



淘宝网Nginx应用、定制与开发实战

朱照远 (@淘叔度)

王晓哲 (@chaoslawful)

2012-06-09

大纲

- 背景介绍
- 应用案例分析
- 开发与定制
- Web应用开发
- 当前工作

1、背景介绍

Nginx简介

- Web服务器、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
- 全球使用量排名第二
- 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
 - 高度模块化，易于扩展

NGINX™

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的内核
 - 启动[Tengine](#)项目并开源

The logo for Tengine, featuring the word "Tengine" in a stylized, blue, italicized font with a 3D effect and a shadow.

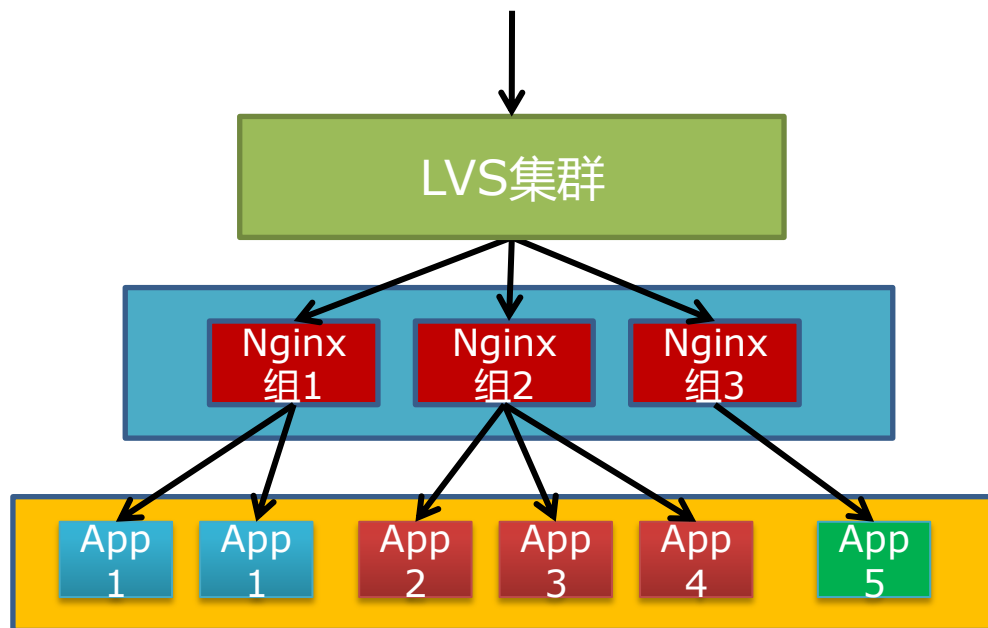
淘宝网应用Nginx的收益

- 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，更不会吃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

2、应用案例分析

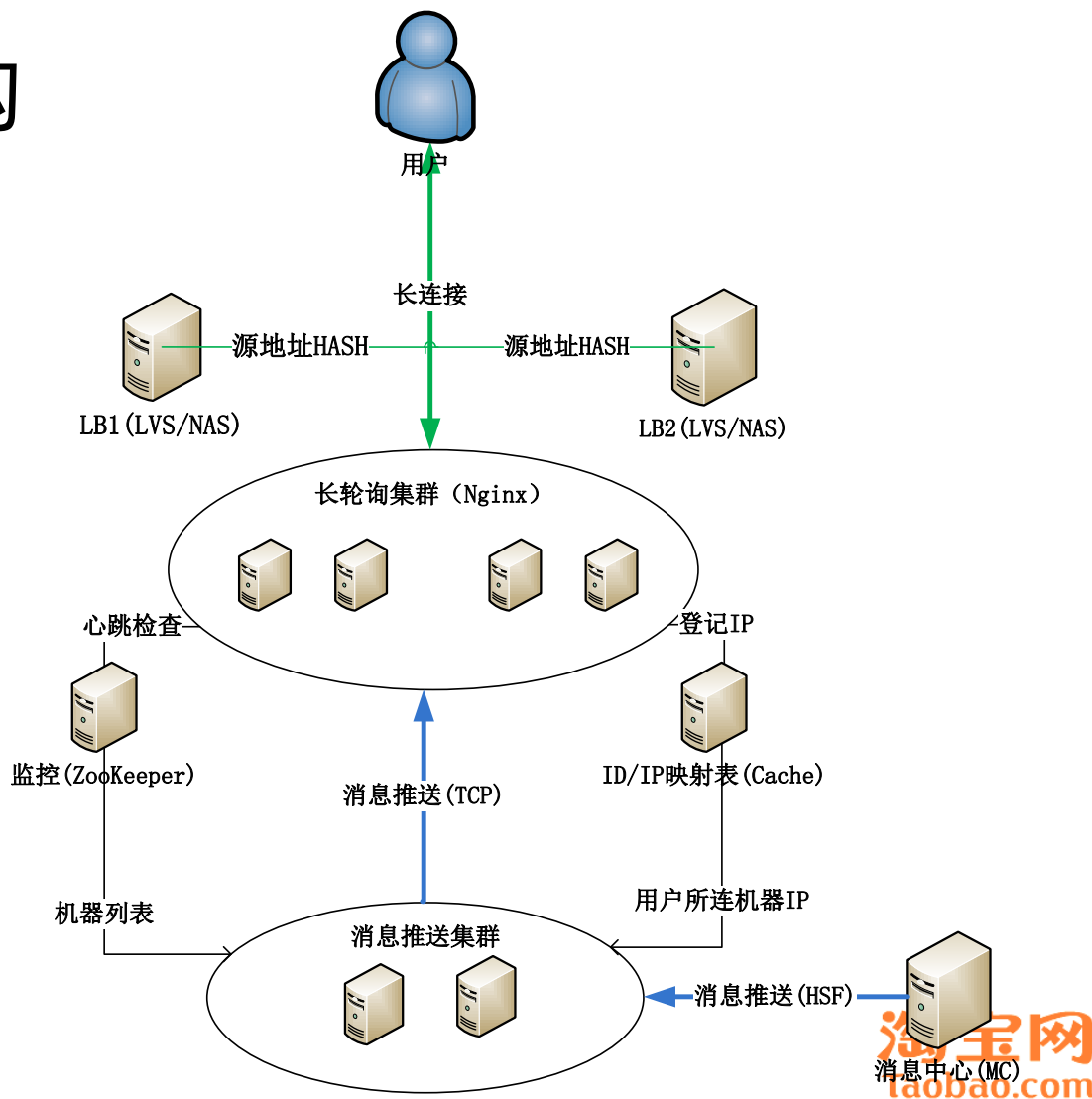
Web接入层

- Nginx的职能
 - 负载均衡
 - SSL卸载
 - 管理接口
 - 安全防御
 - 灰度发布
 - 静态化



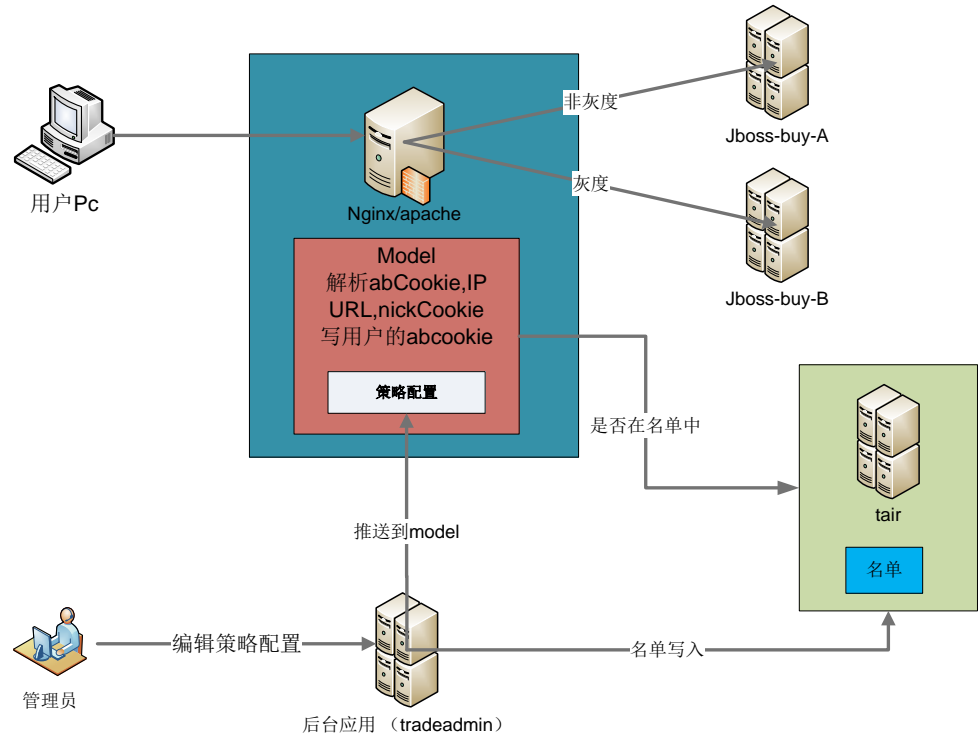
大用户群消息推送

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



发布与A/B Testing

- 灰度发布
 - 逐渐放量
 - 方便的管理接口
- 规则
 - IP
 - Cookie
 - K/V存储
 - ...



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

- [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

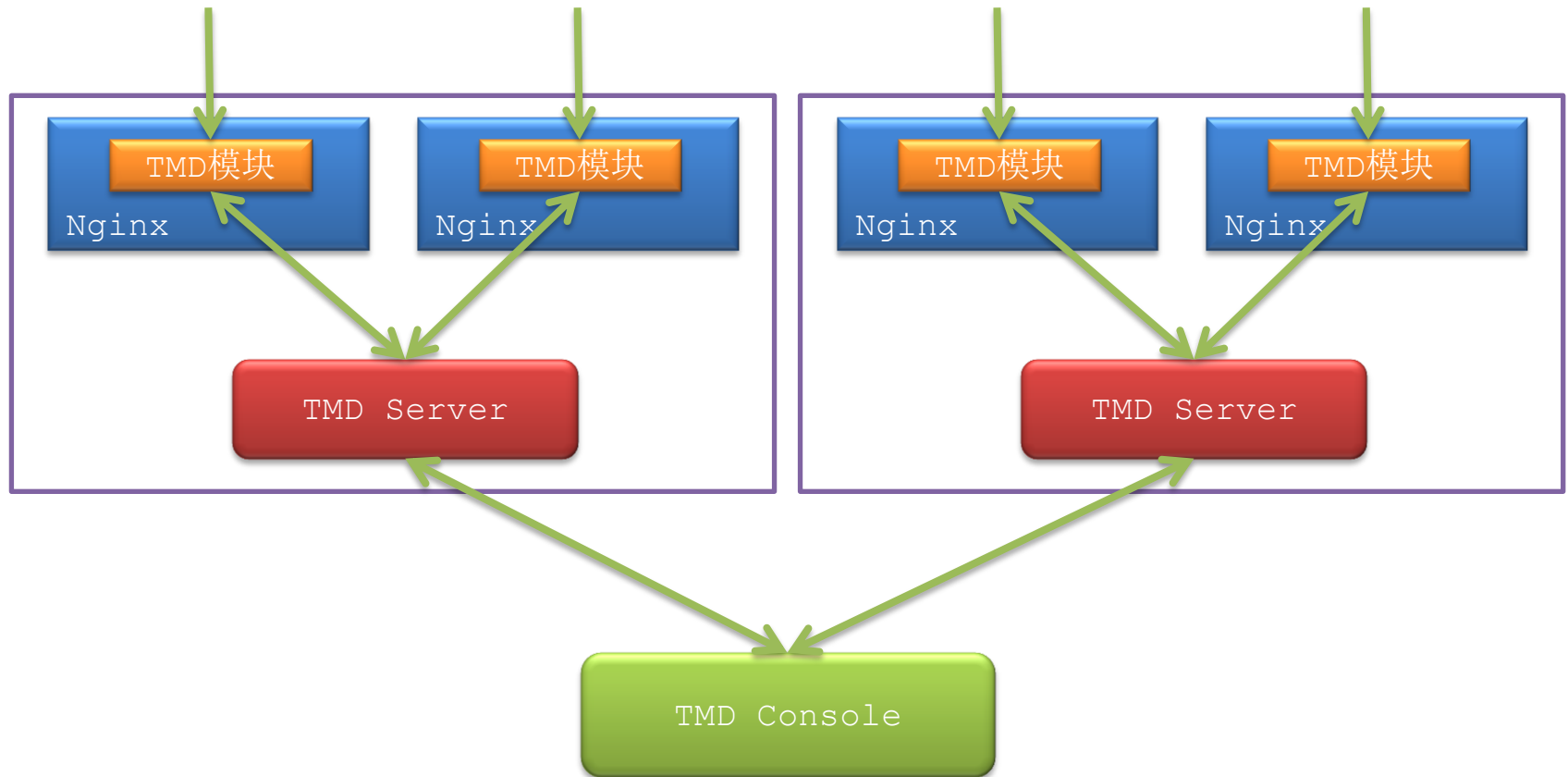
- 从而不必在服务器端渲染页面



分布式防攻击系统

- 应对的问题
 - 小型的DDoS攻击
 - 恶意的爬虫
- 为什么单机版还不够
 - 单机版无法知道全局
- 淘宝TMD (Taobao Missile Defense) 系统
 - Nginx作为防攻击系统的终端
 - TMD Server做策略分析
 - TMD Console执行汇总和控制台

TMD系统架构图



3、开发与定制

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

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=2011092320110301.js
```

系统过载保护

- 判断依据
 - 系统的loadavg
 - 内存使用（swap的比率）
 - QPS
- sysguard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;
```

主机信息调试

- Tengine的footer模块

```
footer $host_comment;
```

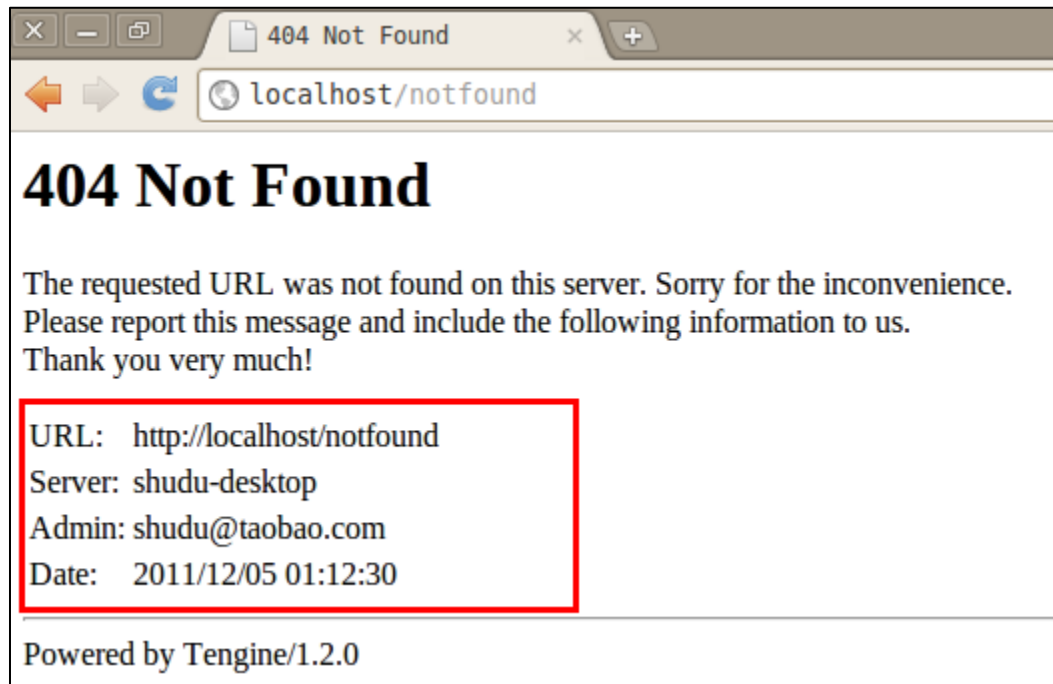
- 输出效果

```
$ curl http://localhost
<html>
<head>
<title>Welcome to nginx!</title>
</head>
<body bgcolor="white" text="black">
<center><h1>Welcome to nginx!</h1></center>
</body>
</html>
<!-- shudu-desktop Sat, 03 Dec 2011 09:27:47 GMT -->
```

Tengine错误信息提示

- 便于定位用户反馈的4xx和5xx错误

```
server_info    on;  
server_admin  shudu@taobao.com;
```



worker进程和CPU亲缘性

- 好处
 - 利用多核
 - 防止CPU的cache失效
- 问题
 - 不同的硬件，CPU核数可能不同
 - 绑定多核的CPU亲缘性比较繁琐

Tengine对于进程设置的简化

- 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模块同user_agent模块相比
 - 算法复杂度 $O(n^3)$
 - 不灵活
- 配置例子
 - <https://www.github.com/taobao/tengine/blob/master/conf/browsers>

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

- 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
- 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无输入体过滤器的问题
 - 如果所有POST内容都在内存中
 - 占用内存过大
 - 否则否则性能不高
 - 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/>

职能进程机制 (proc)

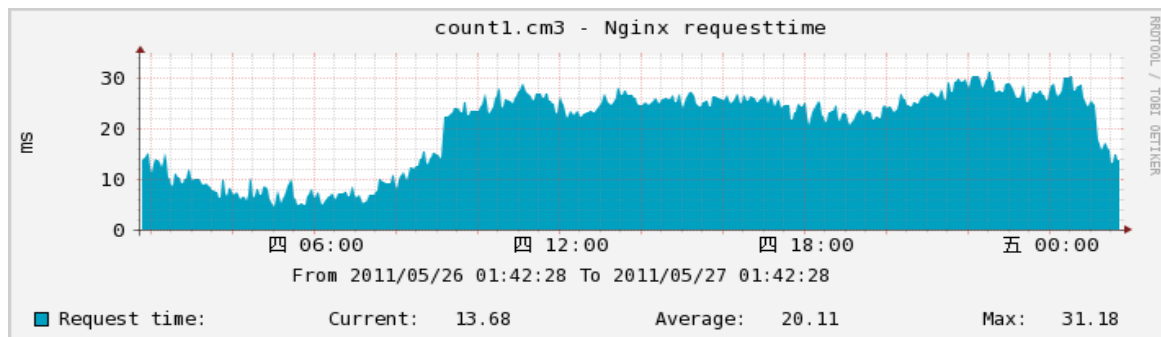
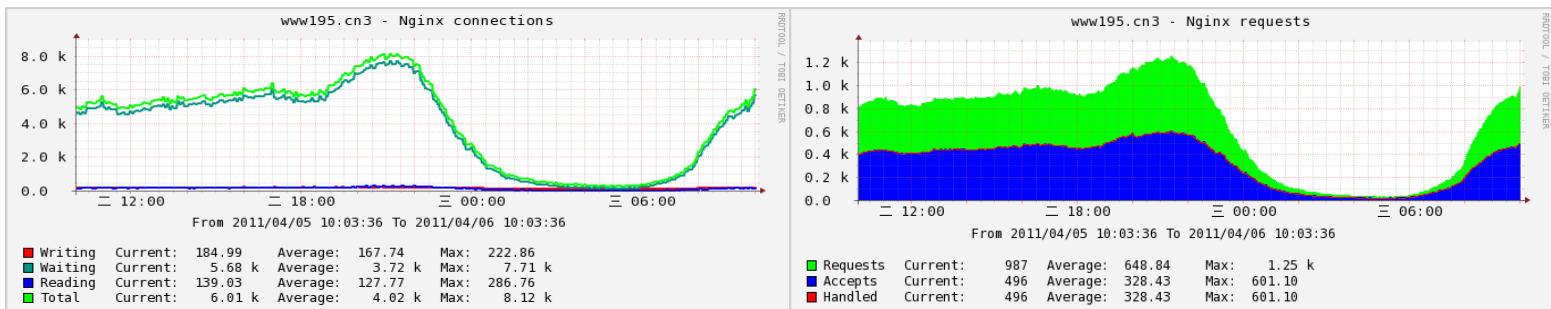
- 为了实现一些特殊用途的进程 (非HTTP)
- 可以实现单体进程的效果

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增加相应时间统计



实时监控工具Tsar

- tsar --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	3.0	14.5K	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

其他

- Slice模块
- error_page可恢复默认
- Server头可自定义
- expires_by_types
- SSL的key加密 (dialog)
- request_time_cache
- 崩溃打印堆栈
- 更多内容请参考：
http://engine.taobao.org/changelog_cn.html

4、Web应用开发

主要思想

- 引进动态脚本语言Lua
 - Lua语言很强大且简单
 - 适合嵌入
 - 支持协程 (coroutine)
 - ngx_lua
- 非阻塞的处理数据库层调用
 - ngx_drizzle
- 价值
 - 用同步的语义来实现异步的调用



ngx_lua

- Proactor模型
 - 业务逻辑以自然逻辑书写
 - 自动获得高并发能力
 - 不会因I/O阻塞等待而浪费CPU资源
- 每Nginx工作进程使用一个Lua VM
- 工作进程内所有协程共享VM
- 将Nginx I/O原语封装后注入Lua VM，允许Lua代码直接访问
- 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

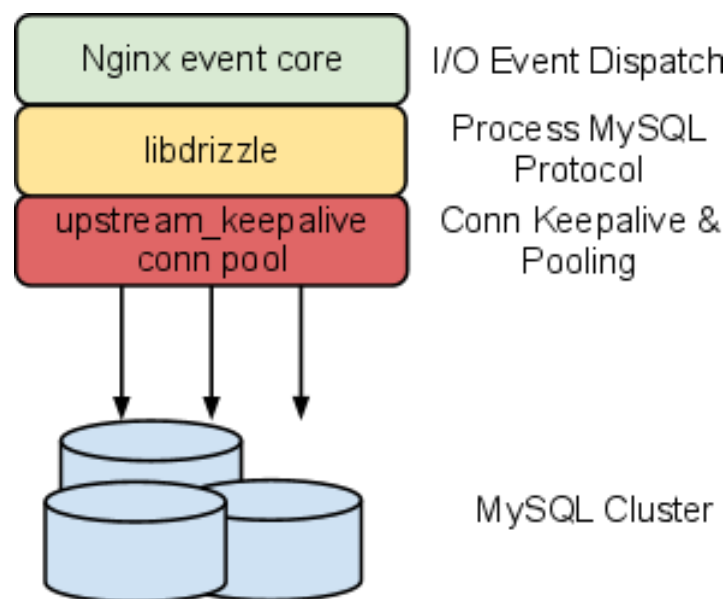
代码示例

```
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 occurred: rc=" .. res.status)  
        end  
    ;  
}
```

ngx_drizzle

- 实现Nginx中同步非阻塞方式访问MySQL
- 具备长连接、进程级可控大小连接池和负载均衡功能
- 返回数据可通过ngx_rds_json/csv等模块转换为JSON/CSV格式

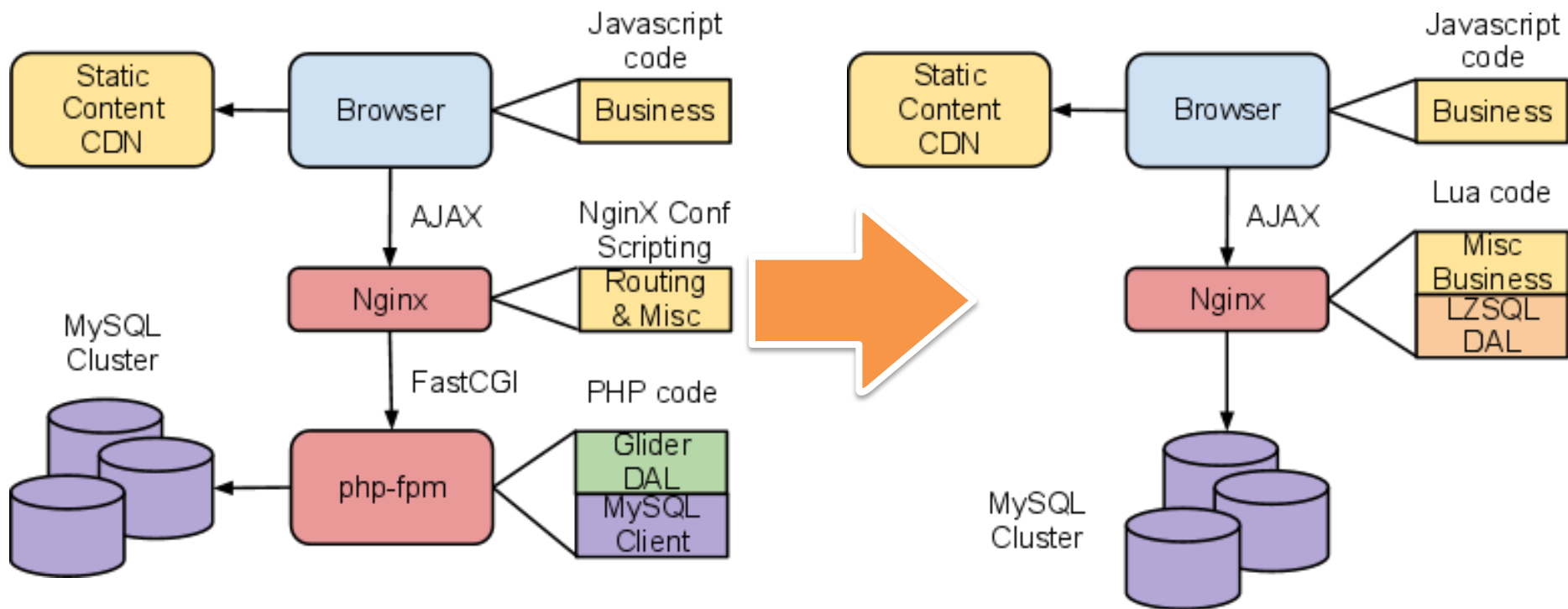


ngx_drizzle示例

```
http {  
    ...  
    upstream dbgroup {  
        drizzle_server host1:3306 dbname=test password=pass user=alice protocol=mysql;  
        drizzle_server host2:3306 dbname=test2 password=pass user=bob protocol=mysql;  
    }  
    ...  
    server {  
        location /mysql {  
            set $sql "select * from cats";  
            drizzle_query $sql;  
            drizzle_pass dbgroup;  
            rds_json on;  
        }  
    }  
}
```

ngx_lua + ngx_drizzle应用案例

- 淘宝量子业务架构变迁



5、当前工作

即将发布的功能

- 动态模块加载
- Timer优化
 - 红黑树->四叉最小堆
- 智能gzip
- 防慢攻击支持

正在开发中的功能

- 内存型HTTP Cache
- 上传buffer机制改进
 - 避免将文件缓存到磁盘文件
- 云支持

关于Tengine的后续发展

- 同国内多个公司合作
- 开发过程已经完全透明化
 - <http://github.com/taobao/tengine>
- 社区化发展

参考资料

- 本演示稿中涉及的大部分技术已经开源：
 - <http://engine.taobao.org>
 - <https://github.com/taobao/engine>
 - <https://github.com/chaoslawful/lua-nginx-module>
 - <https://github.com/chaoslawful/drizzle-nginx-module>

Thank You!

- Q & A