**非凡之星技术解决方案**

|  |  |  |  |
| --- | --- | --- | --- |
| 编写 | 朱国进 | 编写时间 | 2016年01月04日 |
| 审批 | 李标 | 审批时间 | 2016年1月10 |
| 版本 | V1.0 | | |

**文档修订摘要**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **日期** | **修订号** | **描述** | **作者** | **审阅者** |
| 2016-01-04 | 1.0 | * 文档创建 | 朱国进 | 李标 |

**目 录**

第一章 引言.............................................................................................................................3

1.1 文档用途.....................................................................................................................3

1.2 阅读对象.....................................................................................................................3

第二章 技术概述........................................................................................................................3

2.1 技术介绍.....................................................................................................................3

2.2 系统构建标准化.........................................................................................................6

第三章 电子商务平台建设方案.................................................................................................6

3.1 平台设计原则..............................................................................................................6

3.2 电子商务总体架构方案..............................................................................................7

第四章 运行环境.........................................................................................................................8

4.1 服务器软硬件环境......................................................................................................8

4.2 客户端软硬件环境......................................................................................................8

第五章 安全保障建设.................................................................................................................9

5.1 服务分层隔离设计......................................................................................................9

5.2 数据安全设计..............................................................................................................9

5.3容灾设计.....................................................................................................................11

5.4服务器架构.................................................................................................................12

5.5其他安全机制.............................................................................................................12

## 第一章 引言

从公司长远发展来看,需要有一份有关技术的文档作为指导或参考，根据今后公司的发展和技术的发展将不断完善该技术文档。

#### 1.1 文档用途

作为公司技术方向选型参考。

#### 1.2 阅读对象

公司相关技术人员,包括:架构师、项目经理、java工程师、运维人员、Android工程师、IOS工程师、设计师、前端开发人员等。

## 第二章 技术概述

#### 2.1 技术介绍

* **Ajax:** Ajax是Web2.0阶段系列技术和相关产品服务中非常重要的一种技术。其全称为异步JavaScript和XML（即Asynchronous JavaScript and XML），从中可以看到与Ajax直接相关的几个技术点：一是异步、二是JavaScript、三是XML。这几点恰恰集中反映了这项技术关注的两个问 题：一是借助异步JavaScript实现浏览器和服务器之间的异步交互，如无需重新装载整个页面就可以向服务器发送请求，并接受响应。二是对XML文档 的解析和处理。
* **Jquery:** JQuery是继[prototype](http://baike.baidu.com/view/1217697.htm" \t "_blank)之后又一个优秀的[Javascript](http://baike.baidu.com/view/16168.htm" \t "_blank)库。它是轻量级的js库 ，它兼容[CSS3](http://baike.baidu.com/view/1713027.htm)，还兼容各种浏览器（[IE](http://baike.baidu.com/subview/703/10271308.htm" \t "_blank) 6.0+, [FF](http://baike.baidu.com/subview/31049/5089479.htm) 1.5+, [Safari](http://baike.baidu.com/subview/110484/5036395.htm) 2.0+, [Opera](http://baike.baidu.com/subview/10019/7098372.htm) 9.0+），jQuery2.0及后续版本将不再支持IE6/7/8浏览器。jQuery使用户能更方便地处理[HTML](http://baike.baidu.com/view/692.htm" \t "_blank)（[标准通用标记语言](http://baike.baidu.com/view/5286041.htm)下的一个应用）、events、实现动画效果，并且方便地为网站提供[AJAX](http://baike.baidu.com/subview/1641/5762264.htm" \t "_blank)交互。jQuery还有一个比较大的优势是，它的文档说明很全，而且各种应用也说得很详细，同时还有许多成熟的[插件](http://baike.baidu.com/view/18979.htm)可供选择。jQuery能够使用户的html页面保持代码和html内容分离，也就是说，不用再在html里面插入一堆js来调用命令了，只需要定义id即可。
* **Json:** Json（JavaScript Object Notation）是一种轻量级的数据交换格式。它基于ECMAScript的一个子集。 JSON采用完全独立于语言的文本格式，但是也使用了类似于C语言家族的习惯（包括C、C++、C#、Java、JavaScript、Perl、 Python等）。这些特性使JSON成为理想的数据交换语言。 易于人阅读和编写，同时也易于机器解析和生成（网络传输速率）。
* **Node.js:** Node.js是一个基于Chrome JavaScript运行时建立的平台， 用于方便地搭建响应速度快、易于扩展的网络应用。Node.js 使用[事件驱动](http://baike.baidu.com/view/536048.htm" \t "_blank)， 非阻塞[I/O](http://baike.baidu.com/subview/300881/11169495.htm" \t "_blank) 模型而得以轻量和高效，非常适合在分布式设备上运行的数据密集型的实时应用。
* **CSS3:** CSS3是[CSS](http://baike.baidu.com/subview/15916/5236733.htm" \t "_blank)技术的升级版本，CSS3语言开发是朝着模块化发展的。以前的规范作为一个模块实在是太庞大而且比较复杂，所以，把它分解为一些小的模块，更多新的模块也被加入进来。这些模块包括： 盒子模型、列表模块、[超链接](http://baike.baidu.com/view/743.htm)方式 、语言模块 、背景和边框 、文字特效 、多栏布局等。
* **Bootstrap:** Bootstrap，来自 Twitter，是目前最受欢迎的前端框架。Bootstrap 是基于 HTML、CSS、JAVASCRIPT 的，它简洁灵活，使得 Web 开发更加快捷。
* **PAAS:**Paas是Platform-as-a-Service的缩写，意思是平台即服务。把服务器平台作为一种服务提供的商业模式。通过网络进行程序提供的服务称之为Saas(Software as a Service),而云计算时代相应的服务器平台或者开发环境作为服务进行提供就成为了Paas(Platform as a Service)。
* **Spring MVC:** Spring Web MVC是一种基于Java的实现了Web MVC设计模式的请求驱动类型的轻量级Web框架，即使用了MVC架构模式的思想，将web层进行职责解耦，基于请求驱动指的就是使用请求-响应模型，框 架的目的就是帮助我们简化开发，Spring Web MVC也是要简化我们日常Web开发的。
* **Freemarker:** FreeMarker是一个用Java语言编写的模板引擎，它基于模板来生成文本输出。FreeMarker与Web容器无关，即在Web运行时，它并不知道Servlet或[HTTP](http://baike.baidu.com/view/9472.htm" \t "_blank)。它不仅可以用作表现层的实现技术，而且还可以用于生成XML，JSP或Java 等。
* **Ngix:** Nginx（发音同 engine x）是一款轻量级的Web 服务器／反向代理服务器及电子邮件（IMAP/POP3）代理服务器，并在一个BSD-like 协议下发行。由俄罗斯的程序设计师Igor Sysoev所开发，最初供俄国大型的入口网站及搜寻引擎Rambler（俄文：Рамблер）使用。 其特点是占有内存少，并发能力强，事实上nginx的并发能力确实在同类型的网页伺服器中表现较好.目前中国大陆使用nginx网站用户有：新浪、网易、 腾讯,另外知名的微网志Plurk也使用nginx。
* **RabbitMQ:** RabbitMQ是一个很受欢迎的消息中间件，通过它可以很方便地实现异构子系统之间的通讯，还可以将不同子系统之间进行解耦。它用erlang开发，基本上是实现了AMQP 1.0标准的消息协议。
* **Kafka:** Kafka是一种高吞吐量的[分布式](http://baike.baidu.com/view/402382.htm" \t "_blank)发布订阅消息系统，具有高性能和高吞吐率。
* **Redis**: Redis是一个开源的使用ANSI C语言编写、支持网络、可基于[内存](http://product.it168.com/list/b/0205_1.shtml" \t "_blank" \o "内存)亦可持久化的日志型、Key-Value数据库，并提供多种语言的API。
* **Solr:**Solr是一个独立的[企业级搜索](http://baike.baidu.com/view/4173560.htm" \t "_blank)应用服务器，它对外提供类似于Web-service的API接口。用户可以通过http请求，向搜索引擎服务器提交一定格式的XML文件，生成索引；也可以通过Http Get操作提出查找请求，并得到XML格式的返回结果。Solr是Lucene面向企业搜索应用的扩展。
* **Dubbo:** Dubbo是阿里巴巴公司开源的一个高性能优秀的服务框架，使得应用可通过高性能的 RPC 实现服务的输出和输入功能，可以和Spring框架无缝集成。
* **Thrift:**是一个软件框架，用来进行可扩展且跨语言的服务的开发。它结合了功能强大的软件堆栈和[代码生成](http://baike.baidu.com/view/8477718.htm)引 擎，以构建在 C++, Java, Python, PHP, Ruby, Erlang, Perl, Haskell, C#, Cocoa, JavaScript, Node.js, Smalltalk, and OCaml 这些编程语言间无缝结合的、高效的服务。
* **CDN:** CDN系统能够实时地根据网络流量和各节点的连接、负载状况以及到用户的距离和响应时间等综合信息将用户的请求重新导向离用户最近的服务节点上。其目的是使用户可就近取得所需内容，解决 Internet网络拥挤的状况，提高用户访问网站的响应速度。

对于大规模电子商务平台一般需要建CDN做网络加速，大型平台如淘宝、京东都采用自建CDN，中小型的企业可以采用第三方CDN厂商合作，如蓝汛、网宿、快网等。当然在选择CDN厂商时，需要考虑经营时间长短，是否有可扩充的带宽资源、灵活的流量和带宽选择、稳定的节点、性价比。

* **负载均衡:** 负载均衡是由多台服务器以对称的方式组成一个服务器集合，每台服务器都具有等价的地位，都可以单独对外提供服务而无须其他服务器的辅助。通过某种负载分担技术，将外部发送来的请求均匀分配到对称结构中的某一台服务器上，而接收到请求的服务器独立地回应客户的请求。均衡负载能够平均分配客户请求到服务器列阵，籍此提供快速获取重要数据，解决大量并发访问服务问题。这种群集技术可以用最少的投资获得接近于大型主机的性能。硬件采用F5，软件LVS。
* **反向代理:** 反向代理（Reverse Proxy）方式是指以[代理服务器](http://baike.baidu.com/view/751.htm" \t "_blank)来接受internet上的连接请求，然后将请求转发给内部网络上的服务器，并将从服务器上得到的结果返回给internet上请求连接的客户端，此时代理服务器对外就表现为一个反向代理服务器。
* **Zookeeper:** ZooKeeper是一个[分布式](http://baike.baidu.com/view/402382.htm" \t "_blank)的，开放源码的[分布式应用程序](http://baike.baidu.com/view/553502.htm)协调服务，是[Google](http://baike.baidu.com/view/105.htm)的Chubby一个[开源](http://baike.baidu.com/view/9664.htm" \t "_blank)的实现，是Hadoop和Hbase的重要组件。它是一个为分布式应用提供一致性服务的软件，提供的功能包括：配置维护、名字服务、分布式同步、组服务等。
* **Mycat:** 一个彻底开源的，面向企业应用开发的大数据库集群，结合传统数据库和新型分布式数据仓库的新一代企业级数据库产品，一个新颖的数据库中间件产品。
* **Tomcat:** Tomcat是Apache 软件基金会（Apache Software Foundation）的Jakarta 项目中的一个核心项目，由[Apache](http://baike.baidu.com/subview/28283/5418752.htm" \t "_blank)、Sun 和其他一些公司及个人共同开发而成。由于有了Sun 的参与和支持，最新的Servlet 和JSP 规范总是能在Tomcat 中得到体现，Tomcat 5支持最新的Servlet 2.4 和JSP 2.0 规范。因为Tomcat 技术先进、性能稳定，而且免费，因而深受Java 爱好者的喜爱并得到了部分软件开发商的认可，成为目前比较流行的Web 应用服务器。
* **Jboss:** 是一个基于J2EE的[开放源代码](http://baike.baidu.com/view/1708.htm" \t "_blank)的[应用服务器](http://baike.baidu.com/view/995472.htm)。 JBoss代码遵循LGPL许可，可以在任何商业应用中免费使用，而不用支付费用。JBoss是一个管理EJB的容器和服务器，支持EJB 1.1、EJB 2.0和EJB3的规范。但JBoss核心服务不包括支持servlet/JSP的WEB容器，一般与Tomcat或Jetty绑定使用。
* **MongoDB:** MongoDB是一个可扩展、高性能的分布式文档存储数据库，由C 语言编写，旨在为web应用提供可扩展的高性能数据存储解决方案。它的特点是高性能、易部署、易使用，存储数据非常方便。
* **HBase:** HBase是一个分布式的、面向列的开源数据库，该技术来源于 Fay Chang 所撰写的Google论文“Bigtable：一个结构化数据的[分布式存储系统](http://baike.baidu.com/view/1911305.htm" \t "_blank)”。 就像Bigtable利用了Google文件系统（File System）所提供的分布式数据存储一样，HBase在Hadoop之上提供了类似于Bigtable的能力。HBase是Apache的Hadoop 项目的子项目。HBase不同于一般的关系数据库，它是一个适合于非结构化数据存储的数据库。另一个不同的是HBase基于列的而不是基于行的模式。
* **MySql Server:** MySQL是一个[关系型数据库管理系统](http://baike.baidu.com/view/1450387.htm" \t "_blank)，由瑞典 MySQL AB 公司开发，目前属于 [Oracle](http://baike.baidu.com/view/15020.htm) 旗下公司。MySQL 最流行的[关系型数据库管理系统](http://baike.baidu.com/view/1450387.htm" \t "_blank)，在 WEB 应用方面 MySQL 是最好的 RDBMS (Relational Database Management System，关系数据库管理系统) 应用软件之一。MySQL 是一种关联[数据库管理系统](http://baike.baidu.com/view/68446.htm" \t "_blank)， 关联数据库将数据保存在不同的表中，而不是将所有数据放在一个大仓库内，这样就增加了速度并提高了灵活性。MySQL 所使用的 SQL 语言是用于访问数据库的最常用标准化语言。MySQL 软件采用了双授权政策（本词条“授权政策”），它分为社区版和商业版，由于其体积小、速度快、总体拥有成本低，尤其是[开放源码](http://baike.baidu.com/view/394804.htm)这一特点，一般中小型网站的开发都选择 MySQL 作为网站数据库。
* **Oracle:** Oracle Database，又名Oracle RDBMS，或简称Oracle。是[甲骨文公司](http://baike.baidu.com/view/22424.htm" \t "_blank)的一款[关系数据库管理系统](http://baike.baidu.com/view/687948.htm)。它是在数据库领域一直处于领先地位的产品。可以说Oracle数据库系统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。
* **雅虎优化网站规则:**请遵照执行。

#### 2.2 系统构建标准化

* 计算机软件工程规范国家标准汇编GB/T16398-96
* GB/T11475-95软件工程术语
* GB/T15538-95软件标分类法
* GB/T12504-90计算机软件质量保证计划规范
* GB/T14394-93计算机软件可靠性和可维护性
* GB/T15853-95软件支持环境

## 第三章 电子商务平台建设方案

#### 3.1 平台设计原则

为了实现网站商务功能最大化的目标，给目标客户提供方便、实用的信息服务，在设计该平台时，我们充分考虑了以下方面：

**可靠性:** 采用稳定可靠的成熟技术，保证系统长期安全运行。一方面要强调整个系统高可靠性，确保在意 外情况故障或重负载情况下系统的稳定性，另一方面要保证在业务高峰期间的系统响应能力。

**先进性:** 以最先进的观点和设计思路，为客户设计高先进性的网站系统。设计方案将立足先进技术，使项 目具备国内乃至国际领先的水平。服务器和网络方面以优化通讯流量，提高系统的管理性和安全 性为重点。

**可持续性:** 该平台正常运作后，由于面对的是广泛的全球互联网客户，因此系统应能够提供每天24小时， 每周7天的不间断运作能力。为客户提供高度可持续增长的稳定运行保障。

**安全性:** 互联网是一个标准开放的网络，在网上进行各种商务活动，随时可能将面对黑客的攻击，病毒的 侵袭等。因此，确保网上信息流通的系统安全十分重要。安全不仅仅是一个技术的问题，还涉及 到系统的管理、法律法规的保障等。我们会做到保障系统数据和信息安全，为业务及商务提供安 全环境。

**可扩展性:** 互联网具有巨大的商务潜能，没有人可以确切预计系统的最终访问量和最佳的商务运行模式。 因此系统设计的原则之一是可扩展性。随着企业网上平台业务量的扩展和平台访问量的增长, 系统应该能够具有很强的扩展能力，以适应新业务的发展。

**标准性和开放性:** 所有程序及接口具有统一的标准，使程序和系统具备优异的可移植性。企业网上平台的 设计应当严格遵守国际标准，在还没有形成标准的新领域内也积极倡导标准的形成。

**美观性：**良好的视觉效果强大的功能同等重要。它可以突出企业文化特色和定位。

**服务性：**时刻体现着以客户为中心的服务思想，以为客户提供最好的服务为网站的设计思路。

**实用性：**以满足业务需求为首要目标,网站所提供的各项信息、服务等内容要做到实用，真正能够使网站为 用户带来方便。

**便捷性：** 网站使用起来简单方便，尽可能地适应不同年龄、知识层次的群众的需求。

**定制性：** 为企业的不同业务定制相应的业务流程。

**交互性：** 各项在线服务的提供，以及友好的对话关系，极大的改善企业的办事效率和形象。

**宣传性：** 提供良好的实用的宣传功能，使之成为企业自身宣传的重要载体。

#### 3.2 电子商务总体架构方案



**3-1 电子商务总体架构**

整个架构是分层的分布式的架构，纵向包括CDN，负载均衡/反向代理，应用服务器，web应用，基础服务，数据存储层。水平方向包括对整个平台的配置管理部署和监控。

## 第四章 运行环境

#### 4.1 服务器软硬件环境

|  |  |
| --- | --- |
| **需求名称** | **环境要求** |
| 硬件最低配置 | CPU:XEON2.0G四核(以上) 内存:(4G以上)、网卡:100/1000兆 |
| 操作系统 | CentOS6.5以上 |
| 应用服务器 | Apache Tomcat7.0、Nginx1.2.6 |
| 负载均衡 | LVS(Linux Virtual Server) |
| 数据库 | Mysql 5.0 |

#### 4.2 客户端软硬件环境

|  |  |
| --- | --- |
| **需求名称** | **环境要求** |
| 硬件最低配置 | CPU:Intel i5系列3.30G 内存:(4G以上)、网卡:100兆 |
| 操作系统 | Windows系列 |
| 应用服务器 | Apache Tomcat7.0 |
| 浏览器 | IE8+,FireFox3.0以上版本,Chrome,360v3.0以上版本 |
| 最小屏幕区域 | 1024\*768 |

## 第五章 安全保障建设

#### 5.1 服务分层隔离设计



**5-1 服务器分离**

使用linux系统的iptables内核级防火墙，可以根据需要禁止某些或开发某些端口的访问。

在架构中，将各种服务器按照安全级别从高到低划分如下:

* Web服务器集群
* 安全级别最低，可由外部直接通过SSH(Secure Shell)协议登陆
* 向所有IP打开HTTP,HTTPS,SSH,FTP等端口。
* 应用服务器及资源服务器集群
* 安区级别中等，不可直接由外部登陆。
* 只向Web服务器集群打开相应的服务端口(由具体应用决定且可配)
* (分布式)数据库集群和(分布式)文件系统集群
* 安全级别最高，不可直接由外部登录
* 只向应用服务器及资源服务器集群打开绝对必要的服务端口

#### 5.2 数据安全设计

数据在项目中处于核心地位，数据分为两大类即:结构化数据和非结构化数据。结构化数据一般指的是存储在数据库系统的数据，这里可以是Oracle或者Mysql等关系型数据库，非结构化数据指的是如图片、视频、文件等非结构化资源。

从图()我们可以看到数据库和文件系统的安全级别是最高的。这里涉及到两个概念---集群和分布式。

项目数据安全设计可以分为两个阶段，根据项目运营的需求进行调整。

* 第一阶段----集群

集群，是项目初期和中期采用的数据安全设计。这种方式，在项目初期，没有高并发量和大数据的前提下。可以保证数据的安全和完整性。

构化数据以Mysql为例常用的集群技术，目前最常用的方式就是设置主-从(master-slave)库，并实现读写分离，保障数据安全的同时，提高了数据操纵的效率。如图(5-2),在这里我们可以保证，当一台数据库服务器宕机的时候，那项目还能正常运行，而不会有丝毫影响。



**图5-2 结构化数据集群**

非结构化数据，对于电子商务项目来说视频资源是相对较少的，如有也只是产品介绍flash或者宣传短片，主要是图片类资源。在这里有一种简易的方法--采用RAID做多硬盘数据同步。

但是这种方法，只能在硬盘损坏的情况下保证项目正常运行，如果服务器其他部件损坏则无法正常运行。在项目运作初期，特别是B2C项目，如果没有太多SNS交互功能，那一般非结构化资源不会太多。所以在项目初期可以采用以Nginx或者lighttpd为服务器(两者在静态资源的处理上优于Apache Server或IIS)，下面以Nginx为例(更稳定)。如图5-3所示



**5-3 Nginx负载均衡示意图**

* 第二阶段---分布式集群

当项目运营发展到一定阶段，用户并发数和数据量暴增到已经达到第一阶段架构的瓶颈。我们应该采用分布式集群对架构进行拓展，以保障系统的性能以及项目的正常运行，当然这还包括web前端的安全性和性能问题，这个在服务器架构上会详细描述。

集群大家可以理解为，一组服务器集中在一起实现相同的业务。而分布式的每个节点都处理不同的业务，每个分布式的节点都可以是一个集群。从中大家可以看出分布式和集群的关系，分布式是集群的扩展。采用分布式可以解决集群的性能瓶颈问题，资源瓶颈问题，因为它是可以无限扩展的。

对于结构化数据，在项目第二阶段还是采用Mysql数据库的分布式集群方式来存储。因为首先Mysql是免费开源的，大大降低项目运营的成本；其次Mysql是跨平台的，与MS SQL相比大大提高了拓展性；再次Mysql轻便耗资源少，这个是Oracle无法比拟的。我们采用Mycat作为Mysql分布式数据库的中间件。如图5-4所示，解决数据量大的情况下，数据查询效率的问题，以及服务器资源不足的问题。MySql Server可以无限拓展，每个Mysql Server都可以是如图5-4所示的主从库。



**5-4 MySql数据库分布式存储**

#### 5.3容灾设计

所谓容灾，即当发生灾难的时候如:地震、海啸等，淹没了主机房的时候，项目还能正常运行，或者受影响不大。在这里一样根据项目运营的不同阶段做两个阶段的容灾方案。

* 第一阶段，可采用定时异地容灾机制

项目初期，为了防止数据丢失，可以通过定期对数据进行备份包括:结构化数据、非结构化数据。

结构化数据:在线数据库服务器(DB)每日定时运行脚本做整库自备份，生成数据库备份文件，并将备份文件上传到备份数据库(BackUp DB)。这两台服务器每日定时运行脚本将最新的数据库备份文件导入到自身的数据库。

根据上传资源的用户角色不同，非结构化数据又分为:a)用户上传数据资源备份:定期(如30分钟)从DB服务器的media meta信息表中获取最新上传数据资源的列表，将此列表中的每个数据资源从Media服务器拷贝到在线Web服务器(Web1,Web2)、备份Web服务器(BackUp Web),以此达到备份效果。b)静态资源备份:每次在部署新版本时，运行部署脚本将最新的静态资源拷贝到在线Web服务器(Web1,Web2)、备份Web服务器(BackUp Web)，从而进行备份。

* 第二阶段，可采用第三方异地容灾机制

第二阶段，由于项目运营已经进入常态化，用户量和用户数据日趋重要，每时每刻都可能有用户进行交易，定时每分已经无法满足需求，这个时候实时异地备份成为必然趋势。

#### 5.4服务器架构

综合考虑成本和效率服务器在系统中主要分为5大块包括Web服务器、数据库服务器、文件服务器、应用服务器和资源服务器。

* Web服务器，主要指Nginx、Apach Server、IIS服务器。
* 数据库服务器，主要存放数据库和数据仓库等结构化数据。
* 文件服务器，主要存放的是图片、文件、视频等非结构化数据。
* 应用服务器，主要是部署应用的服务器，如：Tomcat、Jboss等。
* 资源服务器，主要放js、css等的服务器。

在Web服务器主要采用LVS作为负载均衡中间件,4台服务器(web1、web2、web-lb1、web-lb2)用于实现web服务器的负载均衡(需5个IP)。web1、web2均为系统的web服务器，web-lb1、web-lb2均为web服务器的负载均衡节点。

web-lb1、web-lb2均为web服务器的负载均衡节点，用于调用web1、web2服务器，其中web-lb1服务器处于工作状态，当web-lb1出现故障或死机时将由web-lb2服务器接管所有的web-lb1的所有工作直到web-lb1服务器恢复正常，保证用户的正常访问。

web1、web2均为网站web服务器，其中负载量较小的将被web-lb1(或web-lb2)首先调用以达到web1、web2的负载均衡，当web1或web2出现故障或死机时将由相应的另一方接管所有工作，保证用户的正常访问。

当在线用户增加导致项目访问缓慢的情况的时候，可以增加负载均衡节点来实现分流，解决访问缓慢的问题。可以进行无限扩容。

#### 5.5其他安全机制

* 安全登录制
* 禁用root帐号
* 使用SSH密钥登录方式登录，密钥采用2048位的DSA密码。该方法被业界证明为最安全的登录方式(该加密算法被美国国防部等高保密机构使用)
* 用户访问权限控制
* 将随时更新系统补丁，以避免系统漏洞
* 服务器将尽量少开服务,只服务于必要项目,这样降低由其他服务带来的安全隐患.
* 用户密码保护,我们将保持sha256位的hash密码，而不是原始密码。
* 开启128位SSL的https访问，使得关键数据加密传输
* 在服务器端审核所有用户操作，以防止非法用户操作
* 采用用户密码强度验证，禁止使用简单的不安全的密码
* 用户密钥一旦丢失将立即作废，防止帐号盗用
* 每天对数据库进行热备份，以防止数据库崩库或非法篡改。备份过程通过SSH安全隧道进行。
* 进行代码级安全校验，防止SQL注入。