

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:
Etienne DA CRUZ DALCOL

Supervisor:
Dr. Olivier REYNET

September 27, 2015

Contents

1	Abstract	1
2	Introduction	2
2.1	Google Summer of Code	2
2.2	LabLua	2
2.3	The Lua programming language	3
3	Analysis of the problem	4
3.1	The Lua ecosystem	6
3.1.1	General ecosystem	6
3.1.2	Web servers supported	7
3.1.3	Existent tools and frameworks	8
3.2	Schedule	9
4	Section	10
4.1	Subsection	10
4.1.1	Subsubsection	10
5	Another section	11
6	References	12
6.1	Books	12
6.2	Articles	12
6.3	Websites	12
7	Appendix	13

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 Ierusalimsky, 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 CGI Lua 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 typed 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 C  u

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 functions³ and metatables⁴. 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

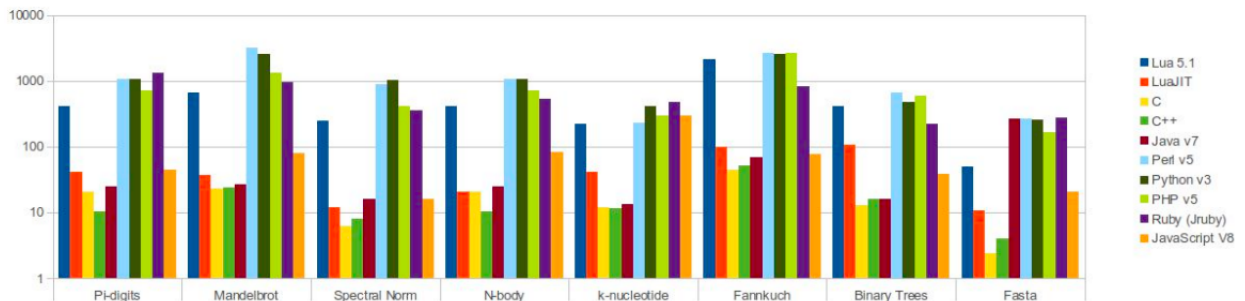


Figure 1: Speed comparison of popular script languages (less is better)

3 Analysis of the problem

Lua is greatly used as an embedded language and has found a niche in game development being the top language of choice for scripting in this area.⁶ In the same time, it is still a great tool as a general purpose language but not much widespread in other domains such as web development. As of September 26th of 2015 Lua is used in less than 0.1% of websites whose server-side programming language we know.⁷

³Support 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)

⁴Allows 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 Ierusalimsky. *Programming in Lua*. Departamento de Informatica, PUC-Rio, 2006, p. 117)

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

⁶Mark DeLoura. *The engine survey: general results*. <http://www.satori.org/2009/03/the-engine-survey-general-results/>. Accessed: 2015-09-26. 2009.

⁷Q-Success. *Usage of server-side programming languages for websites*. http://w3techs.com/technologies/overview/programming_language/all. Accessed: 2015-09-26.

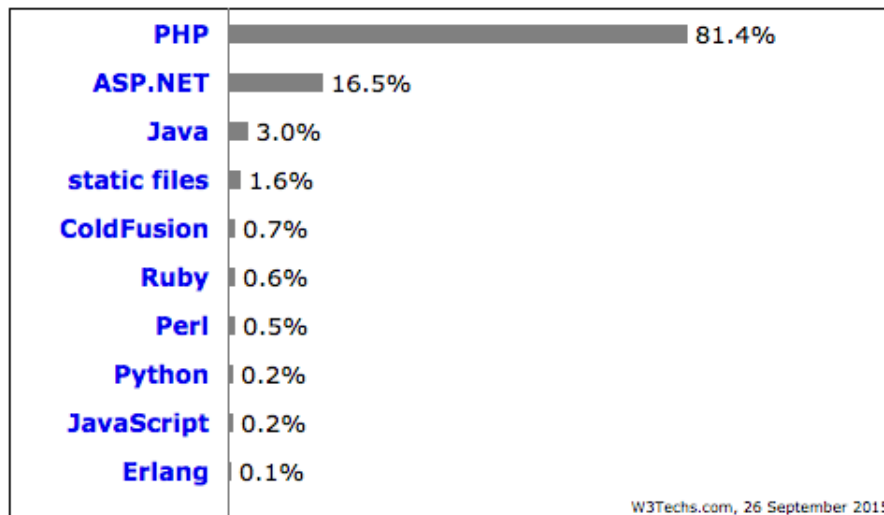


Figure 2: Usage of server-side programming languages for websites. Note: a website may use more than one server-side programming language

In an attempt to understand why this happens, I made an analysis of the context of the top used language: PHP. One may argue that PHP is not such a well-designed language. Then why is it the top language of choice for web development? Why is 82% of the web made of PHP websites? We could speculate that PHP's success is due to a good timing but, 20 years later, allegedly better programming languages have appeared and PHP's popularity continues to grow.⁸

⁸Q-Success. *Historical yearly trends in the usage of server-side programming languages for websites*. http://w3techs.com/technologies/history_overview/programming_language/ms/y. Accessed: 2015-09-26.

	2010 1 Jan	2011 1 Jan	2012 1 Jan	2013 1 Jan	2014 1 Jan	2015 1 Jan	2015 26 Sep
PHP	72.5%	75.3%	77.3%	78.7%	81.6%	82.0%	81.4%
ASP.NET	24.4%	23.4%	21.7%	20.2%	18.2%	17.1%	16.5%
Java	4.0%	3.8%	4.0%	4.1%	2.7%	2.8%	3.0%
static files							1.6%
ColdFusion		1.3%	1.2%	1.1%	0.8%	0.7%	0.7%
Ruby	0.5%	0.5%	0.6%	0.5%	0.4%	0.6%	0.6%
Perl		1.1%	1.0%	0.8%	0.6%	0.5%	0.5%
Python	0.3%	0.3%	0.2%	0.2%	0.2%	0.2%	0.2%
JavaScript			<0.1%	<0.1%	0.1%	0.1%	0.2%
Erlang						0.1%	0.1%

Figure 3: Historical yearly trends in the usage of server-side programming languages for websites

This is what Jeff Atwood, stackoverflow co-founder, calls "The PHP singularity".⁹ According to him, the faults of PHP don't matter because it has a huge ecosystem and easy deployment and the only solution to this is to build compelling alternatives who are as pervasive and as easy to setup and use. In his blog, he calls out the community of programmers to push other options forward.

Having some knowledge of the PHP ecosystem as a beginner and intermediate user, I considered this an interesting challenge. As Lua is a well-designed, fast and easy-to-learn language, I believe it is a perfect candidate for the job in terms of quality. But how about its ecosystem?

3.1 The Lua ecosystem

3.1.1 General ecosystem

Despite being a successful 20-year old language, Lua's community and amount of libraries is small. This is an issue that is being addressed, specially in the recent years. The Kepler Project¹⁰ (2006-2009) was a project developed with the goal to make a platform for web development in Lua.¹¹ While it did not accomplish its ultimate goal, it produced some of the modules that were essential for the later development of the ecosystem. One of these modules that deserves a highlight is LuaRocks,¹² developed by Hisham Muhammad to deploy Kepler modules. LuaRocks is now the package manager for Lua modules, acting as a one-stop repository that significantly reduced the entry barrier for Lua as an application development language. Pierre Chapuis, in his 2013 talk "State of the Lua Ecosys-

⁹Jeff Atwood. *The PHP Singularity*. <http://blog.codinghorror.com/the-php-singularity/>. Accessed: 2015-09-26. 2012.

¹⁰*Kepler 1.1.1*. <https://github.com/keplerproject/kepler>. Accessed: 2015-09-27.

¹¹Andr   Carregal and Tom  s Guisasola. *CGILua: Building Web Scripts with Lua*. <http://keplerproject.github.io/cgilua/>. Accessed: 2015-09-27.

¹²Hisham Muhammad. *LuaRocks: the package manager for Lua modules*. <https://luarocks.org>. Accessed: 2015-09-27.

tem¹³ suggested modules should be even easier to find. He created a website called Lua toolbox,¹⁴ aimed at ranking, endorsing and classification of Lua modules. In the meantime, MoonRocks, a tool aimed at easy uploading and hosting of modules¹⁵ was created by Leaf Corcoran, also in 2013. This project has recently merged with LuaRocks.

All these efforts have been producing results and the ecosystem is growing steadily.¹⁶

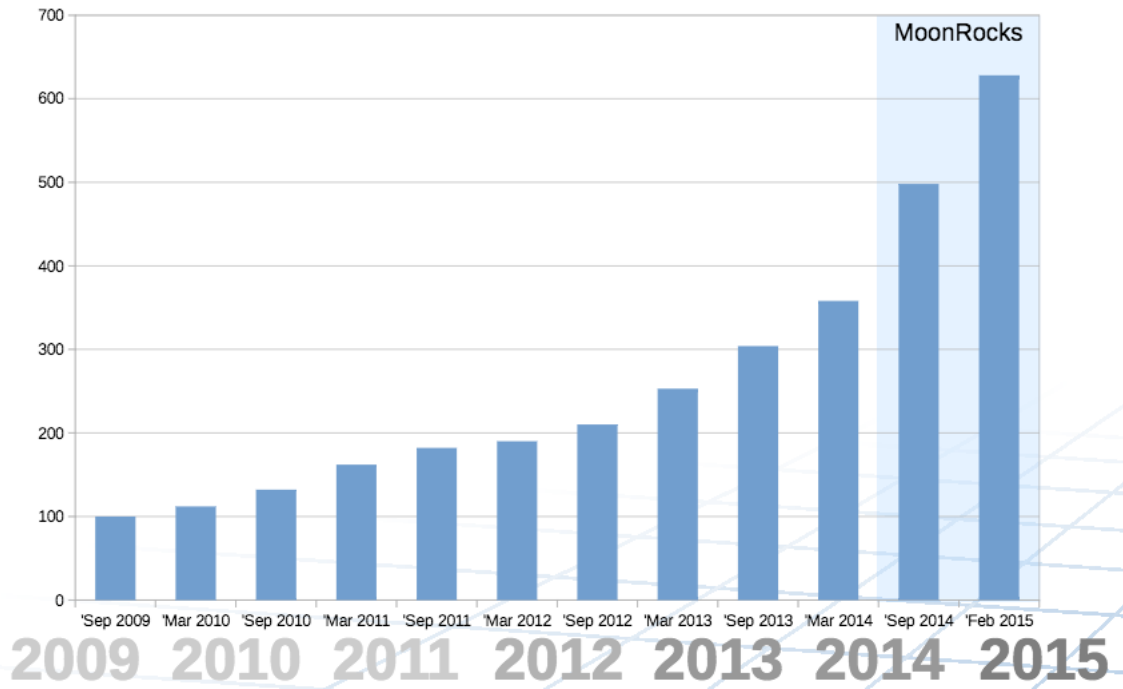


Figure 4: Recent growth of the repository

3.1.2 Web servers supported

Lua can run on a variety of web servers. Apache has a module to run lua called `mod_lua`, Nginx has a distribution called `open resty` that allows to run Lua too. This is great news, because they are the top two web servers used.¹⁷ There's Xavante, which is a web server written in Lua and allows a really quick deploy, as well as plenty of others options like Mongoose, Lwan or Lighttpd.

¹³Pierre Chapuis. *State of the Lua Ecosystem*. <http://files.catwell.info/presentations/2013-11-lua-workshop-lua-ecosystem/>. Accessed: 2015-09-27. 2013.

¹⁴Pierre Chapuis. *Lua Toolbox*. <https://lua-toolbox.com/>. Accessed: 2015-09-27.

¹⁵Leaf Corcoran. *Moonrocks: A command line tool for uploading and installing from the public Lua module hosting site, MoonRocks*. <https://github.com/leafo/moonrocks>. Accessed: 2015-09-27.

¹⁶Hisham Muhammad. *LuaRocks, fostering an ecosystem of Lua modules*. http://hisham.hm/papers/talks/hisham_luarocks_fosdem2015.pdf. Accessed: 2015-09-27. 2015.

¹⁷web servers.

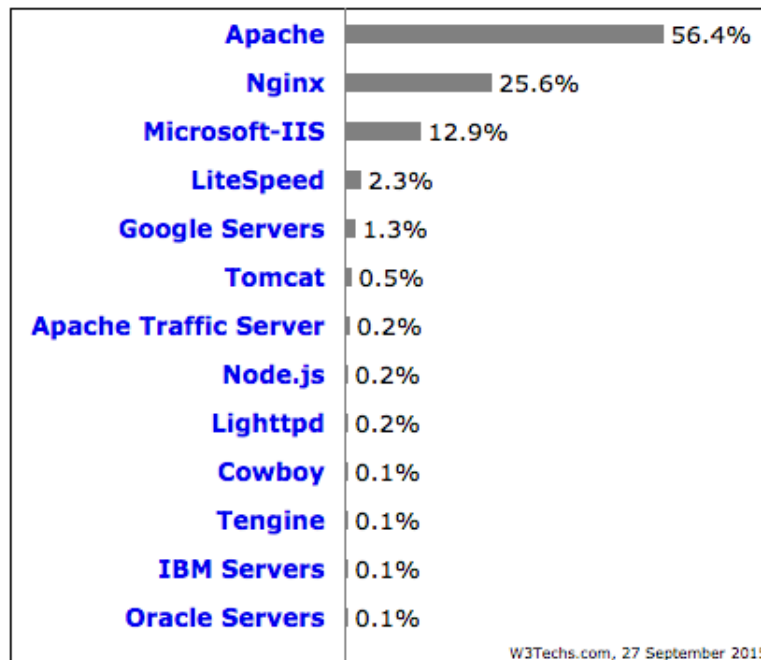


Figure 5: Usage of web servers for websites

3.1.3 Existent tools and frameworks

Orbit:

Orbit is the longest established framework the Lua community has, but it's not in active development anymore. It's very lightweight and follows an MVC architecture.

Can be installed through LuaRocks: Yes

Documentation: Very succinct but sufficient

License: GPL

Website: <http://keplerproject.github.io/orbit/>

Luvit:

Luvit adopts a different kind of approach from Orbit. It's designed to be a port of Node.js to Lua. It accomplishes this task very well, claiming to be 2 to 4 times more performant and making great memory savings. However, for those that are not already proficient in writing applications with Node.js, it can be very complex to use as its documentation is not yet prepared to receive full beginners in the web development world. It's very powerful, it's in intense development and it is maybe the most popular Lua web development framework, but still very little known outside the community.

Can be installed through LuaRocks: No

Documentation: Insufficient

License: Apache 2.0

Website: <https://luvit.io/>

Lapis:

Lapis is a very interesting framework. It's not exactly in Lua, but in MoonScript, a language that compiles to Lua like CoffeeScript compiles to JavaScript. You can still, however, use Lua to write your application. It's in very active development by a dedicated maintainer. It does not provide an MVC architecture out of the box but it's possible to prototype one.

Can be installed through LuaRocks: Yes

Documentation: Extensive and informative

License: MIT

Website: <http://leafo.net/lapis/>

Other projects worth mentioning:

TurboLua | Apache 2.0: <http://www.turbolua.org/>

Tir | BSD: <http://tir.mongrel2.org/>

Ophal | AGPL 3.0: <http://ophal.org/>

3.2 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

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

4 Section

4.1 Subsection

Structuring a document is easy!¹⁸

4.1.1 Subsubsection

More text.¹⁹

Paragraph Some more text.

Subparagraph Even more text.

¹⁸John Doe. *The Book without Title*. Dummy Publisher, 2100, p. 97.

¹⁹Claudio Vellage. “A quick start to L^AT_EX”. In: (2013).

5 Another section

6 References

6.1 Books

Doe, John. *The Book without Title*. Dummy Publisher, 2100.
Ierusalimschy, Roberto. *Programming in Lua*. Departamento de Informatica, PUC-Rio, 2006.

6.2 Articles

Vellage, Claudio. “A quick start to L^AT_EX”. In: (2013).

6.3 Websites

Atwood, Jeff. *The PHP Singularity*. <http://blog.codinghorror.com/the-php-singularity/>. Accessed: 2015-09-26. 2012.

Carregal, Andr   and Tom  s Guisasola. *CGILua: Building Web Scripts with Lua*. <http://keplerproject.github.io/cgilua/>. Accessed: 2015-09-27.

Chapuis, Pierre. *Lua Toolbox*. <https://lua-toolbox.com/>. Accessed: 2015-09-27.

— *State of the Lua Ecosystem*. <http://files.catwell.info/presentations/2013-11-lua-workshop-lua-ecosystem/>. Accessed: 2015-09-27. 2013.

Corcoran, Leaf. *Moonrocks: A command line tool for uploading and installing from the public Lua module hosting site, MoonRocks*. <https://github.com/leafo/moonrocks>. Accessed: 2015-09-27.

DeLoura, Mark. *The engine survey: general results*. <http://www.satori.org/2009/03/the-engine-survey-general-results/>. Accessed: 2015-09-26. 2009.

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.

Kepler 1.1.1. <https://github.com/keplerproject/kepler>. Accessed: 2015-09-27.

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

Muhammad, Hisham. *LuaRocks, fostering an ecosystem of Lua modules*. http://hisham.hm/papers/talks/hisham_luarocks_fosdem2015.pdf. Accessed: 2015-09-27. 2015.

— *LuaRocks: the package manager for Lua modules*. <https://luarocks.org>. Accessed: 2015-09-27.

Q-Success. *Historical yearly trends in the usage of server-side programming languages for websites*. http://w3techs.com/technologies/history_overview/programming_language/ms/y. Accessed: 2015-09-26.

— *Usage of server-side programming languages for websites*. http://w3techs.com/technologies/overview/programming_language/all. Accessed: 2015-09-26.

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.

7 Appendix

List of Figures

1	Speed comparison of popular script languages (less is better)	4
2	Usage of server-side programming languages for websites. Note: a website may use more than one server-side programming language	5
3	Historical yearly trends in the usage of server-side programming languages for websites	6
4	Recent growth of the repository	7
5	Usage of web servers for websites	8

List of Tables