



# Praktické aspekty vývoje software 2015/2016

Projekt 1 – zpráva o profilování

11. března 2016

Autor: Marek Šipoš (ID: 187044, login: xsipos03)

Fakulta Informačních Technologí  
Vysoké Učení Technické v Brně

## Zadání:

Proveďte profilování neefektivní (`fib-sequence_slow.rb`) i vzorové (`fib-sequence.rb`) implementace Fibbonaciho řady např. metodou "code instrumentation" a výsledky profilování diskutujte v krátké textové zprávě o profilování. Profilování provádějte s velkým množstvím vstupů (různé členy Fibonaciho řady).

## Provedení:

Vzorovou implementaci jsem se rozhodl zpomalit tak, že výpočet jednotlivých členů je proveden skrze novou rekurzivní metodu "recursive". Ta počítá členy voláním sebe-sama, čímž vzniká spousta zbytečných volání a matematických operací.

Okolní prostředí upravené implementace zůstává nedotčeno, provádím pouze směřování výpočtu, díky čemuž je upravená implementace pomalejší, ale stejně jako ta vzorová projde všemi testy.

K profilování jsem použil standartní Ruby profiler, který jsem spouštěl při volání daného programu dodatečným argumentem `-rprofile`.

K otestování obou implementací jsem si vytvořil jednoduché kusy kódu, které využívaly třídy `FibonacciSequence` a jejich metod k výpočtům jednotlivých prvků řady. Výsledky jednotlivých měření následují.

## Výsledky profilování:

Profilování proběhlo přesně dle očekávání. Rekurzivní implementace byla MNOHEM pomalejší, profiler ukázal četná volání matematických operací a metody "recursive", což způsobilo zpomalení implementace geometrickou řadou. Původní implementace si uchovávala stabilní efektivitu i při náročnějších výpočtech, jelikož každý prvek počítala pouze jednou a navíc efektivně.

Zajímavé je srovnání třetího a čtvrtého profilování. Obě proběhla na výpočtu dvacátého prvku Fibbonaciho řady – třetí profilování však využívalo indexace (vracení členu na zadaném indexu), čtvrté využívalo opětovného volání metody `next`. Navzdory možnému očekávání, opakované volání `next` bylo téměř o polovinu rychlejší a efektivnější, hlavně díky využití již vypočítaných prvků, kdežto výpočet členu na zadaném indexu v mé implementaci stále volal metodu "recursive" nehledě na fakt, že některé členy mohly být již vypočítány.

Data z měření:

Kód:

```
require_relative 'fib-sequence' # totez i pro fib-sequence_slow

class Exec
  @fibObject = FibonacciSequence.new
  @fibObject[5]
end
```

Profiling:

```
xsipos03@ivs:~$ ruby -rprofile fib-sequence_exec.rb
% cumulative self self total
time seconds seconds calls ms/call ms/call name
0.00 0.00 0.00 4 0.00 0.00 IO#set_encoding
0.00 0.00 0.00 2 0.00 0.00 Class#inherited
0.00 0.00 0.00 6 0.00 0.00 Module#method_added
0.00 0.00 0.00 1 0.00 0.00 Kernel.require_relative
0.00 0.00 0.00 1 0.00 0.00 FibonacciSequence#reset
0.00 0.00 0.00 1 0.00 0.00 FibonacciSequence#initialize
0.00 0.00 0.00 1 0.00 0.00 Class#new
0.00 0.00 0.00 8 0.00 0.00 Fixnum#<
0.00 0.00 0.00 1 0.00 0.00 Fixnum#<=
0.00 0.00 0.00 10 0.00 0.00 Fixnum#+
0.00 0.00 0.00 17 0.00 0.00 Array#[]
0.00 0.00 0.00 2 0.00 0.00 Kernel.nil?
0.00 0.00 0.00 6 0.00 0.00 FibonacciSequence#next
0.00 0.00 0.00 4 0.00 0.00 NilClass#nil?
0.00 0.00 0.00 8 0.00 0.00 Fixnum#-
0.00 0.00 0.00 4 0.00 0.00 Array#[]=
0.00 0.00 0.00 1 0.00 0.00 Fixnum#>=
0.00 0.00 0.00 1 0.00 0.00 FibonacciSequence#current
0.00 0.00 0.00 1 0.00 0.00 FibonacciSequence#[]
0.00 0.01 0.00 1 0.00 10.00 #toplevel
```

```
xsipos03@ivs:~$ ruby -rprofile fib-sequence_slow_exec.rb
% cumulative self self total
time seconds seconds calls ms/call ms/call name
0.00 0.00 0.00 4 0.00 0.00 IO#set_encoding
0.00 0.00 0.00 2 0.00 0.00 Class#inherited
0.00 0.00 0.00 7 0.00 0.00 Module#method_added
0.00 0.00 0.00 1 0.00 0.00 Kernel.require_relative
0.00 0.00 0.00 1 0.00 0.00 FibonacciSequence#reset
0.00 0.00 0.00 1 0.00 0.00 FibonacciSequence#initialize
0.00 0.00 0.00 1 0.00 0.00 Class#new
0.00 0.00 0.00 8 0.00 0.00 Fixnum#<
0.00 0.00 0.00 48 0.00 0.00 Fixnum#<=
0.00 0.00 0.00 27 0.00 0.00 Fixnum#+
0.00 0.00 0.00 8 0.00 0.00 Array#[]
0.00 0.00 0.00 2 0.00 0.00 Kernel.nil?
0.00 0.00 0.00 6 0.00 0.00 FibonacciSequence#next
0.00 0.00 0.00 4 0.00 0.00 NilClass#nil?
0.00 0.00 0.00 42 0.00 0.00 Fixnum#-
0.00 0.00 0.00 47 0.00 0.00 FibonacciSequence#recursive
0.00 0.00 0.00 4 0.00 0.00 Array#[]=
0.00 0.00 0.00 1 0.00 0.00 FibonacciSequence#[]
0.00 0.01 0.00 1 0.00 10.00 #toplevel
```

Kód:

```
require_relative 'fib-sequence' # totez i pro fib-sequence_slow

class Exec
  @fibObject = FibonacciSequence.new
  @fibObject[10]
end
```

Profiling:

```
xsipos03@ivs:~$ ruby -rprofile fib-sequence_exec.rb
% cumulative self self total
time seconds seconds calls ms/call ms/call name
0.00 0.00 0.00 4 0.00 0.00 IO#set_encoding
0.00 0.00 0.00 2 0.00 0.00 Class#inherited
0.00 0.00 0.00 6 0.00 0.00 Module#method_added
0.00 0.00 0.00 1 0.00 0.00 Kernel.require_relative
0.00 0.00 0.00 1 0.00 0.00 FibonacciSequence#reset
0.00 0.00 0.00 1 0.00 0.00 FibonacciSequence#initialize
0.00 0.00 0.00 1 0.00 0.00 Class#new
0.00 0.00 0.00 13 0.00 0.00 Fixnum#<
0.00 0.00 0.00 1 0.00 0.00 Fixnum#<=
0.00 0.00 0.00 20 0.00 0.00 Fixnum#+
0.00 0.00 0.00 32 0.00 0.00 Array#[]
0.00 0.00 0.00 2 0.00 0.00 Kernel.nil?
0.00 0.00 0.00 11 0.00 0.00 FibonacciSequence#next
0.00 0.00 0.00 9 0.00 0.00 NilClass#nil?
0.00 0.00 0.00 18 0.00 0.00 Fixnum#-
0.00 0.00 0.00 9 0.00 0.00 Array#[]=
0.00 0.00 0.00 1 0.00 0.00 Fixnum#>=
0.00 0.00 0.00 1 0.00 0.00 FibonacciSequence#current
0.00 0.00 0.00 1 0.00 0.00 FibonacciSequence#[]
0.00 0.01 0.00 1 0.00 10.00 #toplevel
```

```
xsipos03@ivs:~$ ruby -rprofile fib-sequence_slow_exec.rb
% cumulative self self total
time seconds seconds calls ms/call ms/call name
66.67 0.02 0.02 628 0.03 0.29 FibonacciSequence#recursive
33.33 0.03 0.01 629 0.02 0.02 Fixnum#<=
0.00 0.03 0.00 320 0.00 0.00 Fixnum#+
0.00 0.03 0.00 1 0.00 0.00 Kernel.require_relative
0.00 0.03 0.00 1 0.00 0.00 FibonacciSequence#reset
0.00 0.03 0.00 1 0.00 0.00 FibonacciSequence#initialize
0.00 0.03 0.00 1 0.00 0.00 Class#new
0.00 0.03 0.00 13 0.00 0.00 Fixnum#<
0.00 0.03 0.00 4 0.00 0.00 IO#set_encoding
0.00 0.03 0.00 7 0.00 0.00 Module#method_added
0.00 0.03 0.00 13 0.00 0.00 Array#[]
0.00 0.03 0.00 2 0.00 0.00 Kernel.nil?
0.00 0.03 0.00 11 0.00 1.82 FibonacciSequence#next
0.00 0.03 0.00 9 0.00 0.00 NilClass#nil?
0.00 0.03 0.00 618 0.00 0.00 Fixnum#-
0.00 0.03 0.00 2 0.00 0.00 Class#inherited
0.00 0.03 0.00 9 0.00 0.00 Array#[]=
0.00 0.03 0.00 1 0.00 30.00 FibonacciSequence#[]
0.00 0.03 0.00 1 0.00 30.00 #toplevel
```

## Kód:

```
require_relative 'fib-sequence' # totez i pro fib-sequence_slow

class Exec
  @fibObject = FibonacciSequence.new
  @fibObject[20]
end
```

## Profiling:

```
xsipos03@ivs:~$ ruby -rprofile fib-sequence_exec.rb
% cumulative self self total
time seconds seconds calls ms/call ms/call name
0.00 0.00 0.00 4 0.00 0.00 IO#set_encoding
0.00 0.00 0.00 2 0.00 0.00 Class#inherited
0.00 0.00 0.00 6 0.00 0.00 Module#method_added
0.00 0.00 0.00 1 0.00 0.00 Kernel.require_relative
0.00 0.00 0.00 1 0.00 0.00 FibonacciSequence#reset
0.00 0.00 0.00 1 0.00 0.00 FibonacciSequence#initialize
0.00 0.00 0.00 1 0.00 0.00 Class#new
0.00 0.00 0.00 23 0.00 0.00 Fixnum#<
0.00 0.00 0.00 1 0.00 0.00 Fixnum#<=
0.00 0.00 0.00 40 0.00 0.00 Fixnum#+
0.00 0.00 0.00 62 0.00 0.00 Array#[]
0.00 0.00 0.00 2 0.00 0.00 Kernel.nil?
0.00 0.00 0.00 21 0.00 0.00 FibonacciSequence#next
0.00 0.00 0.00 19 0.00 0.00 NilClass#nil?
0.00 0.00 0.00 38 0.00 0.00 Fixnum#-
0.00 0.00 0.00 19 0.00 0.00 Array#[]=
0.00 0.00 0.00 1 0.00 0.00 Fixnum#>=
0.00 0.00 0.00 1 0.00 0.00 FibonacciSequence#current
0.00 0.00 0.00 1 0.00 0.00 FibonacciSequence#[]
0.00 0.01 0.00 1 0.00 10.00 #toplevel
```

```
xsipos03@ivs:~$ ruby -rprofile fib-sequence_slow_exec.rb
% cumulative self self total
time seconds seconds calls ms/call ms/call name
78.64 1.62 1.62 79180 0.02 0.32 FibonacciSequence#recursive
9.22 1.81 0.19 79181 0.00 0.00 Fixnum#<=
7.28 1.96 0.15 79160 0.00 0.00 Fixnum#-
4.85 2.06 0.10 39601 0.00 0.00 Fixnum#+
0.00 2.06 0.00 4 0.00 0.00 IO#set_encoding
0.00 2.06 0.00 1 0.00 0.00 FibonacciSequence#initialize
0.00 2.06 0.00 1 0.00 0.00 Class#new
0.00 2.06 0.00 23 0.00 0.00 Fixnum#<
0.00 2.06 0.00 1 0.00 0.00 Kernel.require_relative
0.00 2.06 0.00 1 0.00 0.00 FibonacciSequence#reset
0.00 2.06 0.00 23 0.00 0.00 Array#[]
0.00 2.06 0.00 2 0.00 0.00 Kernel.nil?
0.00 2.06 0.00 21 0.00 73.33 FibonacciSequence#next
0.00 2.06 0.00 19 0.00 0.00 NilClass#nil?
0.00 2.06 0.00 7 0.00 0.00 Module#method_added
0.00 2.06 0.00 2 0.00 0.00 Class#inherited
0.00 2.06 0.00 19 0.00 0.00 Array#[]=
0.00 2.06 0.00 1 0.00 2060.00 FibonacciSequence#[]
0.00 2.06 0.00 1 0.00 2060.00 #toplevel
```

## Kód:

```
require_relative 'fib-sequence' # totez i pro fib-sequence_slow

class Exec
  @fibObject = FibonacciSequence.new
  20.times { @fibObject.next }
end
```

## Profiling:

```
xsipos03@ivs:~$ ruby -rprofile fib-sequence_exec.rb
% cumulative self self total
time seconds seconds calls ms/call ms/call name
0.00 0.00 0.00 4 0.00 0.00 IO#set_encoding
0.00 0.00 0.00 2 0.00 0.00 Class#inherited
0.00 0.00 0.00 6 0.00 0.00 Module#method_added
0.00 0.00 0.00 1 0.00 0.00 Kernel.require_relative
0.00 0.00 0.00 1 0.00 0.00 FibonacciSequence#reset
0.00 0.00 0.00 1 0.00 0.00 FibonacciSequence#initialize
0.00 0.00 0.00 1 0.00 0.00 Class#new
0.00 0.00 0.00 38 0.00 0.00 Fixnum#+
0.00 0.00 0.00 58 0.00 0.00 Array#[]
0.00 0.00 0.00 2 0.00 0.00 Kernel.nil?
0.00 0.00 0.00 20 0.00 0.00 FibonacciSequence#next
0.00 0.00 0.00 18 0.00 0.00 NilClass#nil?
0.00 0.00 0.00 36 0.00 0.00 Fixnum#-
0.00 0.00 0.00 18 0.00 0.00 Array#[]=
0.00 0.00 0.00 1 0.00 0.00 Integer#times
0.00 0.01 0.00 1 0.00 10.00 #toplevel
```

```
xsipos03@ivs:~$ ruby -rprofile fib-sequence_slow_exec.rb
% cumulative self self total
time seconds seconds calls ms/call ms/call name
81.73 0.85 0.85 35398 0.02 0.34 FibonacciSequence#recursive
6.73 0.92 0.07 35398 0.00 0.00 Fixnum#<=
6.73 0.99 0.07 35380 0.00 0.00 Fixnum#-
4.81 1.04 0.05 17710 0.00 0.00 Fixnum#+
0.00 1.04 0.00 22 0.00 0.00 Array#[]
0.00 1.04 0.00 1 0.00 0.00 FibonacciSequence#initialize
0.00 1.04 0.00 1 0.00 0.00 Class#new
0.00 1.04 0.00 4 0.00 0.00 IO#set_encoding
0.00 1.04 0.00 1 0.00 0.00 FibonacciSequence#reset
0.00 1.04 0.00 2 0.00 0.00 Kernel.nil?
0.00 1.04 0.00 20 0.00 52.00 FibonacciSequence#next
0.00 1.04 0.00 18 0.00 0.00 NilClass#nil?
0.00 1.04 0.00 1 0.00 0.00 Kernel.require_relative
0.00 1.04 0.00 7 0.00 0.00 Module#method_added
0.00 1.04 0.00 2 0.00 0.00 Class#inherited
0.00 1.04 0.00 18 0.00 0.00 Array#[]=
0.00 1.04 0.00 1 0.00 1040.00 Integer#times
0.00 1.04 0.00 1 0.00 1040.00 #toplevel
```