

FFI:

Библиотека FFI:

Использование расширений
C в Ruby

Александр Косс

6 июня 2009

C + Ruby == '❤️' ?

Зачем?

Скорость выполнения

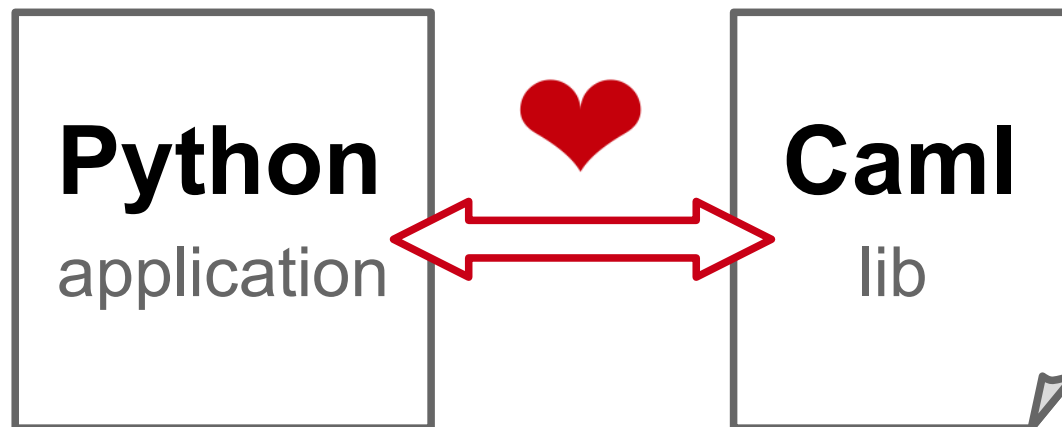
- Интерпретируемые языки медленнее компилируемых

Функциональность

- Большие возможности
- Готовый функционал на языке C

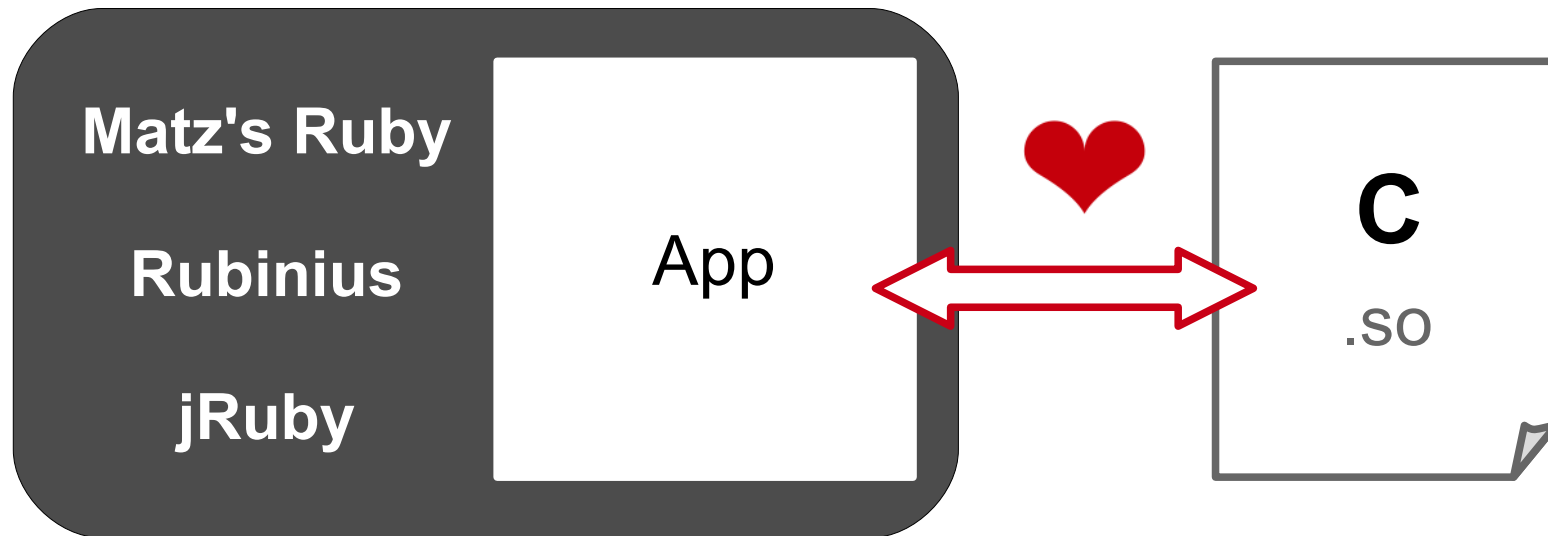
Что такое эф эф ай?

FFI — foreign function interface — это интерфейс вызова функционала написанного на языке отличном от используемого



Ruby FFI

- Автор — Wayne Meissner (+ 42 участника)
- <http://kenai.com/projects/ruby-ffi>
- `sudo gem install ffi`



Пример работы с Ruby FFI

```
1  require 'rubygems'
2  require 'ffi'
3
4  module GetPid
5    extend FFI::Library
6
7    attach_function :getpid, [], :uint
8  end
9
10 puts GetPid.getpid
```

Описание C функций

```
1  int    do_something_a(float a,    float b);
2  char*  do_something_b(char* a_str, char b_chr);
3  double do_something_c(int a,     float b);
4  long   do_something_d(short a,   char b);
```

```
1  module DoSomethingLib
2    extend FFI::Library
3
4    ffi_lib "path/to/do_somethinglib.so"
5
6    attach_function :do_something_a, [:float, :float], :int
7    attach_function :do_something_b, [:string, :char ], :string
8    attach_function :do_something_c, [:int,    :float], :double
9    attach_function :do_something_d, [:short,  :char ], :long
10 end
```

Типы данных

```
1  :char          :uchar          :int8
2  :uint8        :short          :ushort
3  :int16        :uint16         :int
4  :uint         :int32          :uint32
5  :long         :ulong          :int64
6  :uint64       :long_long      :ulong_long
7  :float        :double         :pointer
8  :void         :string         :buffer_in
9  :buffer_out   :buffer_inout   :varargs
10 :char_array
```

Описание C-struct

```
1 struct Dog {  
2     char        *name;  
3     int         age;  
4     int         weight;  
5     struct Dog  *brother;  
6 };  
7
```

```
1 class Dog < FFI::Struct  
2     layout :name,    :string,  
3           :age,     :int,  
4           :weight, :int,  
5           :brother, :pointer  
6
```

Проблемы с C-struct

Проблемы

- Различие между 32-bit и 64-bit платформами
- Утечки памяти

Решения

- Явное указание размера указателя (4 / 8 бит)
- «Деструкторы»

Ruby/DL

```
1  int    do_something_a(float a,    float b);
2  char*  do_something_b(char* a_str, char b_chr);
3  double do_something_c(int a,     float b);
4  long   do_something_d(short a,   char b);
```

```
1  require 'dl/import'
2
3  module DoSomethingLib
4    extend DL::Importable
5
6    dlopen "path/to/do_somethinglib.so"
7
8    extern "int    do_something_a(float, float)"
9    extern "char*  do_something_b(char*, char)"
10   extern "double do_something_c(int,   float)"
11   extern "long   do_something_d(short, char)"
12 end
```

Почему FFI?

- Более красивый синтаксис
- Более быстрое обращение к библиотеке
- Несравнимо быстрый вызов функций из динамической библиотеки

Сравнение быстродействия

```
kossnocorp@kossnocorp-desktop: ~  
=====^  
Performance Test: C vs Ruby; RuPyRu 2009  
=====^  
-----  
Testing C  
-----  
0.000000 sec --- C test  
4.830000 sec --- C this linked shared library test  
0.360000 sec --- C this linked shared library full cicle test  
-----  
Testing Ruby  
-----  
40.12672 sec --- Ruby test  
16.38325 sec --- Ruby FFI test  
65.97072 sec --- Ruby DL test  
0.359066 sec --- Ruby FFI full cycle test  
0.378881 sec --- Ruby DL full cycle test  
-----v
```

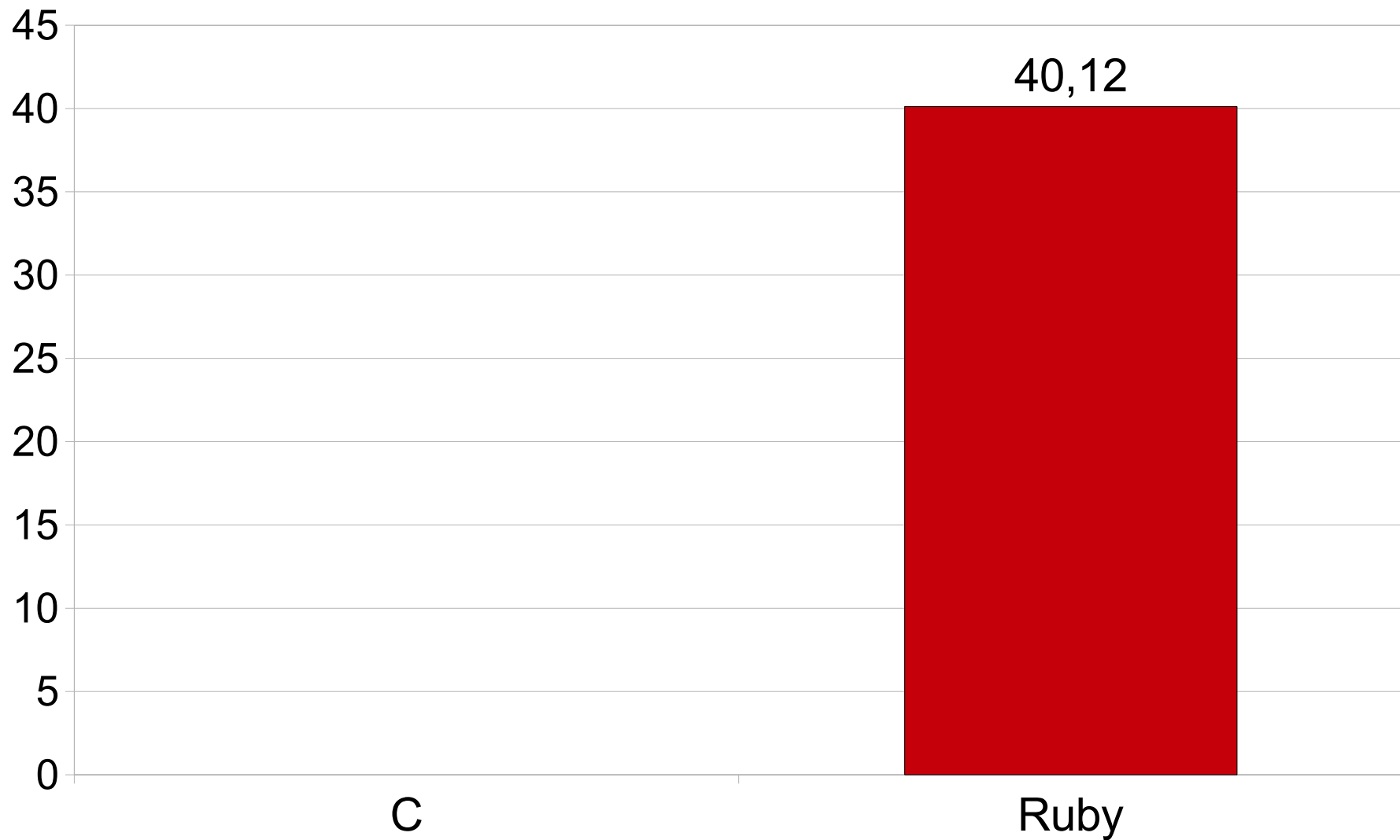
Как проводились тесты

- Сравнение быстродействия C и Ruby
- Сравнение вызова одной «медленной» функции
- Тест производительности вызова функции из динамической библиотеки

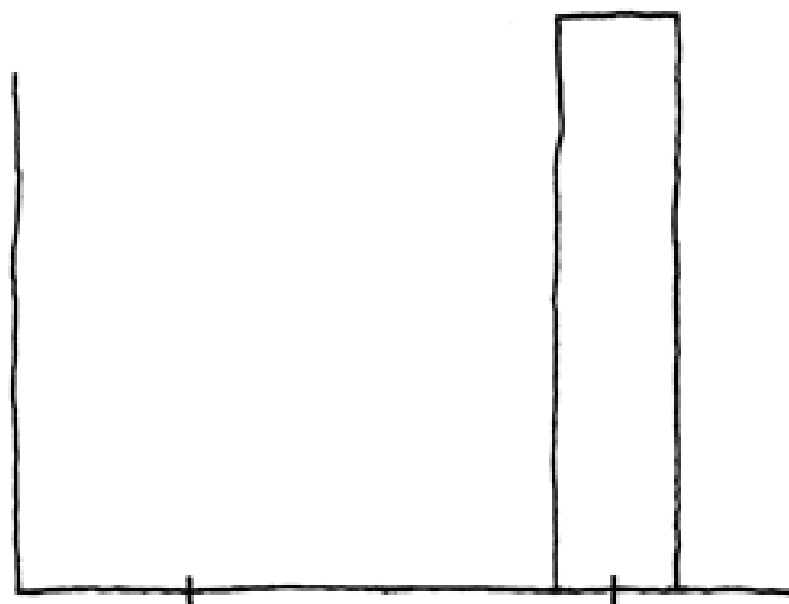
10 000 000 x

$$\frac{a \times b}{a} + a - \frac{b}{a}$$

C vs Ruby



ЗАЯВЛЕНИЯ О СВЕРХЪЕСТЕСТВЕННЫХ СПОСОБНОСТЯХ:

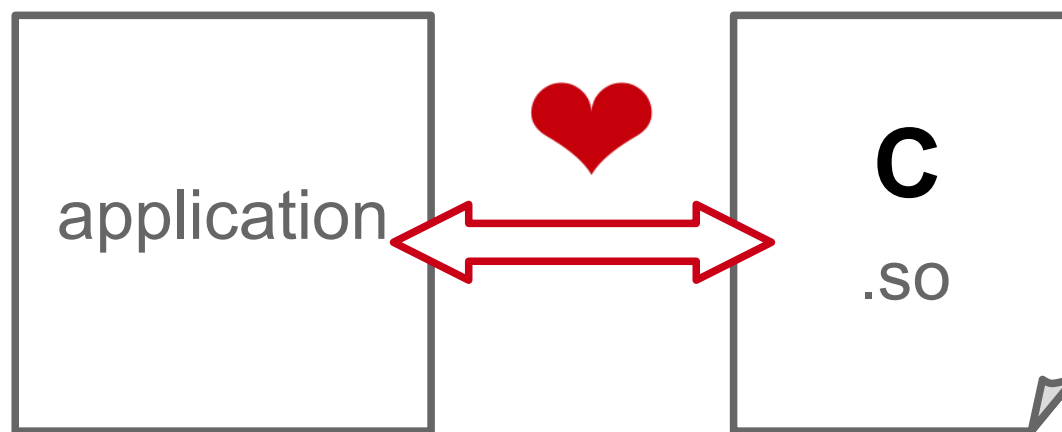


ПОДТВЕРЖДЁННЫЕ
ЭКСПЕРИМЕНТОМ

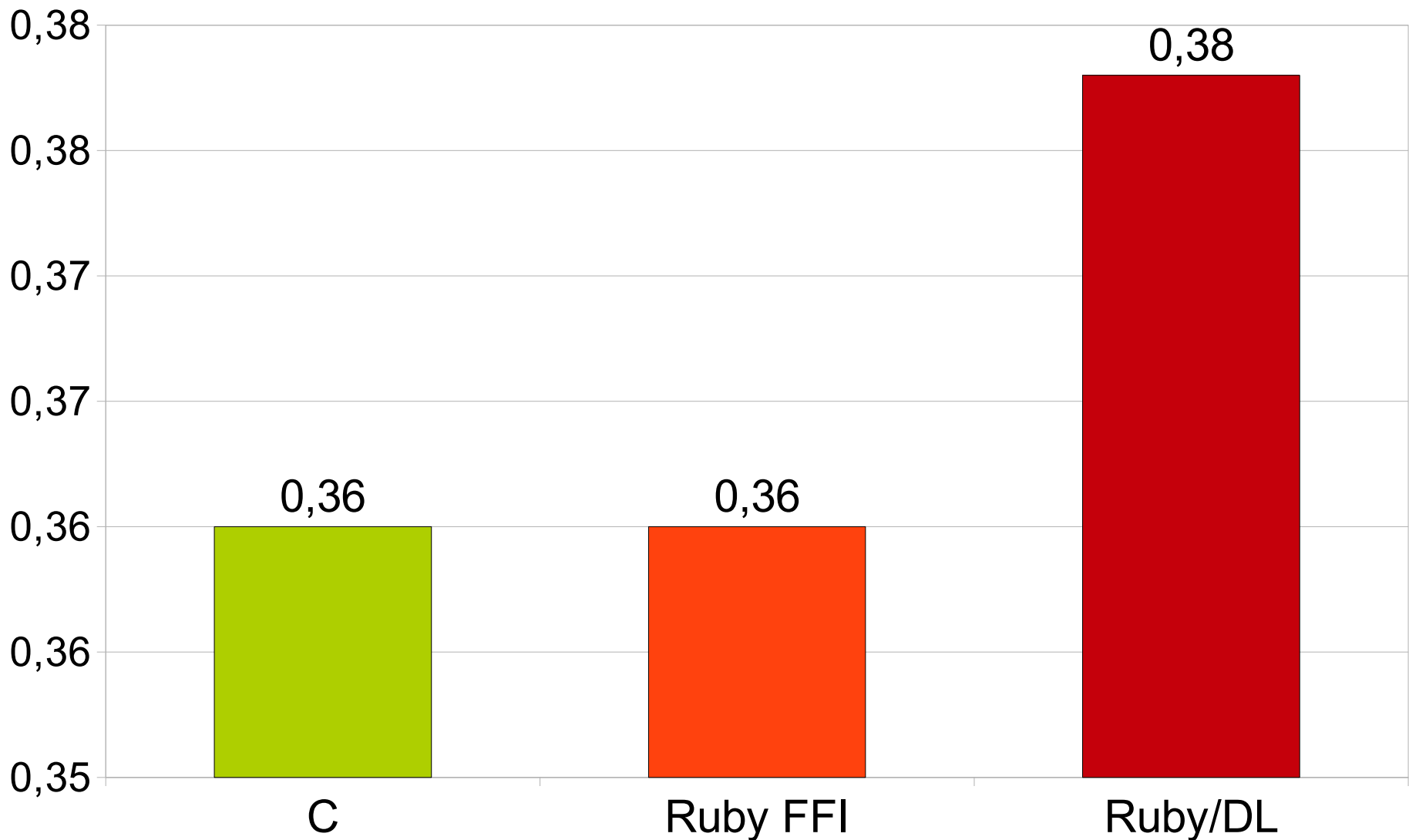
ОПРОВЕРГНУТЫЕ
ЭКСПЕРИМЕНТОМ

Вызов одной функции C

- Из C и Ruby (FFI / DL) приложения вызывается «медленная» функция.
- Наглядно показывает нам потери скорости при вызове одной функции C

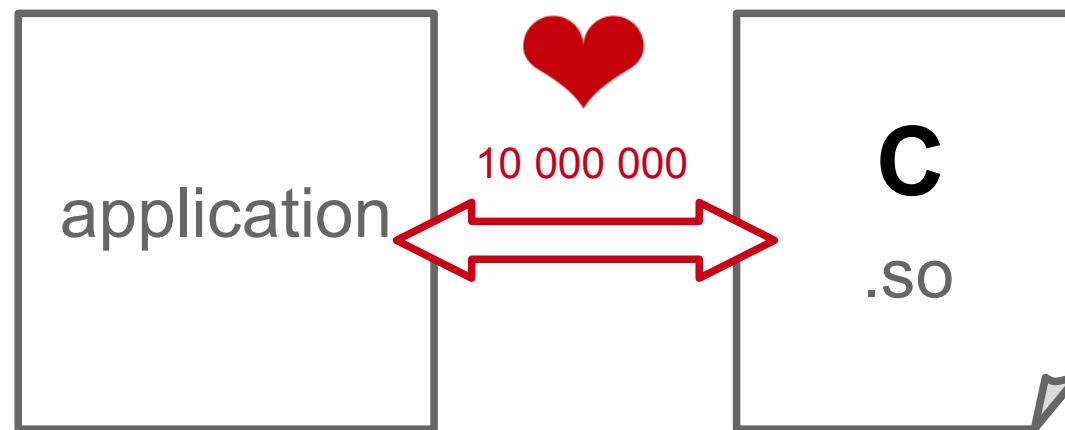


C vs Ruby FFI vs Ruby/DL

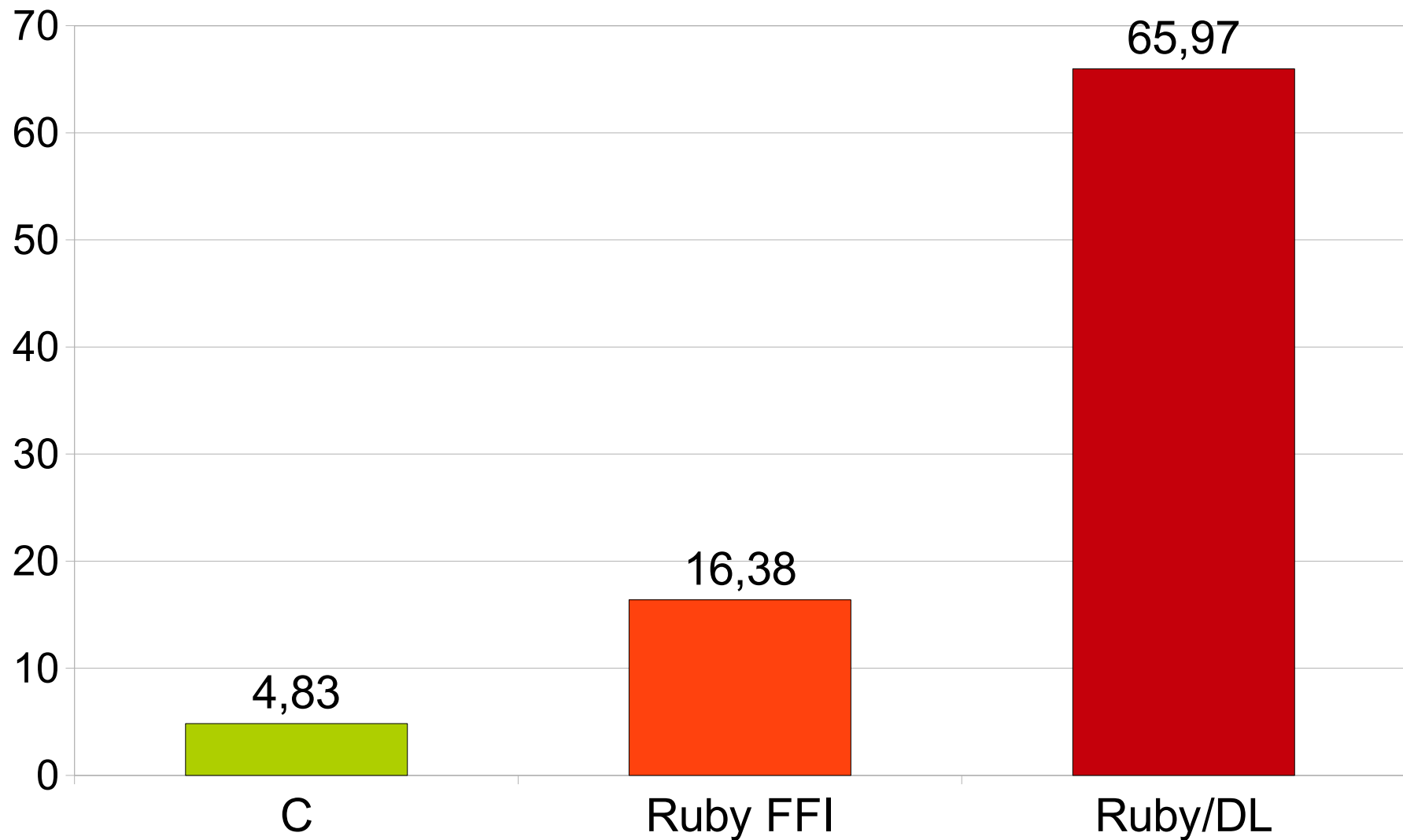


Большое количество вызовов

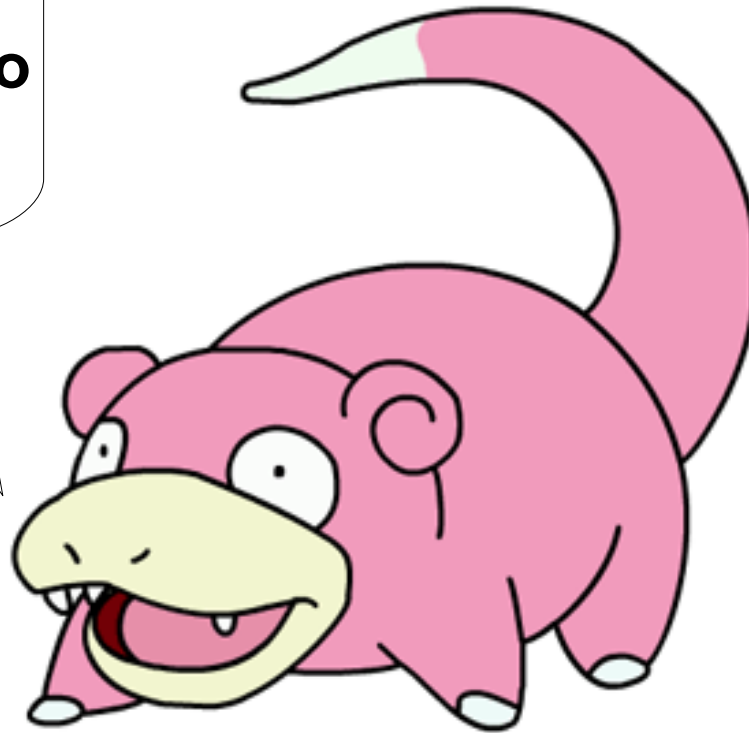
- 10 000 000 вызовов C функции
- Наглядно показывает задержку во время обращения к функции



C vs Ruby FFI vs Ruby/DL



**Ruby/DL
действительно
медлителен!**



C + Ruby = ?

C + Ruby = '❤️' !