NGINX, Lua, and beyond

☺ *agentzh@gmail.com* ☺
Yichun Zhang (@agentzh)

2014.02.25

ngx_echo

```
echo "hello world";

echo_after_body "some footer...";

echo_duplicate 1024 "a";

echo_sleep 1.5;

echo_exec /foo;

echo_location /baz;

echo_subrequest_async DELETE /bar

$echo_request_method
```

ngx_headers_more

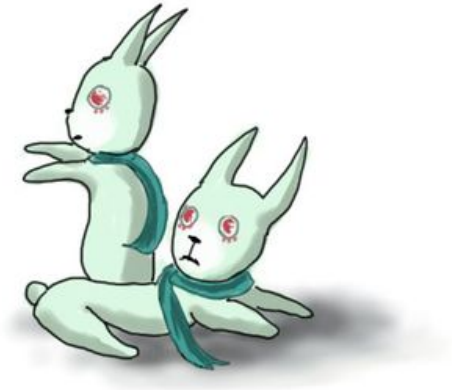
```
    more_clear_headers Server X-Powered-By;

    more_set_headers 'Foo: bar';

    more_set_input_headers 'User-
Agent: curl';
```

ngx_set_misc

```
set_unescape_uri $name $arg_name;

set_if_empty $name "anonymous";

set_random $rand 1 10;

set_formatted_local_time $timestr "%a %b %e %H:%M:%S %Y %Z";
```

NGINX

ngx_memc



```
location = /memc-set {
    set $memc_cmd set;
    set $memc_key foo;
    set $memc_value blah;
    set $memc_exptime 60; # 60 sec
    memc_pass 127.0.0.1:11211;
}
```

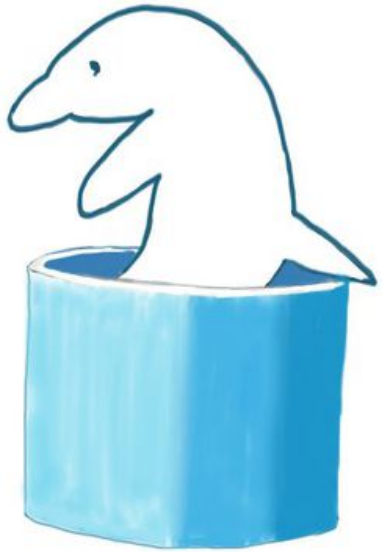
Ragel

ngx_redis2

```
location = /redis-test {
    redis2_query del key1;
    redis2_query lpush key1 A;
    redis2_query lpush key1 B;
    redis2_pass 127.0.0.1:6379;
}
```

NGINX

ngx_drizzle

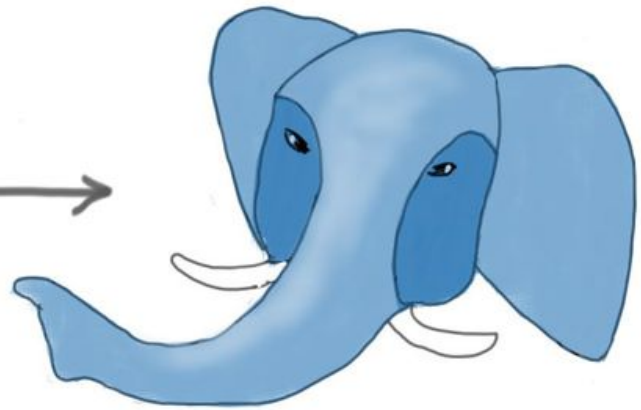


```
upstream mysql_backend {
    drizzle_server 127.0.0.1:3306 dbname=test
         password=some_pass user=monty protocol=mysql;
}

server {
    location = /mysql {
        set $query 'select * from cats';
        drizzle_query $query;
        drizzle_pass mysql_backend;
        rds_json on;  # or rds_csv on
}
```

libdrizzle

NGINX

ngx_postgres



libpq

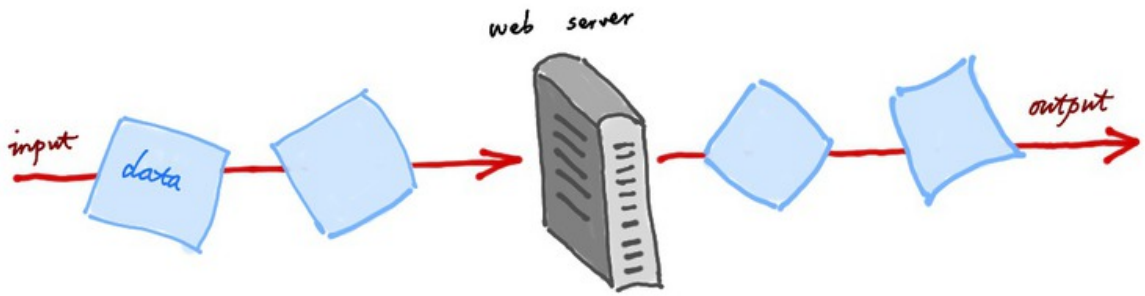
Oracle? OCI?

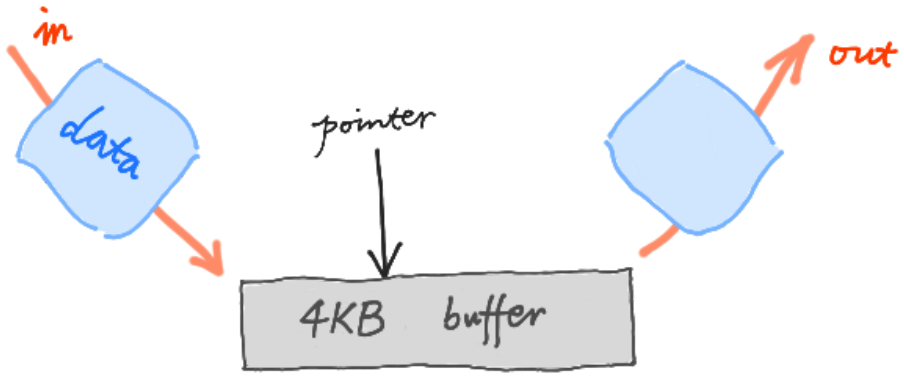
ngx_srcache

```
location /foo {
    set $key $uri$args;

    srcache_fetch GET /memc $key;
    srcache_store PUT /memc $key;

    srcache_store_statuses 200 301 302;

    # proxy_pass/fastcgi_pass/drizzle_pass/echo/etc
}
```

ngx_replace_filter

```
location ~ '\.cpp$' {
    # proxy_pass/fastcgi_pass/root/...
    replace_filter_types text/plain;

    # skip C/C++ string literals:
    replace_filter "'(?:\\[^\n]|[^'\n])*'" $& g;
    replace_filter '"(?:\\[^\n]|[^"\n])*"' $& g;

    # remove all those ugly C/C++ comments:
    replace_filter '/\*.*?\*/|//[^\n]*' '' g;
}
```


web server
input
data
output


in
data
pointer
out
4KB buffer

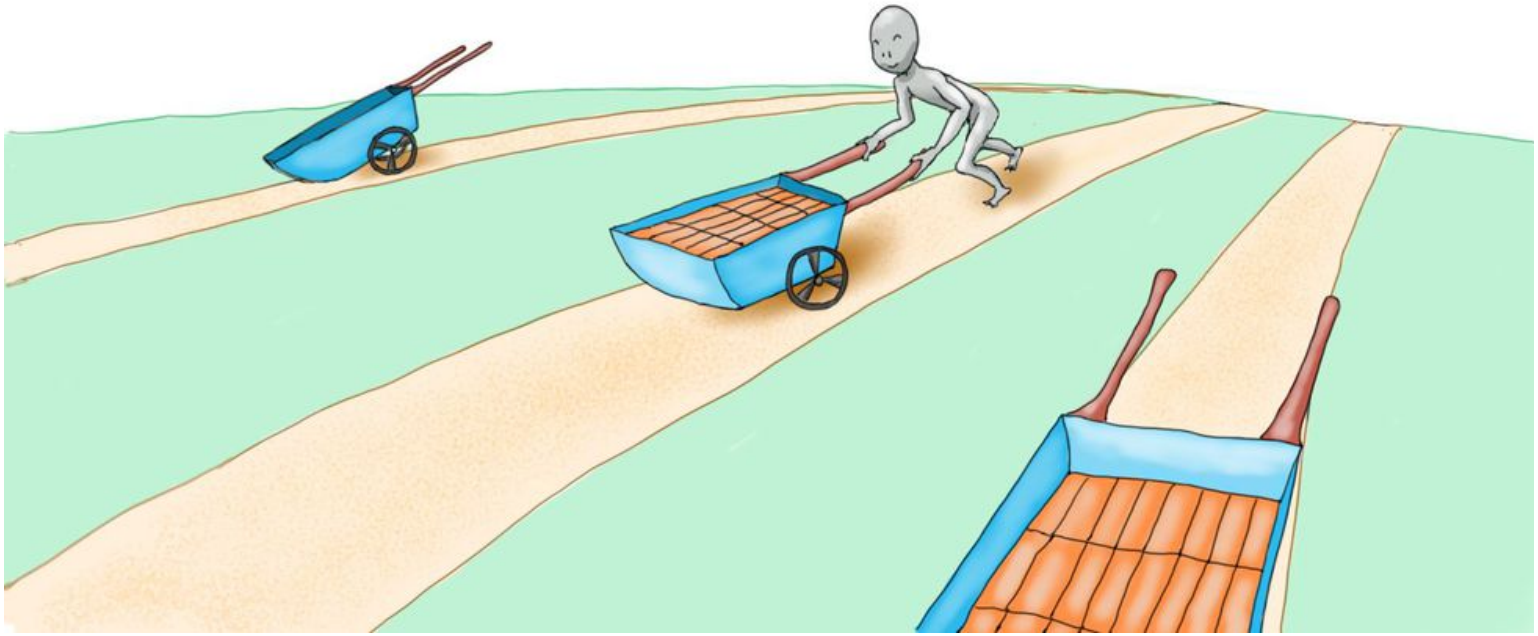
libsregex

nginx.conf *scripting*

ngx_lua

init_by_lua
init_worker_by_lua
set_by_lua
rewrite_by_lua
access_by_lua
content_by_lua
header_filter_by_lua
body_filter_by_lua
log_by_lua

"Light threads" based on Lua *coroutines*



```
ngx.location.capture("/foo")
ngx.location.capture_multi{ {"/foo"}, "/bar"} }
```

```
local sock = ngx.socket.tcp()
local ok, err = sock:connect("127.0.0.1", 1234)
local bytes, err = sock:send("hello")
local data, err = sock:receive("*l")
```

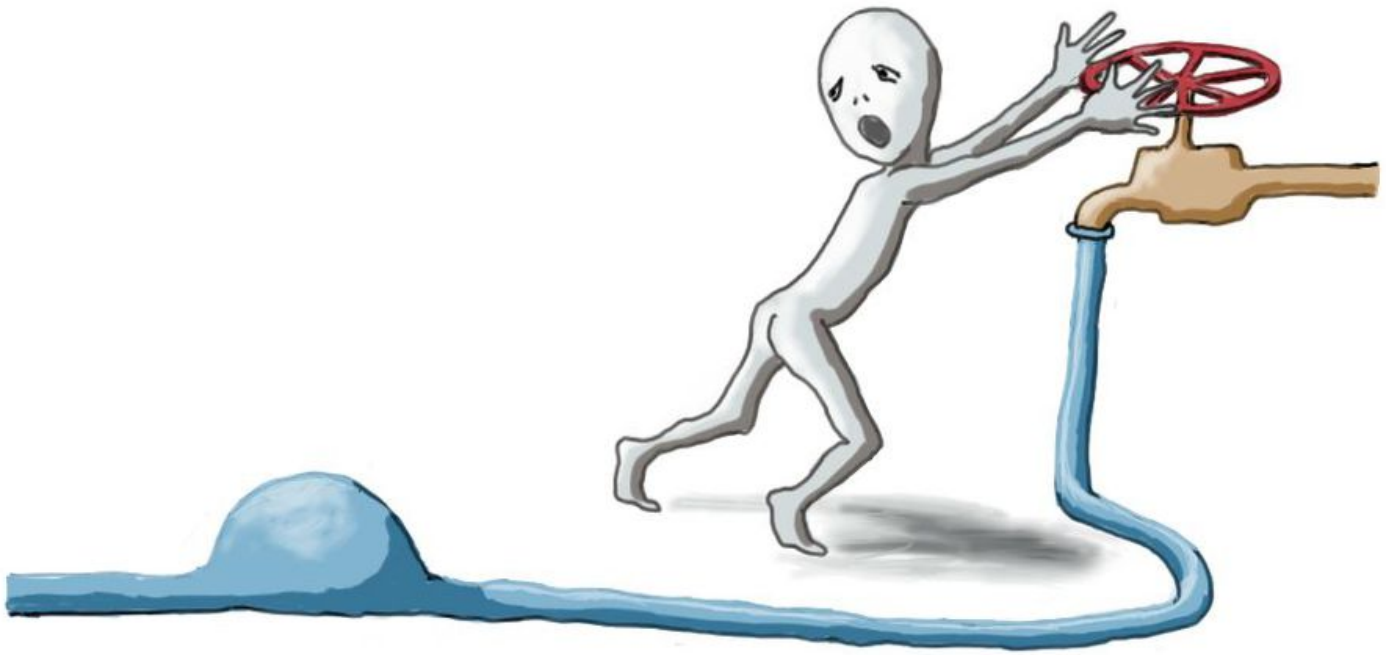
```
local sock = ngx.socket.udp()
local ok, err = sock:setpeer("127.0.0.1", 5432)
local bytes, err = sock:send("my query")
local dgram, err = sock:receive()
```

```
local req_body_sock = ngx.req.socket()
local raw_sock = ngx.req.socket(true)
```

```
local t = ngx.thread.spawn(func)
local ok, res = ngx.thread.wait(t)
```

```
local dogs = ngx.shared.dogs
dogs:add("Tom", 5)
dogs:set("Tom", 7)
dogs:incr("Tom", 1)
dogs:delete("Tom", 1)
```

```
local function do_job()
    ...
end
local ok, err = ngx.timer.at(1.5, do_job)
```



```
while true do
    local chunk, err =
            read_data_chunk_from_backend()
    if not chunk then
        -- error and eof handling
    end

    local ok, err = ngx.print(chunk)
    -- handle errors here if any

    local ok, err = ngx.flush(true)
    -- handle errors here if any
end
```

lua-resty-memcached
lua-resty-redis
lua-resty-mysql
lua-resty-dns
lua-resty-upload
lua-resty-websocket
lua-resty-lock
lua-resty-upstream-healthcheck

LuaJIT

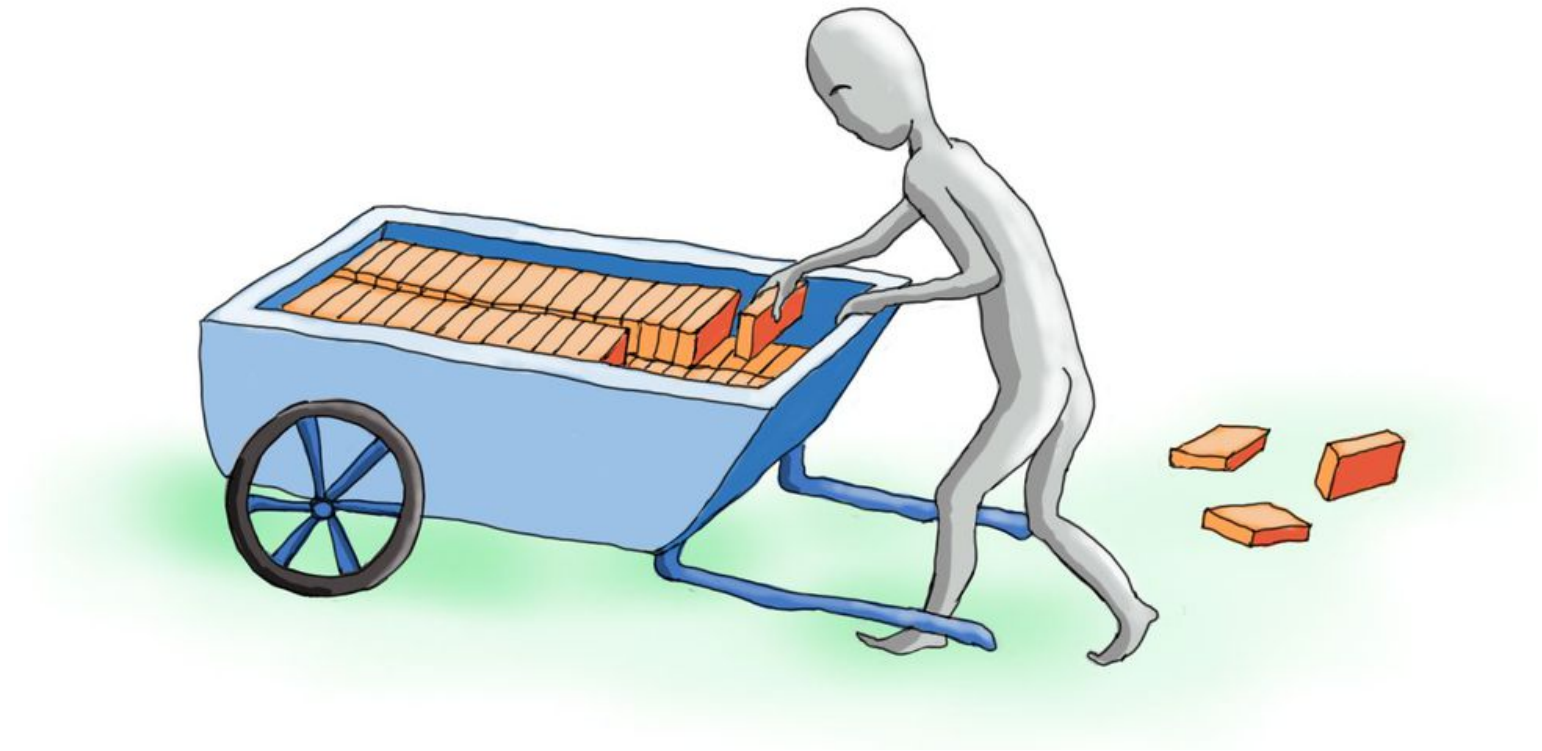
OpenResty

CloudFlare Lua CDN

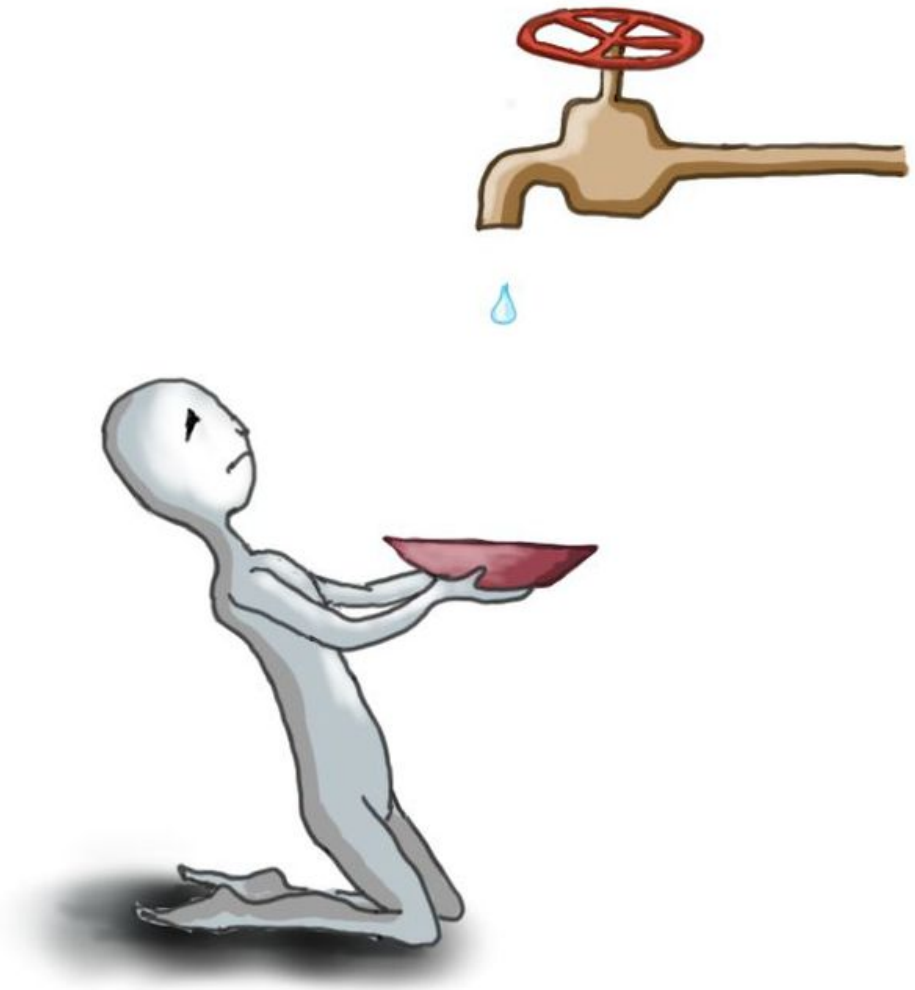
CloudFlare Lua WAF

Testing

No-pool patch for NGINX



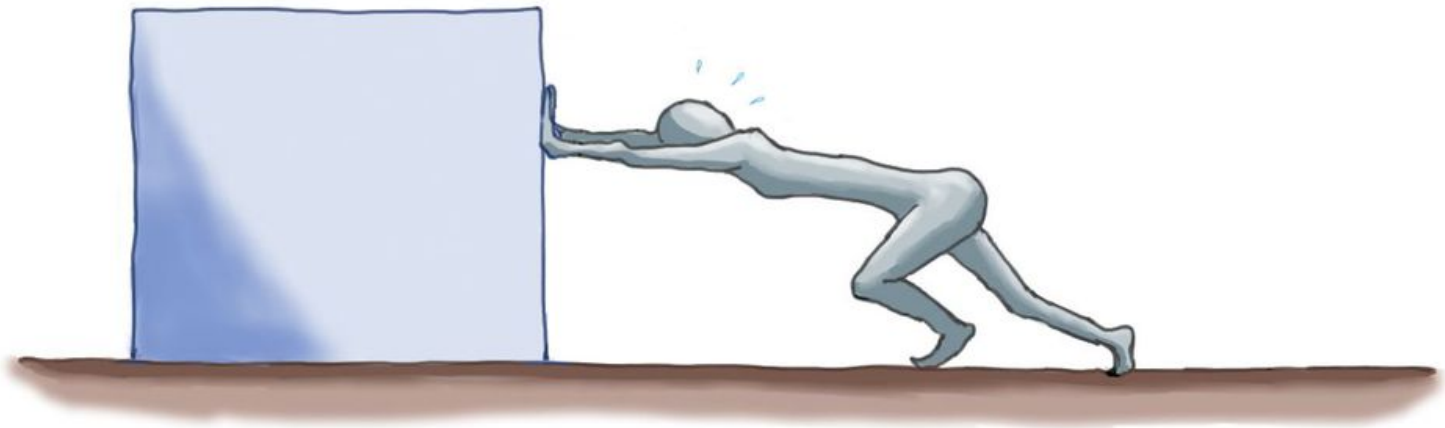
mockeagain



```
export MOCKEAGAIN=r
export LD_PRELOAD=/path/to/mockeagain.so
# edit nginx.conf to add "env" directives
# start nginx and test
```

```
# nginx.conf
env MOCKEAGAIN;
env LD_PRELOAD;
```

Better than the TCP proxy approach



```
export MOCKEAGAIN=w
```

Mocking socket write timeout

```
export MOCKEAGAIN=w
export MOCKEAGAIN_WRITE_TIMEOUT_PATTERN='hello world'
```

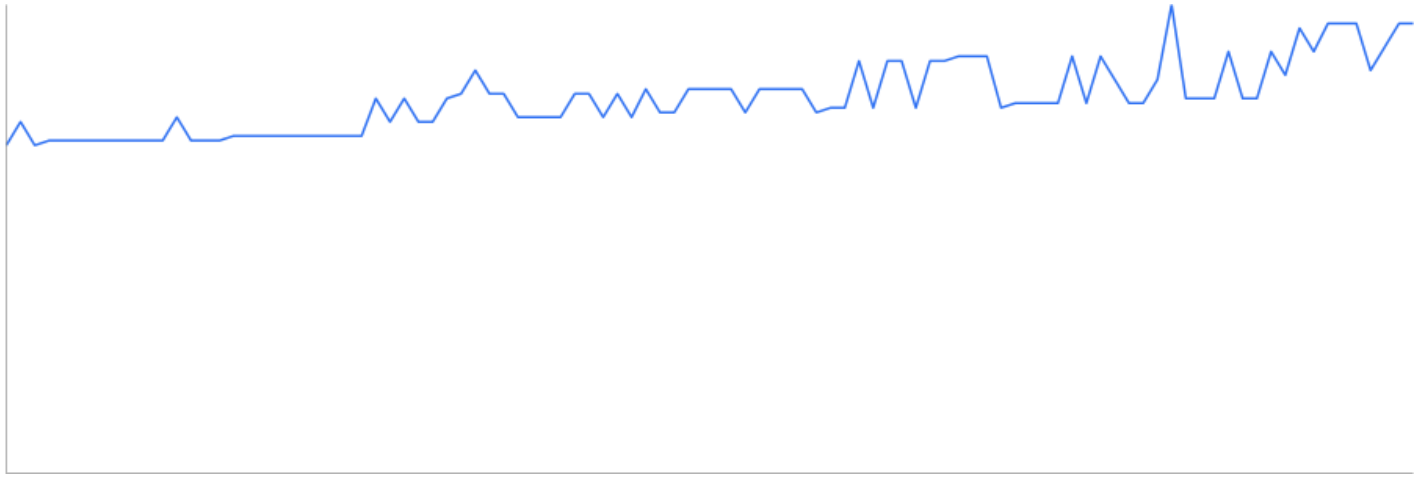
Test::Nginx

```
=== TEST 1: echo hello world
--- config
    location = /t {
        echo "hello world";
    }
--- request
GET /t
--- response_body
hello world
```

```
--- more_headers
User-Agent: Opera/9.80 (Macintosh; Intel Mac OS X 10.7.4; U; en)
--- stap
F(ngx_http_core_content_phase) {
    printf("content: opera: %d\n", $r->headers_in->opera)
}
--- stap_out
content: opera: 0
```

LeakTest: k=10.5
Up-Edges: 52
Stable point: N/A

RSS Samples (in KB):
3596 3860 3604 3608 3612 3616 3620 3624 3628
3628 3632 3636 3904 3648 3652 3656 3660 3668
3672 3676 3680 3684 3688 3692 3696 3700 4096
3844 4104 3848 3852 4116 4120 4388 4128 4132
3876 3880 3884 3888 4156 4160 3904 4168 3912
4176 3920 3928 4192 4192 4196 4200 3944 4208
4212 4216 4220 3964 3968 3972 4496 3980 4508
4512 3992 4520 4524 4528 4532 4536 4016 4020
4024 4028 4032 4560 4044 4568 4312 4056 4060
4320 5116 4068 4072 4076 4604 4084 4088 4620
4356 4884 4628 4892 4900 4900 4384 4648 4916



Test Cluster on Amazon EC2

opsboy

```
$ ./dispatcher -r -t 170 -a 'linux x86_64' ngx_lua
Requires at least 5 machines.
bucket 1: tl-ngx_lua force=1 (242 min)
bucket 2: twv-ngx_lua tw-ngx_lua to-
ngx_lua force=1 (160 min)
bucket 3: trv-ngx_lua tr-ngx_lua force=1 (125 min)
bucket 4: tv-ngx_lua t-ngx_lua force=1 (118 min)
bucket 5: thv-ngx_lua th-ngx_lua force=1 (74 min)
```

qa.openresty.org

(This page was automatically generated by the tools in the opsboy project on 20:08:16 05-Feb-2014 GMT.)

Amazon EC2 Test Cluster Reports for Nginx and its Components

Reports

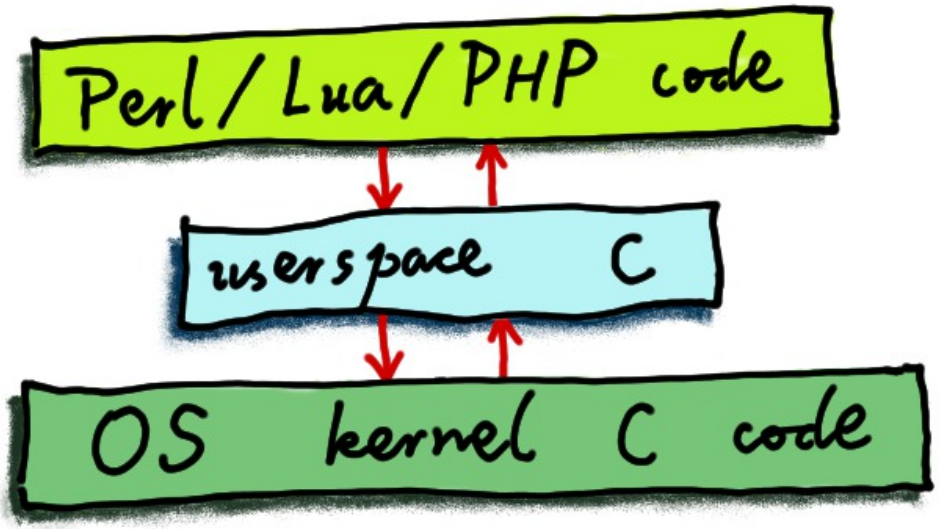
Linux x86_64

nginx/1.5.9 (no pool)

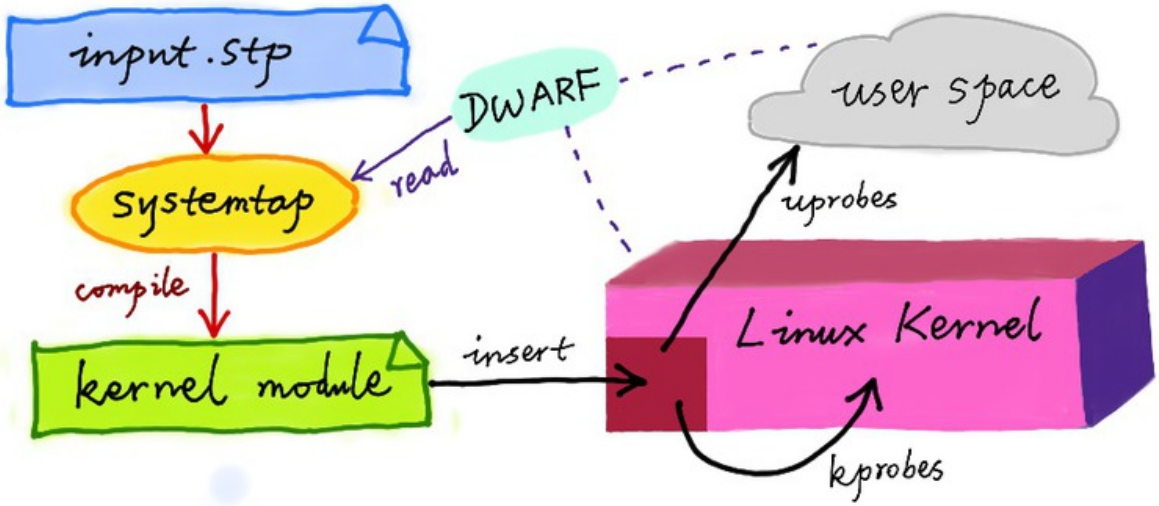
Component	plain	valgrind/memcheck	mockeagain(R)	mockeagain(R) valgrind/memcheck
lua-resty-core	PASS NEW	PASS NEW	PASS NEW	PASS NEW
lua-resty-dns	PASS	PASS	PASS	PASS
lua-resty-lock	PASS	PASS	PASS	PASS
lua-resty-logger-socket	PASS	PASS	PASS	PASS
lua-resty-memcached	PASS	PASS	PASS	PASS

Online debugging and profiling

systemtap



The Software Stack


input.stp
DWARF
user space
read
Systemtap
uprobes
compile
Linux Kernel
kernel module
insert
kprobes

nginx-systemtap-toolkit
stapxx (stap++)

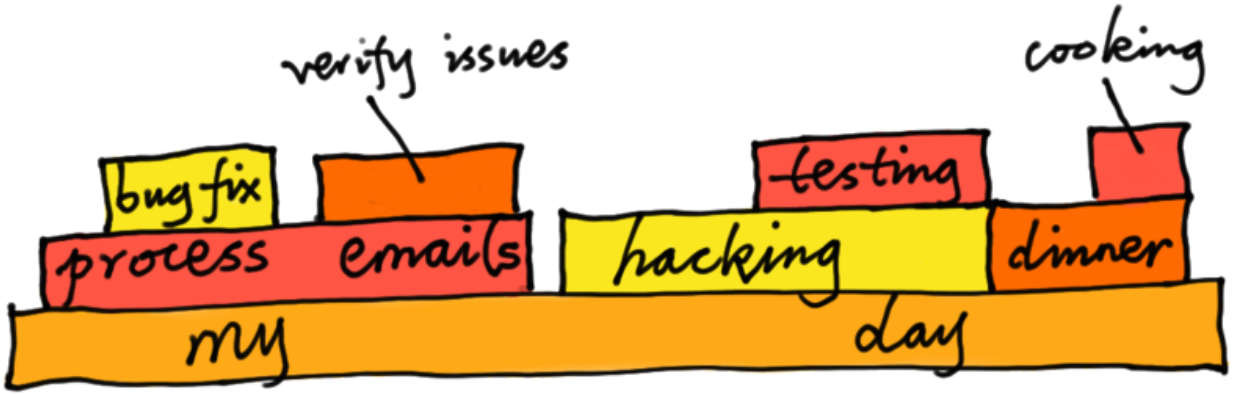
```
$ ./ngx-req-distr -c -
m `cat /opt/nginx/logs/nginx.pid`
Tracing 4394 4395 4396 4397 (/opt/nginx/sbin/nginx)...
Hit Ctrl-C to end.
^C
worker 4394:    0 reqs,       0 conns
worker 4395:    2100 reqs,    21 conns
worker 4396:    501 reqs,     6 conns
worker 4397:    2100 reqs,    21 conns
```

```
# show the details of the shm zone named "dogs"
$ ./ngx-shm -p 15218 -n dogs
Tracing 15218 (/opt/nginx/sbin/nginx)...

shm zone "dogs"
    owner: ngx_http_lua_shdict
    total size: 100 KB
    free pages: 88 KB (22 pages, 1 blocks)

22 microseconds elapsed in the probe.
```

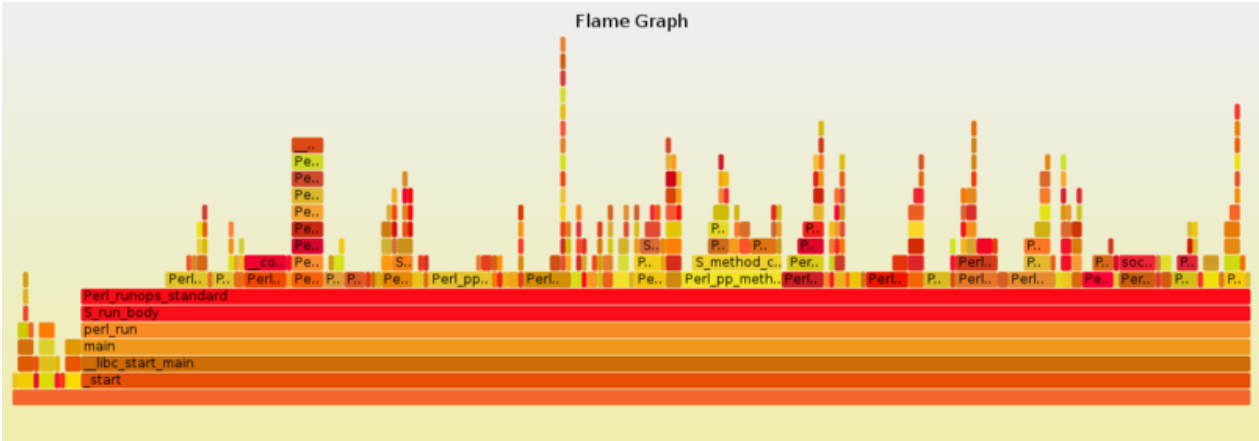
```
$ ngx-rps.sxx -x 19647
WARNING: Tracing process 19647.
Hit Ctrl-C to end.
[1376939543] 300 req/sec
[1376939544] 235 req/sec
[1376939545] 235 req/sec
[1376939546] 166 req/sec
```

```
$ epoll-loop-blocking-distr.sxx -x 19647 --arg time=60
Start tracing 19647...
Please wait for 60 seconds.
Distribution of epoll loop blocking latencies (in milliseconds)
max/avg/min: 1097/0/0
value |-------------------------------------------------- count
    0 |@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@  18471
    1 |@@@@@@@@                                           3273
    2 |@                                                   473
    4 |                                                    119
    8 |                                                     67
   16 |                                                     51
   32 |                                                     35
   64 |                                                     20
  128 |                                                     23
  256 |                                                      9
  512 |                                                      2
 1024 |                                                      2
 2048 |                                                      0
```

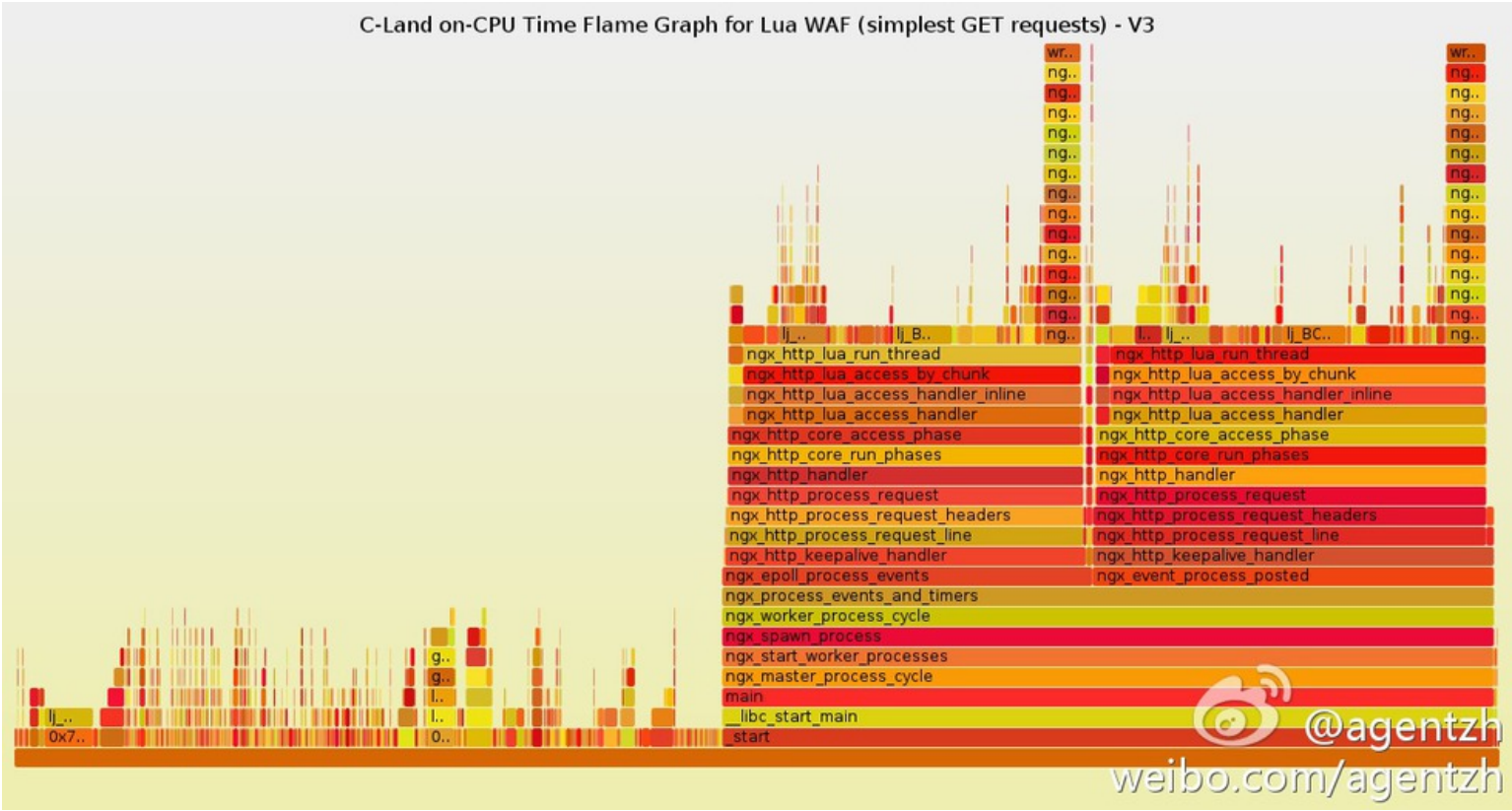
Flame Graphs

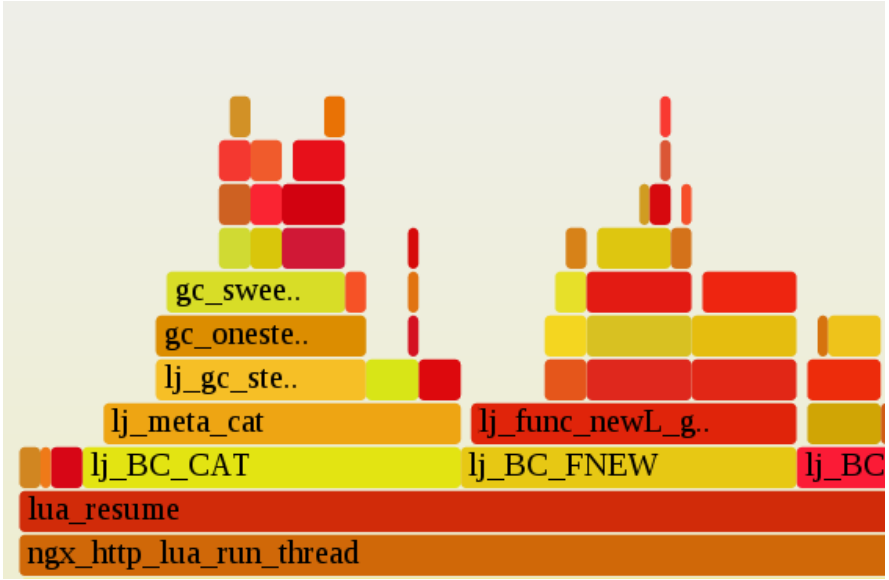


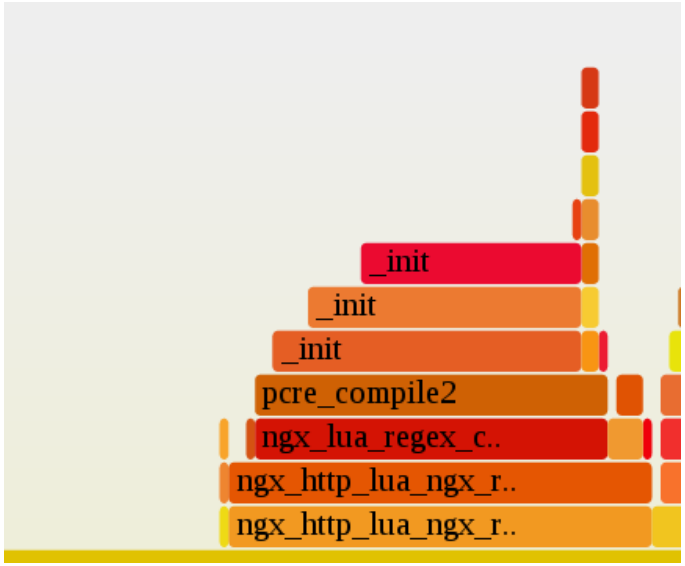
Flame Graph for My Day

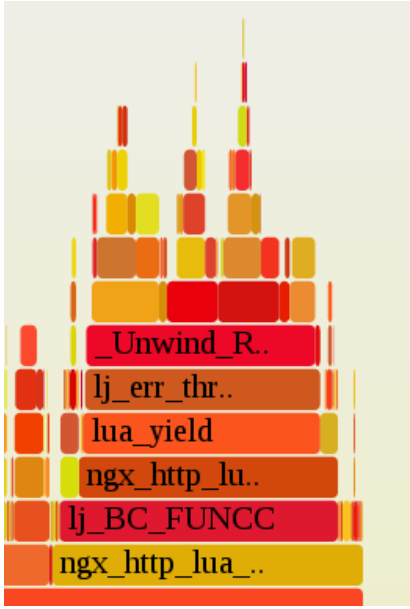

Flame Graph
Perl_runops_standard
S_run_body
perl_run
main
__libc_start_main
_start

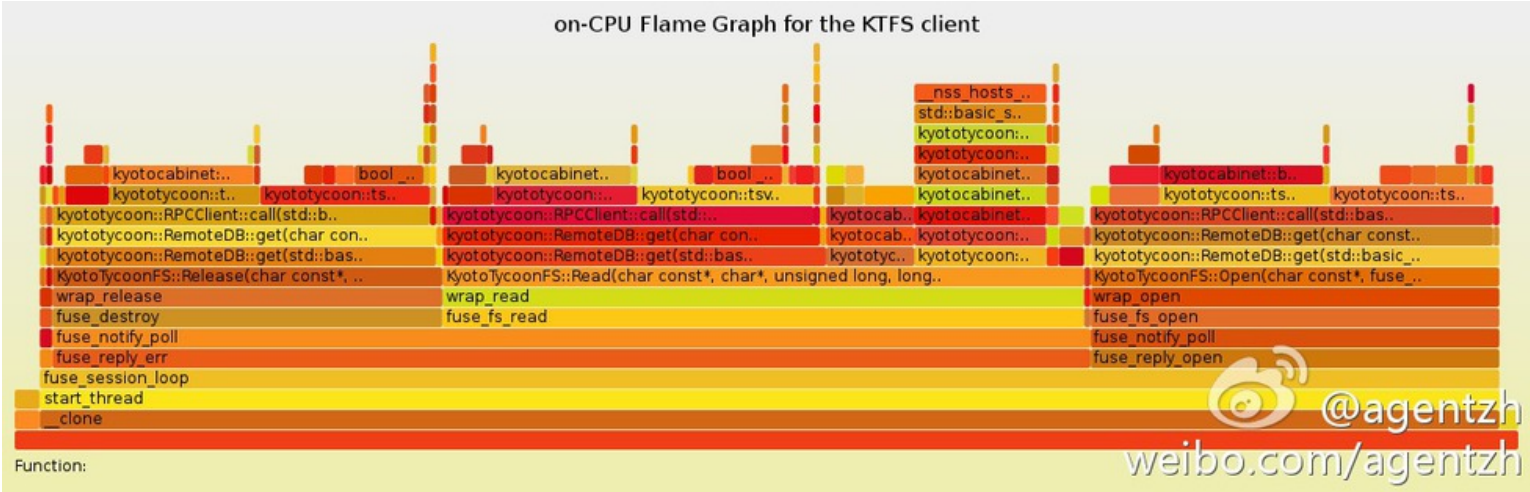
on-CPU Flame Graphs

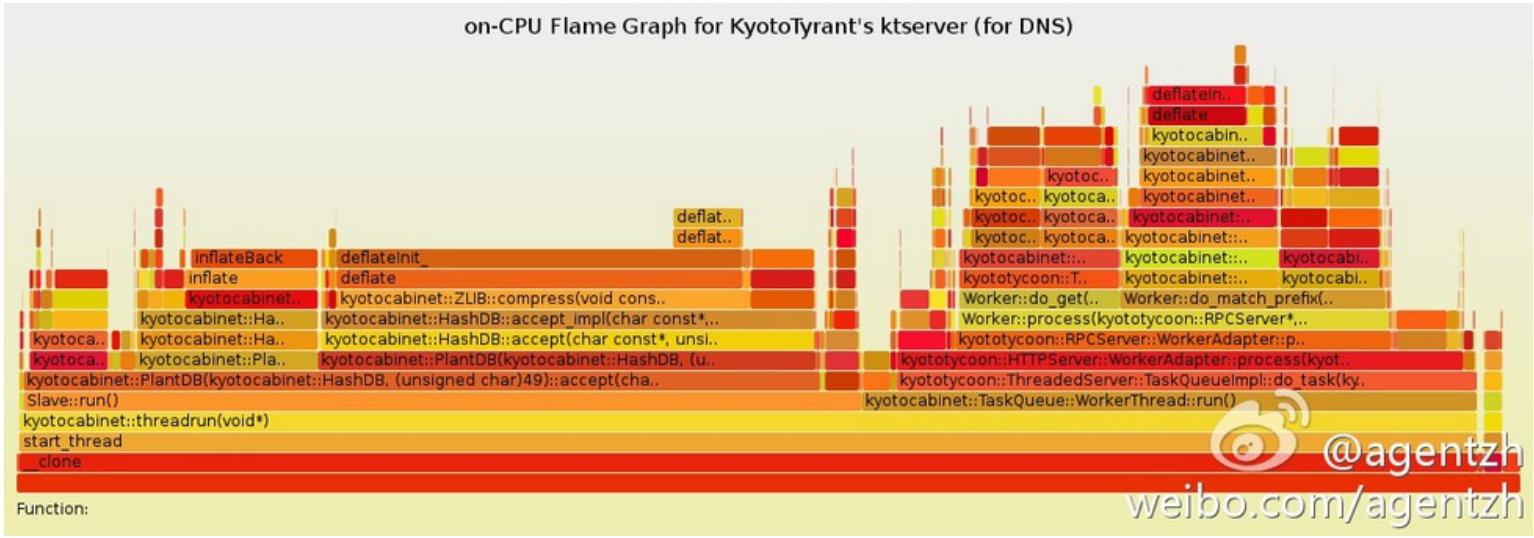

C-Land on-CPU Time Flame Graph for Lua WAF (simplest GET requests) - V3
ngx_http_lua_run_thread
ngx_http_lua_access_by_chunk
ngx_http_lua_access_handler_inline
ngx_http_lua_access_handler
ngx_http_core_access_phase
ngx_http_core_run_phases
ngx_http_handler
ngx_http_process_request
ngx_http_process_request_headers
ngx_http_process_request_line
ngx_http_keepalive_handler
ngx_epoll_process_events
ngx_event_process_posted
ngx_process_events_and_timers
ngx_worker_process_cycle
ngx_spawn_process
ngx_start_worker_processes
ngx_master_process_cycle
main
__libc_start_main
_start
@agentzh
weibo.com/agentzh


gc_swee..
gc_oneste..
lj_gc_ste..
lj_meta_cat
lj_func_newL_g..
lj_BC_CAT
lj_BC_FNEW
lj_BC
lua_resume
ngx_http_lua_run_thread

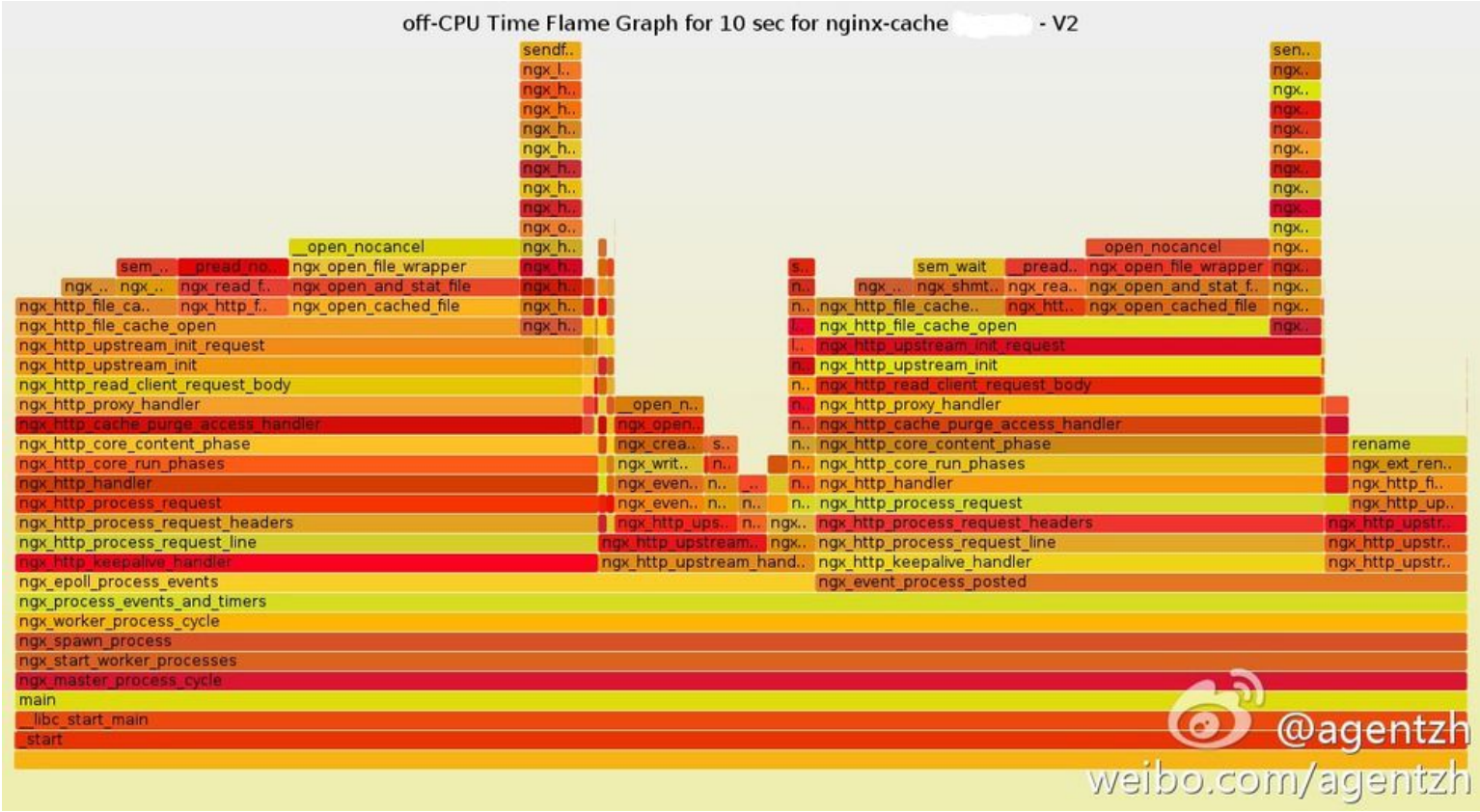

_init
_init
_init
pcre_compile2
ngx_lua_regex_c..
ngx_http_lua_ngx_r..
ngx_http_lua_ngx_r..

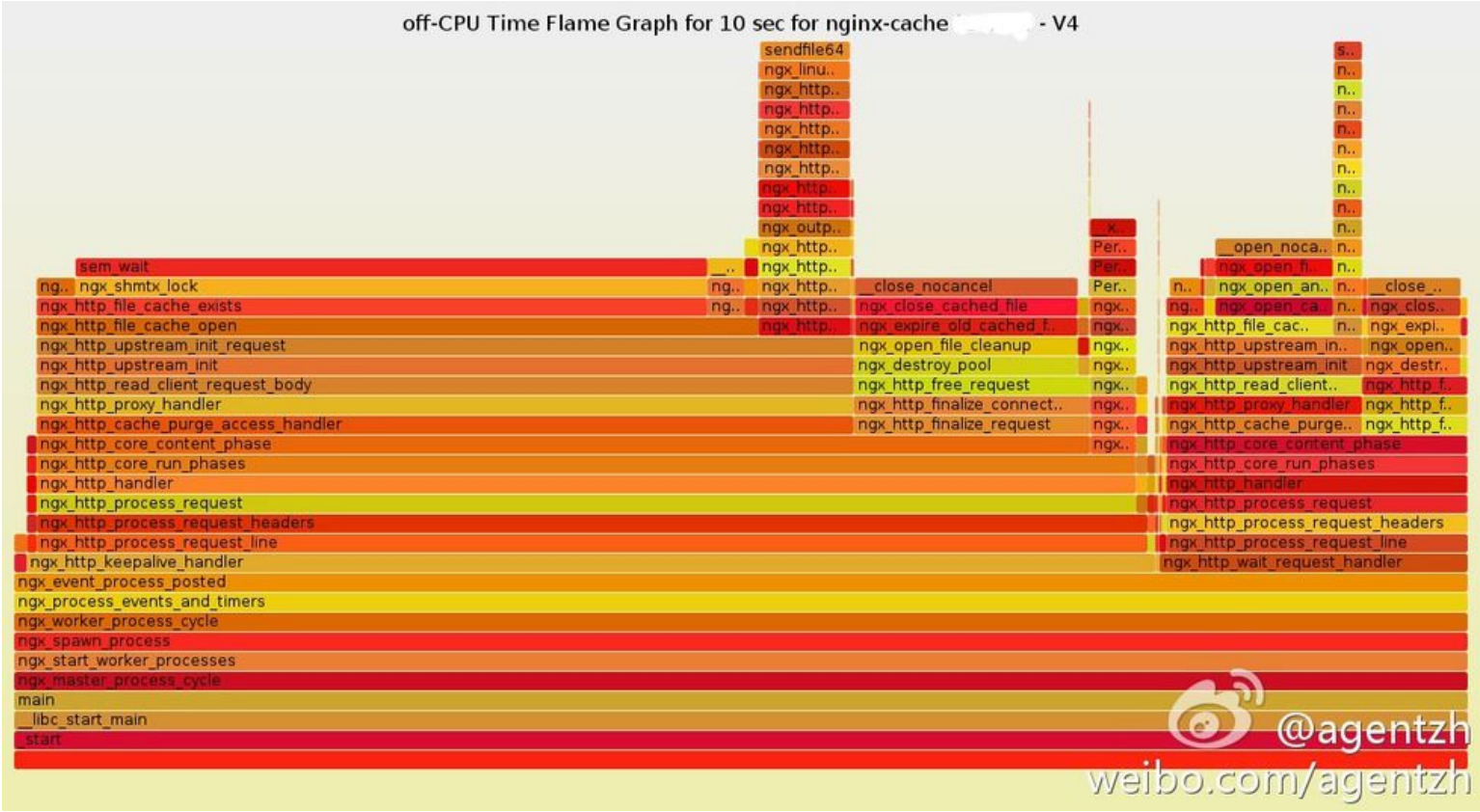

_Unwind_R..
lj_err_thr..
lua_yield
ngx_http_lu..
lj_BC_FUNCC
ngx_http_lua_..

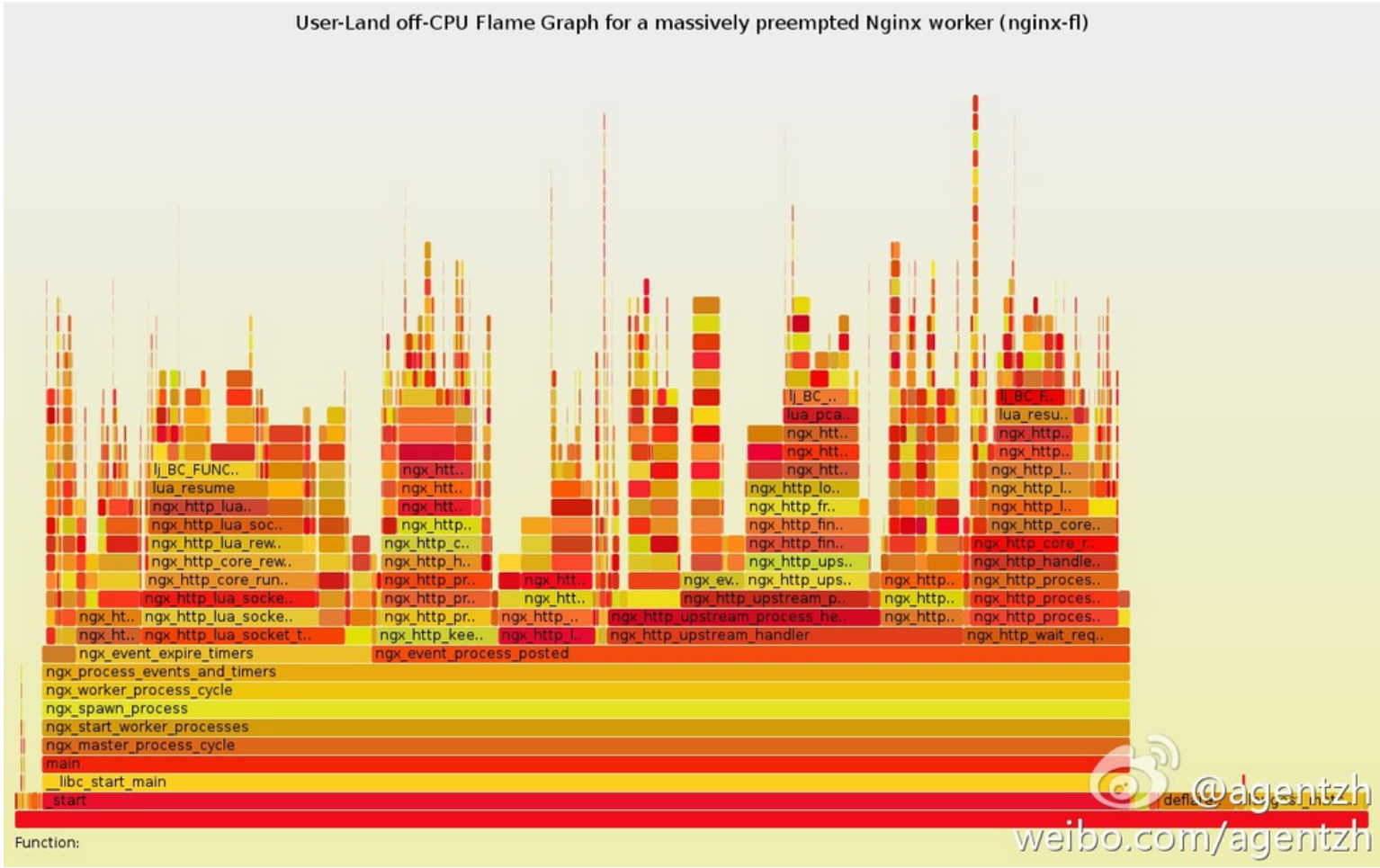

on-CPU Flame Graph for the KTFS client
__nss_hosts_..
std::basic_s..
kyototycoon::..
kyototycoon::..
kyotocabinet::..
bool _..
kyotocabinet..
bool _..
kyotocabinet..
kyotocabinet::b..
kyototycoon::t..
kyototycoon::ts..
kyototycoon::..
kyototycoon::tsv..
kyotocabinet..
kyototycoon::ts..
kyototycoon::ts..
kyototycoon::RPCClient::call(std::b..
kyototycoon::RPCClient::call(std::..
kyotocab..
kyotocabinet..
kyototycoon::RPCClient::call(std::bas..
kyototycoon::RemoteDB::get(char con..
kyototycoon::RemoteDB::get(char con..
kyotocab..
kyototycoon::..
kyototycoon::RemoteDB::get(char const..
kyototycoon::RemoteDB::get(std::bas..
kyototycoon::RemoteDB::get(std::bas..
kyototyc..
kyototycoon::..
kyototycoon::RemoteDB::get(std::basic_..
KyotoTycoonFS::Release(char const*, ..
KyotoTycoonFS::Read(char const*, char*, unsigned long, long..
KyotoTycoonFS::Open(char const*, fuse_..
wrap_release
wrap_read
wrap_open
fuse_destroy
fuse_fs_read
fuse_fs_open
fuse_notify_poll
fuse_notify_poll
fuse_reply_err
fuse_reply_open
fuse_session_loop
start_thread
__clone
Function:
@agentzh
weibo.com/agentzh


on-CPU Flame Graph for KyotoTyrant's ktserver (for DNS)
deflatein..
deflate
kyotocabin..
kyotocabinet..
kyotoc..
kyotocabinet..
kyotoc..
kyotoca..
kyotocabinet..
deflat..
deflat..
kyotoc..
kyotoca..
kyotocabinet::..
kyotoc..
kyotoca..
kyotocabinet::..
kyotocabinet::..
kyotocabinet::..
kyotocabi..
inflateBack
deflateInit_
inflate
deflate
kyototycoon::T..
kyotocabinet::..
kyotocabi..
kyotocabinet..
kyotocabinet::ZLIB::compress(void cons..
Worker::do_get(..
Worker::do_match_prefix(..
kyotocabinet::Ha..
kyotocabinet::HashDB::accept_impl(char const*,..
Worker::process(kyototycoon::RPCServer*,..
kyotoca..
kyotocabinet::Ha..
kyotocabinet::HashDB::accept(char const*, unsi..
kyototycoon::RPCServer::WorkerAdapter::p..
kyotoca..
kyotocabinet::Pla..
kyotocabinet::PlantDB(kyotocabinet::HashDB, (u..
kyototycoon::HTTPServer::WorkerAdapter::process(kyot..
kyotocabinet::PlantDB(kyotocabinet::HashDB, (unsigned char)49)::accept(cha..
kyototycoon::ThreadedServer::TaskQueueImpl::do_task(ky..
Slave::run()
kyotocabinet::TaskQueue::WorkerThread::run()
kyotocabinet::threadrun(void*)
start_thread
__clone
Function:
@agentzh
weibo.com/agentzh

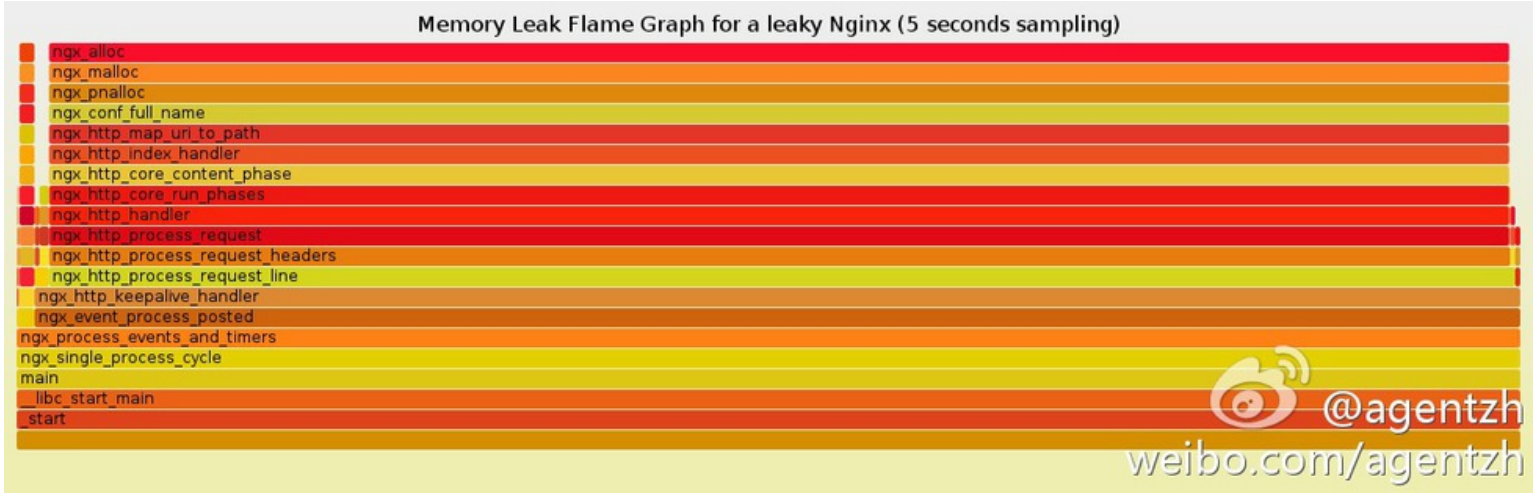
off-CPU Flame Graphs


off-CPU Time Flame Graph for 10 sec for nginx-cache - V2
sendf..
ngx_l..
ngx_h..
__open_nocancel
sem_..
__pread_no..
ngx_open_file_wrapper
ngx_..
ngx_..
ngx_read_f..
ngx_open_and_stat_file
ngx_http_file_ca..
ngx_http_f..
ngx_open_cached_file
ngx_o..
ngx_http_file_cache_open
ngx_http_upstream_init_request
ngx_http_upstream_init
ngx_http_read_client_request_body
ngx_http_proxy_handler
ngx_http_cache_purge_access_handler
ngx_http_core_content_phase
ngx_http_core_run_phases
ngx_http_handler
ngx_http_process_request
ngx_http_process_request_headers
ngx_http_process_request_line
ngx_http_keepalive_handler
ngx_epoll_process_events
ngx_process_events_and_timers
ngx_worker_process_cycle
ngx_spawn_process
ngx_start_worker_processes
ngx_master_process_cycle
main
__libc_start_main
_start
__open_n..
ngx_open..
ngx_crea..
ngx_writ..
ngx_even..
ngx_even..
ngx_http_ups..
ngx_http_upstream..
ngx_http_upstream_hand..
sen..
sem_wait
__pread..
ngx_shmt..
ngx_rea..
ngx_http_file_cache..
ngx_htt..
ngx_event_process_posted
rename
ngx_ext_ren..
ngx_http_fi..
ngx_http_up..
ngx_http_upstr..
@agentzh
weibo.com/agentzh

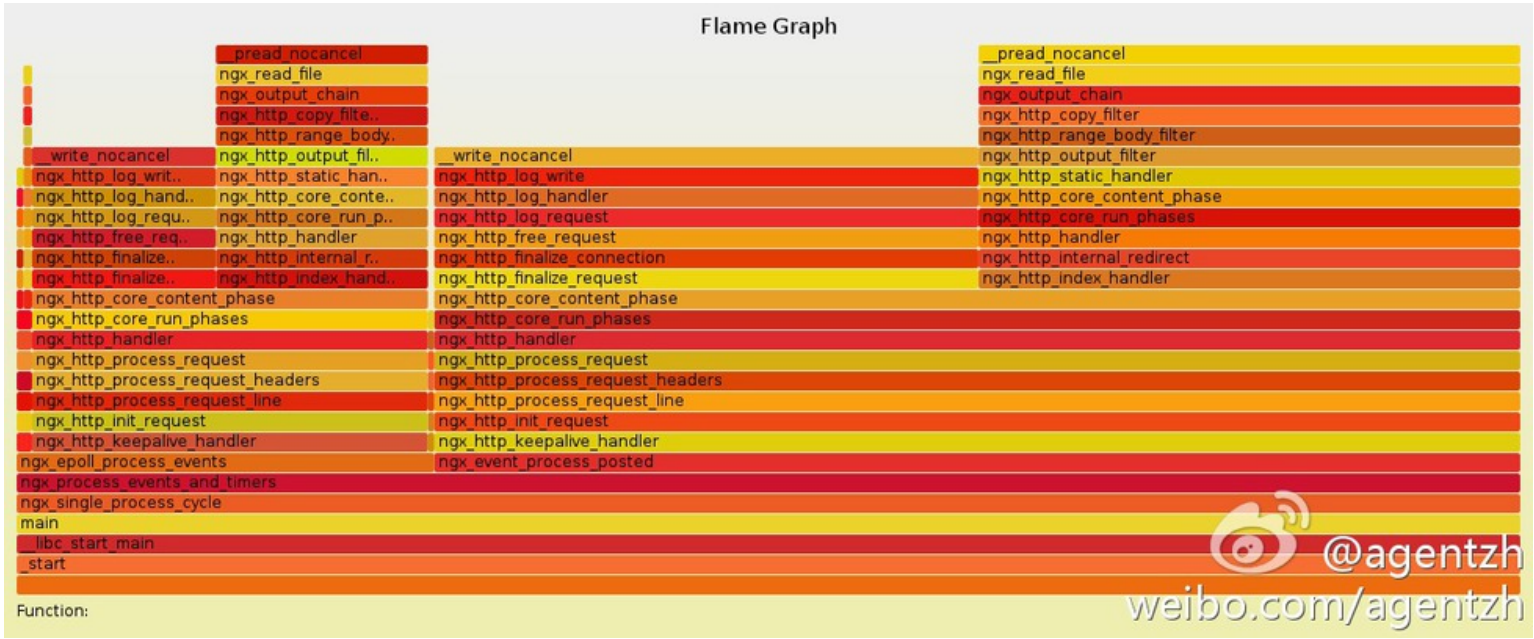

off-CPU Time Flame Graph for 10 sec for nginx-cache - V4
sendfile64
ngx_linu..
ngx_http..
sem_wait
ngx_shmtx_lock
ngx_http_file_cache_exists
ngx_http_file_cache_open
ngx_http_upstream_init_request
ngx_http_upstream_init
ngx_http_read_client_request_body
ngx_http_proxy_handler
ngx_http_cache_purge_access_handler
ngx_http_core_content_phase
ngx_http_core_run_phases
ngx_http_handler
ngx_http_process_request
ngx_http_process_request_headers
ngx_http_process_request_line
ngx_http_keepalive_handler
ngx_event_process_posted
ngx_process_events_and_timers
ngx_worker_process_cycle
ngx_spawn_process
ngx_start_worker_processes
ngx_master_process_cycle
main
__libc_start_main
_start
ngx_outp..
__close_nocancel
ngx_close_cached_file
ngx_expire_old_cached_f..
ngx_open_file_cleanup
ngx_destroy_pool
ngx_http_free_request
ngx_http_finalize_connect..
ngx_http_finalize_request
__open_noca..
ngx_open_fi..
ngx_open_an..
ngx_open_ca..
ngx_http_file_cac..
ngx_http_upstream_in..
ngx_http_upstream_init
ngx_http_read_client..
ngx_http_proxy_handler
ngx_http_cache_purge..
ngx_http_core_content_phase
ngx_http_core_run_phases
ngx_http_handler
ngx_http_process_request
ngx_http_process_request_headers
ngx_http_process_request_line
ngx_http_wait_request_handler
__close_..
ngx_clos..
ngx_expi..
ngx_open..
ngx_destr..
ngx_http_f..
@agentzh
weibo.com/agentzh


User-Land off-CPU Flame Graph for a massively preempted Nginx worker (nginx-fl)
lj_BC_FUNC..
lua_resume
ngx_http_lua..
ngx_http_lua_soc..
ngx_http_lua_rew..
ngx_http_core_rew..
ngx_http_core_run..
ngx_http_lua_socke..
ngx_ht..
ngx_http_lua_socke..
ngx_ht..
ngx_http_lua_socket_t..
ngx_event_expire_timers
ngx_process_events_and_timers
ngx_worker_process_cycle
ngx_spawn_process
ngx_start_worker_processes
ngx_master_process_cycle
main
__libc_start_main
_start
ngx_htt..
ngx_htt..
ngx_htt..
ngx_http..
ngx_http_c..
ngx_http_h..
ngx_http_pr..
ngx_http_pr..
ngx_http_pr..
ngx_http_kee..
ngx_event_process_posted
ngx_htt..
ngx_htt..
ngx_http_..
ngx_http_l..
lj_BC_..
lua_pca..
ngx_htt..
ngx_htt..
ngx_htt..
ngx_http_lo..
ngx_http_fr..
ngx_http_fin..
ngx_http_fin..
ngx_http_ups..
ngx_ev..
ngx_http_ups..
ngx_http_upstream_p..
ngx_http_upstream_process_he..
ngx_http_upstream_handler
ngx_http..
ngx_http..
ngx_http..
lj_BC_F..
lua_resu..
ngx_http..
ngx_http..
ngx_http_l..
ngx_http_l..
ngx_http_l..
ngx_http_core..
ngx_http_core_r..
ngx_http_handle..
ngx_http_proces..
ngx_http_proces..
ngx_http_proces..
ngx_http_wait_req..
Function:
@agentzh
weibo.com/agentzh

Memory Leak Flame Graphs


Memory Leak Flame Graph for a leaky Nginx (5 seconds sampling)
ngx_alloc
ngx_malloc
ngx_pnalloc
ngx_conf_full_name
ngx_http_map_uri_to_path
ngx_http_index_handler
ngx_http_core_content_phase
ngx_http_core_run_phases
ngx_http_handler
ngx_http_process_request
ngx_http_process_request_headers
ngx_http_process_request_line
ngx_http_keepalive_handler
ngx_event_process_posted
ngx_process_events_and_timers
ngx_single_process_cycle
main
__libc_start_main
_start
@agentzh
weibo.com/agentzh

File IO Flame Graphs


Flame Graph
__pread_nocancel
ngx_read_file
ngx_output_chain
ngx_http_copy_filte..
ngx_http_range_body..
__write_nocancel
ngx_http_log_writ..
ngx_http_log_hand..
ngx_http_log_requ..
ngx_http_free_req..
ngx_http_finalize..
ngx_http_finalize..
ngx_http_output_fil..
ngx_http_static_han..
ngx_http_core_conte..
ngx_http_core_run_p..
ngx_http_handler
ngx_http_internal_r..
ngx_http_index_hand..
__write_nocancel
ngx_http_log_write
ngx_http_log_handler
ngx_http_log_request
ngx_http_free_request
ngx_http_finalize_connection
ngx_http_finalize_request
__pread_nocancel
ngx_read_file
ngx_output_chain
ngx_http_copy_filter
ngx_http_range_body_filter
ngx_http_output_filter
ngx_http_static_handler
ngx_http_core_content_phase
ngx_http_core_run_phases
ngx_http_handler
ngx_http_internal_redirect
ngx_http_index_handler
ngx_http_core_content_phase
ngx_http_core_run_phases
ngx_http_handler
ngx_http_process_request
ngx_http_process_request_headers
ngx_http_process_request_line
ngx_http_init_request
ngx_http_keepalive_handler
ngx_epoll_process_events
ngx_event_process_posted
ngx_process_events_and_timers
ngx_single_process_cycle
main
__libc_start_main
_start
Function:
@agentzh
weibo.com/agentzh

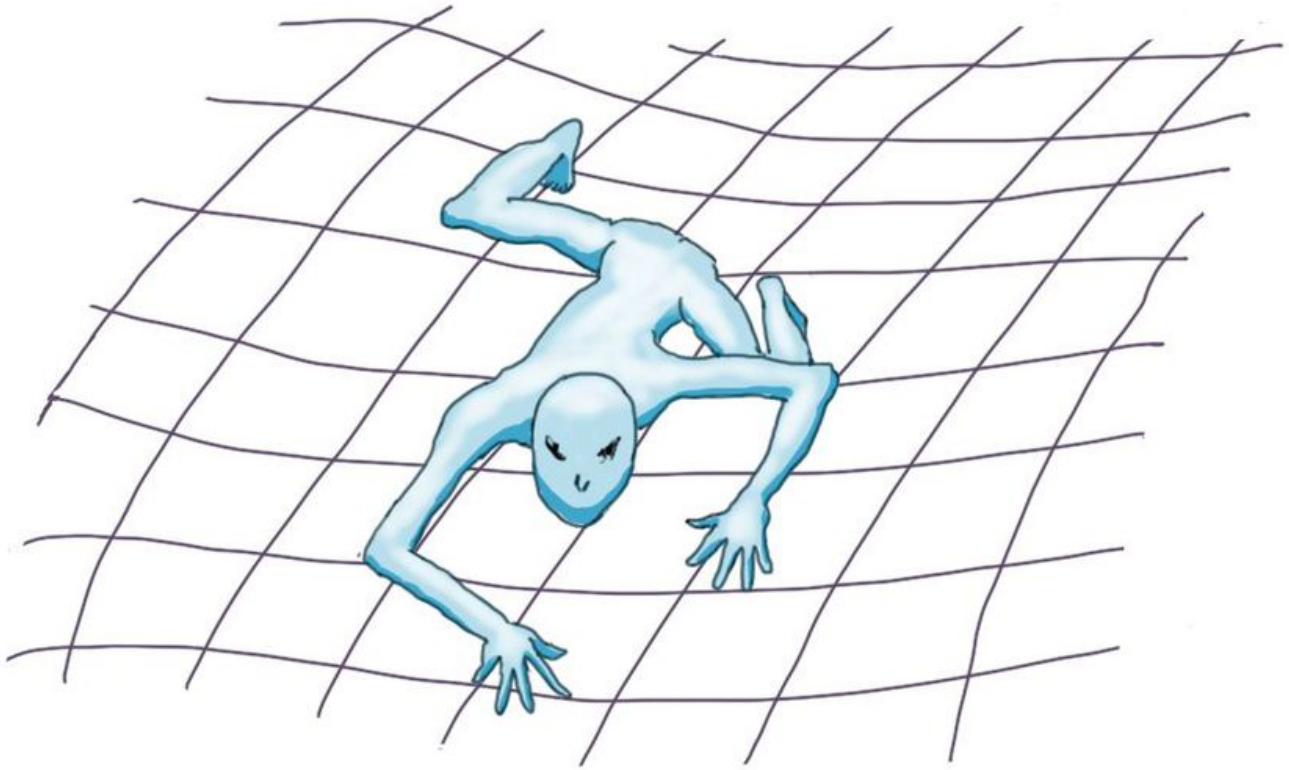
DWARF debug info format

nginx-gdb-utils

```
(gdb) source luajit21.py

(gdb) lvmst
current VM state: C code from intperpreted Lua

(gdb) lbt
builtin#166
builtin#195
builtin#187
@/home/agentzh/git/lua-resty-
core/lib/resty/core/regex.lua:588
content_by_lua:10
```

```
(gdb) source ngx-raw-req.py
(gdb) ngx-raw-req r
GET /foo HTTP/1.0
Host: bar.com
User-Agent: curl
Accept-Encoding: gzip
```

Spider binary data structures in the process space



We are hiring :)

☺ *Any questions*? ☺