CENG311 **Fall 2021**

PROGRAMMING ASSIGNMENT 4

Due date: 15.01.2022 23:00

In this assignment, you are required to run a set of experiments and analyze the results in order to understand the cache behavior of programs, how different cache designs affect program execution.

You will use the cache simulator (dcache) provided in Pin tool [1] to collect cache miss rates. You need to compile and run the matrix multiplication codes provided as part of this assignment, then you will run both naive and blocked matrix multiplication versions by using the simulator as discussed in the lab session (or you can learn from the Pin user guide: https://software.intel.com/sites/landingpage/pintool/docs/98484/Pin/html/)

You will run the matrix multiplication program for a set of configurations with the following L1 cache parameters in the cache simulator:

L1 Size [KB]	Block size [B]	L1 Associativity
8	32	1
8	32	2
8	32	4
16	16	1
16	32	1
16	64	1
16	16	2
16	32	2
16	64	2
16	16	4
16	32	4
16	64	4
32	32	1
32	32	2
32	32	4

You will execute the experiments with the default matrix size (N=1000) and block size 100 for the blocked version. You will have 30 different executions (2 programs (i.e., naive and blocked) and 15 cache configurations).

After executing each experiment, collect the L1 data cache load miss rates (and/or any other relevant values), and draw graphs to demonstrate the effect of the blocking and cache parameters on miss rates. You need to show the L1 load miss rates in the graphs, you can use the other values to explain your results. Examine the results and explain them by providing your comments.

Notes:

- You can run your experiments by setting cache parameters from the command line like this:

```
./pin -t source/tools/Memory/obj-intel64/dcache.so -o output_32_32_4.out -c 32 -b 32 -a 4 -- ./mul
```

- It is recommended (but not mandatory) to prepare a test script including all the experiments.
- Timing in the code does not represent the correct execution time if you execute your code in the simulator. It is just to observe timing difference in case of executing the code in your machine, not in the simulator.
- You are required to submit a report that includes your results with graphs and comments about the results.
- You need to work individually, no group work is allowed.
- No late homework will be accepted.

OPTIONAL PART:

For extra credit (up to 50 points), you can run GPU experiments and submit additional results. Specifically, you need to execute CUDA matrix multiplication codes (both naive and tiled version) on your GPU-enabled computer or colab as introduced in your lab. You should execute codes with 1024 and 2048 input sizes, obtain execution times and prepare a report including a speedup graph (graph similar to one discussed in the lecture).

Submission: You are required to submit your report to cloud-lms in pdf format named as yourstudentnumber_P4.pdf (e.g. If your student number is 201812345678, the file name must be 201812345678_P4.pdf).

References:

[1] Pin - A Dynamic Binary Instrumentation Tool, https://software.intel.com/en-us/articles/pin-a-dynamic-binary-instrumentation-tool