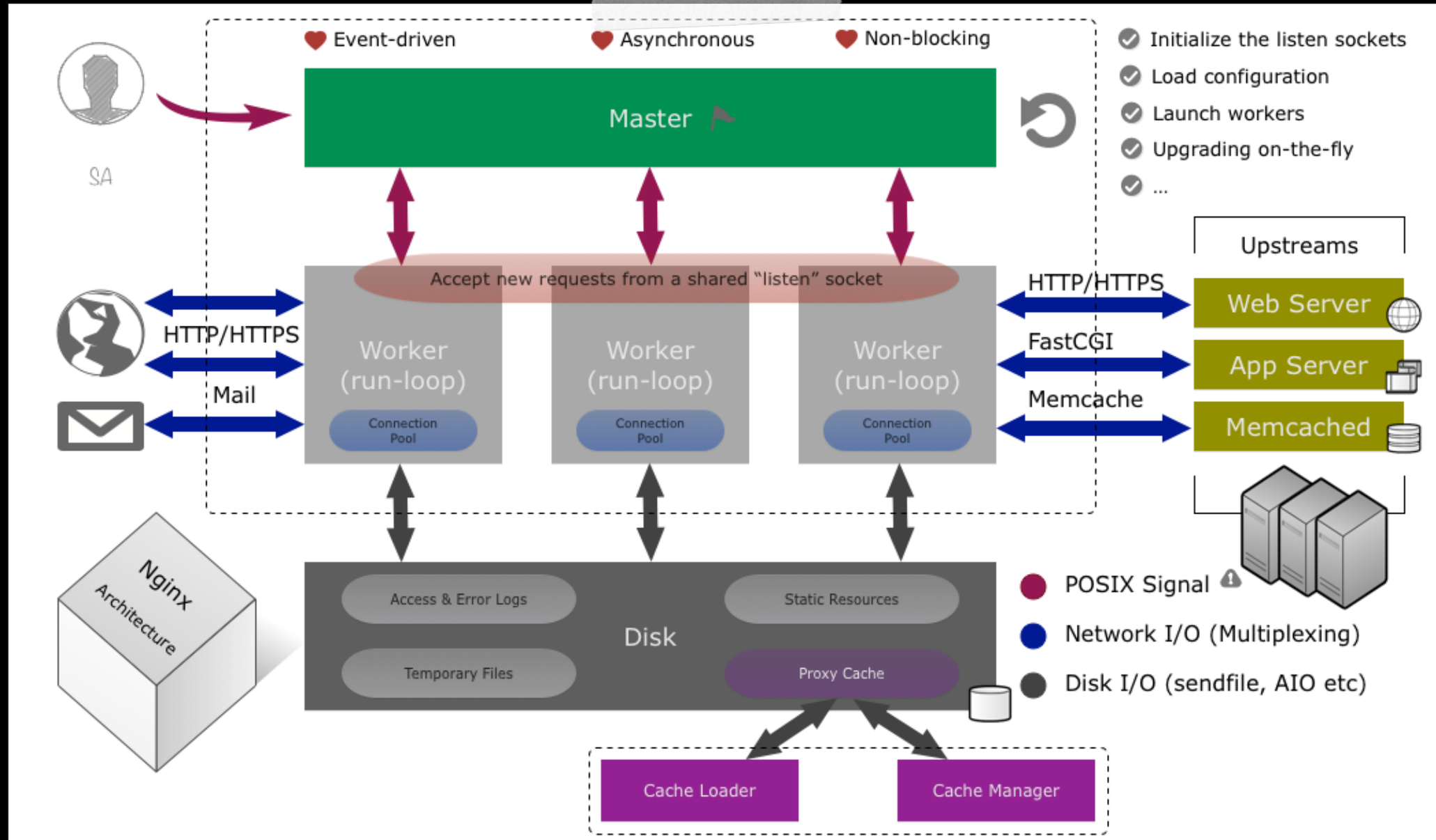
Using ngx_lua in UPYUN

Monkey Zhang (timebug)

2014.11 @ Beijing OSC



Hello Nginx

nginx [engine x] is an HTTP and reverse proxy server, as well as a mail proxy server, written by Igor Sysoev.

A simple example

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

```
http {
    server {
        listen 8080;

        location /hello {
            set $foo "hello";
            echo $foo;

            set $foo "world";
            echo $foo;
        }
    }
}
```

```
$ curl http://localhost:8080/hello
world
world
```

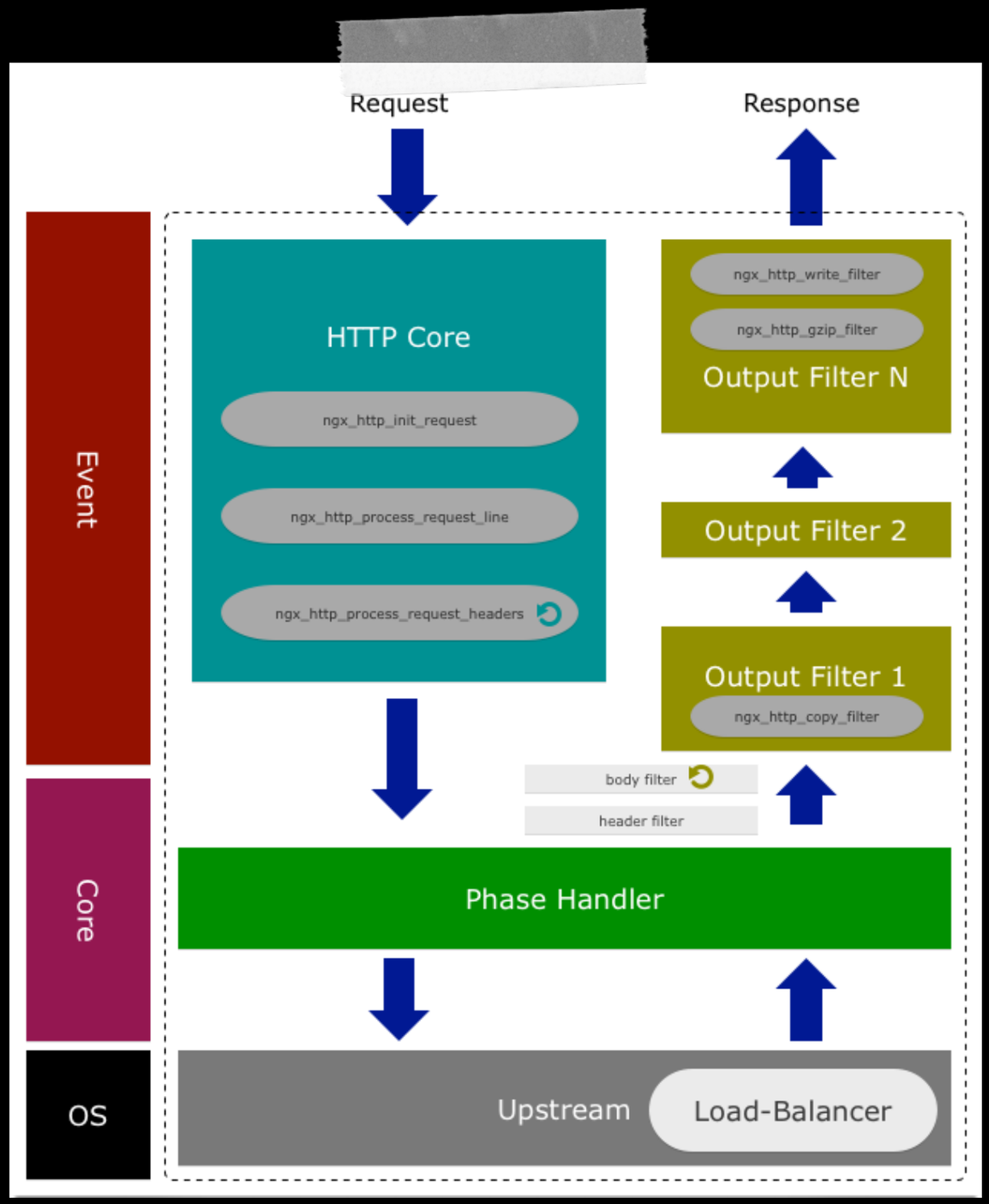
In Fact ...

```
http {
    server {
        listen 8080;

        location /hello {
            set $foo "hello";
            set $foo "world";

            echo $foo;
            echo $foo;
        }
    }
}
```

- REWRITE PHASE: set (ngx_http_rewrite_module)
- CONTENT PHASE: echo (echo-nginx-module)

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



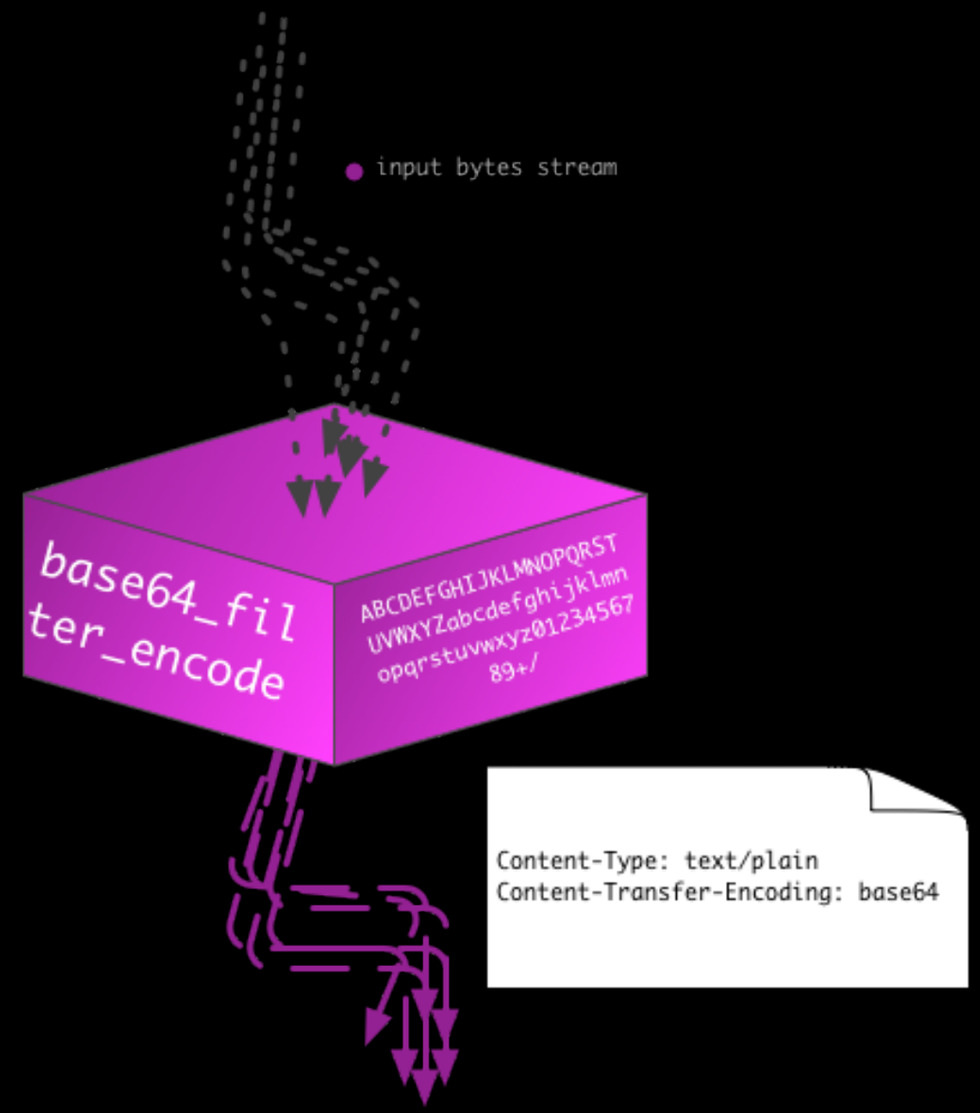
Why Nginx Module

Development is Not Easy ?

A true story

```
$ ./configure --prefix=/opt/nginx \
    --add-module=/path/to/echo-nginx-module \
    --add-module=/path/to/base64-nginx-module
```

```
http {
    server {
        listen 8080;

        location /base64 {
            base64 on;
            base64_max_length 10485760;

            echo "hello world";
        }
    }
}
```




```
$ curl http://localhost:8080/base64
aGVsbG8gd29ybGQK
```

500 lines C

If is Evil

http://agentzh.blogspot.jp/2011/03/how-nginx-location-if-works.html

```
http {
    server {
        listen 8080;

        location /if {
            set $foo 1;

            if ($foo = 1) {
                set $foo 2;
                echo "foo = $foo";
            }

            set $foo 3;
            proxy_pass http://127.0.0.1:$server_port/$foo;
        }

        location ~ /(\d+) {
            echo "bar = $1";
        }
    }
}
```

```
$ curl http://localhost:8080/if
foo = 3
```

`If` is Evil: How it works

- REWRITE PHASE

```
set $foo 1;

if ($foo = 1) {
    set $foo 2;
}
```

```
set $foo 3;
```

- CONTENT PHASE

```
if ($foo = 1) {
    echo "foo = $foo";
}
```

If is Evil:
Break ngx_rewite Directives

```
http {
    server {
        listen 8080;

        location /if {
            set $foo 1;

            if ($foo = 1) {
                set $foo 2;
                break;
                echo "foo = $foo";
            }

            set $foo 3;
            proxy_pass http://127.0.0.1:$server_port/$foo;
        }

        location ~ /(\d+) {
            echo "bar = $1";
        }
    }
}
```

```
$ curl http://localhost:8080/if
foo = 2
```

- REWRITE PHASE

```
set $foo 1;

if ($foo = 1) {
    set $foo 2;
    break;
}
```

- CONTENT PHASE

```
if ($foo = 1) {
    echo "foo = $foo";
}
```

Hello Lua

Lua is a powerful, fast, lightweight, embeddable scripting language.

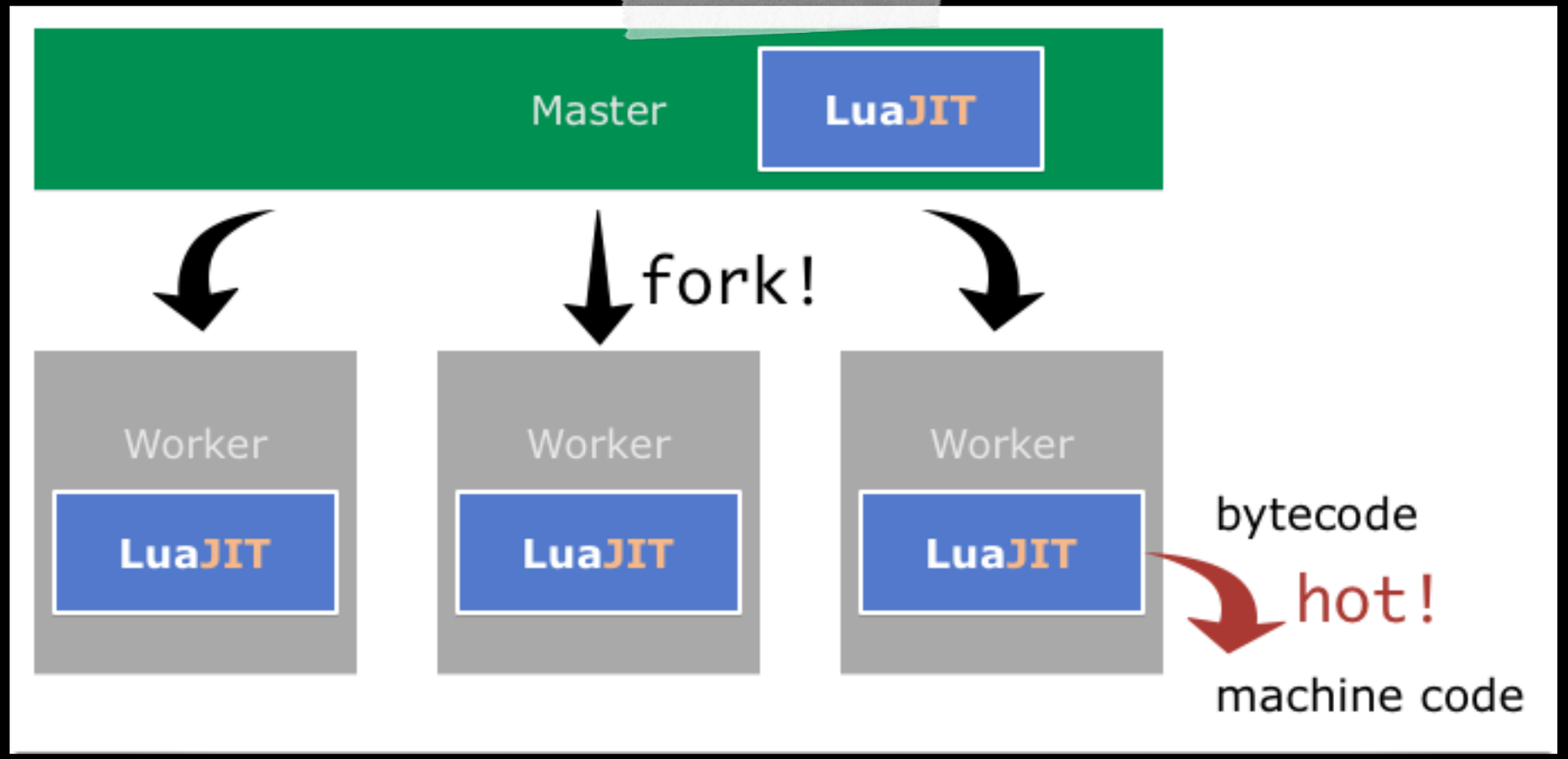

```
$ ./configure --prefix=/opt/nginx \
    --add-module=/path/to/lua-nginx-module
```

```
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

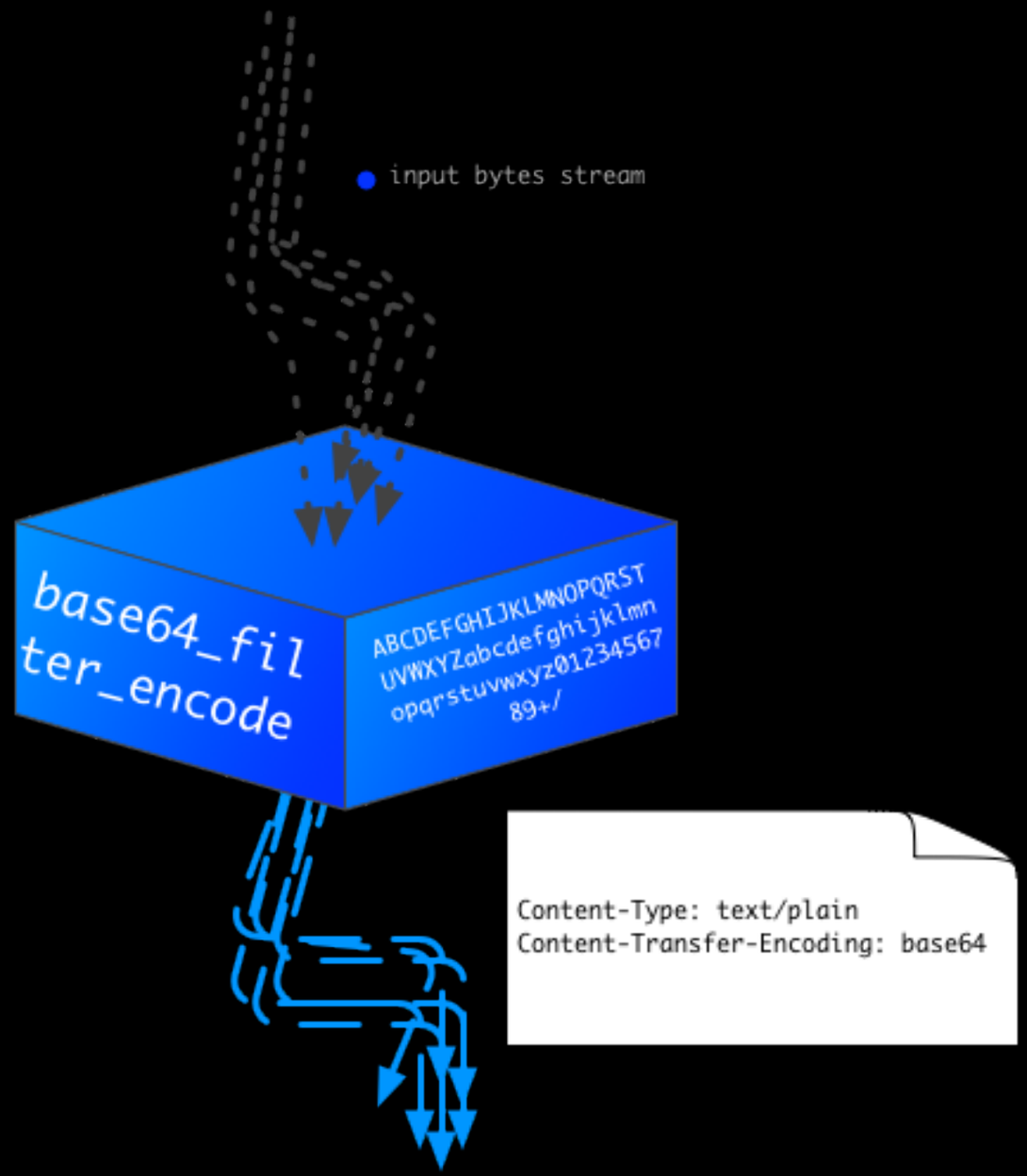
LuaJIT VM embedded into the Nginx

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)
```



30 lines Lua

★ngx_upreferer_module.c ~ 2100 C

★src/modules/referer.lua ~ 500 Lua

ngx.md5 ngx.time ngx.re.* ngx.req.*

ngx.decode_args

string.sub string.find string.byte

table.concat

based on Lua coroutines & synchronous & 100% non-blocking

cosocket API

ngx.socket.* connect send receive

sslhandshake close settimeout etc.

A true story:
Yupoo Referer Redirect (Year 2012)

```
eval_escalate off;
eval_override_content_type text/plain;

eval $answer {
    set $redis_key "$scheme://<key>";
    redis_pass redis;
}

if ($answer = "101") {
    rewrite ^ http://r.yupoo.com/101.gif redirect;
    break;
}

if ($answer = "102") {
    rewrite ^ http://r.yupoo.com/102.gif redirect;
    break;
}

if ($answer ~ "^http://") {
    rewrite ^ $answer redirect;
    break;
}
```

WTF!

- Fork ngx_http_redis to support `ypacl` command
- vkholodkov/nginx-eval-module
  last commit on `Nov 26, 2010`

when upgrade Nginx to the latest version

Coredump

Yupoo Referer Redirect by Lua

lua-resty-redis (based on the cosocket API)

ngx.redirect rewrite_by_lua_file

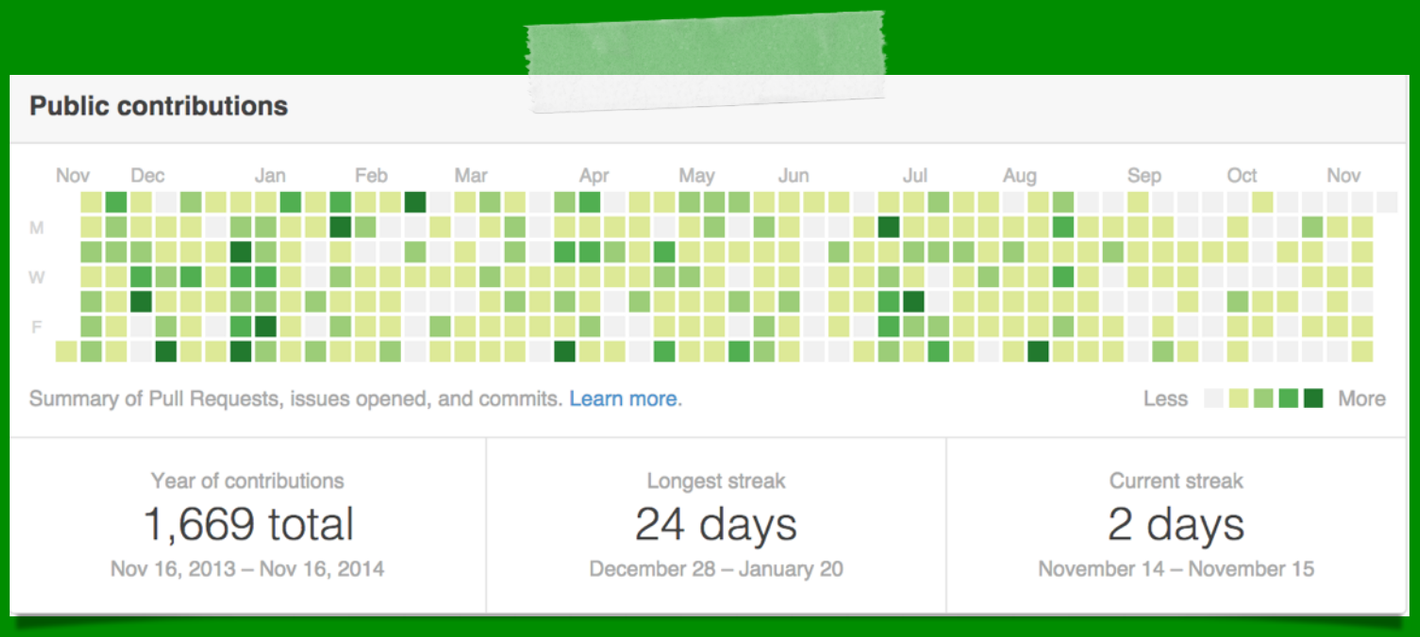
```
Local redis = require "resty.redis"
local red = redis:new()

redis.add_commands("ypacl")

-- set_timeout and connect

local res, err = red:ypacl(key)

if res == "101" then
    return ngx.redirect("http://r.yupoo.com/101.gif")
else
    -- do something else
end
```

LuaJIT FFI

lua-resty-uuid:
Based on **LuaJIT** FFI

```
-- modified version of original pull request by smallfish
-- https://github.com/openresty/lua-resty-string/pull/7

local ffi = require "ffi"
local new = ffi.new
local string = ffi.string

local _M = {}

ffi.cdef[[
    typedef unsigned char uuid_t[16];
    void uuid_generate(uuid_t out);
    void uuid_unparse(const uuid_t uu, char *out);
]]

local libuuid = ffi.load("libuuid")
function _M.generate()
    if libuuid then
        local uuid = new("uuid_t")
        local result = new("char[36]")
        libuuid.uuid_generate(uuid)
        libuuid.uuid_unparse(uuid, result)
        return string(result)
    end
end

return _M
```

Openresty



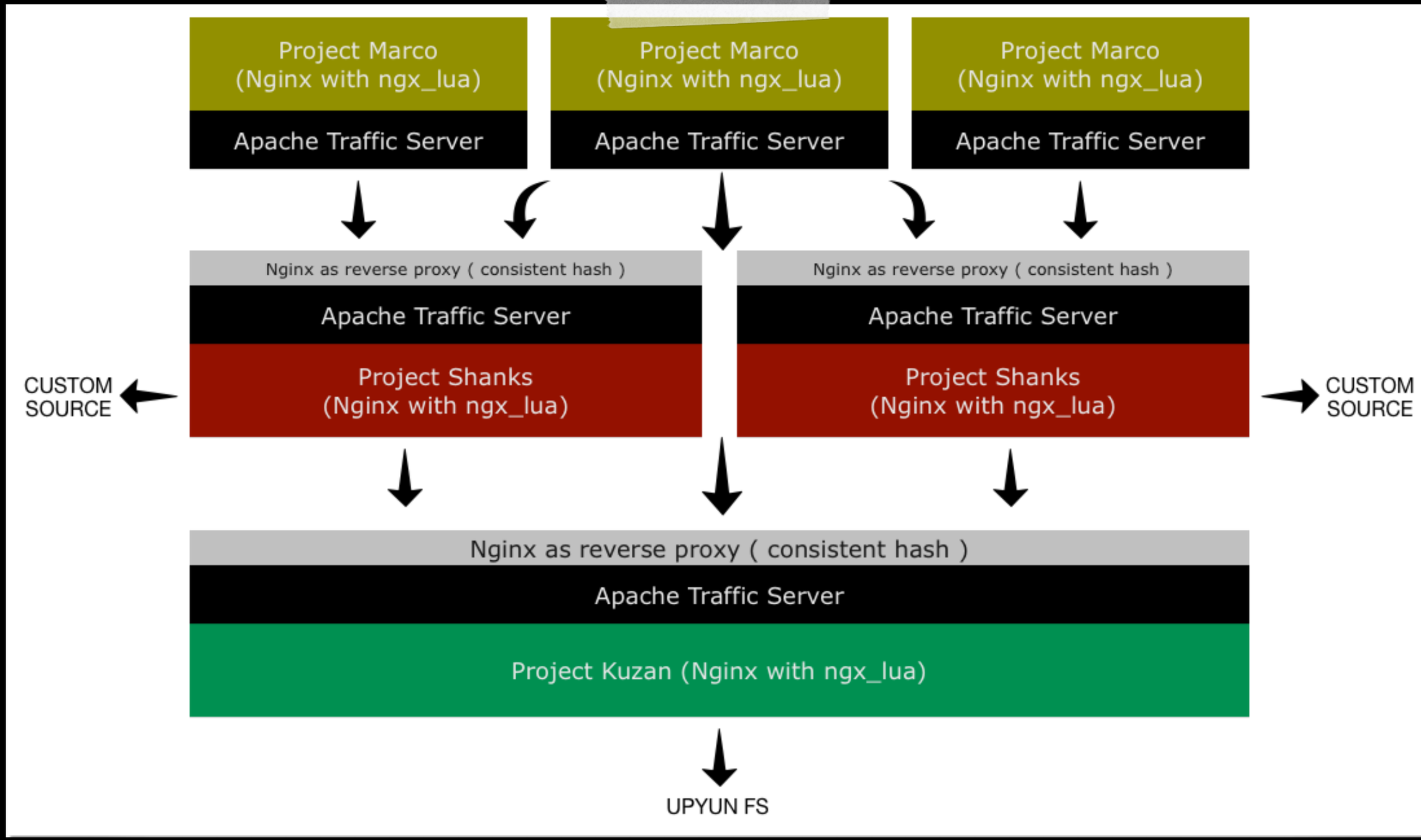
Yichun "agentzh" Zhang (章亦春) agentzh@gmail.com, CloudFlare Inc.

ONEPIECE


WANTED
DEAD OR ALIVE
MARCO THE PHOENIX
฿ 750,000,000-
KONO SAKUHIN HA FICTION DETHUNODE JITSUZAISURU JINBUTSU DANTAI SONOTA NO SOSHIKI TO DOITSU NO MEISHOU GA GEKICHU NI TOUJYOU SHITATOSHITEMO JITSUZAI NA MONOTOHA ISSAI MUKANKEIDETH.
MARINE


WANTED
DEAD OR ALIVE
RED-HAIRED SHANKS
฿ 850,000,000-
KONO SAKUHIN HA FICTION DETHUNODE JITSUZAISURU JINBUTSU DANTAI SONOTA NO SOSHIKI TO DOITSU NO MEISHOU GA GEKICHU NI TOUJYOU SHITATOSHITEMO JITSUZAI NA MONOTOHA ISSAI MUKANKEIDETH.
MARINE

UPYUN CDN


Project Marco
(Nginx with ngx_lua)
Apache Traffic Server
Project Marco
(Nginx with ngx_lua)
Apache Traffic Server
Project Marco
(Nginx with ngx_lua)
Apache Traffic Server
Nginx as reverse proxy (consistent hash)
Apache Traffic Server
Project Shanks
(Nginx with ngx_lua)
Nginx as reverse proxy (consistent hash)
Apache Traffic Server
Project Shanks
(Nginx with ngx_lua)
CUSTOM
SOURCE
CUSTOM
SOURCE
Nginx as reverse proxy (consistent hash)
Apache Traffic Server
Project Kuzan (Nginx with ngx_lua)
UPYUN FS

UPYUN CDN:
Metadata Cache

- CJSON, MessagePack
- ngx.shared.DICT
- lua-resty-lock
- lua-resty-shcache
- lua-resty-lrucache (*)

Metadata Cache:
The original version (Year 2012)

```lua
local metadata = ngx.shared.metadata

-- local key, bucket = ...

local value = metadata:get(key)
if value ~= nil then
    if value == "404" then
        return   -- HIT_NEGATIVE
    else
        return value -- HIT
    end
end

local rds = redis:new()
local ok, err = rds:connect("127.0.0.1", 6379)
if not ok then
    metadata:set(key, "404", 120) -- expires 2 minutes
    return -- NO_DATA
end

res, err = rds:hget("upyun:" .. bucket, ":something")
if not res or res == ngx.null then
    metadata:set(key, "404", 120)
    return -- NO_DATA
end

metadata:set(key, res, 300) -- expires 5 minutes
rds:set_keepalive()

return res -- MISS
```

Issues

- "dog-pile effect"
- NO_DATA when redis crash
- code inflexible

Metadata Cache:

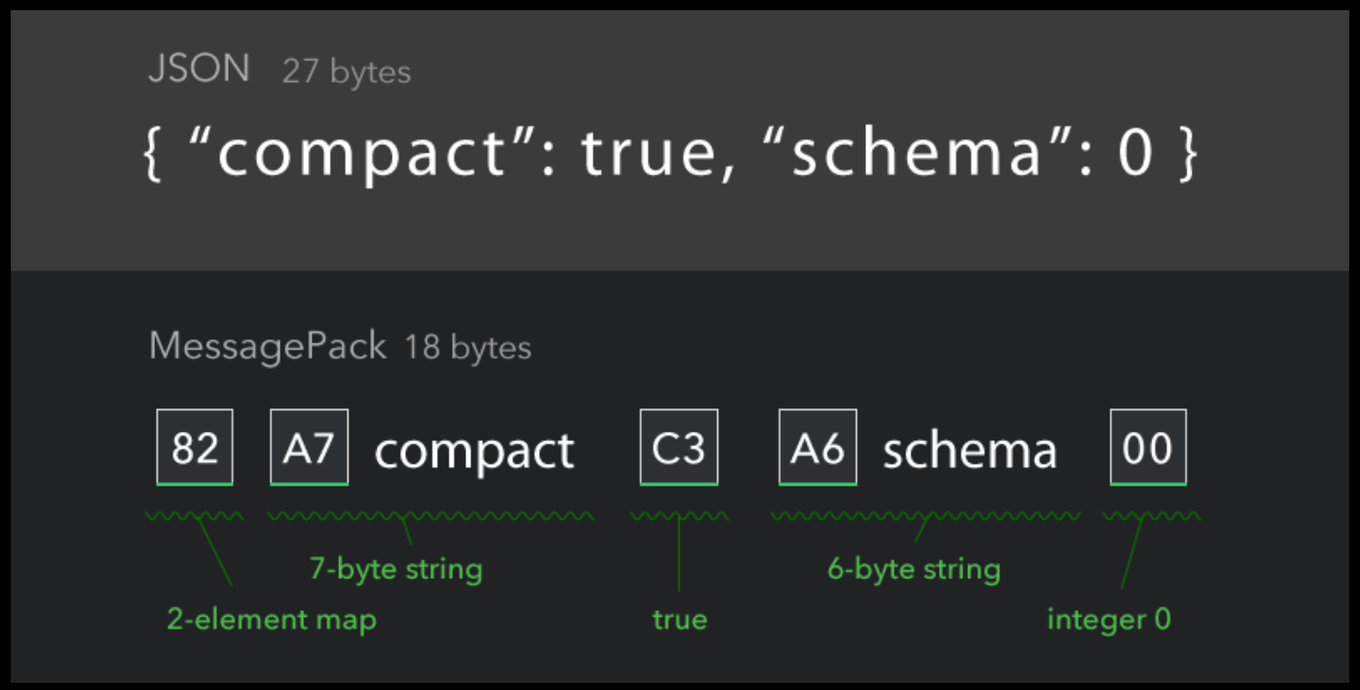
lua-resty-shcache

```
-- app/src/modules/metadata.lua

local shcache = require "resty.shcache"

function _M.get_metadata(bucket)
    local lookup_metadata = function ()
        -- fetch from redis
        return res
    end

    local cache_data = shcache:new(
        ngx.shared.metadata,
        { external_lookup = lookup_metadata,
          encode = cmsgpack.pack,
          decode = cmsgpack.unpack,
        },
        { positive_ttl = cache_positive_ttl,
          negative_ttl = cache_negative_ttl,
          name = "metadata",
        })

    -- local key = ...

    local data, _ = cache_data:load(key)
    if not data then
        return
    end

    return data
end
```

- cache locks (based on lua-resty-lock)
- serialization / de-serialization
- external lookup via Lua closure
- MISS, HIT, HIT_NEGATIVE, STALE, NET_ERR

Metadata Cache:

MessagePack



Bucket	JSON	MessagePack
huab***	6035 bytes	4135 bytes - 69%

Metadata Cache:

lua-resty-lrucache

- based on **LuaJIT** FFI
- HIT_LRU
- avoid serialization / de-serialization

UPYUN CDN:
Upstream Health Check

- ✦lua-resty-checkups (*)
- ✦lua-upstream-nginx-module

Upstream Health Check: lua-resty-checkups

```
-- app/etc/config.lua

_M.global = {
    checkup_timer_interval = 5,
    checkup_timer_overtime = 60,
}

_M.api = {
    timeout = 2,
    typ = "general", -- http, redis, mysql etc.
    cluster = {
        {   -- level 1
            try = 2,
            servers = {
                { host = "127.0.0.1", port = 12354 },
                { host = "127.0.0.1", port = 12355 },
                { host = "127.0.0.1", port = 12356 },
            }
        },
        {   -- level 2
            servers = {
                { host = "127.0.0.1", port = 12360 },
                { host = "127.0.0.1", port = 12361 },
            }
        },
    },
}
```

Upstream Health Check: checkups with nginx.conf

```
# nginx/conf/upstream.conf

upstream api.com {
    server 127.0.0.1:12354;
    server 127.0.0.1:12355;
    server 127.0.0.1:12356;
    server 127.0.0.1:12360 backup;
    server 127.0.0.1:12361 backup;
}
```

type	status	timer	interval
checkup	shm_zone	global	5s
upstream	per worker	per worker	2s

```
-- app/etc/config.lua

_M.global = {
    checkup_timer_interval = 5,
    checkup_timer_overtime = 60,

    ups_status_sync_enable = true,
    ups_status_timer_interval = 2,
}

_M.api = {
    cluster = {
        {   -- level 1
            try = 2,
            upstream = "api.com",
        },
        {   -- level 2
            upstream = "api.com",
            upstream_only_backup = true,
        },
    },
}
```

nginx.conf	service
server_name *.b0.upaiyun.com	Custom Domain Binding
valid_referers, rewrite, **allow, deny**	**Custom Antileech Rules and Redirect:** ip, user-agent, referer, token etc.
expires 7d	**Custom Expires Time:** support specific URI rules etc.
ssl_certificate*	Custom SSL Certificates Load
upstream { server 127.0.0.1 }	Custom CDN Source: 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
...	...

nginx.conf as a service

powered by ngx_lua

UPYUN DevOps

conf hash + project version + upyun.cfg

Ansible Playbook

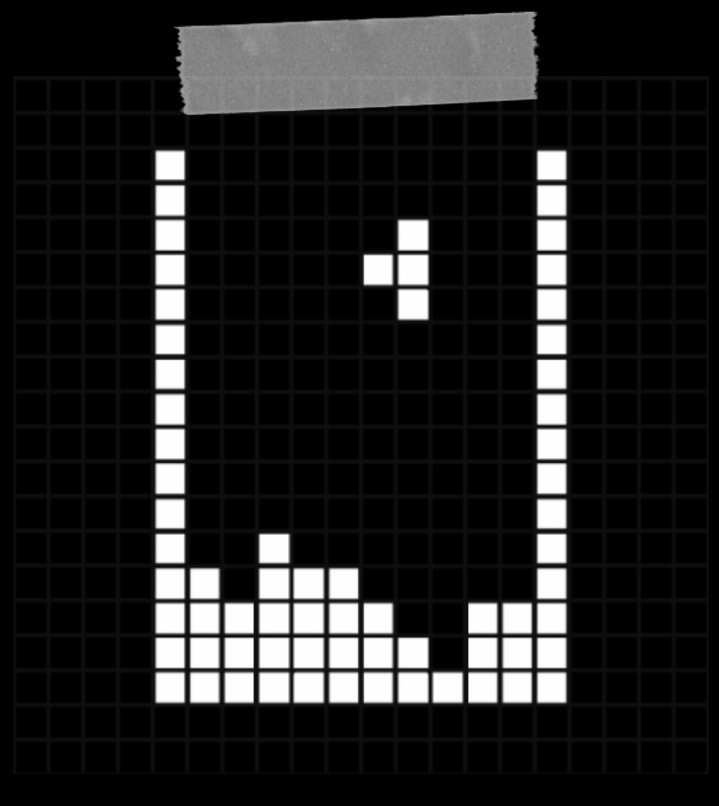
- rsync code and binary
- conf template instantiate
- kill -HUP `cat /var/run/nginx.pid` (*)

Lua CDN

Lua WAF

Lua SSL

Join our team



A Systems Engineer at UPYUN

★Email: timebug.info@gmail.com

★Github: https://github.com/timebug

Nginx ngx_lua agentzh Lua

blog.cloudflare.com Openresty **LuaJIT** Github

Maxim Dounin Igor Sysoev chaoslawful

https://groups.google.com/forum/#!forum/OpenResty

Ansible Michael Pall Open source

Thanks

...

Q & A