

For the small dataset, the latency was measured in both time and number of cycles. Time was calculated by dividing time elapsed in nanoseconds by the number of operations (1E8). Cycles were calculated by measuring the number of cycles it took to start the sync mechanism, perform the operation, and end the mechanism. Then the average number of cycles was calculated for each thread, and those values were averaged to compute the final result. Except for the vanilla function, each type of synchronization saw an increase in latency as the number of threads increased. With only one thread, atomic synchronization performed the best, and it was also the most resistant to the increase in threads, followed by spinlock. Semaphore performed the worst once concurrency was implemented, and even on just one thread the latency was very high. For the small dataset it seems the most advantageous option was atomic.

Type	Threads	Latency (nanoseconds/operation)	Latency (cycles/operation)
vanilla	1	14.83	20.36
vanilla	2	13.78	13.78
vanilla	4	8.96	23.55
mutex	1	43.16	99.39
mutex	2	85.73	423.81
mutex	4	92.83	985.88
semaphore	1	41.55	95.01
semaphore	2	200.31	1096.06
semaphore	4	270.90	2966.96
spinlock	1	31.90	67.83
spinlock	2	37.34	182.11
spinlock	4	53.58	487.98
atomic	1	23.34	43.81
atomic	2	25.27	118.83
atomic	4	26.90	270.62

For the large dataset, only throughput was measured. This was calculated by dividing the total number of operations performed (1E9) by the time elapsed in seconds. Once again, vanilla's trend is opposite the rest. Aside from this, each mechanism generally decreased in throughput as the concurrency increased, which makes sense because throughput and latency are inversely proportional. However, mutex and atomic both showed a slight increase in throughput when switching from 2 to 4 threads, which was not something reflected in the first table. Atomic once again performs the best as it yields the highest amount of throughput for any given thread count, with spinlock again close behind. And also like the first table, semaphore declines the most significantly when thread count is increased.

Type	Threads	Measured Throughput (ops/sec)
vanilla	1	68167422
vanilla	2	83343369
vanilla	4	110445907
mutex	1	23330421
mutex	2	9709425
mutex	4	10383054
semaphore	1	24081572
semaphore	2	5274079
semaphore	4	3612904
spinlock	1	31275394
spinlock	2	28768134
spinlock	4	15469306
atomic	1	41492906
atomic	2	36996272
atomic	4	37816606