# 淘宝技术架构简介

# 大纲

- 背景介绍
- 应用案例分析
- 开发与定制
- 当前工作

# 1、背景介绍

# Nginx简介

- Web服务器、HTTP反向代理和邮件代理服务器
- 俄罗斯程序员Igor Sysoev于2002年开始
- 全球使用量排名第三
- 2011年成立商业公司
- 特点
  - 性能非常高
  - 资源占用（CPU、内存）非常节省
  - 内存池设计，非常稳定
  - 高度模块化，易于扩展

# 淘宝网使用Nginx的过程

- 2009年开始使用和探索
- 2010年开始开发大量模块
  - 通用的
  - 业务的
- 2011年开始
  - 修改Nginx的核心代码
  - 启动Tengine项目并开源
- 2012年Apache全部替换为Tengine

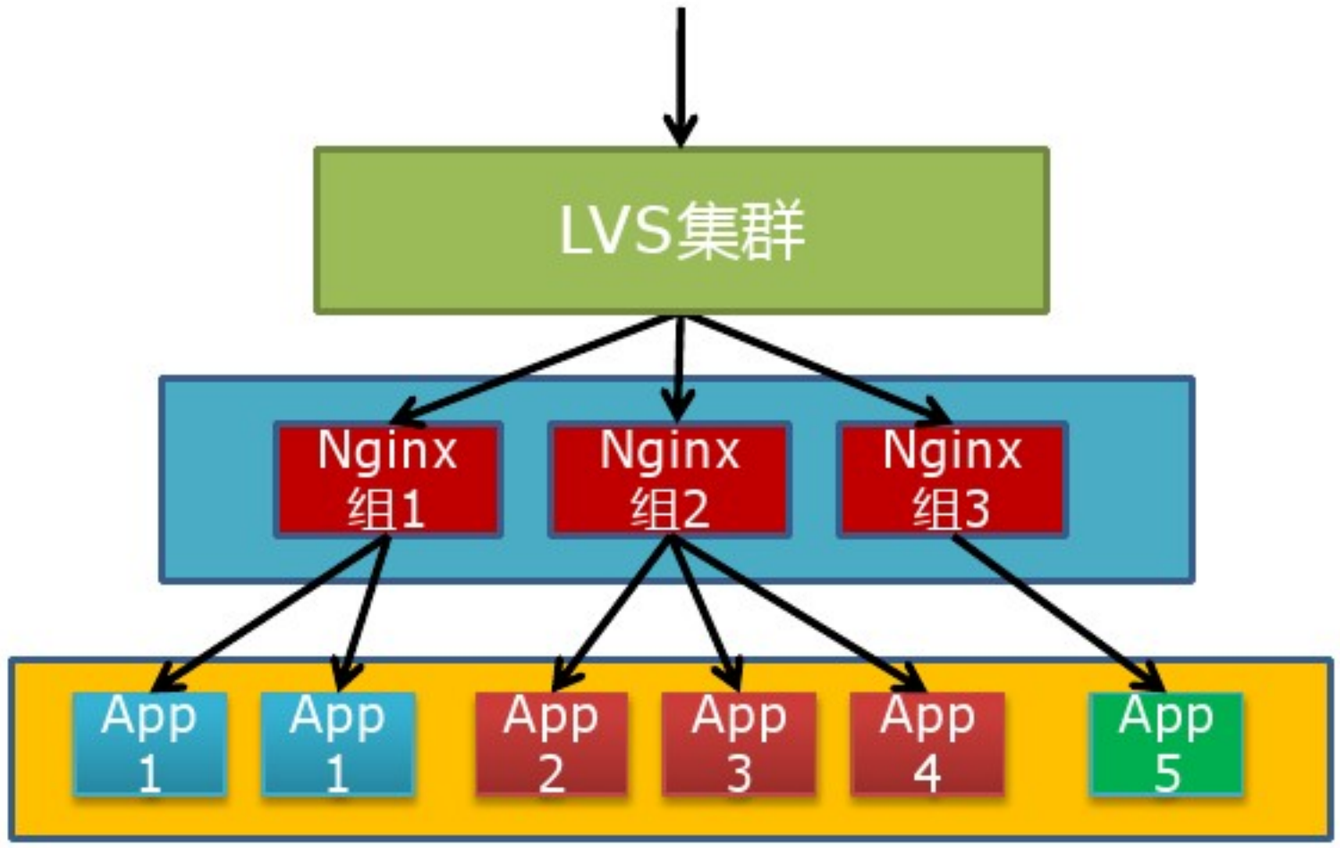
# 淘宝网应用Nginx的收益

- 业务更加稳定
  - Nginx大连接数目支持非常好
  - Nginx本身的内存占用很少，不会吃swap
- 业务性能更高
  - QPS比Apache要好
  - 节省机器数目
  - 基于Nginx的模块性能往往是之前业务的数倍
  - 有效抵御DDOS攻击

# 2、应用案例分析

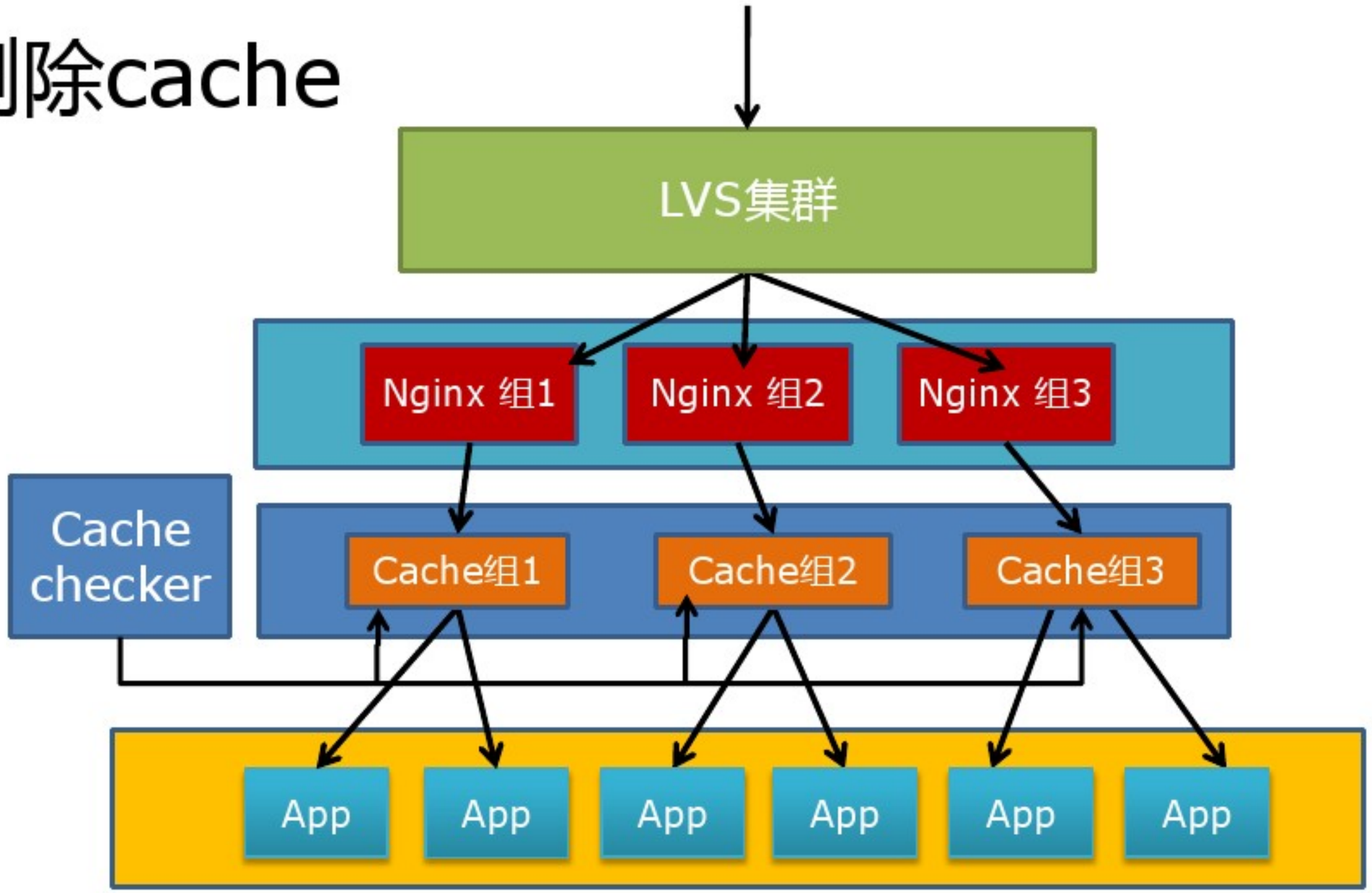
# Web接入层

- Nginx的职能
  - SSL卸载
  - 七层接入管理
  - 安全防御
  - 负载均衡
  - 灰度发布
  - 静态资源



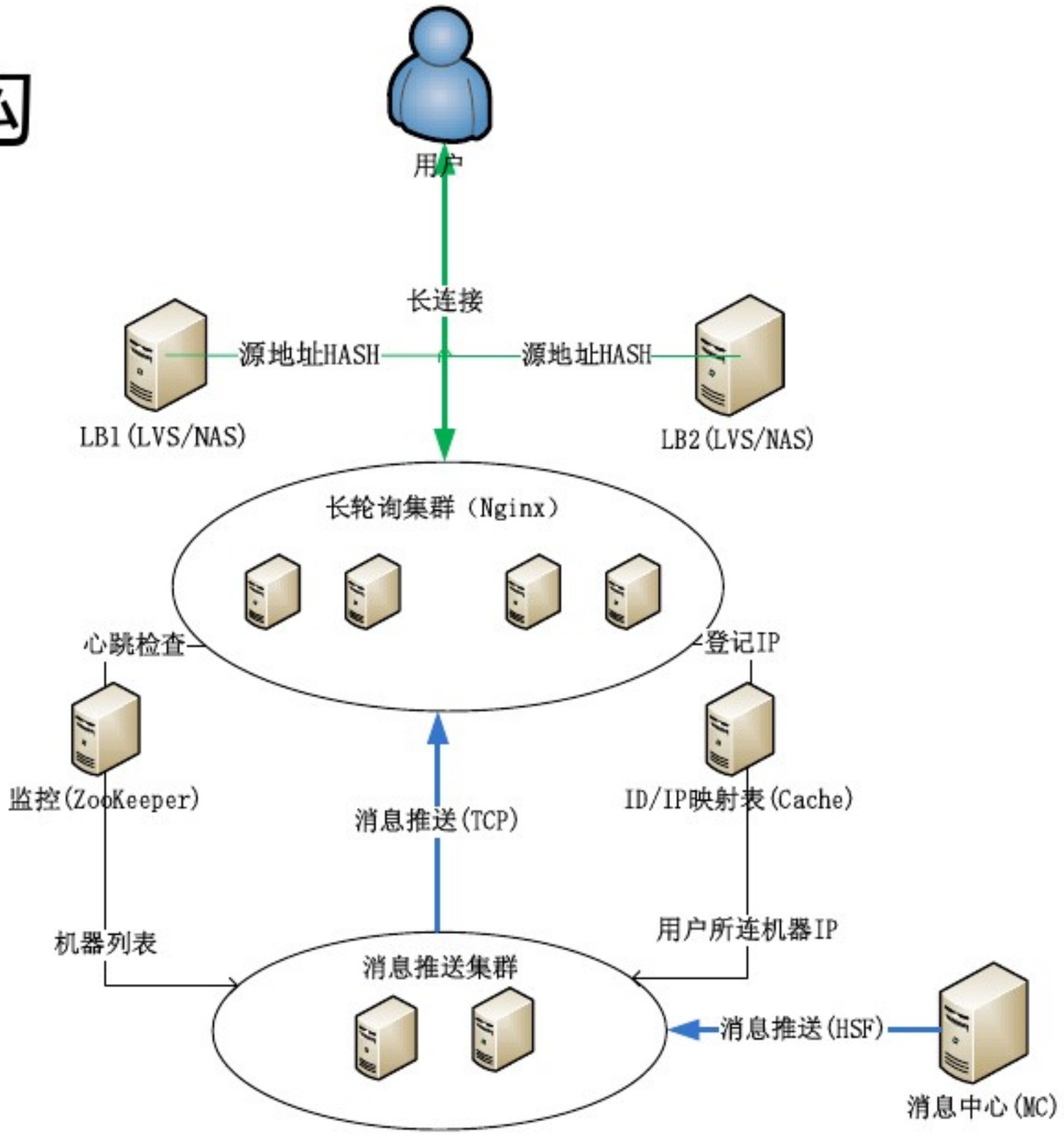
# 动态内容的静态化

- 把所有能cache的内容都cache住
- 主动删除cache



# 大用户群消息推送

- Comet服务架构
- 部署容量
  - 60万连接/台
- 运行数据
  - 30万连接/台



# 日志收集与统计系统

- 功能（可看成私有的Google Analytics）
  - JavaScript埋点
  - 收集日志
  - 分析统计信息
- 实现
  - Nginx模块
  - 分布式传输系统
  - Hadoop上运行MapReduce统计
- 性能
  - 小几十台机器一天几十亿PV
  - 单机处理能力4万QPS

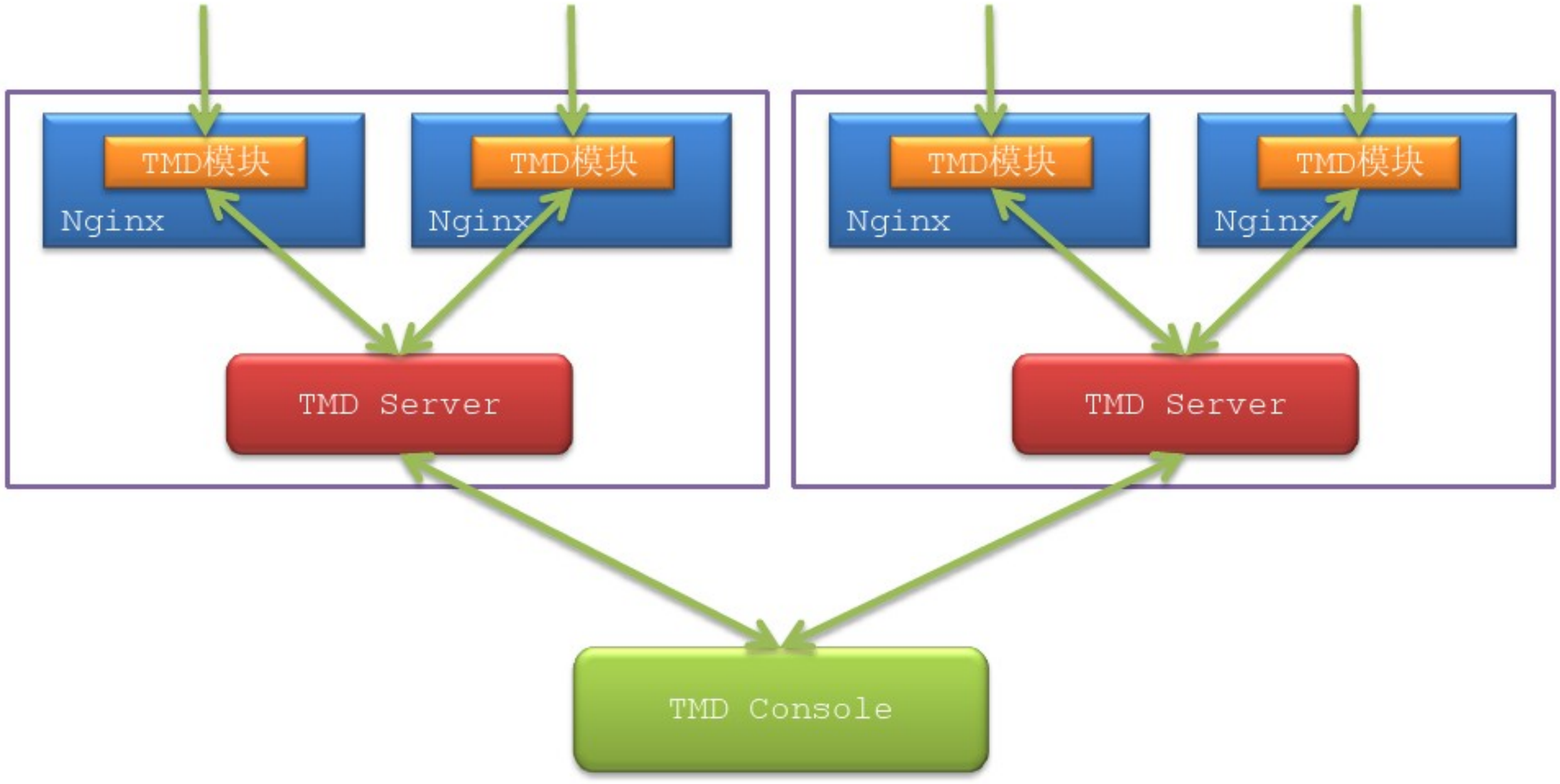
# RESTful接口层

- RESTful接口支持（准备开源）
  - TFS
    - 分布式文件系统，类似于GFS
  - Tair
    - 分布式K/V存储系统
- 简化应用开发
  - 可返回JSON格式直接让浏览器处理
    - 从而不必在服务器端渲染页面

# 分布式防攻击系统

- 功能
  - 抵挡中小型的DDoS攻击
  - 可以封禁IP、Cookie
  - 限流
  - 提供验证码服务
- 淘宝TMD（Taobao Missile Defense）系统
  - Nginx作为防攻击系统的终端，实时发送日志给Server
  - TMD Server做策略分析
  - TMD Console执行汇总和控制台

# TMD系统架构图


TMD模块
Nginx
TMD模块
Nginx
TMD模块
Nginx
TMD模块
Nginx
TMD Server
TMD Server
TMD Console

# 3、开发与定制

# 动态加载模块

- Nginx核心代码静态编译
- 功能模块动态编译成so文件
  - ./dso_tool --add-module=/path/to/lua-nginx-module
- 好处
  - 核心模块跟功能模块去耦合，不必一起编译
  - 对于包管理系统来说，不再需要N多包
  - 修正某个模块，只需编译相应模块

# 动态加载模块使用

- 使用方法

```
dso {
    load ngx_http_lua_module.so;
    load ngx_http_memcached_module.so;
}
```

- 动态库比静态代码性能差？
  - Wangbin：

采用tengine dso编译技术，对我们广告系统进行动态编译，并与原先静态编译进行了性能对比（采用tcpcopy）报告如下，1）吞吐量，静态/动态=100.155% 2）cpu消耗，动态/静态=100.3% 从上面可以看出，性能并没有明显下降。以上只是针对我们系统的6小时测试结果，仅供参考

9月27日16:51 来自新浪微博 | 举报 转发(5) | 收藏 | 评论(3)

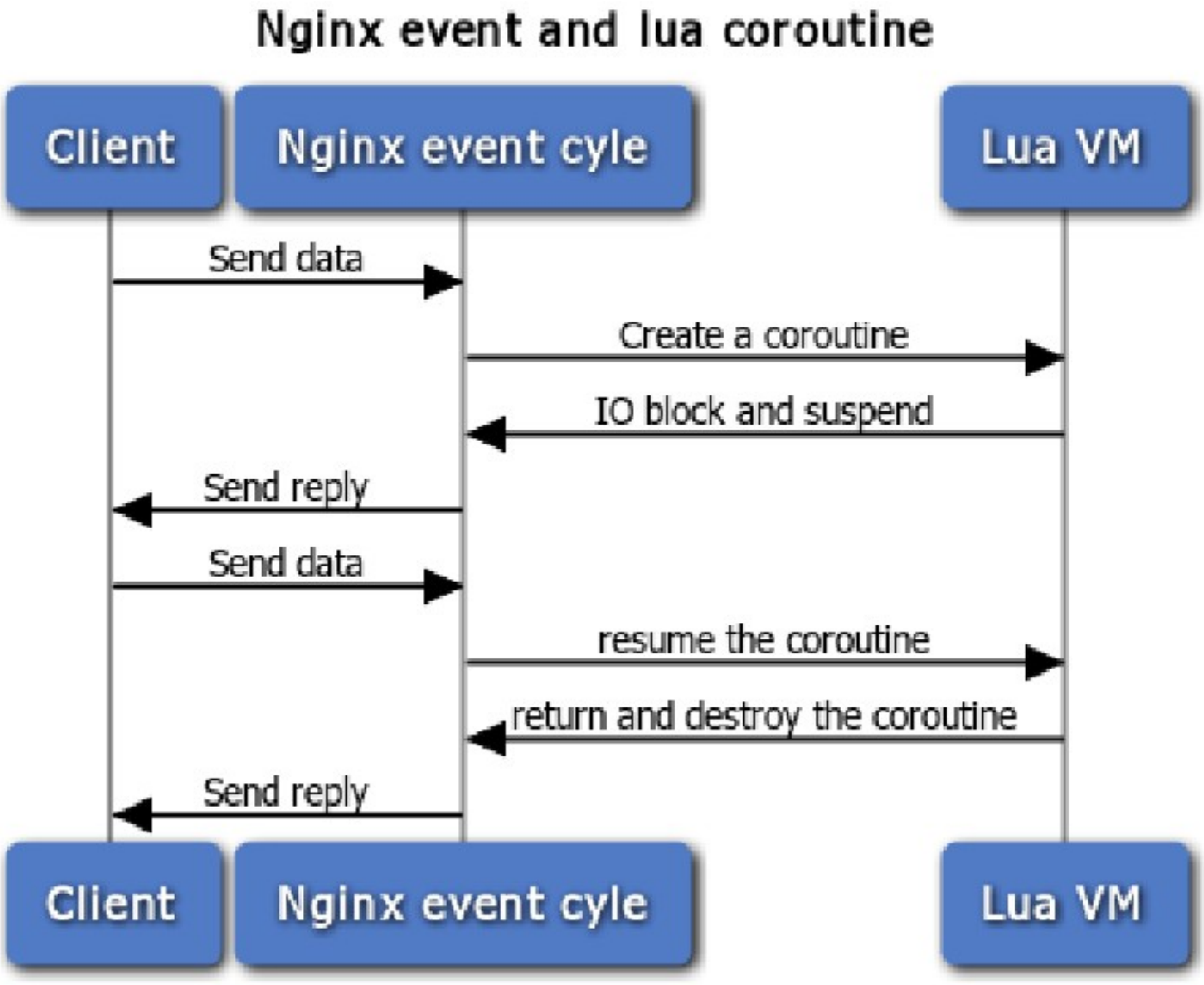
# ngx_lua模块思想

- 引进动态脚本语言Lua
  - Lua语言强大且简单
  - 适合嵌入
  - 支持协程（coroutine）
- 价值
  - 用同步的语义来实现异步的调用

# ngx_lua原理

- 每个Nginx工作进程使用一个Lua VM，工作进程内所有协程共享VM
- 每个外部请求都由一个Lua协程处理，协程之间数据隔离
- Lua代码调用I/O操作接口时，若该操作无法立刻完成，则打断相关协程的运行并保护上下文数据
- I/O操作完成时还原相关协程上下文数据并继续运行

# ngx_lua原理


Nginx event and lua coroutine
Client
Nginx event cyle
Lua VM
Send data
Create a coroutine
IO block and suspend
Send reply
Send data
resume the coroutine
return and destroy the coroutine
Send reply
Client
Nginx event cyle
Lua VM

# 代码示例

```
location /http_client {
    proxy_pass $arg_url;
}

location /web_iconv {
    content_by_lua '
        local from, to, url = ngx.var.arg_f, ngx.var.arg_t, ngx.var.arg_u
        local iconv = require "iconv"
        local cd = iconv.new(to or "utf8", from or "gbk")
        local res = ngx.location.capture("/http_client?url=" .. url)
        if res.status == 200 then
            local ostr, err = cd:iconv(res.body)
            ngx.print(ostr)
        else
            ngx.say("error occured: rc=" .. res.status)
        end
    ';
}
```

# 组合JavaScript和CSS文件

- Yahoo!前端优化第一条原则
  - Minimize HTTP Requests
  - 减少三路握手和HTTP请求的发送次数
- 淘宝CDN combo
  - concat模块
  - 将多个JavaScript、CSS请求合并成一个
- Pagespeed?
  - 自动优化，结合cache，效果显著

# 淘宝CDN Combo的使用

- 以两个问号（??）激活combo特性
- 多个文件之间用逗号（,）分开
- 用一个?来表示时间戳
  - 突破浏览器缓存
- 例子

http://a.tbcdn.cn/??s/kissy/1.1.6/kissy-min.js,p/global/1.0/global-min.js,p/et/et.js?t=20110923201103O1.js

# 系统过载保护

- 判断依据
  - 系统的loadavg
  - 内存使用（swap的比率）

- sysgurad模块

```
sysguard on;
sysguard_load load=4 action=/high_load.html;
sysguard_mem swapratio=10% action=/mem_high.html
```

- 可定制保护页面

# 多种日志方式

- 本地和远程syslog支持

```
access_log      syslog:user:info:127.0.0.1:514 combined;
```

- 管道支持

```
access_log      pipe:/path/to/cronolog combined;
```

- 抽样支持
  - 减少写日志的数量，避免磁盘写爆

```
access_log      /path/to/file combined ratio=0.01;
```

# CPU亲缘性设置的简化

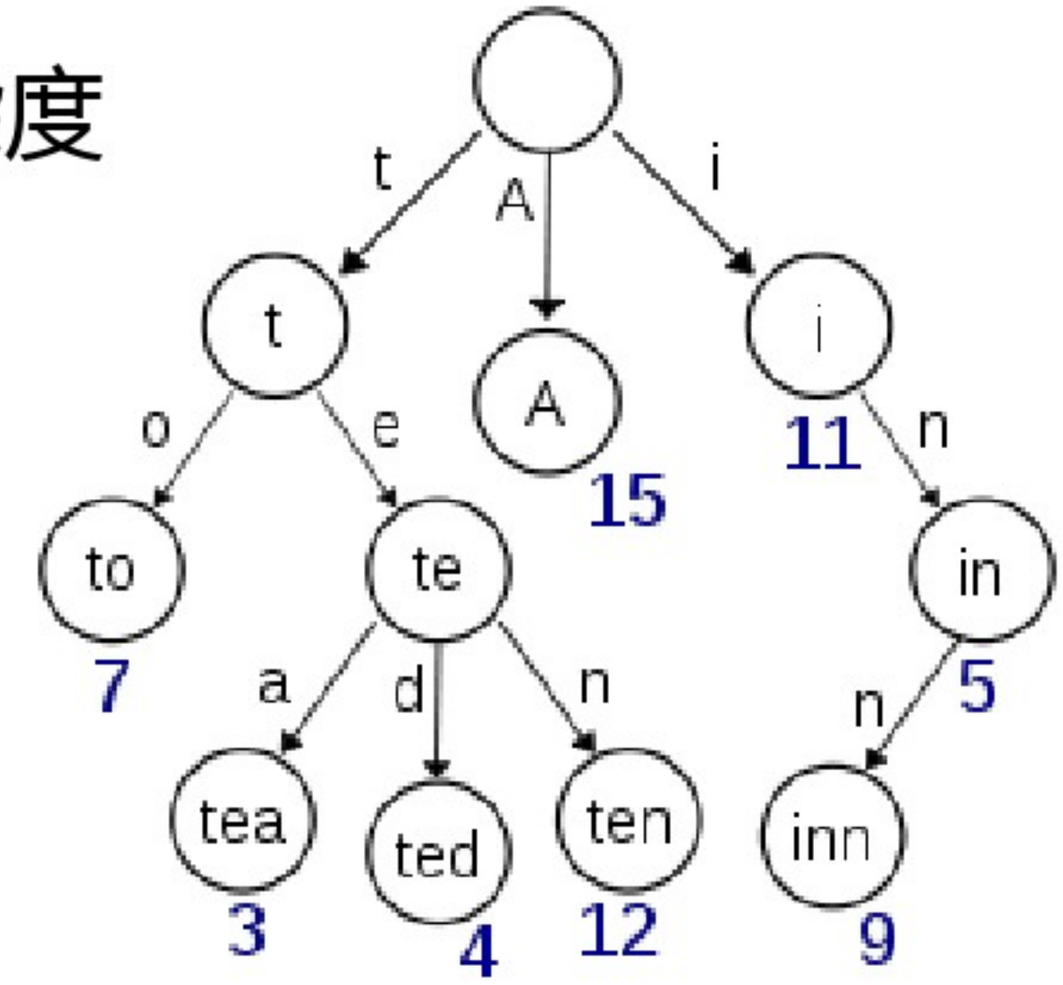
- 使用对比

```
# standard nginx
worker_processes        8;
worker_cpu_affinity     00000001 00000010 00000100 00001000
                        00010000 00100000 01000000 10000000
```

```
# tengine
worker_processes        auto;
worker_cpu_affinity     auto;
```

# user_agent模块

- 功能：将浏览器、爬虫匹配成变量
- 实现
  - Trie树匹配，O(n)复杂度
  - Nginx的browser模块
    - 算法复杂度O(n ^ 3)
    - 不灵活，没有版本匹配



# 对Nginx的limit_req增强

- 白名单支持
- 指定跳转页面支持
- 同一个location下多limit_req支持

```
location / {
    limit_req zone=one burst=5;
    limit_req zone=two forbid_action=@test1;
    limit_req zone=three burst=3 forbid_action=@test2;
}

location /off {
    limit_req off;
}

location @test1 {
    rewrite ^ /test1.html;
}

location @test2 {
    rewrite ^ /test2.html;
}
```

# 主动健康检查

- 心跳检查，发现后端服务器失效的响应快
- L7检查使上线下线很方便
- 后端server的状态监控页面：HTML/JSON/CSV格式
- 可检查多种后端服务器
  - HTTP/HTTPS
  - AJP
  - MySQL
  - ...

**Nginx http upstream check status**

**Check upstream server number: 2, generation: 5**

| Index | Upstream | Name | Status | Rise counts | Fall counts | Check type |
|---|---|---|---|---|---|---|
| 0 | linuxtone | 127.0.0.1:81 | up | 607 | 0 | tcp |
| 1 | linuxtone | 127.0.0.1:82 | up | 70 | 0 | tcp |

# 输入体过滤器（input body filter）

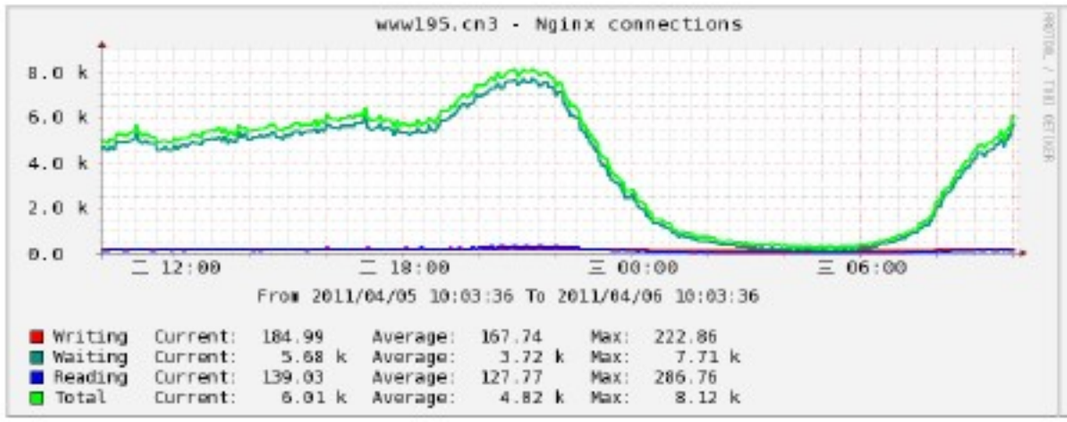
- 目的是做安全过滤如
  - 防hashdos攻击
  - 防SQL注入
  - 防XSS
- 标准Nginx无输入体过滤器机制的问题
  - 如果所有POST内容都在内存中，占用内存过大
  - 否则性能不高，内容可能被buffer到磁盘
- 例子（防hashdos攻击）
  - http://blog.zhuzhaoyuan.com/2012/01/a-mechanism-to-help-write-web-application-firewalls-for-nginx/

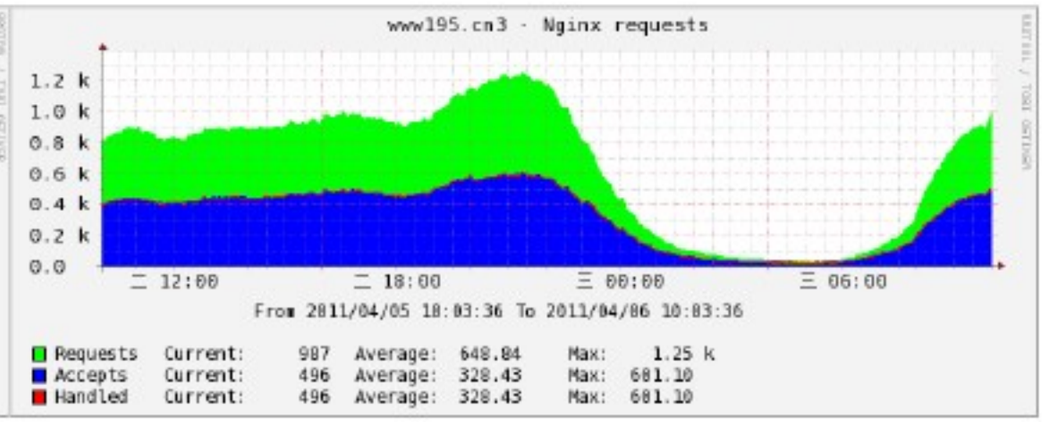
# Tengine中命令行参数的增加

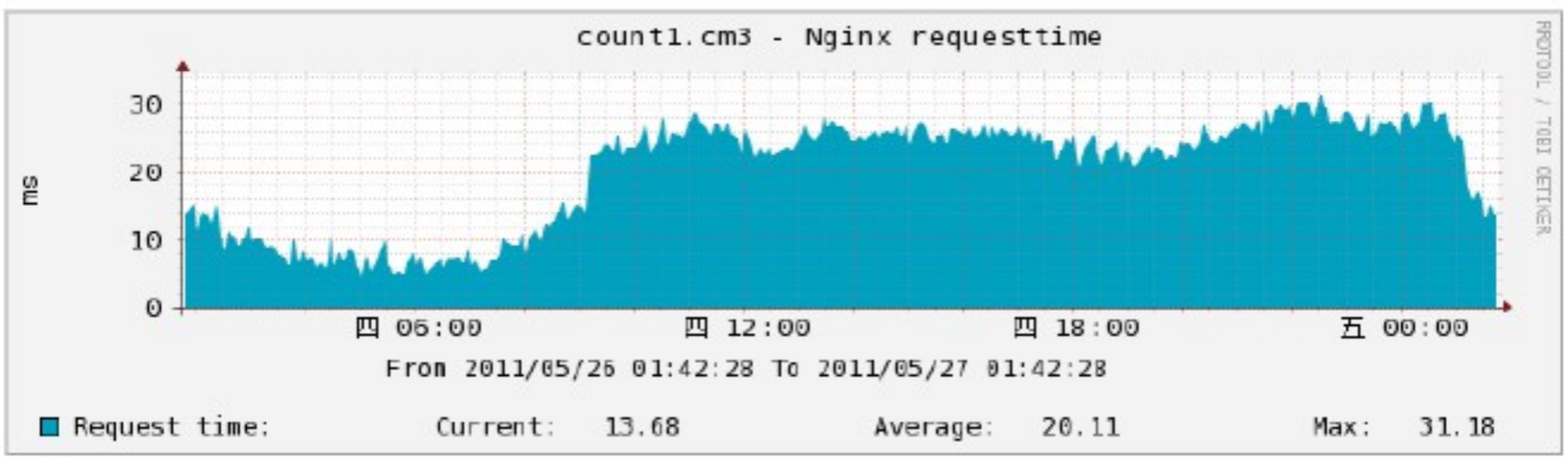
- 列出已经编译的模块
  - nginx -m
- 列出支持的指令
  - nginx -l
- 输出配置文件的全部内容
  - nginx -d
  - 支持include的内容

# Nginx监控增强

- 可集成到统计工具如Cacti中
- Tengine增加响应时间统计



# 实时监控工具Tsar

- tsar --nginx

| Time | nginx | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| Time | accept | handle | reqs | active | read | write | wait | qps | rt |
| 25/03-09:10 | 224.5K | 224.5K | 512.4K | 14.5K | 228.0 | 5.0 | 14.3K | 1.7K | 18.2 |
| 25/03-09:15 | 228.6K | 228.6K | 515.7K | 14.6K | 210.0 | 2.0 | 14.4K | 1.7K | 18.0 |
| 25/03-09:20 | 231.6K | 231.6K | 525.4K | 15.1K | 232.0 | 3.0 | 14.9K | 1.8K | 20.2 |
| 25/03-09:25 | 236.7K | 236.7K | 536.7K | 15.2K | 202.0 | 3.0 | 15.0K | 1.8K | 20.9 |
| 25/03-09:30 | 238.2K | 238.2K | 536.6K | 15.3K | 231.0 | 3.0 | 15.1K | 1.8K | 20.3 |
| 25/03-09:35 | 239.8K | 239.8K | 537.0K | 15.3K | 213.0 | 4.0 | 15.1K | 1.8K | 19.8 |
| 25/03-09:40 | 227.2K | 227.2K | 505.5K | 14.0K | 192.0 | 1.0 | 13.8K | 1.7K | 21.2 |
| 25/03-09:45 | 227.2K | 227.2K | 505.5K | 1.0 | 0.0 | 1.0 | 0.0 | 1.7K | 21.2 |
| 25/03-09:50 | 206.7K | 206.7K | 366.2K | 10.2K | 236.0 | 1.0 | 9.9K | 1.2K | 19.4 |
| 25/03-09:55 | 261.1K | 261.1K | 478.5K | 10.6K | 252.0 | 3.0 | 10.4K | 1.6K | 21.2 |
| 25/03-10:00 | 268.8K | 268.8K | 496.4K | 11.1K | 270.0 | 2.0 | 10.8K | 1.7K | 23.4 |
| 25/03-10:05 | 278.5K | 278.5K | 509.3K | 11.2K | 250.0 | 3.0 | 11.0K | 1.7K | 24.5 |
| 25/03-10:10 | 283.9K | 283.9K | 512.2K | 11.5K | 257.0 | 7.0 | 11.2K | 1.7K | 23.2 |
| 25/03-10:15 | 283.0K | 283.0K | 509.6K | 11.3K | 268.0 | 2.0 | 11.0K | 1.7K | 22.9 |
| 25/03-10:20 | 285.7K | 285.7K | 510.0K | 11.4K | 291.0 | 2.0 | 11.1K | 1.7K | 21.6 |
| 25/03-10:25 | 285.4K | 285.4K | 509.7K | 11.3K | 282.0 | 5.0 | 11.1K | 1.7K | 24.1 |
| 25/03-10:30 | 286.7K | 286.7K | 512.1K | 11.4K | 276.0 | 5.0 | 11.2K | 1.7K | 25.7 |
| 25/03-10:35 | 288.3K | 288.3K | 517.3K | 11.4K | 244.0 | 1.0 | 11.2K | 1.7K | 25.8 |
| 25/03-10:40 | 290.9K | 290.9K | 515.7K | 11.7K | 319.0 | 2.0 | 11.4K | 1.7K | 24.7 |
| Time | nginx | | | | | | | | |

# 其他

- Slice模块
- SSL的key加密（dialog）
- Jemalloc库的支持
- 崩溃时打印堆栈
- 更多内容请参考：
  http://tengine.taobao.org

# 4、当前工作

# 即将发布的功能

- Timer优化：红黑树->四叉最小堆
- 防慢攻击支持
- 一致性hash模块
- Session sticky模块
- 更强的统计模块

# 正在开发中的功能

- 上传buffer机制改进
  - 避免将文件缓存到磁盘文件
- Nginx的远程管理工具，包括监控，远程控制，配置同步等功能
- Pagespeed模块移植

# 关于Tengine的后续发展

- 国内多个公司在使用tengine：土豆、56、PPTV、小米
- 同多个公司合作开发：CloudFlare、搜狗、网易、去哪儿
- 开发过程已经完全透明化
  - http://github.com/taobao/tengine
  - github 上面通过pull request进行代码review
- 社区化发展

# 与Nginx官方协同发展

- 与Nginx进行合作，翻译Nginx中文文档，征求志愿者
- 为Nginx提供若干bugfix
- 内部测试SPDY协议
- 写一本Nginx的书籍：Nginx开发从入门到精通

# 参考资源

- 本演示稿中涉及的大部分技术已经开源：
  - http://tengine.taobao.org
  - https://github.com/taobao/tengine

# Thank You!

- Q & A