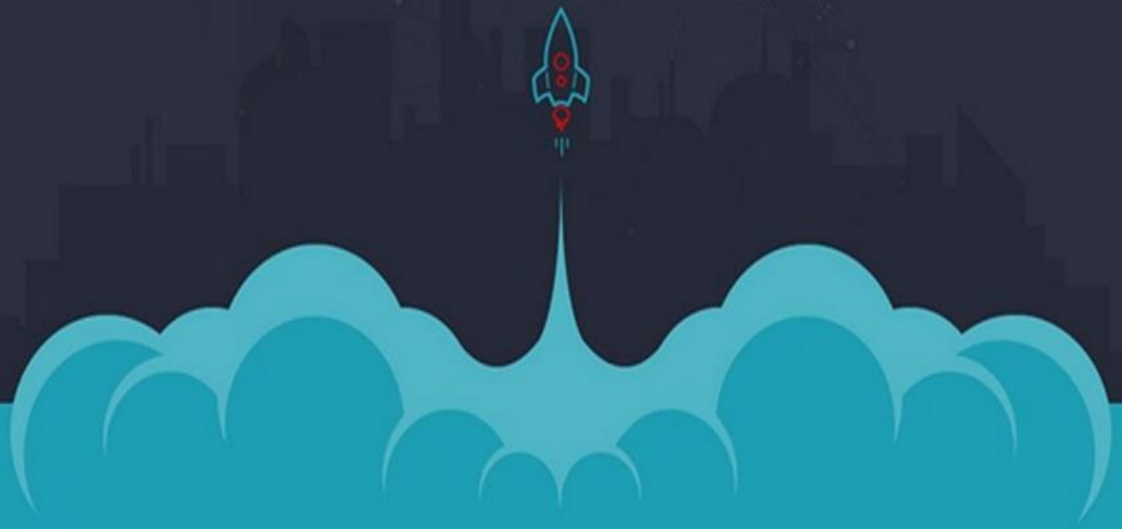
Nignx + Lua [+ Py]

实现高性能图片视频处理服务



速致 - 赖立维 (Jason)

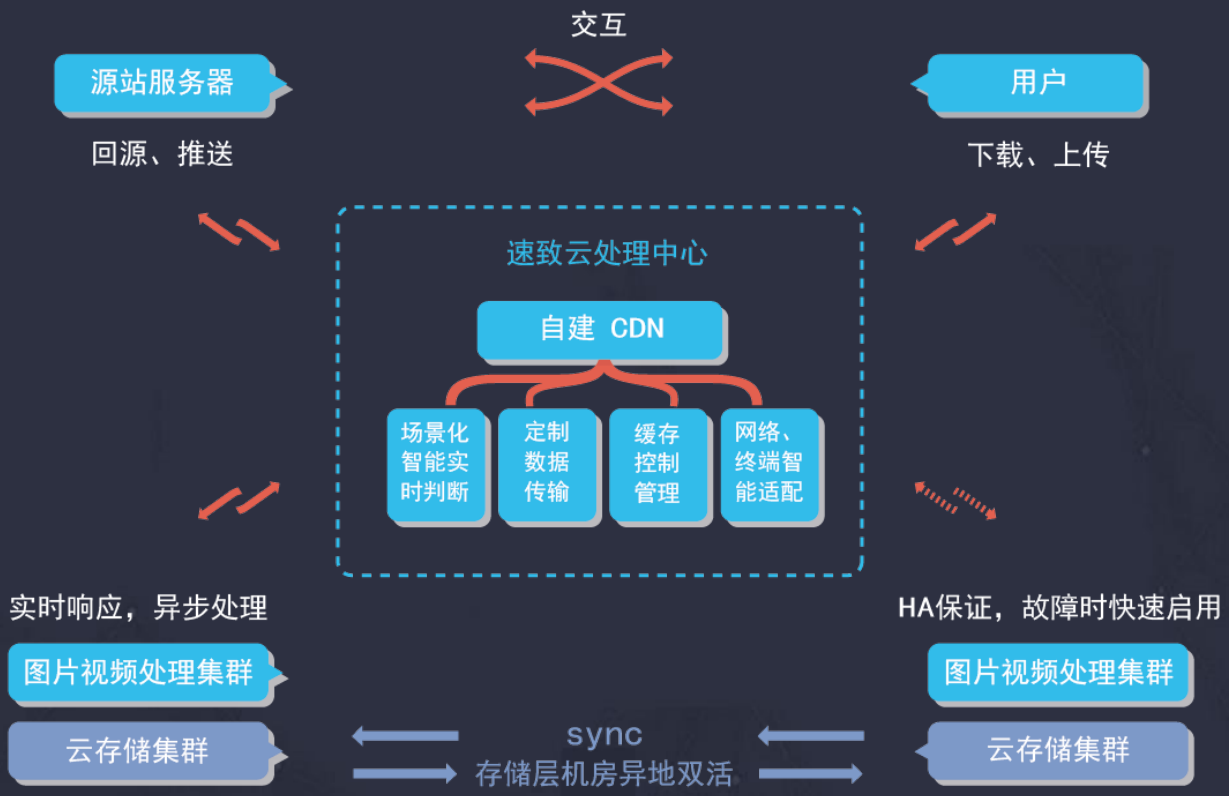
Github: JasonLai256

2015-11-12

技术变迁，业务变迁

- （2012）Vanish -> Nginx + Lua
- （2015）CDN -> 图片视频云处理

现有技术处理概况


交互
源站服务器
用户
回源、推送
下载、上传
速致云处理中心
自建 CDN
场景化智能实时判断
定制数据传输
缓存控制管理
网络、终端智能适配
实时响应，异步处理
HA保证，故障时快速启用
图片视频处理集群
图片视频处理集群
云存储集群
sync
存储层机房异地双活
云存储集群

Nginx + Lua + Python

- Lua -> 相对高效，最简实现 Backend-Nginx-Client 间的薄粘合层；
- Python -> 相对灵活，扛起麻烦的活、耗时的活，与各路英雄好汉通信打交道；

Nginx + Lua + Python

Why?

- Lua 性能很好，别的部件不一定
- 高并发、多逻辑处理情境下，很多机制需要优雅的降级处理
- 期望更好的异步处理相关事情
- 人的原因（对于 py 的积累较多）
- 事的原因（py 上面的单元测试机制较完善）

Nginx + Lua + Python

Lua 的高效

- 过往经验表明，正常情况下 Lua 端处理性能是杠杠的
- Lua 的协程很简洁高效
- Lua 与 Nginx 水乳交融，搭配合适的 ngx 配置，能在 http 代理转发时候做魔术般的处理

Nginx + Lua + Python

Python 的灵活

- 实时监测系统、网络的整体情况
- 实时检测 ngx 的各类日志
- 检测和处理实际的 ngx 缓存文件
- 与各类缓存、消息层进行交互（http api、zeromq 等）

Nginx + Lua + Python

消息传递?

- 基于 lua_shared_dict 缓存信息
- Nginx 暴露 API （监听 127.0.0.1）
- 日志输出处理信息
- Nginx 配置预/实时处理
- HTTP 的自定义头信息

Nginx-Lua 与 CDN

- Ngx_proxy 与 Lua 的结合很美妙
  - access_by_lua
  - header_filter_by_lua
  - body_filter_by_lua
- 传统 CDN 所需要的基本功能可以快速实现；
- 复杂处理解决思路，模块化、测试、测试、测试；

Nginx-Lua 与图片处理

配合处理

- 在 lua 层面判断各种条件情况，包括 UA、 URL Path、请求状态、各类头部等；
- 在 py 层面进行各类实际处理，包括 nginx 缓存处理、实际图片处理、图片信息分析等；
- 整体 ngx + lua + py 对整体 CDN + 处理集群（backend）进行动态调度和控制；

Nginx-Lua 与图片处理

成果

- 图片基本处理，缩略、水印等
- 自动图片判断优化，包括：webp自动适配处理、jpeg质量优惠、渐进式jpeg转换处理；

Nginx-Lua 与视频处理

上传优化

- 针对视频文件分片上传优化；
- 结合 CDN 节点网络；

整合存储

- 结合 ceph 进行文件落地，Request URL to 客户空间存储转换处理；
- 处理后资源下载与 CDN 的打通

Nginx-Lua 与测试

起始 -> 痛苦

Python 习惯的悲剧（比较2）：

```
access_by_lua '
    if condition1 then
        ngx.say("goto condition1")
    elif condition2 then
        ngx.say("goto condition2")
    else
        ngx.say("goto condition3")
    end
';
```

过渡 -> 舒适

重构，测试，重构，测试 … 良性循环

Nginx-Lua 与测试

自动化测试是正路

- Opsboy -> 启发 -> 简易 Python 版本实现
- 通过 Local API 黑盒测试内部功能实现
- Python 模拟 backend 收集回源数据
- Atc 线下模拟各类网络环境
- 不能不提火焰图（systemtap），性能测试佳品

END