Bachelor/Master Thesis

Engineering Faster Sorters for Small Sets of Items

Overview

Most sorting algorithms use insertion sort as base case sorter for small sets of elements. The goal of this thesis is to investigate if this can be improved in practice.

For example, optimal sorting functions with the minimum number of comparisons needed are known for up to around 16 elements, but these may not be very fast in practice due to branch mispredictions. Alternatively, sorting networks have a fixed pattern of swap operations, which may be pipelined with conditional moves by processors if the elements are independent. Another approach would be to use explicit vectorization using SIMD instructions.

Goal is to design, implement in C or C++, and evaluate sorting algorithms for small sets of elements. The algorithms should be profiled on modern processors (Intel and ARM) and designed to utilize the hardware accelerations available on these platforms. The new sorters for small inputs should ultimately be integrated in the currently fastest parallel in-place sorting implementation [1] and other algorithms.

This master thesis builds on previous work on sequential (string) sorting and parallel (string) sorting on multi-core systems [1, 2], and on parallel memory bandwidth experiments [3].

Requirements

- Interest in sorting algorithms and characteristics of modern CPUs
- Excellent programming skills in C++, interest in reading assembly code.
- Ability to think for yourself

