

Retro Profiling

Máté Marjai - Ruby Ireland 2015

Why profiling?

- Find expensive methods in your code
- Analyse execution paths
- (In)Validate code smell
- More comprehensive than just benchmarking

Benchmarking != Profiling

Benchmarking is usually associated with assessing performance characteristics of computer hardware

...

Benchmarks provide a method of comparing the performance of various subsystems across different chip/system architectures.

Benchmark example

```
Benchmark.bm do |x|  
  x.report("UserJoins: ") do  
    User.joins(:social_accounts).all  
  end  
end
```

Benchmark example

	user	system	total	real
UserJoins:	1.050000	0.000000	1.050000	(0.503462)

[Benchmark docs](#)

Benchmark

Other gems:

- [minitest](#)
- [benchmark-ips](#)
- [rails-perftest](#)

Profiling



Profiling

A form of dynamic program analysis that measures space (memory), time complexity of a program, frequency and duration of function calls.

Profiling information provides useful information on program optimisation.

Profiling - -rprofile

Ruby has a [built in profiler](#):

- require 'profile'
- can be called from command line

```
$ ruby -rprofile lib/test.rb
```

Profiling - -rprofile

%	cumulative	self		self	total	
time	seconds	seconds	calls	ms/call	ms/call	name
66.67	0.04	0.04	14	2.86	11.43	Kernel#gem_original_require
16.67	0.05	0.01	1	10.00	10.00	OpenSSL::X509::Store#set_default_paths
16.67	0.06	0.01	72	0.14	0.14	nil#
0.00	0.06	0.00	10	0.00	0.00	Integer#to_int
0.00	0.06	0.00	14	0.00	0.00	MonitorMixin#mon_enter
0.00	0.06	0.00	14	0.00	0.00	Kernel#respond_to_missing?

Profiling - ruby-prof

- C extension
- Can measure call times, memory usage and object allocations
- Provides comprehensive reports in various formats

<https://github.com/ruby-prof/ruby-prof>

Profiling - ruby-prof

Examples:

http://github.com/marjaimate/profiling_example

Profiling - ruby-prof



Profiling - other tools

- [perftools](#)
- [rails-perftest](#)
- [rack-mini-profiler](#)
- [stackprof](#)