IVS – Praktické aspekty vývoje software

Sudo_Apt_Install_Calculator

Profiling – správa

Autori:

Filip Figúr (xfigurf00)

Miroslav Kučík (xkucikm00)

Kristián Rúček (xrucekk00)

Rastislav Šerý (xseryra00)

18.4.2025

Úvod

Účelom tohto dokumentu je analyzovať výstup profileru pri profilovaní programu určeného na výpočet smerodajnej odchýlky s využitím funkcií z math_library.py a navrhnúť možný postup pri ďalšej optimalizácií programu.

Na profiling bol použitý nástroj *cProfile*. Na generovanie vstupných súborov s postupnosťou čísel bol použitý skript v jazyku *Python*. Číselné hodnoty sú náhodné celé čísla z intervalu <100;1 000 000>.

Profiling bol vykonaný v testovacom prostredí s nasledujúcimi špecifikáciami:

- **OS**: Ubuntu 24.04.2 LTS (WSL 2)
- **Platforma**: Windows Subsystem for Linux (WSL 2)
- **CPU**: AMD Ryzen 5 7535HS, 6 cores, 3,30 GHz (až 4.75 GHz)
- **RAM**: 16GB

Všetky testovacie súbory nájdete v zložke */profiling/Test_files*. Konkrétne výpisy profileru pre jednotlivé testovacie súbory nájdete na konci tohto dokumentu.

Výstupy

Na testovanie bolo vytvorených 7 testovacích súborov s postupnosťami 10, 100, 1 000, 10 000, 100 000, 1 000 000 a 10 000 000 číselných hodnôt oddelených bielym znakom.

Funkcie *data_extractor()*, *calculate_average()*, *formula_core()* a *finito()* nie sú súčasťou knižnice math_library.py a boli vytvorené za účelom výpočtu smerodajnej odchýlky.

Pri 10 a 100 hodnotách bol nameraný čas príliš malý na to, aby sa dalo určiť, kde by sa mal program optimalizovať. Pri 1000 hodnotách celkový beh programu bol 0,003 sekundy, čo je ale stále veľmi malý čas. Pre 10 000 hodnôt bol nameraný čas 0,027 sekundy, stále pomerne rýchle.

Pri veľkom objeme dát (100 000 a 1 000 000 hodnôt) je už ale čas 0,214 a niečo nad 2 sekundy. Keď na vstup dáme 10 000 000 hodnôt (nad rámec zadania), program je veľmi pomalý (výpočet trvá vyše 18 sekúnd).

Z výstupov je možné si všimnúť, že najviac volanými funkciami sú **add()**, **sub()**, a **power()** z našej vlastnej matematickej knižnice.

Počet volaní pre N prvkov:

- add() = 2*N krát,
- power() = N+1 krát
- sub() = N krát
- multiply(), divide(), n_root() = 1 krát

Pri ďalšej optimalizácií výpočtu by sme sa zamerali hlavne na zrýchlenie funkcií add(), power() a sub() .

```
      92 function calls in 0.000 seconds

Ordered by: standard name

ncalls  tottime  percall  cumtime  percall filename:lineno(function)
     3    0.000    0.000    0.000    0.000 <frozen codecs>:319(decode)
     1    0.000    0.000    0.000    0.000 <string>:1(<module>)
    20    0.000    0.000    0.000    0.000 math_library.py:17(add)
    10    0.000    0.000    0.000    0.000 math_library.py:24(sub)
     1    0.000    0.000    0.000    0.000 math_library.py:31(multiply)
     1    0.000    0.000    0.000    0.000 math_library.py:38(divide)
    11    0.000    0.000    0.000    0.000 math_library.py:45(power)
     1    0.000    0.000    0.000    0.000 math_library.py:53(n_root)
     1    0.000    0.000    0.000    0.000 profiling.py:14(data_extractor)
     1    0.000    0.000    0.000    0.000 profiling.py:24(calculate_average)
     1    0.000    0.000    0.000    0.000 profiling.py:31(formula_core)
     1    0.000    0.000    0.000    0.000 profiling.py:40(finito)
     1    0.000    0.000    0.000    0.000 profiling.py:46(main)
     3    0.000    0.000    0.000    0.000 {built-in method _codecs.utf_8_decode}
     1    0.000    0.000    0.000    0.000 {built-in method builtins.exec}
     1    0.000    0.000    0.000    0.000 {built-in method builtins.len}
     1    0.000    0.000    0.000    0.000 {built-in method builtins.print}
     1    0.000    0.000    0.000    0.000 {built-in method builtins.round}
     1    0.000    0.000    0.000    0.000 {built-in method math.isclose}
    10    0.000    0.000    0.000    0.000 {method 'append' of 'list' objects}
     1    0.000    0.000    0.000    0.000 {method 'disable' of '_lsprof.Profiler' objects}
    10    0.000    0.000    0.000    0.000 {method 'split' of 'str' objects}
    10    0.000    0.000    0.000    0.000 {method 'strip' of 'str' objects}
```

Obrázok 1 - Výstup pre 10 čísel

```
      722 function calls in 0.000 seconds

Ordered by: standard name

ncalls  tottime  percall  cumtime  percall filename:lineno(function)
     3    0.000    0.000    0.000    0.000 <frozen codecs>:319(decode)
     1    0.000    0.000    0.000    0.000 <string>:1(<module>)
   200    0.000    0.000    0.000    0.000 math_library.py:17(add)
   100    0.000    0.000    0.000    0.000 math_library.py:24(sub)
     1    0.000    0.000    0.000    0.000 math_library.py:31(multiply)
     1    0.000    0.000    0.000    0.000 math_library.py:38(divide)
   101    0.000    0.000    0.000    0.000 math_library.py:45(power)
     1    0.000    0.000    0.000    0.000 math_library.py:53(n_root)
     1    0.000    0.000    0.000    0.000 profiling.py:14(data_extractor)
     1    0.000    0.000    0.000    0.000 profiling.py:24(calculate_average)
     1    0.000    0.000    0.000    0.000 profiling.py:31(formula_core)
     1    0.000    0.000    0.000    0.000 profiling.py:40(finito)
     1    0.000    0.000    0.000    0.000 profiling.py:46(main)
     3    0.000    0.000    0.000    0.000 {built-in method _codecs.utf_8_decode}
     1    0.000    0.000    0.000    0.000 {built-in method builtins.exec}
     1    0.000    0.000    0.000    0.000 {built-in method builtins.len}
     1    0.000    0.000    0.000    0.000 {built-in method builtins.print}
     1    0.000    0.000    0.000    0.000 {built-in method builtins.round}
     1    0.000    0.000    0.000    0.000 {built-in method math.isclose}
   100    0.000    0.000    0.000    0.000 {method 'append' of 'list' objects}
     1    0.000    0.000    0.000    0.000 {method 'disable' of '_lsprof.Profiler' objects}
   100    0.000    0.000    0.000    0.000 {method 'split' of 'str' objects}
   100    0.000    0.000    0.000    0.000 {method 'strip' of 'str' objects}
```

Obrázok 2 - Výstup pre 100 čísel

```
      7022 function calls in 0.003 seconds

Ordered by: standard name

ncalls  tottime  percall  cumtime  percall filename:lineno(function)
     3    0.000    0.000    0.000    0.000 <frozen codecs>:319(decode)
     1    0.000    0.000    0.003    0.003 <string>:1(<module>)
  2000    0.000    0.000    0.000    0.000 math_library.py:17(add)
  1000    0.000    0.000    0.000    0.000 math_library.py:24(sub)
     1    0.000    0.000    0.000    0.000 math_library.py:31(multiply)
     1    0.000    0.000    0.000    0.000 math_library.py:38(divide)
  1001    0.000    0.000    0.000    0.000 math_library.py:45(power)
     1    0.000    0.000    0.000    0.000 math_library.py:53(n_root)
     1    0.002    0.002    0.002    0.002 profiling.py:14(data_extractor)
     1    0.000    0.000    0.000    0.000 profiling.py:24(calculate_average)
     1    0.000    0.000    0.001    0.001 profiling.py:31(formula_core)
     1    0.000    0.000    0.000    0.000 profiling.py:40(finito)
     1    0.000    0.000    0.003    0.003 profiling.py:46(main)
     3    0.000    0.000    0.000    0.000 {built-in method _codecs.utf_8_decode}
     1    0.000    0.000    0.003    0.003 {built-in method builtins.exec}
     1    0.000    0.000    0.000    0.000 {built-in method builtins.len}
     1    0.000    0.000    0.000    0.000 {built-in method builtins.print}
     1    0.000    0.000    0.000    0.000 {built-in method builtins.round}
     1    0.000    0.000    0.000    0.000 {built-in method math.isclose}
  1000    0.000    0.000    0.000    0.000 {method 'append' of 'list' objects}
     1    0.000    0.000    0.000    0.000 {method 'disable' of '_lsprof.Profiler' objects}
  1000    0.000    0.000    0.000    0.000 {method 'split' of 'str' objects}
  1000    0.000    0.000    0.000    0.000 {method 'strip' of 'str' objects}
```

Obrázok 3 - Výstup pre 1000 čísel

```
      70036 function calls in 0.027 seconds

Ordered by: standard name

ncalls  tottime  percall  cumtime  percall filename:lineno(function)
    10    0.000    0.000    0.000    0.000 <frozen codecs>:319(decode)
     1    0.000    0.000    0.027    0.027 <string>:1(<module>)
 20000    0.002    0.000    0.002    0.000 math_library.py:17(add)
 10000    0.001    0.000    0.001    0.000 math_library.py:24(sub)
     1    0.000    0.000    0.000    0.000 math_library.py:31(multiply)
     1    0.000    0.000    0.000    0.000 math_library.py:38(divide)
 10001    0.001    0.000    0.001    0.000 math_library.py:45(power)
     1    0.000    0.000    0.000    0.000 math_library.py:53(n_root)
     1    0.011    0.011    0.017    0.017 profiling.py:14(data_extractor)
     1    0.002    0.002    0.003    0.003 profiling.py:24(calculate_average)
     1    0.005    0.005    0.008    0.008 profiling.py:31(formula_core)
     1    0.000    0.000    0.000    0.000 profiling.py:40(finito)
     1    0.000    0.000    0.027    0.027 profiling.py:46(main)
    10    0.000    0.000    0.000    0.000 {built-in method _codecs.utf_8_decode}
     1    0.000    0.000    0.027    0.027 {built-in method builtins.exec}
     1    0.000    0.000    0.000    0.000 {built-in method builtins.len}
     1    0.000    0.000    0.000    0.000 {built-in method builtins.print}
     1    0.000    0.000    0.000    0.000 {built-in method builtins.round}
     1    0.000    0.000    0.000    0.000 {built-in method math.isclose}
 10000    0.002    0.000    0.002    0.000 {method 'append' of 'list' objects}
     1    0.000    0.000    0.000    0.000 {method 'disable' of '_lsprof.Profiler' objects}
 10000    0.002    0.000    0.002    0.000 {method 'split' of 'str' objects}
 10000    0.002    0.000    0.002    0.000 {method 'strip' of 'str' objects}
```

Obrázok 4 - Výstup pre 10 000 čísel

```
      700190 function calls in 0.214 seconds

Ordered by: standard name

ncalls  tottime  percall  cumtime  percall filename:lineno(function)
    87    0.000    0.000    0.000    0.000 <frozen codecs>:319(decode)
     1    0.001    0.001    0.214    0.214 <string>:1(<module>)
200000    0.015    0.000    0.015    0.000 math_library.py:17(add)
100000    0.010    0.000    0.010    0.000 math_library.py:24(sub)
     1    0.000    0.000    0.000    0.000 math_library.py:31(multiply)
     1    0.000    0.000    0.000    0.000 math_library.py:38(divide)
100001    0.009    0.000    0.009    0.000 math_library.py:45(power)
     1    0.000    0.000    0.000    0.000 math_library.py:53(n_root)
     1    0.074    0.074    0.122    0.122 profiling.py:14(data_extractor)
     1    0.013    0.013    0.021    0.021 profiling.py:24(calculate_average)
     1    0.045    0.045    0.071    0.071 profiling.py:31(formula_core)
     1    0.000    0.000    0.000    0.000 profiling.py:40(finito)
     1    0.000    0.000    0.213    0.213 profiling.py:46(main)
    87    0.000    0.000    0.000    0.000 {built-in method _codecs.utf_8_decode}
     1    0.000    0.000    0.214    0.214 {built-in method builtins.exec}
     1    0.000    0.000    0.000    0.000 {built-in method builtins.len}
     1    0.000    0.000    0.000    0.000 {built-in method builtins.print}
     1    0.000    0.000    0.000    0.000 {built-in method builtins.round}
     1    0.000    0.000    0.000    0.000 {built-in method math.isclose}
100000    0.013    0.000    0.013    0.000 {method 'append' of 'list' objects}
     1    0.000    0.000    0.000    0.000 {method 'disable' of '_lsprof.Profiler' objects}
100000    0.021    0.000    0.021    0.000 {method 'split' of 'str' objects}
100000    0.013    0.000    0.013    0.000 {method 'strip' of 'str' objects}
```

Figure 5 - Výstup pre 100 000 čísel

```
      7001700 function calls in 2.057 seconds

 Ordered by: standard name

 ncalls  tottime  percall  cumtime  percall filename:lineno(function)
    842    0.002    0.000    0.003    0.000 <frozen codecs>:319(decode)
      1    0.008    0.008    2.057    2.057 <string>:1(<module>)
2000000    0.141    0.000    0.141    0.000 math_library.py:17(add)
1000000    0.099    0.000    0.099    0.000 math_library.py:24(sub)
      1    0.000    0.000    0.000    0.000 math_library.py:31(multiply)
      1    0.000    0.000    0.000    0.000 math_library.py:38(divide)
1000001    0.088    0.000    0.088    0.000 math_library.py:45(power)
      1    0.000    0.000    0.000    0.000 math_library.py:53(n_root)
      1    0.699    0.699    1.151    1.151 profiling.py:14(data_extractor)
      1    0.131    0.131    0.205    0.205 profiling.py:24(calculate_average)
      1    0.439    0.439    0.694    0.694 profiling.py:31(formula_core)
      1    0.000    0.000    0.000    0.000 profiling.py:40(finito)
      1    0.000    0.000    2.049    2.049 profiling.py:46(main)
    842    0.002    0.000    0.002    0.000 {built-in method _codecs.utf_8_decode}
      1    0.000    0.000    2.057    2.057 {built-in method builtins.exec}
      1    0.000    0.000    0.000    0.000 {built-in method builtins.len}
      1    0.000    0.000    0.000    0.000 {built-in method builtins.print}
      1    0.000    0.000    0.000    0.000 {built-in method builtins.round}
      1    0.000    0.000    0.000    0.000 {built-in method math.isclose}
1000000    0.125    0.000    0.125    0.000 {method 'append' of 'list' objects}
      1    0.000    0.000    0.000    0.000 {method 'disable' of '_lsprof.Profiler' objects}
1000000    0.202    0.000    0.202    0.000 {method 'split' of 'str' objects}
1000000    0.122    0.000    0.122    0.000 {method 'strip' of 'str' objects}
```

Obrázok 6 - Výstup pre 1 000 000 čísel

```
        70016838 function calls in 18.145 seconds

   Ordered by: standard name

   ncalls  tottime  percall  cumtime  percall filename:lineno(function)
     8411    0.013    0.000    0.026    0.000 <frozen codecs>:319(decode)
        1    0.095    0.095   18.145   18.145 <string>:1(<module>)
 20000000    1.303    0.000    1.303    0.000 math_library.py:17(add)
 10000000    0.916    0.000    0.916    0.000 math_library.py:24(sub)
        1    0.000    0.000    0.000    0.000 math_library.py:31(multiply)
        1    0.000    0.000    0.000    0.000 math_library.py:38(divide)
 10000001    0.814    0.000    0.814    0.000 math_library.py:45(power)
        1    0.000    0.000    0.003    0.003 math_library.py:53(n_root)
        1    6.012    6.012    9.819    9.819 profiling.py:14(data_extractor)
        1    1.211    1.211    1.865    1.865 profiling.py:24(calculate_average)
        1    3.982    3.982    6.362    6.362 profiling.py:31(formula_core)
        1    0.000    0.000    0.003    0.003 profiling.py:40(finito)
        1    0.001    0.001   18.050   18.050 profiling.py:46(main)
     8411    0.013    0.000    0.013    0.000 {built-in method _codecs.utf_8_decode}
        1    0.000    0.000   18.145   18.145 {built-in method builtins.exec}
        1    0.000    0.000    0.000    0.000 {built-in method builtins.len}
        1    0.000    0.000    0.000    0.000 {built-in method builtins.print}
        1    0.000    0.000    0.000    0.000 {built-in method builtins.round}
        1    0.003    0.003    0.003    0.003 {built-in method math.isclose}
 10000000    1.025    0.000    1.025    0.000 {method 'append' of 'list' objects}
        1    0.000    0.000    0.000    0.000 {method 'disable' of '_lsprof.Profiler' objects}
 10000000    1.700    0.000    1.700    0.000 {method 'split' of 'str' objects}
 10000000    1.057    0.000    1.057    0.000 {method 'strip' of 'str' objects}
```

Obrázok 7 - Výstup pre 10 000 000 čísel