Joseph Dewar

Dr. Bellardo

Computer Architecture

22 March, 2023

Lab 6 Report

Introduction

The objective of this lab was to assess the impact of various parallelization strategies, such as vector processing and multi-threaded implementations, on the performance of a well-established program. The application under investigation is a matrix multiplication program, matmul, which multiplies two matrices and records the result.

The first parallelization strategy involves employing ARMv8's neon instructions for vectors in the matmul routine to utilize SIMD instructions. The second parallelization strategy entails using OpenMP pragmas in the C++ implementation of the matmul program to parallelize loops that compute dot products with different numbers of threads.

To gauge the effectiveness of these parallelization strategies, the runtime, number of instructions, clock rate, and instructions per cycle will be measured and compared across different implementations.

Method

For the SIMD implementation, the ARM assembly version of the matmul routine was modified to include SIMD instructions, specifically vectored load and multiply instructions. The inner loop was unrolled to compute four multiplications per iteration.

For the OpenMP implementation, matmul loops were parallelized using OpenMP pragmas in the C++ version of the program. Different instances with varying numbers of threads were measured to assess the performance of the multi-threaded implementation.

Results

The table below presents the performance results for each implementation:

matmul	arm.csc.calpoly.edu Runtime	#Instructions	Clock Rate GHz	Instuctions Per Cycle
Runtime SIMD	39.84407237	31,793,030,939	1.996	0.41
Runtime 1 Thread	48.87057687	72,536,701,960	1.996	0.74
Runtime 2 Threads	31.68210016	72,539,069,394	1.993	0.69
Runtime 4 Threads	18.4796083	72,543,218,471	1.993	0.67
Runtime 8 Threads	10.71296603	72,556,767,580	1.992	0.67
Runtime 16 Threads	7.306737127	72,581,686,339	1.994	0.7
Runtime 32 Threads	5.892482751	72,635,625,773	1.992	0.65
Runtime 64 Threads	4.971285593	72,717,739,357	1.995	0.65

Discussion of Results

The findings demonstrate the significant impact of parallelization strategies on performance. The SIMD implementation exhibits a reduced runtime compared to the single-threaded version, emphasizing the advantages of vector processing. The OpenMP implementation displays a decrease in runtime as the number of threads increases, with the most substantial improvement occurring up to 16 threads. Beyond this point, the enhancements in runtime are less significant, potentially due to overheads associated with thread management.

Conclusion

Both parallelization strategies SIMD and OpenMP both provide performance enhancements for the matrix multiplication program. This lab demonstrates some useful techniques for optimizing memory bound programs. Thank you for the great quarter, Dr. Bellardo! I hope to take another class of yours again maybe in Operating Systems 2.