Speculative Common Path Optimizations for Datacenter Workloads

Compilers utilize techniques such as constant-value propagation to eliminate certain code sections at compile time. However, as compilers must prove that values are constant, optimization opportunities are often limited. In this project, we propose runtime profiling to track value-invariant variables, increasing the opportunities for the compiler. In particular, we propose safe code transformation techniques that optimize the common execution path of applications, substantially improving performance. First, we developed a PIN-based value profiling tool to collect invariant and constant function call parameters. As PIN-based instrumentation tools can introduce substantial overheads, we have developed several performance optimizations enabling us to profile real applications, such as the SPEC2017 and Cortex suites, in acceptable times. Second, we have shown that these insights can be used to optimize code by enabling additional constant-value propagation opportunities. Third, we have developed an LLVM function pass that utilizes our obtained profile data for automatic code transformations. In particular, the pass exploits value invariant function call parameters, inserting an optimized, fast path into the code that eliminates the instructions depending on such invariant variables.