PThreads Comparison

Lucy Gaudlin

March 2, 2023

I used three types of thread synchronizations, no synchronization, PThread locks, and Test-and-Set locks, to show how fast each type of comparison is. For each type of thread synchronization, I compared the result of the program, as well as the time taken for the program to run, using the default amount of threads (4), 8 threads, and 16 threads. For each amount of threads, I used different increment amounts per thread, using the default amount (10,000), one million (1,000,000) increments, and one billion (1,000,000,000).

1 No Synchronization

The first type of synchronization type used in this experiment is actually no synchronization between each thread.

1.1 4 threads

The first test using no synchronization has 4 threads.

- For the default increments per thread, it takes 6 milliseconds to produce a result of 40,000.
- For one million increments per thread, it takes 15 milliseconds to produce a result of 4,000,000.
- For one billion increments per thread, it takes 7.9 seconds to produce a result of -294,967,296.

1.2 8 Threads

The second test using no synchronization uses 8 threads.

- For the default increments per thread, it takes 7 milliseconds to produce a result of 80,000.
- For one million increments per thread, it takes 24 milliseconds to produce a result of 8,000,000.

- For one billion increments per thread, it takes 15.7 seconds to produce a result of -589,934,592.

1.3 16 Threads

The third test using no synchronization uses 16 threads.

- For the default increments per thread, it takes 7 milliseconds to produce a result of 160,000.
- For one million increments per thread, it takes 46 milliseconds to produce a result of 16,000,000.
- For one billion increments per thread, it takes 31.5 seconds to produce a result of -1,179,869,184

2 PThread Locks

The second synchronization type used in this experiment is PThread locks.

2.1 4 Threads

The first test using PThread locks uses 4 threads.

- For the default increments per thread, it takes 6 milliseconds to produce a result of 40,000.
- For one million increments per thread, it takes 13 milliseconds to produce a result of 4,000,000.
- For one billion increments per thread, it takes 7.8 seconds to produce a result of -294967296.

2.2 8 Threads

The second test using PThread locks uses 8 threads.

- For the default increments per thread, it takes 5 milliseconds to produce a result of 80,000.
- For one million increments per thread, it takes 23 milliseconds to produce a result of 8,000,000.
- For one billion increments per thread, it takes 15.8 seconds to produce a result of -589,934,592.

2.3 16 Threads

The third test using PThread locks uses 16 threads.

- For the default increments per thread, it takes 8 milliseconds to produce a result of 160,000.
- For one million increments per thread, it takes 41 milliseconds to produce a result of 16,000,000.
- For one billion increments per thread, it takes 31.7 seconds to produce a result of -1,179,869,184.

3 Test-and-Set Locks

The last synchronization type used in this experiment is Test-and-Set locks.

3.1 4 Threads

The first test using Test-and-Set locks uses 4 threads.

- For the default increments per thread, it takes 5 milliseconds to produce a result of 40,000.
- For one million increments per thread, it takes 15 milliseconds to produce a result of 4,000,000.
- For one billion increments per thread, it takes 8 seconds to produce a result of -294,967,296.

3.2 8 Threads

The second test using Test-and-Set locks uses 8 threads.

- For the default increments per thread, it takes 6 milliseconds to produce a result of 80,000.
- For one million increments per thread, it takes 25 milliseconds to produce a result of 8,000,000.
- For one billion increments per thread, it takes 15.7 seconds to produce a result of -589,934,592.

3.3 16 Threads

The third test using Test-and-Set locks uses 16 threads.

- For the default increments per thread, it takes 6 milliseconds to produce a result of 160,000.
- For one million increments per thread, it takes 42 milliseconds to produce a result of 16,000,000.
- For one billion increments per thread, it takes 31.3 seconds to produce a result of -1,179,869,184.

4 Results

In conclusion, the result produced by each lock did not vary when the same parameters were used. The different types of locks also do not impact performance much, with most time being similar regardless of what type of lock was used.