LuaRocks
past, present and future

Hisham Muhammad

Prologue: what is LuaRocks?

- A package manager for Lua modules
- Modules written in Lua (.lua), binary Lua modules (.so/.dll) and Lua scripts
- Usual features you would expect from a package manager
    - luarocks install <module>
    - luarocks remove <module>
    - ...

Rocks and rockspecs

- Rock: a LuaRocks package
  - archive files (actually .zip files)
  - *.src.rock - contains source code
  - *.win32-x86.rock - “binary rock”, contains compiled binaries for a given platform
- Rockspec: a package specification file
  - A declarative Lua script, with rules on how to build and package rocks
  - *.rockspec - a Lua file containing some tables

A rockspec

```
package = "midialsa"; version = "1.17-1"
source = {
   url = "http://www.pjb.com.au/comp/lua/midialsa-1.17.tar.gz",
   md5 = "0482df57c2262ff75f09cec5568352a7"
}
description = {
   summary = "Provides access to the ALSA sequencer", detailed = [[ ... ]],
   homepage = "http://www.pjb.com.au/comp/lua/midialsa.html", license = "MIT/X11"
}
dependencies = { "lua >= 5.1" }
external_dependencies = { ALSA = { header = "alsa/asoundlib.h", library = "asound" }
}
build = {
   type = "builtin",
   modules = {
      ['C-midialsa'] = {
         incdirs = { "$(ALSA_INCDIR)" }, libdirs = { "$(ALSA_LIBDIR)" },
         libraries = { "asound" }, sources = { "C-midialsa.c" }
      },
      midialsa = "midialsa.lua"
   },
   copy_directories = { "doc", "test" }
}
```

Part I

The past: a short history of LuaRocks

Origins

- Kepler Project: research project to develop a platform for web development using Lua
    - combining modules that already existed (LuaSocket, CGILua) and adding the missing pieces
    - For more of the story, read Yuri Takhteyev’s book, “Coding Places” :)
- I started (re)writing Unix makefiles to automate the packaging/install process
- Common patterns emerged

LuaRocks 0.x-1.x: a bumpy start

- 0.x was a gradual evolution
  - the goal for 1.0 was for it to be able to build all Kepler modules
- The rockspec format is unchanged since 1.0
  - We really care about compatibility
  - Learning the format and writing a rockspec are not disposable efforts
- We got many things right, but we also got some of them wrong...

Annoyances in LuaRocks 1.x

- LuaRocks 1.0 did not use the standard Lua layout for modules
  - It wasn’t clear that there was a standard, especially on Windows (Kepler defined its own)
  - On Unix at least, people expect the Unix defaults
    - We fortunately have a policy there!
- It needed a customized require()
  - People didn’t like this
  - It was a clean approach for versioning, though!
- We changed all of this in 2.0
  - Some bad 1st impressions are hard to dispel!

LuaRocks 2.0

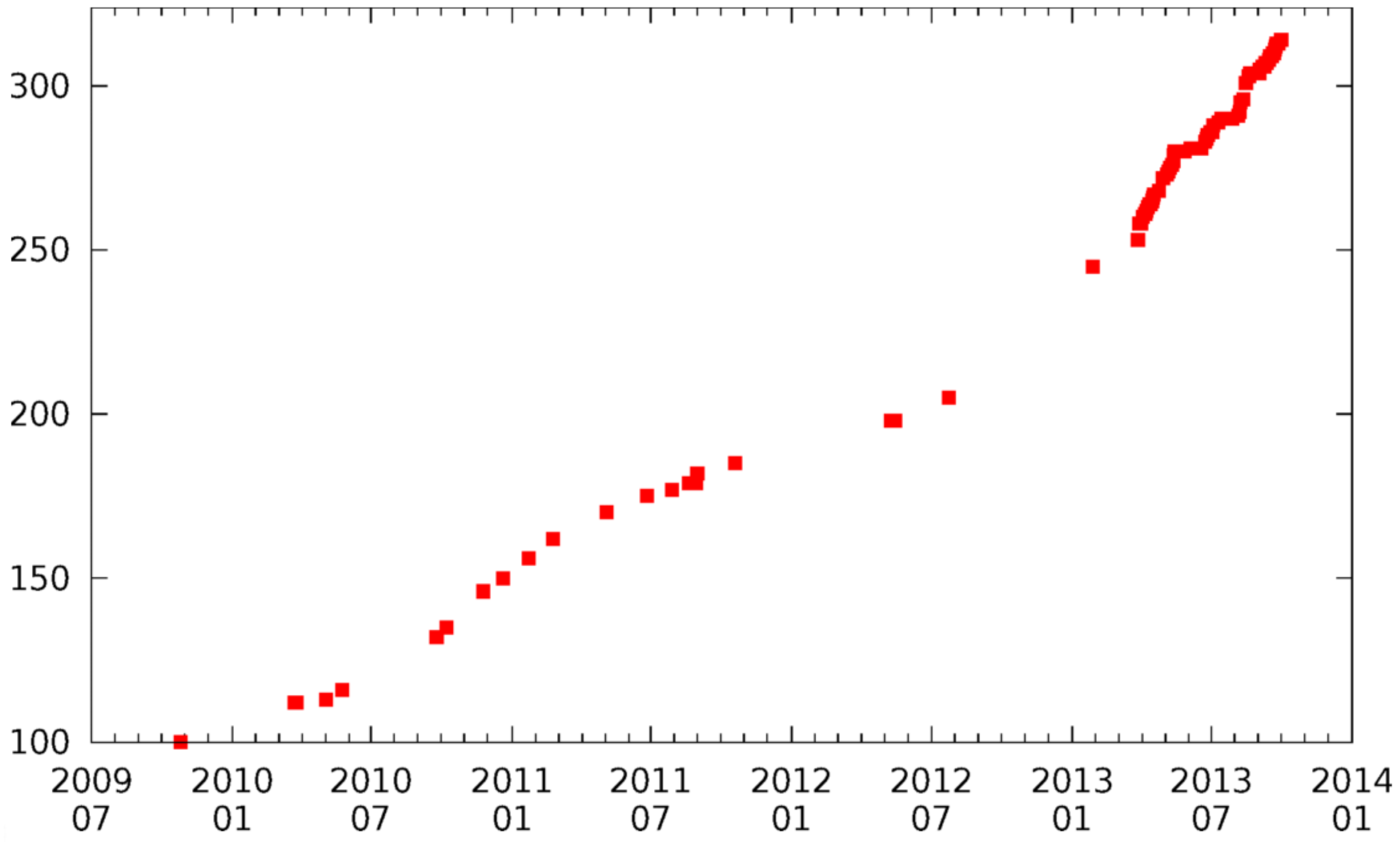
- Plays nice with default paths
    - LR always tried to play nice with distros (first of all, not stepping in their toes)
    - There are limits to what we can do in that front
    - But we’d like more integration! (we’ll get back to this)
- We still support multiple versioning
    - But now we use an optional loader
- You can install modules using LuaRocks but you don’t need it to use them
    - So you could see it just as a build tool like make, scons, etc.
    - This has actually improved recently in 2.1.x

Part II

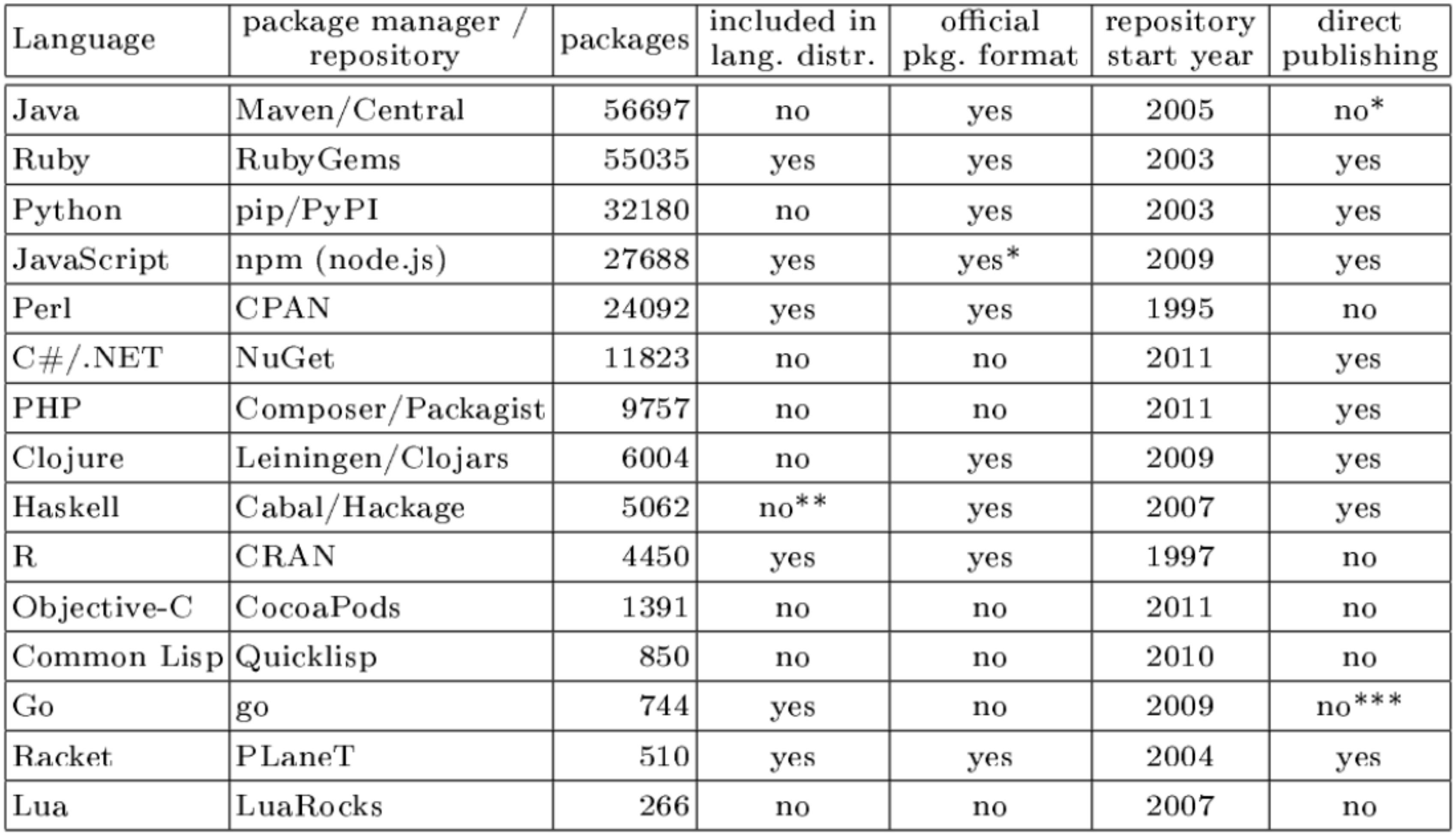
Where we are now

LuaRocks is a reality

- 329 projects, 989 rockspecs



Still, we have a long way to go



Language	package manager / repository	packages	included in lang. distr.	official pkg. format	repository start year	direct publishing
Java	Maven/Central	56697	no	yes	2005	no*
Ruby	RubyGems	55035	yes	yes	2003	yes
Python	pip/PyPI	32180	no	yes	2003	yes
JavaScript	npm (node.js)	27688	yes	yes*	2009	yes
Perl	CPAN	24092	yes	yes	1995	no
C#/.NET	NuGet	11823	no	no	2011	yes
PHP	Composer/Packagist	9757	no	no	2011	yes
Clojure	Leiningen/Clojars	6004	no	yes	2009	yes
Haskell	Cabal/Hackage	5062	no**	yes	2007	yes
R	CRAN	4450	yes	yes	1997	no
Objective-C	CocoaPods	1391	no	no	2011	no
Common Lisp	Quicklisp	850	no	no	2010	no
Go	go	744	yes	no	2009	no***
Racket	PLaneT	510	yes	yes	2004	yes
Lua	LuaRocks	266	no	no	2007	no

as of April 2013

LuaRocks 2.1

- LuaRocks itself is a rock
  - ...on Unix
  - Windows problem: how does a program reinstall itself if you can’t delete open files?
- Making progress in the Windows front
  - Thijs Schreijer has been doing a lot of work there
  - The installer is now a Lua script
  - Better install-time detections all around

The “builtin” mode

- LuaRocks is build-tool agnostic
  - There was no clear leader in the Lua world
  - So we support make, cmake, autoconf, etc.
- But it provides its own Lua-centric build tool
  - build.type=”builtin”
- The numbers show its success
  - ~76% of all rocks use “builtin”, ~10% use “make”
  - 15% of the “builtin” rockspecs used to use “make” and switched
    - mostly because builtin gets portability right automatically
- Use case: BuildRoot

A current annoyance

- Making sure rocks are relocatable is a delicate matter
  - Expected behavior on Windows
  - Unix devs mostly don’t care about it
- LuaDist applies rpath-type patches
- We annoy developers into complying, by building in a temporary sandbox
  - This makes hardcoded paths to data files break
  - (But see http://github.com/hishamhm/datafile)

Rocks server

- http://luarocks.org/repositories/rocks/
- Fueled by rockspec submissions to the luarocks-developers mailing list
- A few ones I still package myself
- A manual process
  - I went with a curated repo early on because of the quality demands of the Lua community
  - ...and also because it was less work then

Part III

Where do we go from here?

Future of the rocks server

- Scalability
  - What happens when/if we reach 50,000 rocks?
    - (Will we ever?)
  - Downloading the whole manifest won’t be feasible
  - We’ll need a proper server-side handler
- Curation
  - I don’t want to take care of the repo forever
  - And I don’t scale, and I miss stuff, take days off, etc.
- MoonRocks
  - Switch the default repo to a “non-curated” one?

LuaDist and Lua for Windows

- LuaDist: CMake-based Lua package manager
  - Some design differences, of course
  - CMake-only is a big con for some
  - Building non-Lua libs is a big pro on Windows
- Lua for Windows: “why not both?”
- Many opportunities for cooperation
  - We’ve been thinking about unifying the rockspec format
  - LuaDist support for the LuaRocks builtin mode
- Looking forward to Peter’s talk!

Improving the interplay with distros

- Is there any interest?
  - from both sides?
- LuaRocks would be happy to be a build tool
  - Bad experiences with other language-specific package managers ruined this for many distros
- What can we do?
  - We try to be system-agnostic
  - It would be nice if we could detect Lua modules already present
  - Perhaps a metafile not unlike pkgconfig .pc files?
- Looking forward to Enrico’s talk!

Further development

- Make LuaRocks embeddable
  so it can work as a plugin manager
- It can do so from the command line today
  - Sputnik, Tarantool
- It would be nice if it could do it as a library
  - This requires some refactoring, but stay tuned...

Long term, where should it go?

- Break it into libraries?
  - luarocks.fs, if cleaned up a bit,
    could be useful on its own
  - LuaDist dependency handling was originally based on the LR codebase -- why don’t we share it?
- Build types (make, cmake, builtin, ...)
  and fetch protocols (file, http, git, svn, …)
  are extensible -- what else can be too?
  - Can we get to a point where whenever someone asks for a new feature we could just reply “just write a new plugin”? :)

In conclusion

- LuaRocks is trying hard to be an enabler for the Lua ecosystem
  - This is happening: we now see some rather nice dependency trees (and less wheel-reinvention?)
  - Reach out to the developers
  - The rockspec format is its main contribution
- but there’s just so much it can do… it’s up to us to build upon the ecosystem
  - Looking forward to Pierre’s talk, which follows!

```
luahue
  penlight
    luafilesystem
  luasocket
  luajson
    lunit
    lpeg
```

```
lua-jet
   lua-cjson
   lua-websockets
      luabitop
      copas
         coxpcall
   luasocket
   lua-ev
   lpack
```