



---

# WEB 中小型企业网络架构

---



撰写：王瑞

审核：孟亚鹏

成员：彭顺品

刘田华

马建华

蒋璐

## 目录

一、 需求环境分析 .....	2
二、 设计目标 .....	3
三、 项目介绍 .....	3
四、 运行环境 .....	3
五、 架构设计 .....	4
六、 模块划分 .....	4
七、 部署方案 .....	8

## 一、 需求环境分析

随着网络应用的丰富和发展，很多网站往往不能迅速跟进大量信息衍生及业务模式变革的脚步，常常需要花费许多时间、人力和物力来处理信息更新和维护工作；遇到网站扩充的时候，整合内外网及分支网站的工作就变得更加复杂，甚至还需重新建设网站；如此下去，用户始终在一个高成本、低效率的循环中升级、整合。对于网站建设和信息发布人员来说，他们最关注的系统的易用性和的功能的完善性，因此，这对网站建设和信息发布工具提出了一个很高的要求。

为此，我们想出了一个方案，来有效解决用户网站建设与信息发布中常见的

问题和需求。

## 二、设计目标

在 Linux 系统中搭建一个高并发、高可用的运行环境，实现服务稳定、数据安全、高并发、易扩展、便管理的基本功能，从而有效解决用户网站建设与信息发布中常见的问题和需求，最终实现一个高并发高可用性的中小型企业级 Web 网站架构，架构中包括了负载均衡，热备，以及监控等技术。

## 三、项目介绍

本项目设计一个适用于中小型企业的网络架构解决方案，案列为搭建一个 Web 论坛，使用 LVS+Keepalived 实现高可用，采用 MooseFS 分布式文件系统作为网站文件的存储，数据库集群使用 MyCat 中间件实现 MySQL 数据库的读写分离其中读库 2 台写库 1 台，Web 集群则使用 Nginx+Squid 实现访问静态页面的高并发，如果非静态页面则走 Apache 执行动态 PHP 脚本。

## 四、运行环境/人员分配

CentOS 7 /RHEL 7 64bit 17 台，Zabbix 监控 1 台

内网中所有主机关闭防火墙和 SELinux 外部使用专业防火墙防护。

内网网段：192.168.1.0/24 VIP 地址：192.168.1.10

彭顺平：主要负责两台 Nginx 服务的配置以及一台 Squid 配置缓存服务，三台 Apache 服务并配置好 PHP 执行环境。

刘田华：主要负责数据库初始数据的导入任务，并配置三台数据库主从从，以及 MyCat 读写分离。

马建华：主要负责 MooseFS 分布式文件系统的配置，主要任务包括一台主 MFS 一台日志 MFS 服务器，以及后端的实际存储节点的部署任务。

蒋璐：主要负责最后的两台 LVS 的 DR 模式配置，以及配置好两台服务器的故障切换功能。

## 五、架构设计

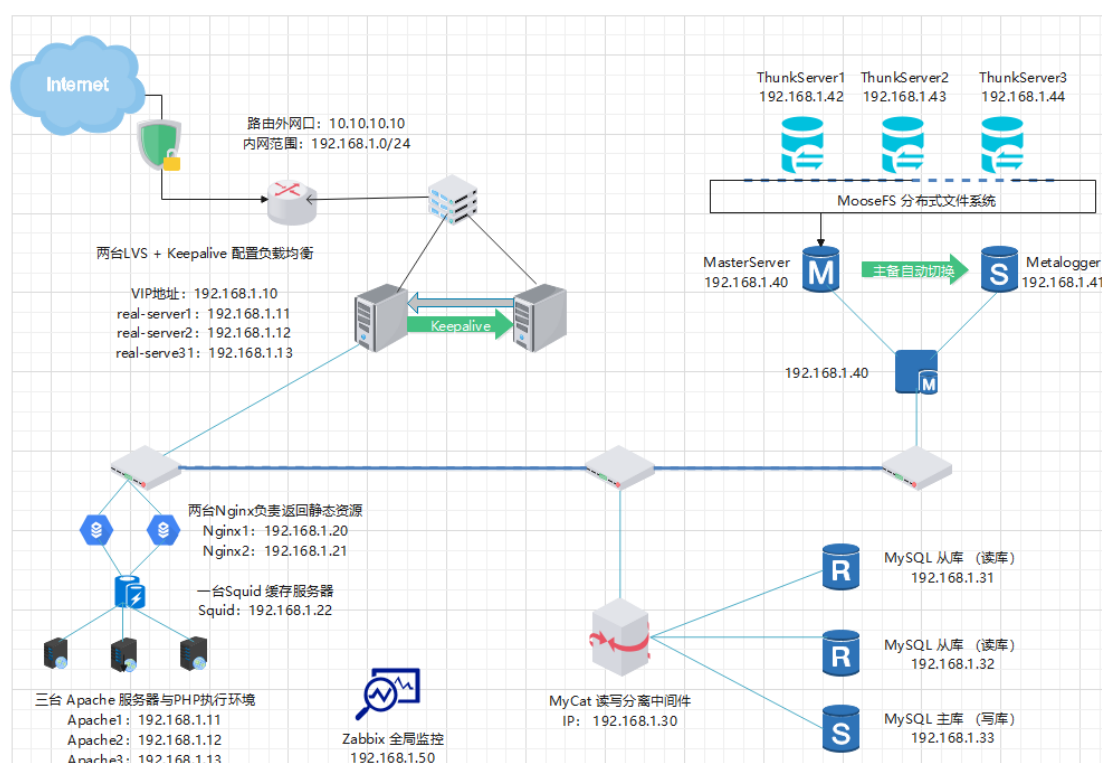


图 1 架构设计图

## 六、模块划分

### 6.1、模块介绍

**LVS 负载均衡：**对外提供给用户接入，提供一个虚拟 ip 进行前端系统的负载均衡，只用直接路由技术实现虚拟服务器。实现服务器之间的负载调度，减轻服务器压

力，提高利用率。

**高可用 HA (High Availability)** 指的是通过尽量缩短日常维护操作和突发的系统崩溃（非计划）所导致的停机时间，以提高系统和应用的可用性。

**Web 服务：**Apache/Nginx ， Nginx 在前端，实现静态页面的一些缓存和处理，Apache 提供动态页面的访问。

**MySQL 数据库：**按照数据结构来组织、存储和管理数据，并通过 MyCat 实现读写分离，分担使用数据库的压力。

**MFS 分布式文件系统：**用户访问到的网站网页，网站服务器统一使用存储服务器进行挂载，实现高可靠性 、可动态扩展 、可回收的文件共享。

## 6.2、Web 集群

Web 集群前端使用 LVS+Keepalived 实现高可用，后面使用 2 台 Nginx 服务器承受高并发的访问，为了加快静态文件的访问速度，在 Nginx 后端加入一台 Squid 作为缓存服务器，同时可降低后端 Apache 的访问压力，最后面使用 3 台 Apache 服务器，上面安装 PHP+Apache 用于提供动态网页文件的解析和访问请求。

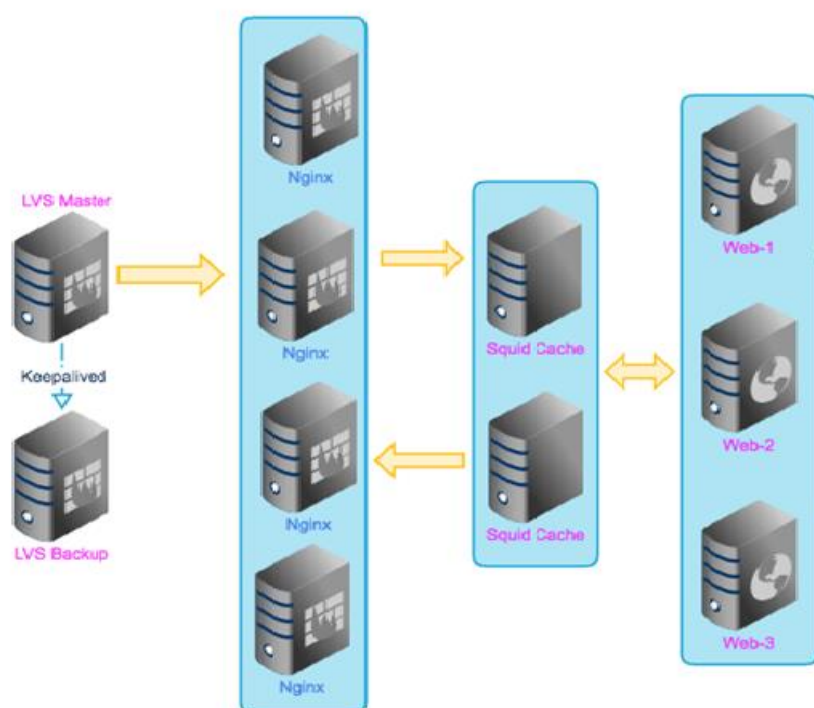


图 2 Web 集群

### 6.3、数据库集群

Amoeba 是 MySQL 的分布式数据库前端代理，在应用层访问 MySQL 的时候充当 SQL 路由功能，处于应用层和 MySQL 之间，对客户端是透明的。

数据集群采用 Amoeba 中间件实现 MySQL 数据库的读写分离，降低 MySQL 服务器的磁盘 I/O，实现数据访问的高并发。

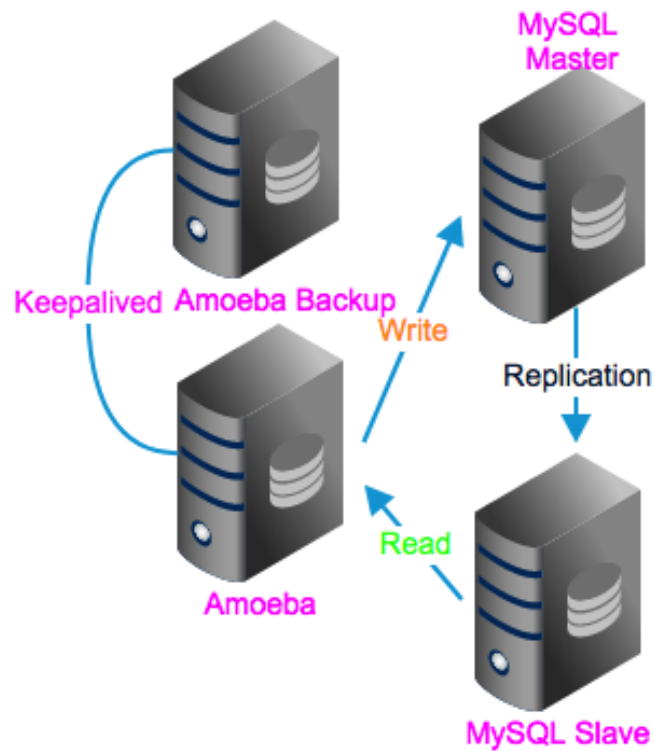


图 3 数据库集群

## 6.4、存储集群

分布式文件系统就是把一些分散在多台计算机上的共享文件夹，集合到一个共享文件夹内，用户要访问这些文件夹的时候，只要打开一个文件夹，就可以的看到所有链接到此文件夹内的共享文件夹，它把数据分散存放在多个物理服务器上，而呈现给用户的则是一个统一的资源。

存储集群采用 MFS 分布式文件系统作为网页文件的存储解决方案，当有数据丢失时，能及时恢复；分布式的文件系统加快了网页的访问速度。

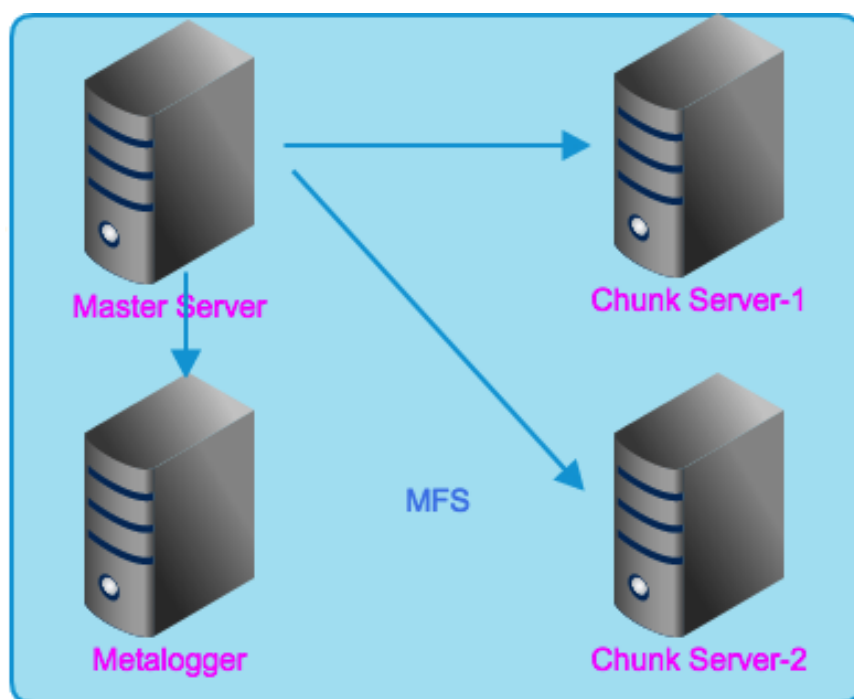


图 4 MFS

## 七、部署方案

### 7.1、Web 服务器 Nginx + Apache 部署

相对于 apache 来说，Nginx 占用更少的内存及资源，并且还能支持更高的并发连接，具有更高的效率。Nginx 静态处理性能比 Apache 高出 3 倍以上，但是对动态页面 PHP 的支持则远不如 Apache。

在部署时，将 Nginx 置于前端，作为一个分发器，将静态请求留给 Nginx 自己处理，而对于动态请求则发送给后端的 Apache 处理。利用 Nginx 的高性能和 Apache 的高稳定性来提供更好 Web 服务。

### 7.2、LVS 部署（DR 模式 + Keepalived）

LVS 即 Linux Virtual Server——Linux 虚拟服务器。利用集群技术将多台同构或者异构的 Linux 服务器结合起来，协同完成某些特定任务。

LVS 集群采用三层结构：负载调度器、服务器池、共享存储。所有的用户都通过一个虚拟的 IP（VIP）来访问 WEB 服务器集群。负载调度器是整个集群的前



端，负责将用户的请求通过轮询算法发送到服务器池的节点主机上。调度器具有很好的吞吐率，将用户请求均衡地发送到不同的节点主机上，由节点主机处理用户的请求。并且能自动屏蔽掉发生故障的节点主机，从而将一组服务器构成一个高性能、高可用的虚拟服务器。

负载调度器在分发大量的用户请求时，难免会出现故障。为了在负载调度器发生故障的情况下仍然能对用户提供服务，可以准备一台备用的负载调度器，通过 keepalived 实现两台负载调度器的切换。

### 7.3、MFS 分布式文件系统部署

MooseFS 是一个具备冗余容错功能的分布式网络文件系统，它将数据分别存放在多个物理服务器或单独磁盘或分区上，确保一份数据有多个备份副本，然而对于访问 MFS 的客户端或者用户来说，整个分布式网络文件系统集群看起来就像一个资源一样，从其对文件系统的情况看 MooseFS 就相当于 UNIX 的文件系统。具有高可靠性、高可扩展性、高可容错性、高数据一致性等优点，部署使用一台服务器作为主服务器，存储元数据信息，一台作为日志服务器，两台 chunk 服务器。

### 7.4、MySQL 的主从、读写分离部署

当网站达到一定规模时，就需要大量地在数据中读取和写入数据，通过主从备份来保存、备份数据，通过读写分离实现负载均衡。

通过一台装有 MyCat 的服务器作为中间件，实现在主从上的读写分离。考虑到一般情况下，对数据库的读取操作要高于写入操作，所以让从服务器来提供读取，主服务器提供写入操作，分配上 2 读 1 写。

## 7.5、Zabbix 监控服务器部署

Zabbix 是一个基于 Web 界面的提供分布式系统监视以及网络监视功能的企业级的开源解决方案, 使用一台 Zabbix 服务器监控主要的节点, 以便在出现问题时能及时处理。