Contribution ID: **67** Type: **Poster**

AdaptivePerf: a portable, low-overhead, and comprehensive code profiler for single- and multi-threaded applications

Wednesday, March 13, 2024 4:15 PM (30 minutes)

Considering the slowdown of Moore's Law, an increasing need for energy efficiency, and larger and larger amounts of data to be processed in real-time and non-real-time in physics research, new computing paradigms are emerging in hardware and software. This requires bigger awareness from developers and researchers to squeeze as much performance as possible out of applications. However, this is a difficult task in case of complex and large codebases, where pinpointing a specific bottleneck is hard without profiling. At the same time, finding a suitable low-overhead and extensive profiler which works for a wide range of homogeneous and heterogeneous architectures proves to be challenging, especially when multi-threaded application support, user-friendliness, and reliability are important.

To address this, we introduce AdaptivePerf, developed in the context of the SYCLOPS EU project: an open-source comprehensive profiling tool based on sampling and system call tracing through patched Linux perf. It improves shortcomings sometimes observed in Linux perf such as broken stacks, no off-CPU data, and reports unclear to developers. The tool also further builds on its functionality by profiling how threads and processes are spawned within a program and what code parts are most time-consuming on- and off-CPU for each thread/process. Additionally, AdaptivePerf presents results in a user-friendly way by showing an interactive timeline with stack traces of functions starting new threads/processes and the browser of per-thread/per-process non-time-ordered flame graphs and time-ordered flame charts.

AdaptivePerf is designed with architecture portability in mind: the main features are based on hardware-independent metrics like wall time, while hardware-specific extensions such as profiling non-CPU devices (GPUs, FPGAs etc.) or capturing CPU-dependent perf events are possible. Thanks to our tool, detecting bottlenecks and addressing them within software and/or hardware is significantly easier for developers and researchers.

In this presentation, we will discuss the status of the profiler and its future prospects.

Significance

References

Experiment context, if any

Author: GRACZYK, Maksymilian (CERN)

Co-author: ROISER, Stefan (CERN)

Presenter: GRACZYK, Maksymilian (CERN)

Session Classification: Poster session with coffee break

Track Classification: Track 1: Computing Technology for Physics Research