



Contribution ID: 100

Type: Talk

## Enhancing software-hardware co-design for HEP by low-overhead profiling of single- and multi-threaded programs on diverse architectures with AdaptivePerf

Tuesday 22 October 2024 13:48 (18 minutes)

Given the recent slowdown of the Moore's Law and increasing awareness of the need for sustainable and edge computing, physicists and software developers can no longer just rely on computer hardware becoming faster and faster or moving processing to the cloud to meet the ever-increasing computing demands of their research (e.g. the data rate increase in HL-LHC). However, algorithmic optimisations alone are also starting to be insufficient, so novel computing paradigms spanning both software and hardware appear. Adapting existing and new software to them may be difficult though, especially for large and complex applications. This is where profiling can help bridge the gap, but finding a suitable profiler is challenging when a low overhead, wide architectural support, and reliability are important.

As a response to the above problem, AdaptivePerf was developed. It is an open-source, architecture-portable, and low-overhead profiling tool with custom-patched Linux perf as its main foundation, currently available on GitHub. Thanks to the extensive research and modifications, AdaptivePerf improves the main shortcomings of perf such as incomplete stack traces. It profiles how threads and processes are created within a program and what code segments within each thread/process should be considered on- or off-CPU bottlenecks, in terms of both consumed time and other hardware metrics like cache misses. If a user-friendly visualisation is needed, AdaptivePerf can present results as a timeline with the process tree, where corresponding non-time-ordered and time-ordered flame graphs can be browsed along with functions spawning new threads/processes.

The tool has already been shown to work on x86-64 and RISC-V and is designed in the context of the SYCLOPS EU project, which CERN is part of and where solutions for heterogeneous architectures are developed, e.g. custom RISC-V cores tailored to a specific problem, RISC-V support for SYCL, and SYCL-accelerated algorithms in ROOT. In this presentation, we will talk about the profiler, its place within the project, and how it can be used for software-hardware co-design for HEP.

**Primary author:** GRACZYK, Maksymilian (CERN)

**Co-author:** ROISER, Stefan (CERN)

**Presenter:** GRACZYK, Maksymilian (CERN)

**Session Classification:** Parallel (Track 6)

**Track Classification:** Track 6 - Collaborative software and maintainability