Resource sharing occurs when multiple active processes or software components compete for system resources, which influences the observed performance compared to an individual execution. Isolated benchmarking of durations of key operations for solving of performance prediction models may therefore yield imprecise results. Resource sharing also occurs between the measured code and the benchmark infrastructure for obtaining and storing samples, imposing an indirect overhead.

This thesis quantifies the effects of sharing on performance for several resources that are often shared, namely the processor caches and the file systems. The highest possible performance impact of cache sharing is determined by synthetic benchmarks. Impact on practical code and its dependency on a number of factors such as cache trashing frequency and intensity are then determined by experiments with existing implementations of FFT and LZW algorithms and a video stream processing application. Effects of file system sharing are measured by experiments that read and write multiple files simultaneously. For both resources, situations with significant performance impact of sharing have been observed.

Based on the results of the experiments, several suggestions for dealing with the overhead of performance monitoring infrastructure are proposed, and applicability of the experiments and their results for performance modeling is discussed.