

Опыт работы с LuaIT в нагруженных интернет-проектах

Александр ГЛАДЫШ
CTO, LogicEditor



**Российские
интернет-
технологии**

1. О Lua и LuaJIT

2. Наш нынешний стек

3. Грабли

4. Каким мы видим стек следующего поколения?

5. Хотите знать больше?



Российские
интернет-
технологии

О Lua и LuaJIT

Lua:

- мощный,
- быстрый,
- лёгкий,
- расширяемый,
- встраиваемый

скриптовый язык программирования.



Кратко о языке

О Lua и LuaJIT

- Происхождение.
- Рост популярности в последние годы.
- Где используется?
- LuaRocks.



Российские
интернет-
технологии

Популярные диалекты

О Lua и LuaJIT

- Lua 5.1 vs. Lua 5.2,
- LuaJIT 2.0,
- Metalua.



Российские
интернет-
технологии

LuaJIT 2.0

О Lua и LuaJIT

- JIT, FFI, производительность.
- Поддерживаемые платформы.
- Ограничения для x86_64.
- LuaJIT vs. Lua 5.2.
- Следующие версии LuaJIT.



Почему Lua?

О Lua и LuaJIT

Исторически: Мы вышли из игровой индустрии, где Lua "правит миром".

Прагматически:

- Работает — быстро!
- Писать — удобно!
- Освоить — легко!



Российские
интернет-
технологии

Где искать людей?

О Lua и LuaJIT

Переучивать.



Российские
интернет-
технологии

Основные проблемы при переучивании на Lua

О Lua и LuaJIT

- Неуёмное перетачивание языка под себя:
 - NIH-синдром и лёгкость доработки напильником.
 - Цена и выгоды отхода от мейнстрима.
- Идеосинкразии языка:
 - Переменные по умолчанию — глобальные.
 - Массивы индексируются с 1.
 - Размер массива с `nil` внутри — не определён.
 - Всё, что не `nil` и `false` — истина (включая 0).



Самое главное!

О Lua и LuaJIT

Когда пишете код на Lua — пишите его на Lua!



Российские
интернет-
технологии

Место для Lua / LuaJIT в вашем стэке

О Lua и LuaJIT

В первую очередь:

- Настраиваемая пользователем логика.
- Код, который иначе был бы написан на C/C++/OCaml.



Российские
интернет-
технологии

Фреймворки для построения веб-сервисов на Lua

Наш нынешний стек

Некоторые популярные:

- Kepler/WSAPI
- Luvit
- OpenResty

У нас — велосипед.



Российские
интернет-
технологии

Фреймворки для построения веб-сервисов на Lua

Наш нынешний стек

а) Какие задачи мы решаем?



Российские
интернет-
технологии

Фреймворки для построения веб-сервисов на Lua

Наш нынешний стек

b) На каком железе мы живём?



Российские
интернет-
технологии

Фреймворки для построения веб-сервисов на Lua

Наш нынешний стек

с) Архитектура взаимодействия. XEN, Ubuntu (и её тюнинг), nginx (и его тюнинг), spawn-fcgi, multiwatch, LuaJIT 2, WSAPI, OMQ, Redis (и его тюнинг). DNS-ы. Отдельностоящие сервисы. Почему так?



Российские
интернет-
технологии

Фреймворки для построения веб-сервисов на Lua

Наш нынешний стек

d) Какие луашные библиотеки мы используем и почему? Годные альтернативы нашим историческим opensource-велосипедам (и какие из велосипедов — лучше альтернатив).



Российские
интернет-
технологии

Фреймворки для построения веб-сервисов на Lua

Наш нынешний стек

е) Как сделано High Availability?



Российские
интернет-
технологии

Фреймворки для построения веб-сервисов на Lua

Наш нынешний стек

f) Как устроен деплоймент?



Российские
интернет-
технологии

Фреймворки для построения веб-сервисов на Lua

Наш нынешний стек

g) Как устроен мониторинг?



Российские
интернет-
технологии

Фреймворки для построения веб-сервисов на Lua

Наш нынешний стек

h) Какие показатели по производительности? По стабильности?



Российские
интернет-
технологии

Фреймворки для построения веб-сервисов на Lua

Наш нынешний стек

i) DSL для описания обработчиков запросов. Кодогенерация. Прочие рюшечки и сахар (бонус: DSL для описания SQL-данных с возможностью автогенерации продвинутого UI бэкофиса для этих данных).



Российские
интернет-
технологии

Габли

а) Какие были основные проблемы? Как их решали? Несколько общих советов по отладке и оптимизации производительности при работе с Lua. Отладка отладчиком и по логам, оптимизация GC, какие параметры нужно мониторить. Профайлинг кода на LJ2. Автотесты.



Грабли

б) Какие проблемы не решены, и как с этим жить?



Российские
интернет-
технологии

Габли

i. Long polling / comet.



Российские
интернет-
технологии

Грабли

ii. TODO



Российские
интернет-
технологии

Каким мы видим стек следующего поколения?

а) Ориентироваться на openresty, но не использовать его напрямую. Почему?



Российские
интернет-
технологии

Каким мы видим стек следующего поколения?

b) Новая архитектура.



Российские
интернет-
технологии

Каким мы видим стек следующего поколения?

i. Проще! Ещё проще!



Российские
интернет-
технологии

Каким мы видим стек следующего поколения?

ii. Отказ от LuaRocks.



Российские
интернет-
технологии

Каким мы видим стек следующего поколения?

iii. Полный переход на FFI.



Российские
интернет-
технологии

Каким мы видим стек следующего поколения?

iv. Отказ от FCGI и WSAPI. Переход на eroll и библиотеку парсинга HTTP.



Каким мы видим стек следующего поколения?

v. Отказ от сервера конфигураций.



Российские
интернет-
технологии

Каким мы видим стек следующего поколения?

vi Улучшенная High Availability.



Российские
интернет-
технологии

Каким мы видим стек следующего поколения?

vii Неблокирующее API на корутинах, без коллбэков. Архитектура. Особенности реализации для основных сервисов (HTTP[S], Redis, MySQL/Postgres).



Каким мы видим стек следующего поколения?

viii. Новый дизайн DSL.



Российские
интернет-
технологии

Каким мы видим стек следующего поколения?

ix. ...



Российские
интернет-
технологии

Хотите знать больше?

Official Site lua.org, luajit.org

Wiki lua-users.org/wiki, wiki.luajit.org

Mailing Lists lua.org/lua-l.html, luajit.org/list.html

StackOverflow stackoverflow.com/questions/tagged/Lua

IRC #lua at irc.freenode.net



Российские
интернет-
технологии

Вопросы?

ag@logiceditor.com



Российские
интернет-
технологии