Icewind: Automatic, Hardware-Accelerated Emulation of Formal Instruction Set Architecture Descriptions

Ferdia McKeogh
fm208@st-andrews.ac.uk
University of St Andrews

Dr Tom Spink
tsc6@st-andrews.ac.uk
University of St Andrews

The Sail language for defining ISA specifications has widespread industry adoption; Arm generates Sail specifications for their ISAs automatically from their internal specifications, and RISC-V use Sail to define their authoritative specification. These ISA definitions can be used for formal verification research and generating tests for hardware. They can also be used to generate emulators for those ISAs, such as the included C-based Sail emulator and the Pydrofoil project. However, these generated emulators do not achieve the same performance as handwritten dynamic binary translation (DBT) systems such as QEMU and Captive. This higher performance comes at the cost that the handwritten models are time-consuming to produce and error-prone compared to automatic methods.

We present the `icewind` toolchain to demonstrate how functional elements from these formal models can be extracted automatically to achieve the high speeds of the handwritten systems, but with the ease and correctness of the automatic emulators. The toolchain consists of an offline compiler and a DBT runtime system employing aggressive offline optimizations, partial evaluation during binary translation, use of the host MMU to accelerate guest memory translations, and Intel Extended Page Table to accelerate the invalidation of modified code pages to avoid executing stale code. `icewind` achieves a 66.9× speedup over Pydrofoil and an ≈15000× speedup over the Sail C emulator (the two automatic tools) while only being 0.55× as slow as the current state-of-the-art handwritten DBT, QEMU, in the CPU2017 `mcf` benchmark.