Lua and its Ecosystem

(45')

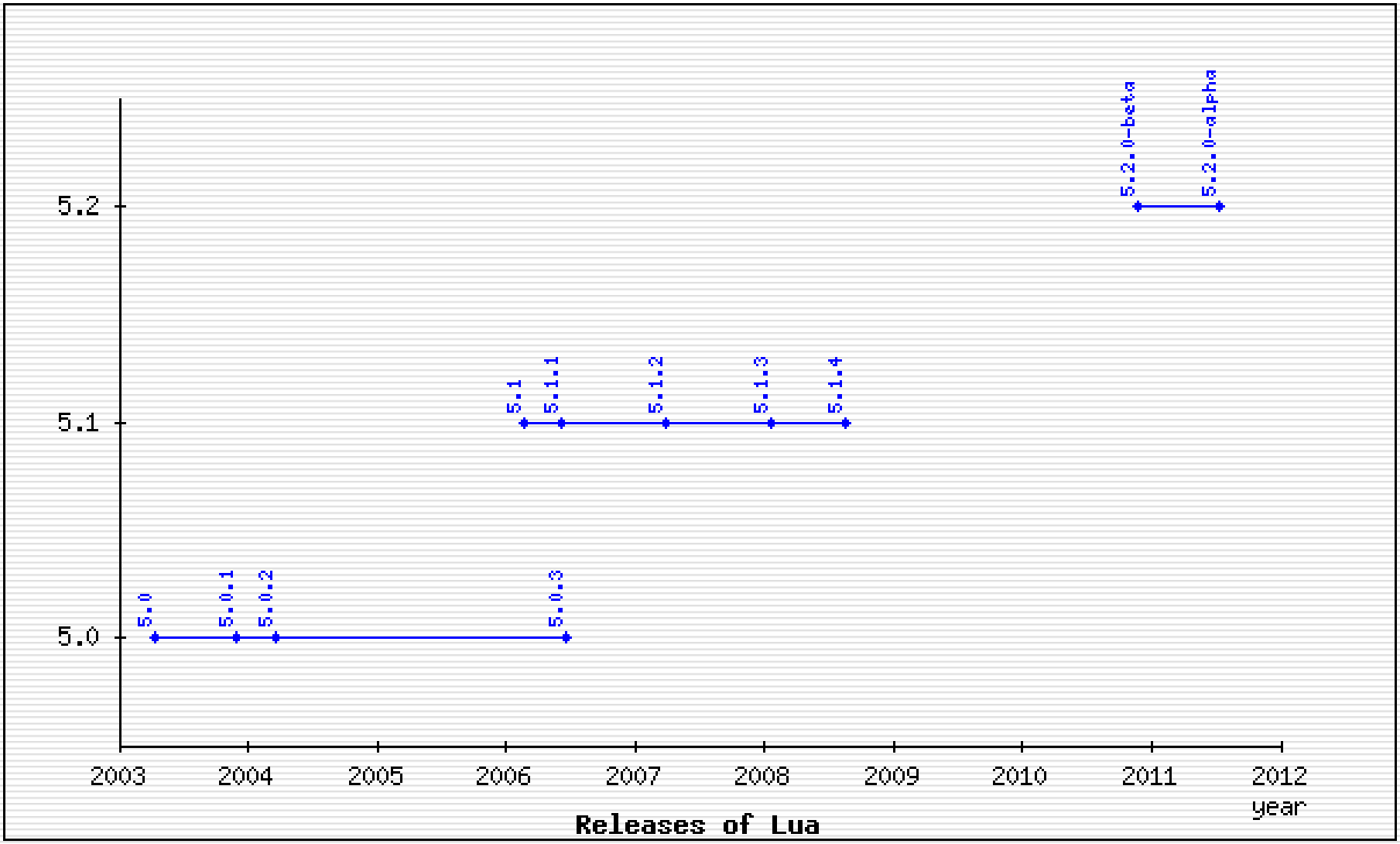
François Perrad

francois.perrad@gadz.org

Overview

- Lua is a powerful, fast, lightweight, embeddable scripting language.
- from Brazil (PUC-Rio) since 1993
- Open Source, but closed development
- A core team (academic)
  - Roberto Ierusalimschy
  - Luiz Henrique de Figueiredo
  - Waldemar Celes
- MIT License
- A mailing list since 1997
- lastest Workshop on September 8–9, 2011

History of recent releases



Lua is small

- Reference Manual
  - < 100 pages
- Grammar EBNF
  - 1 page
- Code size : Sloccount -> 12.5 KLoC of C
- Binary ~150 kB
- 8 types
  - nil, boolean, number, string, function, userdata, thread, table
- 21 keywords
  - No builtin function,
  - only standard libraries

Lua is portable

- Only C89
- Linux, OSX, Windows
- Android, iOS
  - Corona SDK
- eLua runs on the "bare-metal"
  - LEGO Mindstorm NXT
  - ...

Alternate Implementations

- LuaJIT
  - Very fast, full compatible
- LLVM-Lua
- Mochalua
  - Java JME
- Jill
  - Java JME, JSE

Lua is powerful

- function as first class
- closure
- exception (as library)
- coroutine (as library)
- iterator
- regex (with its own dialect)
- some OO mechanisms (prototype based)
- tail call

Factorial - Loop

```
function factorial (n)
    local a = 1
    for i = 1, n, 1 do
        a = a * i
    end
    return a
end

print(factorial(7))  --> 5040
```

Factorial – recursive

```
function factorial (n)
    if n == 0 then
        return 1
    else
        return n * factorial(n-1)
    end
end

print(factorial(7))  --> 5040
```

Factorial - Iter

```
function factorial (n)
    local function iter (prod, cnt)
        if cnt > n then
            return prod
        else
            return iter(cnt*prod, cnt+1)
        end
    end
    return iter(1, 1)
end

print(factorial(7)) --> 5040
```

Fibonacci - iterator

```
local function fibo_iterator ()
    local x, y = 0, 1
    return function ()
        local res = x
        x, y = y, x+y
        return res
    end
end

for v in fibo_iterator() do
    print(v)
    if v >= 144 then break end
end
```

Fibonacci - Coroutine

```
local function fibo_generator ()
    local x, y = 0, 1
    while true do
        coroutine.yield(x)
        x, y = y, x+y
    end
end

local function fibo_iterator ()
    return coroutine.wrap(fibo_generator)
end

for v in fibo_iterator() do
    print(v)
    if v >= 144 then break end
end
```

Fibonacci – metamethods

```
local fibo = setmetatable({
    [0] = 0,
    [1] = 1,
}, {
    __index =   function (t, n)
                    local res = t[n-1] + t[n-2]
                    t[n] = res -- memoize
                    return res
                end,
    __call =    function (t, n)
                    return t[n]
                end,
})

for i = 0, 12 do
    print(fibo(i))
end
```

Lua is embeddable / extensible

- Designed to be integrated with software writen in C
- C API
  - Comprehensible
  - Well documented
  - Stack model
- All standard libraries are built with
- Userdata is a core type and allows metatable/metamethods

Lua Module Land / Heterogenority

- Build system
  - nothing
  - Makefile
  - CMake
- Documentation
  - LuaDoc : *à la* JavaDoc
  - LuaPOD : with Perl
- QA – Test
  - `assert`
  - lunit : *à la* xUnit
  - Lunity
  - lua-TestMore : *à la* Perl (Test Anything Protocol)
- Packaging / Deployment
  - LuaDist (CMake)
  - LuaRocks

A rockspec sample (plain Lua)

```
package = 'lua-CodeGen'
version = '0.2.2-1'
source = {
   url = 'http://cloud.github.com/downloads/fperrad/lua-CodeGen/lua-codegen-0.2.2.tar.gz',
   md5 = '782a40b6ac55ee3077e10f302e301706',
   dir = 'lua-CodeGen-0.2.2',
}
description = {
   summary = "a template engine",
   detailed = [[
       lua-CodeGen is a "safe" template engine.
       lua-CodeGen enforces a strict Model-View separation.
       lua-CodeGen allows to split template in small chunk, and encourages the reuse of them by inheri
       lua-CodeGen is not dedicated to HTML, it could generate any kind of textual code.
   ]],
   homepage = 'http://fperrad.github.com/lua-CodeGen',
   maintainer = 'Francois Perrad',
   license = 'MIT/X11'
}
dependencies = {
   'lua >= 5.1',
   'lua-testmore >= 0.2.3',
}
build = {
   type = 'builtin',
   modules = {
      ['CodeGen']         = 'src/CodeGen.lua',
      ['CodeGen.Graph']   = 'src/CodeGen/Graph.lua',
   },
   copy_directories = { 'doc', 'test' },
}
```

Binding modules

- LuaSocket : socket, http, smtp
- LuaSec : https
- LuaPOSIX
- LuaExpat : XML
- LuaSQL : mysql, postgres, sqlite, ...
- wxLua : wxWidget
- LuaGnome : GTK
- LuaZMQ
- ...

Other modules / projects

- Kepler : web development platform
  - Orbit : an MVC web framework
  - WSAPI : *à la* WSGI, Rack, PSGI/Plack
  - Sputnik : a wiki engine
- Lpeg : Parsing Expression Grammars
- LuaJSON
- Lua Lanes - multithreading in Lua
- OiL : an Object Request Broker (CORBA)
- ...

Use case : textadept

- Textadept is a fast, minimalist, and ridiculously extensible text editor for Linux, Mac OSX, and Windows
- Lead dev : Mitchell
- started on 2007, 1.0 on Jan 2009
- 2 KLoC of C + 6 KLoc of Lua
- Textadept 4.0 embeds
  - Lua 5.1.4
  - LPeg 0.9
  - LuaFileSystem 1.4.1
  - Scintilla 2.28 / GTK+
- MIT License

Use case : wireshark

- Wireshark (Ethereal) is the world's foremost network protocol analyzer
- > 2 MLoC of C
- Lua can be used to write dissectors, post-dissectors and taps.
- Lua introduced around 0.99.4
- GPL License

Use case : LuaTeX

- LuaTeX is an extended version of pdfTeX using Lua as an embedded scripting language.
- started on 2007
- source size
  - 300 KLoC of C
  - 200 KLoc of C++
  - 10 KLoC of Lua
- GPL License

Use case : awesome

- awesome is an extensible, highly configurable window manager for X.
- started on 2007
- 10 KLoC of C + 7 KLoC of Lua
- It's extremely fast, small, dynamic and heavily extensible using the Lua programming language.
- GPL License

Use case : Redis

- Redis is an open-source, networked, in-memory, persistent, journaled, key-value data store
- Lead dev : antirez
- started on 2009
- 30 KLoC of C
- Scripting branch released on May 2011
  - Server-side scripting with Lua
  - Easy to embed, and FAST
  - `scripting.c` : 500 LoC
- will available with Redis 2.6
- License : BSD

Sponsors

- Adobe
  - Photoshop Lightroom
    - 40% is written in Lua
    - Penlight : A Portable Lua Library
  - The workshop 2005 held at Adobe’s headquarters in San José, California
- SocialMediaPress
- CSTUG
- Océ
  - Printer, copier, plotter
  - The workshop 2006 held at Océ's R&D department in Venlo, The Netherlands

Philosophy of Lua

- Mechanisms instead of policies
- Zen of Lua :
  - *Doing more with less.*

Conclusion

- An embeddable scripting language that is simple, efficient, portable and lightweight

- supports several paradigm of programming :
  - procedural
  - Object-Oriented
  - functional
  - data-driven

Bibliography / Webography

- www.lua.org
- the Lua 5.1 Reference Manual
- www.luafaq.org
- http://lua-users.org/wiki/
- The evolution of Lua
- Small is Beautiful: the design of Lua
- Lua Short Reference
- Programming in Lua
- Lua programming Gems
- LuaForge (frozen since 2009)