École Nationale Supérieure de Techniques Avancées Bretagne

Summer Internship Report

Web development in Lua programming language

# Improvements to Sailor framework during Google Summer of Code

*Author:*
Etiene da Cruz Dalcol

*Supervisor:*
Dr. Olivier Reynet

September 25, 2015

# Contents

# 1 Abstract

Lua is a very fast and powerful scripting language that can be easily embeddable. For that reason, it has gained a niche in the game development industry. Lua has great potential and incredible benchmarks. Despite being also an excellent tool as a general purpose language to develop robust applications, its use in web developments needs to be more widespread.

Sailor was invented to increase the ecosystem of web development in Lua but it is still in early development. The purpose of this work is to turn Sailor into a more mature software by adding new features and improving existing ones such as adding automated tests and improving the use of Lua instead of Javascript when programming for the browser.

## Resumé

Lua est un langage très rapide et puissant qui peut être embarqué facilement. Pour cette raison, il a gagné une place importante dans l'industrie de développement de jeux. Lua a des potentiels et benchmarks incroyables. En dépit d'être aussi un excellent outil en tant que langue d'usage général pour développer des applications robustes, son utilisation dans le développement web devrait être plus généralisée.

Sailor a été inventé pour augmenter l'écosystème du développement web en Lua, mais il est encore dans un stage de développement précoce. Le but de ce travail est de transformer Sailor dans un logiciel plus mature en ajoutant de nouvelles fonctionnalités et l'amélioration de ceux existants, tels que l'ajout de tests automatisés et d'amélioration de l'utilisation de Lua lieu de Javascript lors de la programmation pour le navigateur.

# 2 Introduction

## 2.1 Google Summer of Code

Google Summer of Code (GSoC) is a global program that connects students with open source, free software and technology-related organizations. During the a 3 month period on the summer, the students get familiarised with open source projects, work with the community and write code.

Google identifies open source projects and organizations that will receive funding and participate on the program. The organizations will provide mentors to guide students during the program. Students submit projects to the organizations, who rank them. Organizations may suggest a list of ideas for projects. Once Google defines how many student slots for projects are allocated to an organization, the organization decides which students and projects are accepted and pair them with a mentor.

While most students come from a Computer Science and Software Engineering background, this is not mandatory and the educational area of participants can be very wide.

The program is centered on some goals:[1]

1. Get more open source code written and released for the benefit of all.
2. Inspire young developers to begin participating in open source development.
3. Help open source projects identify and bring in new developers.
4. Provide students the opportunity to do work related to their academic pursuits during the summer: "flip bits, not burgers."
5. Give students more exposure to real-world software development (for example, distributed development and version control, software licensing issues, and mailing list etiquette)."

There are midterm and final evaluations, and the code completed must be submitted to GSoC's website by the end of the program. All development happens online, Google does not provide an office space and there's no requirement to travel.

Since the start of the program, in 2005, over 8500 successful students have participated, from over 109 countries, with 8000 mentors making 55 million lines of code.

## 2.2 LabLua

LabLua was one of the organizations selected to participate in the 2015 version of Google Summer of Code. It is a research lab at the Pontifical Catholic University of Rio de Janeiro (PUC-Rio) affiliated with its Computer Science Department. Its researches are on the field of programming languages, with an emphasis on the Lua programming language. The founder of LabLua, Prof. Roberto Ierusalimschy, is one of the creators of the Lua language.

LabLua proposed the following list of ideas to GSoC:

1. LuaRocks add-ons system

[1] *Google Summer of Code Student Guide*. `http://en.flossmanuals.net/GSoCStudentGuide/index/`. Accessed: 2015-09-25.

2. Port Lua Test Suite to NetBSD Kernel
3. Elasticsearch Lua client (elasticsearch-lua)
4. Add support for left recursion to LPeg
5. Switch Typed Lua from optional typing to gradual typing
6. Adapt CGILua SAPI launcher to explore all WSAPI features
7. Add support for WSDL generation to LuaSOAP
8. Add support for prepared statements in LuaSQL
9. Multi-CPU usage in VLC
10. Multi-CPU usage in wireshark
11. Develop a binary serialization format with support for dynamically types values and an RPC protocol for dynamically typed invocations based on this format.
12. Develop a library for Lua that allows Lua programs to access features provided by the platform's underlying operating system (OS) kernel, such as process control, network access, file system, event notification, etc.
13. Port an SDL-based C++ open source game to Céu

They received in total 6 slots, from which, four of them were filled by students working on ideas 2, 3, 7 and 13 and two of them were ideas proposed by students themselves. One of the student-proposed ideas was my project to improve Sailor, a web development framework I had been developing on my free time.

The project was mentored by Dr. Fabio Mascarenhas, professor at Universidade Federal do Rio de Janeiro. Since Google did not provide for office space or work placement, ENSTA Bretagne offered a placement during the summer to work on this project under the supervision of Professor Dr. Olivier Reynet.

## 2.3 The Lua programming language

> Lua is a powerful, fast, lightweight, embeddable scripting language.[2]

This description provided by lua.org does not fully grasp what's interesting about this language. Lua was created in a very specific context: Petrobras, a multinational energy and oil corporation headquartered in Rio de Janeiro, Brazil had interactive graphical programs for engineering applications. These programs needed some flexibility and they were used by non professional programmers. Lua was created to be very simple yet powerful, allowing the customisation of these softwares through scripting. Avoiding cryptic syntax and semantics, Lua is a very readable language with a short learning curve.

[2] *The Programming Language Lua*. `http://lua.org/`. Accessed: 2015-09-25.

It's possible to have an idea about the simplicity of this language just by comparing the brevity of the Programming in Lua book versus books on pretty much any other language. This means you can have all of the language in your head. Its source code is also very succinct. As of version 5.3.1, described in about 23000 lines of Standard C, the whole distribution which includes documentation has only 276Kb. This allows easily porting Lua code to things that run standard ANSI C, even if they have typically low resources, and embedding Lua to applications without bloating them. The result is that Lua is a very widely used language, from micro controllers and washing machines to operating systems, power computers and video games.

Its power can be observed on its multi-paradigm extensible semantics and speed. For example, while Lua does not come with Object Oriented Programming out of the box, it is possible to prototype it using first class functions [3] and metatables [4]. Lua also performs way better than other popular programming languages on speed tests. A special dialect of Lua called LuaJIT is the fastest it can get while still being a dynamically typed scripting language:[5]

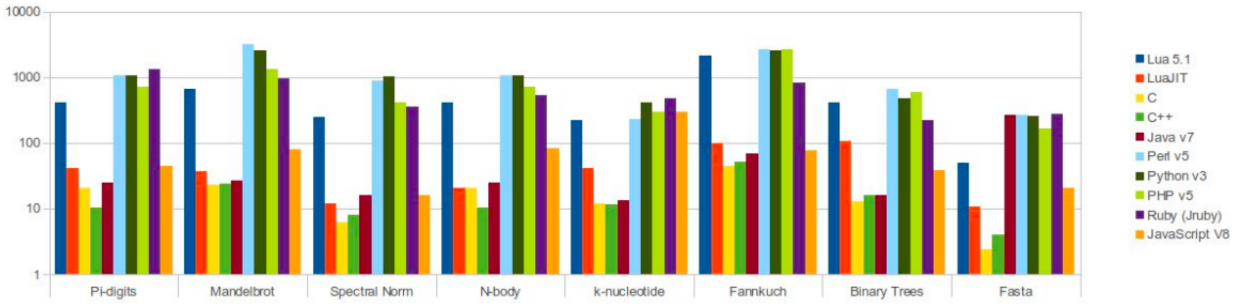

Figure 1: Speed comparison of popular script languages (less is better)

## 2.4 Schedule

**May 25th - June 5th**

Researching how other frameworks use their test suites

**June 6th - June 15th**

Researching and testing existent test Lua modules

**June 16th - July 1st**

Either integrating an existing test module with Sailor or developing a new one

[3] Support of passing functions as arguments to other functions, returning them as the values from other functions, and assigning them to variables or storing them in data structures. (*Wikipedia: First Class Function*. `https://en.wikipedia.org/wiki/First-class_function`. Accessed: 2015-09-25)

[4] Allows to change the behavior of a table. It is part of an extension mechanism which allows you to overload certain operations on Lua objects. (*lua-users wiki: Metamethods Tutorial*. `http://lua-users.org/wiki/MetamethodsTutorial`. Accessed: 2015-09-25) (Roberto Ierusalimschy. *Programming in Lua*. Departamento de Informatica, PUC-Rio, 2006, p. 117)

[5] Daniel Gruno. *Introducing mod lua*. `http://humbedooh.com/presentations/ACNA-mod_lua.odp`. Accessed: 2015-09-25, p. 8.

**July 2nd - July 16th**
Testing, bug fixing and documenting

**July 17th - July 23rd**
Researching and testing Lua to JavaScript VMs. E.g. MoonshineJS

**July 24th - August 6th**
Improving current way to manipulate DOM from Lua and load Lua modules to be used on client side.

**August 7th - August 16th**
Testing, bug fixing and documenting

**August 17th - August 21st**
Polishing and making sure nothing was missed

# 3 Section

## 3.1 Subsection

Structuring a document is easy![6]

### 3.1.1 Subsubsection

More text.[7]

**Paragraph** Some more text.

**Subparagraph** Even more text.

[6] John Doe. *The Book without Title.* Dummy Publisher, 2100, p. 97.

[7] Claudio Vellage. "A quick start to LaTeX". In: (2013).

# 4 Another section

# 5 References

## 5.1 Books

Doe, John. *The Book without Title*. Dummy Publisher, 2100.

Ierusalimschy, Roberto. *Programming in Lua*. Departamento de Informatica, PUC-Rio, 2006.

## 5.2 Articles

Vellage, Claudio. “A quick start to LaTeX”. In: (2013).

## 5.3 Websites

*Google Summer of Code Student Guide*. `http://en.flossmanuals.net/GSoCStudentGuide/index/`. Accessed: 2015-09-25.

Gruno, Daniel. *Introducing mod lua*. `http://humbedooh.com/presentations/ACNA-mod_lua.odp`. Accessed: 2015-09-25.

*lua-users wiki: Metamethods Tutorial*. `http://lua-users.org/wiki/MetamethodsTutorial`. Accessed: 2015-09-25.

*The Programming Language Lua*. `http://lua.org/`. Accessed: 2015-09-25.

*Wikipedia: First Class Function*. `https://en.wikipedia.org/wiki/First-class_function`. Accessed: 2015-09-25.

# 6 Appendix

## List of Figures

## List of Tables