Magia (ou talvez não) com…

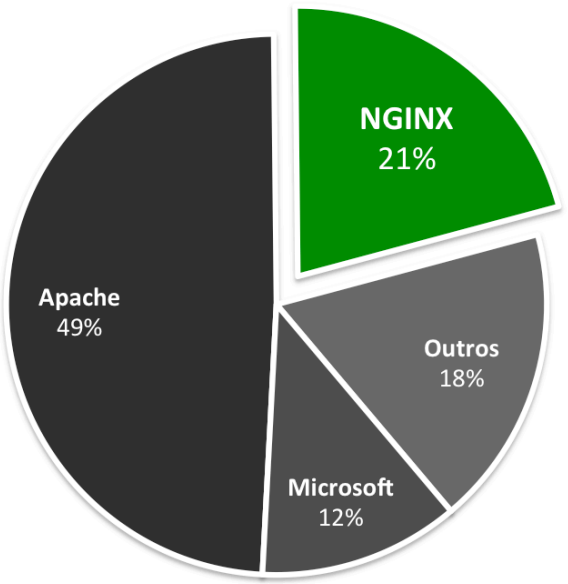
[engine-x]

+

Carlos Rodrigues
Maio de 2015

NGINX*

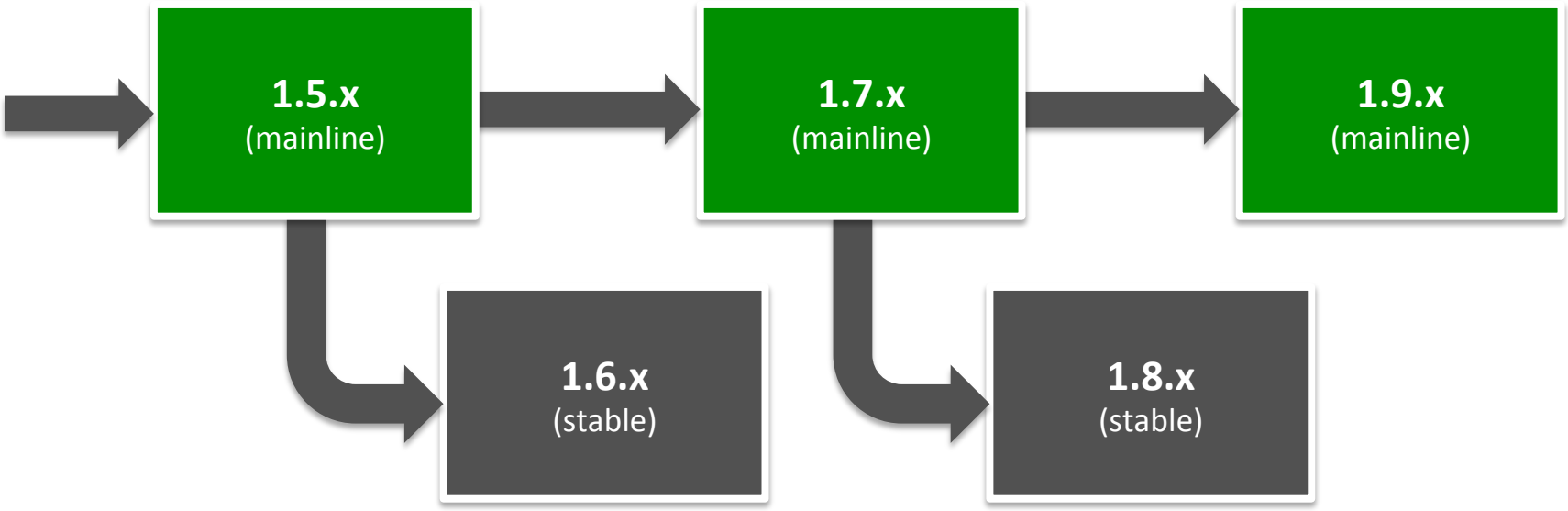
- Servidor *web* de alto desempenho
  …assíncrono/*non-blocking* (desafio C10K**)
- Open-Source (licença BSD)
  …com suporte comercial
- Corre em Linux, FreeBSD, etc.
  …em Windows não está *production-ready*
- Existe há mais de 10 anos
  …com um nível crescente de utilização



milhão de *sites* mais activos
(Netcraft, Abril de 2015)

* nginx.org
** www.kegel.com/c10k.html

Versões

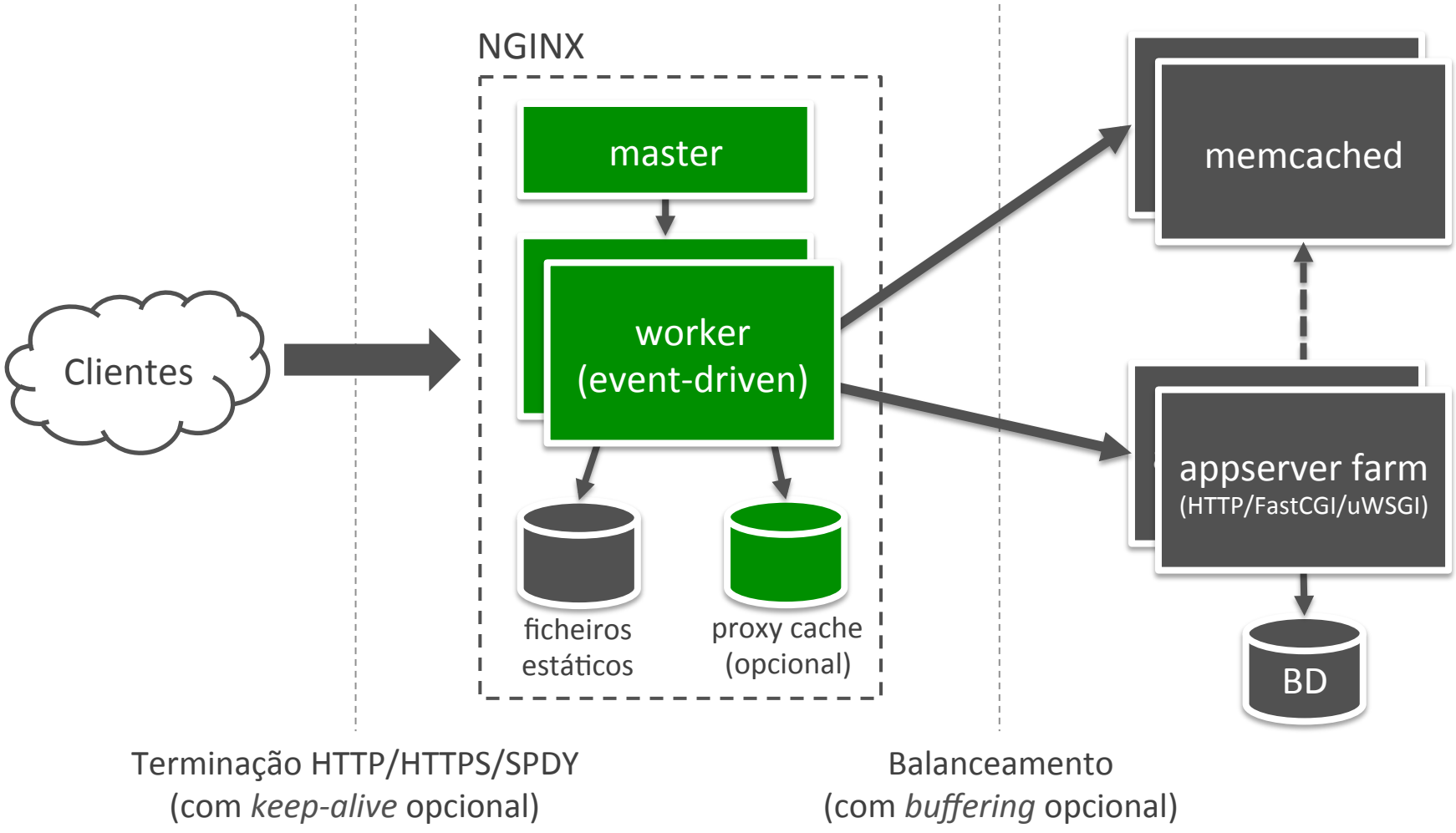


- Ambos os *branches* são usáveis em produção

  …mas a versão *stable* é mais previsível (só correcções importantes)

- Ambos têm repositórios de pacotes binários *standard* para Linux

  …que costumam ser rapidamente actualizados após cada *release*


NGINX
master
worker
(event-driven)
Clientes
ficheiros
estáticos
proxy cache
(opcional)
memcached
appserver farm
(HTTP/FastCGI/uWSGI)
BD
Terminação HTTP/HTTPS/SPDY
(com *keep-alive* opcional)
Balanceamento
(com *buffering* opcional)

Exemplo: Proxy

```
# /etc/nginx/nginx.conf

user nginx;

master_process on;
worker_processes auto;       # um por CPU
worker_rlimit_nofile 4096;   # ulimit -H -n

events {
  use epoll;
  worker_connections 1024;
}

http {
  # [...]

  server {
    listen *:80 default_server;
    server_name _;

    # [...]
  }
}
```

Exemplo: Proxy

```
# /etc/nginx/nginx.conf

user nginx;

master_process on;
worker_processes auto;       # um por CPU
worker_rlimit_nofile 4096;   # ulimit -H -n

events {
  use epoll;
  worker_connections 1024;
}

http {
  # [...] ---------------------------------------->

  server {
    listen *:80 default_server;
    server_name _;

    # [...]
  }
}
```

```
server_tokens off;  # não mostrar a versão

include /etc/nginx/mime.types;
default_type application/octet-stream;
charset utf-8;

log_format main '$remote_addr - [...]';
access_log /var/log/nginx/access.log main;

keepalive_requests 100;
keepalive_timeout 60;

tcp_nopush on;
tcp_nodelay on;
sendfile on;

proxy_buffering on;  # default
```

Exemplo: Proxy

```
# /etc/nginx/nginx.conf

user nginx;

master_process on;
worker_processes auto;       # um por CPU
worker_rlimit_nofile 4096;   # ulimit -H -n

events {
  use epoll;
  worker_connections 1024;
}

http {
  # [...] ---------------------------------------->

  server {
    listen *:80 default_server;
    server_name _;

    # [...] ---------------------------------------->
  }
}
```

```
server_tokens off;  # não mostrar a versão

include /etc/nginx/mime.types;
default_type application/octet-stream;
charset utf-8;

log_format main '$remote_addr - [...]';
access_log /var/log/nginx/access.log main;

keepalive_requests 100;
keepalive_timeout 60;

tcp_nopush on;
tcp_nodelay on;
sendfile on;

proxy_buffering on;  # default
```

```
location / {
  proxy_pass http://127.0.0.1:8080;
  proxy_redirect default;
}
```

Exemplo: Memcache

```
# /etc/nginx/nginx.conf

events {};

http {
  upstream appserver_farm {
    server 10.0.0.1:8080;
    server 10.0.0.2:8080;
    least_conn;  # tipo de balanceamento
  }

  include /etc/nginx/mime.types;
  default_type application/octet-stream;
  charset utf-8;

  server {
    listen *:80;
    server_name www.example.com example.com;

    # [...]
  }

  server { [...] }  # default_server
}
```

Exemplo: Memcache

```
# /etc/nginx/nginx.conf

events {};

http {
  upstream appserver_farm {
    server 10.0.0.1:8080;
    server 10.0.0.2:8080;
    least_conn;  # tipo de balanceamento
  }

  include /etc/nginx/mime.types;
  default_type application/octet-stream;
  charset utf-8;

  server {
    listen *:80;
    server_name www.example.com example.com;

    # [...] ----------------------------------------->
  }

  server { [...] }  # default_server
}
```

```
if ($request_method !~ "^(GET|HEAD)$") {
  return 405;  # method not allowed
}

location / {
  default_type text/html;

  expires 5m;
  add_header X-Cache "HIT";

  # tentar na memcache primeiro (body)
  set $memcached_key "$uri$is_args$args";
  memcached_pass 10.0.0.3:11211;
  error_page 404 502 504 = @backends;
}

location @backends {
  proxy_pass http://appserver_farm;
  proxy_redirect default;
}
```

Entretanto…


MONGODB
MICROSOFT
TOUCH
STUDENT
LANGUAGES
SMARTALKS
MANAGEMENT
CHAPTER
UBUNTU
ACM
DEVELOPMENT

Lua

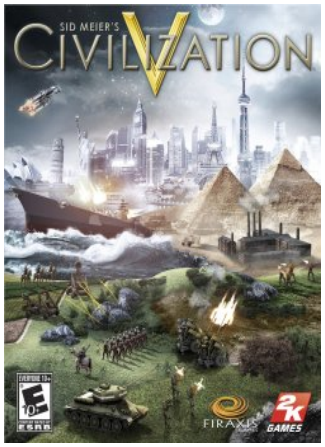
- Sintaxe simples e fácil de aprender
  - …mas flexível para usos avançados
  - …orientada para *scripting* embebido
- Implementação minimalista, rápida e universal
  - …menos de 200 KB para a totalidade do *runtime standard**
  - …com o LuaJIT** como alternativa ainda mais rápida
- Open-Source (licença MIT)
  - …sem restrições à integração em *software* proprietário
- Mais de 20 anos de existência
  - …com uma grande (mas pouco visível) base de utilização —

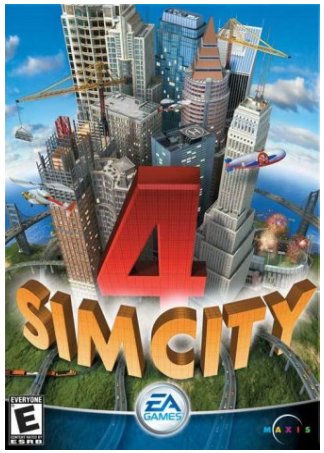
* www.lua.org

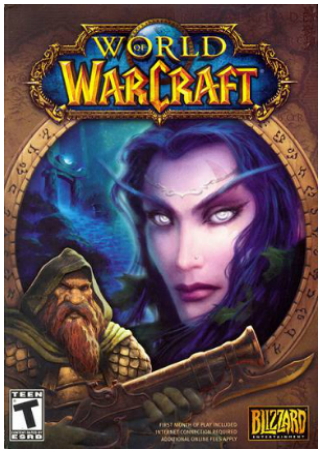
** luajit.org

— principalmente na área dos jogos…

*

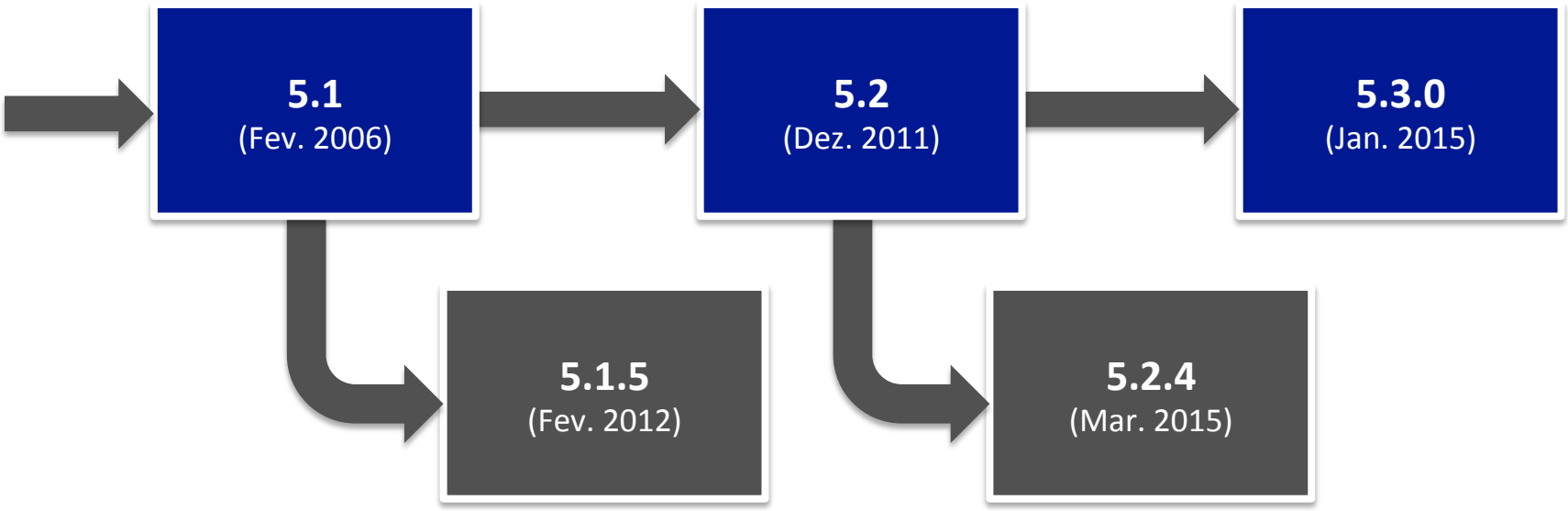






* www.grimfandango.net/features/articles/lua-in-grim-fandango

Versões



- Desenvolvimento contínuo de novas funcionalidades

  …as versões mais antigas só recebem correcções

- A versão 5.1 continua a ser a mais popular

  …porque é a especificação implementada pelo LuaJIT

Hello, World!

```lua
print("Hello, World!”)
print(1 == 1, 1 ~= 1, x)                          -- true false nil


if "" and 0 and {} then print("ok") end           -- só nil e false são false
if not (nil == false) then print("ok") end        -- mas nil ≠ false


--[[
  A tabela é o único tipo estruturado nativo em Lua e serve de base a
  tudo o resto, incluindo algumas funcionalidades da própria linguagem.
--]]

local a = {10, 20, 30}                            -- tabela como array simples
for index, value in pairs(a) do                   -- índice “index” começa em 1
    print(index .. " -> " .. value)
end

local t = {["key one"]=1, key_two=2, [3]=3}       -- tabela como array associativo
for key, value in pairs(t) do                     -- índice “key” sem ordem definida
    print(string.format("%s -> %d", key, value))
end
```

lua-users.org/wiki/LuaTutorial

Funções

```lua
function fact1(n)
    if n <= 1 then
        return 1
    else
        return n * fact1(n - 1)
    end
end

local fact2 = function(n)    -- as variáveis (e funções) são globais por omissão
    local f = 1              -- mas o acesso a variáveis locais é mais eficiente

    while n > 1 do
        f = f * n
        n = n - 1
    end

    return f
end

for i = 1, 10, 1 do
    print(string.format("%d! = %d == %d", i, fact1(i), fact2(i)))
end
```

lua-users.org/wiki/LuaTutorial

Módulos

```lua
-- module.lua

local function add(n, i)     -- função local ao módulo
    return n + i
end

return {                     -- tabela de exports do módulo
    add = add
}

-- test_module.lua

local m = require("module")

m.add(10, 1)
```

lua-users.org/wiki/LuaTutorial

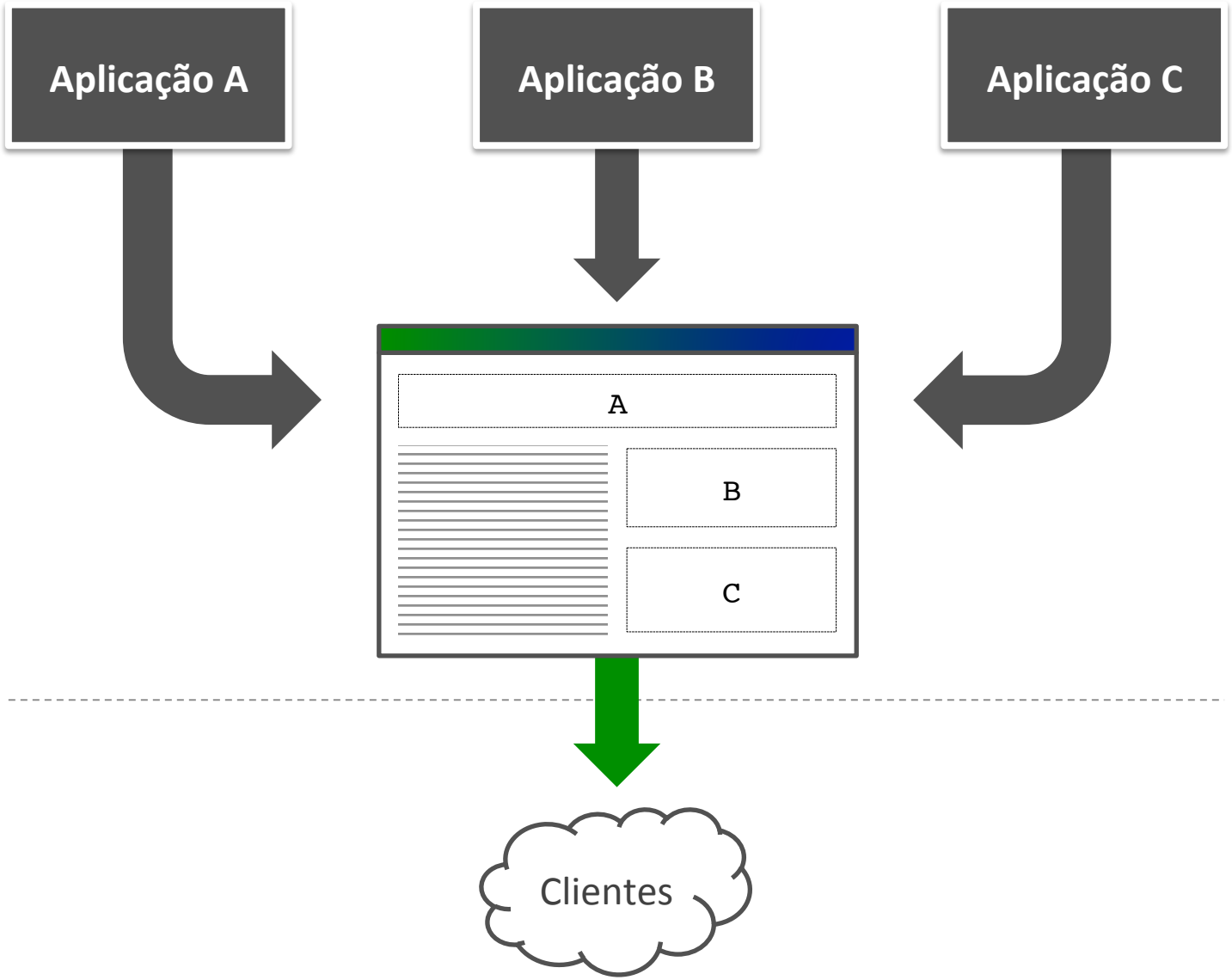
Objectos

```lua
-- class.lua

local Class = {}                    -- representação da classe e metatable das instâncias
Class.__index = Class               -- fallback do lookup de atributos (métodos) na classe

function Class.new(n)
    local self = setmetatable({}, Class)   -- a metatable adiciona "magia" à tabela que
    self.n = n; return self                -- pode incluir, p.ex. operator overloading
end

function Class.add(self, i)
    self.n = self.n + i; return self.n
end

return Class

-- test_class.lua

local class = require("class")
local instance = class.new(10)

instance:add(1)                     -- sintaxe equivalente a "instance.add(instance, 1)"
```

lua-users.org/wiki/LuaTutorial

$$1 + 1 = 2$$


MONGODB
MICROSOFT
STUDENT
SCTE
SMARTALKS
CHAPTER
UBUNTU
ACM
DEVELOPMENT

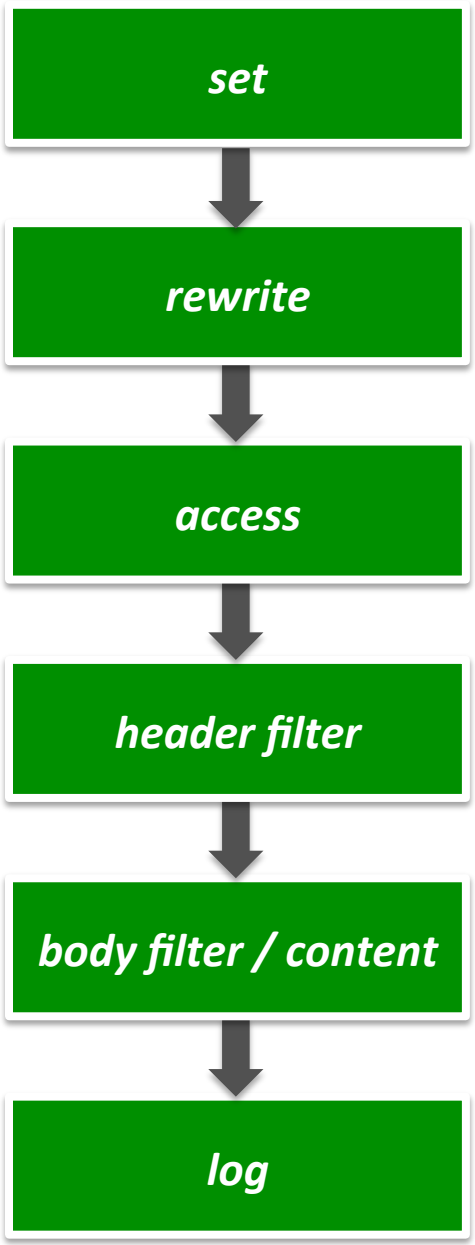

Aplicação A
Aplicação B
Aplicação C
A
B
C
Clientes

NGINX + Lua

- Simplifica **muito** as configurações avançadas
  …quase eliminando a necessidade de módulos extra
- Permite trazer lógica para a camada de entrega
  …partes que devem estar a cargo dos *devops*
  …ou para maximizar o desempenho do serviço
- Fiável e com impacto mínimo nos tempos de resposta
  …o projecto arrancou no Taobao (grupo Alibaba)
  …usado intensivamente pela CloudFlare (actual patrocinador)
- OpenResty Bundle* ou "Do-It-Yourself"**

* openresty.org

** github.com/openresty/lua-nginx-module



Funcionalidades

- Encaixa nas várias fases do processamento
  - …com partilha de contexto *per-request* entre elas
  - …controlo do fluxo e conteúdo final das respostas
- API interna exposta em Lua*
  - …com *regexes* compiladas, *timers*, *encoding*, etc.
  - …com *sub-requests*, *co-sockets* e *shared dictionaries*
  - …quase tudo *main-loop friendly* (não-bloqueante)
- Suficiente para uma *framework*** completa

* wiki.nginx.org/HttpLuaModule

** leafo.net/lapis

Exemplo: Routing

```
# /etc/nginx/nginx.conf

user nginx;
events {}

http {
  server {
    listen *:80;
    server_name _;

    location / {
      set $backend "http://10.0.0.1";

      rewrite_by_lua '[...]';

      # com variáveis requer URL completo:
      proxy_pass "$backend$uri$is_args$args";
    }
  }
}
```

Exemplo: Routing

```
# /etc/nginx/nginx.conf

user nginx;
events {}

http {
  server {
    listen *:80;
    server_name _;

    location / {
      set $backend "http://10.0.0.1";

      rewrite_by_lua '[...]';

      # com variáveis requer URL completo:
      proxy_pass "$backend$uri$is_args$args";
    }
  }
}
```

```
-- fase de rewrite

local uri = ngx.var.uri
local regex = "^/user/([0-9]+)$";

local m = ngx.re.match(uri, regex, "o")

if m then
  ngx.req.set_uri("/profile")
  ngx.req.set_uri_args("id=" .. m[1])

  ngx.var.backend = "https://10.0.0.2"
end
```

Exemplo: Throttling

```
# /etc/nginx/nginx.conf

user nginx;
events {}

http {
  lua_shared_dict counters 1m;

  server {
    listen *:80;
    server_name _;

    location / {
      default_type text/plain;

      access_by_lua '[...]';
      header_filter_by_lua '[...]';
      content_by_lua '[...]';
    }
  }
}
```

Exemplo: Throttling

```
# /etc/nginx/nginx.conf

user nginx;
events {}

http {
  lua_shared_dict counters 1m;

  server {
    listen *:80;
    server_name _;

    location / {
      default_type text/plain;

      access_by_lua '[...]';
      header_filter_by_lua '[...]';
      content_by_lua '[...]';
    }
  }
}
```

```
-- fase de controlo de acesso

local counters = ngx.shared.counters
local client = ngx.var.remote_addr

counters:add(client, 0, 20)  -- expira em 20s
local hits = counters:incr(client, 1)

if hits > 10 then
  return ngx.exit(429)
end

ngx.ctx["hits"] = hits  -- contexto do pedido
```

Exemplo: Throttling

```
# /etc/nginx/nginx.conf

user nginx;
events {}

http {
  lua_shared_dict counters 1m;

  server {
    listen *:80;
    server_name _;

    location / {
      default_type text/plain;

      access_by_lua '[...]';
      header_filter_by_lua '[...]';
      content_by_lua '[...]';
    }
  }
}
```

```
-- fase de controlo de acesso

local counters = ngx.shared.counters
local client = ngx.var.remote_addr

counters:add(client, 0, 20)  -- expira em 20s
local hits = counters:incr(client, 1)

if hits > 10 then
  return ngx.exit(429)
end

ngx.ctx["hits"] = hits  -- contexto do pedido
```

```
-- fase de manipulação de headers

ngx.header["X-Hits"] = ngx.ctx["hits"]
```

Exemplo: Throttling

```
# /etc/nginx/nginx.conf

user nginx;
events {}

http {
  lua_shared_dict counters 1m;

  server {
    listen *:80;
    server_name _;

    location / {
      default_type text/plain;

      access_by_lua '[...]';
      header_filter_by_lua '[...]';
      content_by_lua '[...]';
    }
  }
}
```

```
-- fase de controlo de acesso

local counters = ngx.shared.counters
local client = ngx.var.remote_addr

counters:add(client, 0, 20)  -- expira em 20s
local hits = counters:incr(client, 1)

if hits > 10 then
  return ngx.exit(429)
end

ngx.ctx["hits"] = hits  -- contexto do pedido
```

```
-- fase de manipulação de headers

ngx.header["X-Hits"] = ngx.ctx["hits"]
```

```
-- fase de output de conteúdo

ngx.say(ngx.localtime())
```


você está aqui
you are here

Obrigado!

Questões?

Carlos Rodrigues

cefrodrigues@gmail.com
twitter.com/carlosefr

Slides: speakerdeck.com/carlosefr