Compilers for Embedded Systems

Winter semester 2023-24

P. Hadjidoukas

Set 1 - Performance

Issued: November 20, 2023

Question 1: Roofline Model

Given the following serial GEMM code snippet:

```c
for (int k=0; k<N; k++)
    for (int i=0; i<N; i++)
        for (int j=0; j<N; j++)
            C[i+j*N] += A[i+k*N]*B[k+j*N];
```

a) Can you identify potential problems and/or difficulties in the loop structure of the code, in particular regarding parallelization and locality?

b) What is the operational intensity (single precision) for a matrix of size N, assuming there is no caching? Count floating point operations and memory accesses.

c) What is the operational intensity (single precision) for a matrix of size N, assuming an infinite cache size? Count floating point operations and memory accesses.

d) Draw the roofline plot for your system and add vertical lines corresponding to the minimum and maximum operational intensities computed in parts b) and c) for $N = 80$.

e) What is the maximum performance you can reach with the two operational intensities computed for $N = 80$ given that a compute node has a bandwidth of 96 GB/s and a peak single precision floating point performance of 844.8 GFLOP/s? Show your calculations.

Question 2: Performance Measurements

Use PAPI to measure the performance (GFLOP/s and Cache Hit Ratio) of the GEMM code, implemented with simple looping, for each of the 6 possible loop execution orders ("ijk", "ikj", "jik", "jki", "kij", "kji").

Make sure that your implementation gives correct results in all cases. Report the results in a suitable table.

Perform your experiments for two cases of matrix size: the dataset (e.g. arrays) fits and does not fit in the cache of your system.

Question 3: Code Improvements

Use PAPI to measure the performance (GFLOP/s and Cache Hit Ratio) of the GEMM code, for the following code optimizations:

1. Loop tiling
2. Loop unrolling and explicit SIMD vectorization
3. Use of the BLAS implementation

Report your design decisions (e.g. block size) and the corresponding results.

Perform your experiments for two cases of matrix size: the dataset (e.g. arrays) fits and does not fit in the cache of your system.