

第 10 章

Nginx 在国内知名网站中的应用案例

Nginx 在国内知名网站中的应用案例主要可分为三类：Nginx 反向代理与负载均衡类网站应用案例、Nginx+PHP/JSP 类网站应用案例、Nginx 静态内容 Web 服务器应用案例。下面，我们找了一些截至 2009 年 9 月，使用 Nginx 作为 Web 服务器的网站代表：

1. 使用 Nginx 运行 PHP (FastCGI)、JSP 程序的网站

新浪播客 (<http://v.sina.com.cn/>) : nginx/0.7.62 + PHP

金山逍遥网 (<http://www.xoyo.com/>) : nginx/0.8.15 + PHP 5.2.9

金山爱词霸 (<http://www.iciba.com/>) : nginx/0.6.32 + PHP

六间房视频 (<http://www.6.cn/>) : nginx/0.7.21 + PHP

Discuz!官方论坛 (<http://www.discuz.net/>) : nginx/0.7.59 + PHP/5.2.10

赶集网 (<http://www.ganji.com/>) : nginx/0.7.62 + PHP/5.2.8

搜狐通行证 (<http://passport.sohu.com/>) : nginx/0.6.37 + JSP

网易博客 (<http://blog.163.com/>) : nginx/0.7.59 + JSP

人人网 (原校内网) (<http://renren.com/>) : nginx/0.6.32 + JSP

2. 使用 Nginx 作反向代理、规则过滤的网站

新浪播客接口服务器 (<http://interface.video.sina.com.cn/>) : nginx/0.5.35

新浪博客 (<http://blog.sina.com.cn/>) : nginx/0.7.62

YUPOO 相册 (<http://www.yupoo.com/>) : nginx/0.5.35

金山游戏用户中心 (<http://my.xoyo.com/>) : nginx/0.8.15

豆瓣 (<http://www.douban.com/>) : nginx

3. 使用 Nginx 运行静态 HTML 页、图片、FLV 视频的网站

网易新闻 (<http://news.163.com/>) : nginx/0.6.36

酷 6 网 (<http://www.ku6.com/>) 视频服务器: nginx/0.5.36

迅雷看看 (<http://www.xunlei.com/>) : nginx/0.6.31

新华网 RSS 订阅频道 (<http://rss.xinhuanet.com/>) : nginx

腾讯网 (<http://www.qq.com/>) : nginx/0.6.39

开心网图片服务器 (<http://img1.kaixin001.com.cn/>) : nginx

无论是新浪、搜狐、网易、腾讯等门户网站，还是目前红红火火的 Web 2.0 网站，都已经使用 Nginx 作为某些项目的 Web 服务器案例。Nginx 大有取代 Apache 之势。下面，我们就来结合实际案例，介绍一下 Nginx 在国内知名网站中的应用。

10.1 Nginx 反向代理与负载均衡类网站应用案例

10.1.1 Nginx 负载均衡在新浪播客中的应用

2008 年的新浪播客 (v.sina.com.cn、you.video.sina.com.cn) 由静态服务器集群和动态服务器集群两部分组成，分别采用不同的域名。静态服务器集群即我们在浏览器地址栏经常能看到 v.sina.com.cn 和 you.video.sina.com.cn 域名，采用 Squid 做前端缓存，服务器分布在全国各地机房。动态服务器集群采用 interface.video.sina.com.cn 域名，也称接口服务器，主要用来实时显示播放数、记录播放日志、为 Flash 视频播放器提供数据、与新浪内部产品、外部合作产品交互。

接口服务器最上层采用 F5 BIG-IP 硬件四/七层负载均衡交换机，对 4 台 Nginx 反向代理服务进行四层负载均衡，由这 4 台 Nginx 服务器判断 URL，进行分组，对后端的 3 组 Web 服务器进行七层负载均衡。F5 BIG-IP 也支持七层负载均衡，但是，由于一对 F5 BIG-IP 要服务新浪的多个产品，七层交换需要耗费 F5 BIG-IP 不少的 CPU 资源，而 F5 BIG-IP 四层交换有专门的硬件芯片来处理，耗费的资源较少，所以，在新浪，F5 BIG-IP 一般只用四层负载均衡。

F5 BIG-IP 后端的 3 组 Web 服务器，配置不一样，第 1 组为内存密集型服务器，技术类型主要是 PHP+Memcached 服务；第 2 组为 CPU 密集型服务器，主要耗费的是 CPU 资源；第 3 组为磁盘密集型服务器，为记录日志等操作，要求磁盘空间大、磁盘转速高。

代码 10-1 是新浪播客接口服务器的 Nginx 负载均衡配置，提供按 URL 分组服务、负载均衡服务：

代码 10-1

```
user www www;

worker_processes 10;

error_log /data1/logs/nginx_error.log crit;

pid /tmp/nginx.pid;

worker_rlimit_nofile 51200;

events
{
    use epoll;

    worker_connections 51200;
}

http
{
    include conf/mime.types;
    default_type application/octet-stream;

    charset gb2312;

    server_names_hash_bucket_size 128;
    .
    keepalive_timeout 15;

    sendfile on;
    tcp_nopush on;
    tcp_nodelay on;

    #第 1 组接口机: Memcache 相关 (点击数)
    upstream count.interface.video.sina.com.cn {
        server xx.xx.xx.55:80;
        server xx.xx.xx.58:80;
        server xx.xx.xx.47:80;
    }

    #第 2 组接口机: 外部提供数据类程序
    upstream data.interface.video.sina.com.cn {
```

```

server xx.xx.xx.59:80;
server xx.xx.xx.64:80;
server xx.xx.xx.48:80;
}

#第3组接口机: 打日志类程序、功能相关、嵌套页面
upstream log.interface.video.sina.com.cn {
    server xx.xx.xx.72:80;
    server xx.xx.xx.49:80;
}

server
{
    listen 80;
    server_name interface.video.sina.com.cn;

    location / {
        proxy_redirect off;

        #后端的Web服务器可以通过X-Forwarded-For获取用户真实IP
        proxy_set_header X-Forwarded-For $remote_addr;

        #按URL进行分组, 第1组: Memcache 相关 (点击数)
        if ($request_uri ~ "^/app/count/")
        {
            proxy_pass http://count.interface.video.sina.com.cn;
        }
        if ($request_uri ~ "^/app/online/")
        {
            proxy_pass http://count.interface.video.sina.com.cn;
        }
        if ($request_uri ~ "^/interface/user/getLoginGap.php")
        {
            proxy_pass http://count.interface.video.sina.com.cn;
        }

        #按URL进行分组, 第2组: 外部提供数据类程序
        if ($request_uri ~ "^/crossdomain.xml")
        {
            proxy_pass http://data.interface.video.sina.com.cn;
        }
        if ($request_uri ~ "/interface/client/topVideoClient.php")
        {
            proxy_pass http://data.interface.video.sina.com.cn;
        }
        if ($request_uri ~ "^/interface/common/")
        {
            proxy_pass http://data.interface.video.sina.com.cn;
        }
        if ($request_uri ~ "^/interface/randplay/randplay.php")
        {
            proxy_pass http://data.interface.video.sina.com.cn;
        }
    }
}

```

```

    }
    if ($request_uri ~ "^\/interface\/topic\/suggTopic.php")
    {
        proxy_pass http://data.interface.video.sina.com.cn;
    }
    if ($request_uri ~ "^\/interface\/uploadClient\/")
    {
        proxy_pass http://data.interface.video.sina.com.cn;
    }
    if ($request_uri ~ "^\/interface\/xml\/")
    {
        proxy_pass http://data.interface.video.sina.com.cn;
    }
    if ($request_uri ~ "^\/outinterface\/")
    {
        proxy_pass http://data.interface.video.sina.com.cn;
    }

    #按 URL 进行分组, 第 3 组: 打日志类程序
    if ($request_uri ~ "^\/interface\/flash\/")
    {
        proxy_pass http://log.interface.video.sina.com.cn;
    }
    if ($request_uri ~ "^\/interface\/playrank\/playrank2008_10.php")
    {
        proxy_pass http://log.interface.video.sina.com.cn;
    }

    #按 URL 进行分组, 其他组: 功能相关、嵌套页面等未匹配到的 URL
    proxy_pass      http://log.interface.video.sina.com.cn;
}

#定义日志格式
log_format count '$remote_addr - $remote_user [$time_local] $request '
                '$status' $body_bytes_sent "$http_referer" '
                '$http_user_agent' "$http_x_forwarded_for";

#打日志
access_log /data/logs/interface.log count;

#允许客户端请求的最大的单个文件字节数
client_max_body_size 10m;

#缓冲区代理缓冲用户端请求的最大字节数 可以理解为先保存到本地再传给用户
client_body_buffer_size 128k;

#跟后端服务器连接的超时时间_发起握手等候响应超时时间
proxy_connect_timeout 600;

#连接成功后_等候后端服务器响应时间_其实已经进入后端的排队等候处理
proxy_read_timeout 600;

```

```

#后端服务器数据回传时间_就是在规定时间内后端服务器必须传完所有的数据
proxy_send_timeout      600;

#代理请求缓存区_这个缓存区间会保存用户的头信息以供Nginx进行规则处理_一般只要能保存
下头信息即可
proxy_buffer_size        8k;

#同上 告诉Nginx保存单个用的几个Buffer 最大用多大空间
proxy_buffers             4 32k;

#如果系统很忙的时候可以申请更大的proxy_buffers 官方推荐*2
proxy_busy_buffers_size  64k;

#proxy缓存临时文件的大小
proxy_temp_file_write_size 64k;
}
}

```

10.1.2 Nginx 负载均衡在金山逍遥网中的应用

在金山逍遥网（www.xoyo.com）中，前端的负载均衡服务器采用的是Nginx，两台Nginx服务器为一组，承担多种类型的负载均衡服务，两台负载均衡服务器都处于活动状态，各自绑定一个公网虚拟IP，作为负载均衡服务器，当其中一台服务器发生故障时，另一台服务器接管发生故障服务器的虚拟IP。这种方式的详细介绍请见第6章。

代码 10-2 是 Nginx 负载均衡在金山逍遥网中的配置（nginx.conf）：

代码 10-2

```

user www www;

worker_processes 8;

error_log /data/logs/nginx_error.log crit;

pid /usr/local/webserver/nginx/nginx.pid;

#Specifies the value for maximum file descriptors that can be opened by this process.
worker_rlimit_nofile 51200;

events
{
    use epoll;
    worker_connections 51200;
}

http
{

```

```
include mime.types;
default_type application/octet-stream;

#charset utf-8;

server_names_hash_bucket_size 128;
client_header_buffer_size 32k;
large_client_header_buffers 4 32k;

sendfile on;
#tcp_nopush on;

keepalive_timeout 30;

tcp_nodelay on;

fastcgi_connect_timeout 300;
fastcgi_send_timeout 300;
fastcgi_read_timeout 300;
fastcgi_buffer_size 64k;
fastcgi_buffers 4 64k;
fastcgi_busy_buffers_size 128k;
fastcgi_temp_file_write_size 128k;

gzip on;
gzip_min_length 1k;
gzip_buffers 4 16k;
gzip_http_version 1.1;
gzip_comp_level 2;
gzip_types text/plain application/x-javascript text/css application/xml;
gzip_vary on;

limit_zone anti_attack $binary_remote_addr 10m;

#允许客户端请求的最大的单个文件字节数
client_max_body_size 300m;

#缓冲区代理缓冲用户端请求的最大字节数 可以理解为先保存到本地再传给用户
client_body_buffer_size 128k;

#跟后端服务器连接的超时时间_发起握手等候响应超时时间
proxy_connect_timeout 600;

#连接成功后_等候后端服务器响应时间_其实已经进入后端的排队之中等候处理
proxy_read_timeout 600;

#后端服务器数据回传时间_就是在规定时间内后端服务器必须传完所有的数据
proxy_send_timeout 600;

#代理请求缓存区_这个缓存区间会保存用户的头信息以供Nginx进行规则处理_一般只要能保存下头信息即可
proxy_buffer_size 16k;
```



#同上 告诉 Nginx 保存单个用的几个 Buffer 及最大用多大空间

```
proxy_buffers 4 32k;
```

#如果系统很忙的时候可以申请更大的 proxy_buffers 官方推荐*2

```
proxy_busy_buffers_size 64k;
```

#proxy 缓存临时文件的大小

```
proxy_temp_file_write_size 64k;
```

#缓存

```
proxy_temp_path /data2/proxy_temp_path;
```

```
proxy_cache_path /data2/proxy_cache_path levels=1:2 keys_zone=cache_one:2000m  
inactive=1d max_size=5m;
```

```
upstream my_server_pool {
```

```
server xx.xx.xx.1:80 weight=1 max_fails=2 fail_timeout=30s;
```

```
server xx.xx.xx.2:80 weight=1 max_fails=2 fail_timeout=30s;
```

```
server xx.xx.xx.3:80 weight=1 max_fails=2 fail_timeout=30s;
```

```
}
```

```
upstream php_server_pool {
```

```
server xx.xx.xx.4:80 weight=1 max_fails=2 fail_timeout=30s;
```

```
server xx.xx.xx.5:80 weight=1 max_fails=2 fail_timeout=30s;
```

```
server xx.xx.xx.6:80 weight=1 max_fails=2 fail_timeout=30s;
```

```
server xx.xx.xx.7:80 weight=1 max_fails=2 fail_timeout=30s;
```

```
server xx.xx.xx.8:80 weight=1 max_fails=2 fail_timeout=30s;
```

```
}
```

```
upstream bbs_server_pool {
```

```
ip_hash;
```

```
server xx.xx.xx.9:80 max_fails=2 fail_timeout=30s;
```

```
server xx.xx.xx.10:80 max_fails=2 fail_timeout=30s;
```

```
server xx.xx.xx.11:80 max_fails=2 fail_timeout=30s;
```

```
server xx.xx.xx.12:80 max_fails=2 fail_timeout=30s;
```

```
}
```

```
upstream cms_server_pool {
```

```
server xx.xx.xx.13:80 weight=1 max_fails=2 fail_timeout=30s;
```

```
server xx.xx.xx.14:80 weight=1 max_fails=2 fail_timeout=30s;
```

```
}
```

```
upstream pic_server_pool {
```

```
server xx.xx.xx.15:80 weight=1 max_fails=2 fail_timeout=30s;
```

```
server xx.xx.xx.16:80 weight=1 max_fails=2 fail_timeout=30s;
```

```
}
```

```
upstream xoyohimsg_server_pool {
```

```
server xx.xx.xx.17:3245;
```

```
server xx.xx.xx.18:3245 down;
```

```
}
```

```
#xoyo.com 域名跳转到 www.xoyo.com
```



```
server
{
    listen      80;
    server_name xoyo.com;

    rewrite ^/(.*) http://www.xoyo.com/ permanent;

    access_log /data1/logs/xoyo.com_access.log;
}

#用户中心 HTTPS/SSL 加密浏览
server
{
    listen      443;
    server_name my.xoyo.com;

    ssl on;
    ssl_certificate my.xoyo.com.crt;
    ssl_certificate_key my.xoyo.com.key;

    location /
    {
        proxy_pass http://php_server_pool;
        proxy_set_header Host my.xoyo.com;
        proxy_set_header X-Forwarded-For $remote_addr;
    }

    access_log /data1/logs/my.xoyo.com_access.log;
}

#图片服务器，不同的路径访问后端不同的服务器
server
{
    listen      80;
    server_name pic.xoyo.com;

    location /cms/
    {
        proxy_pass http://cms_server_pool;
        proxy_set_header Host pic.xoyo.com;
        proxy_set_header X-Forwarded-For $remote_addr;
    }

    location /
    {
        proxy_pass http://pic_server_pool;
        proxy_set_header Host pic.xoyo.com;
        proxy_set_header X-Forwarded-For $remote_addr;
    }

    access_log /data1/logs/pic.xoyo.com_access.log;
}
```

```
#音频电台文件下载, 进行简单防盗链
#limit_zone media $binary_remote_addr 10m;
server
{
    listen 80;
    server_name media.xoyo.com;

    location /
    {
        proxy_pass http://cms_server_pool;
        proxy_set_header Host $host;
        proxy_set_header X-Forwarded-For $remote_addr;

        valid_referers none blocked www.xoyo.com *.xoyo.com www.kingsoft.com
            *.kingsoft.com www.kingsoft.cn *.kingsoft.cn;
        if ($invalid_referer) {
            rewrite ^/ http://www.xoyo.com;
        }
    }

    access_log /data1/logs/media.xoyo.com_access.log;
}

# “逍遥有聊” WebIM 产品的负载均衡, 反向代理两种 HTTP 服务器
server
{
    listen 80;
    server_name hi.xoyo.com;

    #反向代理一款定制开发的高性能消息队列 HTTP 服务器
    location /recmessage.xoyo
    {
        proxy_pass http://xoyohimsg_server_pool;
        proxy_set_header Host $host;
    }

    location /
    {
        proxy_pass http://php_server_pool;
        proxy_set_header Host $host;
        proxy_set_header X-Forwarded-For $remote_addr;
    }

    access_log /data1/logs/hi.xoyo.com_access.log;
}

#论坛负载均衡, 并对图片、Flash、JavaScript、CSS、静态 HTML 进行 Web 缓存
server{
    listen 80;
    server_name bbs.xoyo.com *.bbs.xoyo.com bbs.xoyo.kingsoft.com;
```



```
location /
{
    proxy_set_header Host $host;
    proxy_set_header X-Forwarded-For $remote_addr;
    proxy_pass http://bbs_server_pool;
}

location ~ .*\. (gif|jpg|jpeg|png|bmp|swf|js|css|html|shtml)$
{
    proxy_cache cache_one;
    proxy_cache_valid 200 10m;
    proxy_cache_valid 304 1m;
    proxy_cache_valid 301 302 1h;
    proxy_cache_valid any 1m;
    proxy_cache_key $host$uri$is_args$args;
    proxy_set_header Host $host;
    proxy_set_header X-Forwarded-For $remote_addr;
    proxy_pass http://bbs_server_pool;
}

log_format bbs '$remote_addr $host $remote_user [$time_local] "$request" '
    '$status $body_bytes_sent "$http_referer" '
    '"$http_user_agent" $http_x_forwarded_for';
access_log /data1/logs/bbs.xoyo.com_access.log bbs;
}
```

#论坛附件反向代理，限制下载速度为 256KB/秒

```
server{
    listen      80;
    server_name att03.bbs.xoyo.com att02.bbs.xoyo.com att01.bbs.xoyo.com;

    location /
    {
        #限制下载速度为 256KB/秒
        limit_rate 256k;
        proxy_pass http://xx.xx.xx.19;
        proxy_set_header Host $host;
        proxy_set_header X-Forwarded-For $remote_addr;
    }

    access_log off;
}
```

#逍遥江湖 SNS 社区，管理后台定位到一台服务器上，并对图片、Flash、JavaScript、CSS 进行 Web 缓存区

```
server{
    listen      80;
    server_name hu.xoyo.com *.hu.xoyo.com;

    location /
    {
```

```
    proxy_pass http://php_server_pool;
    proxy_set_header Host $host;
    proxy_set_header X-Forwarded-For $remote_addr;
}

location ~ .*\. (gif|jpg|jpeg|png|bmp|swf|js|css)$
{
    proxy_cache cache_one;
    proxy_cache_valid 200 10m;
    proxy_cache_valid 304 1m;
    proxy_cache_valid 301 302 1h;
    proxy_cache_valid any 1m;
    proxy_cache_key $host$uri$is_args$args;
    proxy_set_header Host $host;
    proxy_set_header X-Forwarded-For $remote_addr;
    proxy_pass http://php_server_pool;
}

location ~ ^/admincp.php
{
    #管理后台定位到一台服务器上
    proxy_pass http://xx.xx.xx.4;
    proxy_set_header Host $host;
    proxy_set_header X-Forwarded-For $remote_addr;
}

access_log /data1/logs/hu.xoyo.com_access.log;
}
}
```

10.2 Nginx+PHP 类网站应用案例

10.2.1 Nginx+PHP 在金山逍遥网 CMS 发布系统中的应用

金山逍遥网 CMS 发布系统采用 PHP 编写，自主开发的 MVC 框架，和其他的 MVC 框架一样，拥有单一请求处理入口/application/cmsmanage/index.php，逻辑清晰，Nginx 的配置也就简单明了：如果文件不存在，则直接 Rewrite（重写）到单一入口文件/application/cmsmanage/index.php 上。

MVC 是 Model（模型）、View（视图）、Controller（控制器）这三个单词的缩写组合。MVC 是一种普遍的软件敏捷开发模式，在许多领域特别是桌面编程领域早已经得到了广泛的应用，随着众多框架的涌现，MVC 在 PHP 的各个框架中也得到了实现。

单一入口指在一个网站中，所有的动态请求都是指向一个脚本文件的，例如 `http://cms.xoyo.com/application/cmsmanage/index.php`，所有对应用程序的访问都必须通过这个入口。正是单一入口才使得 MVC 模式得以实现，因为当你访问一个 URL 的时候，index.php 会调用基础类库，做

初始化工作，并通过从地址栏的 URL 路径获取参数，加载控制器、视图、模型等内容信息。

代码 10-3 则是金山游戏官方网站——逍遥网的 CMS 发布系统 Nginx 配置：

代码 10-3

```
user www www;

worker_processes 8;

error_log /data1/logs/nginx_error.log crit;

pid /usr/local/webserver/nginx/nginx.pid;

#Specifies the value for maximum file descriptors that can be opened by this process.
worker_rlimit_nofile 51200;

events
{
    use epoll;
    worker_connections 51200;
}

http
{
    include mime.types;
    default_type application/octet-stream;

    #charset utf-8;

    server_names_hash_bucket_size 128;
    client_header_buffer_size 32k;
    large_client_header_buffers 4 32k;
    client_max_body_size 300m;
    client_body_buffer_size 128k;

    sendfile on;
    #tcp_nopush on;

    keepalive_timeout 65;

    tcp_nodelay on;

    fastcgi_connect_timeout 300;
    fastcgi_send_timeout 300;
    fastcgi_read_timeout 300;
    fastcgi_buffer_size 64k;
    fastcgi_buffers 4 64k;
    fastcgi_busy_buffers_size 128k;
    fastcgi_temp_file_write_size 128k;
```



```
gzip off;
gzip_min_length 1k;
gzip_buffers 4 16k;
gzip_http_version 1.1;
gzip_comp_level 2;
gzip_types text/plain application/x-javascript text/css application/xml;
gzip_vary on;

#limit_zone crawler $binary_remote_addr 10m;

server
{
    listen 80;
    server_name cms.xoyo.com;
    index index.html index.htm index.php;
    root /data0/htdocs/cms.xoyo.com;

    location ~ .*\. (sh|bash)?$ { return 403; }

    location / {
        index index.html index.php;
        #如果请求的文件不存在,则重定向到单一入口文件上
        if (!-f $request_filename)
        {
            rewrite ^/(.*)$ /application/cmsmanage/index.php last;
        }
    }

    location ~ .*\. (php|php5)?$
    {
        #fastcgi_pass unix:/tmp/php-cgi.sock;
        fastcgi_pass 127.0.0.1:9000;
        fastcgi_index index.php;
        include fcgi.conf;
    }

    log_format cms '$remote_addr - $remote_user [$time_local] "$request" '
        '$status $body_bytes_sent "$http_referer" '
        '"$http_user_agent" $http_x_forwarded_for';
    access_log /data1/logs/cms.xoyo.com_access.log cms;
}
}
```

10.2.2 Nginx+PHP 在某分类信息网站中的应用

由于历史原因,该网站没有任何开发框架,针对搜索引擎的 URL 优化,兼容旧的 URL 地址,几乎全靠 Rewrite 规则来实现,这也使得该网站 nginx.conf 配置中拥有大量的 Rewrite 规则。虽然这种方法不推荐使用,但是,这些实现各种功能的 Rewrite 规则的应用,具有不小的学习价值。

由于篇幅有限，本书就不列出配置文件的完整内容了，感兴趣的读者可以访问以下网址查看：

http://blog.s135.com/book/nginx/nginx_rewrite_conf.txt

10.3 Nginx 视频点播类网站应用案例

10.3.1 Nginx 视频点播在金山游戏视频网站中的应用

金山逍遥视频网站为金山软件公司网络游戏的宣传视频，提供 Flash 视频流媒体点播服务。

整个逍遥视频站点由 4 部分组成：

- (1) v.xoyo.com（逍遥视频网站页面部分）；
- (2) FLV 视频节点部分；
- (3) api.v.xoyo.com（逍遥视频 Flash 播放器与 API 接口部分）；
- (4) admin.v.xoyo.com（逍遥视频管理后台部分、集成在 CMS 系统中）。

视频发布流程如下：

金山软件公司的视频编辑人员通过 CMS 统一登录入口，登录进入 admin.v.xoyo.com 视频管理后台。在此后台可以上传 FLV 视频、文字描述。视频上传完成后，系统会自动生成视频 ID 编号（即 vid），并将视频 Rsync 推送到各节点服务器。

视频播放流程（以访问 vid=387 的视频为例）如下：

(1) 用户访问逍遥视频页面 <http://v.xoyo.com/play/387>，页面内嵌入了 Flash 播放器地址：<http://api.v.xoyo.com/external/player.swf?config=http://api.v.xoyo.com/external/video-387.swf>，其中 <http://api.v.xoyo.com/external/player.swf> 为 Flash 播放器，<http://api.v.xoyo.com/external/video-387.swf> 为经过 Rewrite 重写规则伪静态的视频 API 接口，其真实地址为 <http://api.v.xoyo.com/external/video.php?vid=387>。

(2) 视频 API 接口（<http://api.v.xoyo.com/external/video.php?vid=387>）记录视频播放数等信息，并根据用户来源 IP 所属的地域、ISP 网络接入商进行判断，选择一个当前用户下载速度最快的 FLV 视频节点，输出包含 FLV 视频下载地址的 XML 文件，内容如下。该 API 接口还会根据各 FLV 视频节点报告的实时带宽使用情况，在多个 IDC 节点动态调节视频带宽资源。当一个 IDC 节点的使用带宽超过设定值后，新的视频播放请求将自动按照地域、ISP 类型等顺序，分到邻近的带宽尚有空余的其他 IDC，代码如下。

