

# Profiling - IVS Projekt 2

xzvara01

24. marca 2021

## 1 Zadanie

Pomocou funkcií z našej matematickej knižnice sme mali vytvoriť program pre výpočet **výberovej smerodajnej odchýlky** z postupnosti čísel, ktorú náš program načítava zo štandardného vstupu. Pre výpočet odchýlky sme použili vzorec

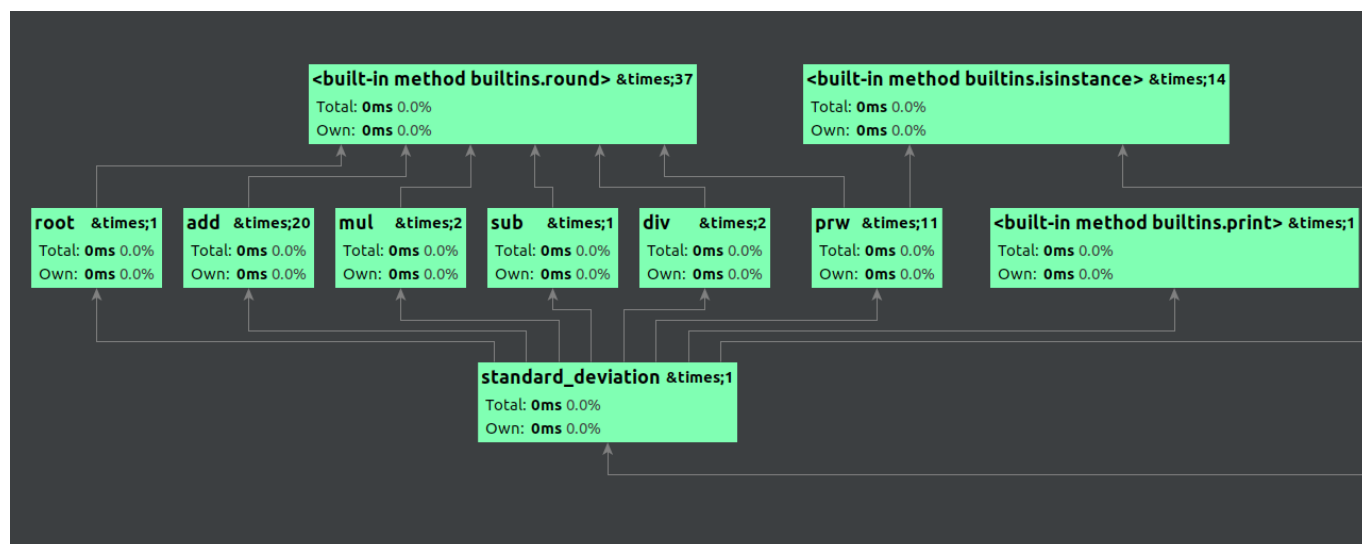
$$s = \sqrt{\frac{1}{N-1} \left( \sum_{i=1}^N x_i^2 - N\bar{x}^2 \right)}$$

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

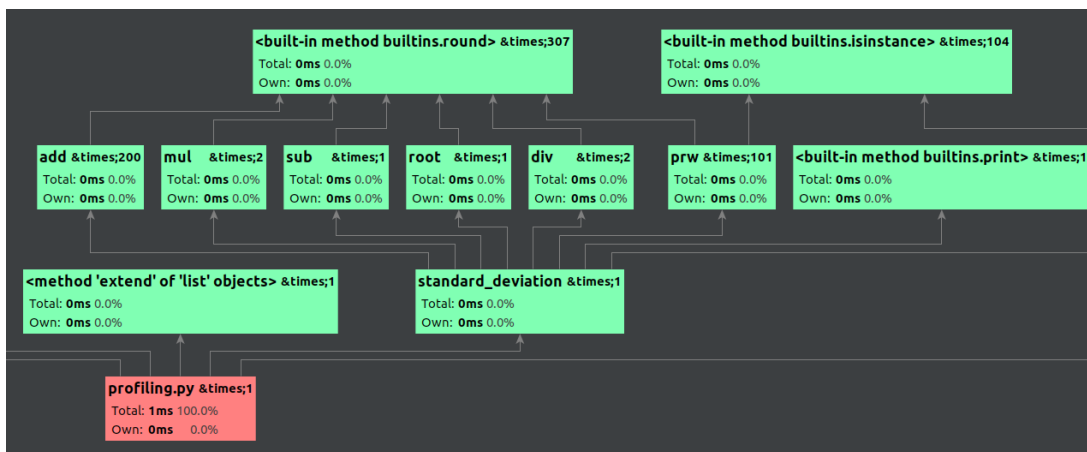
## 2 Výsledky

Výsledky profileru sú v priečinku profiling. Snímky grafu volania funkcií (call graph) sú v oddelenom priečinku image\_out. Profiling prebehol v prostredí PyCharm (cProfile) s verziou Pythonu 3.8.5. Merania boli uskutočnené štyrikrát - s rôznym počtom vstupných čísel, od 10 až po 10 000.

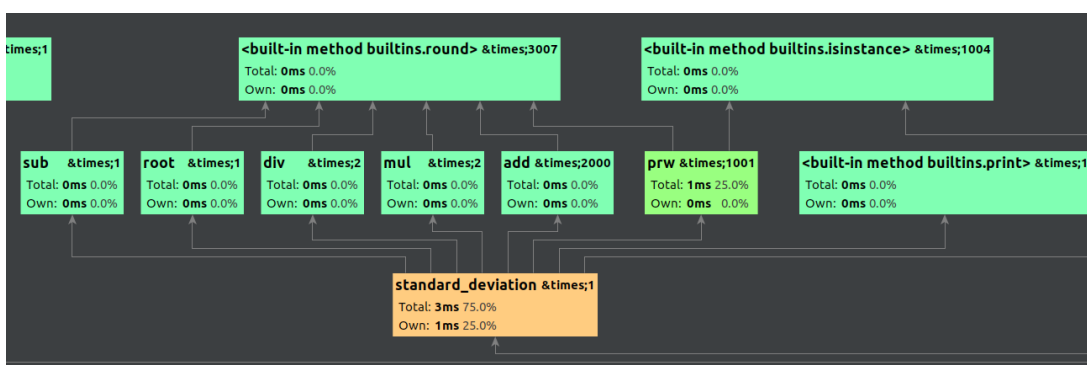
Na ďalších obrázkoch sú výstupy profilingu pre 10, 100 a 1000 vstupných čísel. Je vidieť, že program najviac využíva funkcie **add**, **prw** a **round**.



Obr. 1: Vstup s 10 číslami

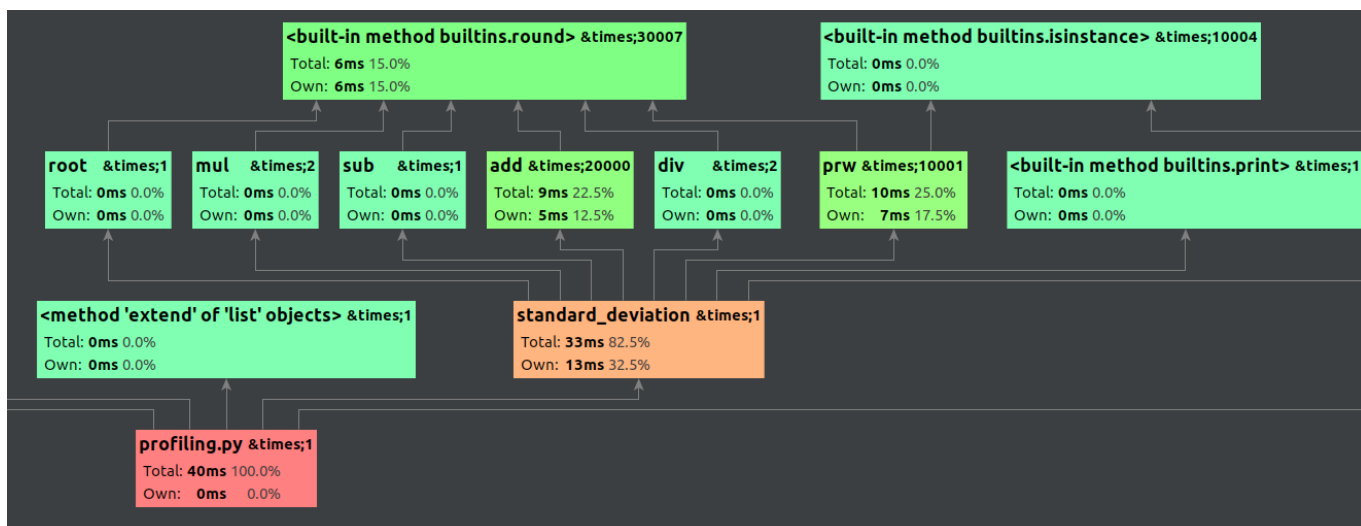


Obr. 2: Vstup s 100 číslami



Obr. 3: Vstup s 1000 číslami

Počet volaní funkcií `add`, `prw` a `round` rastie lineárne s počtom vstupných čísel. Konkrétne sa pre  $N$  čísel zavolajú funkcie **round**  $3N + 7$ -krát, **add**  $2N$ -krát a **prw**  $N + 1$ -krát. Teda pri optimalizáciách by som sa zameral na tieto funkcie, keďže sú používané najčastejšie. Ďalší nápad, ako optimalizovať tento program by bolo prenechať zaokrúhľovanie operácií implicitne na jazyku Python, prípadne na užívateľa. Tým pádom by sa nemusela volať funkcia `round`. Najlepšie je to vidieť v prípade, kedy to bolo testované pre 10000 čísel.



Obr. 4: Vstup s 10000 číslami