a little Nginx (1)

detailyang@gmail.com

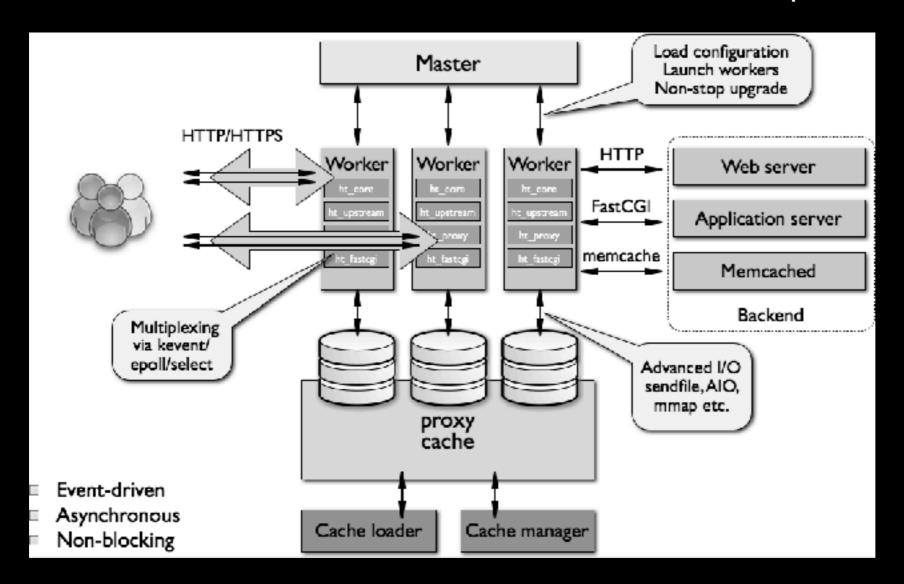
2016.10

Nginx

- 1. HTTP and reverse proxy server
- 2. a mail proxy server
- 3. a generic TCP/UDP proxy server

Why Nginx is fast

1. reactor architect based multi IO dispatch

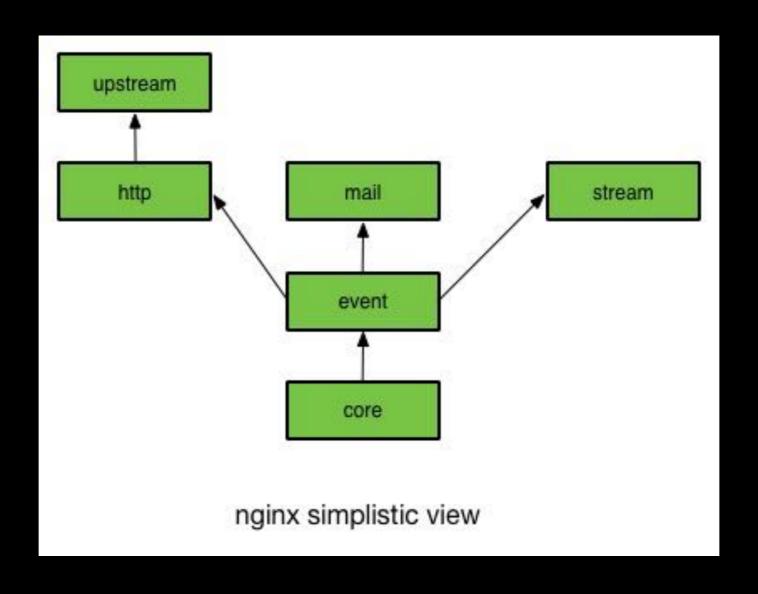


One master and several worker processes; worker processes run under an unprivileged user;

Why Nginx is fast

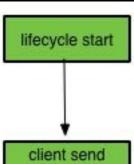
2. The Nginx codebase is written in the C programming language and is extremely optimized by the Igor Sysoev.

simplistic view



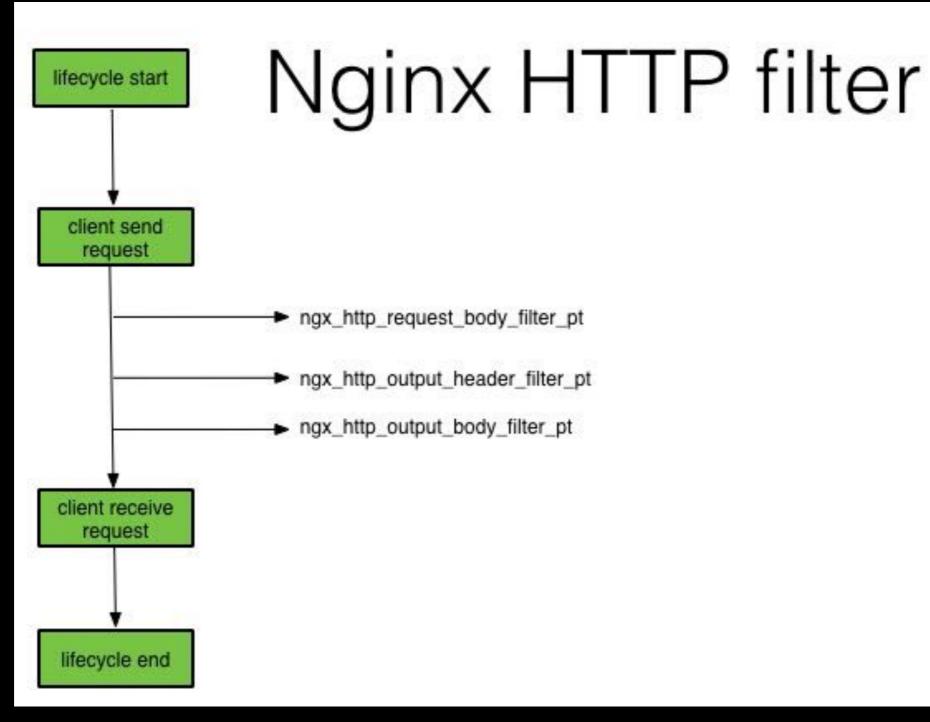
key concept

all is modular(e)



Nginx HTTP phase

```
request
     ngx_http_process_request_headers
                                                     ngx_http_realip_module
                       NGX_HTTP_POST_READ_PHASE
                                                         ngx_http_rewrite_module
                       NGX_HTTP_SERVER_REWRITE_PHASE
                                                      location match
                       NGX_HTTP_FIND_CONFIG_PHASE
                       NGX_HTTP_REWRITE_PHASE
                                                       ngx_http_rewrite_module
                       NGX_HTTP_POST_REWRITE_PHASE
                                                      check dead recursion
                       NGX_HTTP_PREACCESS_PHASE
                                                      ngx_http_limit_conn_module ngx_http_limit_req_module
                       NGX HTTP ACCESS PHASE
                                                  ngx http access module
                       NGX_HTTP_POST_ACCESS_PHASE
                                                   try_files
                       NGX_HTTP_TRY_FILES_PHASE
                                                  ngx_http_autoindex_module
                       NGX_HTTP_CONTENT_PHASE
                                                                  ngx_http_proxy_module
                                                    ngx http fastcgi module
client receive
 request
                                               ngx http log module
                       NGX_HTTP_LOG_PHASE
lifecycle end
```



nginx

Nginx Upstream Hook

input_filter_init

create_request

M2 - SEREE - 175

input_filter

reinit_request

rewrite_redirect

process_header

rewrite_cookie

abort_request

finalize_request

peer.init

peer.get

ip_hash

round_robin

upstream

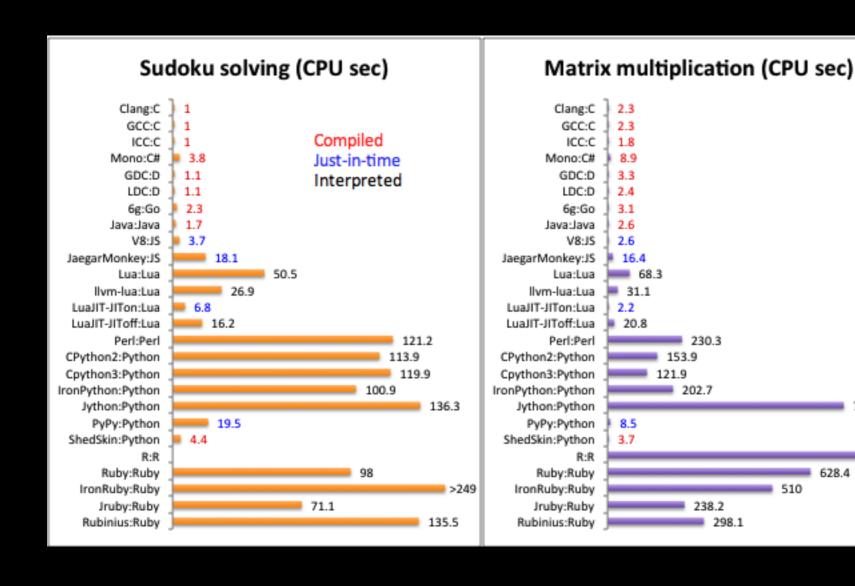
Nginx + Lua Beyond Nginx

What's the ngx_http_lua

Embed the power of Lua into Nginx

Why Lua

High Performance



easy to develop

20 row in C = 1 row in Lua

```
part = &r->headers_in.headers.part;
                                                         ngx.req.set header("Content-Type", "text/css")
h = part->elts;
for (i = 0; /* void */; i++) {
 if (i >= part->nelts) {
      \cdot \cdot \cdot if \cdot (part->next \cdot == \cdot \overline{N}ULL) \cdot {
             break;
       part = part->next;
        h = part->elts;
        i = 0;
    if (key.len == h[i].len &&
         ngx_strncasecmp(key.data, h[i].data,
             h[i].len) == 0)
         h[i].value = *value;
```

How Does Nginx Connect Lua

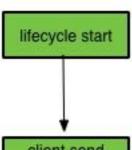
Inject Nginx C API To Lua VM

```
int main(int argc, char *argv[]) {
45
         lua_State *l;
46
47
        -l = luaL newstate();
48
        luaL_openlibs(l);
49
50
     lua_pushcfunction(l, l_dir);
51
        lua_setglobal(l, "mydir");
52
53
54
        if (luaL_loadfile(l, "./dir.lua") != 0) {
             fprintf(stderr, "luaL_loadfile err: %s.\n", lua_tostring(l, -1));
55
             goto end;
56
         }
57
58
{ 59
       --if (lua_pcall(l, 0, 0, 0, 0) != 0) {
             fprintf(stderr, "lua_pcall err: %s.\n", lua_tostring(l, -1));
60
361
         }
62
63
     end:
64
         lua_close(l);
65
         return 0;
66
67
```

```
rua_pushcrunction(L, ngx_nttp_iua_get_raw_phase_context);
    lua_setfield(L, -2, "_phase_ctx");
    ngx_http_lua_inject_arg_api(L);
    ngx_http_lua_inject_http_consts(L);
    ngx http lua inject core consts(L);
    ngx_http_lua_inject_log_api(L);
    ngx_http_lua_inject_output_api(L);
    ngx_http_lua_inject_time_api(L);
    ngx_http_lua_inject_string_api(L);
    ngx_http_lua_inject_control_api(log, L);
    ngx_http_lua_inject_subrequest_api(L);
    ngx_http_lua_inject_sleep_api(L);
    ngx_http_lua_inject_phase_api(L);
#if (NGX_PCRE)
    ngx_http_lua_inject_regex_api(L);
#endif
    |ngx_http_lua_inject_req_api(log, L);
    ngx_http_lua_inject_resp_header_api(L);
    ngx_http_lua_create_headers_metatable(log, L);
    ngx_http_lua_inject_variable_api(L);
    ngx_http_lua_inject_shdict_api(lmcf, L);
    ngx_http_lua_inject_socket_tcp_api(log, L);
    ngx_http_lua_inject_socket_udp_api(log, L);
    ngx_http_lua_inject_uthread_api(log, L);
    ngx_http_lua_inject_timer_api(L);
    ngx_http_lua_inject_config_api(L);
    ngx_http_lua_inject_worker_api(L);
    ngx_http_lua_inject_misc_api(L);
```

How Does NginxLua Work

Over Nginx Hooks



Nginx HTTP phase

client send request

```
ngx_http_process_request_headers
                                                     ngx_http_realip_module
                       NGX_HTTP_POST_READ_PHASE
                                                         ngx_http_rewrite_module
                       NGX_HTTP_SERVER_REWRITE_PHASE
                       NGX_HTTP_FIND_CONFIG_PHASE
                                                      location match
                       NGX_HTTP_REWRITE_PHASE
                                                       ngx_http_rewrite_module
                       NGX_HTTP_POST_REWRITE_PHASE
                                                      check dead recursion
                       NGX_HTTP_PREACCESS_PHASE
                                                      ngx http limit conn module ngx http limit reg module
                       NGX HTTP ACCESS PHASE
                                                  ngx http access module
                       NGX_HTTP_POST_ACCESS_PHASE
                       NGX HTTP TRY FILES PHASE
                                                   try files
                                                  ngx_http_autoindex_module
                       NGX_HTTP_CONTENT_PHASE
                                                                  ngx_http_proxy_module
                                                    ngx http fastcgi module
client receive
 request
                       NGX_HTTP_LOG_PHASE
                                               ngx http log module
lifecycle end
```

Order of Lua Nginx Module Directives init_by_lua* init_worker_by_lua* Initialization Phase Yes ssl_certificate_by_lua* Secure request? Request start No set_by_lua* rewrite_by_lua* access_by_lua* Rewrite/Access Phase Upstream content_by_lua* Content generated by ? balancer_by_lua* Other directive header_filter_by_lua* body_filter_by_lua* Content Phase log_by_lua* Log Phase

Be Continued