

CSE141L Lab 4 Threads

Name: _____

Student ID: _____

Instructions

- Complete this worksheet while reading/working through the lab write up. The worksheet doesn't make sense without the lab.
- The point values are listed for each question. Altering the size of the cells will cost you 1 point. There are 42 points total for the write up portion of the lab.

Tier 1: Adding Multithreading to `fc_layer:calc_grads()`

P1 (4pt) Look at the spec sheet of the CPU we are using in our servers:**[<https://ark.intel.com/content/www/us/en/ark/products/93848/intel-xeon-processor-e3-1578l-v5-8m-cache-2-00-ghz.html>]. Fill in the blanks with the specifications of the CPU in our cloud servers.**

Number of cores: _____

Is hyperthreading enabled?: _____

Maximum number of threads that can run concurrently: _____

Number of threads available per core: _____

P2 (4pt) Try adding multithreading with 2 threads above the following loops in `fc_layer:calc_grads()`: `nn`, `b`, `n`, `i`. Fill in the table below with the runtime of `fc_layer:calc_grads()` for multithreading each of these loops.

Loop	Runtime in seconds	Speedup over single threaded
single threaded	_____	_____
nn	_____	_____
b	_____	_____
n	_____	_____
i	_____	_____

P3 (4pt) Adding multithreading to which loops resulted in threads writing to the same memory space and which variable needed to be created for each loop locally? Answer yes if the loop resulted in this, no if not. Write the variable name after your yes or no answer

nn: _____ b: _____ n: _____ i: _____

P4 (1pt) Multithreading which loop provided the best speedup?

Loop: _____

P5 (4pt) Try multithreading the loop that gave the best speedup with different number of threads. Fill in the table below with the runtime of `fc_layer:calc_grads()` for each number of threads.

# of threads	Runtime in seconds	Speedup over single threaded
single threaded	_____	_____
2	_____	_____
4	_____	_____
6	_____	_____

P6 (4pt) Plot "Speedup over single threaded" vs "# of threads".

Your graph here.

P7 (4pt) At how many threads does speedup start to decrease? Why?

Number of threads: _____

Explain why here.

Tier 2: Adding Multithreading to fc_layer:fix_weights()

P1 (4pt) Try adding multithreading with 4 threads above the following loops in fc_layer:fix_weights(): b, n, i. Fill in the table below with the runtime of fc_layer:fix_weights() for multithreading each of these loops.

Loop	Runtime in seconds	Speedup over single threaded
single threaded	_____	_____
b	_____	_____
n	_____	_____
i	_____	_____

P2 (1pt) Multithreading which loop provided the best speedup?

Loop: _____

Tier 3: Applying More Optimizations

P1 (5pt) Give a brief description of which functions you added more optimizations to and why. Report the speedup you achieved for each one.

Your answer here

P2 (5pt) Give a description of the additional optimizations you implemented to speedup training. Mention the loops, of the all the functions, on which you applied multithreading. Also include other loop optimizations you added to the functions. The answer should give us a fair understanding of your optimizations without looking at your code.

Your answer here

Lab Reflection

Follow this link 24 hours before or after the due date to fill out the reflection survey. It is worth 5% of your lab grade.

<https://forms.gle/XQYzHEbkxbxQUFvFF9>

Describe What you Did

(2pt) For the best version of each of your functions write out the loops you added multithreading to and how many threads you used.

Example:

```
do_stabilize_tile_y_1_omp_critical_fast() (Found in example/stableize.cpp)
offset_x[X]
```

This means the loop with the variable `offset_x` was parallelized with `X` threads.

There is no correct answer here, we just want to know what you did.

Your answer here