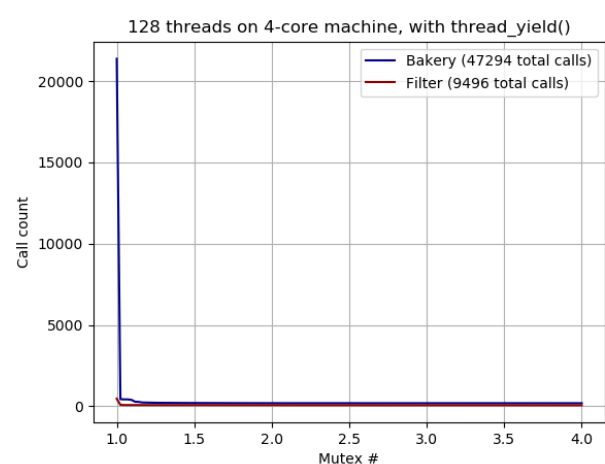
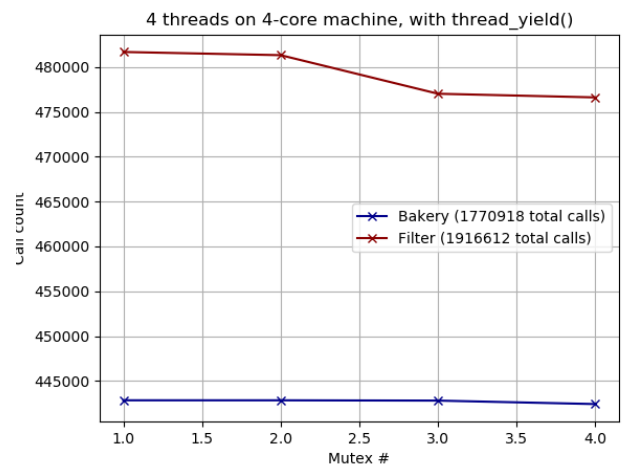
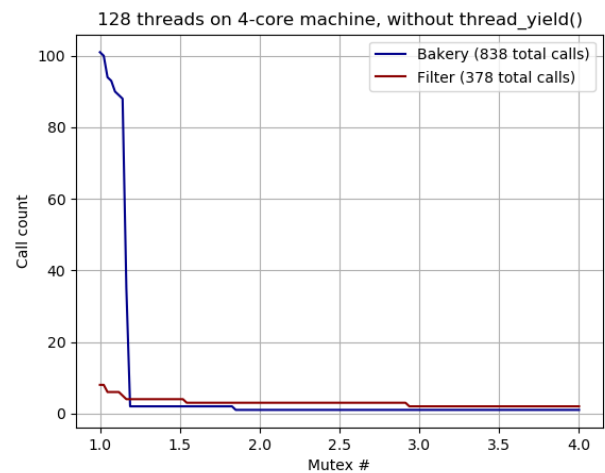
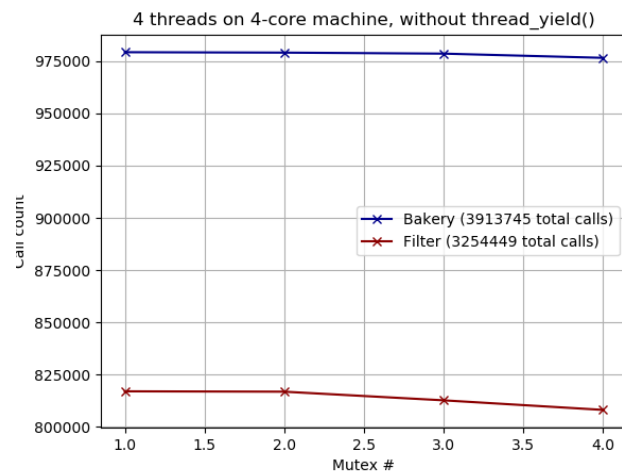


Ari 'aith' I.



In part 1 I implemented 2 different mutexes: the Filter Lock and the Bakery Lock; once using the yield() function and once without. Using yield() is part of an optimization known as **backoff**, and it's known to work best when there are many more threads than cores. In the version without yield, I found that the locks performed around 3.6 million calls, with a pretty Fair distribution of load between the threads. On the other hand, running 128 threads with yield() demonstrated very poor performance- resulting in <1000 calls in both the Bakery and the Filter implementations. Of course, this is 4x the amount of cores my machine has, but **most of the performance loss can be attributed to brute-force, iterative searches and comparisons (based on thread count) found in both implementations.** A better implementation would avoid $O(n^2)$ comparisons.

Interestingly, using `yield()` cuts the performance in half *in the 4-threaded version*, yet boosts the performance of the 128-threaded version by a factor $>10\times$. The `yield()` function's impact can vary from OS to OS, and in this case I found it to scale well with thread count. I attribute the performance gain to an efficient sleeping of threads being performed under the hood. Of course, with a lot of threads, most threads will be sleeping when waiting for a long time, which explains the gain with 128 threads. However, with only 4 threads, such a protocol isn't needed since threads won't be stuck in the busy-wait loop for long, hence the decrease in performance.

Standard Deviation

	Bakery	Filter
4 Threads, no <code>yield()</code>	1237.387	4205.437
128 Threads, no <code>yield()</code>	21.258	1.0929
4 Threads, with <code>yield()</code>	204.962	2712.750
128 Threads, with <code>yield()</code>	1872.343	35.337

Better exemplified by the graphs, we find that the greatest variance occurs in the Bakery Lock with 128 threads, while the Filter lock stays relatively Fair. I find this behavior odd, and I attribute it to an error in the implementation. The Bakery Lock should be more Fair than the Filter because of its first-come-first-served nature, while in the Filter lock the victim check can be rewarded to one thread an arbitrary amount of times.