LAB01 - version 00

November 8, 2022

Overview: The purpose of this lab is to illustrate the parallelization of applications using the Single Instruction Multiple Data (SIMD) model, Shared Memory Parallelism, caching and the use of low-level optimizations. For this we will examine the matrix transposition and stencils, which have sufficient complexity to demonstrate the concepts of the lab without requiring siginificant outside domain knowledge.

You will be provided correct, unoptimized, baseline implementations of this operation, along with the appropriate testing and timing harnesses. Your primary objective will be to iterate and create multiple variants of this operation using the concepts learned in class. From those iterations you will select a specified number of variants for the operation and provide a writeup that sumarizes the results and the relationship to the material covered.

The assumption is that no significant prior knowledge of low-level hardware details is necessary. The expectation is that you acquire this knowledge through reference manuals and more importantly through practice. The exercise of iterating over the code allows you to test your hunches and refine your ideas. Do not copy outside code, because that is plagairism.

Deliverables: Your team will be responsible for pushing several key deliverables for each part to a GitLab repository that you will create and give maintainer access to the instructor and TA.

1. `sow_timeline.txt`: Before you begin code describe in this file the work to be done, how the work will be divided and a tentative timeline that you and your team will follow.

2. Code: For each problem you will have these code artifacts.

   `optimized_op_XX_var_YY.c` These are your team's implementations of the target operations.

   `op_XX_dispatch_vars.sh` Modify these files to point to your variants that you want to be graded on.

3. `writeup.pdf`: Your writeup should make clear how your work relates to the material covered in class. Specifically, your selected variants and writeup should use and relate to SIMD parallelization, blocking/striding for caches, low level optimizations, and specilization to corner cases. The writeup should be a well written document that includes, performances plots for each variant and the following sections for each of the three operations:

   Overview For each operation answer the following questions: What is the problem you are trying to solve? What are the pain points in this problem? How do you plan to solve this? And how does this relate back to the material covered in class.

Refinements For each variant of each operations answer the following questions: What did you change between each variant? What should we expect to see in the results and why? Include performance plots, what do the results show? What interesting features should we note in the plot? What would you do to improve these result?

4. `selfassessment.txt`: In this file perform a postmortem. What was the work that was done? How was it divided? How would you distribute the grade (50/50, 60/40, 33/33/33?). What would you do differently next time?

If there any any issues with this document or the provided code, please let me know so I can update it.