OpenResty与高性能Web应用

尹吉峰

2019/8/31

自我介绍

- 2010年北航软件工程毕业
  - Web开发背景，FullStack/FullCycle工程师
  - 酷爱原型开发，偏向异步函数式 编程
  - 喜欢把玩工具，语言中立者
- 2014年底加入链家网，使用OR构建了3个Web应用
  - image.lianjia.com (图片水印服务第一版)
  - dig.lianjia.com (Web Beacon)
  - t.ke.com (短链接服务)
- 目前在寻找新的工作机会（jifeng.yin@gmail.com）

如何打造单机QPS过万的服务？

Disclaimer:
过早优化是万恶之源

Why ?

不适合水平扩展的服务

- 有状态的服务
  - Cache：Redis
  - Queue: Kafka
  - DB：PG/MySQL
  - NoSQL: MongoDB/Elastic
  - Embedded: SQLite3/H2/Boltdb
  - TimeSeries: InfluxDB/Prometheus
- 基础服务
  - Gateway: Nginx
  - Logging / Tracing / ...
- 平台服务
  - 用户中心Session/Token
  - API认证授权

高性能的好处

- 水平扩展非线性
- 成本低，运维简单
  - 伸缩容不敏感
- 便于流量分发
  - 部署快 => 回滚快
  - 红绿部署
- 简化设计
  - 应用内缓存更高效
  - ABTest
- “程序员的自我修养”

How?

绝大多数Web应用：
IO密集 >> CPU密集

CPU性能:

π 秒 ≈ 纳世纪

异步编程:
IO操作 => CPU操作

我眼中的:

同步模型 = 线程池 + 上下文切换 (+ 锁)

异步模型 = 高并发

异步 + 缓存 = 高性能

异步语言和框架

- C / C++
- PHP (swoole)
- Java (Reactor/RxJava with Netty)
- JavaScript/NodeJS (callback -> Promise -> async/await)
- Python (yield [send & from]/ asyncio)
- Rust (async/await stable in 1.39.0)
- Golang
- ...

OpenResty comes to Rescue ~2015

Why OpenResty?

Nginx

- 一流的Reverse Proxy
- 开源生态 + 官方商业支持
- 易用的NginxScript, 已有学习成本
- 天然的架构设计:
  - Event Driving
  - Master-Worker Model
  - URL Router
  - Processing Phases

Lua

- 小巧灵活的编程语言
- C亲和
- 支持Coroutine
- LuaJit
  - 性能强大
  - 支持FFI

OpenResty = Nginx + Lua(JIT)

OpenResty应用实践

Web Beacon服务

- dig.lianjia.com
  - 接受HTTP请求，返回1x1的gif图片
  - Universal Access Logging
    - time/url/method/ua/referer/...
    - uuid/udid/ucid/ssid/reqid/...
    - 拆解POST聚合请求
  - 推送到Kafka
    - Streaming Process
    - OLAP

Prior Art

- 第一版由PHP完成
  - FastCGI
  - PHP Logging to file
  - rsyslog imfile
  - rsyslog omkafka

挑战

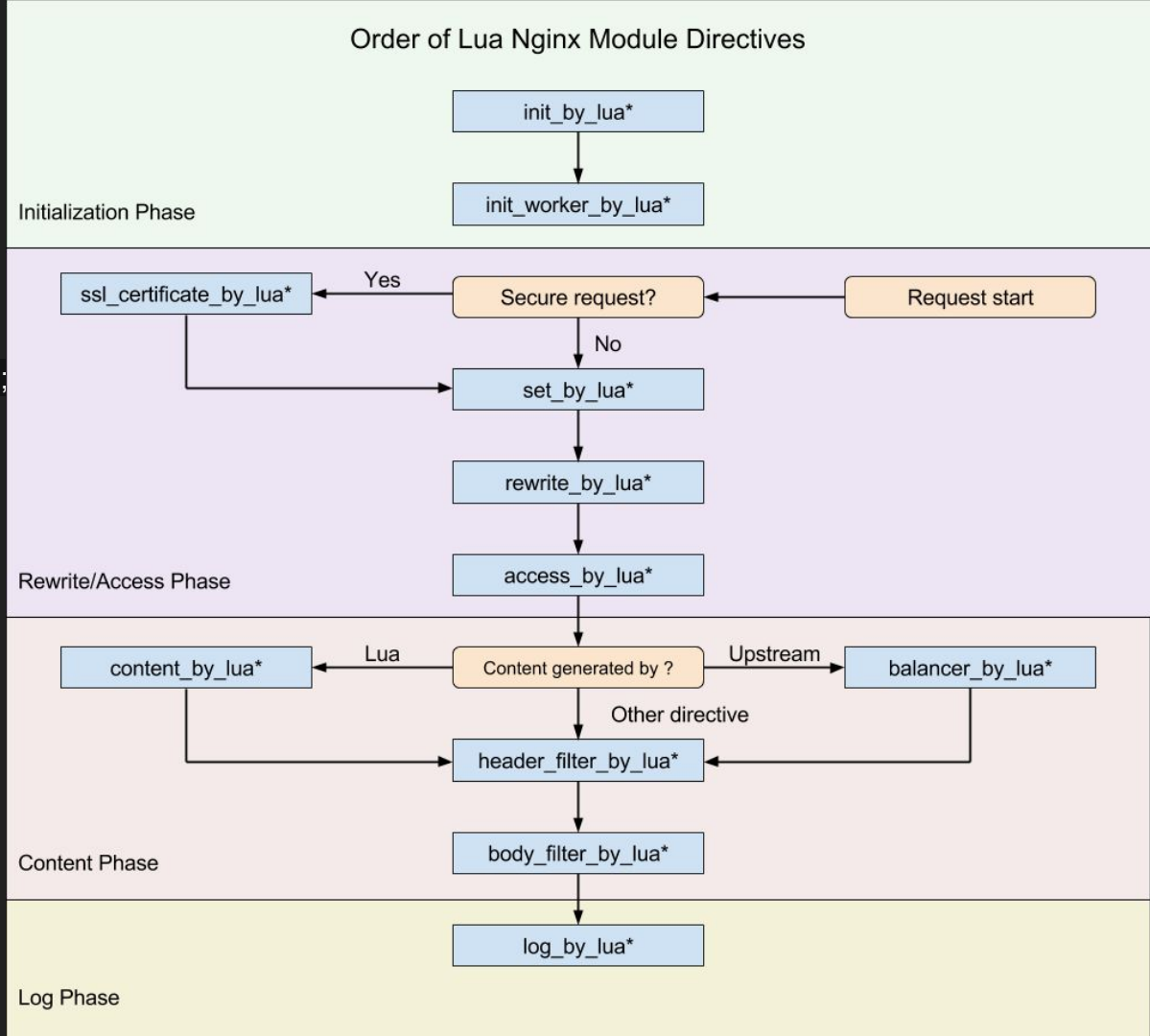
- 重要程度高
- 高吞吐
- 有业务
- 资源隔离性差

原则

- 性能Max
- 避免对磁盘的(硬)依赖
- 即刻响应用户
- 重构成本尽可能低

主体逻辑

```
location = /t.gif {
  access_by_lua_file .../access.lua;
  content_by_lua_file .../content.lua;
  header_filter_by_lua_file .../header_filter.lua;
  log_by_lua_file .../log.lua;
}
```



content_by_lua

```
-- return 1x1 gif by default
-- http://www.xarg.org/2011/01/create-a-simple-and-small-gif-with-php/

ngx.header["Content-Length"] = 43

ngx.print("\x47\x49\x46\x38\x39\x61\x01\x00\x01\x00\z
 \x90\x01\x00\xff\xff\xff\x00\x00\x00\x21\xf9\x04\x01\z
 \x00\x00\x01\x00\x2c\x00\x00\x00\x00\x01\x00\x01\x00\z
 \x00\x02\x02\x4c\x01\x00\x3b");
```

access_by_lua

- 解析cookie（cloudflare/lua-resty-cookie）
- 生成uuid (openssl over libuuid)
  - C.RAND_bytes 16 (= 128 bit)
  - C.ngx_hex_dump
  - string.format("%s-%s-%s-%s-%s", …)
- 更新ssid
  - 30分钟内持续刷新
  - 不跨自然天
- 解析body
  - 避免落盘：client_{body_buffer|max_body}_size 64k;
  - 解码：zlib_inflate -> ngx.decode_args -> json_decode -> `list`
  - 编码：table.new(#list, 0) -> table.new(0, 30)/table.clear-> json_encode -> ngx.escape_uri
  - 降级：log_escape

header(body)_filter_by_lua

- 解析X-Forwarded-For
- 解析　　lianjia_token
  - ffi.new(“uint64_t”, 0):tostring():sub(1, -4)
  - shared_dict 缓存
- 汇总所有字段到ngx.ctx

log_by_lua备选方案

- access_log
  - TB sized and rotating
  - Disk is over-utilized
  - Read/Write is blocking
- doujiang24/lua-resty-kafka
  - Not investigated
- cloudflare/lua-resty-logger-socket
  - Nonblocking
  - Remoting

rsyslog

- MaxMessageSize: 8k
- Queue: Disk-Assisted Memory
- Parser: RFC3164
- imptcp
  - dgram unix socket not supported
  - SupportOctetCountedFraming
- omkafka
  - dynaTopic: on
  - confParam: compression.codec=snappy
- omfile: backup
- impstats: monitor

```
return function(topic, msg, timestamp)
 -- 142 = 17 * 8 (local1 facility)  + 6 (info severity)
 local msg = "<142>"..timestamp.." "..topic..":"..msg
 -- octet-counted framing, and extra linefeed for debug
 local _, err = logger.log(#msg.." "..msg.."\n")
 if err then
   ngx.log(ngx.ERR, "failed to log message: ", err)
   return
 end
end
```

部署方案

- 线上准入：tar包 + `run.sh`
  - OpenResty: precompiled with static libraries
  - Bundle everything and rsync to production server
  - supervisord / systemctl
- 测试环境：“程序员的自我修养”
  - Ansible Provision

性能数据

- 2018年初服务状态
  - QPS
    - 峰值:26K
    - 单机压测: 30K
  - 日志:
    - 传输:~30M/s (snappy)
    - 数量:~1B/d
    - 大小:~100T/d
  - 服务器:
    - 3台EC2 C3.2xlarge

移动端”协作”

- 状况:
  - Android & iOS
  - 客户端BUG, 服务端做兼容
- Cases:
  - 数据两次json encode
    - 对string额外做次decode
  - Cookie重复插入uuid, 一次有值, 一次为空
    - data=A&data=B
    - data[]=A&data[]=B
    - data=A,B
  - Body过大413

总结

- Architecture First
  - OpenResty + RSyslog + Kafka
  - “another level of indirection”
- Keep Boundary
  - No disk
- Little Things Count
  - Avoid NYI
  - table.new / ngx.now
  - uuid generation
  - shared dict
  - ...

Q & A