

学院： 自动化学院

专业： 自动化

班级： 1403

学生姓名： 田祥

学号： 06141090

导师姓名： 程伟 职称： 讲师

起止时间： 2017 年 12 月 5 日 至 2018 年 6 月 10 日

## 毕业设计（论文）声明书

本人所提交的毕业论文《基于商城系统的高可用负载均衡架构》是本人在指导教师指导下独立研究、写作的成果，论文中所引用他人的文献、数据、图件、资料均已明确标注；对本文的研究做出重要贡献的个人和集体，均已在文中以明确方式注明并表示感谢。

本人完全理解《西安邮电大学本科毕业设计（论文）管理办法》的各项规定并自愿遵守。

本人深知本声明书的法律责任，违规后果由本人承担。

论文作者签名：

日期：      年    月    日

# 西安邮电大学本科毕业设计(论文)选题审批表

申报人	程伟	职 称	讲师	学 院	自动化			
题目名称	基于商城系统的高可用负载均衡架构							
题目来源	科研				教学		其它	√
题目类型	硬件设计		软件设计	√	论文		艺术作品	
题目性质	应用研究		√		理论研究			
题目简述	<p>(为什么申报该课题)</p> <p>LVS (Linux Virtual Server) 是一种负载均衡架构, 应用于大型网站的架构方面, 可以大大减轻服务器的访问处理压力。</p> <p>Keepalived 是一种高可用服务, 用于 LVS 的调度机器在单点故障的情况下的处理保障能力, 保证 LVS 集群的正常调度运作。</p> <p>随着网络的发展, 互联网公司也风生水起, 业务要求也越来越多, 如何用有限的服务器处理更多的业务量成为了当下互联网公司执着追求的方向, LVS 刚好能够解决负载问题, Keepalived 能够保证故障切换, 所以这两种组合十分应景。跟商城系统搭配起来更能体现出负载均衡的优点。</p>							
对学生知识与能力要求	<p>要求学生清楚互联网服务 http 的请求流程, 能够熟悉数据包的处理阶段, 以及在架构层面的网络知识。关于网站分布式架构相关的知识和实际动手能力要求十分到位。</p>							
具体任务以及预期目标	<p>(应完成的具体工作, 预期目标和成果形式)</p> <p>一个基于商城系统的高可用负载均衡架构和相对论文</p>							
时间进度	<p>2017. 12 - 2018. 3: 查找相关资料, 准备相关的硬件环境</p> <p>2018. 4 - 2018. 5: 搭建好整个系统</p> <p>2018. 5 - 2018. 6: 准备好论文, 做好答辩准备</p>							
系(教研室)主任签字	2017 年 12 月 9 日			主管院长签字		2017 年 12 月 9 日		

## 西安邮电大学本科毕业设计（论文）开题报告

学生姓名	田祥	学号	06141090	专业班级	自动化 1403
指导教师	程伟	题目	基于商城系统的高可用负载均衡架构		
<p>选题目的（为什么选该课题）</p> <p>在当下 B2B &amp; B2C 互联网商业模式盛行的前提下，一个好的网站服务对于网站经营者来说至关重要，可以在产品活跃的前提下帮助网站已最优模式运营，保障后台数据的快速访问和消费者用户的愉快体验。加上高可用系统就能避免单点故障，防止对外的 VIP 出现安全问题导致网站崩溃。接入的商城系统就是网站的一个缩影，完全可以当做网站来进行模拟，介于代价等原因综合考虑，以此来达到满意的设计效果。</p>					
<p>前期基础（已学课程、掌握的工具，资料积累、软硬件条件等）</p> <p>已学课程：《工业控制网络》，《自动控制原理》，《c 语言程序设计》等。</p> <p>资料积累：《TCP/IP 详解卷 1：协议》（Lawrence Berkeley，机械工业出版社）。《鸟哥私房菜：基础篇》，《鸟哥私房菜：架设篇》（鸟哥，机械工业出版社）。《操作系统概念》（西尔伯查茨，高等教育出版社）等。 软硬件条件：软件条件：nginx。硬件条件：笔记本电脑，linux 宿主机和虚拟机。</p>					
<p>要研究和解决的问题（做什么）</p> <p>lvs 负载均衡架构主要对于网络的依赖比较大，所以我使用的是宿主机和虚拟机之间的通信来解决这个问题，但仍然需要解决防火墙，selinux 和网关接口问题，这需要大量时间来调试。另一个问题就是需要跟商城系统的对接，因为是两个不同的架构，</p>					
<p>工作思路和方案（怎么做）</p> <p>工作思路：这个课题主要是通过通过在虚拟机上搭建 lvs 系统，再与商城系统对接。运用网络知识对整体架构进行疏导。高可用系统就是在此之上进行的一些防止单点故障的系统。在此过程中，需要将配置过程，配置文件等记录下来，便于总结。</p> <p>方案：第 1 周-第 2 周：搜集相关资料，完成开题报告。 第 3 周-第 5 周：学习 lvs 基本实现原理及扩展网络相关知识。 第 6 周-第 8 周：配置好 linux 服务器，虚拟机等基础环境。 第 9 周-第 10 周：完成 lvs 整体架构。 第 11 周-第 12 周：与商场系统实现对接。 第 13 周-第 14 周：完成论文撰写，准备答辩。</p>					
<p>指导教师意见</p> <div style="text-align: right; margin-top: 100px;">             签字：             <div style="margin-left: 200px;">2018 年 1 月 9 日</div> </div>					

# 西安邮电大学毕业设计（论文）成绩评定表

学生姓名	田祥	性别	男	学号	06141090	专业 班级	自动化 1403
课题名称	基于商城系统的高可用负载均衡架构						
指导教师 意见	(从开题论证、论文内容、撰写规范性、学习态度、创新等方面进行考核)  评分(百分制):      指导教师(签字): _____ 2108 年 1 月 9 日						
评阅 教师 意见	(从选题、开题论证、论文内容、撰写规范性、创新和预期成果等方面进行考核)  评分(百分制):      评阅教师(签字): _____ 年 月 日						
验收 小组 意见	(从毕业设计质量、准备、操作情况等方面进行考核)  评分(百分制):      验收教师(签字): _____ 年 月 日						
答辩 小组 意见	(从准备、陈述、回答、仪表等方面进行考核)  评分(百分制):      答辩小组组长(签字): _____ 年 月 日						
评分比例	指导教师评分    (%) 评阅教师评分    (%) 验收小组评分    (%) 答辩小组评分    (%)						
学生总评 成绩	百分制成绩				等级制成绩		
答辩委员 会意见	毕业论文(设计)最终成绩(等级):  学院答辩委员会主任(签字、学院盖章): _____ 年 月 日						

# 目 录

第一章 绪 论.....	1
1.1 课题背景.....	1
1.2 目的和意义.....	1
1.3 课题任务和本人工作.....	1
1.4 论文结构.....	2
第二章 集群系统和负载均衡.....	3
2.1 集群系统.....	3
2.1.1 集群的概念.....	3
2.1.2 集群的分类.....	3
2.1.3 典型集群系统举例.....	3
2.2 负载均衡.....	4
2.3 小结.....	4
第三章 Linux 虚拟服务器（LVS）.....	5
3.1 LVS 介绍.....	5
3.2 LVS 的三种模式.....	6
3.2.1 网络转换 NAT 模式.....	6
3.2.2 隧道 TUN 模式.....	7
3.2.3 直接路由 DR 模式.....	8
3.2.5 三种模式的比较.....	9
3.3 IPVS 调度算法.....	9
3.4 小结.....	10
第四章 集群的高可用.....	11
4.1 高可用的概念.....	11
4.2 常见的高可用技术方案.....	11
4.3 小结.....	11
第五章 高可用负载均衡集群系统的分析与设计.....	12
5.1 集群系统的需求分析.....	12
5.2 集群系统的整体设计.....	12
5.2.1 集群系统的三层机构与作用.....	12
5.2.2 集群系统的网络划分.....	13
5.2.3 集群系统的均衡策略.....	13
5.2.4 集群系统的整体设计图.....	13
5.3 集群系统配置方案设计.....	13

5.3.2 负载均衡配置方案设计.....	13
5.3.3 商城系统编写设计.....	14
5.3.4 群网络构建方案设计.....	14
5.4 集群系统整体架构拓扑图.....	14
5.5 小结.....	15
第六章 商城系统介绍.....	16
6.1 系统功能.....	16
6.2 系统编写.....	16
6.3 过程中遇到的问题与解决办法.....	19
第七章 高可用负载均衡集群系统的实现与测试.....	20
7.1 安装与配置集群系统.....	20
7.1.1 构建基础镜像环境.....	20
7.1.2Nginx 服务器的安装与配置.....	20
7.1.3 负载均衡器的配置与测试.....	22
7.1.4 高可用架构的配置.....	22
7.1.5 真实服务器上的项目配置.....	24
7.1.6 集群配置常见的问题与解决办法.....	24
7.2 集群在模拟商城系统中的应用.....	24
7.3 小结.....	25
结束语.....	26
致谢.....	27
参考文献.....	28

## 摘 要

当今的计算机技术时代，无疑是以网络计算机为中心的时代。网络的高效实用和计算机的稳定运行必将是经济繁荣，信息传播的主要渠道。用户—服务器模式的简易性和可维护性，使得该种模式在实际应用中被大肆使用。成千上万的服务和应用（如电子商务，博客论坛，新闻服务，网上银行等）都是围绕着 web 进行。这些都促使着网络用户剧烈增加和网络流量的急速增长。

这种趋势给网络带宽和服务器端都带来了巨大的挑战，站在网络技术的角度看，服务器响应速度和存储访问速度远远不及网络带宽的增长速率，由此可以预见的是越来越多的问题瓶颈都会出现在服务器端。由此可见，建立一个集高可用的，易维护和高性价比的网络服务系统平台来满足不断增长的网路服务需求已是迫在眉睫的任务。在这种现象下，基于 Linux 系统平台的负载均衡的虚拟服务器集群技术便显得尤为重要。

Linux 虚拟服务器集群技术（简称 LVS），它的实现目标是创建一个具有高可用的，易维护和高性价比的以 Linux 作为操作系统的服务器集群，来满足庞大的网路需求。本文主要讲的就是 LVS 是怎样在若干个单独连接的独立服务器之上架构一个整体对外提供单一服务的集群服务器，使这个集群结构对外访问的用户来说表现为一个整体服务来提供强大的网路服务。

该毕设的另一个重点是在设计的负载均衡系统的基础上建立一个高可用系统，用来处理避免当整体系统的调度节点故障之后，系统的可用性出现问题。在这些基础上探讨探讨负载均衡算法对于系统性能的影响，怎样彻底提高系统性能，适用于各种场景。在整体高可用负载均衡系统架构的基础上，我尝试开发了一个商城系统，用于展示整体高可用负载均衡系统的效果。

本文我将进行了一下几个主要的工作：

- 1，介绍我的论文选题的背景还有意义等，解释网路应用对服务器性能要求。
- 2，阐述和本课题相关的集群的概念和负载均衡相关理论知识，包括集群的定义，负载均衡技术的相关概念等。
- 3，解释 LVS 虚拟服务的相关知识，阐述 keepalived 高可用的相关概念只是。
- 4，利用自己的开发知识，编写了一个商城系统，用于结合整体架构系统，进行展示。
- 5，讲解实现了这一系统的过程并且对实现的集群系统进行了模式测试和高可用性测试。结果表明，整个架构系统是可行的，完全可靠的。
- 6，对本课题进行了小结及提出尚需改进完善的问题，并且对下一步研究的方向做了展望。

**关键词：**负载均衡；高可用；Linux 虚拟服务器；调度算法



## ABSTRACT

Today is a computer technology age, which is undoubtedly a network computer as the center of the era. The high efficiency and practicality of the network and the stable operation of the computer will be the main channels of economic prosperity and information dissemination. The simplicity and maintainability of the user-server mode make it widely used in practical applications. Thousands of services and applications (e-commerce, blog forums, news services, internet banking, etc.) revolve around the web. These all urge the sharp increase of network user and the rapid growth of network traffic.

This trend brings great challenges to both network bandwidth and server. From the point of view of network technology, server response speed and storage access speed are far less than the growth rate of network bandwidth. Thus, it can be predicted that more and more problem bottlenecks will appear on the server side. Therefore, it is an urgent task to set up a network service system platform with high availability, easy maintenance and high cost performance to meet the increasing demand for network services. Under this phenomenon, load balancing virtual server cluster technology based on Linux platform is particularly important.

Linux virtual server cluster technology (LVS), its goal is to create a highly available, easy to maintain and cost-effective server cluster with Linux as an operating system to meet the huge network requirements. This article is mainly about how the LVS is built on a number of separate connected independent servers. A cluster server provides a single service to the outside world, such that it can service users who have access to the outside world as a whole service with offer of a powerful network.

Another important point of this system is to build a high availability system based on the designed load balancing system, which is used to deal with the problem of system availability after the failure of the scheduling node of the whole system. On the basis of these, the influence of load balancing algorithm on system performance is discussed and how to improve the system performance that suitable for all kinds of scenarios. On the basis of the overall high availability load balancing system architecture, I have tried to develop a mall system to show the overall high availability load balancing system effect.

In this paper , a few key tasks is listed as following :

- 1 , This paper introduces the background and significance of my thesis, and explains the requirements of network application to server performance.
- 2 , This paper expounds the concept of cluster and the theory of load balancing, including the definition of cluster, the concept of load balancing technology and so on.
- 3 , Explain the knowledge of LVS virtual services and explain that the highly available concepts of Keepalived are only.

4, Using their own development knowledge, a shopping mall system is used for the integration of the overall architecture system.

5, The process of implementing the system is explained and the pattern test and high availability test of the implemented cluster system are carried out. The results show that the whole architecture system is available and completely reliable.

6, This paper summarizes this subject and puts forward some problems that need to be improved and perfect, and the future research direction is prospected.

Keywords: load balancing; high availability Linux virtual server; scheduling algorithm;

**Keywords:** load balancing; high availability Linux virtual server; scheduling algorithm;

## 第一章 绪 论

二十一世纪的现在，什么发展最快呢？你可能有很多想法，我举几个例子：：网上购物，社交通讯，听歌看电影等等，对！就是互联网计算机。这些行业的迅速发展都需要建立在强大的互联网基础上。在互联网发展之初，大家便沉迷于其简单的操作方式，能够方便快捷地分享网上的信息，让使用者能够很快地掌握到自己想要了解的这些信息。在此趋势下，海量的服务应用都以 web 为中心进行搭建分享，用户数量的增长和流量的迅速飞升也成为了当下的互联网环境所要迎接的巨大挑战。因此，当务之急就是让网络架构变强壮，服务质量得到提升。

### 1.1 选题背景

随着电子商务的兴起，日益增长的用户数量对电子商务平台构成了很大的挑战，要提升网站运行速度和质量刻不容缓。网络硬件设备使得网络带宽的问题得到很大程度地改善，所以，接下来最大的问题就只剩下网络服务器的性能问题了。

可以如下思考提高服务器性能：一是提高服务器的硬件性能。这在如今发展迅速的计算机时代，性能日新月异，但是单纯的提高计算机性能也跟不上指数增长的用户需求，另一方面代价就比较高，对于企业来说，很不划算；二是相对于单服务器组建服务器集群。单服务器存在很多不足：

- 1，服务器单一节点故障，一旦出问题就会影响业务；
- 2，要替换更高端的服务器，花销更大；
- 3，还会导致在升级的步骤比较繁琐并且还会让服务停止；
- 4，受到硬件性能极限限制。

所谓的多服务器集群就是把多个服务器集群通过一种技术手段绑定成一个整体，也就是一个虚拟服务器。

建立在 Linux 基础之上，LVS 是一种虚拟化技术，技术的基础是在 IP 层面进行内容请求分发这样的负载调度，而且在 Linux 内核中通过修改添加内核参数（ARP 协议相关）将这些方法实现，将一些可以毫无关系的单个服务器组合连接成了一个具有很高伸缩性的服务器集群。LVS 具有可以支持几百万个并发的很好的伸缩性。实践证明，LVS 集群是个很不错的集群技术选择。

将我自主开发的商城系统部署在 LVS 的后端服务器上，在这样的基础上，整合 keepalived 技术，使得集群系统更具有可靠健壮性。

### 1.2 目的和意义

因为 LVS 技术手段和 keepalived 集群软件是免费的开源软件，想要使用的话直接在官网下载使用即可。除此之外，我们可以在 Linux 系统平台上使用很多已有的开源软件，比如 Nginx。所以，在前人在 Linux 上研究的成果下，结合自己的前端知识，具体应用系统场景，更具有通用性。

### 1.3 课题任务和本人工作

课题任务：结合自己前端知识设计开发的商城系统，实现一个切实际的集群服务方案。

本人工作：

- 1，讲述本课题的实际背景意义，介绍服务性能依赖网络应用的原因；
- 2，阐述什么是集群与负载；
- 3，探索了 LVS 的工作原理，介绍实现技术模式。
- 4，研究 keepalived 的原理，考虑与负载调度的结合。
- 5，开发一个商城系统，用于测试实现的集群系统。
- 6，对本课题作出内容总结工作以及提出仍须改进的地方，以及对下一步的实现方向做出规划。

## 1.4 论文结构

第一章是绪论，这一部分我主要会介绍了选题的实际背景意义。

第二章内容主要阐述了集群系统，并介绍了负载均衡系统。

第三章 LVS 的介绍，对 LVS 的相关知识内容进行阐述。

第四章介绍高可用相关概念，解释高可用的理论与技术。

第五章对整体系统设计与分析，找出适合的最佳设计方案。

第六章介绍自己开发的商城系统，用于展示。

第七章整个系统的实现，所有环节的安装与测试。

第八章对设计过程进行总结，归纳仍然需要改进的地方。

## 第二章 集群系统和负载均衡

本章主要从集群和负载均衡两方面进行讲解。通过了解集群的概念，能够帮助我们理解本课题的背景意义和实现思路，在分析设计方案的同时，也能够对系统的整体概念有个深入的了解。负载均衡概念是在集群的基础上建立起来的一种运行模式，帮助服务提高质量，两者相辅相成。

### 2.1 集群系统

#### 2.1.1 集群的概念

一般来说，集群大致可如下理解：利用一定的技术手段把物理连接的计算机结合成一个整体，这种所谓的技术手段有 LAN（或者 WAN）等高性能网络。集群中的这些服务器之间相互协作，协同工作来完成相同的服务任务（服务商要对外提供的服务）。一般都把集群中的每个服务机叫做节点，这种所谓的节点可以是各种计算机，比如大型机、云主机等。

综上所述，集群的概念可以这样理解：一是在硬件方面声张单独作用的服务；二是一致性的系统集群。

#### 2.1.2 高可用集群（High Availability Cluster）

集群有很多特征，也有很多分类方法，这里主要介绍高可用集群，其他两个集群不做分析。

高可用集群，作用是以减少如果出现服务中断（节点意外宕机）状况的服务类型，其存在的意义就是把服务变得健壮可靠。

什么是系统的可用性呢？一个系统始终保持着在线状态（上线之后）而且可以对外提供服务，外部可以访问——这就是可用性。它是究于服务是否强大或者可变换来评判的。能让系统不再服务的状况数不胜数，配置高可用的目的就是最大程度缩减这个时间以及重新试业务上线提供服务的所有时间。从工作方式出发，可以把集群分为主—主模型，主—从模型以及混合型三种：

##### ①：主—主型

双主模型会各自分配负责的业务，当一个节点出现故障情况之后，另一个主节点会接管它的所有业务，保证整体服务的正常运行。这样的话会最大程度的使用规划硬件资源，但是在接替之后的服务器处理效率明显下降。

##### ②：主—从模型

主从模型得配置一个调度节点（master）在正常工作时另一台调度节点（backup）处于备用状态，这样，集群的可用性达到最大，对性能的影响也降低到了最小化。master 在工作时，备节点 free；如果 master 由于意外导致出现故障不能再提供服务之后，backup 将自动代替主节点的工作进而继续照常为客户提供服务（前提是主节点和备节点的硬件一致）。

#### 2.1.3 Linux Virtual Server（LVS）

1998 年 5 月，章文嵩博士及其团队成立了 LVS 的开源软件项目组，这就标志着虚拟 Linux 服务器的开始。

LVS 负载均衡的核心是内核补丁，所以第七层用户态的应用程序都不会受到集群系统组建而带来的影响，即不需要做任何代码层面的修改就可以在 LVS 组建的集群服务器上运行。

LVS 应用相当广泛，介于我的课题是基于 web 服务，就主讲对于 web 的影响，可以支持几百万个并发连接。介于 LVS 软件是可以免费获得源码的开源软件，所以应用市场比较广泛。很多国内国际重负载的大公司都使用了 LVS 的架构。

## 2.2 负载均衡

负载均衡（Load Balance），其形式就是建立一个服务器集群，把来自用户的请求报文依据某种事先设定好的策略分配到这个集群中，因为集群中的服务器比较多，在面对大量的请求访问的时候，能够处理的效率变高，这样就提高了整个服务的健壮性。

集群最主要的特点是多个服务节点共同提供服务（调度节点进行请求流量分配）。为了让系统更可靠，更健壮，怎样合理地让所有硬件资源承受到的负载平均下来（或者分配合理），是负载均衡的主要目的。如果出现局部超负荷负载，服务主机的损坏，服务的供应不足将在所难免；同样，轻负载不能很好的利用硬件等条件。

## 2.3 小结

本章主要介绍了本课题的基础——集群和负载均衡的概念。集群是架构的基础，是负载均衡的依赖基础，在解释课题之前，介绍这些概念很有必要，让我们能够更深入的理解课题的作用和意义。集群和负载均衡相辅相成，都是现代网络的基础。

### 第三章 Linux 虚拟服务器（LVS）

本章将主要介绍 LVS 虚拟服务—基于 Linux 内核支持的负载均衡技术，也是整个毕设系统的核心负载均衡部分。这个服务的实现是在 IP 层创建一个基于内容请求分发的调度，并且具有高健壮可用性和强大的可伸缩性的基于 Linux 系统服务器集群。还介绍了 LVS 中负载均衡的三种模式以及互相之间的比较。还有 LVS 调度算法的学习探讨。

#### 3.1 LVS 介绍

即 Linux virtual server，在第二章介绍的集群基础上进行整合，对用户不透明，当请求到达之后，调度机进行分配，让后端服务器来处理，这些过程用户都不知道，所以就是虚拟服务。

LVS 所创建的虚拟服务是利用的集群技术和 Linux 系统平台实现的，具有不错的稳定性和可靠性。下面这个框架能够好的帮助我们理解，如下图 3.1 所示。

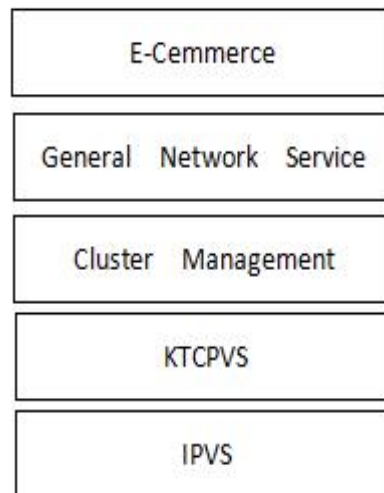


图 3.1 LVS 框架

在 LVS 框架中，最重要的软件就是 IPVS，它是有三种调度模式的虚拟服务软件，是一个用于内容请求分发的内核交换和集群管理的软件。利用这个软件和框架，我们可以实现很多搞可用性和可伸缩性的网络集群服务的部署。

通用架构如图 3.2 所示。

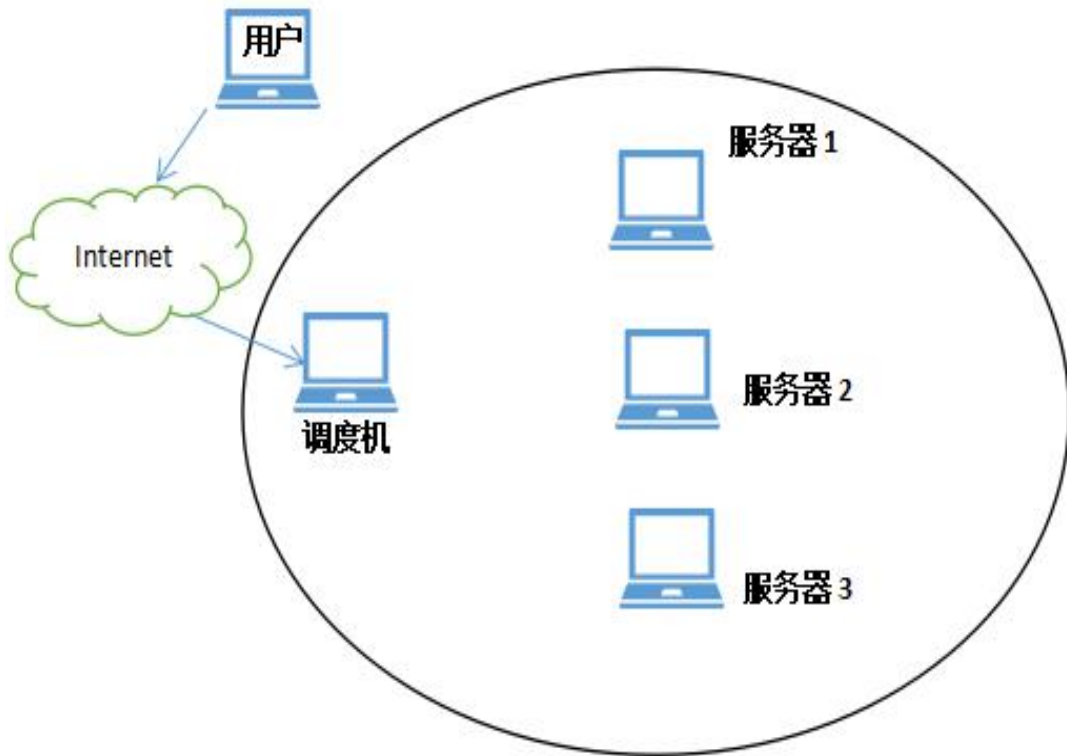


图 3.2 LVS 集群的通用体系结构

由图可看出，主要组成部分如下：

1，负载调度器（load balancer）

是个应答器，也是调节调度器，把用户请求给服务器发送过去，起着一个类似转发的作用，而客户端在收到处理结果后始终认为其来源于自己之前所发送的那个 IP 地址。

2，服务器池（server pool）

实际作为的服务器，可以在其上发布自己的实际业务。

### 3.2 LVS 的三种模式

#### 3.2.1 网络转换 NAT 模式



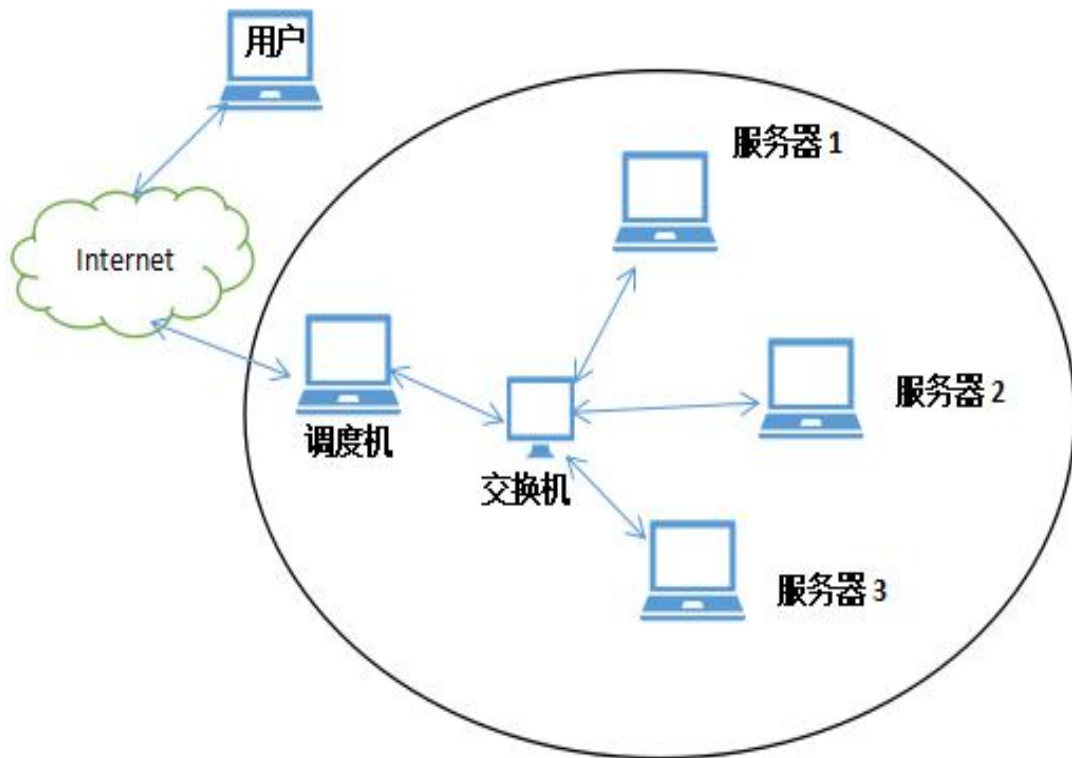


图 3.3 NAT 模式架构图

如图 3.3 所示，在所有服务器之前会有一个调度器，他们是通过交换机相连接的。这些服务器就是提供相同的网络服务后端真实服务器，即是无论来自用户端的服务请求被转发到哪一台后端服务器上，返回给用户的结果都是一样的。

工作流程：当调度器收到了来自用户端的服务请求之后，进行一个转发行为（给 server pool），调度器将处理过后的报文返回给用户。

优点：这个模式应用范围比较广，一般计算机系统都支持，服务器支持自己的私有 IP。

缺点：流量都会经过调度机，所以这对调度器要求比较高，也是模式的问题所在。

### 3.2.2 隧道 TUN 模式

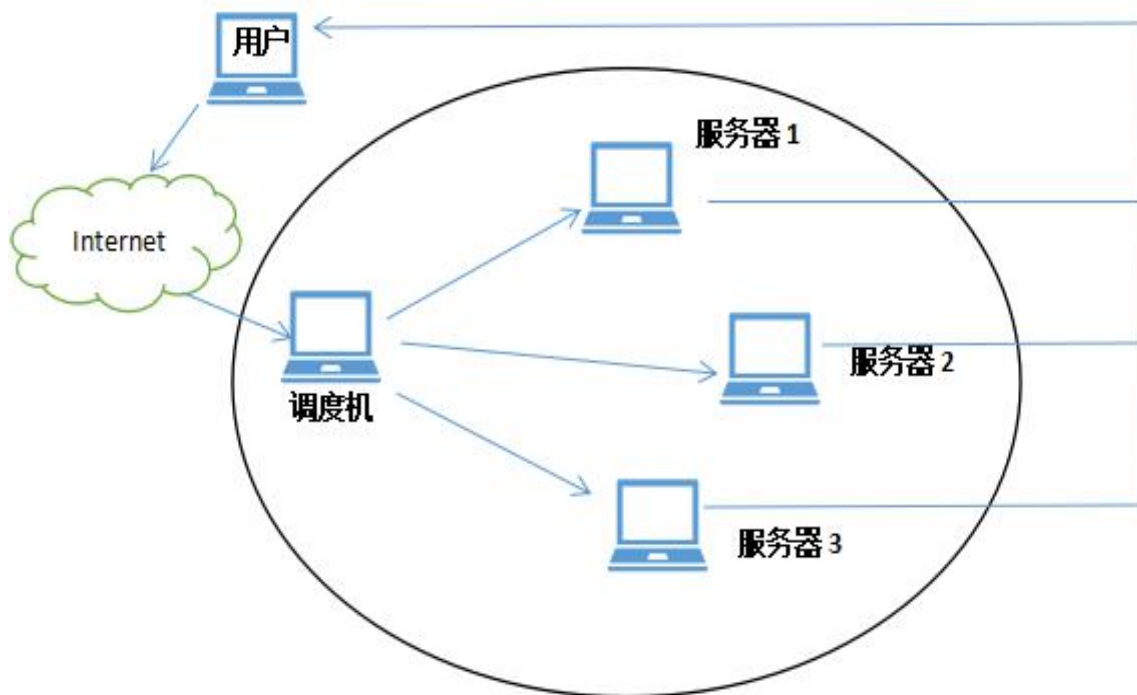


图 3.4 TUN 模式架构图

如图 3.4 所示，隧道模式的架构与转换模式相似，但区别体现在结果返回过程。当后端服务器处理了请求之后，自己返回给用户，这样就减少了调度器的压力。

缺点：需要集群的所有服务器（包括调度器）满足 TUN 协议，这对构建较大规模的集群来说就显得代价偏高。

### 3.2.3 直接路由 DR 模式

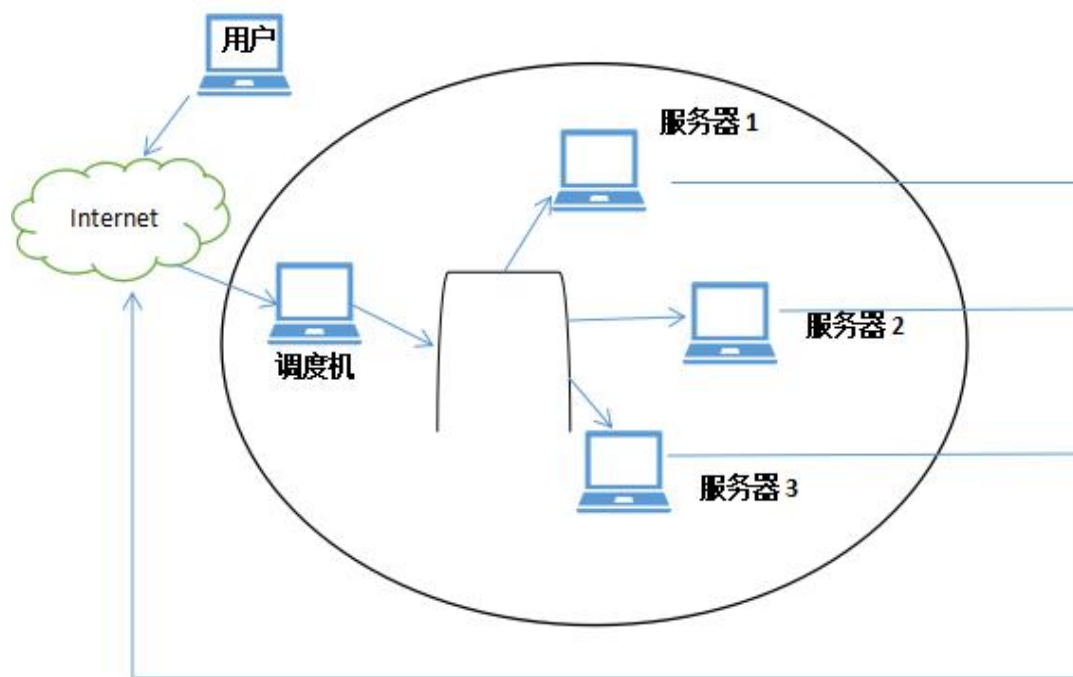


图 3.5 DR 模式架构图

如图 3.5 所示，直接路由模式延续了隧道模式的优点（后端服务器直接返回给用户），但区别是路由模式大大减少了对服务器的要求，加大了该种模式使用的范围。

优点：调度器和服务器群都必须在网络上能够通信。在返回给用户的报文的源 IP 是整个系统的 VIP，而不是服务器的真实 IP，大大提高了集群的安全性。

### 3.3 三种模式的比较

上面介绍了各个模式，先来比较差异，如表 3.6 所示：

表 3.6 模式比较

	NAT	DR	TUN
建立难度	易	有一定难度	
可扩展性	差	好	很大
带宽	小	大	大
支持的服务数量	少（10~20）	大于 100	
延迟	最大	小	大
IP 包修改	IP 地址和端口	请求包的 MAC 地址	进行 IP 封装
后台服务器 OS	任何支持 NAT 的 OS	多数	支持 IP TUN
网络连接要求	局域网		本地或远程
缺省路由	负载均衡服务器	不限	
实际服务器限制	无	lo 不影响 ARP	TUN 设备不响应 ARP
服务器端口映射	可以	不可以	

### 3.4 IPVS 调度算法

这是基于虚拟服务的调度算法，是 IPVS 自带的调度算法，每个算法都是依据 TCP 连接进行的，所以这在处理很多类似 DOS 攻击的访问上有很大帮助。

IPVS 调度算法（主要介绍轮询调度算法和加权轮询调度算法，其他算法这里不做分析）：

#### 1，轮询调度

所谓轮叫调度，就是挨着顺序，一个接下一个的将请求转发到不同的服务器上（ $i=i+1$ ），简洁明了。

#### 2，加权轮询调度

这种算法的作用就是解决像服务器的性能不一致的问题，对于性能较好的服务器赋予较高的缺省值，这样在发配请求的时候，缺省值高的服务器会优先收到请求进而处理。

#### 3，最小连接调度

#### 4，加权最小连接调度

#### 5，基于局部性的最少连接

#### 6，带复制基于局部性的最少连接

#### 7，目标地址散列调度

#### 8，源地址散列调度

### 3.5 小结

通过介绍 LVS 的原理及负载均衡模式的介绍，体现出了 LVS 的架构简洁明了，让大家对 LVS 的理解更加清晰。还有调度算法的介绍，充分解释了 LVS 的工作机制。所以 LVS 的可靠性和伸缩性显而易见。。

## 第四章 集群的高可用

本章将介绍整体系统的另一部分——高可用部分，将会对高可用的概念和常见方案进行讲解，并举例说明。高可用方案能够有效解决服务器集群单点故障，大大加强网站系统的可用性。

### 4.1 高可用的概念

减少服务中断的技术，就是保证服务不会坏掉（在出现意外故障时），能够一直运行，高度可用的意思。实现原理主要是自动检测，自动切换（故障转移）和自动恢复。

### 4.2 Keepalived

在所有的高可用技术中，keepalived 是和 LVS 最贴切的，配合度最高，所以在本题目中，使用 keepalived 方法。

在 master 端，集群模块一直负责监听各个服务器的状态，当出现问题随时进行切换，依据配置文件的设置每监听一次的时间。当 master 出现故障，backup 接收不到返回的信号，则直接将 backup 提升为主要负责节点，进行业务处理。

### 4.3 小结

本章主要介绍了高可用的概念和几种常见的高可用技术实现软件。高可用本身就是为了实现业务的正常运行，保证在出现意外因素的时候会实现基本的检测，故障切换和自动恢复，保证业务的正常服务。

## 第五章 高可用负载均衡集群系统的分析与设计

### 5.1 集群系统的需求分析

结合自己开发的商城系统，使其能够在收到宿主机访问之后，均衡地分配到虚拟机上的服务器，或者说在出现意外情况后，系统还是能够正常访问的。所以要满足下面的需求：

① 集群系统需要对用户透明，用户无法知道架构状态，也不知道自己访问的具体是什么东西，跟正常的服务器如出一辙，只不过这台计算机的性能比较卓越；

② 必须设定一定的策略来分配，最大化利用率。

③ 如果一台服务器出现问题，立刻会在集群中剔除，将流量分配到其他服务端处理。

④ 一台调度器（一般为 master）出现问题，backup 能取而代之进行工作，将来自用户的请求根据策略照常分配到后端。

基于自己的硬件资源，将依据这四个条件对系统进行分析选择。

### 5.2 集群系统的整体设计

前后架构如图 5.1 所示：

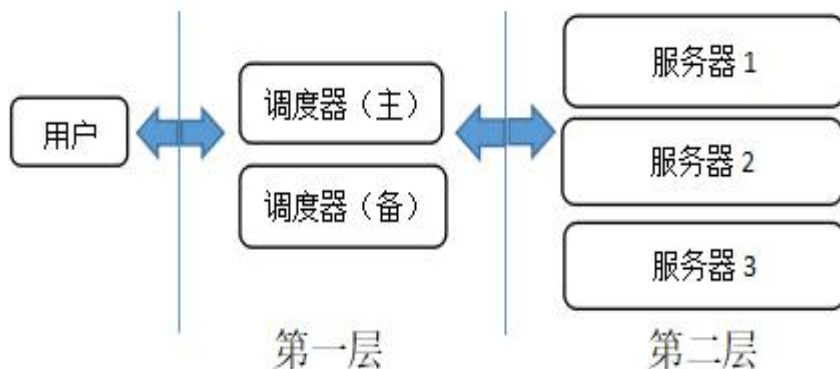


图 5.1 系统总体架构

分为用户区域，高可用块，负载均衡部分，后端服务部分。

#### 5.2.1 集群系统的机构与作用

第一层：高可用负载均衡层

负载均衡层节点负责流量调度，另一个任务是负责主备节点切换，当主备节点出现故障宕机时，进行主备切换。一般情况下选用一个服务器做主节点，另一个服务器做备节点。用户的请求到达该层的时候会先被分配到主节点，由主节点根据策略分配到后端 server，当主节点出现故障的时候将来自用户的请求分配到备节点。如果主节点恢复正常，则切换回主节点。

第二层：应用服务层

这一层的作用就是用来处理来自调度器的用户请求。该层主要是 web 服务器，通过将自己的业务项目发布在服务器上，起到用户访问的目的。应用服务层由两个或两个以上的物理服务器组成，运行 nginx，ftp 或者其他应用。

### 5.2.2 集群系统的网络划分

由于硬件条件限制约束，本人的毕设都是在自己的 Linux 系统上通过虚拟机完成的，通过宿主机充当用户端，调度机和后端服务器都是在虚拟机上部署的，互相之间的通信是通过网络桥接来搭建的，所以所有的本地 IP 都是自己设定的。

提供给服务端的虚拟 IP（VIP）是 172.25.24.100，整个系统都是在 172.25.24.xx 网段配置的。所以不存在任何依赖外部 IP 的因素。

### 5.2.3 集群系统的均衡策略

服务器的负载情况不能仅仅由当前的活动连接数或活动进程数等单一的指标来确定，因为我部署服务项目的环境是用的 Linux 系统里面的虚拟机，所以在 IPVS 算法里面，RR 轮询调度是最合适的选择。简洁方便的同时，很好地演示出来才是目的。

### 5.2.4 集群系统的整体设计图

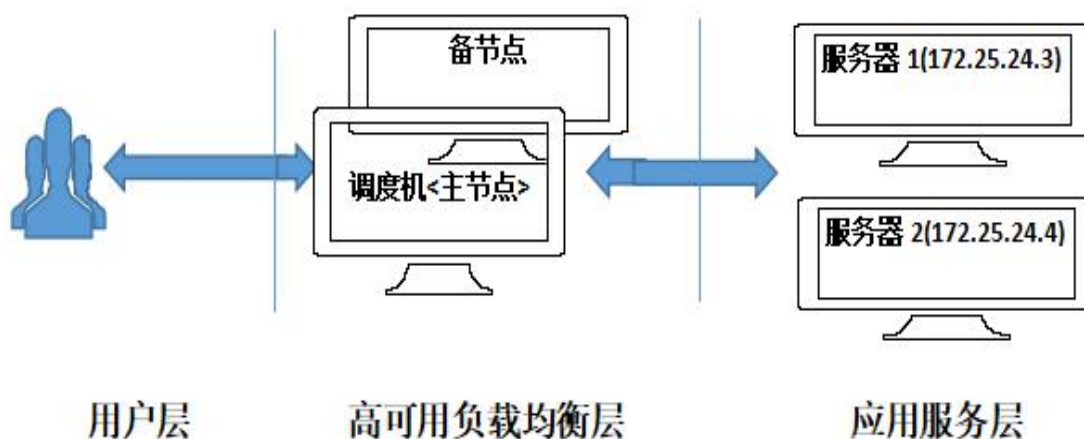


图 5.2 集群系统的整体设计图

## 5.3 集群系统配置方案设计

本人在构建之处就决定将所有的服务搭建在本人的 Linux 系统里面的虚拟机来实现，所有的虚拟机都是自己在 Redhat 官网下载下来的最小镜像，分析自己的题目要求，搭建部署自己需要的环境基础，并作为母镜像。所有用到的虚拟机都是依赖于此镜像建立的。

### 5.3.1 高可用负载均衡配置方案设计

依据上文分析，负载均衡使用 lvs-DR 模式进行配置，后端分配策略选用 RR 策略，高可用选择 keepalived，介于其简洁明了性。

### 5.3.2 真实服务器配置方案设计

服务器的硬件基础是母镜像建立的虚拟机，应用发布软件是 nginx 服务器，来发布我的商城系统，处理用户请求。

### 5.3.3 商城系统编写设计

模仿天猫淘宝等电子商城系统，建立一个类似的商城系统，用于测试高可用负载均衡系统的健壮性和效果。因为要测试负载均衡的流量分配，所以两个服务器上部署的商城要做区别。在 server1 上部署“潮流男女店铺一号店”，在 server2 上部署“潮流男女店铺二号店”，并且在两个店铺上发布的衣服效果不同，用于区别。

### 5.3.4 集群网络构建方案设计

集群角色	系统 VIP	系统域名	内网 IP	网络接口
主调度器	172.25.24.100	<a href="http://www.taxing.com">www.taxing.com</a>	172.25.24.1	eth0:0
被调度器			172.25.24.2	eth0:0
server1			172.25.24.3	eth0:0
server2			172.25.24.4	eth0:0

图 5.3 设计安排

系统的虚拟 IP（VIP）是一个方便访问的 IP，依据自己设定，系统域名也是自己设定的，都是局域内网自己设定的，无关其他。

### 5.4 集群系统整体架构拓扑图



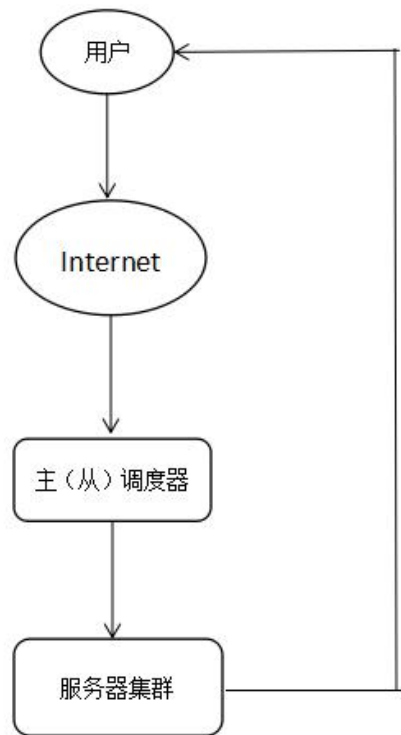


图 5.4 集群系统整体架构拓扑图

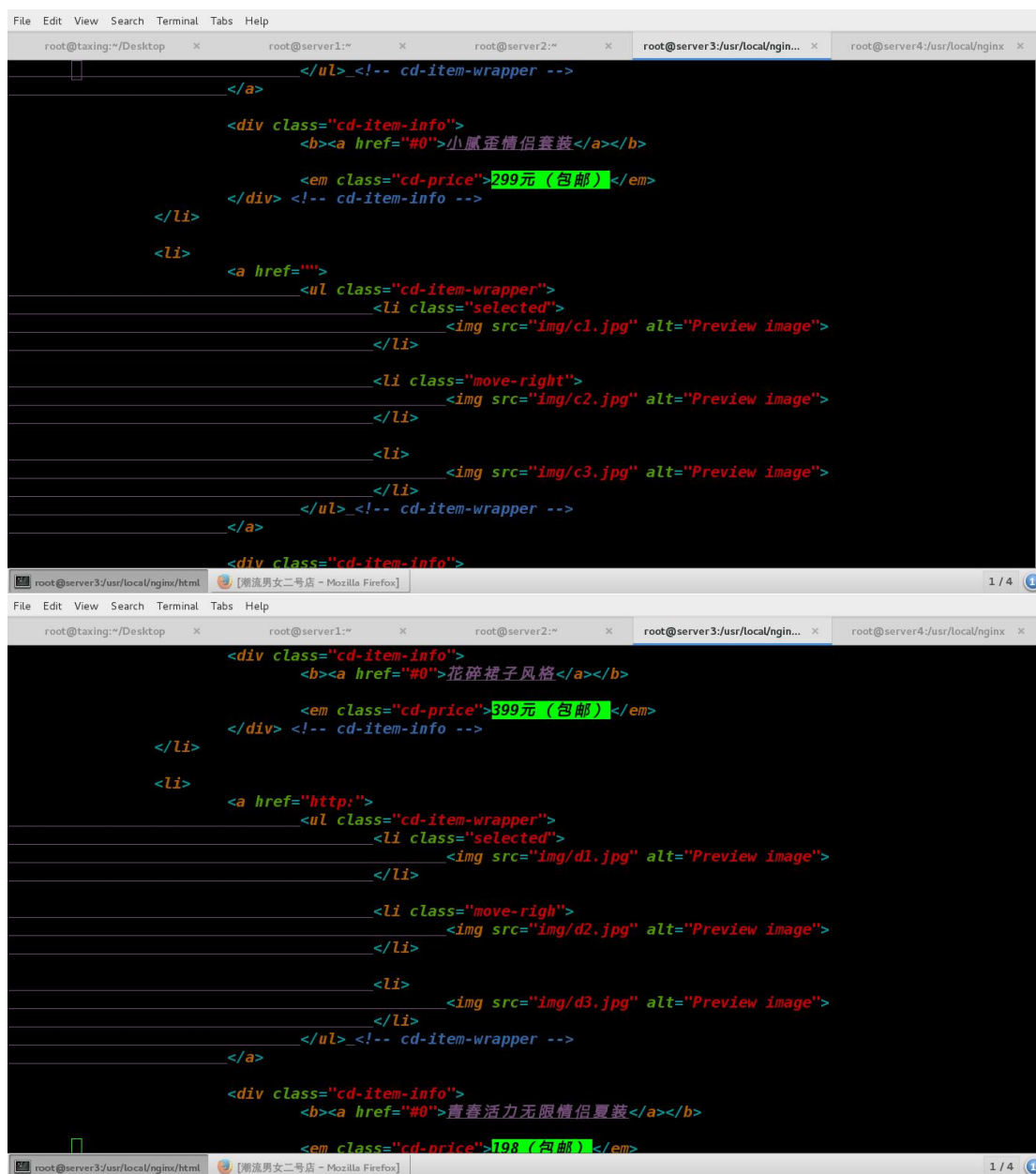
## 5.5 小结

这一章主要总体介绍了整个系统的各个部分的配置设计，分析每个系统的设计思路和分析对比，找出合适自己整体系统的方案。介于硬件基础，只能选择适合虚拟机的方案进行配置设计，但丝毫不想影响系统的效果和性能。

## 第六章 商城系统介绍

## 6.1 系统功能

- ①可以在线挑选衣服;  
②可以支付购买;



```

File Edit View Search Terminal Tabs Help
root@taxing:~/Desktop x root@server1:~ x root@server2:~ x root@server3:/usr/local/nginx... x root@server4:/usr/local/nginx x

<em class="cd-price">198 (包邮)</em>
</div> <!-- cd-item-info -->
</li>
<li>
  <a href="">
    <ul class="cd-item-wrapper">
      <li class="selected">
        
      </li>
      <li class="move-right">
        
      </li>
      <li>
        
      </li>
    </ul> <!-- cd-item-wrapper -->
  </a>
  <div class="cd-item-info">
    <b><a href="#0">灰白色调的休闲拉风款</a></b>
    <em class="cd-price">288 (包邮)</em>
  </div> <!-- cd-item-info -->
</li>
</ul> <!-- cd-gallery -->
<div style="width: 100%; height: 50px; line-height: 50px; text-align: center;">
  <a href=""><b></b></a>
</div>

```

root@server3:/usr/local/nginx/html [潮流男女二号店 - Mozilla Firefox] 1 / 4

```

File Edit View Search Terminal Tabs Help
root@taxing:~/Desktop x root@server1:~ x root@server2:~ x root@server3:/usr/local/nginx... x root@server4:/usr/local/nginx x

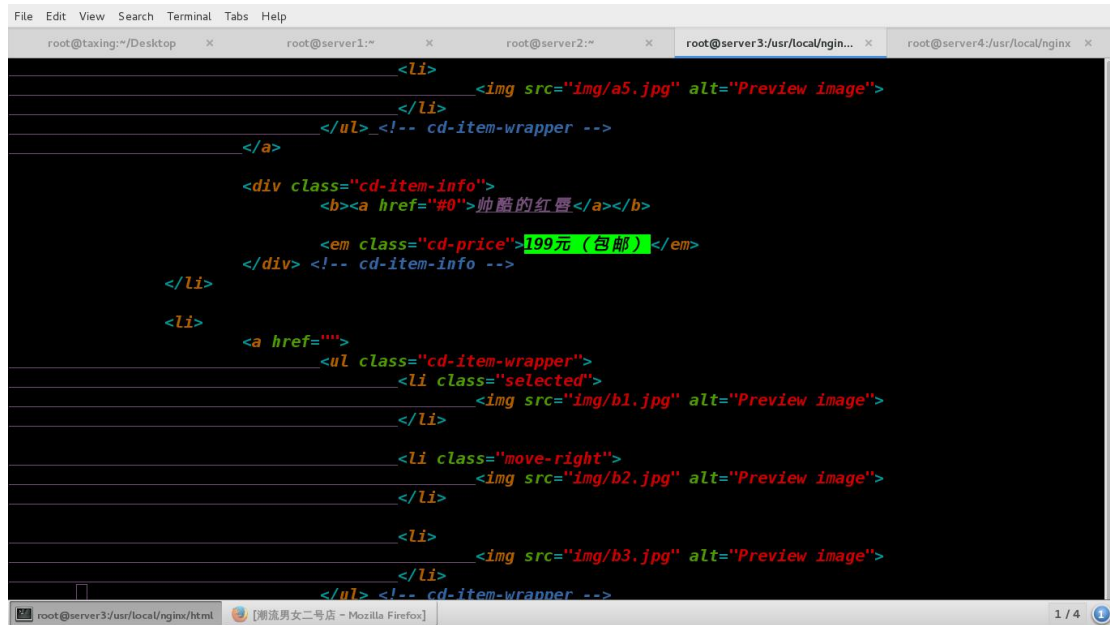
<li>
  <a href="">
    <ul class="cd-item-wrapper">
      <li class="selected">
        
      </li>
      <li class="move-right">
        
      </li>
      <li>
        
      </li>
    </ul> <!-- cd-item-wrapper -->
  </a>
  <div class="cd-item-info">
    <b><a href="#0">海滩格子调调</a></b>
    <em class="cd-price">388 (包邮)</em>
  </div> <!-- cd-item-info -->
</li>
</ul> <!-- cd-gallery -->
<div style="width: 100%; height: 50px; line-height: 50px; text-align: center;">
  <a href=""><b></b></a>
</div>

```

root@server3:/usr/local/nginx/html [潮流男女二号店 - Mozilla Firefox] 1 / 4

```
File Edit View Search Terminal Tabs Help
root@taxing:~/Desktop x root@server1:~ x root@server2:~ x root@server3:/usr/local/nginx... x root@server4:/usr/local/nginx x

!doctype html
<html lang="en" class="no-js">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <link rel="stylesheet" href="css/reset.css"> <!-- CSS reset -->
  <link rel="stylesheet" href="css/style.css"> <!-- Resource style -->
  <script src="js/modernizr.js"></script> <!-- Modernizr -->
  <title>潮流男女一号店 </title>
  <meta name="keywords" content="" />
  <meta name="description" content="" />
  <style>body{font-family: "microsoft yahei";}</style>
</head>
<body>
  <p class="credits">时尚潮品</p>
  <ul class="cd-gallery">
    <li>
      <a href="">
        <ul class="cd-item-wrapper">
          <li class="selected">
            
          </li>
          <li>
            
          </li>
          <li>
            
          </li>
        </ul> <!-- cd-item-wrapper -->
      </a>
      <div class="cd-item-info">
        <b><a href="#0">海滩风格子调调</a></b>
        <em class="cd-price">388 (包邮)</em>
      </div> <!-- cd-item-info -->
    </li>
  </ul> <!-- cd-gallery -->
  <div style="width: 100%; height: 50px; line-height: 50px; text-align: center;">
    <a href=""><b></b></a>
  </div>
  <div style="width: 220px; height: 150px; position: fixed; z-index: 1000; right: 0px; top: 0px; background: #fff;">
    <script type="text/javascript">
      var cpro_id = "u2586467";
    </script>
    <script src="http://cpro.baidustatic.com/cpro/ui/c.js" type="text/javascript"></script>
  </div>
<script src="js/jquery-2.1.1.js"></script>
<script src="js/jquery.mobile.min.js"></script>
<script src="js/main.js"></script> <!-- Resource jQuery -->
</body>
</html>
```



```
File Edit View Search Terminal Tabs Help
root@taxing:~/Desktop x root@server1:~ x root@server2:~ x root@server3:/usr/local/nginx... x root@server4:/usr/local/nginx... x

<li>
    
</li>
</ul>_<!-- cd-item-wrapper -->
</a>

<div class="cd-item-info">
    <b><a href="#0">帅酷的红霞</a></b>

    <em class="cd-price">199元（包邮）</em>
</div> <!-- cd-item-info -->
</li>

<li>
    <a href="">
        <ul class="cd-item-wrapper">
            <li class="selected">
                
            </li>

            <li class="move-right">
                
            </li>

            <li>
                
            </li>
        </ul> <!-- cd-item-wrapper -->
    </a>
</li>
</ul> <!-- cd-item-wrapper -->

root@server3:/usr/local/nginx/html [潮流男女二号店 - Mozilla Firefox] 1 / 4
```

### 6.3 过程中遇到的问题与解决办法

商城系统和高可用负载均衡系统单独实现都较为现实，但是在结合的时候会出现部分显示错误，导致的问题就是页面混乱，排版不一致。最终在将后端服务器减少并对 Nginx 服务器进行了相关参数修改，问题才得以解决。

## 第七章 高可用负载均衡集群系统的实现与测试

经过前几章的分析对比，已经确定好了系统需要的设计方案，在这一章内容中，主要对设计方案予以实现。

本次实现共有五台 Linux 服务器组成：一台宿主机+四台虚拟机。使用的是 LVS 的 DR 模式进行搭建，高可用是 Keepalived，服务器是 Nginx，用于展示的是自己开发的商城系统。

### 7.1 安装与配置集群系统

#### 7.1.1 构建基础镜像环境

构建基础环境镜像包括以下内容：

- ①关闭防火墙；
- ②配置好网络问题；
- ③关闭 selinux；
- ④设置好 IP 和网关。

以上这四点内容都是整个系统正常运行所必须的基础，所以在最小镜像建立之初就配置好，创建虚拟机之后就不需要再做冗余工作了。

#### 7.1.2 nginx 服务器的安装与配置

因为 Nginx 是开源软件，所以直接下载于 [www.nginx.org](http://www.nginx.org)。官网有安装文档，这里不再赘述，大概流程如下：

解压下载的压缩包；

- 1、进入到解压后的目录，`make && make install` 进行安装；
- 2、将生成的相关命令复制到常用位置，便于使用；
- 3、运行 `nginx` 命令，检查端口是否打开；
- 4、用宿主机访问该主机，检查是否会出现 `nginx` 测试页（测试页如下图

7.1）。



图 7.1 nginx 测试页

#### 7.1.3 负载均衡配置和测试

调度器的配置分为两部分：

- 1、调度器的配置：

## ①加载规则：

```
$ ipvsadm -C ##清空 IPVS 的转发表
$ ipvsadm -A (添加一个虚拟服务) -t (选择 tcp 服务) 172.25.24.100:80 -s
(设置转发策略) rr (轮询调度)
$ ipvsadm -a (添加一个后端服务) -t 172.25.24.100:80 -r (指定真实服务器) 172.25.24.3:80 -g (全局设置)
$ ipvsadm -a -t 172.25.24.100:80 -r 172.25.24.4:80 -g
```

## ②保存规则：

```
$ service ipvsadm save
```

## ③绑定 VIP：

```
$ ifconfig eth0:0 172.25.24.100 netmask 255.255.255.0 up
```

## 2、服务器（两台）的配置：

## ①修改内核参数：

```
$ vim /etc/sysctl.conf
net.ipv4.conf.lo.arp_ignore = 1
net.ipv4.conf.lo.arp_announce = 2
net.ipv4.conf.all.arp_ignore = 1
net.ipv4.conf.all.arp_announce = 2
```

这四个内核参数的意义就是只对来自调度器（VIP）的请求进行处理

```
$ sysctl -p ##保存设置
```

```
2, $ ifconfig lo:0 172.25.24.100 netmask 255.255.255.255 up
```

##将虚拟 IP 添加到本地 lo:0 接口

```
3, $ route add -host 172.25.24.100 dev lo:0
```

##添加一条对于 VIP：172.25.24.100 的路由规则

测试：

在两台后端服务器的 nginx 里面的测试页 index.html 填写相区别的信息，检测流量的分配情况。如图 7.2, 7.3 所示：



图 7.2 nginx 测试页



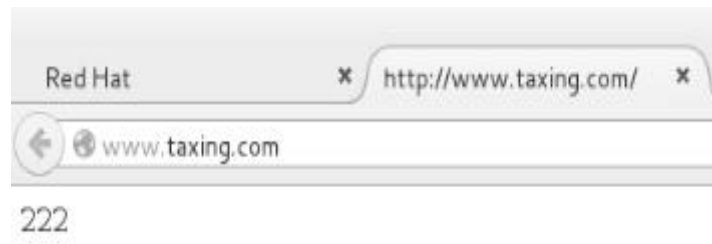


图 7.3 nginx 测试页

由此可见，www.taxing.com 的负载均衡搭建完成。

#### 7.1.4 高可用架构的配置

高可用系统是基于 keepalived 来搭建的，配置主要在两台调度机上完成。

解压安装好 keepalived 软件（命令不再展示），主要问题就是配置环节。keepalived 的配置需要在两台调度器上进行配置，以下是配置文件展示：

```
vrrp_instance VI_1 {
    state MASTER
    interface eth0
    virtual_router_id 51
    priority 100
    advert_int 1
    authentication {
        auth_type PASS
        auth_pass 1111
    }
    virtual_ipaddress {
        172.25.24.100
    }
}

virtual_server 172.25.24.100 80 {
    delay_loop 6
    lb_algo rr
    lb_kind DR
    persistence_timeout 0
    protocol TCP

    real_server 172.25.24.3 80 {
        weight 1
        TCP_CHECK {
            connect_timeout 10
            nb_get_retry 3
            delay_before_retry 3
```



```

virtual_server 172.25.24.100 80 {
    delay_loop 6
    lb_algo rr
    lb_kind DR
    persistence_timeout 0
    protocol TCP

    real_server 172.25.24.3 80 {
        weight 1
        TCP_CHECK {
            connect_timeout 10
            nb_get_retry 3
            delay_before_retry 3
        }
    }
    real_server 172.25.24.4 80 {
        weight 1
        TCP_CHECK {
            connect_timeout 10
            nb_get_retry 3
            delay_before_retry 3
        }
    }
}

```

以上是 master 主节点的配置，slave 备节点的配置只需要将 state 字段的 master 改为 slave 就行。

启动 keepalived，查看 IP，对外的虚拟 IP (VIP) 就出现了，如 7.4 所示：

```

File Edit View Search Terminal Tabs Help
root@taxing:~/Desktop x root@server1:~ x root@server2:~ x root@server3:~ x
[root@server1 ~]# ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 16436 qdisc noqueue state UNKNOWN
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        inet6 ::1/128 scope host
            valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP qlen 1000
    link/ether 52:54:00:a7:68:16 brd ff:ff:ff:ff:ff:ff
    inet 172.25.24.1/16 brd 172.25.255.255 scope global eth0
        inet 172.25.24.100/32 scope global eth0
            inet6 fe80::5054:ff:fea7:6816/64 scope link
                valid_lft forever preferred_lft forever
[root@server1 ~]#

```

图 7.4 虚拟 IP

利用 ipvsadm 命令就可以看到整个集群的信息，如图 7.5 所示：

```

File Edit View Search Terminal Tabs Help
root@taxing:~/Desktop x root@server1:~ x root@server2:~ x root@server3:~ x
[root@server1 ~]# ipvsadm -L
IP Virtual Server version 1.2.1 (size=4096)
Prot LocalAddress:Port Scheduler Flags
-> RemoteAddress:Port Forward Weight ActiveConn InActConn
TCP 172.25.24.100:http rr
-> server3.taxing.com:http Route 1 0 0
-> server4.taxing.com:http Route 1 0 0
[root@server1 ~]#

```

图 7.5 后端服务集群

### 7.1.5 真实服务器上的项目配置

项目配置是将前端代码以及相关文件放置到 nginx 的 html 目录下，便于 nginx 服务器读取。发布在 html，使用 nginx（命令为 `nginx -s reload`）重新读取发布内容。

### 7.1.6 集群配置常见的问题与解决办法

配置安装中最常见的命令就是遗忘了或者打错部分命令，导致的服务没有安装配置成功，需要在配置结束之后仔细校对，利用网络思维，检查出现的问题，并重新更正。

## 7.2 集群在模拟商城系统中的应用

整个集群的最终效果如图 7.6，7.7 下：



图 7.6 一号店铺商城系统



图 7.7 二号店铺商城系统

### 7.3 小结

本章中主要说明了安装配置的具体过程，展现了所有服务的配置文件和商城系统的代码，并对其进行了测试。配置过程是一个考验细心程度的过程，需要仔细检查每一步。包含了每个软件的解压安装以及服务部署，均衡的配置测试，高可用系统的配置及检查，还有整个商城系统的开发以及在高可用负载均衡系统中的应用。

## 结束语

随着社会的发展，技术的进步，互联网计算机已经成为了当前社会交流的必要组成部分。用户基数的增加，单个的服务器不再能够提供满意的服务，进而有了集群的产生。高可用负载均衡集群能够大幅度提高网络服务的健壮性，成为当下热门。

本毕设主要究于服务器实现负载均衡问题，运用虚拟服务 LVS 负载均衡技术和高可用 keepalived 技术，结合自己开发的商城系统，实现了一个整体系统架构。所有的服务都是在 Linux（RedHat6.5）下实现的，调度机和后端服务器都是在虚拟机上部署的，但麻雀虽小五脏俱全，该有的功能都实现了。

介于个人时间和精力有限，论文中提到的设计方案和部署配置过程都比较粗糙，尤其是在商城系统的部署上又很多不足。进一步的工作就是往这方面学习研究，争取早日解决这些问题。

本设计中的各种虚拟机都是独立的，对于服务的响应来说，还存在很大方面的问题。如果各部分的后端服务器是共享存储，那么响应前端用户请求的速度应该会提高不少，这对于大型的项目来说就比较客观了。当然，这都只是我的一些想法，能不能实现还需要进一步研究。

## 致 谢

感谢李朕老师悉心指导，从毕业设计系统的思路设计到最终论文的完成，都离不开李老师的帮助。在于李老师进行沟通请教问题的过程中，李老师总是细心指导，循循善诱，就好像朋友一般热情，他在学识上的严谨细致和认真负责的工作态度，将成为我毕业走上工作岗位之后的榜样。

感谢周围所有同学对我的帮助，在我碰到问题时，他们给予我解答，在论文出现问题时总能帮我解决。感谢四年来我的任课教师及指导我的老师，我的成长，离不开你们的帮助。感谢家人和同学朋友对我的支持鼓励，你们的态度是我最大的动力。

最后，感谢我的母校—西安邮电大学。四年时间，你见证了我的成长，给予了我的现在。怀着感恩之心，真诚的感谢老师、家人、朋友、同学，谢谢你们的一路陪伴。

## 参考文献

- [1] 章文嵩. LVS 项目介绍. <http://www.linuxvirtualserver.org/zh/LVS.html>, 2002 年 3 月.
- [2] Lawrence Berkeley. TCP/IP 详解(卷 1: 协议). 机械工业出版社, 2016 年 6 月.
- [3] 杨晔. 集群服务器的性能分析与研究. 西安电子科技大学, 2007 年 1 月.
- [4] LVS 手册. <http://www.austintek.com/LVS/LVS-HOWTO/HOWTO/index.html>.
- [5] 孟相武, 程劲, 罗克露, 韩淙. 基于 Linux 的高可用集群系统的设计及实现. 电子科技大学学, 2005 年 8 月.
- [6] 刘玉艳, 明玉. LVS 负载均衡技术在网络服务中的应用. 合肥工业大学学报, 2007 年 1 月.
- [7] 单康康, 王勇超, 郭晔. 基于 LVS+keepalived 的 DNS 集群研究, TP393. 0.
- [8] 邓越, 张静, 方俊, 基于 LVS 负载均衡在数字城市云环境中的应用, P203.
- [9] 王志刚, 常兴磊, 胥茜, LVS 集群的一种动态负载均衡方法, TP368. 5.
- [10] V. Fernández-Quiruelas, C. Blanco, A. S. Cofiño, J. Fernández, Large-scale climate simulations harnessing clusters, grid and cloud infrastructures, 2015.
- [11] Broadcom Corp, DIAN ZHOU, LVS Verification across Multiple Power Domains for Quad-Core Microprocessor, University of Texas at Dallas.
- [12] Lizhe Wang, Samee U. Khan, Dan Chen, Joanna Kołodziej, Rajiv Ranjan, Cheng-zhong Xu, Albert Zomaya, Energy-aware parallel task scheduling in a cluster, 2013.