

# Nginx 的介绍

姚伟斌

( yaoweibin@gmail.com )

2010 年 8 月

# 提纲

- Nginx 是什么？
- 为什么选择 Nginx ？
- 如何使用 Nginx ？
- 本人参与的一些作品

Nginx 是什么？

# Nginx 的用途

Nginx (pronounced “engine X”) is a lightweight, high-performance Web server/reverse proxy and e-mail (IMAP/POP3) proxy, licensed under a BSD-like license. It runs on UNIX, GNU/Linux, BSD variants, Mac OS X, Solaris, and Microsoft Windows.



--From wiki

# Nginx 的作者



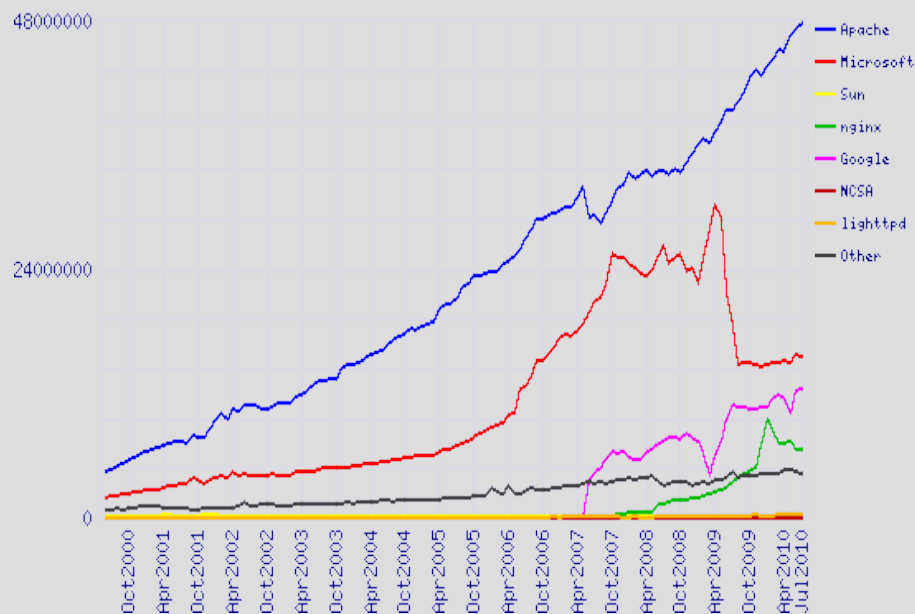
Igor Sysoev: 一个接近 40 岁的老程序员，拥有 10 余年的系统管理员经验，FreeBSD 爱好者，曾经开发过多个 Aapche 模块，对于 Apache 的设计优缺点深有感触。为人相当低调，很少参与开发讨论或者活动。整个 Nginx 几乎都是他一个人开发，很少接纳第三方模块。

主页: <http://sysoev.ru/en/>

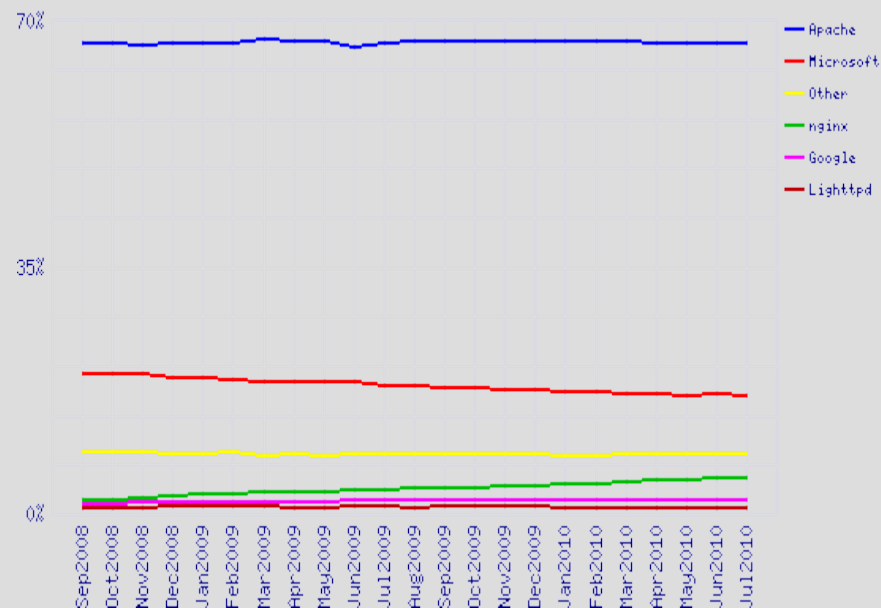
# Nginx 的开发历程

- 2001年春, 计划
- 2002年春, 原始版本 (drafts)
- 2003年秋, 第一个完善版本 (kqueue/select), 多线程
- 2004年春, epoll, master/workers, 在生产环境开始使用
- 2004年夏, mail proxy, 开始在rambler.ru使用
- Nginx-0.1.0 发布于 2004 年 10 月
- Nginx-0.2.0 发布于 2005 年 9 月
- Nginx-0.3.0 发布于 2005 年 10 月
- Nginx-0.4.0 发布于 2006 年 8 月
- Nginx-0.5.0 发布于 2006 年 12 月
- Nginx-0.6.0 发布于 2007 年 6 月
- Nginx-0.7.0 发布于 2008 年 5 月
- Nginx-0.8.0 发布于 2009 年 6 月
- 还在不断更新

# 当今 Web 服务软件的现状



Web 服务软件趋势图（所有域名）



Web 服务软件趋势图（前 100 万最繁忙的网站）

来源: <http://news.netcraft.com/archives/2010/07/16/july-2010-web-server-survey-16.html>

# 大网站偏爱 Nginx

根据 BuiltWith.com 的报告：

	前1百万的网站占有率	前1万的网站占有率
Nginx	3.14%	4.95%
Apache	63.60%	55.79%
IIS	21.15%	20.05%

\* 以上数据来自 <http://trends.builtwith.com/Web%20Server/nginx>

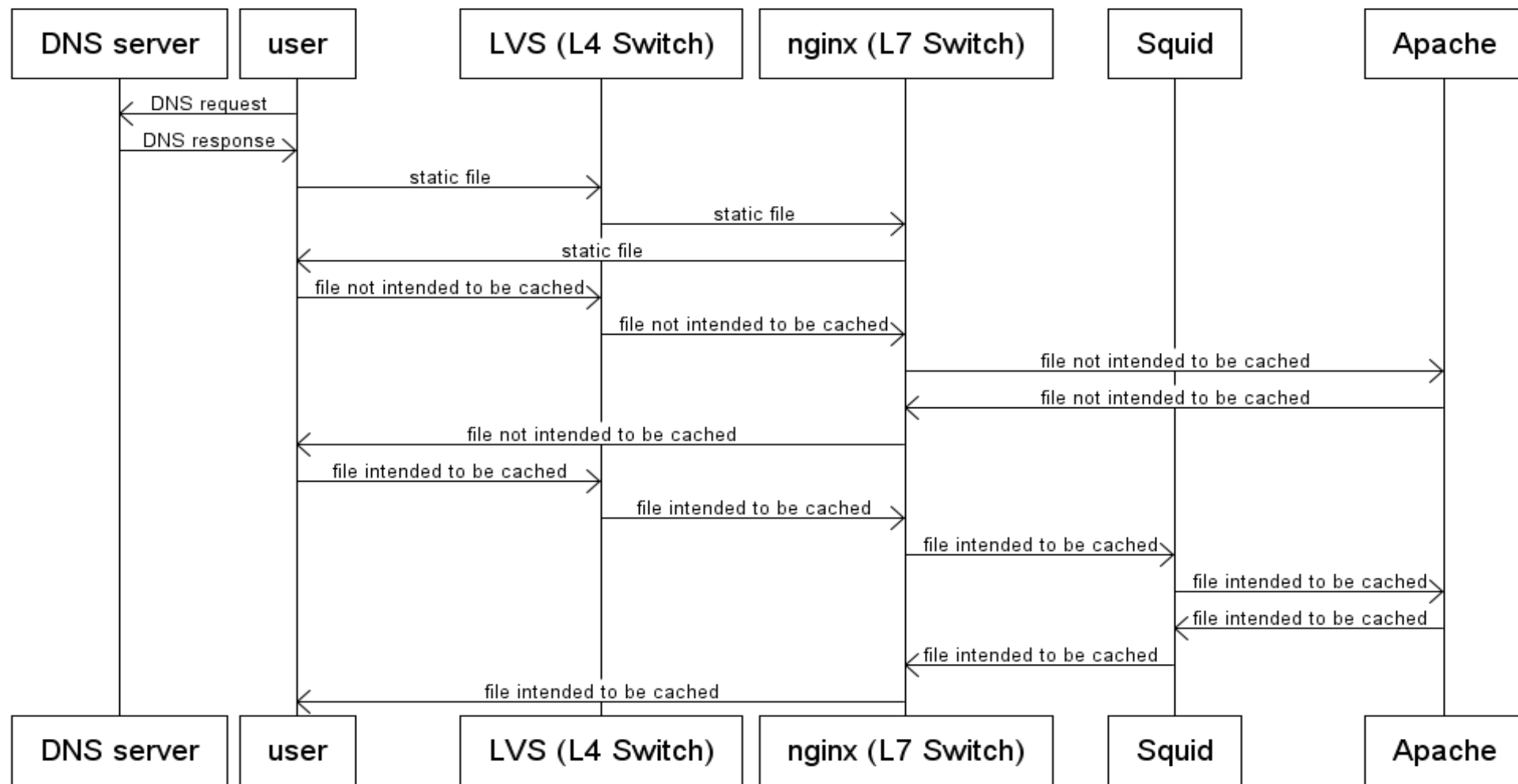


# 哪些网站在使用 Nginx ?

国外：wordpress、sourceforge、github、hulu、...

国内：新浪、网易、搜狐、水木，豆瓣，校内、迅雷、CSDN、...

# 网易博客使用 Nginx 的现状



为什么选择 Nginx ?

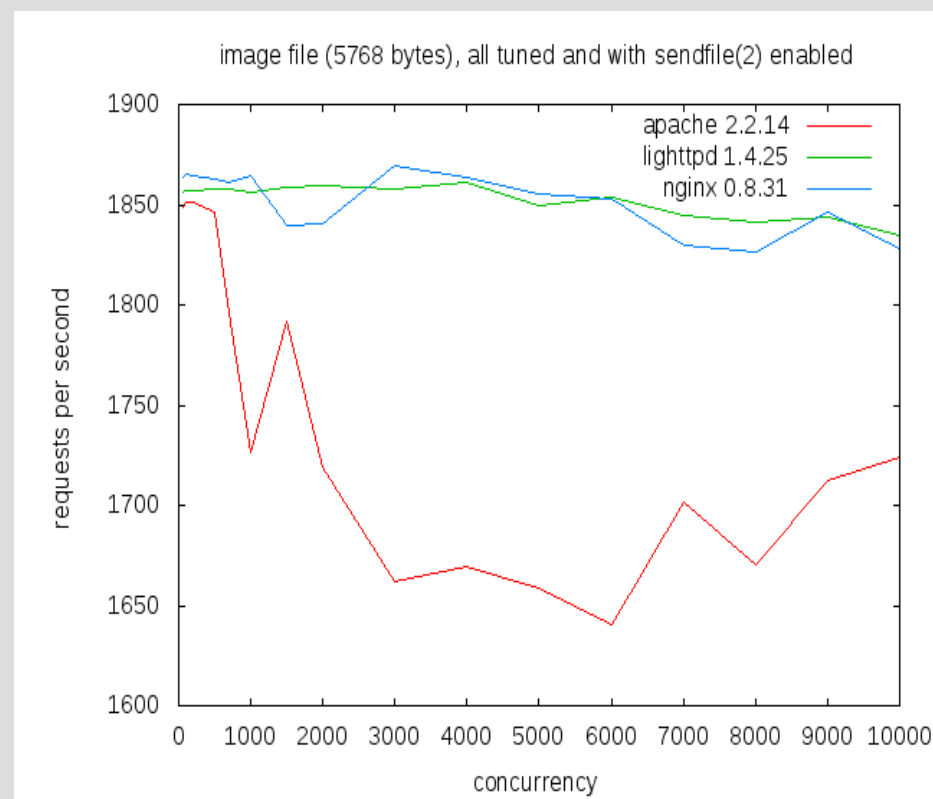
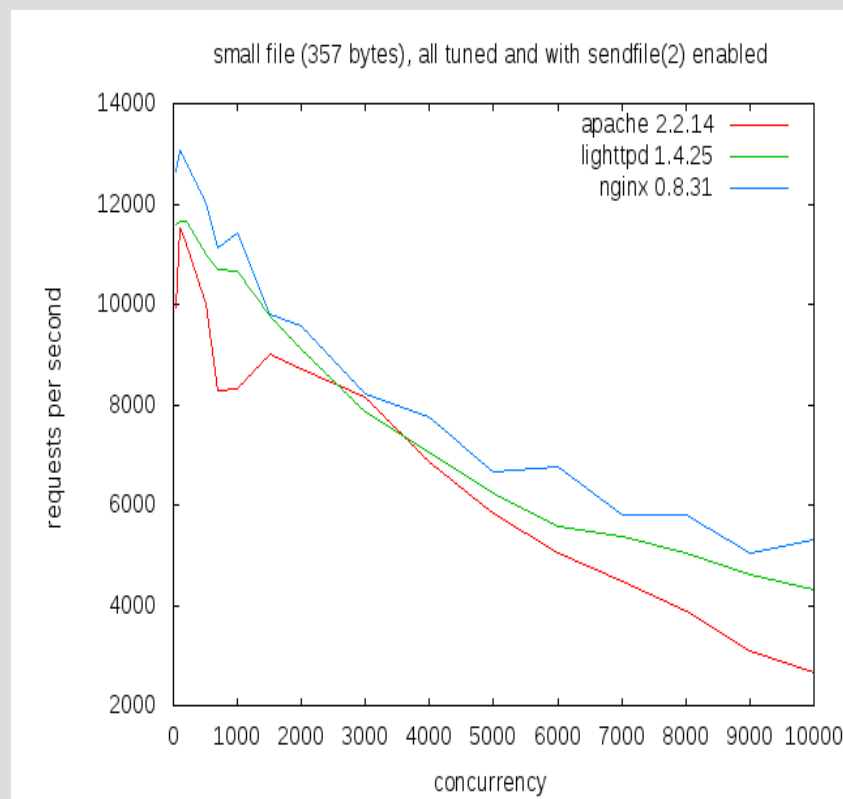
# 性能比较

比较对象： Apache 2.2.14 ， Nginx 0.8.3  
1 ， Lighttpd 1.4.25

测试内容： 小文件（ 357 字节）、图片（ 5.7  
K ）

配置： 3 个都优化过，基本上是公平测试，都  
开启了 `sendfile(2)`

# Apache VS. Lighttpd VS. Nginx



# 设计上的不同带来的性能差异

- Apache：采用 worker 模式，多线程 + 多线程，使用 select。内存及线程调度消耗大。而 event 模式与其他模块是不可调和的。
- Lighttpd：多进程非阻塞 IO，使用 epoll，但代码风格较差，1.5 已经基本停止开发。
- Nginx：跟 Lighttpd 架构类似，但是 Nginx 代码书写更加成熟，细节处理较好，高度模块化，有内存池的支持，代码长期更新。

# 功能比较

Web 服务器	Nginx	Apache	Lighttpd
反向代理	非常好	好	一般
Rewrite 规则	非常好	好	一般
FastCGI	好	差	非常好
热部署	支持	不支持	不支持
系统压力比较	很小	小	很大
稳定性	非常好	好	一般
安全性	一般	好	一般
技术资料	很少	非常多	一般
静态文件处理	非常好	一般	好
虚拟主机	支持	支持	支持
内存消耗	非常小	很大	非常小

\* 摘自张宴新书《实战 Nginx：取代 Apache 的高性能 Web 服务器》

# 选择 Nginx 的理由

- 高并发
- 内存消耗低
- 成本低
- 配置简单
- 支持重写规则
- 在线 gzip ， 节省带宽
- 稳定
- 支持热升级



如何使用 Nginx ?

# Nginx 的启动、停止、重启

- 启动: `./nginx -c nginx.conf`
- 停止: `kill -INT nginx` 的主进程
- 优雅停止: `kill -QUIT nginx` 的主进程
- reload: `kill -HUP nginx` 的主进程
- 平滑升级: `kill -USER2 nginx` 的主进程

# Nginx 的配置语法

- 块： events/http/mail/upstream/server/location 。
- 指令：读入相应配置，用分号结尾。
- 继承：上层的配置会被下层继承。
- 正则比较：使用 PCRE 库，目录或者 uri 之类可以任意比较。
- 变量：一般是请求相关的。

# 静态文件的默认配置

```
worker_processes 1;
error_log logs/error.log info;
events {
    worker_connections 1024;
}
http {
    include mime.types;
    default_type application/octet-stream;
    keepalive_timeout 65;
    server {
        listen 80;
        server_name localhost;
        location / {
            root html;
            index index.html index.htm;
        }
        error_page 500 502 503 504 /50x.html;
        location = /50x.html {
            root html;
        }
    }
}
```

# 反向代理

- 现在支持 http、fastcgi、memcache 协议的反向代理。
- 负载均衡方式多样：round\_robin、ip\_hash、fair、business、session。
- 错误恢复。
- 支持在线 gzip 功能，节省带宽。
- 后端健康检查。

# 文件镜像和缓存

- 文件镜像（ proxy\_store ）： 可以将代理过的文件存在本地，下次就不用取了。
- 文件缓存（ proxy\_cache ）： 对 URL 及相关部分作组合生成 MD5 ， 将此 MD5 文件名存在本地，同时设置过期时间。有第三方模块支持清理特定缓存文件。
- 在一般功能上可以替换 squid 。

# Rewrite 功能

- if:
  - If (condition) {...}
- rewrite
  - rewrite regex replacement
- set
  - set variable value
- example:

```
if ($host ~* www\.(.*)) {  
    set $host_without_www $1;  
    rewrite ^(.*)$ http://$host_without_www$1 permanent;  
}
```

# 什么情况下需要 Nginx

- apache 、 squid 不能负荷 , lighttpd 不够稳定
- 业务增长太快, 老板又要降低成本
- 核心组件, 负载较高的部分, 推荐用 Nginx 。



# Nginx 的展望

- 后端服务器 http 1.1 支持，改进 cache
- 内置脚本功能：lua、javascript
- 与数据库的通信：Mysql、PostgreSQL、Oracle
- 实现 HTTP 的长连接：push 模块
- ...
- 未来 Nginx 就是一个 Web 处理的框架，覆盖前端到后端。

本人参与的一些作品

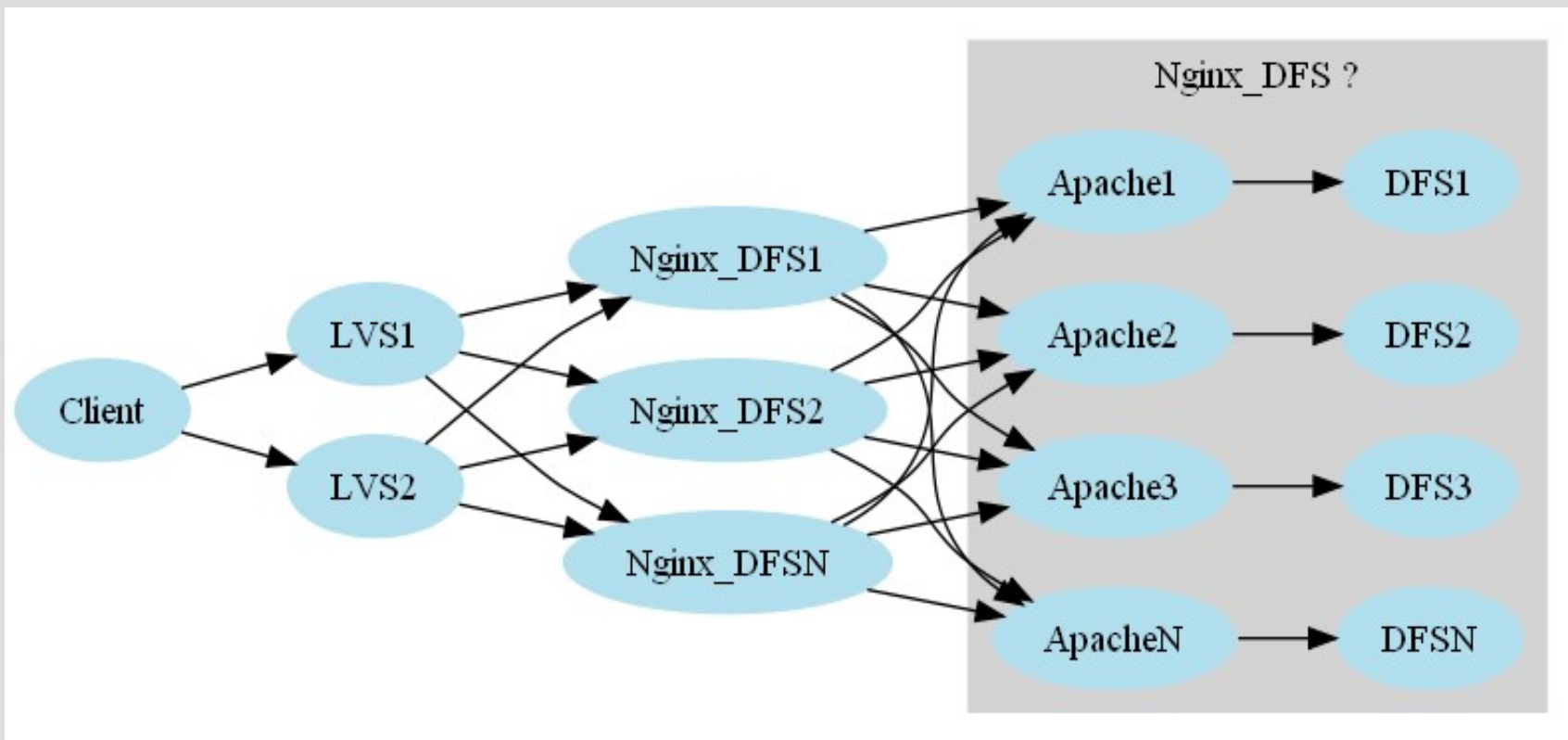
# 本人参与的一些作品

- Nginx\_netease\_dfs 模块：用以 dfs 图片代理
- Nginx 的 ajp 模块 ([http://github.com/yaoweibin/nginx\\_ajp\\_module](http://github.com/yaoweibin/nginx_ajp_module))
- Nginx\_netease\_pci 模块：与硬件加速卡配合
- Nginx 的 tcp proxy 模块 ([http://github.com/yaoweibin/nginx\\_tcp\\_proxy\\_module](http://github.com/yaoweibin/nginx_tcp_proxy_module))
- Nginx 的 substitutions 模块 (<http://code.google.com/p/substitutions4nginx/>)
- Nginx 的 jvm\_route 模块 (<http://code.google.com/p/nginx-upstream-jvm-route/>)
- Nginx 的 carp 模块 (<http://code.google.com/p/nginx-upstream-carp/>)

# Nginx\_netease\_dfs\_module

- 读取 DFS 文件的 docid，计算其 MD5 值，判断是否匹配
- 获取桶号，卷号，目录。查找本地的 IP 与桶号对应表，找到目标服务器。
- 将请求发送到对应的服务器。

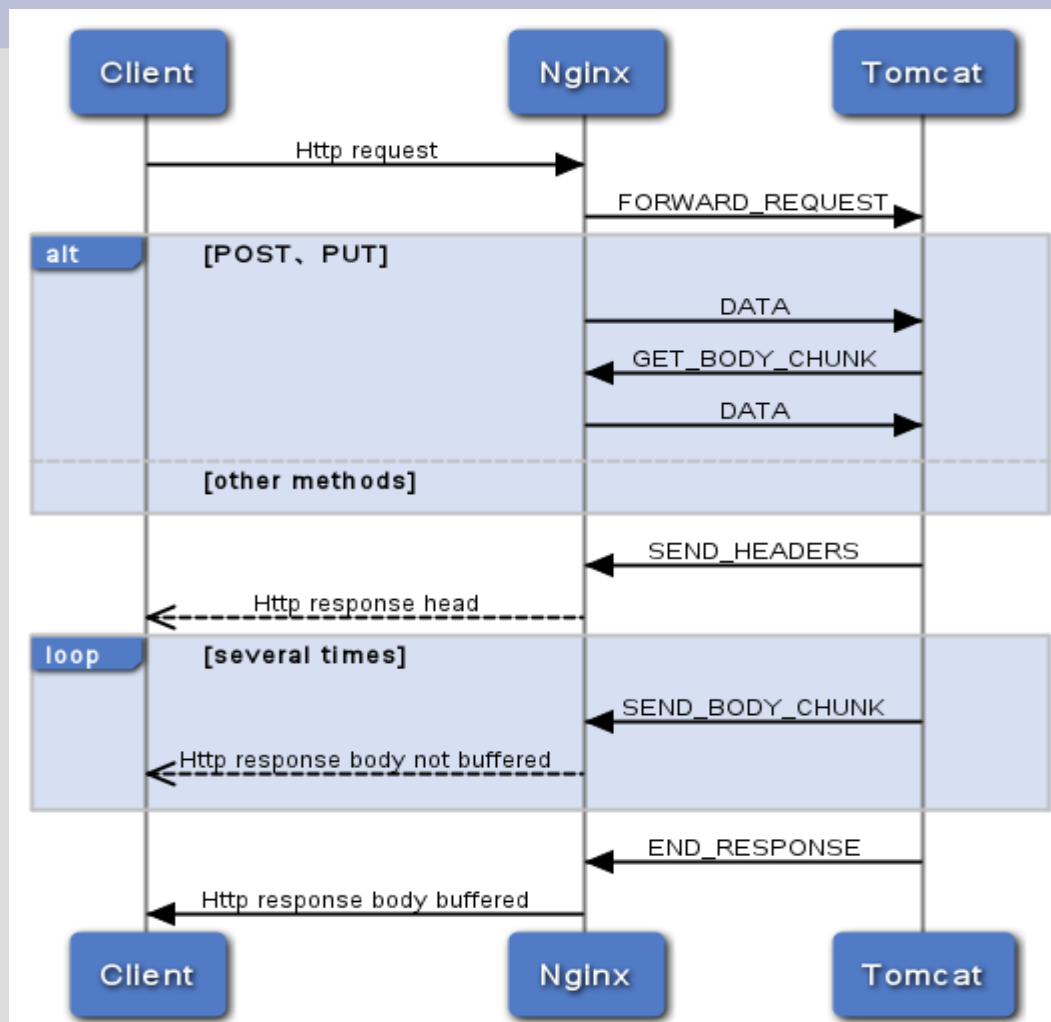
# Nginx\_netease\_dfs\_module



\* 与原有的架构相比，服务器负载减低了 30%-50%

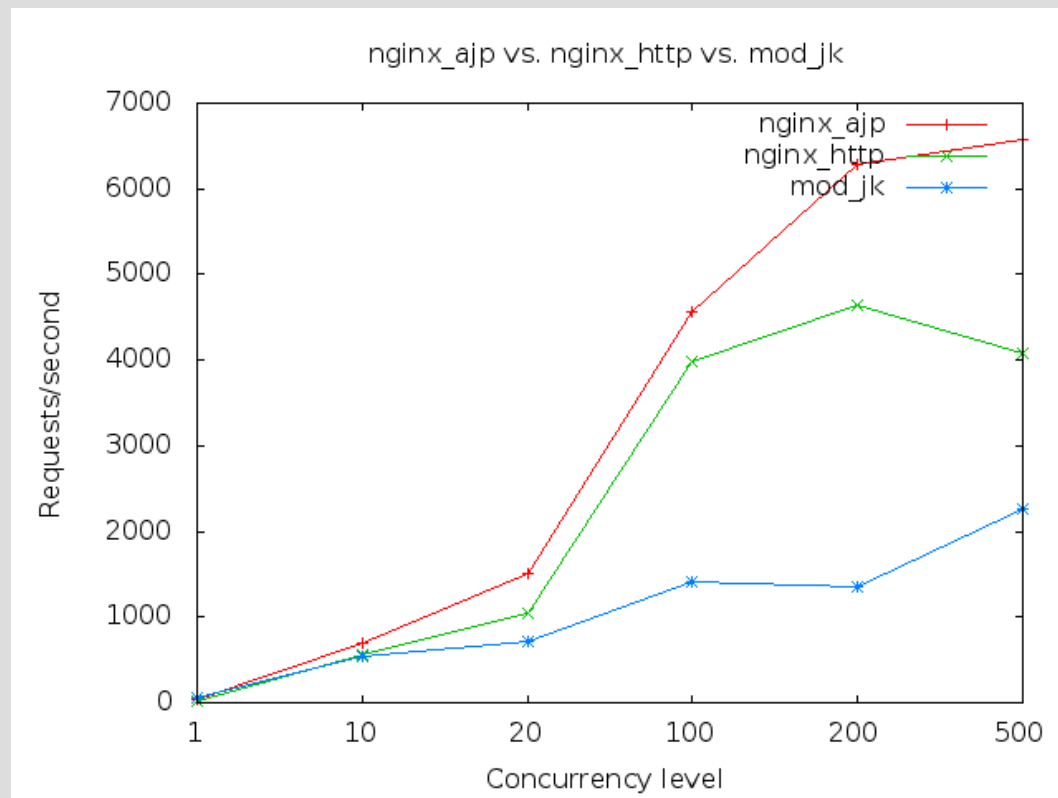
# Nginx\_AJP\_module

- AJP 处理流程



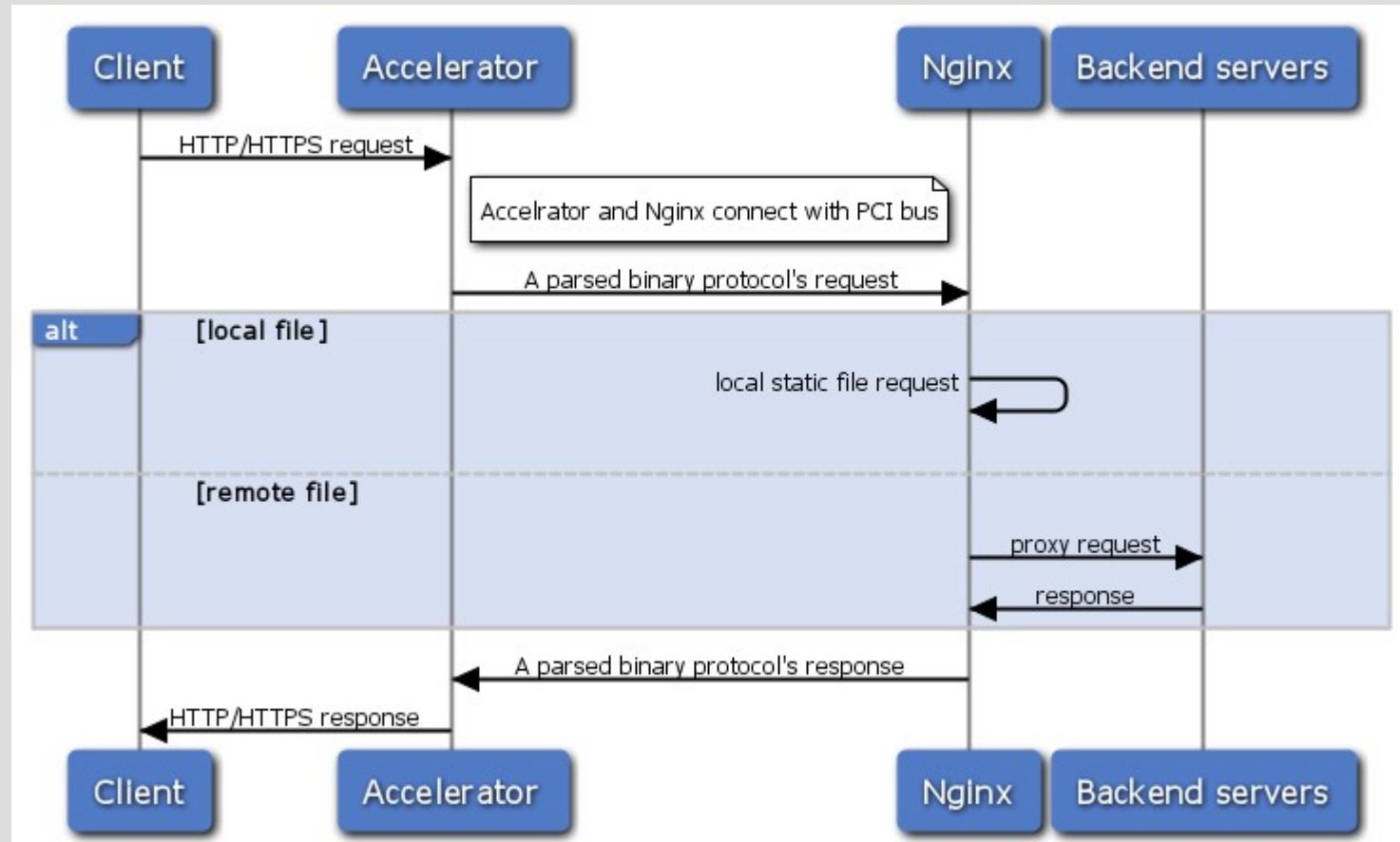
# Nginx\_AJP\_module

- 支持 Nginx 与 Tomcat AJP connector 直接通信。
- Session sticky
- Keepalive



# Nginx\_PCI\_module

- 研发中





# Nginx\_PCI\_module

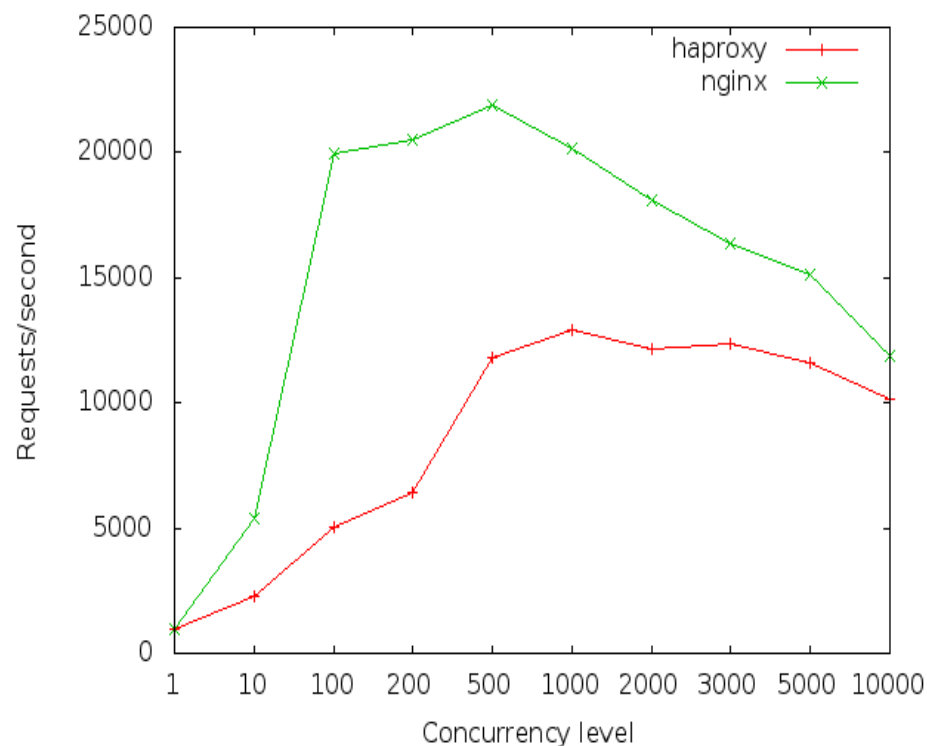
- Nginx 可以对加速卡的 HTTP\HTTPS 监听进行配置。
- 加速卡类似一个应用层的网卡，负责 TCP 层和应用层的解析。
- 加速卡和 Nginx 的请求和回复头通过 mmap 内存共享，实现零拷贝；回复 body 通过 sendfile 实现发送。

# Nginx\_TCP\_proxy\_module

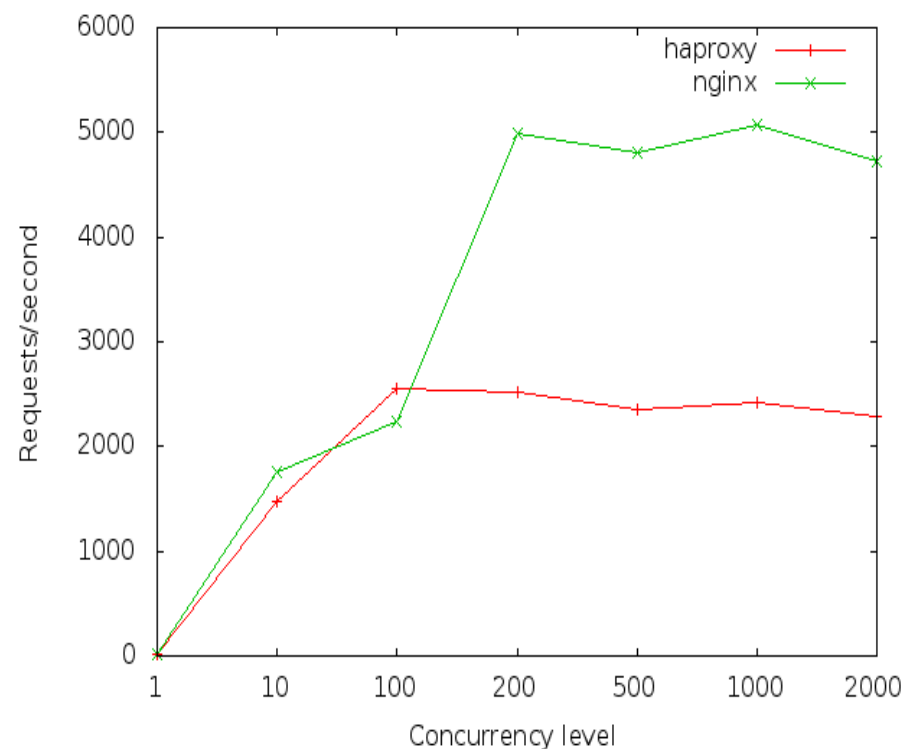
- 提供四层代理功能。
- 支持 round\_robin 和 ip\_hash 方式的反向代理。
- 支持 SSL、HTTP、MYSQL、SMTP、POP3、IMAP 等协议的后端健康检查。
- 简单的 ACL 控制功能。
- 简单的后端服务器连接状态显示。

# Nginx\_TCP\_proxy\_module

Haproxy vs. Nginx\_Tcp\_Proxy\_Module, keepalive requests



Haproxy vs. Nginx\_Tcp\_Proxy\_Module, non-keepalive requests



# Nginx\_substitutions\_module

- 按行扫描，提供对 HTTP body 的正则匹配替换。
- 甚至可以提供类似 SSI 的功能。
- 经过多次优化，性能不错。

例子：

```
subs_filter st(\d*).example.com $1.example.com ir;
```

# Nginx\_jvm\_route\_module

- 模块原理：resin、tomcat 如何 session sticky 的
- 提供按照 Cookie 或 URL 方式 session sticky 的解决方法

# Ngix\_carp\_module

- CARP : 提供固定 hash 路由。
- 比一致性 hash 稍微差点, 在加入或者退出服务器的 hash 路由需要重新计算。

# 谢谢

Question ?

...

The end