

Lab 4

[Start Assignment](#)

Due Oct 9 by 11:59pm **Points** 100 **Submitting** a file upload

In this lab, we are going to experiment with the p-chasing method that we have previously learned.

The implementation is provided to you in the lecture slides. For your convenience, it is shown below:

```
__device__ void P_chasing2(int *A, long long int iterations, int starting_index){

    __shared__ long long int s_tvalue[1024 * 4];
    __shared__ int s_index[1024 * 4];
    int j = starting_index;

    long long int start_time = 0;
    long long int end_time = 0;
    long long int time_interval = 0;

    asm(".reg .u64 t1;\n\t"
        ".reg .u64 t2;\n\t");

    for (long long int it = 0; it < iterations; it++){
        asm("mul.wide.u32 t1, %2, %4;\n\t"
            "add.u64 t2, t1, %3;\n\t"
            "mov.u64 %0, %clock64;\n\t"
            "ld.global.u32 %1, [t2];\n\t"
            : "=l"(start_time), "=r"(j) : "r"(j), "l"(A), "r"(4));

        s_index[it] = j;

        asm volatile ("mov.u64 %0, %clock64;": "=l"(end_time));

        time_interval = end_time - start_time;
        s_tvalue[it] = time_interval;
    }
}
```

```

    }
}

```

However, this is only the device part of the code and it may not be enough for you to observe the actual output. What you might want to add are:

- 1) Some more arguments for the function.
- 2) Some code in the function to offload the timing measurement to the host.
- 3) The host side code for the kernel launch (you should use <<<1,1>>> as the launch parameter).
- 4) The host side code for post-processing.
- 5) The host side code to initialize A. For example:

```

void init_cpu_data(int* A, long long int size, int stride, long long int mod){
    for (long long int i = 0; i < size; i = i + stride){
        A[i]=(i + stride) % mod;
    }
}

```

Next, what we want to profile here are some simple information:

- 1) The cache line size of the L1 cache
- 2) The cache line size of the L2 cache

You should use some data patterns in A to create hits and misses in the caches and find out the answer by observing the timing output. To make it easier for you this time, you can use the compiler options to enable/disable the L1 cache when targeting different caches.

Note that you can perform the profiling on your preferred GPU. However, make sure your parameters match the corresponding specifications.

Things to submit:

- 1) Your working code in a zip file (including your Makefile)
- 2) A text or PDF file showing your input, output, and explanation for your conclusions.