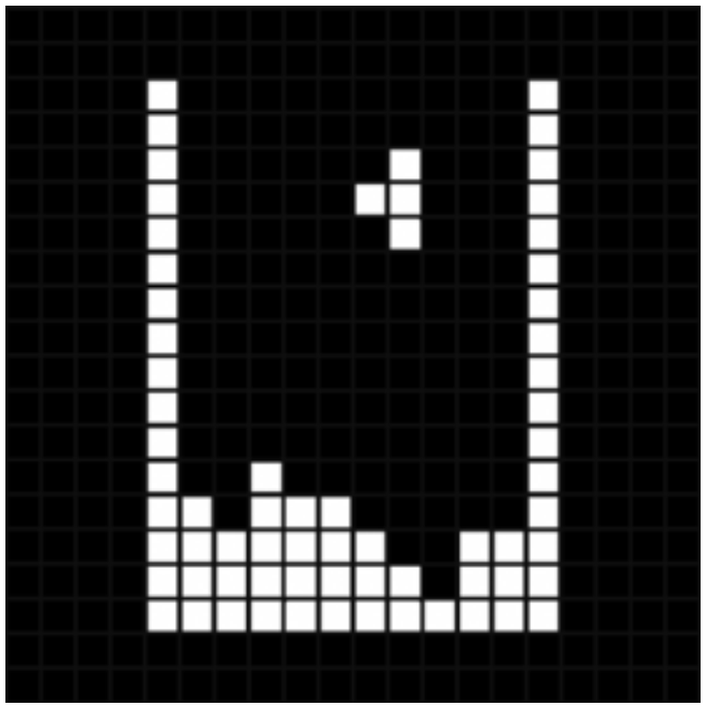
ngx_lua 在又拍云的应用

日志收集、性能调优与服务化

Monkey Zhang (timebug)

2016.07 @ Shenzhen ArchSummit

upyun



A Systems Engineer at UPYUN

- Email: timebug.info@gmail.com
- Github: https://github.com/timebug

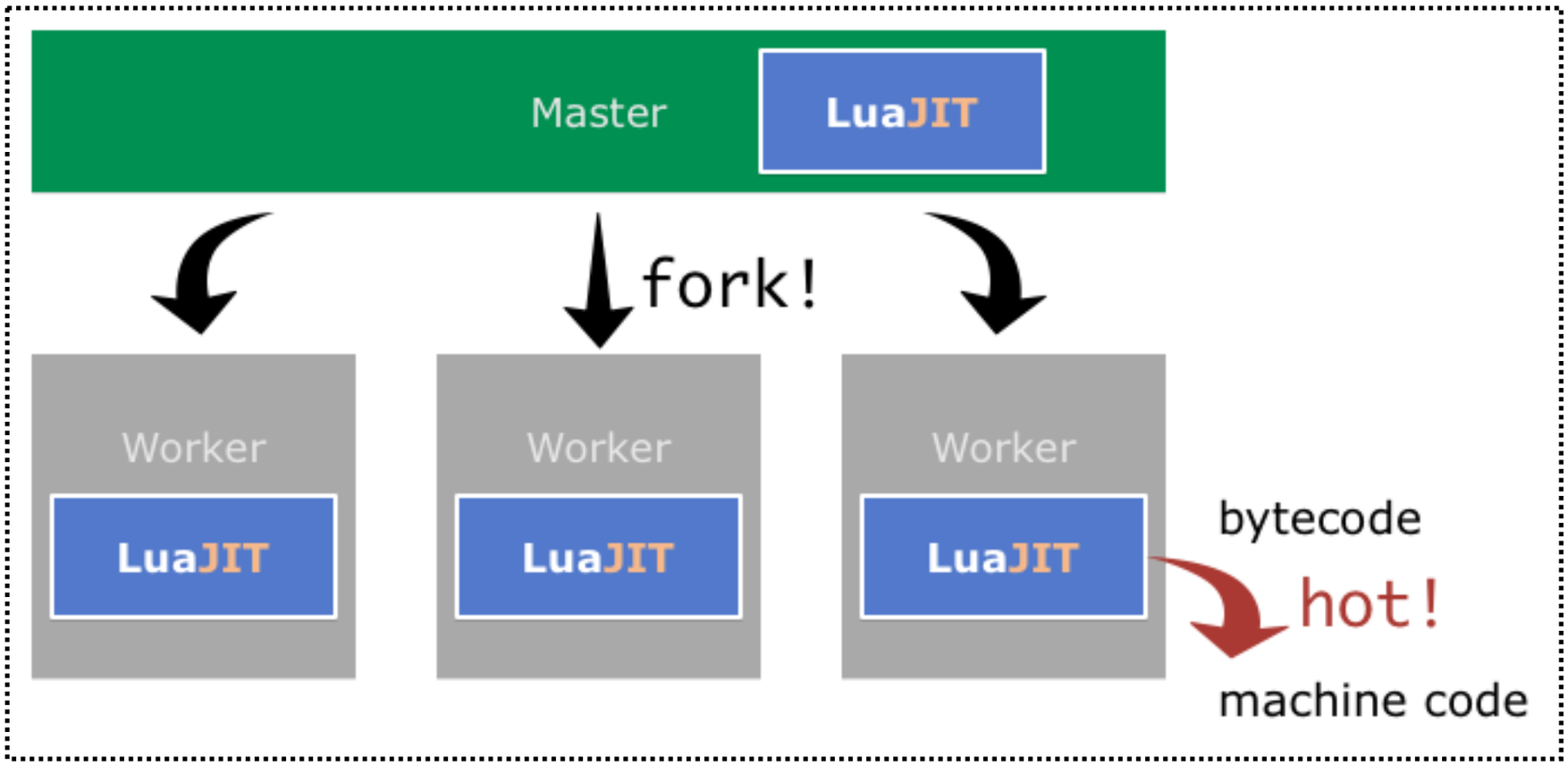

```
$ ./configure --prefix=/opt/nginx \
    --add-module=/path/to/lua-nginx-module
```

```
http {
    server {
        listen 8080;

        location /add {
            set $res '';

            rewrite_by_lua '
                local a = tonumber(ngx.var.arg_a) or 0
                local b = tonumber(ngx.var.arg_b) or 0
                ngx.var.res = a + b
            ';

            content_by_lua '
                ngx.say(ngx.var.res)
            ';
        }
    }
}
```

```
$ curl 'http://localhost:8080/add?a=6&b=7'
13
```

LuaJIT

LuaJIT is a Just-In-Time Compiler (JIT) for the Lua programming language.



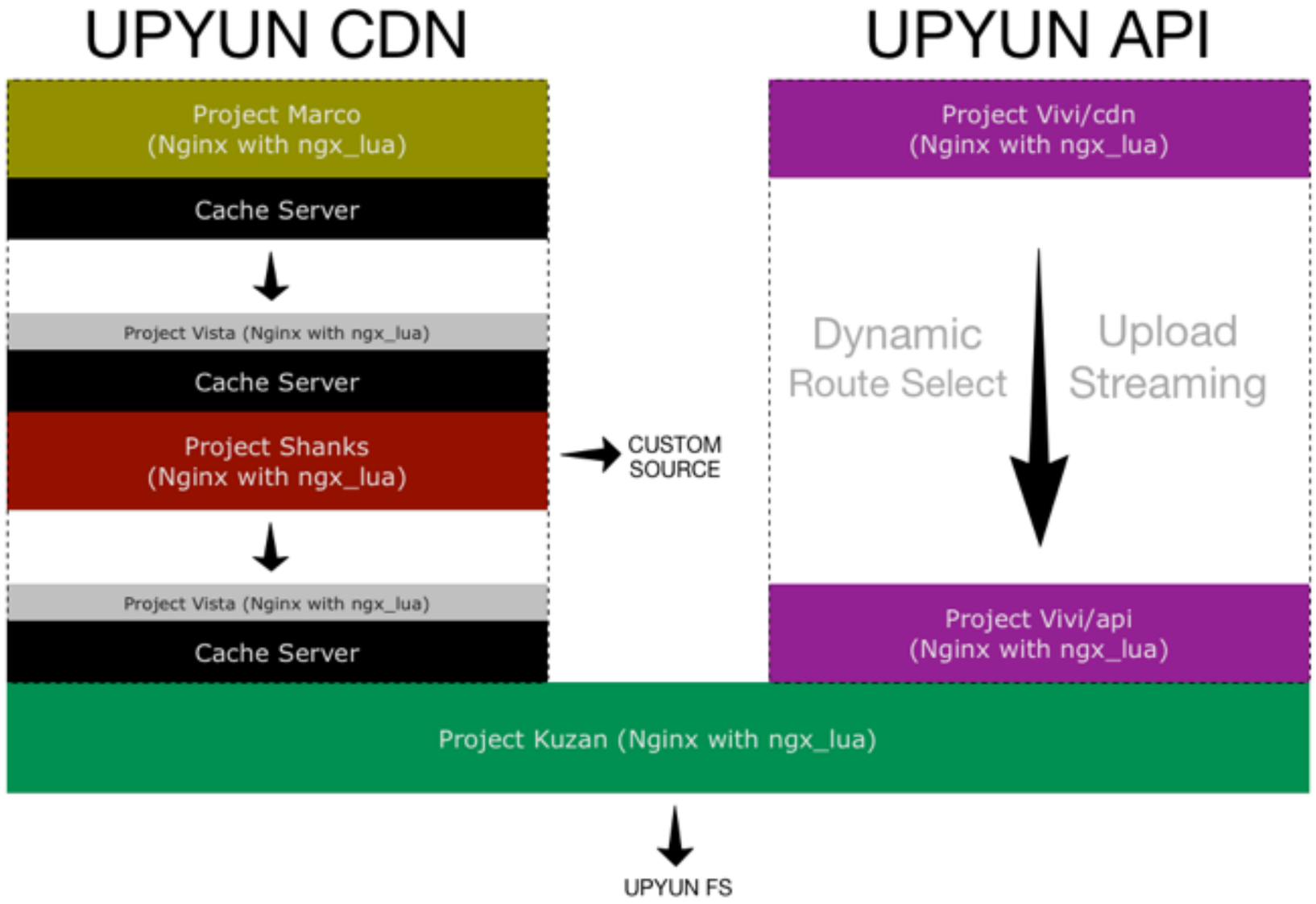
How it works

LuaJIT VM embedded into the Nginx

cosocket API

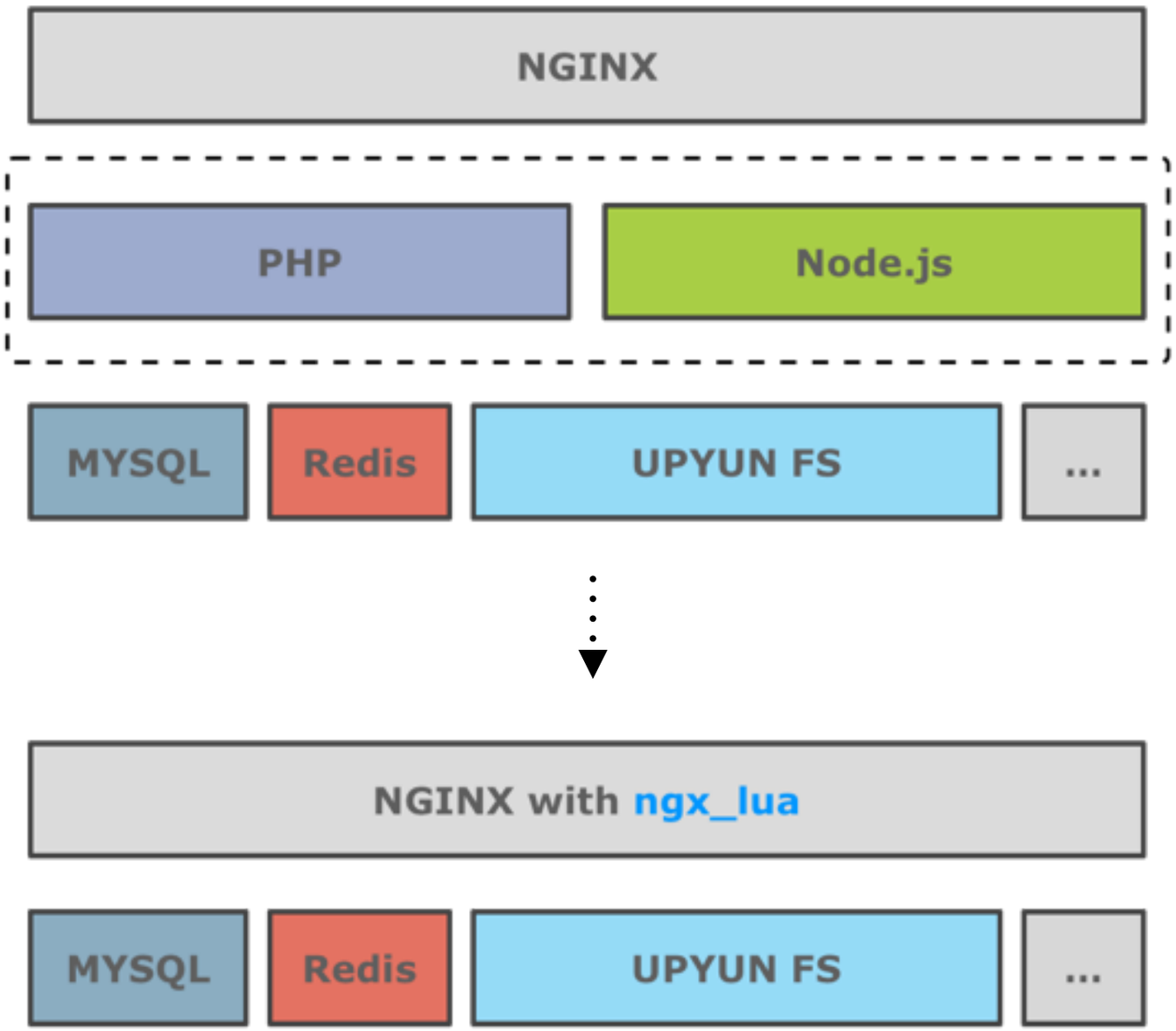
ngx.socket.* connect send receive

sslhandshake close settimeout etc.



UPYUN CDN & API is built on top of NGINX with ngx_lua

UPYUN API

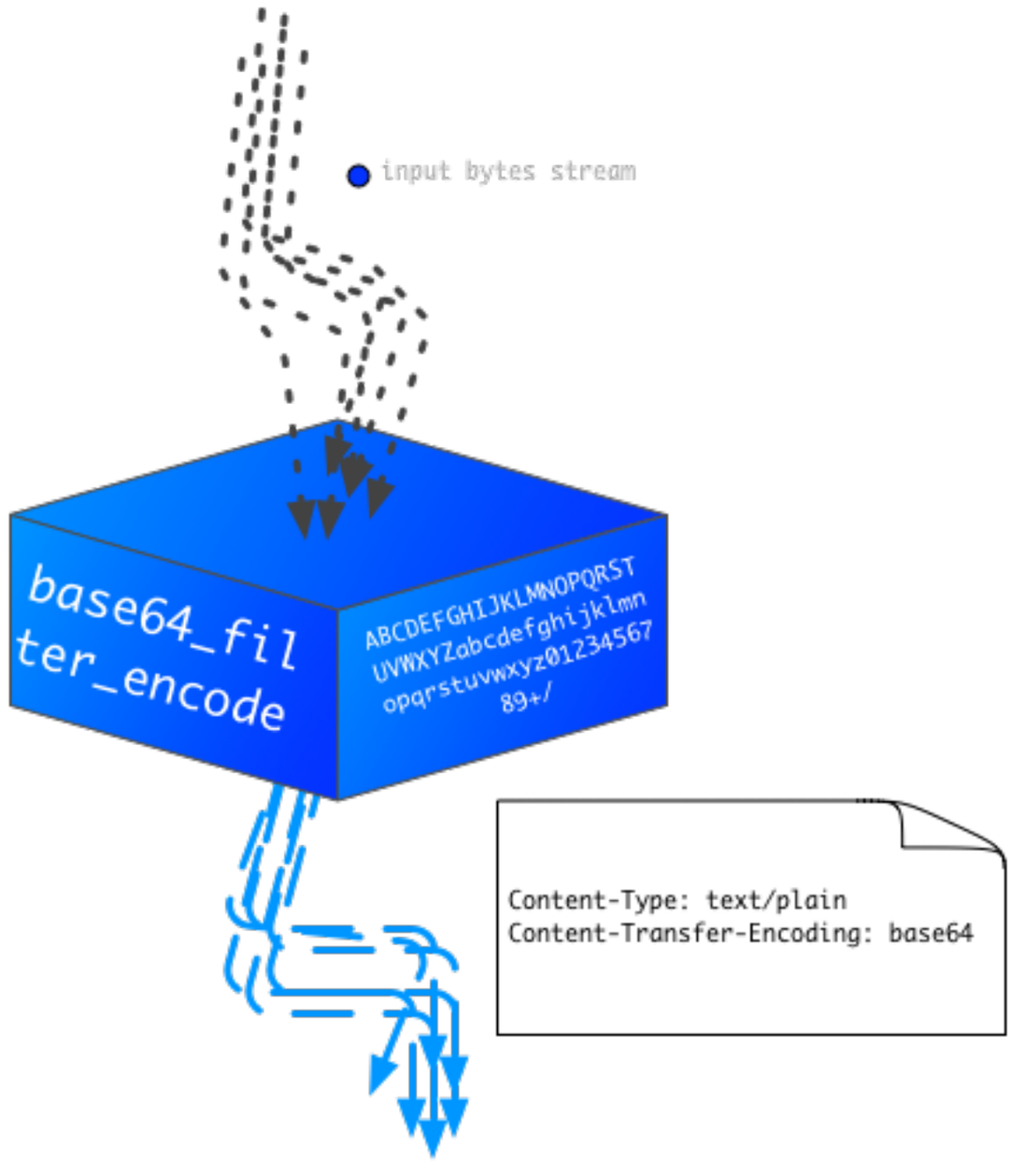

NGINX
PHP
Node.js
MYSQL
Redis
UPYUN FS
...
NGINX with ngx_lua
MYSQL
Redis
UPYUN FS
...

Base64 Filter by Lua

```
http {

    lua_package_path “$prefix/app/src/?.lua;;";

    server {
        listen 8080;

        location /base64 {
            set $b64_en '';
            set $b64_e0 '';
            set $b64_e1 '';

            echo_duplicate 1000 hello;

            header_filter_by_lua '
                ngx.header.content_length = nil -- ((n + 2) / 3 ) * 4
                ngx.header.content_type = "text/plain"
                ngx.header.content_transfer_encoding = "base64"
            ';

            body_filter_by_lua_file app/src/b64_body_filter.lua;
        }
    }
}
```

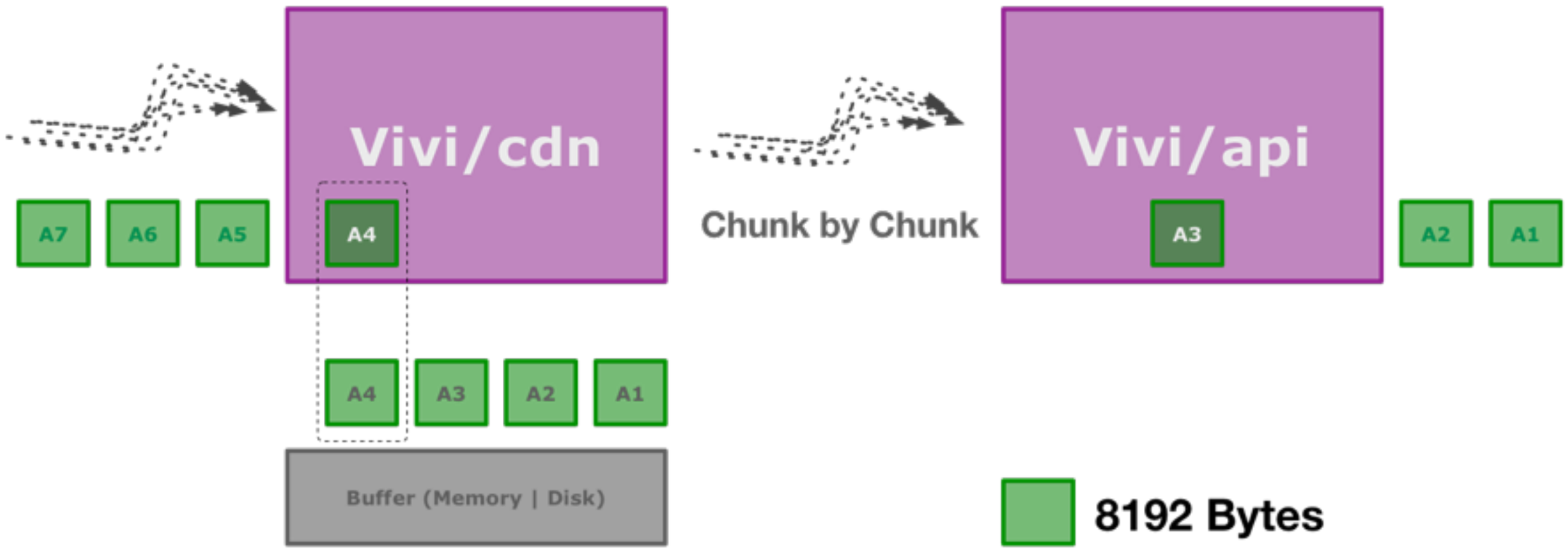
Base64 Filter by Lua: Chunk by Chunk

```lua
local chunk = ngx.arg[1]

local e0 = ngx.var.b64_e0 or ''
local e1 = ngx.var.b64_e1 or ''
local en = tonumber(ngx.var.b64_en) or 0

if en == 1 then
    chunk = e0 .. chunk
elseif en == 2 then
    chunk = e0 .. e1 .. chunk
end

if not ngx.arg[2] then
    en = #chunk % 3
    if en == 1 then
        e0 = chunk:sub(-1)
    elseif en == 2 then
        e1 = chunk:sub(-1)
        e0 = chunk:sub(-2, -2)
    end
    chunk = chunk:sub(1, #chunk - en)
else -- eof
    en = 0
end

ngx.var.b64_en = en
ngx.var.b64_e0 = e0
ngx.var.b64_e1 = e1

ngx.arg[1] = ngx.encode_base64(chunk)
```



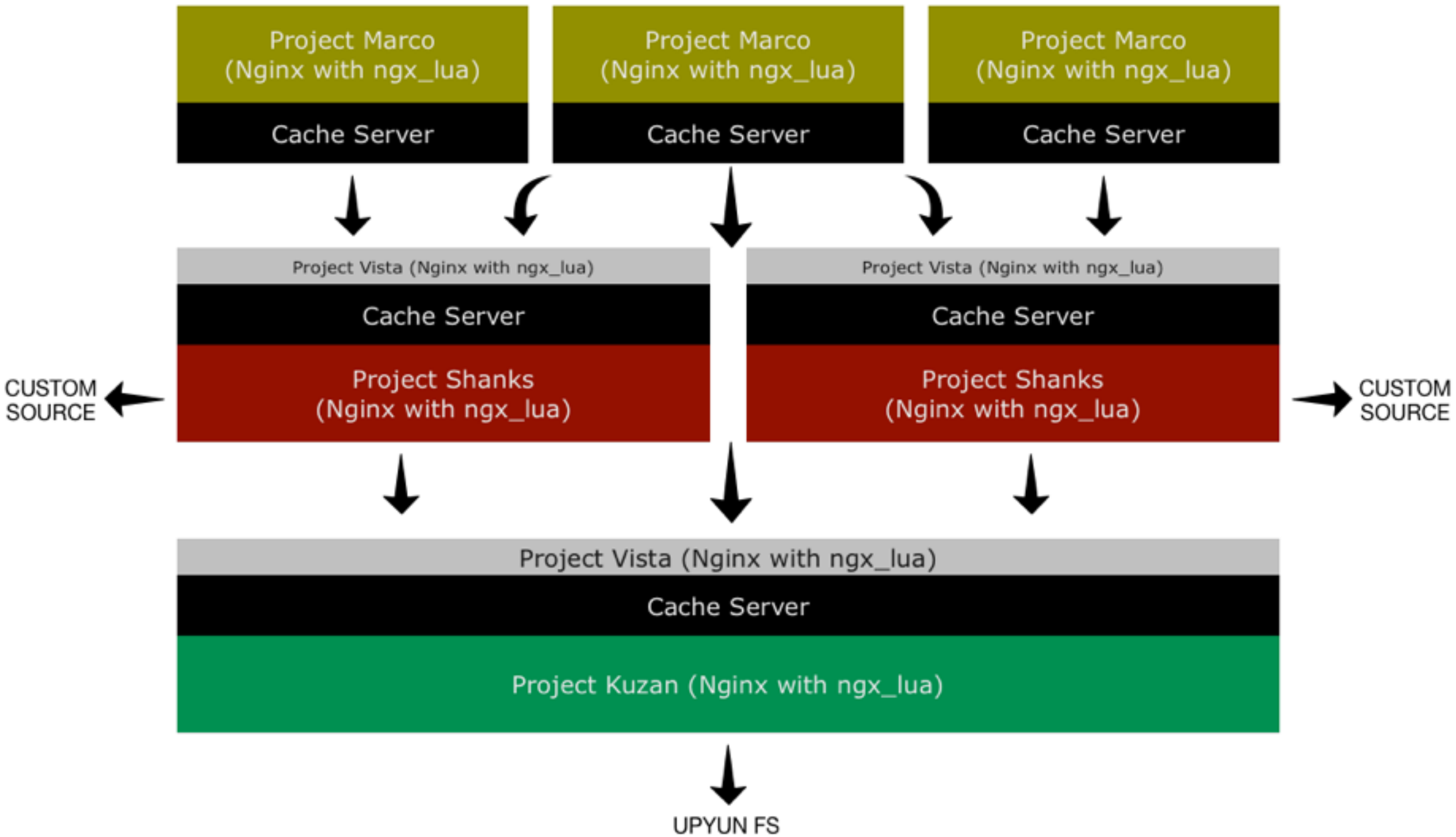
```
location /upload {
    proxy_request_buffering off;
    ...
}
```

Lua **Streaming** Upload




```
ngx.req.init_body()
ngx.req.append_body(chunk)
ngx.req.finish_body()
```

UPYUN CDN


Project Marco
(Nginx with ngx_lua)
Cache Server
Project Marco
(Nginx with ngx_lua)
Cache Server
Project Marco
(Nginx with ngx_lua)
Cache Server
Project Vista (Nginx with ngx_lua)
Cache Server
Project Shanks
(Nginx with ngx_lua)
CUSTOM
SOURCE
Project Vista (Nginx with ngx_lua)
Cache Server
Project Shanks
(Nginx with ngx_lua)
CUSTOM
SOURCE
Project Vista (Nginx with ngx_lua)
Cache Server
Project Kuzan (Nginx with ngx_lua)
UPYUN FS

全网 CDN **日志规模**有多大?

保守估计每天约 5 ~ 10T 原始日志。

日志的**多种用途**

- 日志每日归档；按不同服务名称提供下载。
- **日志实时分析**；供近实时多维分析、问题排查和监控报警。
- **日志聚合计算**；数据结果提前根据需求确定，实时后台展示。
- 日志离线分析；复杂的数据模型计算和分析，定期生成报表。

日志收集常见的几种方式

```
access_log /path/to/access.log combined buffer=4096 flush=5s;
```

- 传统运维脚本：日志文件**定时**或**定量**切割上报
- **Logstash** / **Heka** Agent：Input File 模块（tail -f）
- **NGINX** 1.7.1+： Logging to **syslog**
- 更多选择？以及日志一定要直接落到**磁盘**吗？

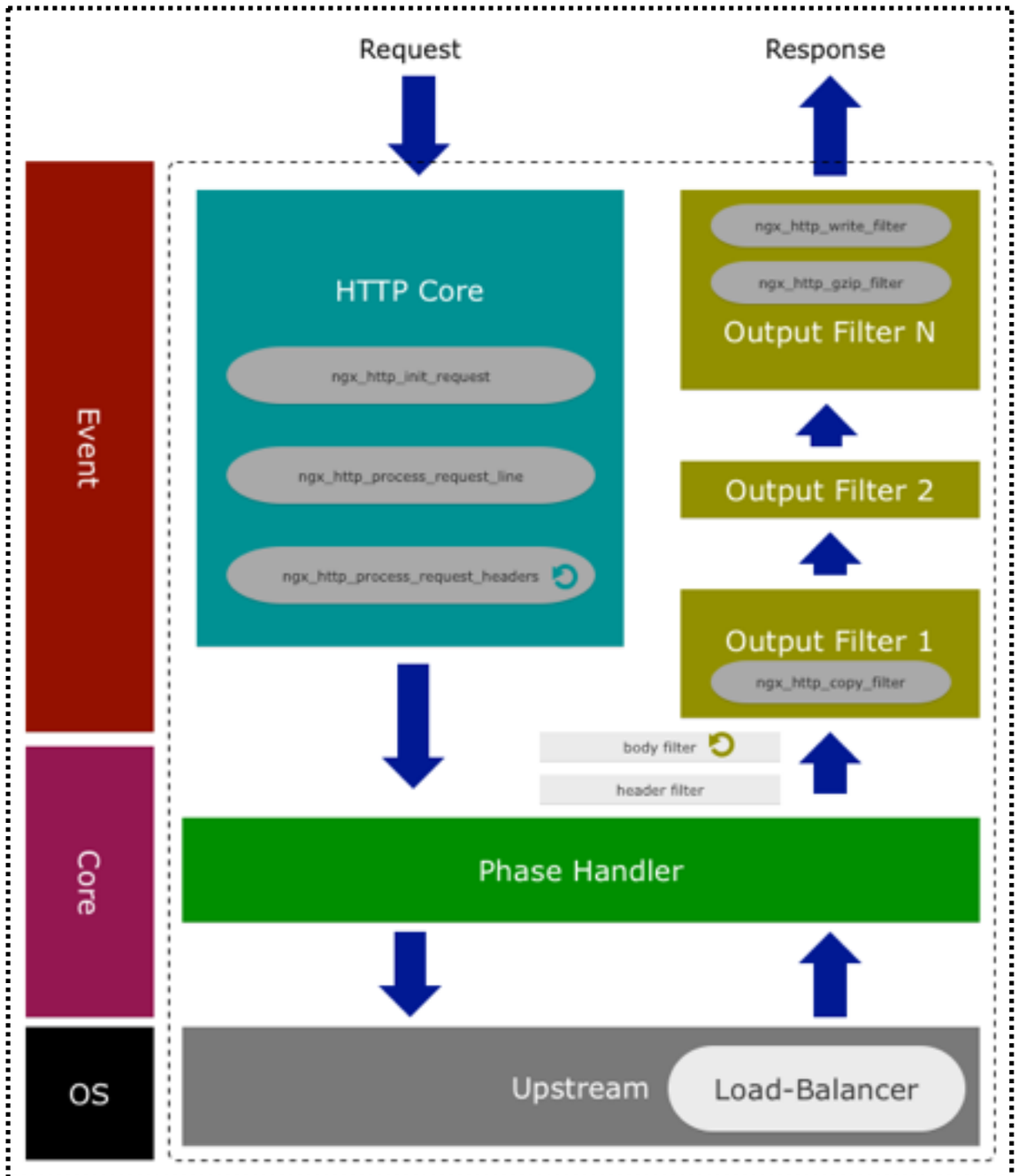
log_by_lua with cosocket

ngx.timer.at（异步） + Lua Buffer + tcp:**send to nsqd**

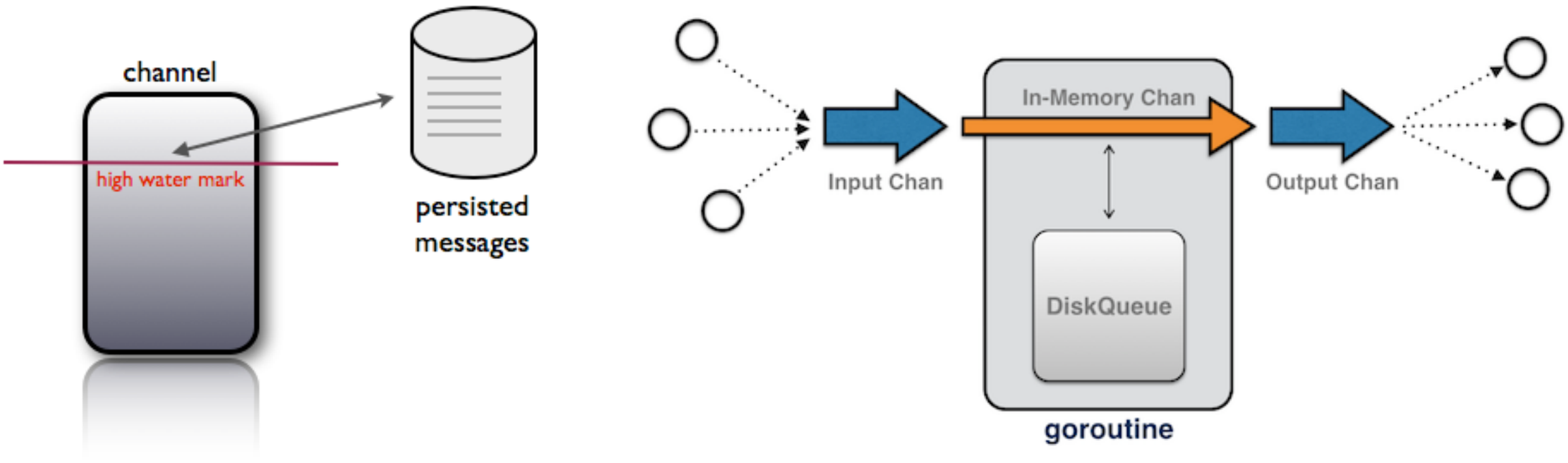
```
_M.nsqd = {

    config = {
        topic = "marco_ngx_log",
        flush_limit = 32768,  -- 32KB
        drop_limit = 2097152, -- 2MB
    },

    cluster = {
        {
            servers = {
                { host = "127.0.0.1", port = 3900 },
            },
        },
    },
}
```

Nginx Internals:
HTTP request phase handlers

- **POST READ PHASE**
- **SERVER REWRITE PHASE**
- FIND CONFIG PHASE
- **REWRITE PHASE**
- POST REWRITE PHASE
- **PRE ACCESS PHASE**
- **ACCESS PHASE**
- POST ACCESS PHASE
- TRY FILES PHASE
- **CONTENT PHASE**
- **LOG PHASE**



nsqd --mem-queue-size



- 每台机器起一个 **nsqd** 单实例和一个对应的 **nsq_to_http** 消费者。
- NSQ 设计上保证消息**至少**传递一次,以确保消息可以**最终**成功发送。

关于**日志实时分析**

- 很多场景中，不需要全部数据都导入进来，**按需分析即可**。
- 目前分析集群（**ELK 架构**）规模约 30 台，资源有限。
- 需要一套**控制规则**来决定当前我们需要获取哪些日志数据。
- **老板说**：我想**现在**抽样看下某个大客户的实时访问情况。
- 当然，我们默认还是收集了全量的**回源日志**和边缘节点的**非健康日志**。

Lua Custom **Logging**:

lua-resty-logger-socket

```
log_format combined '$remote_addr - $remote_user [$time_local] '
                    '"$request" $status $body_bytes_sent '
                    '"$http_referer" "$http_user_agent"';

server {
    access_log /path/to/access.log combined buffer=4096;
    . . .
}
```

Edge Server

bucket:hbimg = {"enable":true,"ratio":0.1}

◀ . . . . . .

redis slaveof

nsqd -> nsq_to_http

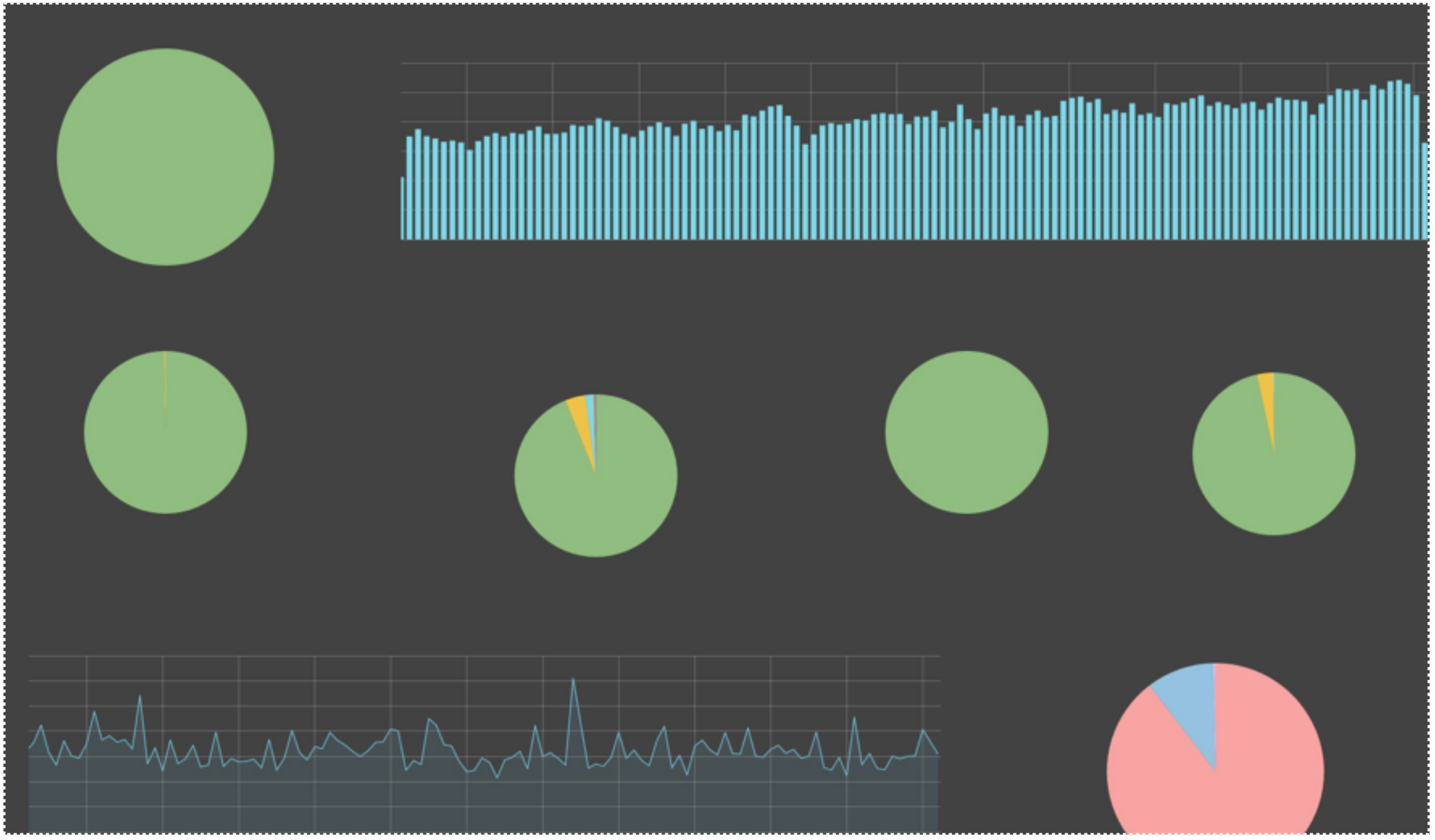
. . . . . ▶

logger.log(cjson.encode(log_msg_table) .. "\n")

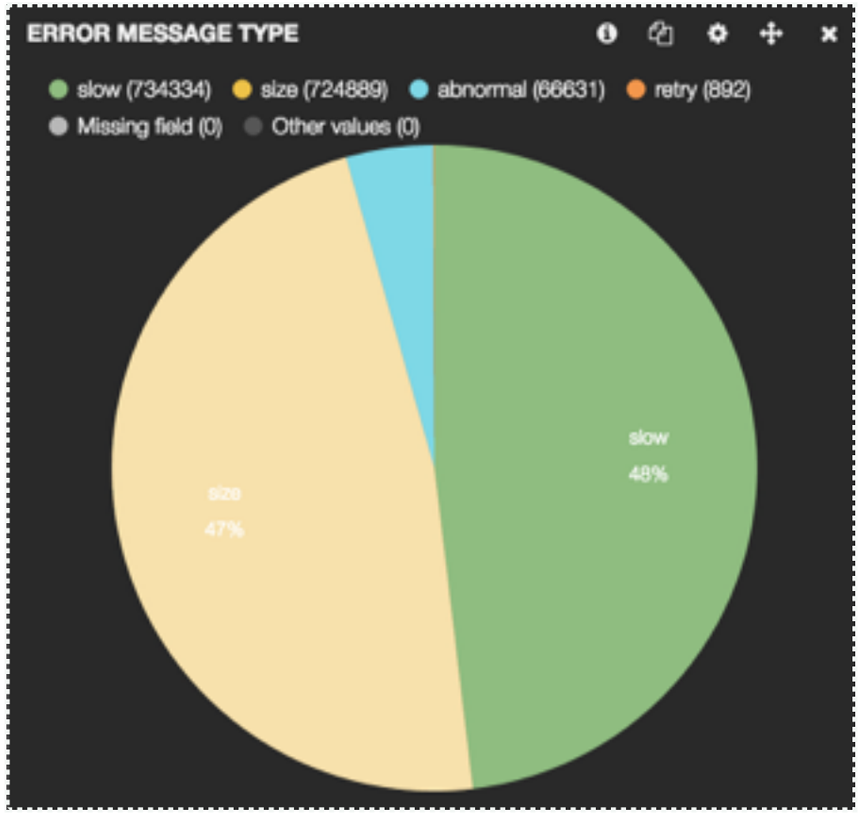
UPYUN LOG

UPYUN LOG Platform:

HAProxy + Heka + Kafka + Elasticsearch + Kibana

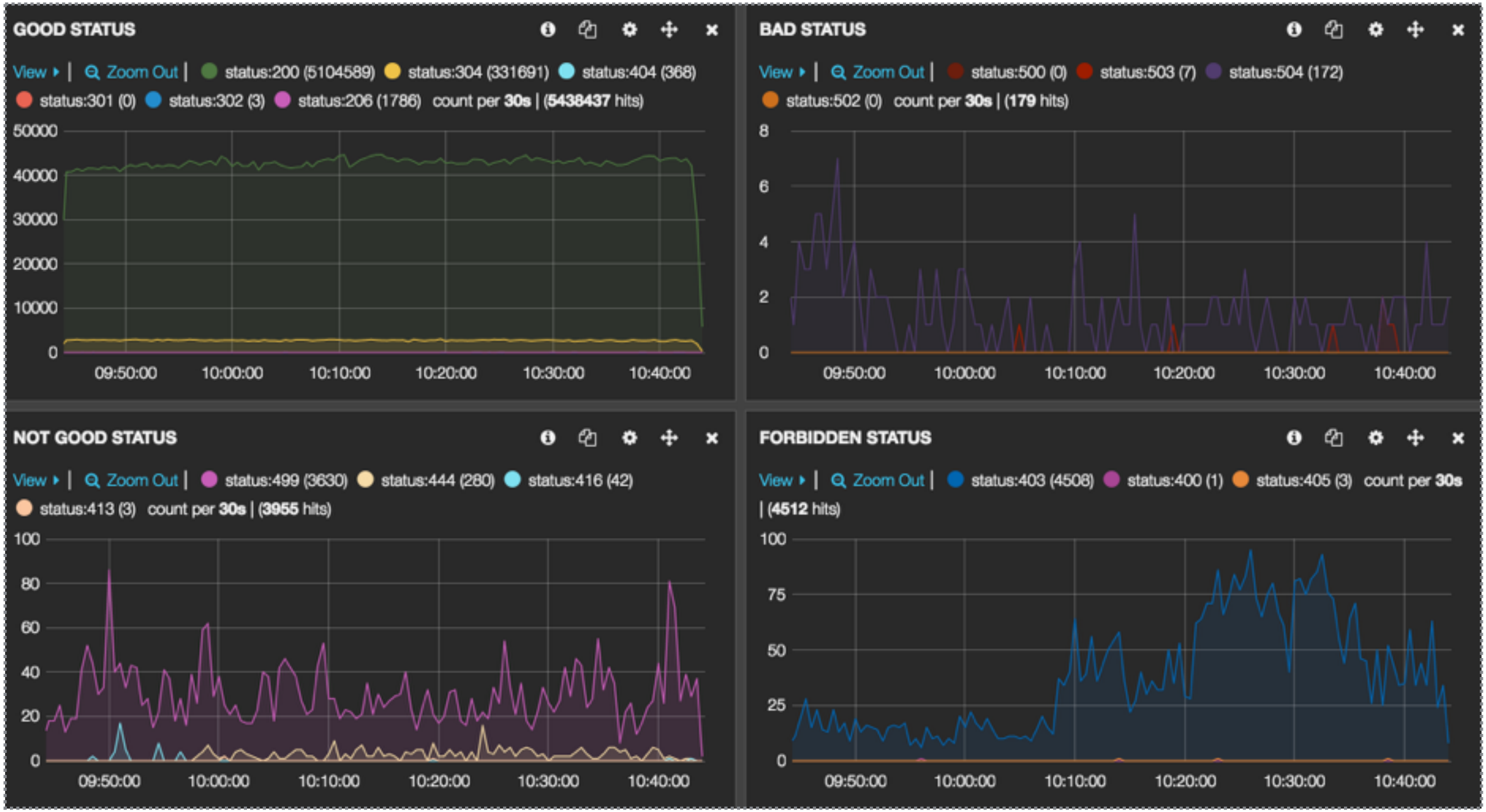


非健康日志有哪些



- **show** - 表示速率小于 **50kb/s** 的请求。
- **size** - 表示发送出去的字节数和文件 **Content-Length** 不一致。
- **retry** - 表示在后端节点重试了至少一次。
- **abnormal** - 表示非正常响应，这里排除客户源站返回的部分。

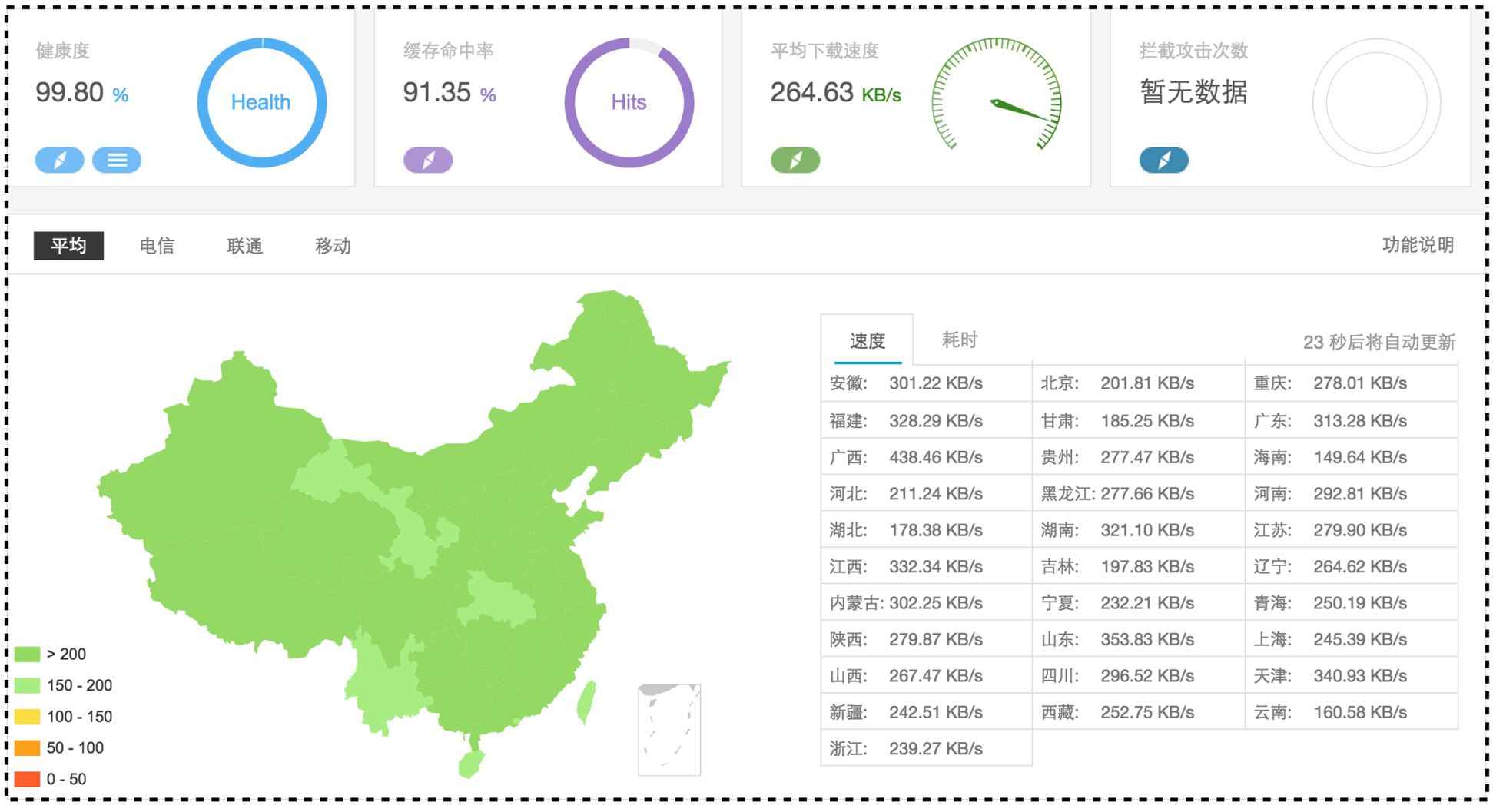
响应状态码分类

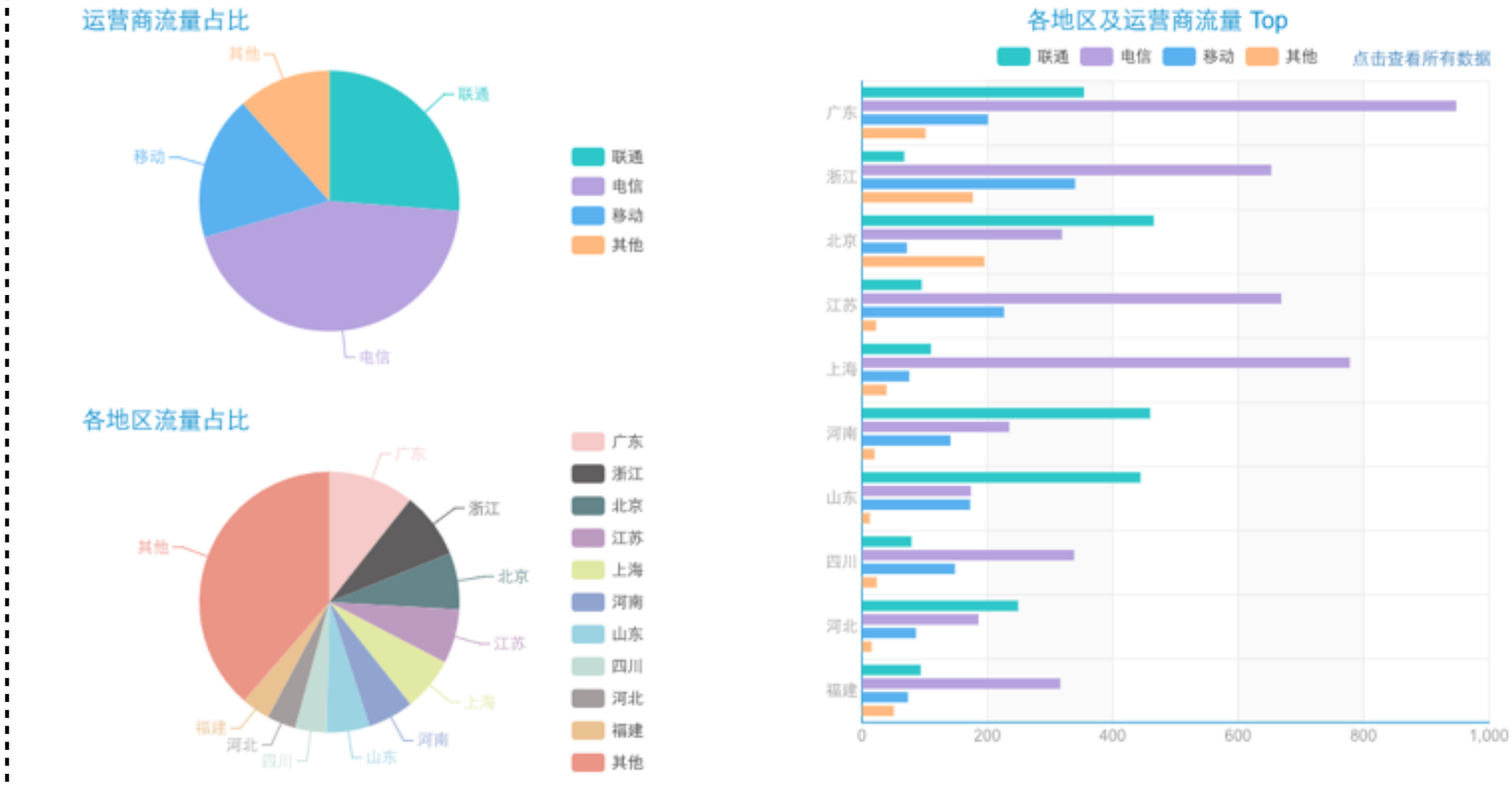

GOOD STATUS
View | Zoom Out | status:200 (5104589) status:304 (331691) status:404 (368)
status:301 (0) status:302 (3) status:206 (1786) count per 30s | (5438437 hits)
50000
40000
30000
20000
10000
0
09:50:00
10:00:00
10:10:00
10:20:00
10:30:00
10:40:00
BAD STATUS
View | Zoom Out | status:500 (0) status:503 (7) status:504 (172)
status:502 (0) count per 30s | (179 hits)
8
6
4
2
0
09:50:00
10:00:00
10:10:00
10:20:00
10:30:00
10:40:00
NOT GOOD STATUS
View | Zoom Out | status:499 (3630) status:444 (280) status:416 (42)
status:413 (3) count per 30s | (3955 hits)
100
80
60
40
20
0
09:50:00
10:00:00
10:10:00
10:20:00
10:30:00
10:40:00
FORBIDDEN STATUS
View | Zoom Out | status:403 (4508) status:400 (1) status:405 (3) count per 30s
| (4512 hits)
100
75
50
25
0
09:50:00
10:00:00
10:10:00
10:20:00
10:30:00
10:40:00

关于**日志聚合计算**

- 全量收集 CDN 边缘请求指标，**实时上报、聚合计算、查询**。
- 支持**多种时间维度**的数据展示（分钟，5分钟，小时，天）。
- 支持域名、服务名称、账号名查询。
- 支持列表、TopN、数据汇总等**复杂查询**请求。

130+ Edge Nodes





ngx_lua 负责业务计算，Redis 进行分阶段聚合，最终结果存储到 ES。

有了**日志数据**，就可以指导我们把系统**做得更好**。

一些有助于定位**请求来源**的字段

- **x-request-id**: ce4fc776afd2b74175695d239b67e3ed
- **via**: T.2428.**H**.1, V.mix-gd-can-007, T.2415.**R**.1, M.cun-ha-cgo-005
- **x-source**: C/200

一些有助于判断**网络行为**的字段

- **first_byte_time**：发送给客户端的**首包时间**统计。
- **client_block_time**：由于客户端**写阻塞**造成的等待时间统计。
- **prematurely_closed**：标记后端是否**提前断开。**

实例：DNS 纠正调度

某个天津电信的用户由于 DNS 配置问题解析到了江苏移动 ...

```
timebug@harmless:~$ http http://img.huaban.com/test.mp4

HTTP/1.1 302 Moved Temporarily
Connection: keep-alive
Content-Length: 161
Content-Type: text/html
Location: http://123.150.200.130/img.huaban.com/test.mp4?
_up_sum=a738dc&_up_id=0f8b04a82480906a2a09bfda8d864c84&_up_from=112.21.160.135
Server: marco
```

江苏常州 移动（112.21.160.135）-> **天津市 电信**（123.150.200.130）

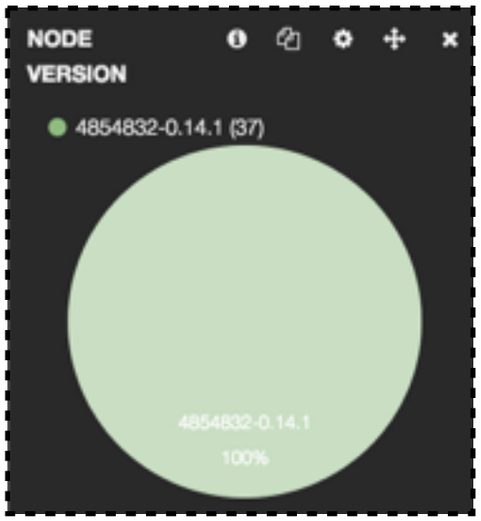
新增统计字段：

- **correct_dns_to**：表示应该往哪个目标节点跳转。
- **correct_dns_from**：表示当前请求是通过哪个节点跳转过来的。

UPYUN DevOps

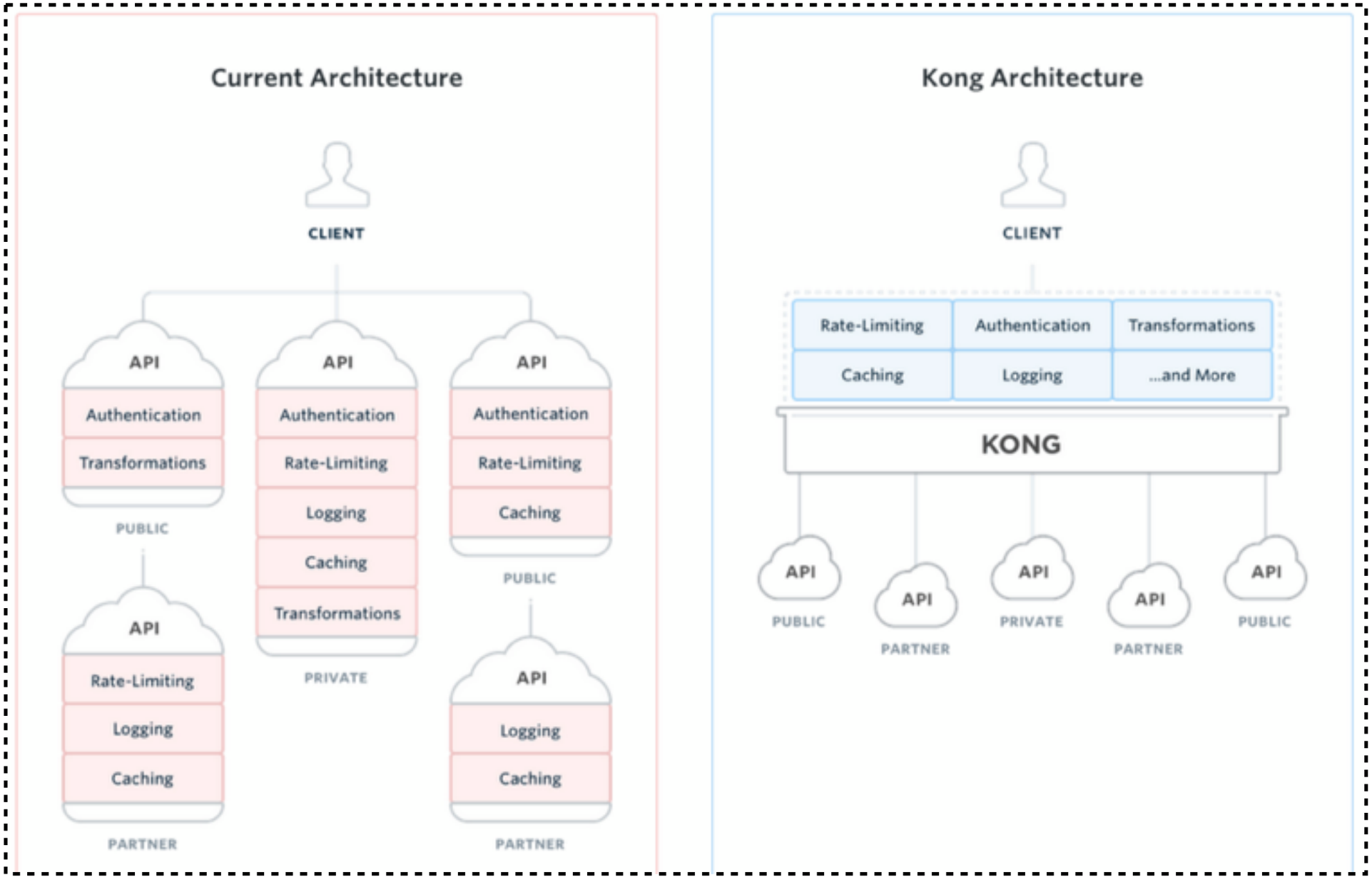
conf hash + project version + upyun.cfg

Ansible Playbook



- rsync code and binary
- conf template instantiate
- kill -HUP `cat /var/run/nginx.pid` (*)

KONG


Current Architecture
CLIENT
API
Authentication
Transformations
PUBLIC
API
Authentication
Rate-Limiting
Logging
Caching
Transformations
PRIVATE
API
Authentication
Rate-Limiting
Caching
PUBLIC
API
Rate-Limiting
Logging
Caching
PARTNER
API
Logging
Caching
PARTNER
Kong Architecture
CLIENT
Rate-Limiting
Authentication
Transformations
Caching
Logging
...and More
KONG
API
PUBLIC
API
PARTNER
API
PRIVATE
API
PARTNER
API
PUBLIC

云 CDN 时代，开通个功能，是否还需要走工单?

nginx.conf	service
server_name *.b0.upaiyun.com	Custom Domain Binding
valid_referers, **allow, deny**	**Custom Antileech Rules and Redirect:** ip, user-agent, referer, token etc.
expires 7d	**Custom Cache Control:** support specific URI rules etc.
ssl_certificate* **ssl_stapling***	Custom SSL
upstream { server 127.0.0.1 }	Custom CDN Origin: support multi-network routing etc.
max_fails=3 fail_timeout=30s **health_check (*)**	Custom Health Check Strategy: passive, active
round-robin, ip_hash, **hash (1.7.2+)**	Custom Load Balancing Strategy
rewrite	Custom URL rewrite
...	...

nginx.conf as a service

powered by ngx_lua

http://io.upyun.com/2015/03/09/hello-world/?foo=bar

[scheme] [host] [path] [query]

tianchaijz:

"$WHEN($MATCH($_URI, '^/foo/.*'))$ADD_REQ_HEADER(X-Foo, bar)"

Marco: **I GOT IT !**

Edge Server

Lua Custom URL **rewrite**:

lua-resty-rewrite | variables

$_METHOD

$_SCHEME

$_HOST

$_POST_name

$_SYM_sym

$_HEADER_name

$_HOST_n

$_COOKIE_name

$_URI

$_GET_name

$_RANDOM_n

$_RANDOM

$_QUERY

Lua Custom URL **rewrite**:

lua-resty-rewrite l functions

$ENCODE_BASE64(E)

$ALL(E1, E2, ...)

$UPPER(E)

$DECODE_BASE64(E)

$ANY(E1, E2, ...)

$LOWER(E)

$WHEN(E1, E2, ...)

$SUB(E1, from, to)

$PCALL(E)

$MATCH(E1, E2)

$GT(E1, E2)

$GE(E1, E2)

$EQ(E1, E2)

$ADD_REQ_HEADER(E1, E2)

$DEL_REQ_HEADER(E1)

$ADD_RSP_HEADER(E1, E2)

Join our team

Q & A