Application of *Lua* in Nginx

Lua 在 Nginx 中的应用

☺ *agentzh@gmail.com* ☺
章亦春 (agentzh)

2015.04

淘宝量子店铺统计V3.0 - 店铺概况 - Windows Internet Explorer

http://lz.taobao.com/#shopsummary/

文件(F) 编辑(E) 查看(V) 收藏夹(A) 工具(T) 帮助(H)

淘宝量子店铺统计V3.0 - 店铺概况

您好， ！ 退出

淘宝网首页 | 我的淘宝 | 收藏量子 | 搜索 | 网站导航 | 帮助

量子统计 测试版
lz.taobao.com

最新公告：淘宝量子店铺统计V3.0测试版震撼登场，诚邀免费试用！ NEW

跟淘宝说几句 去论坛提问

店铺基本
- 店铺概况
- 实时客户访问
- 按小时流量分析
- 按天流量分析
- 宝贝被访排行
- 分类页被访排行
- 店内搜索关键词

营销效果
- 流量来源构成
- 淘宝搜索关键词

客户分析
- 访客地区分析

百宝箱
- 个性化统计图标

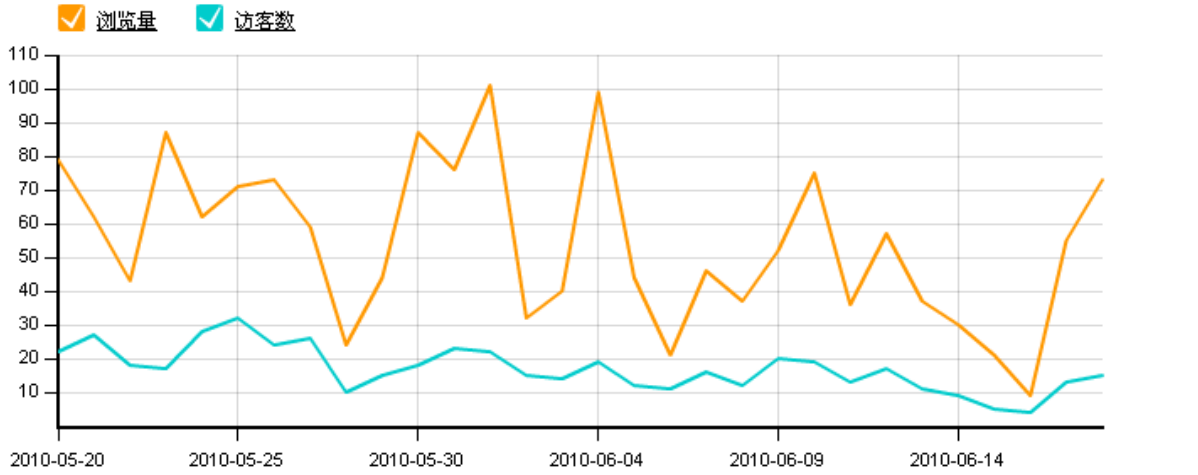
添加功能模块

管理功能模块

流量概况

按天 按小时

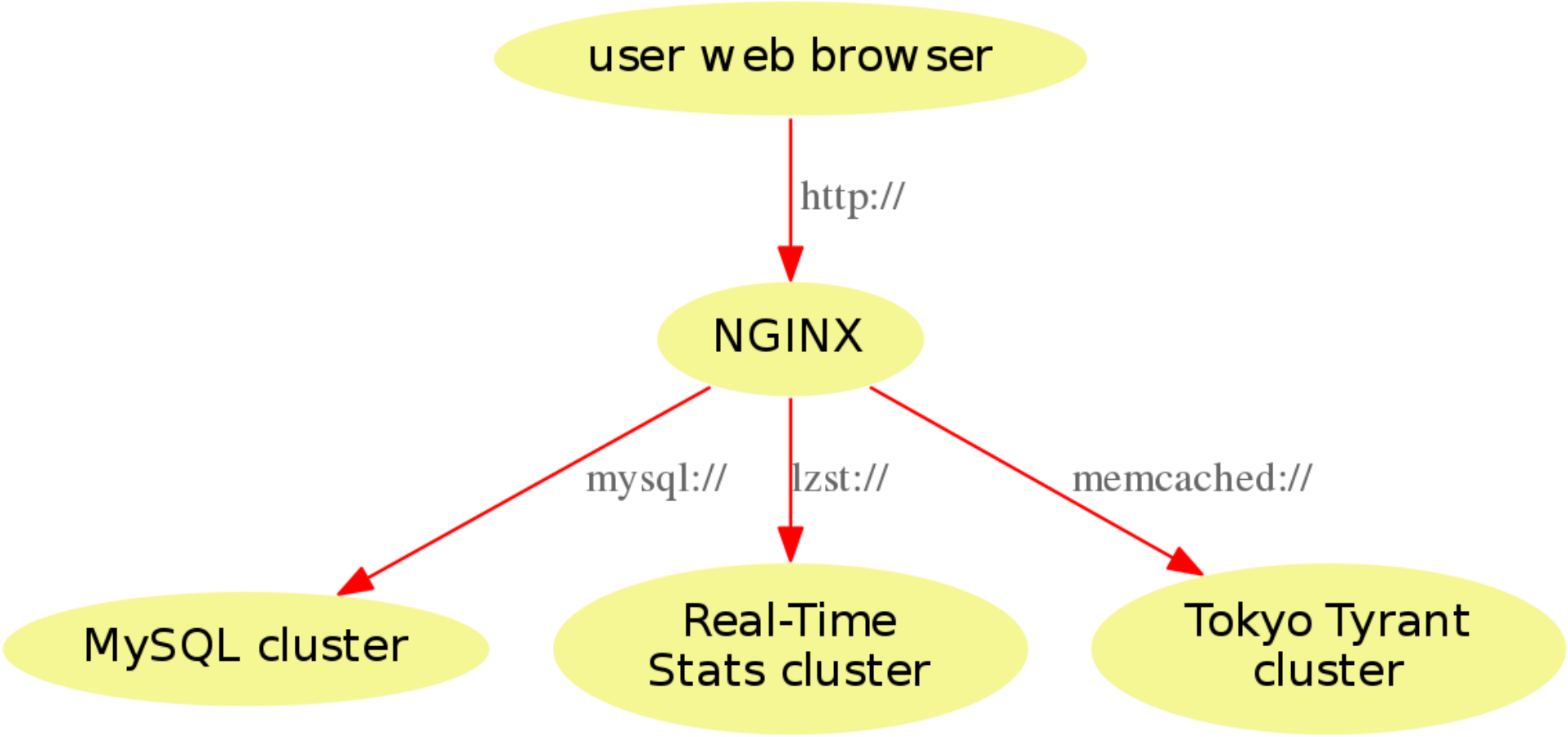
详细信息>>



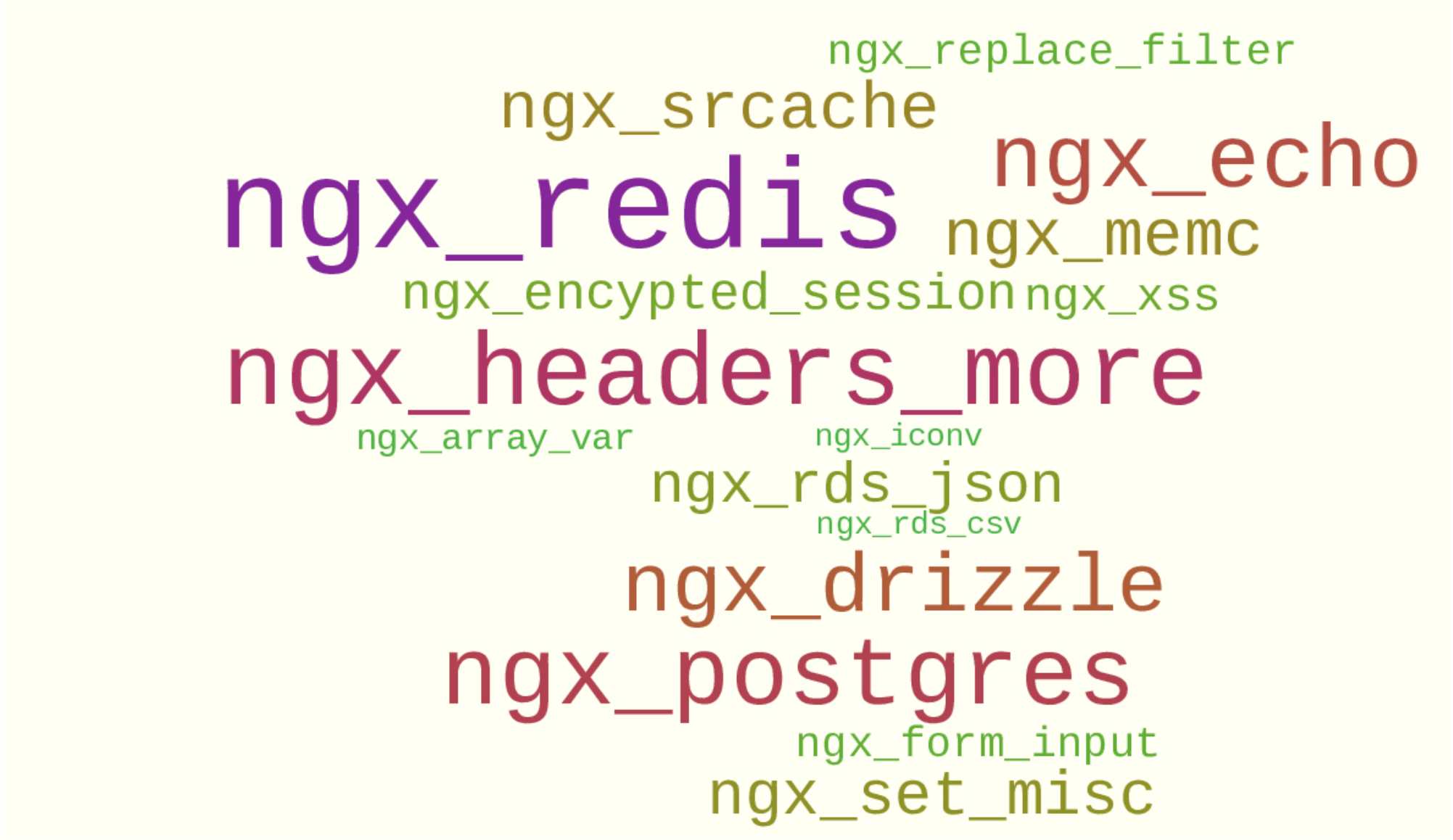
数据汇总

	浏览量（PV）	访客数（UV）	人均访问页面数	人均店内停留时间（秒）	浏览回头率

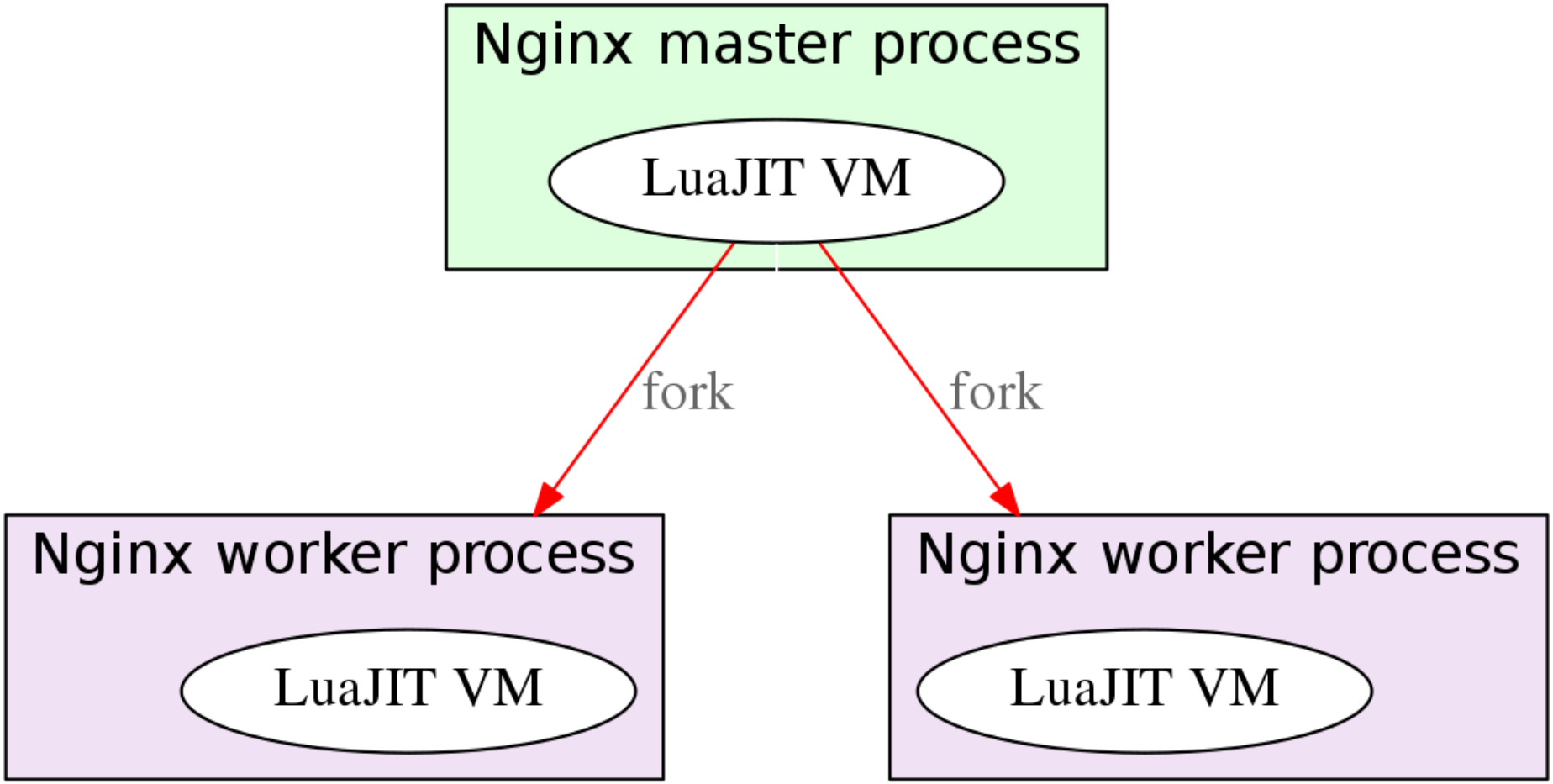
完成 Internet 100%



Architecture Diagram for lz.taobao.com (year 2011)


ngx_replace_filter
ngx_srcache
ngx_redis
ngx_echo
ngx_memc
ngx_encypted_session
ngx_xss
ngx_headers_more
ngx_array_var
ngx_iconv
ngx_rds_json
ngx_rds_csv
ngx_drizzle
ngx_postgres
ngx_form_input
ngx_set_misc


NGINX
LUA



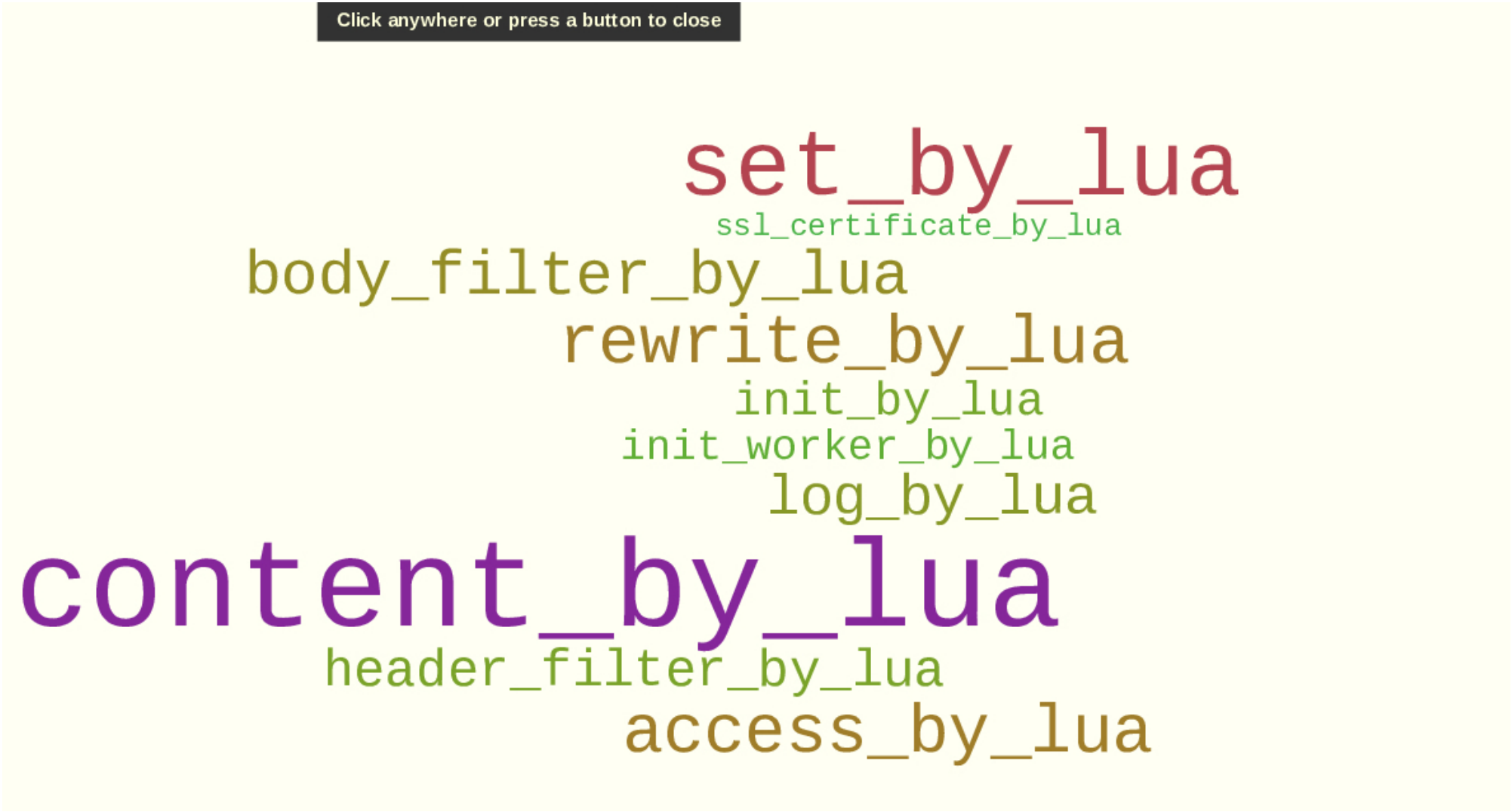
How ngx_lua works

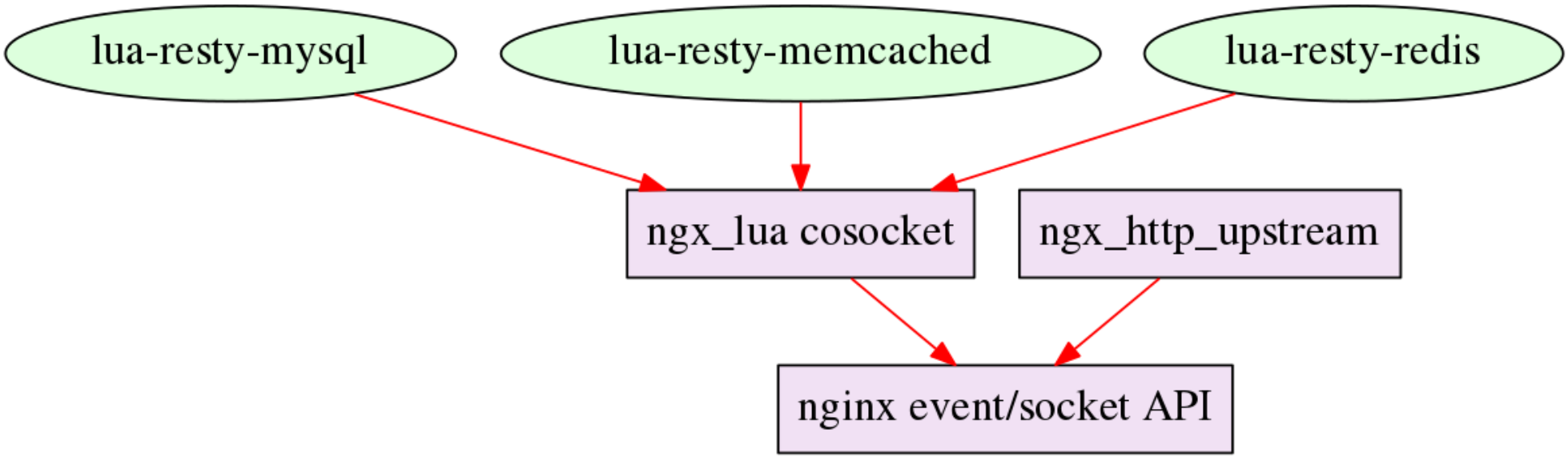
☺ No "callback hell".

☺ 100% nonblocking IO

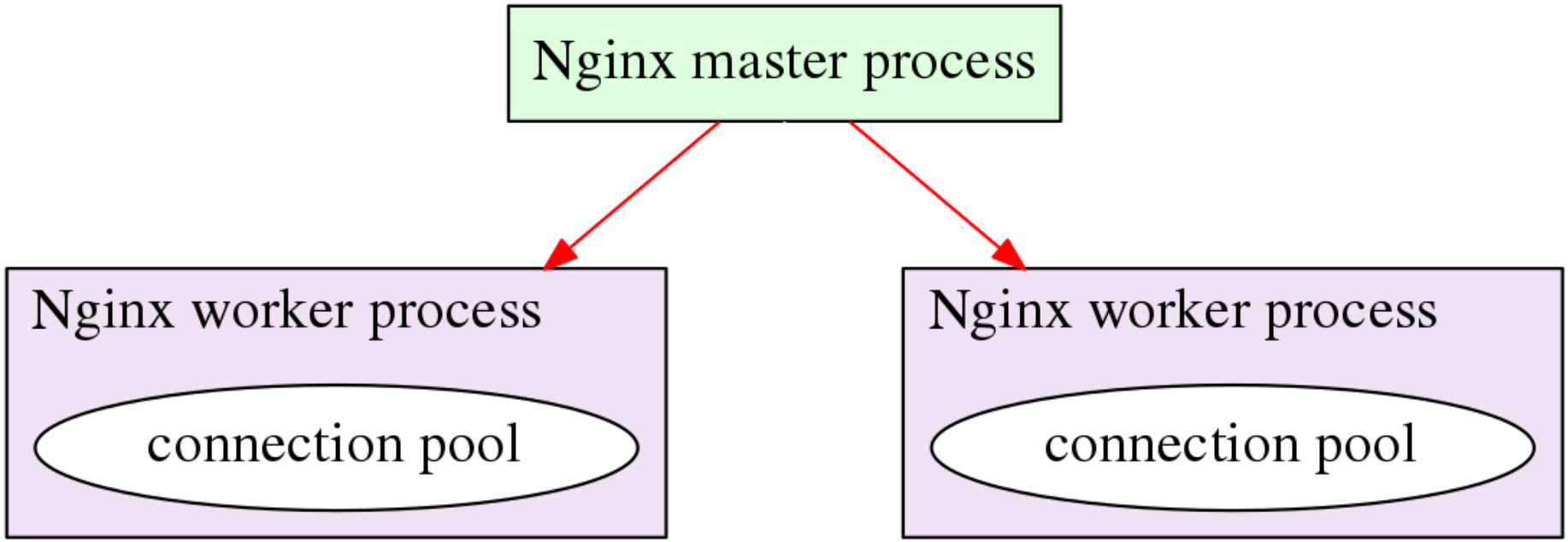
没有“回调地狱”。

100% 非阻塞I/O.

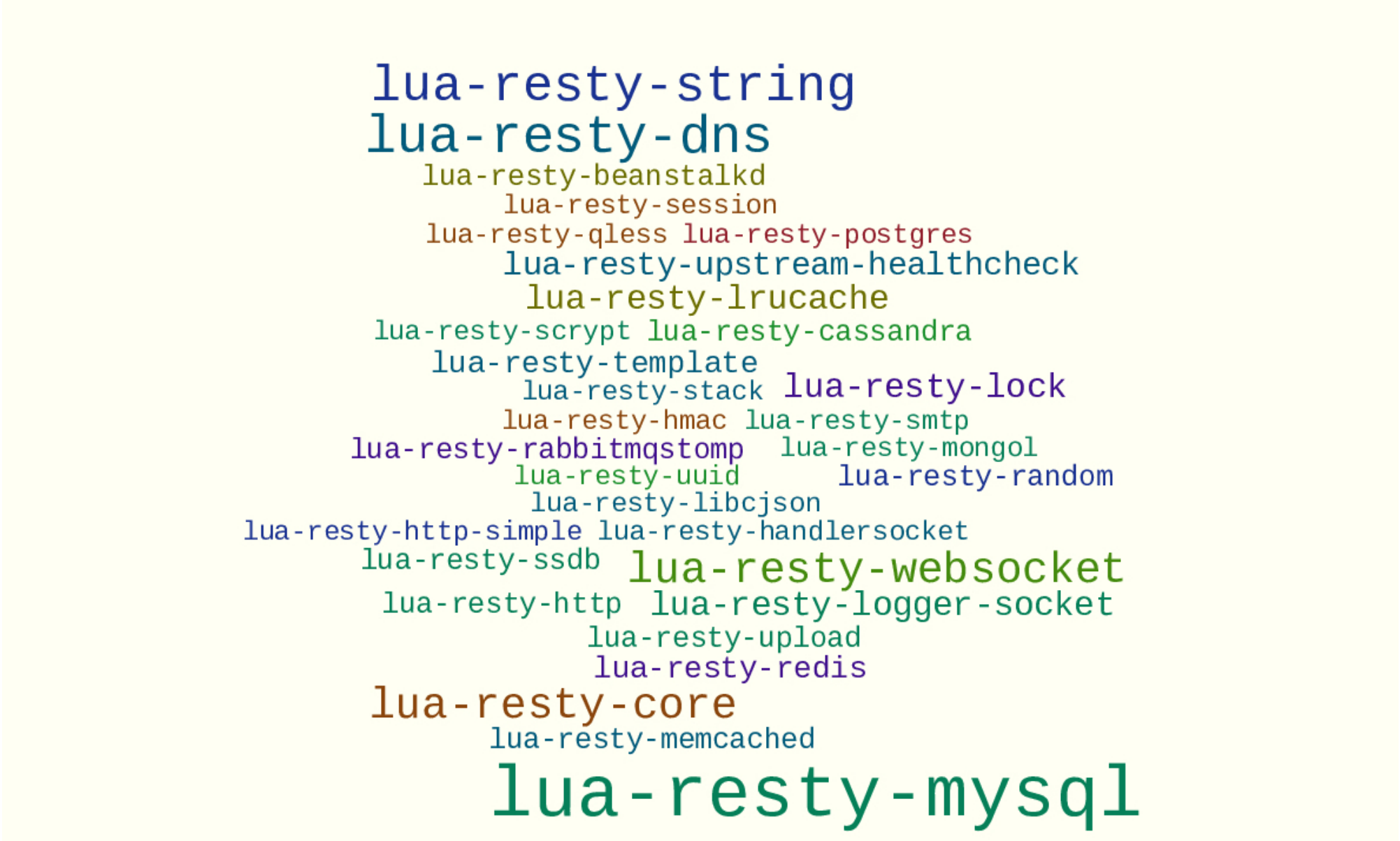




ngx_lua cosocket mechanism



Nginx Multi-Worker Model and Cosocket Connection Pools


lua-resty-string
lua-resty-dns
lua-resty-beanstalkd
lua-resty-session
lua-resty-qless
lua-resty-postgres
lua-resty-upstream-healthcheck
lua-resty-lrucache
lua-resty-scrypt
lua-resty-cassandra
lua-resty-template
lua-resty-stack
lua-resty-lock
lua-resty-hmac
lua-resty-smtp
lua-resty-rabbitmqstomp
lua-resty-mongol
lua-resty-uuid
lua-resty-random
lua-resty-libcjson
lua-resty-http-simple
lua-resty-handlersocket
lua-resty-ssdb
lua-resty-websocket
lua-resty-http
lua-resty-logger-socket
lua-resty-upload
lua-resty-redis
lua-resty-core
lua-resty-memcached
lua-resty-mysql

 openresty.org

 qa.openresty.org

OpenResty 测试集群

(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

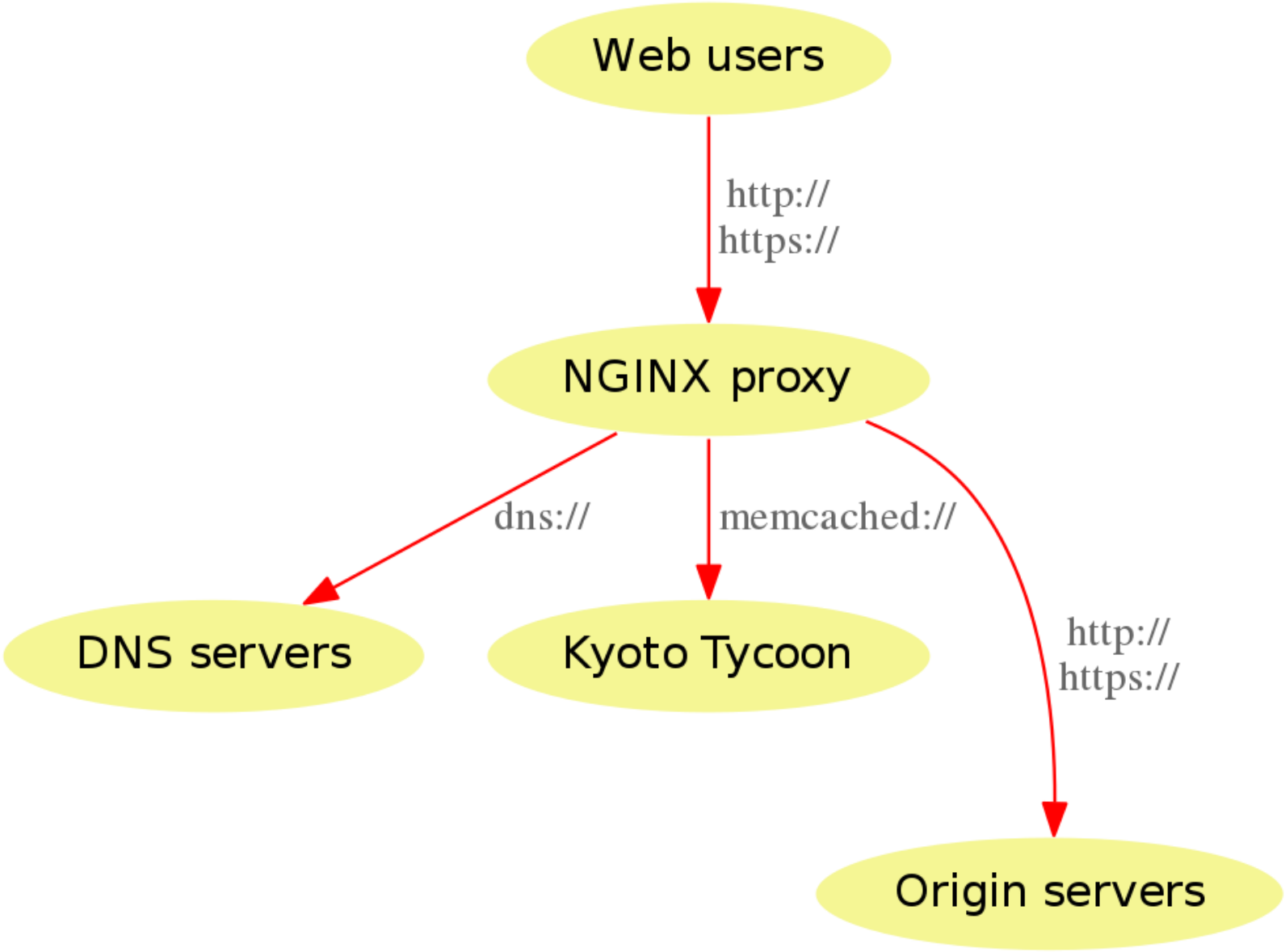
Reports

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

```
$ ./dispatcher -r -t 170 -a 'linux x86_64' ngx_echo ngx_lua
Requires at least 5 machines.
bucket 1: tl-ngx_lua (264 min)
bucket 2: twv-ngx_lua tw-ngx_lua trv-ngx_echo th-ngx_echo (167 min)
bucket 3: trv-ngx_lua th-ngx_lua t-ngx_lua tv-ngx_echo tr-ngx_echo (162 min)
bucket 4: tv-ngx_lua tl-ngx_echo to-ngx_lua thv-ngx_echo t-ngx_echo (145 min)
bucket 5: thv-ngx_lua tr-ngx_lua twv-ngx_echo tw-ngx_echo to-ngx_echo (90 min)
...
```

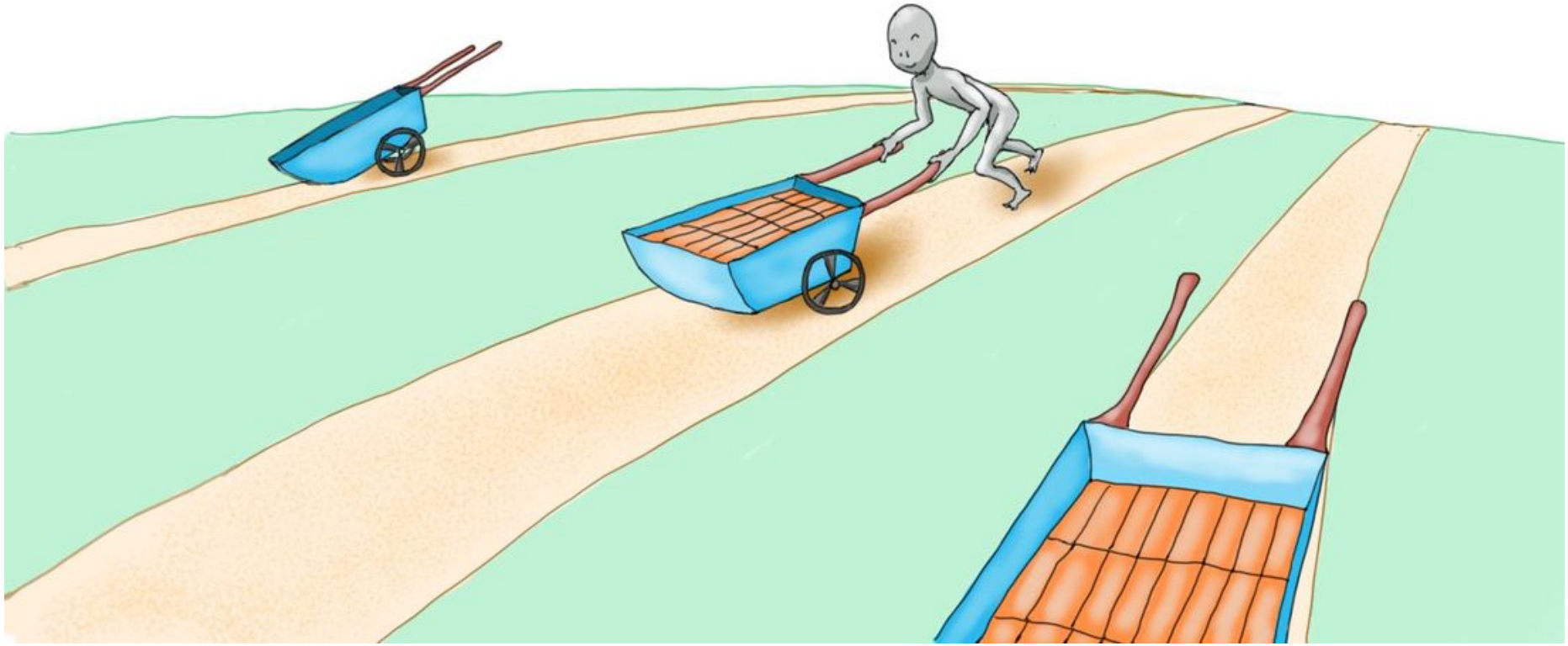


Basic Architecture Diagram for CloudFlare (year 2012)

- ☺ Lua SSL
- ☺ Lua CDN
- ☺ Lua WAF (Lua Web 防火墙)

☺ Light threads

轻量级线程



```
local thread_A, err =
        ngx.thread.spawn(func1)

-- thread_A keeps running asynchronously
-- in the background of the current
-- "light thread".
```

```
local ok, res1, res2 =
            ngx.thread.wait(thread_A, thread_B)
```

```
local ok, err = ngx.thread.kill(thread_A)
```

☺ lua-resty-websocket

WebSocket 服务器和客户端

```
local server = require "resty.websocket.server"

local wb, err = server:new{
    timeout = 5000,  -- in milliseconds
    max_payload_len = 65535,
}

local data, typ, err = wb:recv_frame()

local bytes, err = wb:send_text("Hello world")
```

☺ full-duplex cosockets

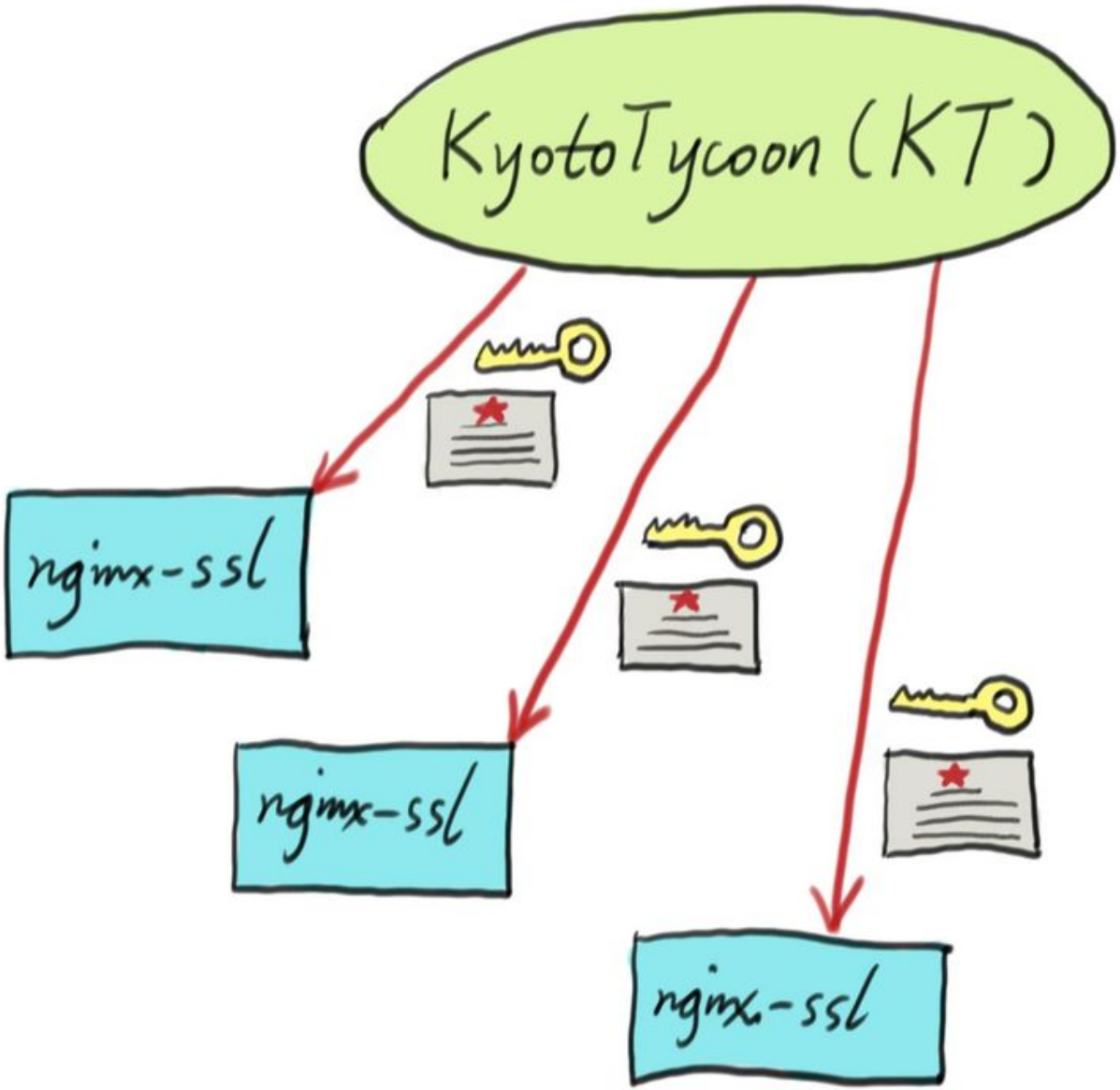
*全双工*的基于协程的套接字

☺ SSL/TLS cosocket

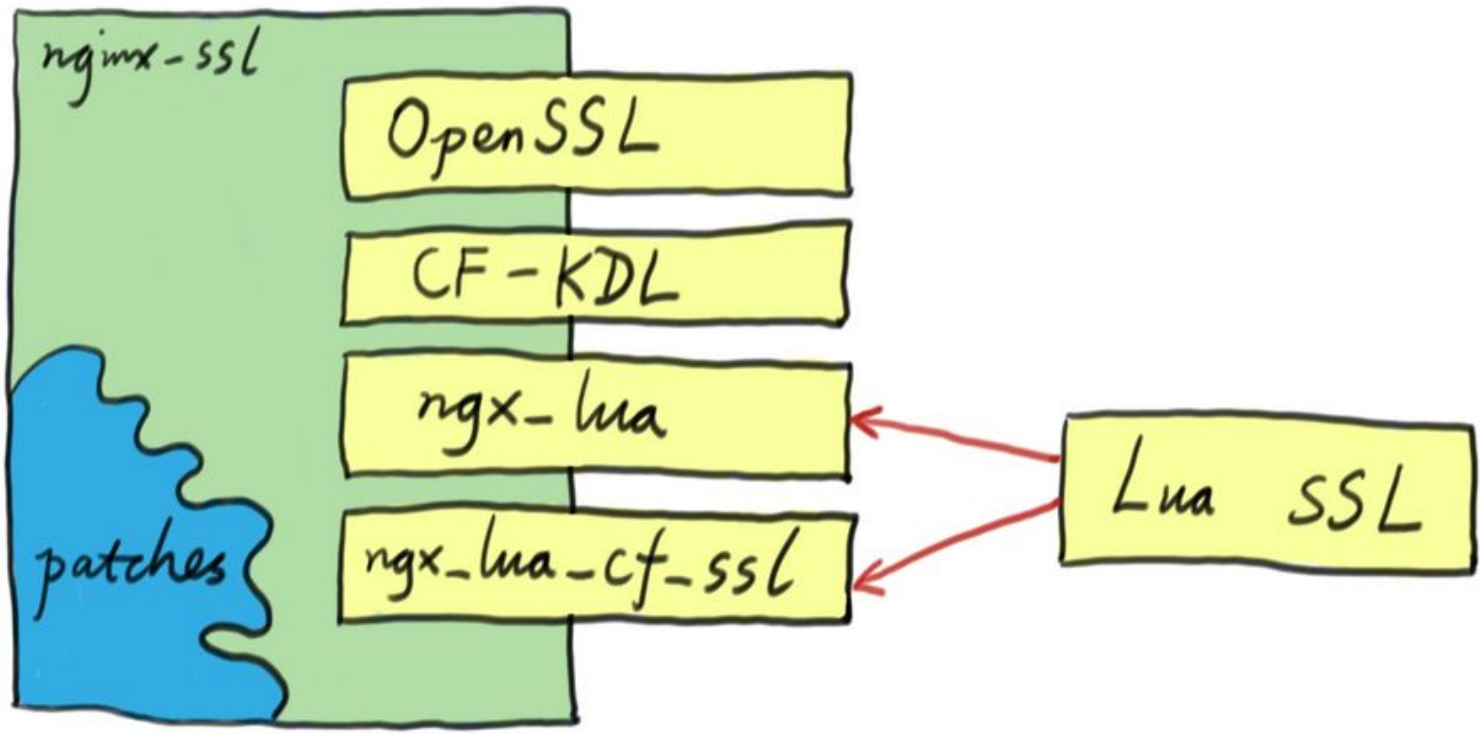
```
local sock = ngx.socket.tcp()
local ok, err = sock:connect("www.cloudflare.com",
                             443)

ok, err = sock:sslhandshake(
    false,  -- disable SSL session
    "www.cloudflare.com",  -- SNI name
    true  -- verify everything
)
```

☺ ssl_certificate_by_lua

纯 Lua 动态服务 SSL 证书和私钥


KyotoTycoon (KT)
nginx-ssl
nginx-ssl
nginx-ssl

```
ssl_certificate_by_lua '
    local ssl = require "ngx.ssl"
    local cert_chain, priv_key = my_load_data()
    assert(ssl.set_der_cert(cert_chain))
    assert(ssl.set_der_priv_key(priv_key))
';
```


nginx-ssl
OpenSSL
CF-KDL
ngx_lua
patches
ngx_lua_cf_ssl
Lua SSL

☺ ssl_balancer_by_lua

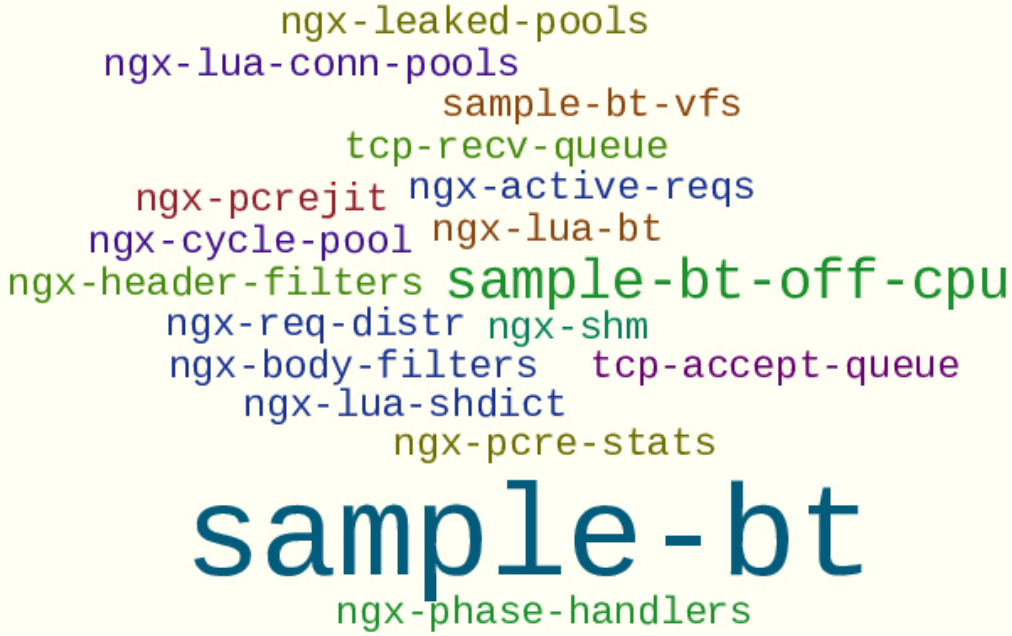
纯 Lua 编写 Nginx 全动态负载均衡器

```
upstream {
    server 0.0.0.0;
    balancer_by_lua '
        local balancer = require "ngx.balancer"
        local host, port = my_load_backend_addr()
        assert(balancer.set_cur_peer(host, port))
    ';
}
```

☺ nginx-systemtap-toolkit

基于 *Systemtap* 的 Nginx 工具箱

Click anywhere or press a button to close



☺ stapxx (stap++)

Systemtap 加上一点宏魔法……

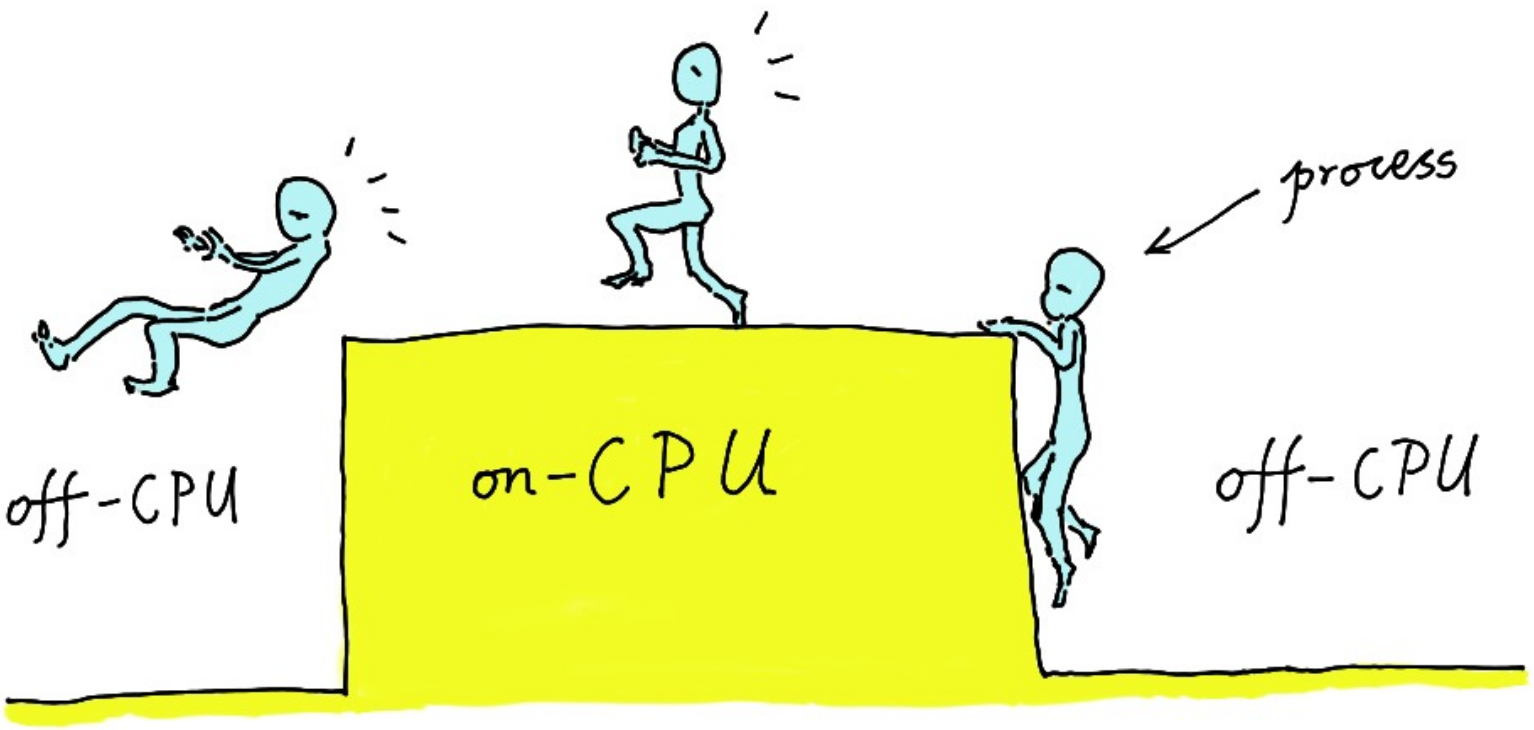
☺ How CPU time is spent inside LuaJIT

CPU 时间在 LuaJIT 内部是如何分布的

```
$ lj-vm-states.sxx -x 5521 --arg time=60
Start tracing 5521 (/usr/local/nginx-waf/sbin/nginx-waf)
Please wait for 60 seconds...

Observed 1203 Lua-running samples and ignored 49 unrelated samples.
Compiled: 64% (779 samples)
C Code (by interpreted Lua): 14% (172 samples)
Interpreted: 13% (158 samples)
Garbage Collector (compiled): 3% (48 samples)
Garbage Collector (not compiled): 3% (42 samples)
Trace exiting: 0% (4 samples)
```

☺ When an nginx worker's CPU is too high...

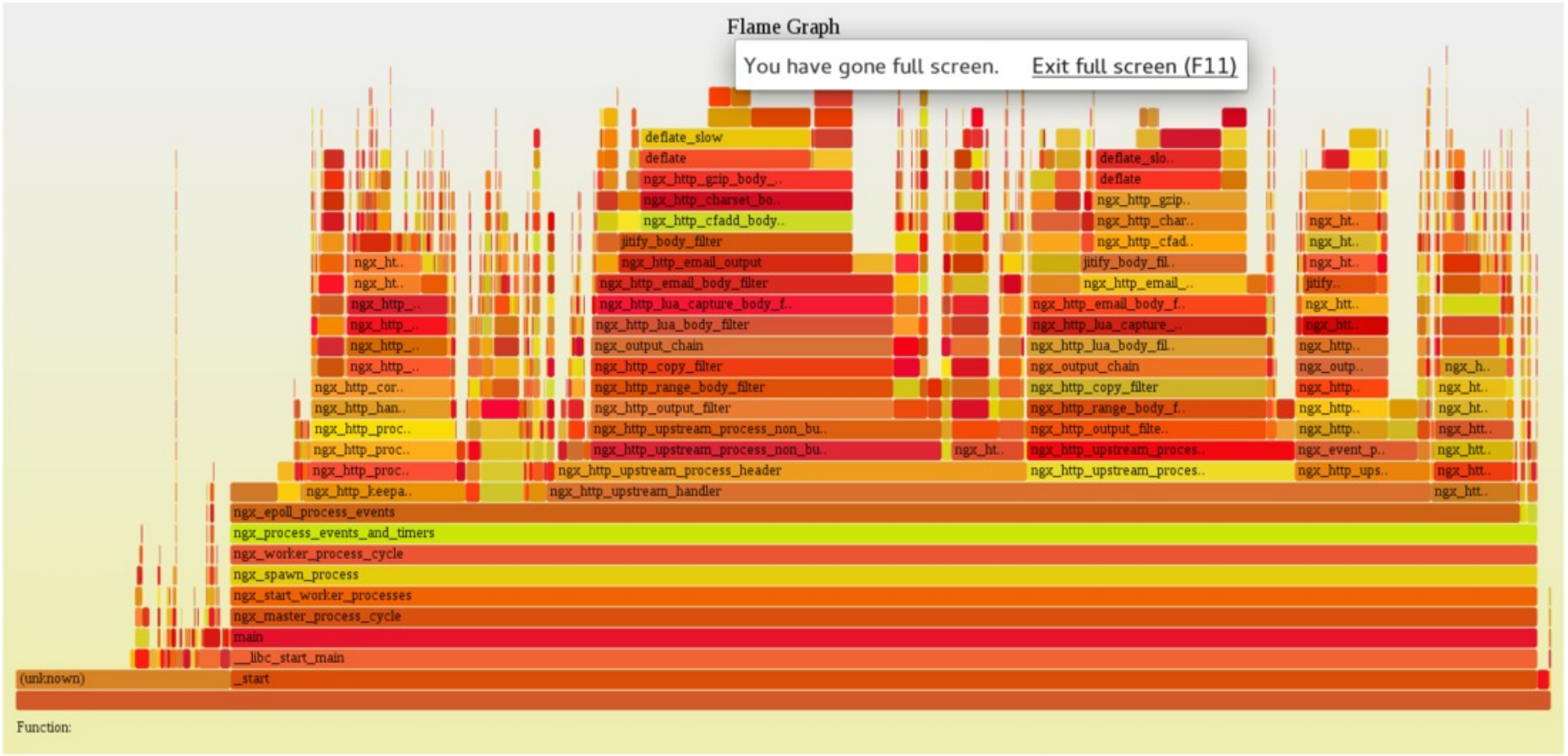
当 nginx worker 进程的 CPU 太高时……


process
off-CPU
on-CPU
off-CPU

```
# assuming one nginx worker process has the pid 19647.
$ 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
[1376939547] 238 req/sec
[1376939548] 234 req/sec
^C
```

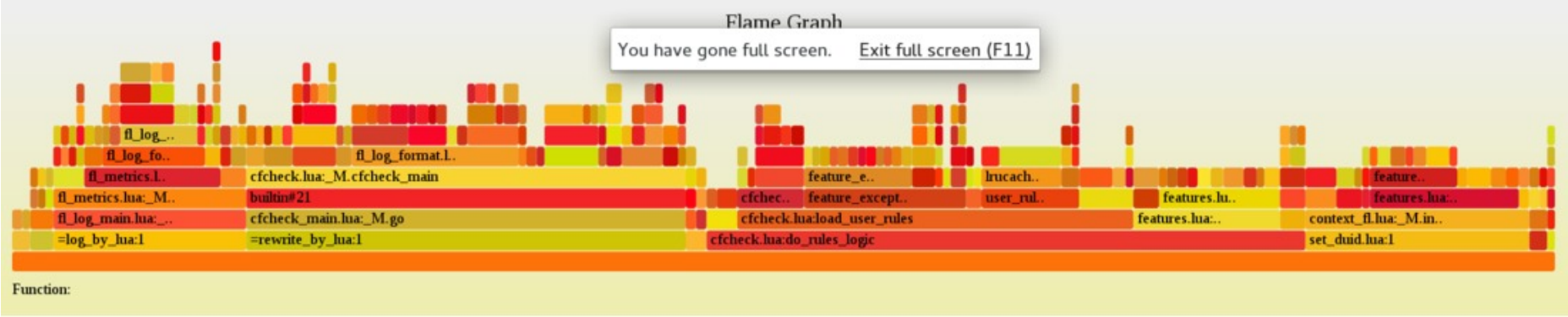
```
$ ./sample-bt -p 19647 -t 20 -u > a.bt
WARNING: Tracing 19647 (/opt/nginx/sbin/nginx) in user-space only...
WARNING: Time's up. Quitting now...(it may take a while)
```

```
# using Brendan Gregg's flame graph tools:
$ stackcollapse-stap.pl a.bt > a.cbt
$ flamegraph.pl a.cbt > a.svg
```


Flame Graph
You have gone full screen. Exit full screen (F11)
deflate_slow
deflate
ngx_http_gzip_body_..
ngx_http_charset_bo..
ngx_http_cfadd_body..
jitify_body_filter
ngx_http_email_output
ngx_http_email_body_filter
ngx_http_lua_capture_body_f..
ngx_http_lua_body_filter
ngx_output_chain
ngx_http_copy_filter
ngx_http_range_body_filter
ngx_http_output_filter
ngx_http_upstream_process_non_bu..
ngx_http_upstream_process_header
ngx_http_upstream_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
(unknown)
_start
Function:

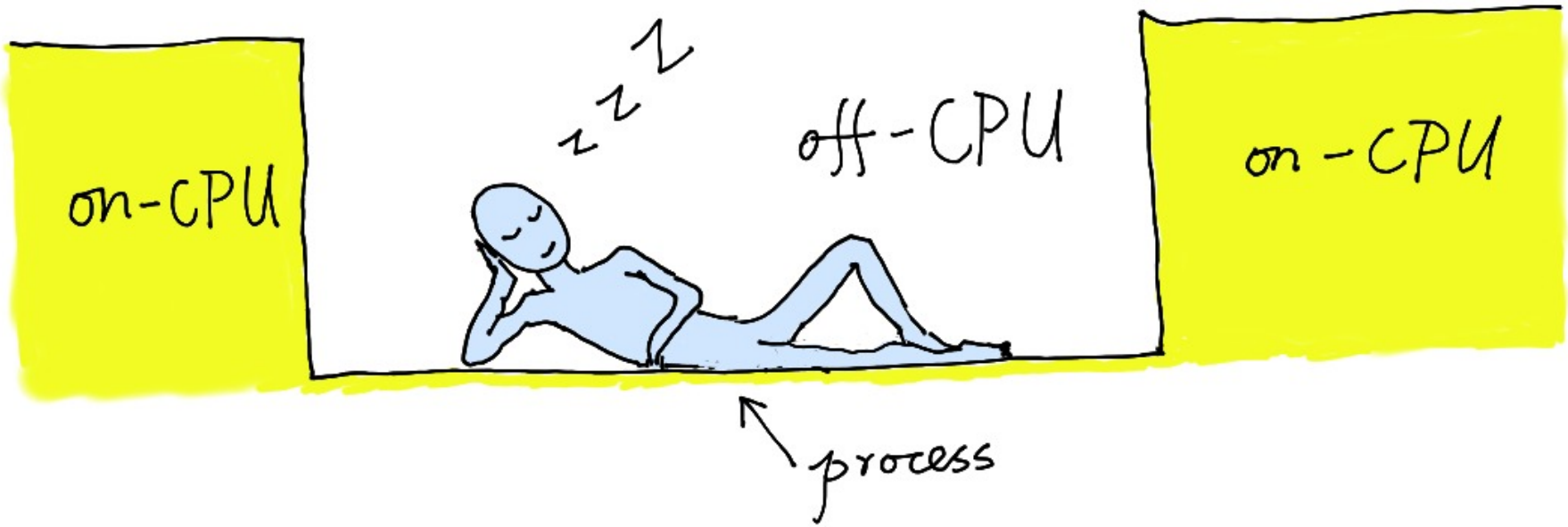
```
# assuming the nginx worker process pid is 19647:
$ lj-lua-stacks.sxx --arg time=20 --skip-badvars -x 19647 > a.bt
Start tracing 19647 (/opt/nginx/sbin/nginx)
Please wait for 20 seconds
```

```
# using Brendan Gregg's flame graph tools:
$ stackcollapse-stap.pl a.bt > a.cbt
$ flamegraph.pl a.cbt > a.svg
```

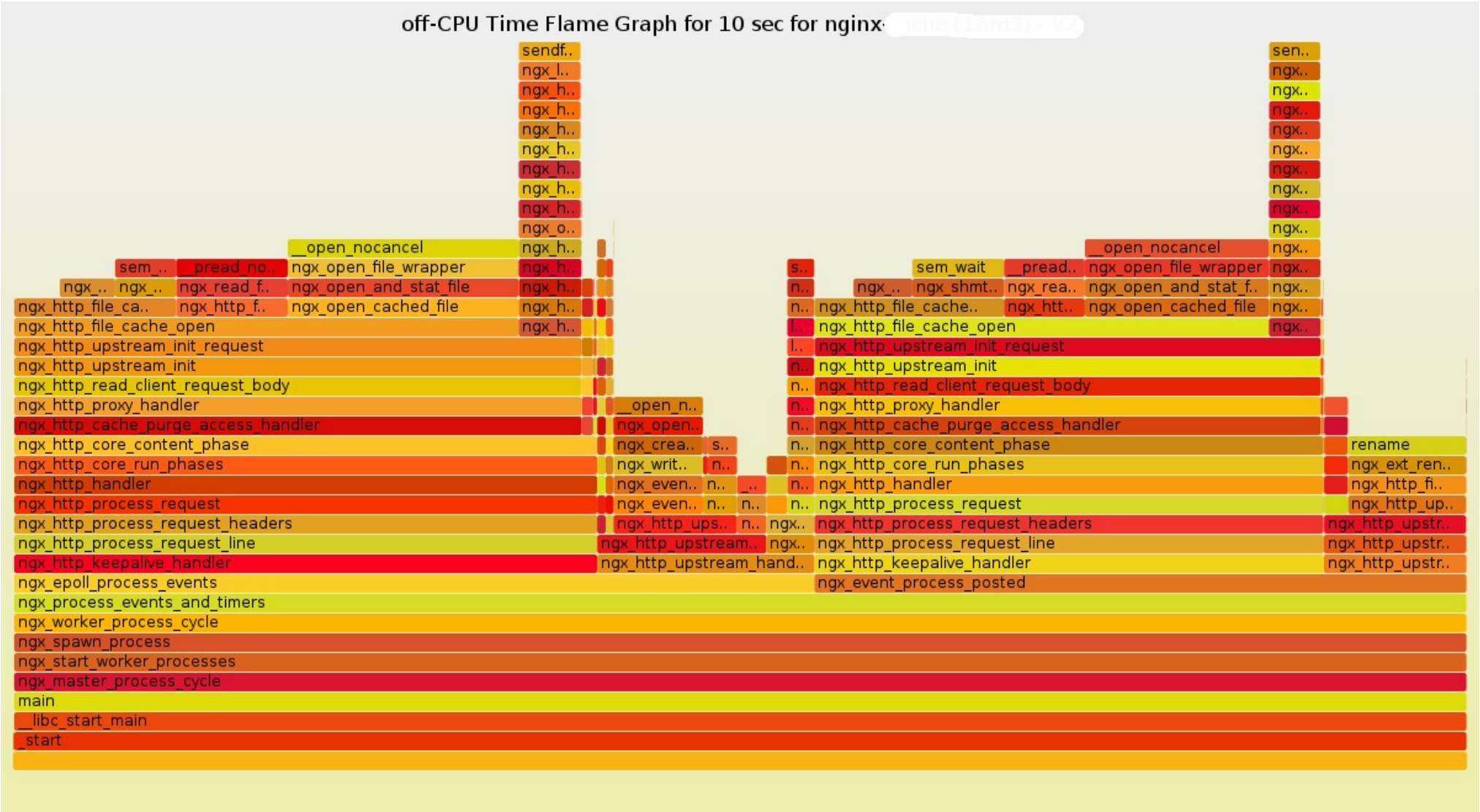

Flame Graph
You have gone full screen. Exit full screen (F11)
fl_log_..
fl_log_fo..
fl_log_format.l..
fl_metrics.l..
cfcheck.lua:_M.cfcheck_main
feature_e..
lrucach..
feature..
fl_metrics.lua:_M..
builtin#21
cfchec..
feature_except..
user_rul..
features.lu..
features.lua:..
fl_log_main.lua:_..
cfcheck_main.lua:_M.go
cfcheck.lua:load_user_rules
features.lua:..
context_fl.lua:_M.in..
=log_by_lua:1
=rewrite_by_lua:1
cfcheck.lua:do_rules_logic
set_duid.lua:1
Function:

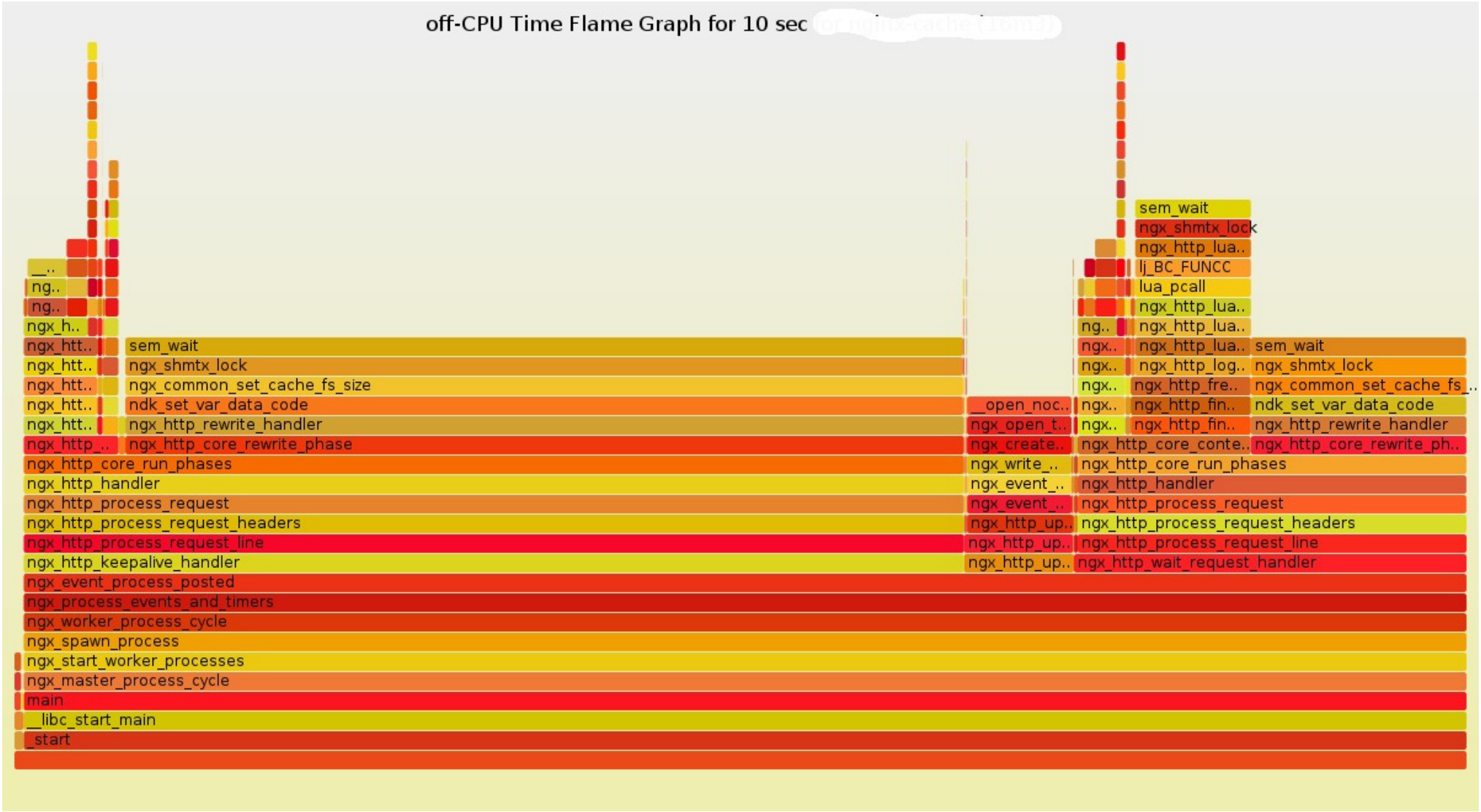
☺ When an nginx worker's CPU is low and the throughput is low...

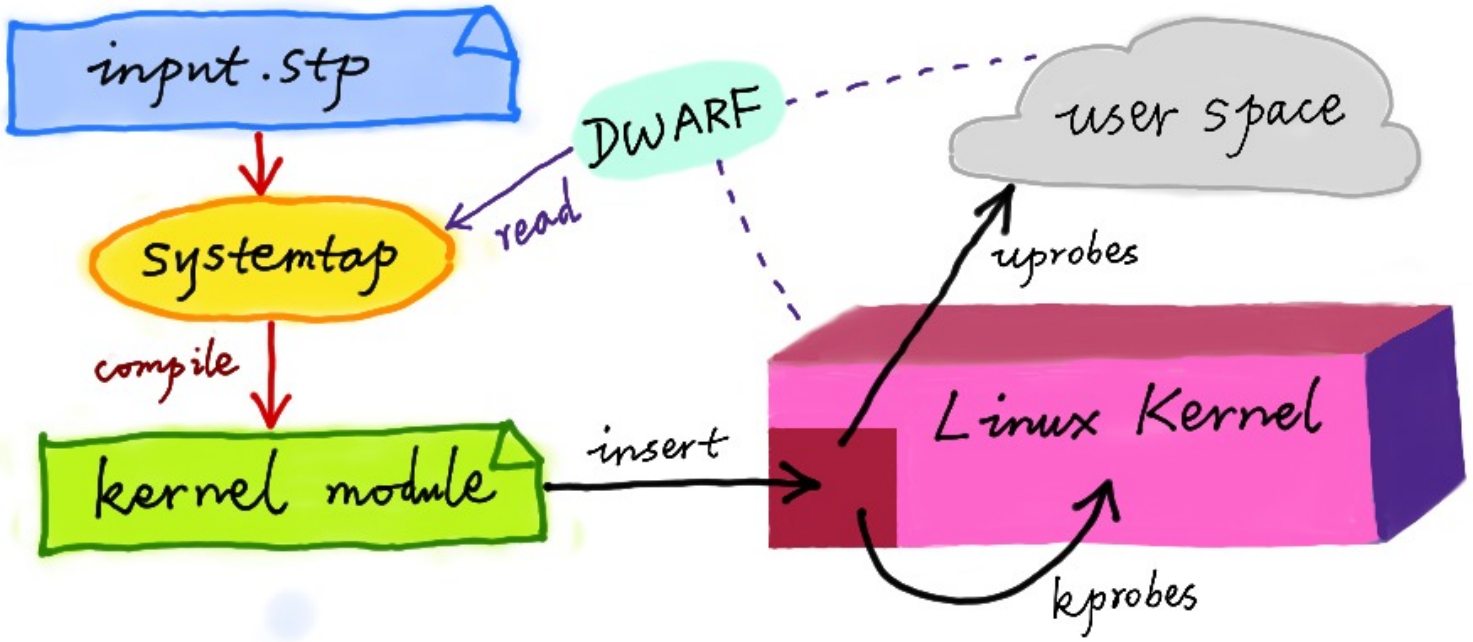
当 nginx worker 进程的 CPU 很低，同时吞吐量上不去的时候……


on-CPU
off-CPU
on-CPU
process

☺ *off-CPU* flame graphs
from the sample-bt-off-cpu tool


off-CPU Time Flame Graph for 10 sec for nginx
sendf..
ngx_l..
ngx_h..
ngx_o..
__open_nocancel
sem_..
__pread_no..
ngx_open_file_wrapper
ngx_..
ngx_read_f..
ngx_open_and_stat_file
ngx_http_file_ca..
ngx_http_f..
ngx_open_cached_file
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..
ngx..
sem_wait
__pread..
ngx_open_file_wrapper
ngx_shmt..
ngx_rea..
ngx_open_and_stat_f..
ngx_http_file_cache..
ngx_htt..
ngx_open_cached_file
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
rename
ngx_ext_ren..
ngx_http_fi..
ngx_http_up..
ngx_http_upstr..
ngx_http_upstr..
ngx_http_upstr..

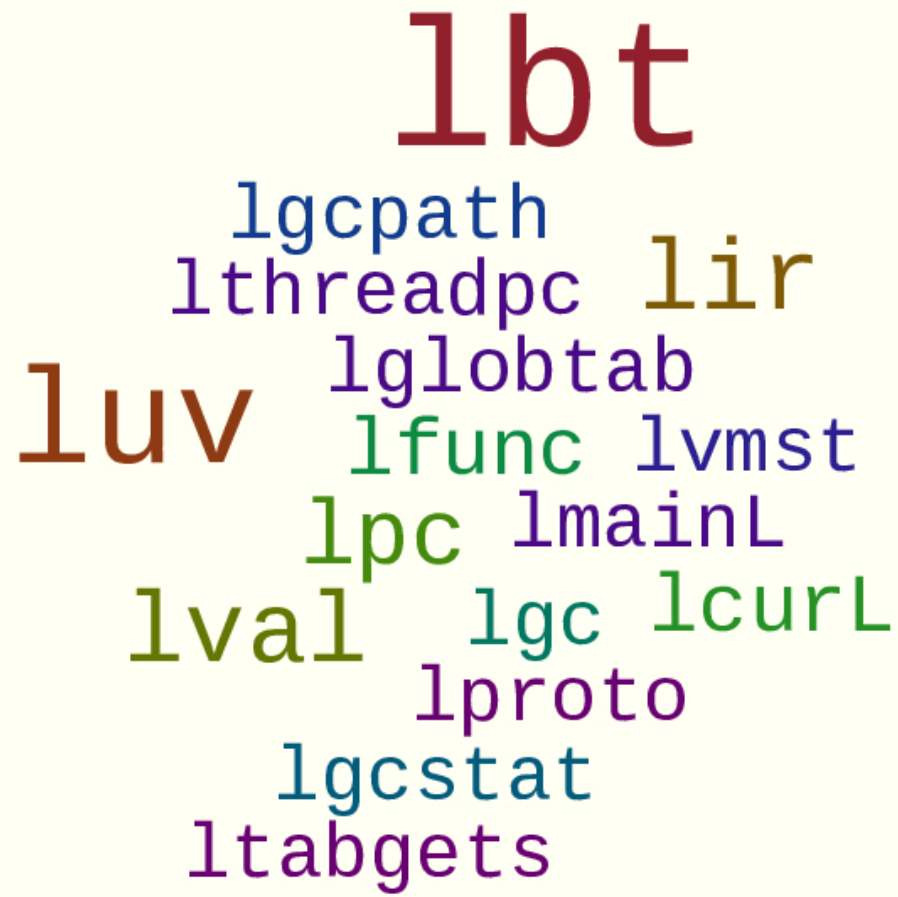

off-CPU Time Flame Graph for 10 sec
__..
ng..
ng..
ngx_h..
ngx_htt..
ngx_htt..
ngx_htt..
ngx_htt..
ngx_htt..
ngx_http_..
sem_wait
ngx_shmtx_lock
ngx_common_set_cache_fs_size
ndk_set_var_data_code
ngx_http_rewrite_handler
ngx_http_core_rewrite_phase
__open_noc..
ngx_open_t..
ngx_create..
ngx_write_..
ngx_event_..
ngx_event_..
ngx_http_up..
ngx_http_up..
ngx_http_up..
sem_wait
ngx_shmtx_lock
ngx_http_lua..
lj_BC_FUNCC
lua_pcall
ngx_http_lua..
ng..
ngx..
ngx..
ngx..
ngx..
ngx..
ngx_http_lua..
ngx_http_log..
ngx_http_fre..
ngx_http_fin..
ngx_http_fin..
ngx_http_core_conte..
sem_wait
ngx_shmtx_lock
ngx_common_set_cache_fs_..
ndk_set_var_data_code
ngx_http_rewrite_handler
ngx_http_core_rewrite_ph..
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
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


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

☺ nginx-gdb-utils

基于*GDB* 的Nginx 工具箱

Click anywhere or press a button to close



```
(gdb) lbt
C:ngx_http_lua_socket_tcp_receive
@.../lib/resty/mysql.lua:191
@.../lib/resty/mysql.lua:530
content_by_lua:10
```

```
(gdb) lbt full
C:ngx_http_lua_socket_tcp_receive
@.../lib/resty/mysql.lua:191
    local "self":
        table (0x40f181a8)
    local "sock":
        table (0x40f181b0)
@.../lib/resty/mysql.lua:530
    local "self":
        table (0x40f18148)
    local "opts":
        table (0x40f18150)
...
```

```
(gdb) lgc
The current memory size (allocated by GC): 898960 bytes
```

```
(gdb) lgcstat
15172 str         objects: max=2956, avg = 51, min=18, sum=779126
 987 upval       objects: max=24, avg = 24, min=24, sum=23688
 104 thread      objects: max=1648, avg = 1622, min=528, sum=168784
 431 proto       objects: max=226274, avg = 2234, min=78, sum=963196
 952 func        objects: max=144, avg = 30, min=20, sum=28900
 446 trace       objects: max=23400, avg = 1857, min=160, sum=828604
2965 cdata       objects: max=4112, avg = 17, min=12, sum=51576
18961 tab         objects: max=24608, avg = 207, min=32, sum=3943256
   9 udata       objects: max=176095, avg = 39313, min=32, sum=353822
```

☺ *Any questions*? ☺

欢迎提问