

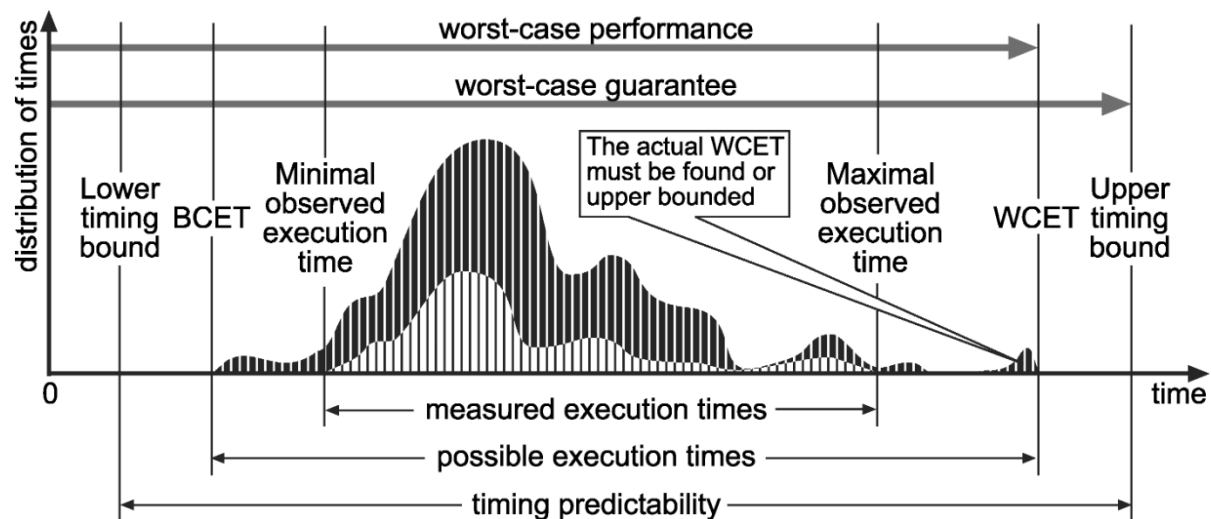
Bachelor Thesis

Worst-Case Execution Time Computation for Python Programs

Problem Statement

In our research project, we develop a runtime verification framework with look-ahead capabilities for Python programs. We analyze Python programs not directly, but the generated Python Bytecode which is an assembly-like intermediate representation for Python programs. Runtime verification checks, while the program is executing, whether a given property is fulfilled and returns a verdict. In order to derive a verdict about future program states information about the worst-case execution time is needed.

Since the Python program is executed on a not real-time capable system and also uses dynamic memory allocation a precise analysis is not possible. In these cases, a measurement-based approach can help finding lower and upper bound on the worst-case execution time.



WORST-CASE EXECUTION TIME DIAGRAM¹

Your Tasks

In this thesis a worst-case execution time analysis for Python should be developed and is to be evaluated using an industrial use case.

- ▶ Literature research on slicing techniques for assembly-like languages
- ▶ Adaption and implementation of a promising approach which can handle pointer aliasing
- ▶ Evaluation on an industrial use case

Your Profile

The analysis should be implemented in Python and therefore intermediate language skills are preferred. Optional is knowledge about formal methods and worst-case execution time computation, but general understanding of program analysis is helpful.

Contact

Thomas Henn, M. Sc. RWTH
henn@embedded.rwth-aachen.de

[1] Wilhelm u. a., „The Worst-Case Execution-Time Problem—Overview of Methods and Survey of Tools“.