A Deep Dive into C-States, Idle Governors and the Prospects of an eBPF Idle Governor

Linux is one of the most utilized Operating Systems in Embedded Systems and Cloud Infrastructure worldwide. Sustainability will become more relevant in the future and saving power is a crucial aspect. This shows the increasing importance of efficient Linux Power Management.

The Power Management in Linux is implemented in several kernel subsystems correlating to hardware characteristics, like P-States (Frequency Scaling) and C-States (Sleep States). This thesis examines the Idle Power Management of Linux, and therefore focuses on C-States. C-States are per Core states and allow parts of the core to shut down individual features. Each processor implements C-States in different ways. Increasing C-State number, e.g. C6, translate to a deeper sleep with lower energy consumption and higher power-on reaction time.

The recently released eBPF functionality makes the kernel more programmable, bypassing the original monolithic characteristics. This mechanism can be divided into four components: the eBPF hooks in the kernel, the interfaces, the in-kernel eBPF infrastructure to execute eBPF bytecode and compile into native code and verify the code and finally the eBPF application itself, which can be written in a C like dialect and compiled into eBPF bytecode by LLVM and GCC.

This thesis aims to analyze and compare the idle governors in the current Kernel in specific situations. It also should provide insight in the C-State usage depending on the architecture. The data is acquired using specific Tracepoints within the Kernel, which can be recorded and parsed with the Kernel Tool perf. Furthermore, we explore the feasibility of a custom eBPF powered idle governor.

Advisors

Marco Liess
Hagen Pfeifer (Rohde & Schwarz)