

## Master Thesis

# Vectorization on ARM

## Description

Vectorization is a key technique for code optimization. It exploits Data Level Parallelism (DLP) by grouping instructions into vectors and applying vector instructions instead of scalar ones. In today's general purpose processors, vectorization is supported by Single Instruction, Multiple Data (SIMD) ISA extensions and hardware. Nonetheless, utilizing this special hardware efficiently is still a challenge. Best performance is typically achieved with manual code re-writing, which is time-consuming and limits code portability across platforms, i.e. SIMD ISAs. That's why this task should be performed by compilers.

For this purpose, modern compilers run one or more vectorization passes. Due to intensive research in the past years, significant progress has been made for both, the vectorization rate (*"How many code patterns are vectorized?"*) and the vectorization quality (*"How much speedup is achieved?"*). Nonetheless, vectorization results depend on the base ISA, and with the rise of ARM microarchitectures, the question is how well compilers perform for this platform.

In this work, the vectorization capabilities of the most common C/C++ compilers on ARM shall be investigated. Using a set of benchmarks, ranging from basic loop patterns to full applications, the vectorization rate and the vectorization quality have to be assessed. Furthermore, a qualitative analysis of why the vectorization fails should be performed, providing insight into what is still missing in com-

compilers to utilize the ARM Neon instruction set to its fullest potential.

**Keywords:** SIMD, vectorization, ARM, Neon, compiler

## Required Skills

- knowledge of computer architecture
- programming skills in C/C++

## Desired Skills

- basic understanding of compilers

## Contact Person

Angela Pohl  
(angela.pohl@tu-berlin.de)