Criticality: Static Profiling for the Worst Case

Motivation

WCET analysis yields a 1-dimensional result: worst-case path. But what about code not on this path?

The Criticality Metric

\[ \text{Crit}(bb) = \frac{\text{WCET}(bb)}{\text{WCET}} \]

WCET(bb)… worst-case execution over block bb

Yields information beyond worst-case path.

How to compute it (efficiently)?

- Repeatedly solve the IPET problem (an ILP) with additional constraints.
  - use dominance properties to propagate results
  - prune Criticality values at a threshold value
  - obtain estimates through less precise WCET analysis (Criticality is robust)

How can it be used?

- By the programmer: as WCET profiling tool
- Compiler: for optimizations targeting WCET
- WCET Analysis: to focus static analysis to critical parts of a complex program

Example CFG (before & after Criticality computation)