



Praktické aspekty vývoje software

2023/2024

Projekt č. 2 - zpráva z profilingu

Tým "Kuře před E112"

Jakub Havlík (xhavlij00)
Martin Vaculík (xvaculm00)
David Bujzaš (xbujzad00)
Erik Polák (xpolake00))

Brno, 13. dubna 2024

1 Zadání

Pomocí funkcí z Vaší matematické knihovny vytvořte program (jako samostatný spustitelný soubor) pro výpočet výběrové směrodatné odchylky z posloupnosti čísel, kterou program čte ze standardního vstupu (v C např. pomocí funkce `scanf`) až do konce souboru a musí být schopen načíst min. 1000 čísel (zkontrolujeme). Na vstupu budou pouze čísla oddělená bílými znaky (mezera, konec řádku nebo tabulátor) a jejich počet není předem dán. Vzorec pro výběrovou směrodatnou odchylku, který bude využit:

$$\bar{x} = \frac{1}{N} \sum_{i=1}^N x_i$$

Obrázek 1: Vzorec pro směrodatnou odchylku

2 Implementace profilingu

Tým pro realizaci profilingu zvolil profiler pro jazyk Python - cProfile, jehož výstupem je Flat profile, ze kterého lze analyzovat, jaká místa je třeba optimalizovat. Výstup je řazen dle "tottime"-stráveného času v jednotlivých funkcích. Kromě toho sledujeme i počet volání jednotlivých funkcí.

cProfile dle dokumentace funguje na základě "deterministického profilingu". Instrumentace kódu není potřeba, jelikož Python je skriptovací jazyk a je kód je tedy již interpretován.

3 Měření

Program byl spuštěn s pseudo-náhodně generovanými vstupy, hodnoty se vyskytovaly typicky v rámci tisíců, pomocí kterých se vypočítala směrodatná odchylka. Pro toto měření jsme uvažovali $10^1, 10^3, 10^6$ vstupů, dodatečně i v řádech desítek milionů¹ vstupů. Jednotlivé výstupy viz sekce 5.

Program byl spouštěn pomocí příkazu `python3 profiling.py < input_pocetvstupu.txt`

Meření bylo prováděno na systému s nainstalovanou verzí Python 3.10.12, Windows 10 22H2. Specifikace zařízení: Intel Core i5-12400F, 16GB RAM DDR4 2667 MHz, RTX 3060Ti, MSI PRO B660-A, Samsung M.2 NVMe 980.

4 Analýza

Na základě naměřených hodnot je zřejmé, že při velkém počtu vstupu je program velmi pomalý. Jedním z důvodů je samotný interpretovaný jazyk Python. Velkým problémem je námi implementovaná funkce `pow`, která počítá mocninu čísla. V této funkci program totiž tráví příliš mnoho času, proto by ji bylo třeba optimalizovat. Kromě toho další velkou zátěží je implementovaná funkce `contains_decimal` používaná pro správné vyhodnocení vstupních čísel. Tuto funkci by také bylo vhodné implementovat jiným způsobem, případně se snažit zredukovat počet volání této funkce.

¹Při více vstupech (např. 100 milionů) program byl moc pomalý a byl násilně ukončen.

5 Výstupy

Měření bylo s jednotlivými sadami vstupů opakovaně a celkový čas se obvykle lišil pouze v řádech setin sekundy.

```
3.0276503540974917
62 function calls in 0.001 seconds

Ordered by: internal time

ncalls  tottime  percall  cumtime  percall filename:lineno(function)
1      0.000    0.000    0.000    0.000 profiling.py:9(profiler_function)
1      0.000    0.000    0.000    0.000 {built-in method builtins.print}
1      0.000    0.000    0.001    0.001 {built-in method builtins.exec}
11     0.000    0.000    0.000    0.000 mathlib.py:108(pow)
3      0.000    0.000    0.000    0.000 codecs.py:319(decode)
3      0.000    0.000    0.000    0.000 {built-in method _codecs.utf_8_decode}
20     0.000    0.000    0.000    0.000 mathlib.py:40(add)
2      0.000    0.000    0.000    0.000 mathlib.py:60(div)
1      0.000    0.000    0.000    0.000 <string>:1(<module>)
10     0.000    0.000    0.000    0.000 {method 'append' of 'list' objects}
1      0.000    0.000    0.000    0.000 mathlib.py:83(root)
2      0.000    0.000    0.000    0.000 mathlib.py:50(sub)
2      0.000    0.000    0.000    0.000 mathlib.py:73(mul)
3      0.000    0.000    0.000    0.000 {built-in method builtins.len}
1      0.000    0.000    0.000    0.000 {method 'disable' of '_lsprof.Profiler' objects}
```

Obrázek 2: 10 vstupů

```
2961.3204060464964
4501 function calls in 0.012 seconds

Ordered by: internal time

ncalls  tottime  percall  cumtime  percall filename:lineno(function)
1      0.012    0.012    0.012    0.012 profiling.py:9(profiler_function)
1001    0.000    0.000    0.000    0.000 mathlib.py:108(pow)
2000    0.000    0.000    0.000    0.000 mathlib.py:40(add)
479     0.000    0.000    0.000    0.000 mathlib.py:23(contains_decimals)
1000    0.000    0.000    0.000    0.000 {method 'append' of 'list' objects}
1      0.000    0.000    0.000    0.000 {built-in method builtins.print}
1      0.000    0.000    0.012    0.012 {built-in method builtins.exec}
3      0.000    0.000    0.000    0.000 codecs.py:319(decode)
3      0.000    0.000    0.000    0.000 {built-in method _codecs.utf_8_decode}
1      0.000    0.000    0.012    0.012 <string>:1(<module>)
2      0.000    0.000    0.000    0.000 mathlib.py:60(div)
2      0.000    0.000    0.000    0.000 mathlib.py:50(sub)
1      0.000    0.000    0.000    0.000 {method 'disable' of '_lsprof.Profiler' objects}
2      0.000    0.000    0.000    0.000 mathlib.py:73(mul)
3      0.000    0.000    0.000    0.000 {built-in method builtins.len}
1      0.000    0.000    0.000    0.000 mathlib.py:83(root)
```

Obrázek 3: 1000 vstupů

```

2885.525151230861
    4501954 function calls in 0.885 seconds

Ordered by: internal time

ncalls  tottime  percall  cumtime  percall filename:lineno(function)
      1    0.489    0.489    0.880    0.880 profiling.py:9(profiler_function)
1000001    0.171    0.000    0.249    0.000 mathlib.py:108(pow)
2000000    0.103    0.000    0.103    0.000 mathlib.py:40(add)
499888    0.078    0.000    0.078    0.000 mathlib.py:23(contains_decimals)
1000000    0.037    0.000    0.037    0.000 {method 'append' of 'list' objects}
      1    0.005    0.005    0.885    0.885 <string>:1(<module>)
    1025    0.001    0.000    0.001    0.000 {built-in method _codecs.utf_8_decode}
    1025    0.001    0.000    0.001    0.000 codecs.py:319(decode)
      1    0.000    0.000    0.000    0.000 {built-in method builtins.print}
      1    0.000    0.000    0.885    0.885 {built-in method builtins.exec}
      2    0.000    0.000    0.000    0.000 mathlib.py:60(div)
      3    0.000    0.000    0.000    0.000 {built-in method builtins.len}
      1    0.000    0.000    0.000    0.000 mathlib.py:83(root)
      1    0.000    0.000    0.000    0.000 {method 'disable' of '_lsprof.Profiler' objects}
      2    0.000    0.000    0.000    0.000 mathlib.py:73(mul)
      2    0.000    0.000    0.000    0.000 mathlib.py:50(sub)

```

Obrázek 4: milion vstupů

```

2886.70720951306
    225096270 function calls in 43.853 seconds

Ordered by: internal time

ncalls  tottime  percall  cumtime  percall filename:lineno(function)
      1   23.901   23.901   43.630   43.630 profiling.py:9(profiler_function)
50000001    8.737    0.000   12.650    0.000 mathlib.py:108(pow)
100000000    5.199    0.000    5.199    0.000 mathlib.py:40(add)
24993934    3.912    0.000    3.912    0.000 mathlib.py:23(contains_decimals)
50000000    1.823    0.000    1.823    0.000 {method 'append' of 'list' objects}
      1    0.223    0.223   43.853   43.853 <string>:1(<module>)
    51160    0.032    0.000    0.032    0.000 {built-in method _codecs.utf_8_decode}
    51160    0.024    0.000    0.056    0.000 codecs.py:319(decode)
      1    0.000    0.000    0.000    0.000 {built-in method builtins.print}
      1    0.000    0.000   43.853   43.853 {built-in method builtins.exec}
      2    0.000    0.000    0.000    0.000 mathlib.py:60(div)
      3    0.000    0.000    0.000    0.000 {built-in method builtins.len}
      2    0.000    0.000    0.000    0.000 mathlib.py:50(sub)
      1    0.000    0.000    0.000    0.000 {method 'disable' of '_lsprof.Profiler' objects}
      2    0.000    0.000    0.000    0.000 mathlib.py:73(mul)
      1    0.000    0.000    0.000    0.000 mathlib.py:83(root)

```

Obrázek 5: 50 milionů vstupů

```

2886.6553368060668
    315142047 function calls in 63.564 seconds

Ordered by: internal time

ncalls  tottime  percall  cumtime  percall filename:lineno(function)
      1   34.852   34.852   63.253   63.253 profiling.py:9(profiler_function)
70000001   12.570    0.000   18.273    0.000 mathlib.py:108(pow)
140000000    7.451    0.000    7.451    0.000 mathlib.py:40(add)
34998785    5.703    0.000    5.703    0.000 mathlib.py:23(contains_decimals)
70000000    2.592    0.000    2.592    0.000 {method 'append' of 'list' objects}
      1    0.312    0.312   63.564   63.564 <string>:1(<module>)
    71623    0.049    0.000    0.049    0.000 {built-in method _codecs.utf_8_decode}
    71623    0.036    0.000    0.085    0.000 codecs.py:319(decode)
      1    0.000    0.000    0.000    0.000 {built-in method builtins.print}
      1    0.000    0.000   63.564   63.564 {built-in method builtins.exec}
      3    0.000    0.000    0.000    0.000 {built-in method builtins.len}
      2    0.000    0.000    0.000    0.000 mathlib.py:60(div)
      1    0.000    0.000    0.000    0.000 {method 'disable' of '_lsprof.Profiler' objects}
      2    0.000    0.000    0.000    0.000 mathlib.py:73(mul)
      2    0.000    0.000    0.000    0.000 mathlib.py:50(sub)
      1    0.000    0.000    0.000    0.000 mathlib.py:83(root)

```

Obrázek 6: 70 milionů vstupů