Почему язык Lua — это интересно?

Ник Заварицкий
mejedi@tarantool.org

http://www.devconf.ru

Индекс TIOBE (2016)

1. Java
2. C
3. C++
4. Python
5. C#
6. PHP
7. JavaScript

31. Lua

Кто использует Lua?

OpenResty

Lua

HTML

NGINX

WebSockets

snabb.co

torch

Terra

Кто еще использует Lua?

TARANTOOL

Luvit <3

redis

Далее...

- Знакомство с Lua
- Lua WAT
- Почему в Tarantool LuaJIT, а не V8
- Как устроен трассирующий JIT-компилятор

Lua: Пример

```lua
print("Hello, world!")
```

```lua
local function left_pad(str, len, ch)
    return string.rep(ch or " ", len - #str) .. str
end
```

Lua: Типы данных

nil		
boolean	`true`	`false`
number	`42`	`3.14159265`
string	`“Hello, world!”`	
table	`{ 1,2,3 }` `{ name=“John”, last=“Doe” }`	
function		
thread, userdata, cdata		

Lua: if / then / else

```
local function say(animal)
    if animal == "Duck" then
        print("Quack")
    elseif animal == "Cat" then
        print("Meow")
    else
        print("♫")
    end
end
```

Lua: for

```lua
local sum = 0
for i = 1,100 do
    sum = sum + i
end


local t = { language="Lua", notes="It rocks!" }
for k,v in pairs(t) do
    print(k, "=", v)
end
```

Lua: Функции (1)

```lua
local function max(a, b)
    return a > b and a or b
end


local first, last = string.find("Dark room", "a cat")
```

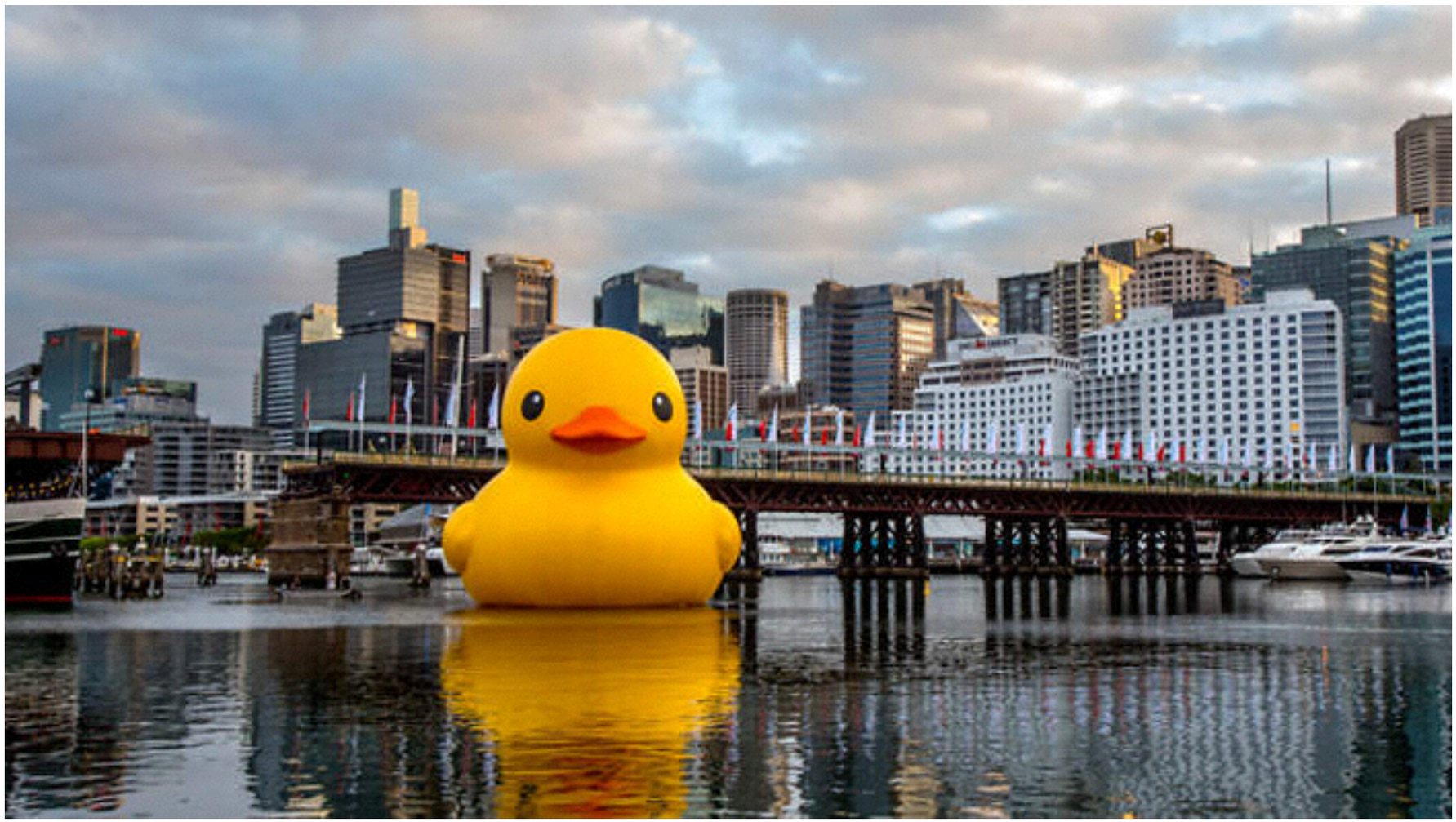
Lua: Функции (2)

```
local function wrap(fn)
    return function(...)
        print("Calling ", fn)
        return fn(...)
    end
end
```

Lua: Осталось за кадром...

`while`	`repeat`
`local module = require("module")`	
мета-таблицы	
GC	
`error()`	`pcall()`
`loadstring()`	`string.dump()`
`coroutine.create()`	`coroutine.yield()`

WAT



WAT

```
#{ 1, 2, nil, 4 } = ?
```

```
0 or 1 = ?
```

```
"" or "apple" = ?
```

```
1 .. "bar" = ?
```

WAT

```
local function chunky_bacon()
    return "chunky", "bacon"
end

{ chunky_bacon() } = ?

{ (chunky_bacon()) } = ?

{ "OMNOMNOM", chunky_bacon() } = ?

{ chunky_bacon(), "Tastes good!" } = ?
```

WAT

```
local i
for i = 1,100 do
    -- nothing
end
i = ?
```

```
local function factorial(n)
    return n > 1 and n * factorial(n-1) or 1
end
factorial(10) = ?
```

Почему в Tarantool LuaJIT, а не V8?

- Проще встраивается
- FFI
- Работает быстрее*
- 78,098 LOC vs. 2,155,836 LOC

LuaJIT FFI

```
local ffi = require("ffi")

ffi.cdef[[
unsigned char *SHA512(
        const unsigned char *d, size_t n,
        unsigned char *md);]])

local crypto = ffi.load("libcrypto.so")

local function sha512(s)
    local md = ffi.new("char[?]", 64)
    crypto.SHA512(s, #s, md)
    return ffi.string(md, 64)
end
```

Как устроен трассирующий JIT-компилятор

- Компилируются «горячие куски кода» (правило 80% / 20%)
- Full-method JIT vs. tracing JIT

```
local function sum_positive(sum, val)
    if val > 0 then
        return sum + val
    else
        return sum
    end
end
```

```
local function reduce(array, func, initial)
    local res = initial
    for i = 1,#array do
        res = func(res, array[i], i, array)
    end
    return res
end

reduce({ ... }, sum_positive, 0)
```

```
local function sum_positive(sum, val)
    if val > 0 then
        return sum + val
    else
        return sum
    end
end

local function reduce(array, func, initial)
    local res = initial
    for i = 1,#array do
        res = func(res, array[i], i, array)
    end
    return res
end

reduce({ ... }, sum_positive)
```

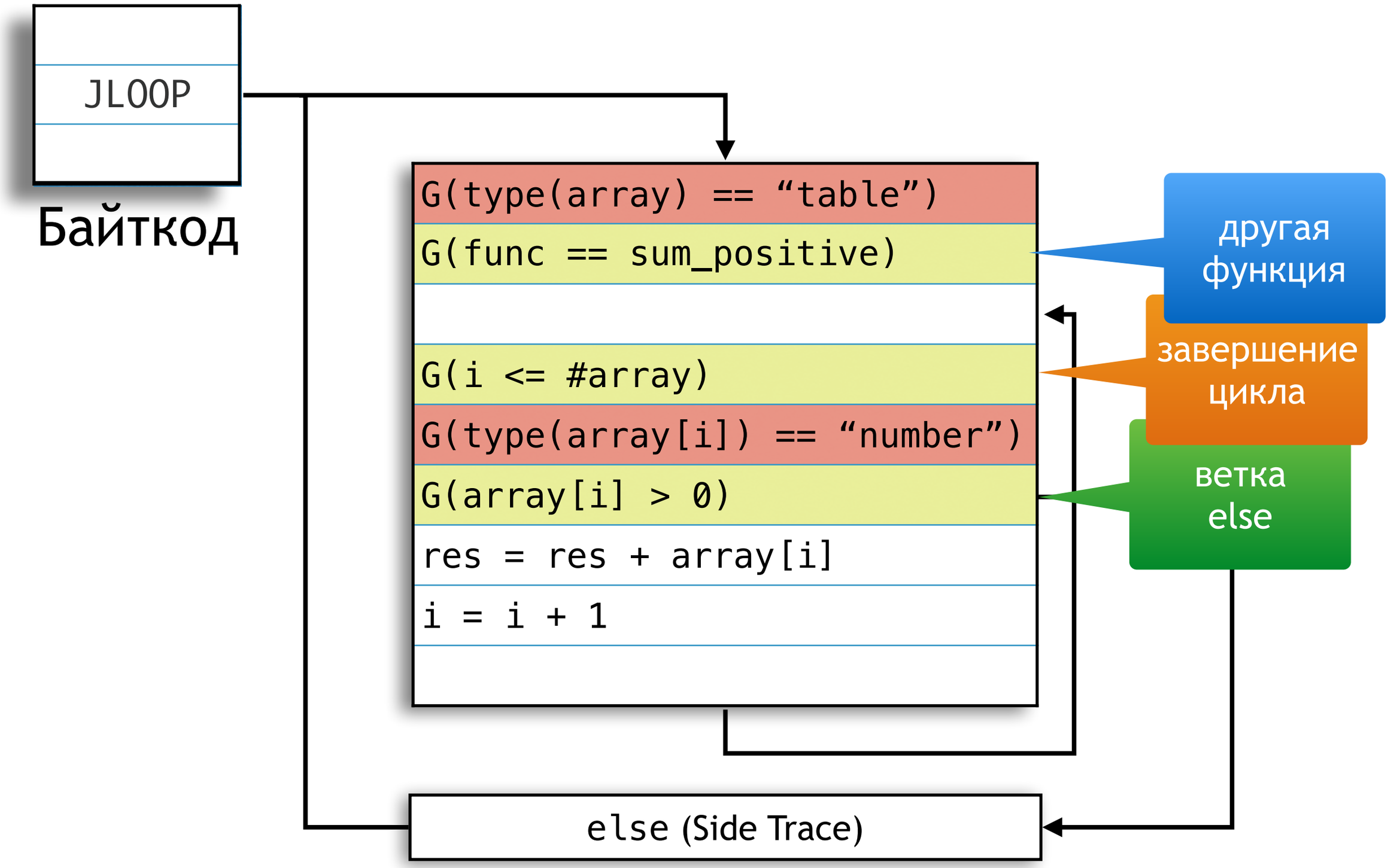
③ ④ ① ②

```
local function sum_positive(sum, val)
③       if val > 0 then
④           return sum + val
        else
            return sum
        end
    end

    local function reduce(arra
        local res = initial
①       for i = 1,#array do
②           res = func(res, a
        end
        return res
    end

    reduce({ ... }, sum_positive)
```

G(type(array) == “table”)
G(func == sum_positive)
G(i <= #array) ①
G(type(array[i]) == “number”)
G(array[i] > 0) ③
res = res + array[i] ②④
i = i + 1 ①


JLOOP
Байткод
G(type(array) == “table”)
G(func == sum_positive)
G(i <= #array)
G(type(array[i]) == “number”)
G(array[i] > 0)
res = res + array[i]
i = i + 1
другая функция
завершение цикла
ветка else
else (Side Trace)

Оптимизации

- DCE, CSE, FOLD, ABC, SINK…

- `string.format("%04d/%02d/%02d", yy, mm, dd)`

- `sql(“SELECT name FROM users WHERE ID=%1", id)`

Заключение

- Lua — удобный высокоуровневый язык.
- У нас есть приложения >10000 LOC на Lua.
- LuaJIT очень быстрый.
- Легко встроить.
- При желании, можно влезть внутрь и понять, как это работает.

OpenResty

Lua

HTML

WebSockets

NGINX

torch

TARANTOOL

LuaJIT

Luvit

Terra

Ссылки

http://lua.org

http://luajit.org

http://tarantool.org

https://openresty.org

http://torch.ch

https://luvit.io

https://snabb.co

http://terralang.org