

Calculandum Machina - profiling

Řešitelé:

Ondřej Lukášek (xlukas15)
Ondřej Koumar (xkouma02)
Jonáš Morkus (xmorku03)
Marek Konečný (xkonec85)

Datum: 25.04.2022

Generace a zobrazení výsledků profilingu:

Použili jsme nástroj Callgrind, který je součástí nástroje Valgrind. Pro grafické zobrazení výsledků jsme použili program Kcachegrind.

Pro překlad lze použít příkaz: `g++ -g stddeviation.cpp ./src/matlib.cpp -o stddeviation`

Příkaz na spuštění Callgrindu: `valgrind --tool==callgrind ./stddeviation cisla.txt`
Příkaz na spuštění Kcachegrindu: `kcachegrind callgrind.out.pidprocesu`

Vizualizaci výsledků (grafy atd.) můžete najít ve složce profiling:

pro soubor s 10 čísly – `vystup10.pdf` pro

soubor s 100 čísly – `vystup100.pdf`

pro soubor s 1000 čísly – `vystup1000.pdf`

pro představu je tu přepsána část výstupu s 1000 čísly

%	called	function
0.86	1000	power(double, unsigned short)
0.54	2000	add(double, double)
0.27	1001	sub(double, double)
0.03	1	root(double, unsigned short)
0.00	2	div(double, double)
0.00	1	mul(double, double)

Závěr:

V programu Kcachegrind můžeme vidět, že funkce **div** a **mul** nezabírají téměř žádný čas.

Nejnáročnější se zdá být funkce **power** – jelikož vždy voláme funkci s argumenty **power(number,2)**, můžeme jako vhodnou optimalizaci nahradit použitím mocninné funkce například `takto – number * number`.

Funkce **power** vykoná 3x více instrukcí než **mul**, proto by to bylo řešení jak ušetřit čas.

Kolikrát a v jaké části se volá jaká funkce můžete najít ve složce profiling, v příložených souborech s výstupy z Kcachegrind, jejichž součástí je fotografie, na které je vidět kde a kolikrát se daná funkce volá.

Například funkce **add**, **power** nebo **sub** je volaná pro všechny čísla (funkce **add** je použita pro výpočet součtu všech čísel).