

基于集群与镜像技术的学习平台部署架构改造

王晓¹, 唐洪鹏¹, 潘淑芬²

(1. 青岛远洋船员职业学院网络教育学院, 山东 青岛 266071; 2. 青岛远洋船员职业学院职业培训学院, 山东 青岛 266071)

摘 要: 基于视频课程的学习平台用户体验差的问题是一个困扰运维人员的技术难点。本文采用集群与镜像技术对某学习平台进行部署架构改造: 通过 Nginx 代理服务器做负载均衡的 Session 保持, 通过 Redis 技术实现 Session 保存与共享, 解决了集群环境下学习平台 Session 丢失的问题; 通过镜像技术和学习平台 Web 服务器、FTP 课程服务器与学员用户设备 IP 地址绑定, 实现了视频课程稳定输出, 解决了学习平台用户体验差、视频播放不流畅的问题。

关键词: Session 保持与共享; 镜像; 负载均衡; IP 地址 hash 绑定

中图分类号: TP311

文献标识码: A

基于视频课程的学习平台用户体验不佳问题是困扰运维人员的一个技术难点。用户体验不佳, 在学员端主要表现为课程播放不流畅、视频长时间加载不上, 从而导致学员无法顺利学习课程。由于事前没有参与学习平台代码开发, 平台运维人员很难对平台代码进行修改, 所以不能从程序层面进行优化, 只能从部署架构上进行改进。本文基于某学习平台改进实践, 探讨基于集群与镜像技术改造学习平台技术架构策略。

1 某学习平台存在问题分析

图 1 是某学习平台原有部署架构图。运维人员通过对该学习平台部署架构进行研究发现, 其视频访问瓶颈主要有两个: 一是数据传输过程中, 需要频繁进行内外网地址转换; 二是原有部署架构为单机模式。在基于单机模式部署架构中, 每台服务器的职能彼此独立, 没有统筹协调机制, 不能统一调度服务器资源对外提供服务。随着访问平台学员数量的增加, 带来平台访问并发量的倍增, 导致单机模式下运行的学习平台资源崩溃。

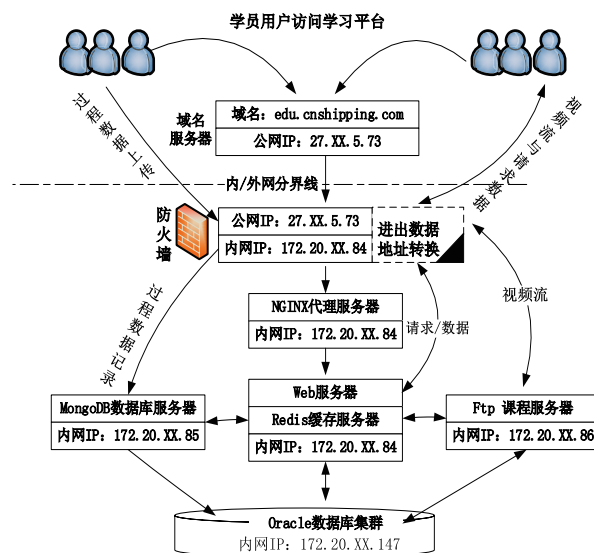


图 1: 单机模式学习平台网络部署架构图

2 平台改进方案

针对平台存在的问题, 运维人员对平台网络架构与部署架构进行了技术改造 (图 2)。一是采用集群与镜像技术, 解决了服务器资源彼此独立、互不协作问题; 二是通过配置公网地址与防火墙策略, 避免了数据传输过程中频繁地进行地址转换的问题。

收稿日期: 2021—08—02

第一作者简介: 王晓 (1971—), 男, 副教授

基金项目: 2019 年青岛远洋船员职业学院立项: “基于 IPV6 的数字化校园升级改造” (项目编号: 2019—R—009)

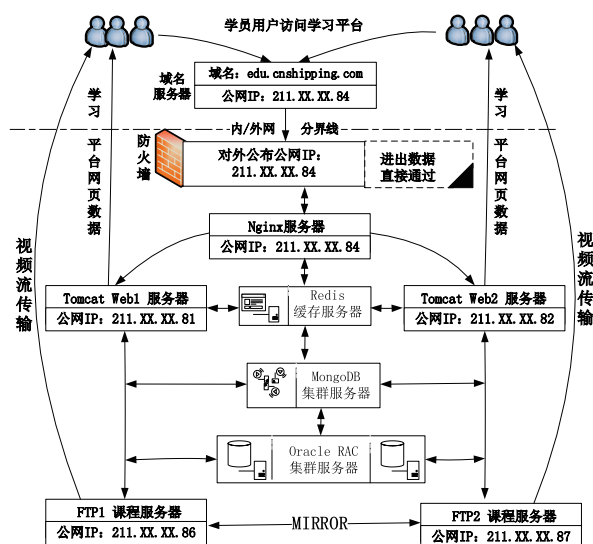


图 2：基于集群与镜像技术学习平台网络部署架构图

从图 2 可以看出，改进的学习平台 Web 服务器是集群，由 Tomcat Web1 与 Tomcat Web2 两台服务器组成；课程服务器是镜像，由 FTP1、FTP2 两台课程服务器组成。这种部署方案具有负载均衡、资源共享、视频分流、效能优化等优点，实现了服务器间分工与互助协作的关系。

采用集群与镜像技术改进学习平台的部署架构，需要解决两个技术问题：一个是 Session 丢失问题，一个是课程视频输出不稳定问题。通常 Web 服务器创建的 Session 只能保存在自己的服务器上，为用户访问提供服务的 Session 会话也只能由这个 Web 服务器提供。当 Web 服务器采用集群以后，用户请求响应从一台 Web 服务器转移到另一台 Web 服务器的时候，Web 服务器对应的 Session 并不能跟随请求响应的转移而转移，从而导致原 Web 服务器 Session 丢失。Session 丢失在学员端表现为登录状态或课程播放被迫中断，学员需要重新登录学习平台，重新建立视频流传输通道，从头开始学习过程。

视频课程输出不稳定问题表现在两个方面：一是原学习平台完整的视频课程路由 IP 地址 + 端口号 + 文件夹名 + 文件名组成。由于镜像服务器 IP 地址不同，所以不同课程服务器上的课程路径就不同，从而导致了镜像服务器之间提供视频播放服务的错乱。二是学习平台数据库中保存的课程路径不包含 IP 地址与端口号，完整的课程路径的 IP 地址与端口号由 Web 服务器从运行环境中获取，并以前缀的方式添加到取自数据库的路径信息中，形成完整的课程播放路径。在集

群状态下，从运行环境中获取 IP 地址与端口号，存在较大的随机性，因而导致视频输出过程中发生视频服务器改变的情况。

3 Session 丢失问题解决方案

3.1 学习平台 Session 的工作原理

Session 是同 Cookie 相互配合进行工作的。Cookie 是保存在客户浏览器中的用户访问信息。它是学习平台登录成功后，由 Web 服务器签发的、允许用户自动登录学习平台的一个令牌——Session ID。Session 是服务器端的一块内存区域，存放着客户端浏览器窗口的编号。当浏览器访问某个页面时，会在服务器端开辟一块内存，用来关联浏览器的窗口或者子窗口。这块内存就是 Session。Session 保存了用户的访问信息，是 Cookie 的集合。学员第一次访问学习平台时，Web 服务器生成 Session ID，并保存到用户的 Cookie 中。以后用户通过浏览器访问学习平台时，浏览器会把保存的 Cookie 信息发送给 Web 服务器，Web 服务器会在服务器端查找与之匹配的 Session。查找成功，就自动登录，并从前一次保存的运行状态接着运行；查找不成功，就重新登录，平台程序会将新输入的用户名、密码等信息与数据库中的信息进行比较，开启一个新的认证过程，并生成新的 Session 与 Cookie。

3.2 Session 丢失解决方案

解决 Session 丢失的方法是采用 Session 保持与共享。学习平台 Web 服务器集群是通过 Nginx 请求转发与负载均衡以及 Redis 缓存技术实现 Session 保持与共享的（如图 3）。

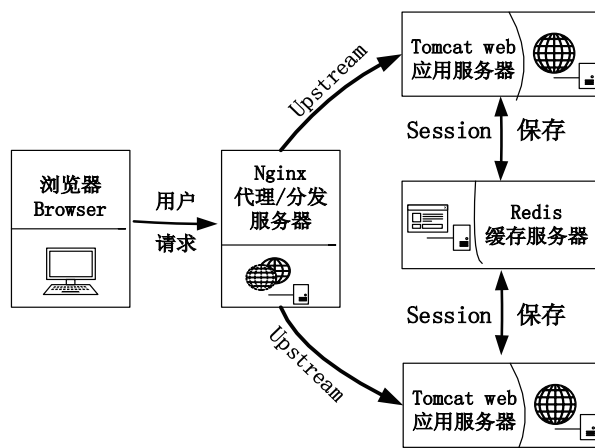


图 3：session 保持与共享

3.2.1 通过 Nginx 代理服务器做负载均衡的 Session 保持

图 3 中 Nginx 服务器有两个作用: 请求转发与负载均衡。所有学员对学习平台的访问请求都不是直接提交给具体业务服务器处理。访问请求首先被 Nginx 服务器拦截, 由 Nginx 服务器对访问请求进行转发, 以 Upstream 方式分配给绑定的 Web 服务器进行处理。这一过程叫请求转发。Web 服务器之间根据承受的压力与约定好的规则, 将客户请求提交给效率最优的 Web 服务器处理。这一过程叫做负载均衡。

具体操作如下:

首先配置 Nginx 服务器, 以 Upstream 方式为每个用户请求分配一个访问 IP_hash 结果。这样每一个学员用户就与学习平台的一台 Web 服务器相绑定, 达到了 Session 保持的结果。

以下脚本是 Nginx 代理服务器通过 Nginx.conf 配置文件, 以 Upstream 方式为 2 台 Web 服务器配置的 Session 保持。

```
Upstream Elearning {  
    Ip_hash;  
    server 211.XX.XX.82:8081;  
    server 211.XX.XX.83:8081;  
}
```

当学员通过集群的一台 Web 服务器首次访问学习平台时, Nginx 会将学员设备的 IP 地址与第一次提供服务的 Web 服务器进行绑定; 以后再通过这个 IP 地址访问学习平台, 用户请求都会通过这个绑定的 Web 服务器进行响应。这一过程也即 Session 保持。只有这台被绑定的 Web 服务器出现故障, 不能提供请求响应时, 学员的请求才会转移给集群中的其他 Web 服务器进行处理。由此可见, 学习平台的 Session 保持是通过学员用户与 Web 服务器地址绑定实现的。

负载均衡的 Session 保持避免了为学员提供服务的 Web 服务器在集群服务器间频繁切换。同一个用户的 Session 可以一直保持在一台 Web 服务器中, 从而实现 Web 服务的稳定输出。

3.2.2 通过 Redis 技术实现 Session 保存与共享

通过 Redis 技术实现用户登录信息——Session ID 与整个学习平台的绑定, 解决学员用户加载新页面或者页面刷新超时需要重新认证的问题。

Redis 的一个重要作用是充当缓存服务器。

通过 Redis 可把最新或者频繁访问的学习平台内容保存到缓存服务器上。当 Web 服务器给客户端提供请求服务时, 会首先查询 Redis 缓存服务器。如果 Redis 缓存服务器中存在要访问的内容, Web 会直接从缓存中读取数据提交给用户; 如果缓存服务器中找不到需要的内容, Web 就会到磁盘上去查找所要的内容。由于从缓存服务器上读写数据要比从磁盘中读写数据速度快得多, 所以采用 Redis 缓存服务器后, 能够极大地提高学习平台的访问效率。

Redis 的另一个重要作用是为集群内不同 Web 服务器提供 Session 共享。当集群中的一个 Web 服务器出现请求中断时, Redis 会将该 Web 服务器的 Session 保存到 Redis 缓存服务器中; 同时, 把这台 Web 服务器提供的请求响应提交给集群内的其他 Web 服务器接管; 接管请求的 Web 服务器会根据请求传递的 Session ID 到 Redis 缓存服务器中查找与请求匹配的 Session, 并将查找到的 Session 内容提交给学习平台应用程序, 由应用程序接着请求中断时的运行状态继续执行。

Redis 配置 Session 共享需要做两方面的工作:

一是在 Redis 服务器端进行参数设置。Redis.conf 文件部分脚本修改如下:

```
bind 127.0.0.1 (Redis 服务器环路 IP 地址)  
bind 211.87.160.XX (web1IP 地址)  
bind 211.87.160.XX (web2IP 地址)
```

二是在集群 Web 服务器端配置 Tomcat-redis-session 的共享依赖。具体操作与部分代码如下:

1) 引入 Redis 运行依赖的类文件 RedisHttpSessionConfiguration

```
<bean id="redisHttpSessionConfiguration"  
class="org.springframework.session.data.redis.config.  
annotation.web.http.RedisHttpSessionConfiguration">  
    <property name="maxInactiveIntervalInSeconds" value="${redis.maxInactiveIntervalInSeconds}"/>  
</bean>
```

2) 配置 Redis 服务器的主机 hostName、端口 port、密码 password 与链接数据库 database 参数:
<bean id="jedisConnectionFactory"


```
class="org.springframework.data.redis.
connection.jedis.JedisConnectionFactory" destroy-
method="destroy">
```

```
<property name="hostName" value="{redis.
host}"/>
```

```
<property name="port" value="{redis.port}"/>
```

```
<property name="password" value="{redis.
pass}"/>
```

```
<property name="database" value="{redis.
default.db}"/>
```

```
</bean>
```

3) 在学习平台程序的 zhy.properties 文件里面配置访问 Redis 的参数值。

Redis.host=211.87.XX.84 //Redis 服务器 IP 地址

Redis.port=6379 //Redis 服务器端口号

Redis.pass=myredispassword // 链接 Redis 密码

Redis.default.db=172.20.XX.84 // 链接 Redis 数据库

通过以上技术改造, 解决了学习平台的 Session 共享问题。

4 视频课程稳定输出问题解决方案

针对前述两个方面的视频课程稳定输出问题, 我们采取视频课程服务器镜像技术和访问线路上的 FTP 课程服务器、Web 服务器与学员访问设备 IP 地址绑定两种措施予以解决。

4.1 课程服务器镜像技术可行性

从视频播放过程看, 学员观看视频, 首先通过学习平台应用程序到数据库中提取到视频所在的位置信息, 然后根据这个位置信息找到课程视频, 再通过流媒体播放器把视频播放到学员的设备上。

从学员的角度来说, 课程视频是从服务器单向流向学员的, 因此课件服务器对于学员而言只有读的操作——即学员只是通过平台看视频, 而不需要为平台上传视频或者加工视频。这就为学员通过不同的 Web 服务器访问不同的 FTP 课程服务器提供了可能。

要实现不同的学员通过不同的 FTP 课程服务器学习同一个视频课程, 需要具备一个条件: 按照数据库提供的课程路径, 在每一台 FTP 课程服务器上都能找到这个课程视频。也就是说要保

证两台 FTP 课程服务器上的课程路径相同、内容相同, 亦即提供视频课程的两台 FTP 服务器彼此互为镜像。

4.2 利用镜像技术解决视频服务错乱问题方案

解决视频服务错乱问题首先要解决除 IP 地址外, 镜像服务器的课程路径相同、课程内容相同问题。这一问题是通过两台 FTP 课程服务器的镜像技术实现的, 两个 FTP 课程服务器做 mirror, 实现互为镜像。

4.2.1 配置镜像服务器的 ssh 互信

配置服务器间的 ssh 互信, 是为了实现服务器之间免密码登陆。配置服务器之间 ssh 互信以后, 服务器之间传输数据就可以不必进行登录验证了。

图 4 是 FTP 课程服务器间配置 ssh 互信原理图。两台服务器在各自服务器中存放对方服务器的证书, 当执行登陆时, 自动完成认证, 从而不需要再次进行身份认证或者输入密码。

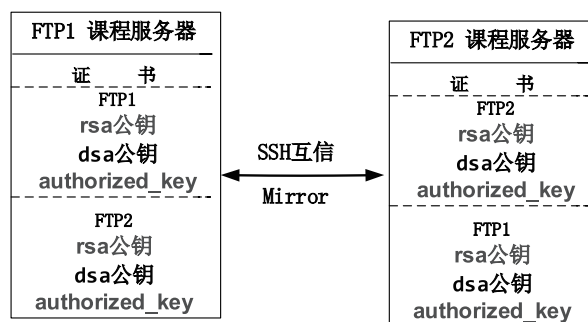


图 4: FTP 课程服务器配置 ssh 互信原理图

具体配置过程如下:

1) 登陆镜像服务器 FTP1 与 FTP2, 在两个镜像服务器上分别执行下面操作;

2) 生成 .ssh 目录 (命令: mkdir -p ~/.ssh);

3) 设置 ssh 的权限 (命令: chmod -R 700 ~/.ssh);

4) 生成 rsa 公钥文件 id_rsa.pub (命令: ssh-keygen -t rsa);

5) 生成 dsa 公钥文件 id_dsa.pub (命令: ssh-keygen -t dsa);

6) 将两台 FTP 课程服务器的 .ssh 目录下的四个公钥文件追加到 FTP1 服务器的认证文件 authorized_keys 中:

首先,把 FTP1 服务器的公钥文件追加到 FTP1 的认证文件 authorized_keys 中:

```
(命令: cat ~/.ssh/id_rsa.pub >> ~/.ssh/authorized_keys cat ~/.ssh/id_dsa.pub >> ~/.ssh/authorized_keys)
```

其次,把 FTP2 服务器的公钥文件追加到 FTP1 的认证文件 authorized_keys 中:

```
(命令: ssh FTP2 cat ~/.ssh/id_rsa.pub >> ~/.ssh/authorized_keys ssh ftp2 cat ~/.ssh/id_dsa.pub >> ~/.ssh/authorized_keys)
```

7)FTP1 服务器的认证文件 authorized_keys 复制到 FTP2 服务器上:

```
(命令: scp -r authorized_keys FTP2:~/.ssh)
```

8) 分别设置两个服务器上认证文件的权限:

```
(命令: chmod 600 ~/.ssh/authorized_keys)
```

通过以上操作,完成镜像服务器之间的互信。

4.2.2 配置镜像服务器之间的实时同步

镜像服务器之间需要保持内容完全一致,两边 FTP 服务器执行的上传、删除、修改等操作需同步进行。因此,需要在镜像服务器之间安装 Rsync 触发式同步更新程序,用于在镜像服务器之间实时更新对应的镜像服务器内容。

1) 在两台镜像服务器上分别安装 Rsync 同步程序和 Xinetd 超级守护进程。(命令: yum -y install Xinetd Rsync)

2) 修改镜像服务器的 Rsync 服务配置文件,在 /etc/Rsyncd.conf 文件中添加如下内容:

```
#Global Settings
```

```
hosts allow =211.XX.XX.86 211.XX.XX.87 //
```

两台镜像服务器的 IP 地址

```
[kejian]
```

```
path = /data/kejian
```

read only = no // read only = no 表示允许读写操作

```
auth users = ftpuser
```

3) 创建同步目录: (命令: mkdir -p /data/kejian)

4) 编写 Rsync 脚本,实现镜像服务器数据实时同步: vim /var/kejian/rsync.sh

```
#!/bin/bash
```

```
src=/data/kejian/
```

```
des=/data/kejian/
```

```
host1=211.XX.XX.86 // 镜像服务器 FTP1IP
```

地址

```
host2=211.XX.XX.87 // 镜像服务器 FTP2IP
```

地址

```
user=ftpuser
```

```
/usr/local/bin/inotifywait -mrq --timefmt '%d/%m/%y %H:%M' --format '%T %w %f' -e modify,delete,create,attrib $src | while read file DATE TIME DIR;
```

```
do
```

```
/usr/bin/rsync -vzrtopg --delete --progress $src $user@$host1:$des
```

```
echo "${files} was rsynced" >> /var/log/rsync.log 2>&1
```

```
done
```

通过以上具体配置,完成镜像服务器之间的数据实时同步。

4.3 解决 Web 服务器、课程服务器与学员用户地址绑定问题

通过 Nginx 代理服务器做负载均衡的 Session 保持,实现了用户 IP 地址与集群中一台 Web 服务器绑定。同时,为避免给学员用户提供视频服务的 FTP 服务器中途发生改变,还需要完成学习平台 Web 服务器与 FTP 课程服务器之间的绑定(图 5)。绑定后,通过 Web1 服务器访问学习平台的学员,只能通过 Web1 绑定的课程服务器 FTP1 获取课程视频;通过 Web2 访问学习平台的学员,只能通过 Web2 服务器绑定的课程服务器 FTP2 获取课程视频。

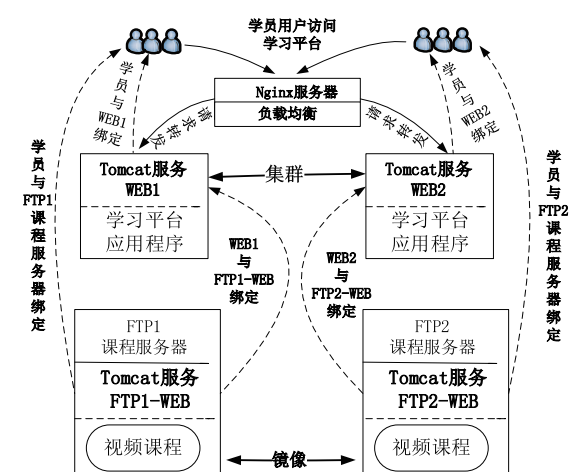


图 5: FTP 课程服务器、平台 web 服务器与学员地址绑定架构图

为了实现 Web 服务器与 FTP 课程服务器的绑定,需要在视频课程服务器上安装 Tomcat web 服务(图 5),称为 FTP-Web,用于专门对接学习平台 Web 应用程序服务器的视频请求。具体操作如下:

1) 在 FTP1、FTP2 服务器上分别安装 Tomcat 服务,命名为 FTP1-Web、FTP2-Web;

2) 配置 Web1 服务器与 FTP1-Web 绑定, Web2 服务器与 FTP2-Web 绑定;

3) 配置 FTP1-Web 接受 Web1 的请求与课程路径;配置 FTP2-Web 接受 Web2 的请求与课程路径;

4) FTP1-Web、FTP2-Web 按照课程路径,分别在各自服务器上查找课程视频;

5) FTP1-Web 将视频播放给 Web1 绑定的学员;FTP2-Web 将视频播放给 Web2 绑定的学员。

通过以上操作,一方面实现了视频课程播放分流,另一方面实现了视频流的稳定输出。

5 测试结果

通过对学习平台“外科手术式的改造”,实现了学习平台从单机模式向集群与镜像模式的技术转变,解决了视频课程播放卡顿、长时间加载不上的技术问题。经在上海、广州、海南、天津、大连等不同地域测试,课程加载时延明显变短,

视频播放清晰、流畅,用户体验明显改善;同时测试学习平台手机 APP(安卓、苹果)课程播放效果也非常好。对学习平台采用联通、电信、移动、教育网等不同网络进行视频传输测试,视频播放效果都非常流畅,完全达到了改善用户体验的效果。

参考文献:

[1] 刘卓,张向利.基于 Nginx 的负载均衡集群设计与实现[J].桂林电子科技大学学报.2017,(06): 490-493.

[2] 曾玮,王丹.基于 Spring Session 及 Redis 的高校 Web 服务器会话保持技术研究[J].实验室科学.2019,22(02): 68-75.

[3] CSDN 中国软件开发者联盟. linux ssh 互信配置——服务器间免密码登录 [EB/OL].
https://blog.csdn.net/chenfeng_sky/article/details/108058370,2020 年 8 月 17 日.

[4] CSDN 中国软件开发者联盟. Nginx 深入详解之 upstream 分配方式 [EB/OL].
https://blog.csdn.net/daily886/article/details/84313917?utm_medium=distribute.pc_relevant.none-task-blog-2-default-baidujs_baidulandingword-default-0.control&spm=1001.2101.3001.4242,2018 年 11 月 21 日:

Transformation of Learning Platform Deployment Architecture Based on Cluster and Mirroring Technology

WANG Xiao¹, TANG Hong—peng¹, PAN Shu—fen²

(1.Online Education School, 2.Vocational Training School, Qingdao Ocean Shipping Mariners College, Qingdao 266071, China)

Abstract: The problem of poor user experience in the learning platform based on video course is a technical difficulty that disturbs the operation and maintenance personnel. In this paper, cluster and mirroring technologies are adopted to transform the deployment architecture of the learning platform. Session retention is performed by Nginx proxy server for load balancing and Session storage and sharing is realized by Redis technology, which solves the problem of Session loss of the learning platform in the cluster environment. By binding the learning platform WEB server, FTP course server and the student user device IP address, the stable output of the video course is realized. It solves the problems of poor user experience and uneven video playback on the learning platform.

Keywords: session retention and sharing, mirroring technique, load balancing, IP address hash binding