Vysoké učení technické v Brně
Fakulta informačních technologií
Praktické aspekty vývoje software
2018 / 2019

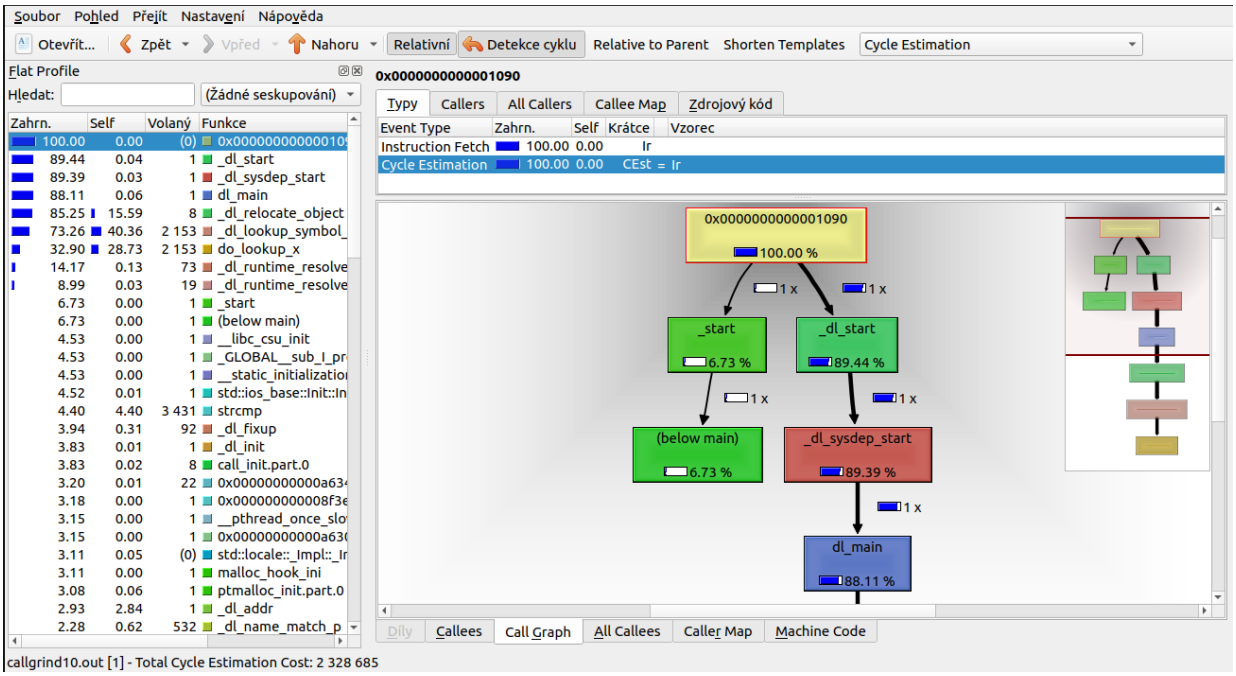
Profilování

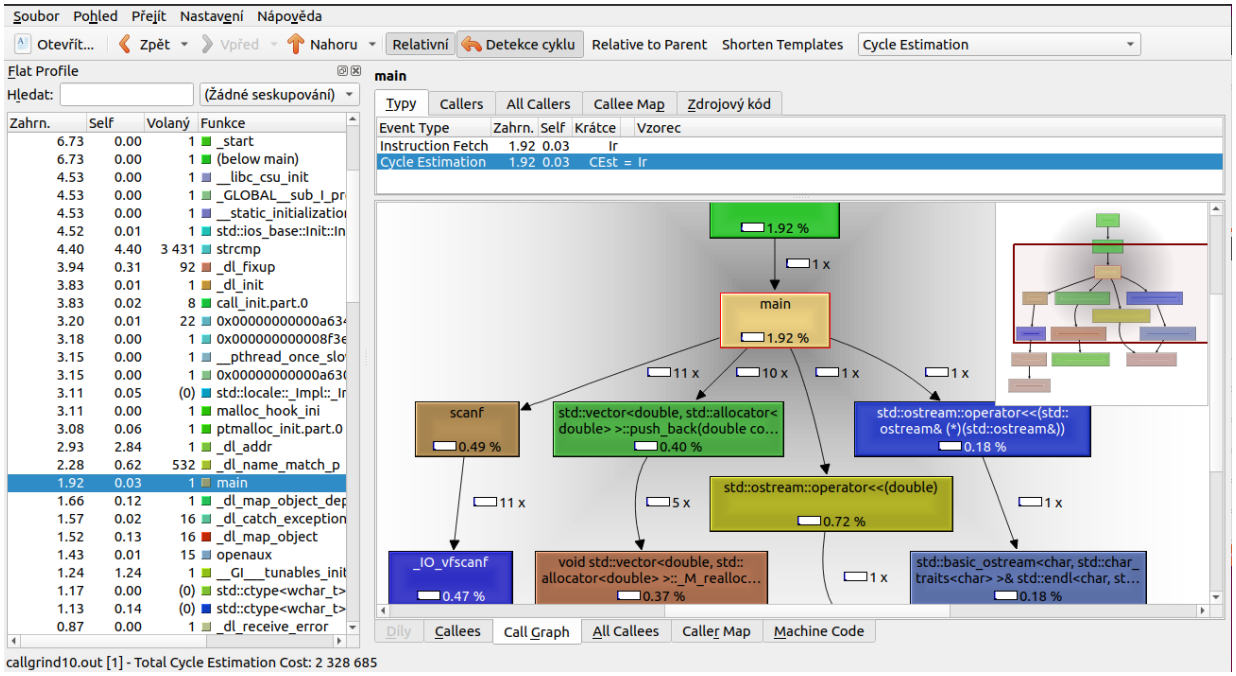
Radim Lipka xlipka02@stud.fit.vutbr.cz
Roman Ondráček xondra58@stud.fit.vutbr.cz
Pavel Raur xraurp00@stud.fit.vutbr.cz
David Reinhart xreinh00@stud.fit.vutbr.cz

1 Výstupy z profilování

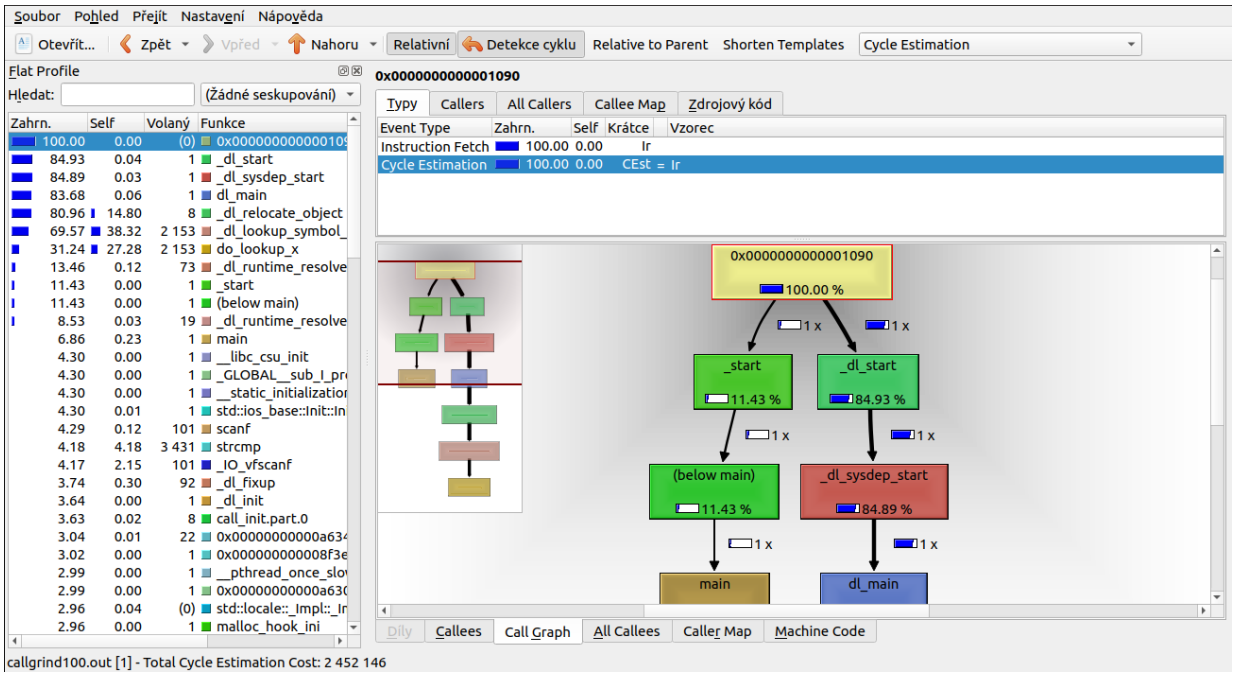
Pro profilování jsme použili program valgrind a pro zobrazení výsledků jsme použili program KCachegrind.
První obrázek pro každou hodnotu vstupu je graf volání z rootu, druhý je pak graf volání z mainu programu.

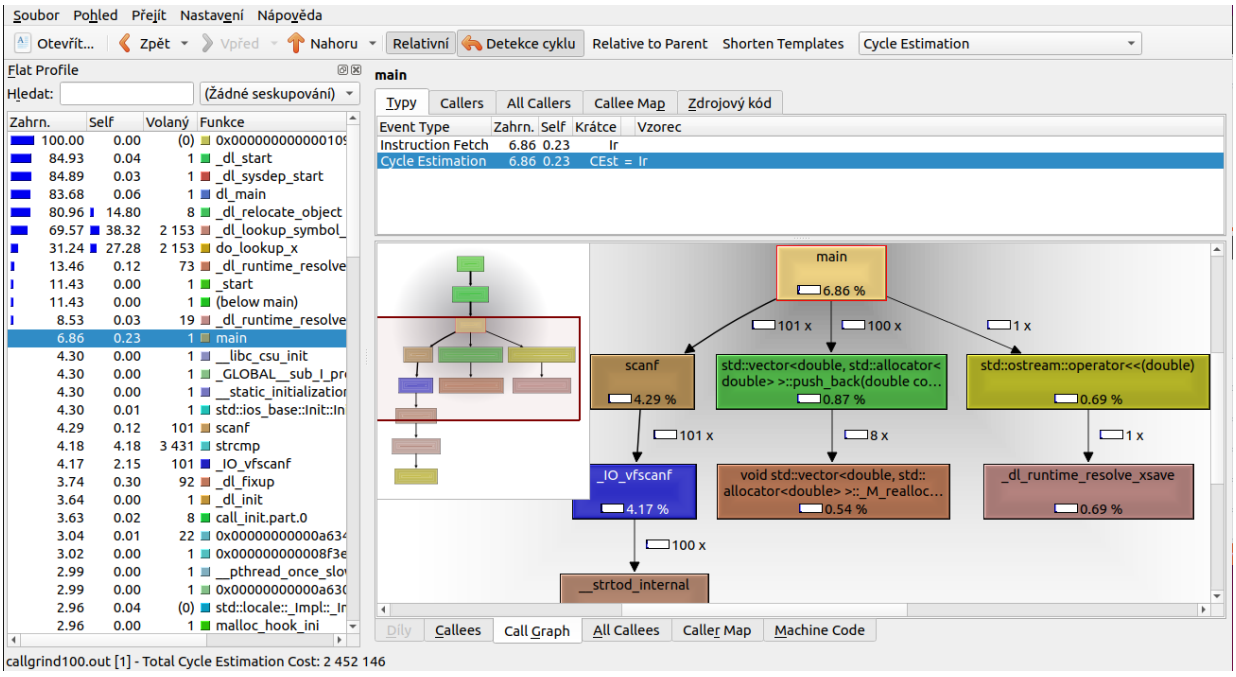
1.1 Grafy volání při spuštění programu s deseti čísly na vstupu



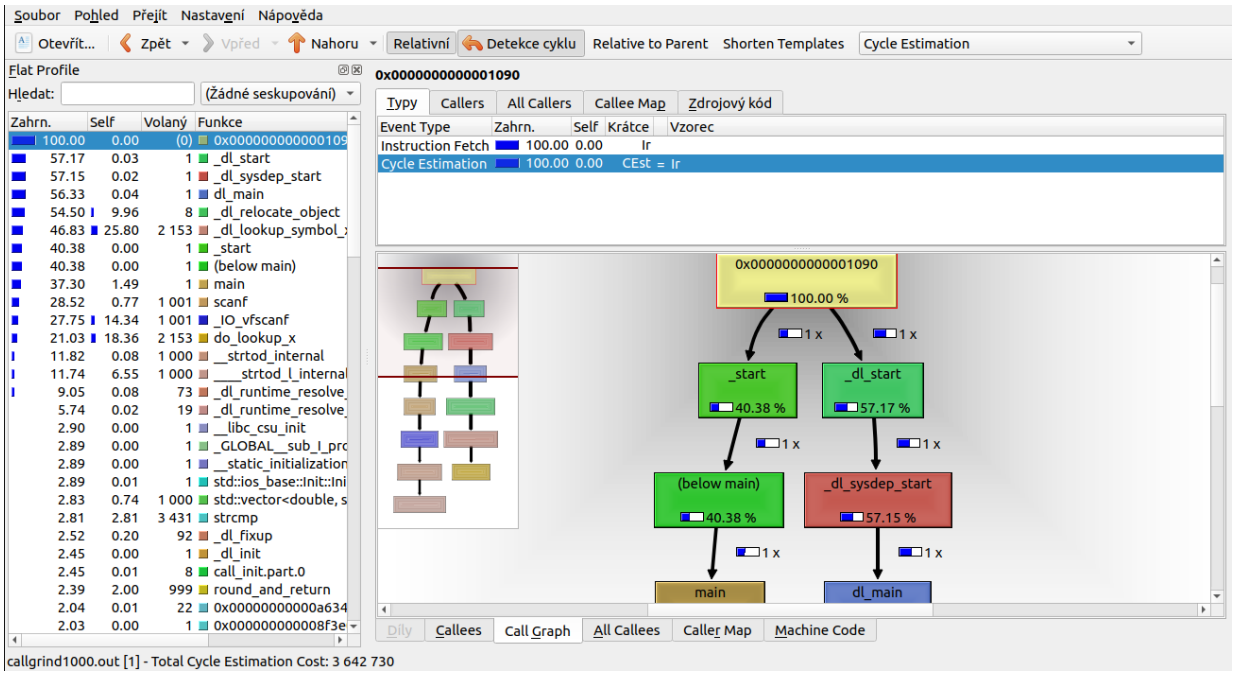


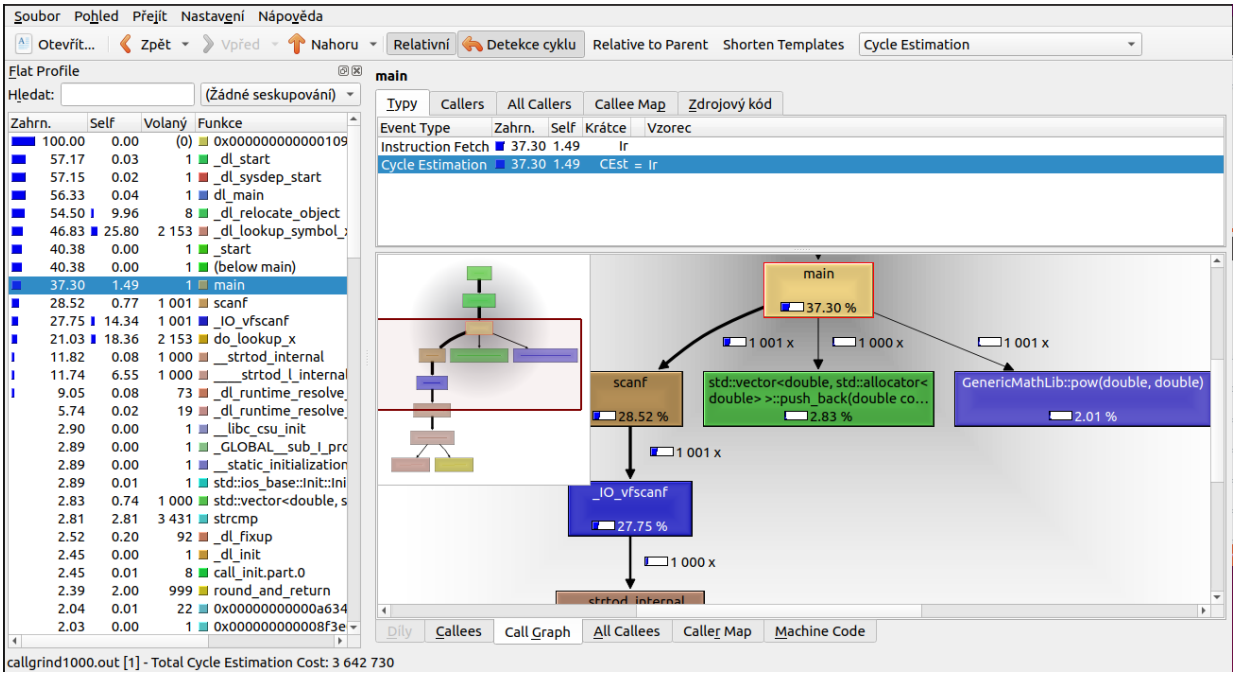
1.2 Grafy volání při spuštění programu se sto čísly na vstupu





1.3 Grafy volání při spuštění programu s tisíci čísly na vstupu





2 Shrnutí

Na všech třech grafech volání, které jsou zobrazeny z kořene je vidět, že úplně nejvíce času tráví program na načítání **dynamických knihoven**.
Pokud se podíváme na grafy volání zobrazené z mainu daného programu, můžeme vidět, že v mainu trvá nejdéle provedení funkce **scanf**.

3 Na co se zaměřit při optimalizacích

Z těchto výstupů vyplývá, že nejlépe je zaměřit se na způsob sestavování programu a zamyslet se nad tím, zda-li není lepší použít **statické sestavení knihoven**, které se volají při sestavování programu než **dynamické sestavení knihoven**, které se volají až při spuštění programu, což daný program zpomaluje a co nejvíce omezit volání různých knihovních funkcí v rozsáhlých cyklech.