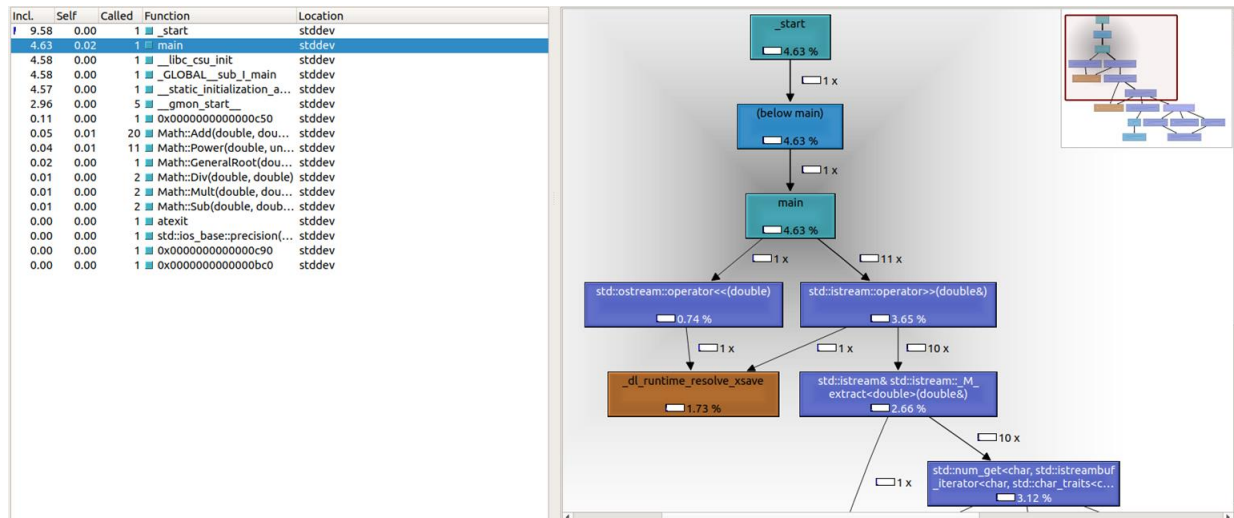
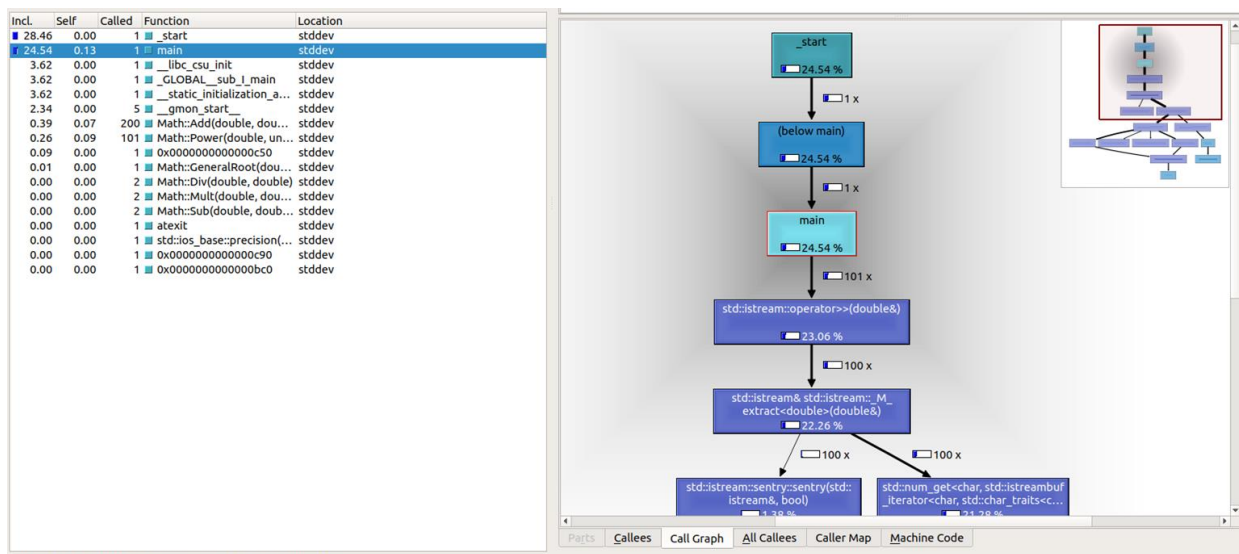


Proffiling

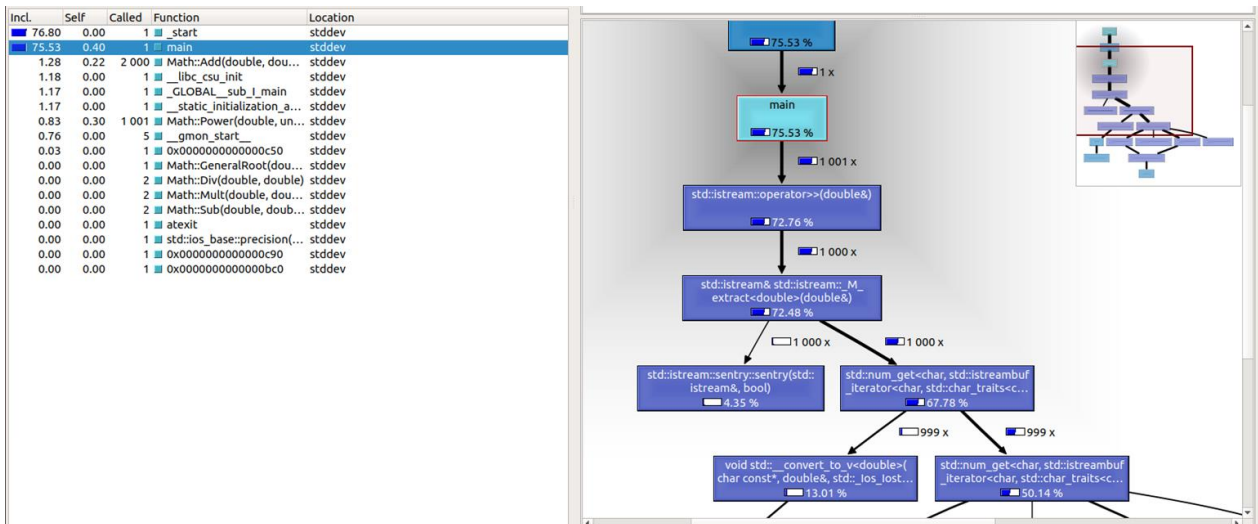
Za účelem proffilingu byl vytvořen program, který počítá výběrovou směrodatnou odchylku. V implementaci tohoto programu je použita námi vytvořená matematická knihovna. Profilování bylo provedeno třikrát a to tak, že program byl spuštěn na 10, 100, 1000 náhodných číslech generovaných v rozsahu $(-100, 1000, 5)$. Proffiling byl vytvořen pomocí programu *gprof* a *callgrind*.



Obr. 1 Výstup Callgrindu pro vstup 10 čísel

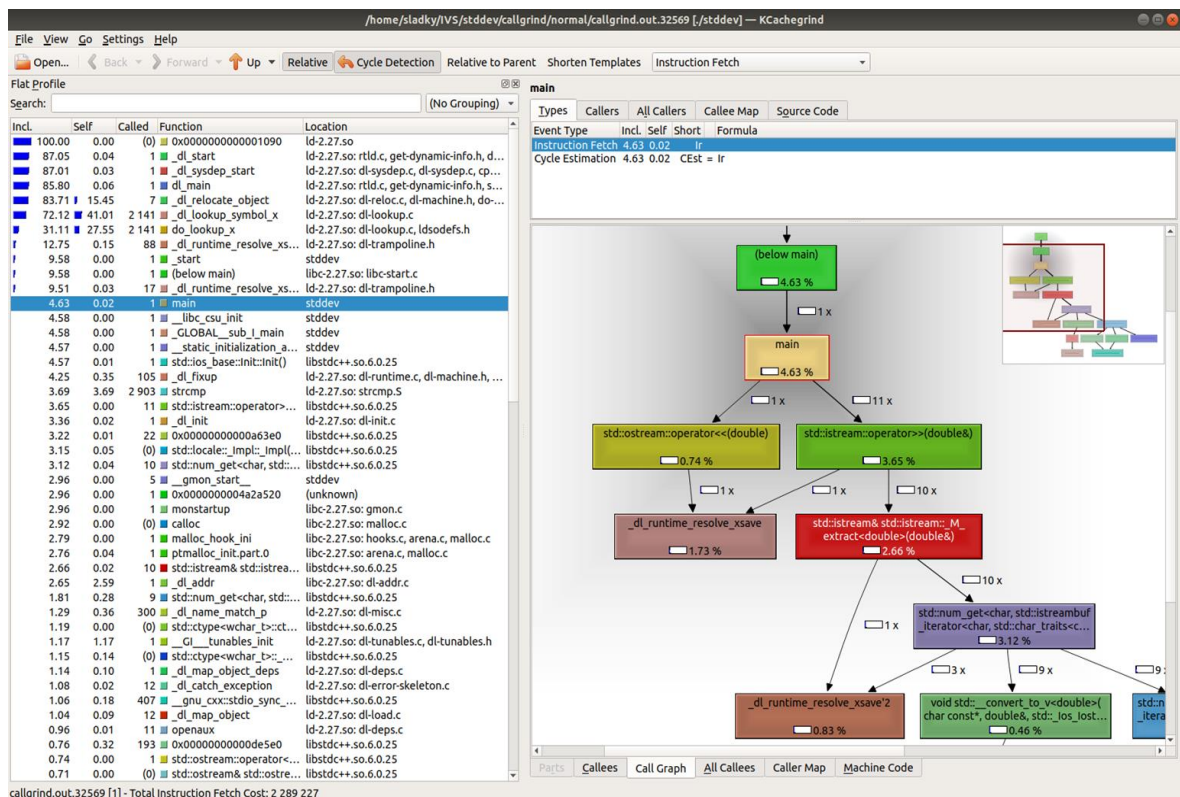


Obr. 2 Výstup Callgrindu pro vstup 100 čísel

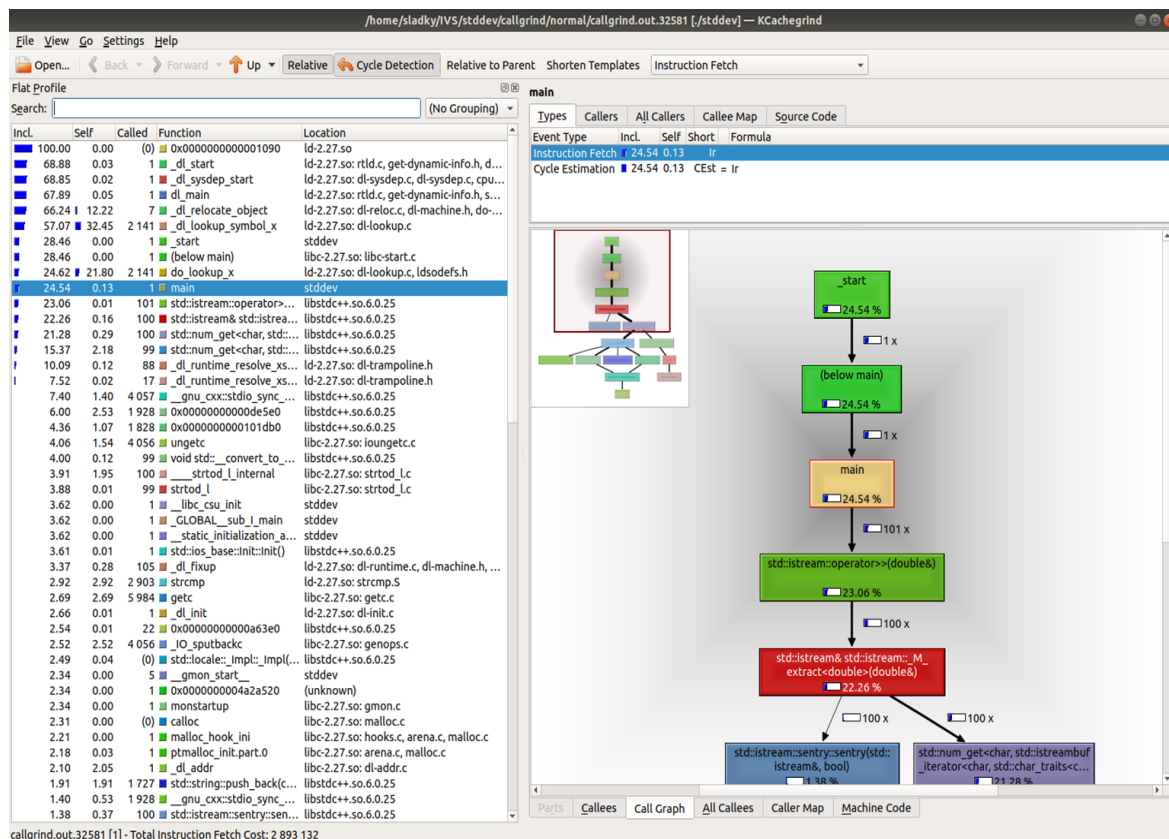


Obr. 3 Výstup Callgrindu pro vstup 1000 čísel

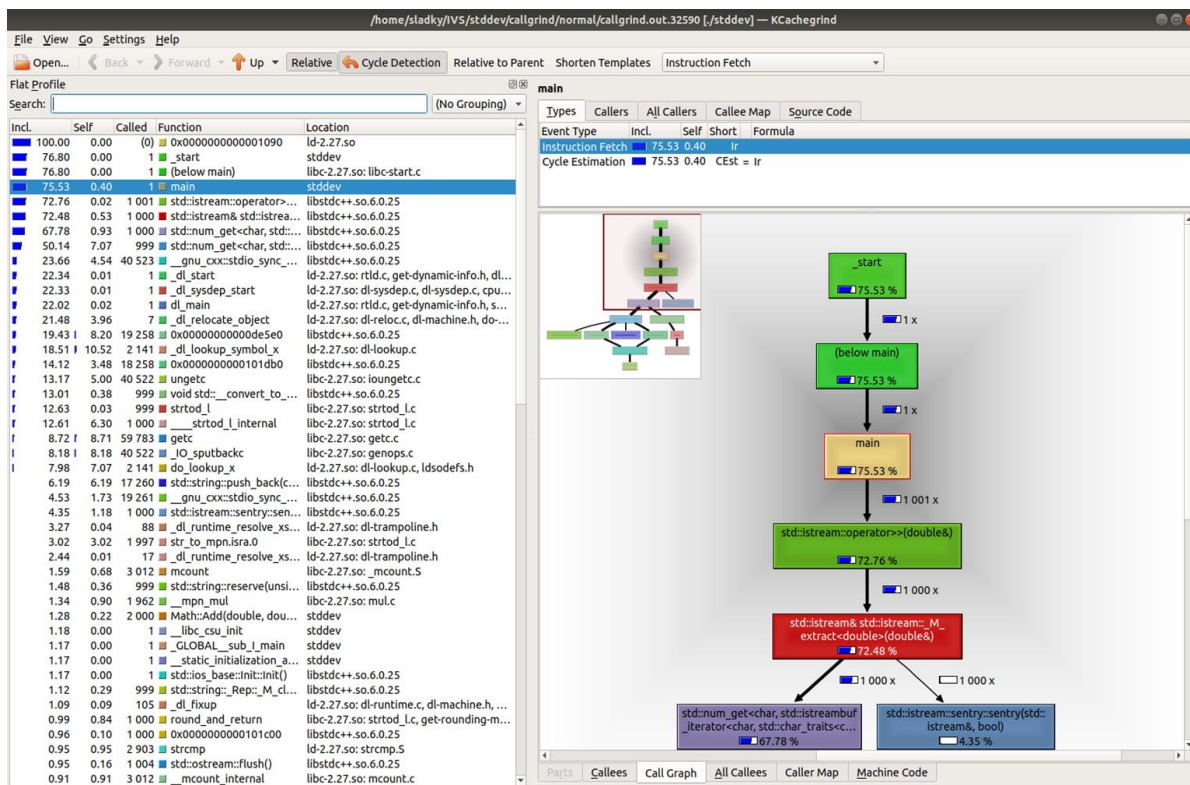
Z výstupu Callgrindu vidíme očekávané chování. S přibývajícím počtem vstupů přibývá pouze počet volání funkce `Math::Add` a `Math::Power`. Toto chování je předvídatelné podle daného algoritmu na výběrovou směrodatnou odchylku a žádné nepředvídatelné chování programu zde není. I s 1000 vstupními čísly však tyto funkce stále běží poměrně málo s ohledem na celkový čas programu. Srovnání s celkovým časem lze porovnat na obrázcích níže:



Obr. 4 Výstup Callgrindu pro vstup 10 čísel



Obr. 5 Výstup Callgrindu pro vstup 100 čísel



Obr. 6 Výstup Callgrindu pro vstup 1000 čísel

Závěr:

Z přiložených výstupů z Callgrindu lze videt, že snažit se optimalizovat náš kód by bylo zbytečné protože nezabírá mnoho času. Například při 10 a 100 číslech na vstupu v main funkci a jejich potomcích program strávil méně než 50% celého běhu. U 1000 čísel už program strávil v main funkci 75%, ale většinu času zabírají systemové knihovny, jako je inicializace a čtení z stdin. Proto při optimalizaci by jsme se měli zaměřit na tyto problémy.

Ve složce profiling přikládáme ještě další soubory z výstupu Callgrindu a z gprof.