



# **Vysoké učení technické v Brně**

Fakulta informačních technologií  
Elektronika pro informační technologie  
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

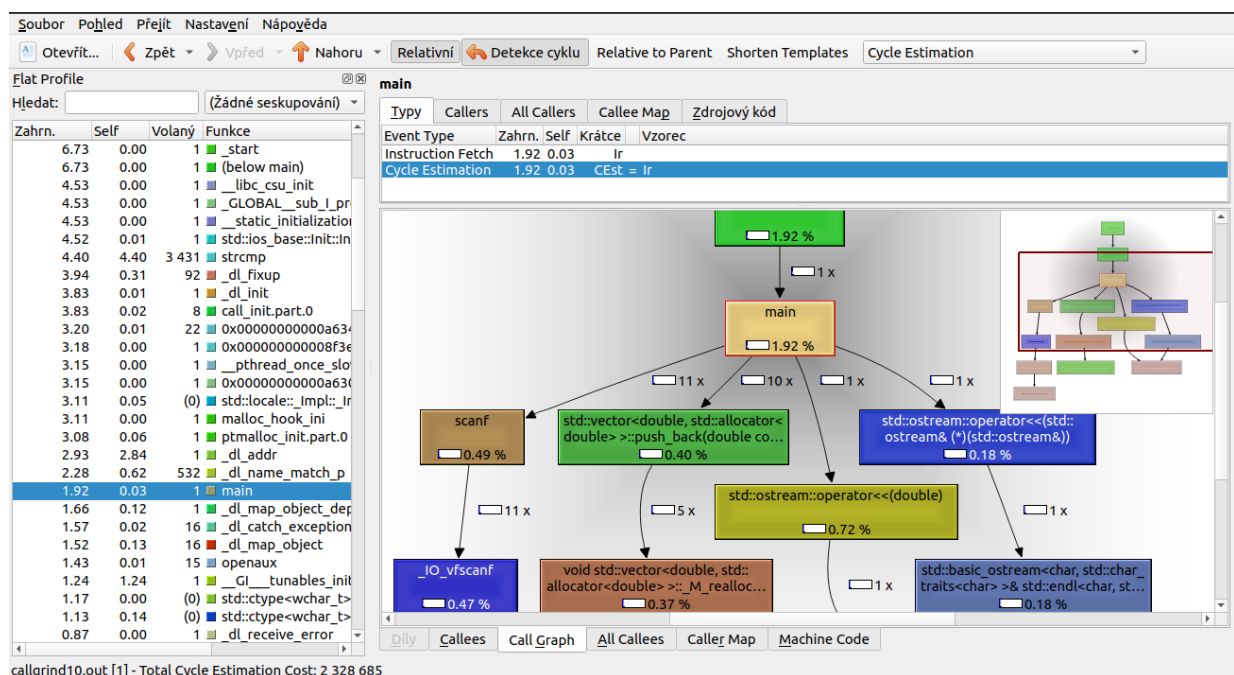
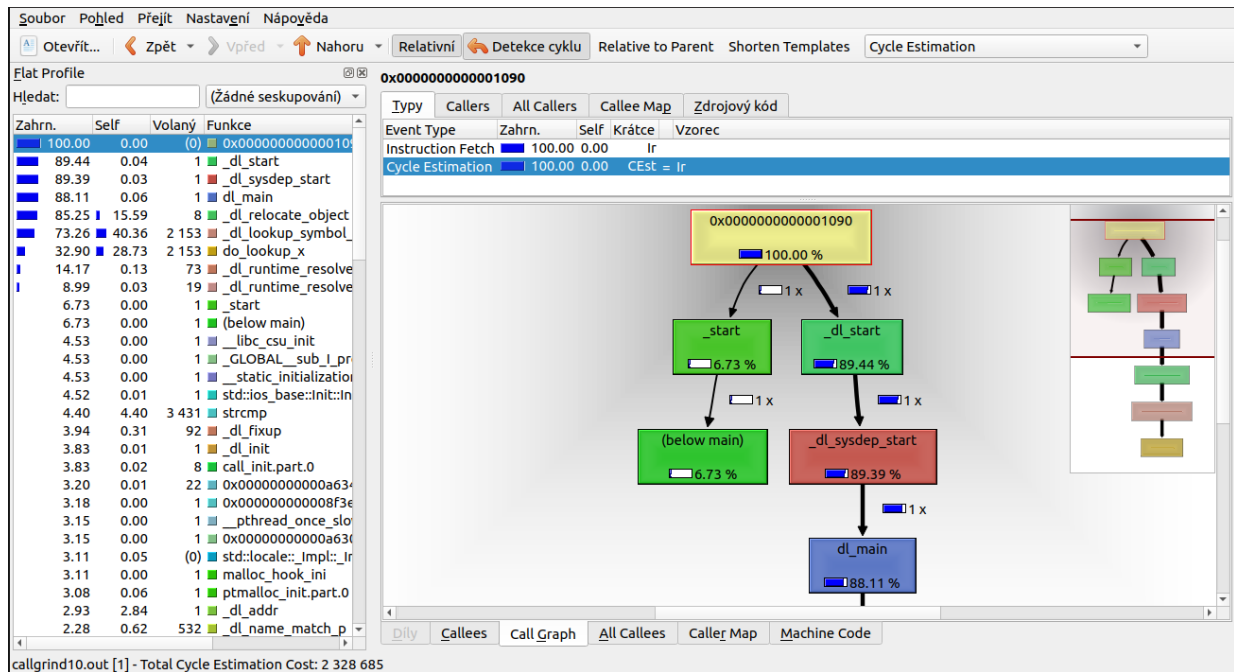
# 1 Výstupy z profilování

Pro profilování jsme použili program KCachegrind.

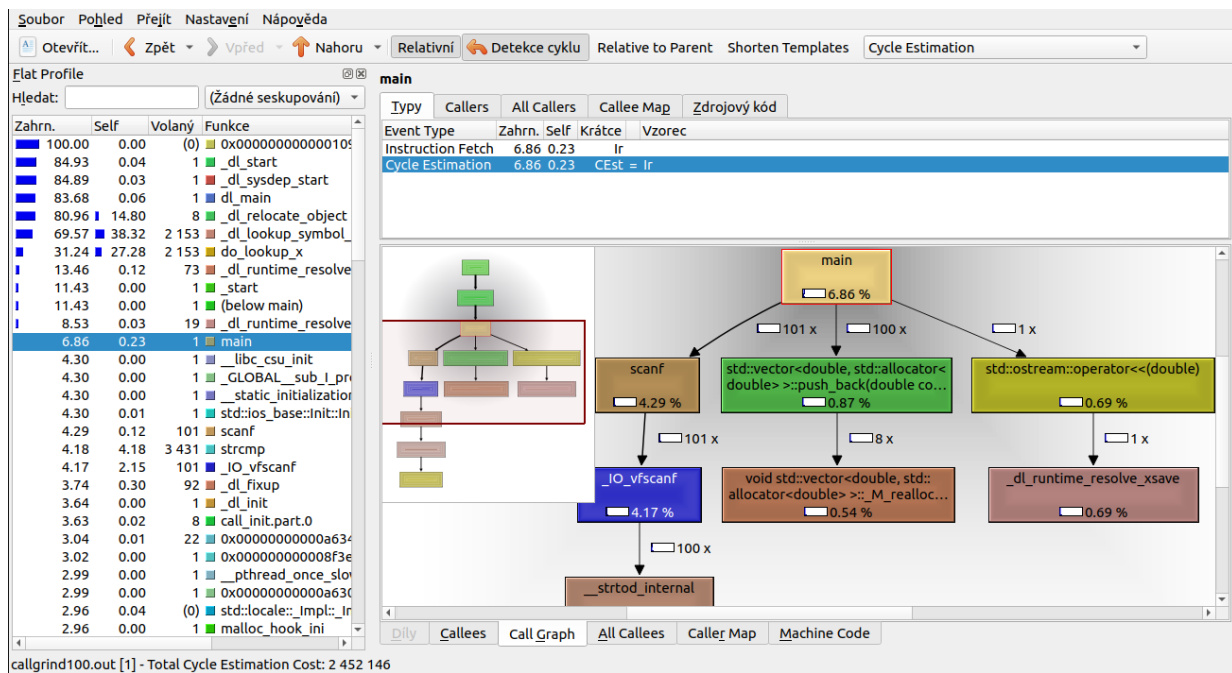
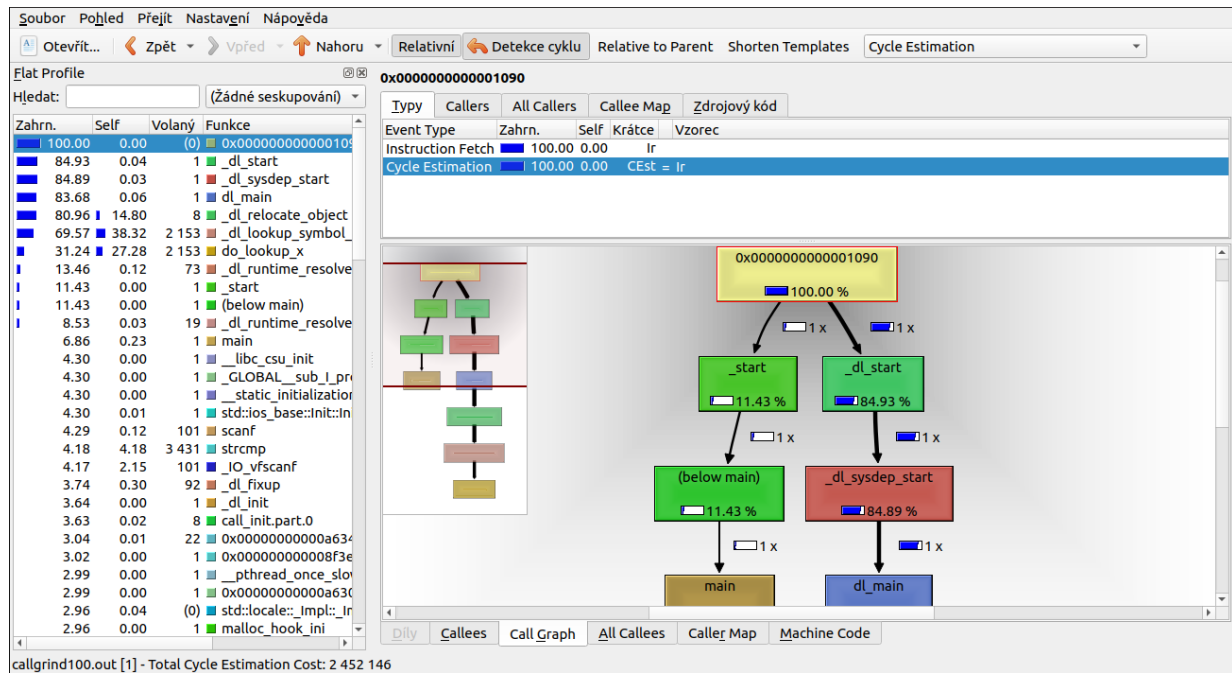
## 1.1 Výstupy

První obrázek pro každou hodnotu vstupu je graf volání z rootu, druhý je pak graf volání z mainu programu.

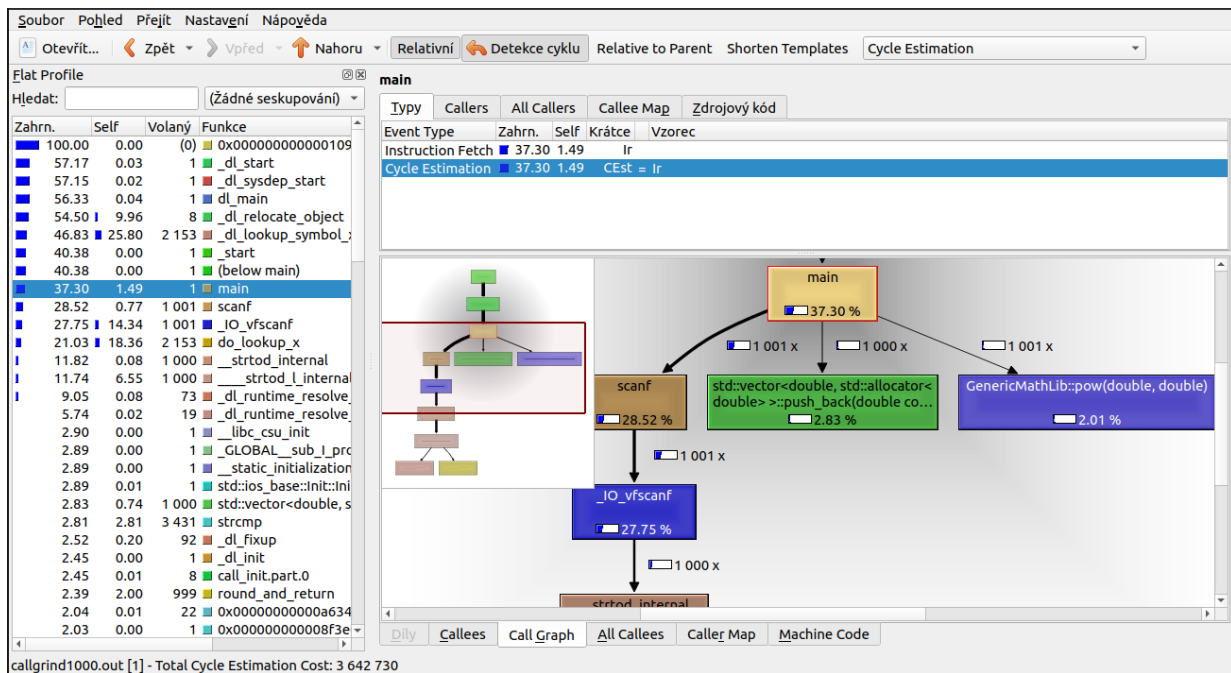
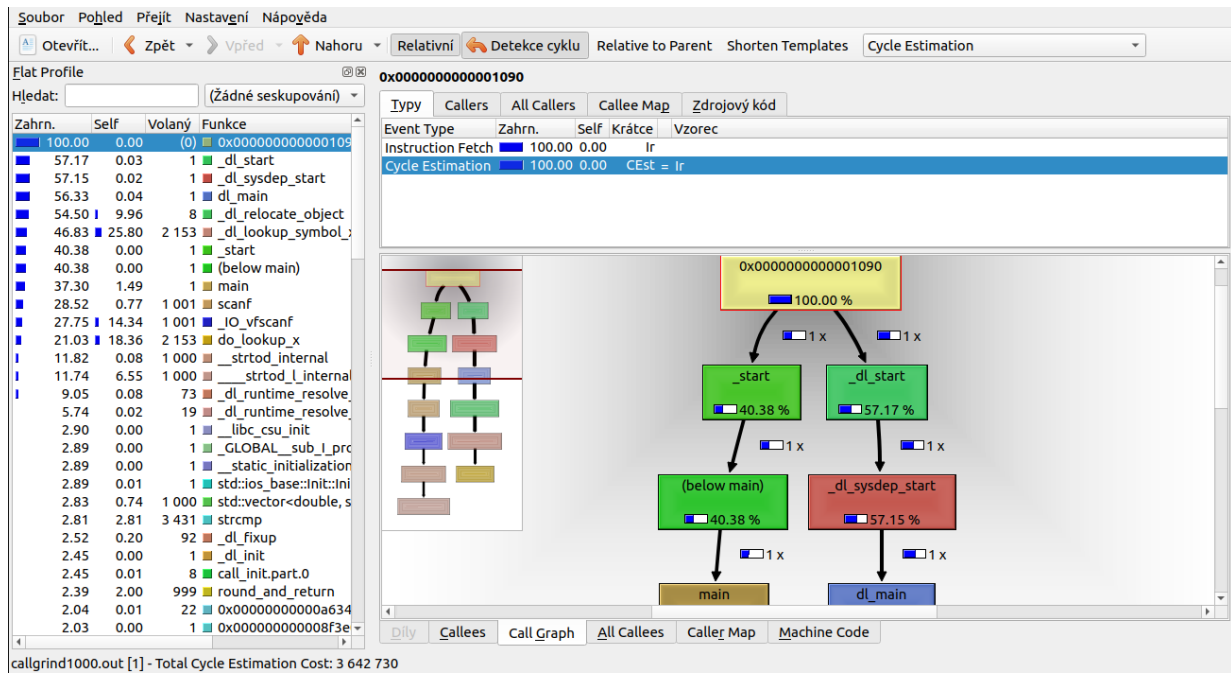
Graf volání při spuštění programu s deseti čísly na vstupu:



Graf volání při spuštění programu se sto čísly na vstupu:



Graf volání při spuštění programu s tisíci čísly na vstupu:



## 1.2 Shrnutí

Na všech třech grafech volání, které jsou zobrazeny z kořene je vidět, že úplně nejvíce času tráví program na volání **dynamických knihoven**.

Pokud se podíváme na grafy volání zobrazené z mainu daného programu, můžeme vidět, že v mainu trvá nejdéle provedení funkce **scanf**.

## 1.3 Na co se zaměřit při optimalizacích

Z těchto výstupů vyplývá, že nejlépe je zaměřit se na linkování a používání dynamických knihoven, protože jejich voláním program tráví nejvíce času a co nejvíce omezit volání různých knihovnických funkcí v cyklech.