

Bachelor/Master Thesis

LLVMTA: Adding AARCH64 (ARMv8,9) support to a static WCET analyser

Nils Hölscher
 Prof. Dr. Jian-Jia Chen
 Otto-Hahn Str. 16
 Technische Universität Dortmund
 Email: nils.hoelscher@tu-dortmund.de
 10.03.2022

LLVMTA is a static Worst Case Execution Time (WCET) analyser published recently. Such analysers determine an upper bound of the program's execution time (so-called WCET), which is very useful for real-time systems. To estimate the WCET of a program, its Control Flow Graph (CFG) has to be reconstructed from the binary, by which all necessary analyses approaches can be conducted, like value analysis, path-analysis and cache analysis, as shown in Figure 1.

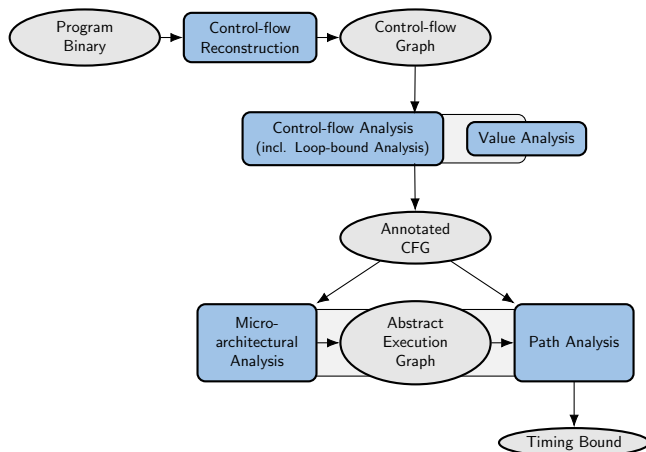


Figure 1: Overview of the general steps in WCET analysis.

Conventional tools like OTAWA or aiT take the generated binary code to reconstruct the CFG. However, it is already generated during a programs compilation, and the reconstructed one might lose important information for analysis.

Alternatively, LLVMTA takes a different approach. As it is integrated into LLVM, the program can be analysed from its native code in C, instead of the generated binary. The whole compilation flow of LLVM and LLVMTA is shown in Figure 2. The current version of LLVMTA supports two 32-bit ISAs, namely RISC-V and ARMv4. To make LLVMTA more comprehensive, an additional support to 64-bit ISAs such as AARCH64 is highly relevant, which will align with the recent development in the industry, e.g., ARM.

In this thesis, students should make themselves familiar with LLVMTA, LLVM and the AARCH64 ISA. After

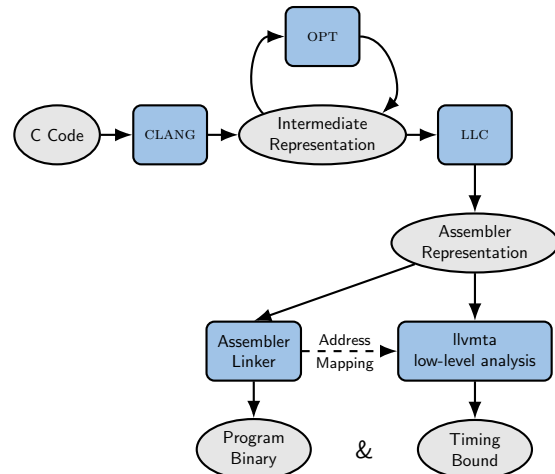


Figure 2: Overview of the common LLVM compilation flow (CLANG, OPT, LLVM) including the integration of our low-level analysis tool LLVMTA.

that the abstract address system of LLVMTA has to be adapted to 64 Bit. Finally the students should implement as many AARCH64 instructions into the LLVMTA analyser as possible, starting with simple load, store instructions and simple arithmetic instructions. The minimal goal should be to successfully analyse a simple loop example.

Other suggestions and related topics are also welcome. Please do not hesitate to make an appointment.

Required Skills:

- Knowledge of computer architecture
- Basic/Advanced knowledge of C++

Acquired Skills after the thesis:

- Deep knowledge about ARMv9 ISA.
- Insights into the LLVM compiler ecosystem
- Knowledge about static WCET analysis.

Literature:

- [LLVMTA: An LLVM-Based WCET Analysis Tool](#)