Performance Measurement

This lesson gives an overview of the performance measurement for the problem of thread-safe initialization of a singleton in C++.

I want to measure how expensive it is to access a singleton object. For reference timing, I will use a singleton which I will access 40 million times sequentially. Of course, the first access will initialize the singleton object. In contrast, the accesses from four threads will be done concurrently. I'm only interested in the performance numbers; therefore, I will sum up the execution time of the four threads. I will measure the performance using a static variable with block scope (Meyers Singleton), a lock `std::lock_guard`, the function `std::call_once` in combination with the `std::once_flag`, and atomics with sequential consistency and acquire release semantic.

The program will run on two PCs. My Linux PC with the GCC compiler has four cores while my Windows PC with the cl.exe compiler has two. That being said, I compile the program with maximum optimization. Now, I want to answer two questions:

1. What are the performance numbers of the various singleton implementations?
2. Is there a significant difference between Linux (GCC) and Windows (cl.exe)?

Finally, I will collect all the numbers in a table. You will be seeing it in the conclusion.