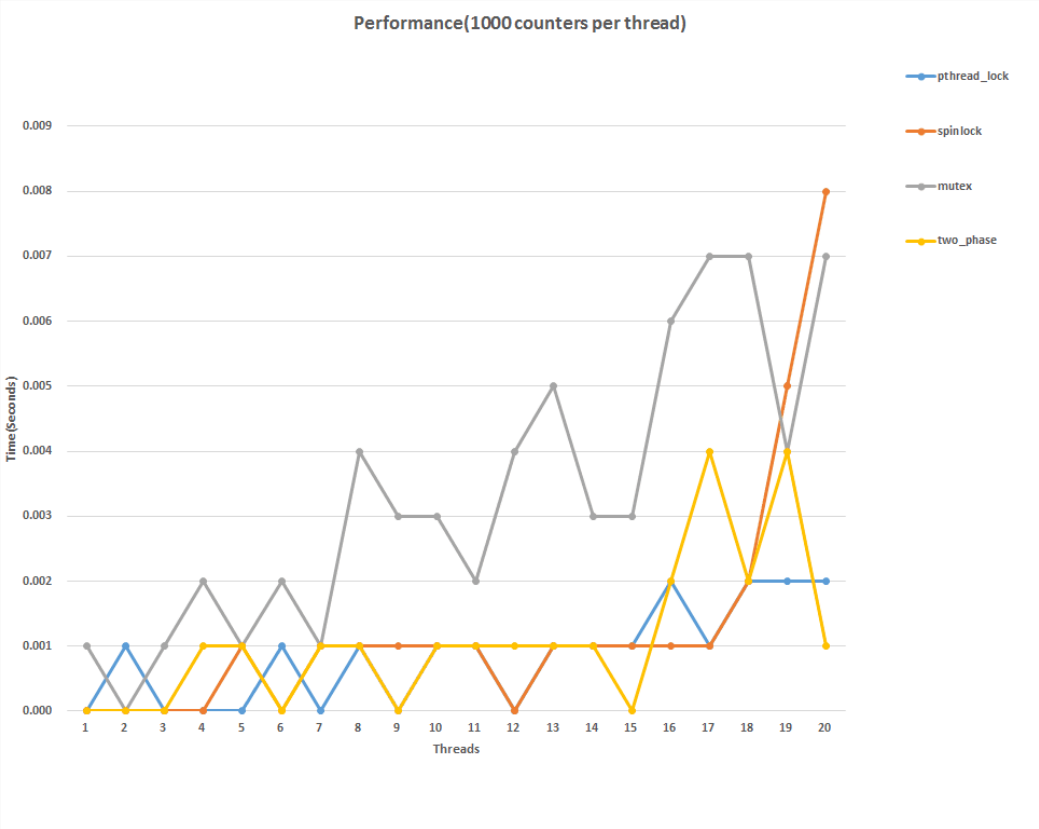
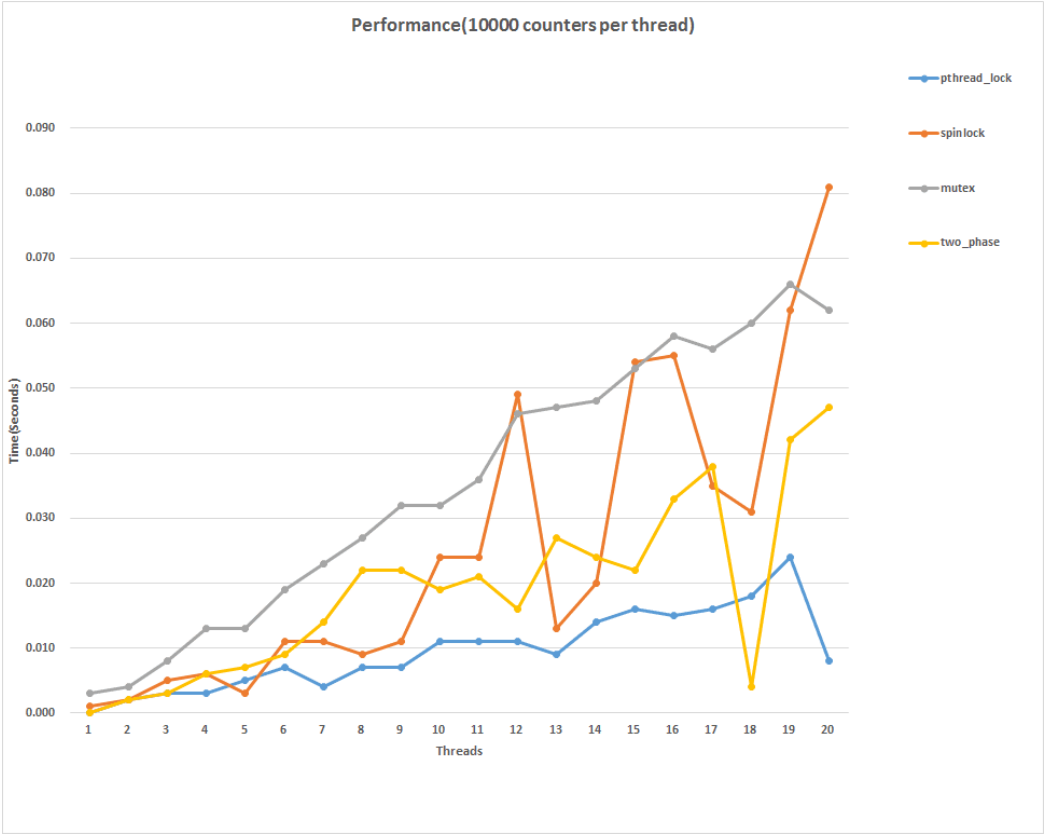
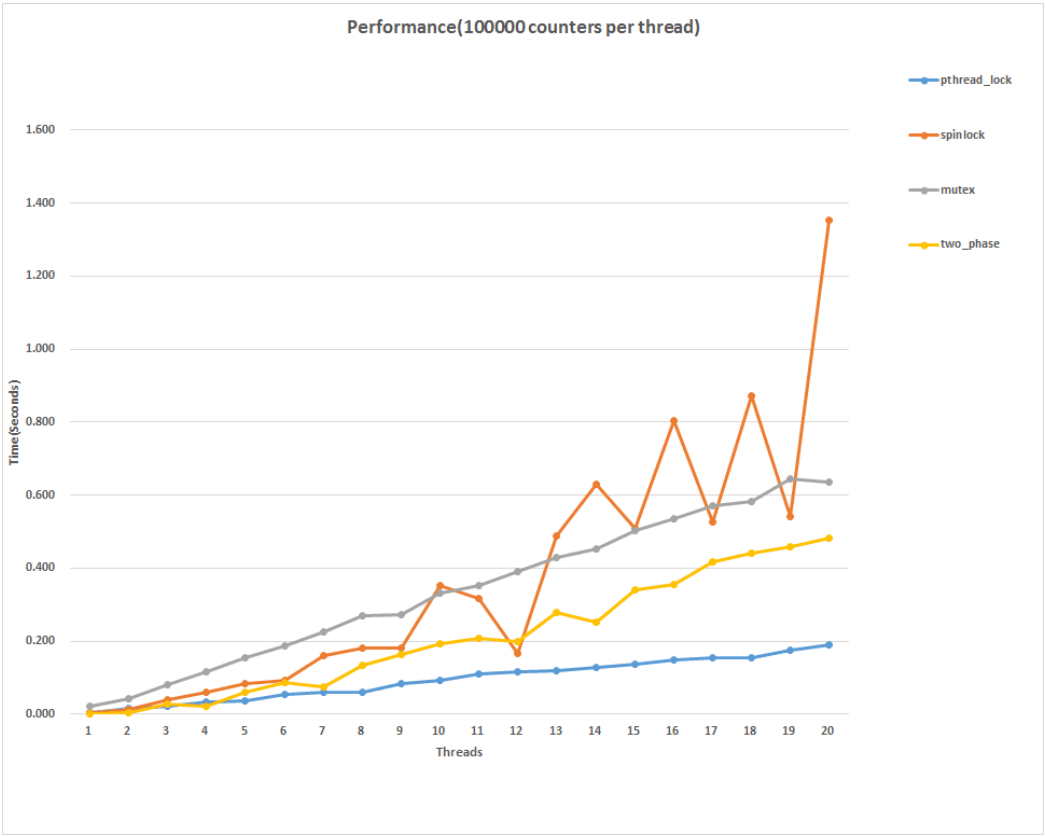
Performance Comparison

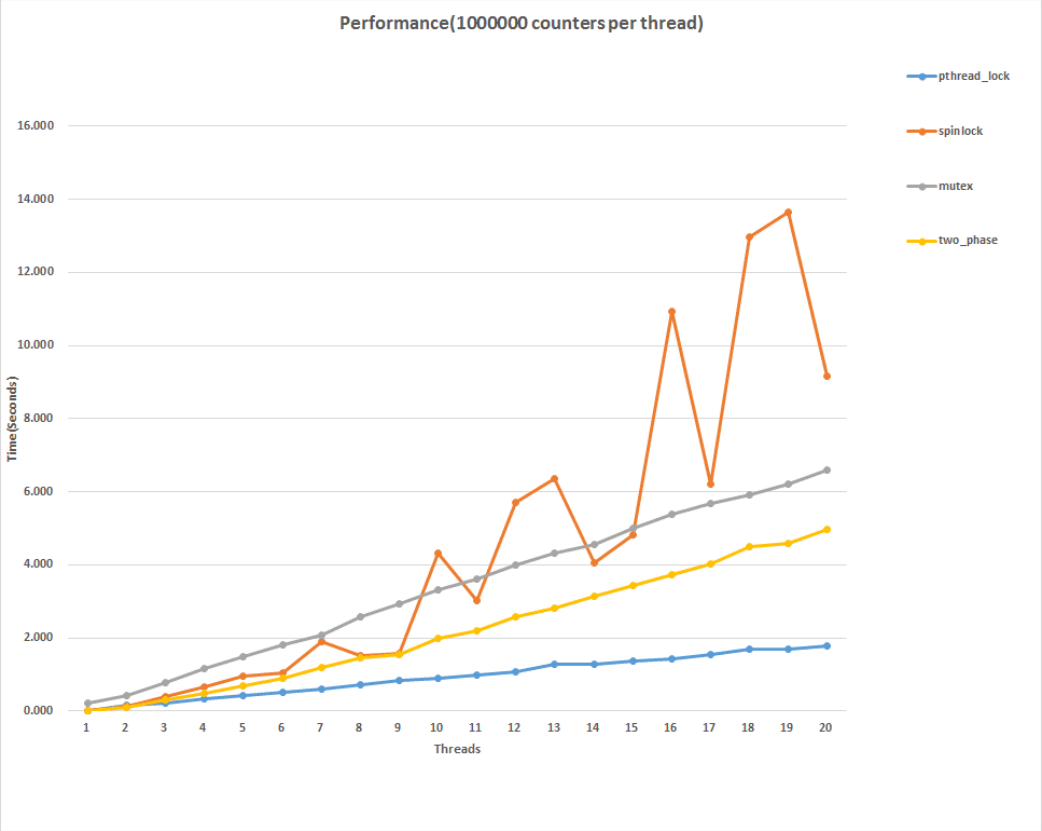
1. Counter

在 *counter* 数量较小时，四种锁的时间波动很大，但是总体上还是呈上升趋势。在 *counter* 数量很大时，时间趋于稳定。而且可以发现，如果线程数量较小，*mutex* 的时间是最长的，因为 *mutex* 需要不停的调用 *CPU* 。如果线程较多，四种锁的时间相对大小为 $spinlock > mutex > two\_phase > pthread\_lock$ 。



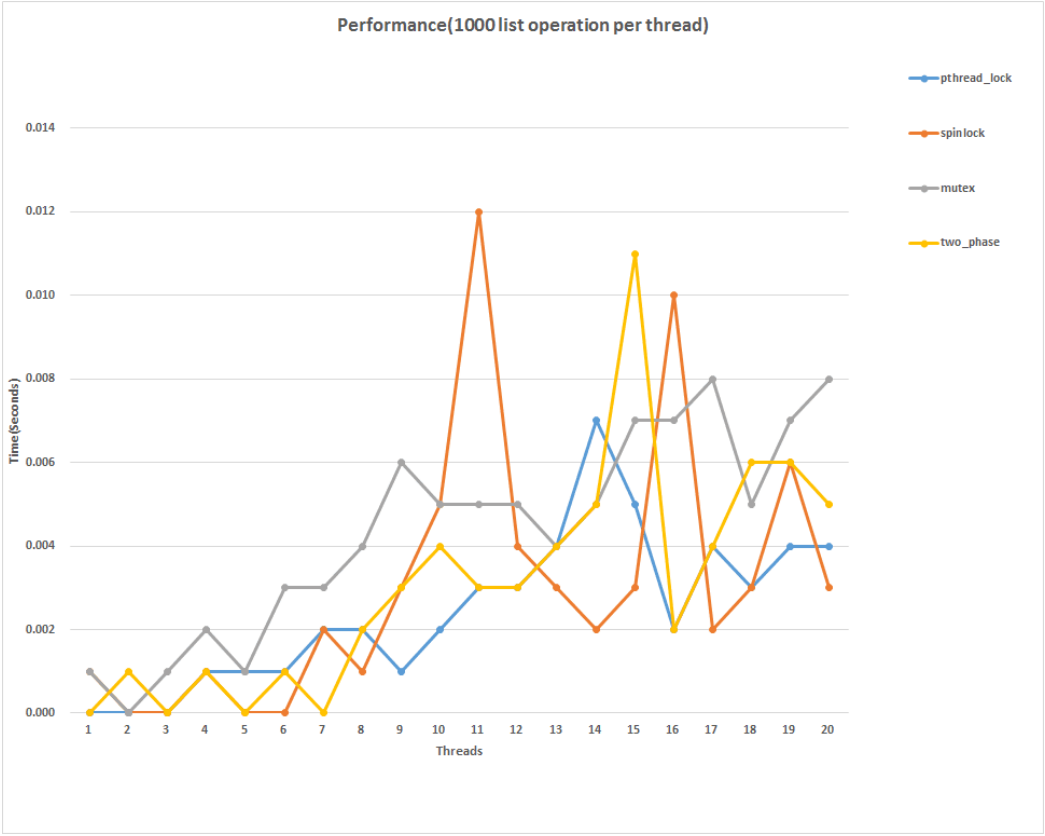

Performance(10000 counters per thread)
pthread_lock
spinlock
mutex
two_phase
Time(Seconds)
0.090
0.080
0.070
0.060
0.050
0.040
0.030
0.020
0.010
0.000
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20
Threads

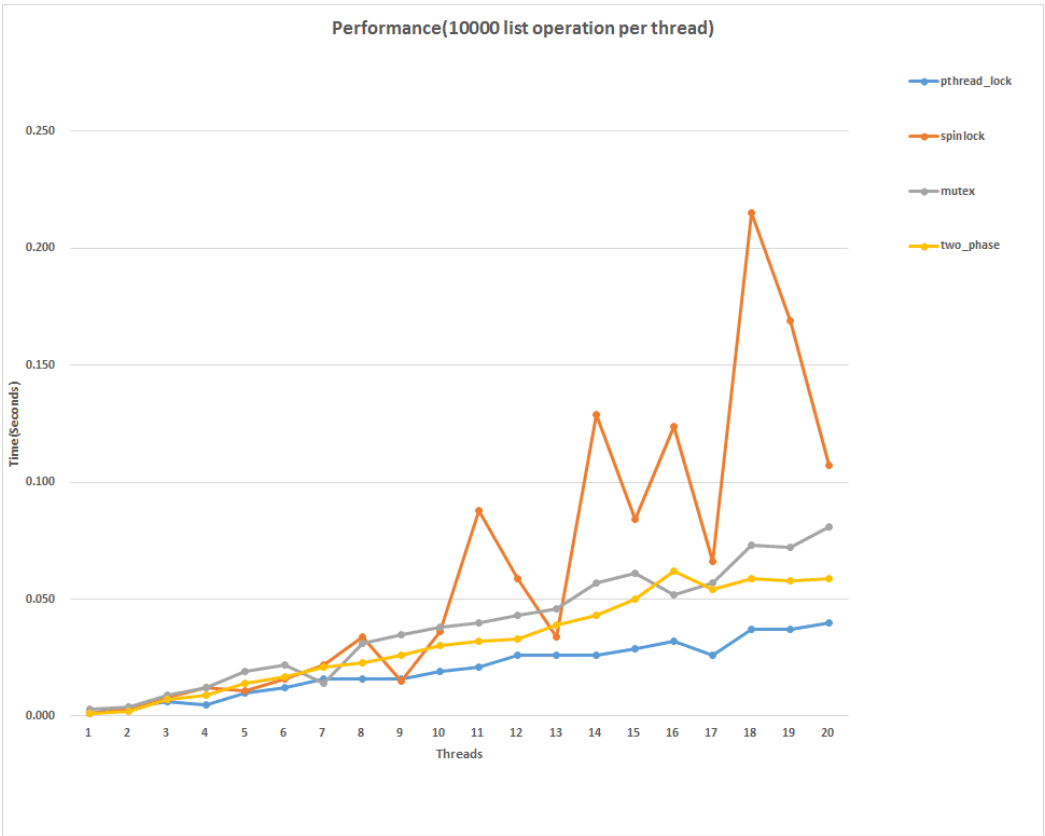

Performance(100000 counters per thread)
pthread_lock
spinlock
mutex
two_phase
Time(Seconds)
1.600
1.400
1.200
1.000
0.800
0.600
0.400
0.200
0.000
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20
Threads

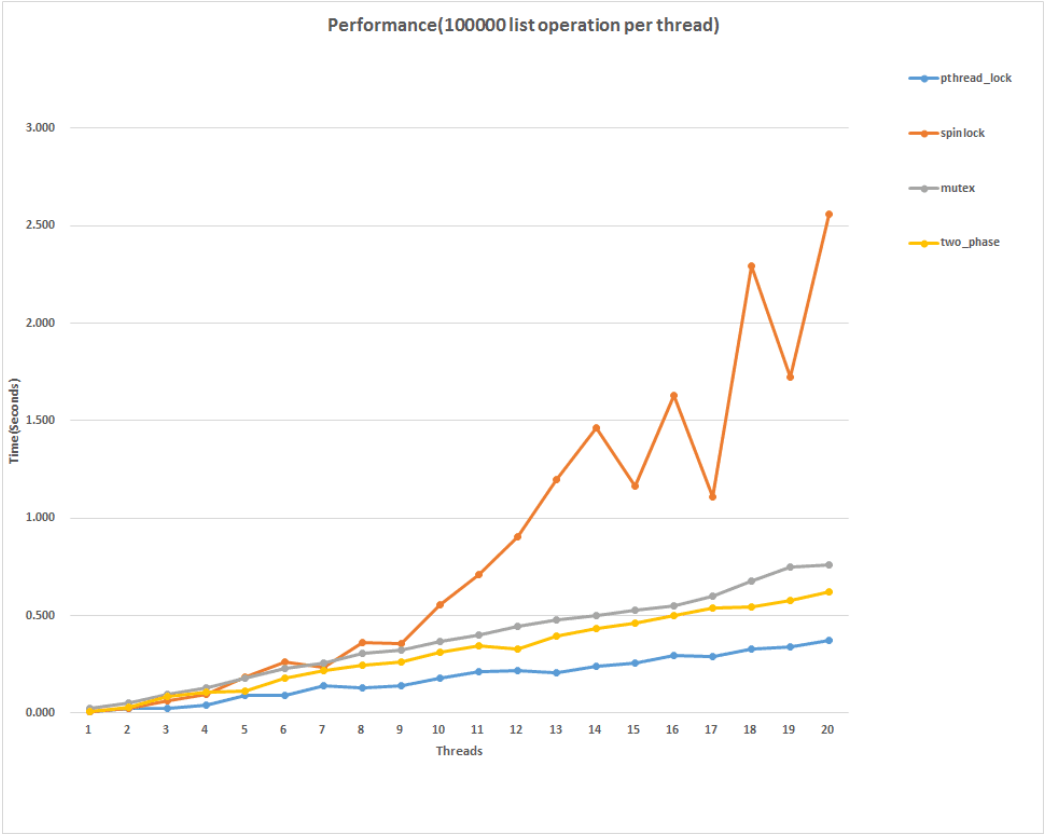


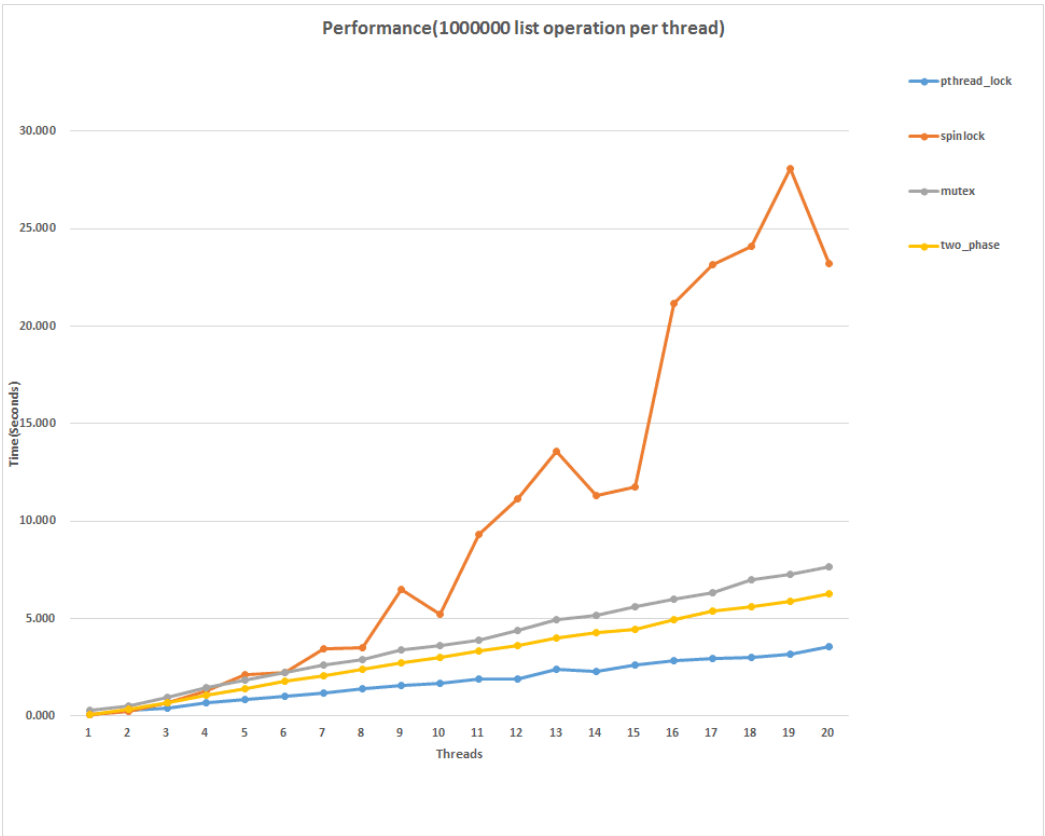
2. List

在 *list* 操作数量较小时，四种锁的时间波动很大，但是总体上还是呈上升趋势。在 *list* 操作数量很大时，时间趋于稳定。而且可以发现，如果线程数量较小，*mutex* 的时间是最长的，因为 *mutex* 需要不停的调用 *CPU* 。如果线程较多，四种锁的时间相对大小为 $spinlock > mutex > two\_phase > pthread\_lock$ 。


Performance(1000 list operation per thread)
pthread_lock
spin lock
mutex
two_phase
0.014
0.012
0.010
0.008
0.006
0.004
0.002
0.000
Time(Seconds)
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20
Threads

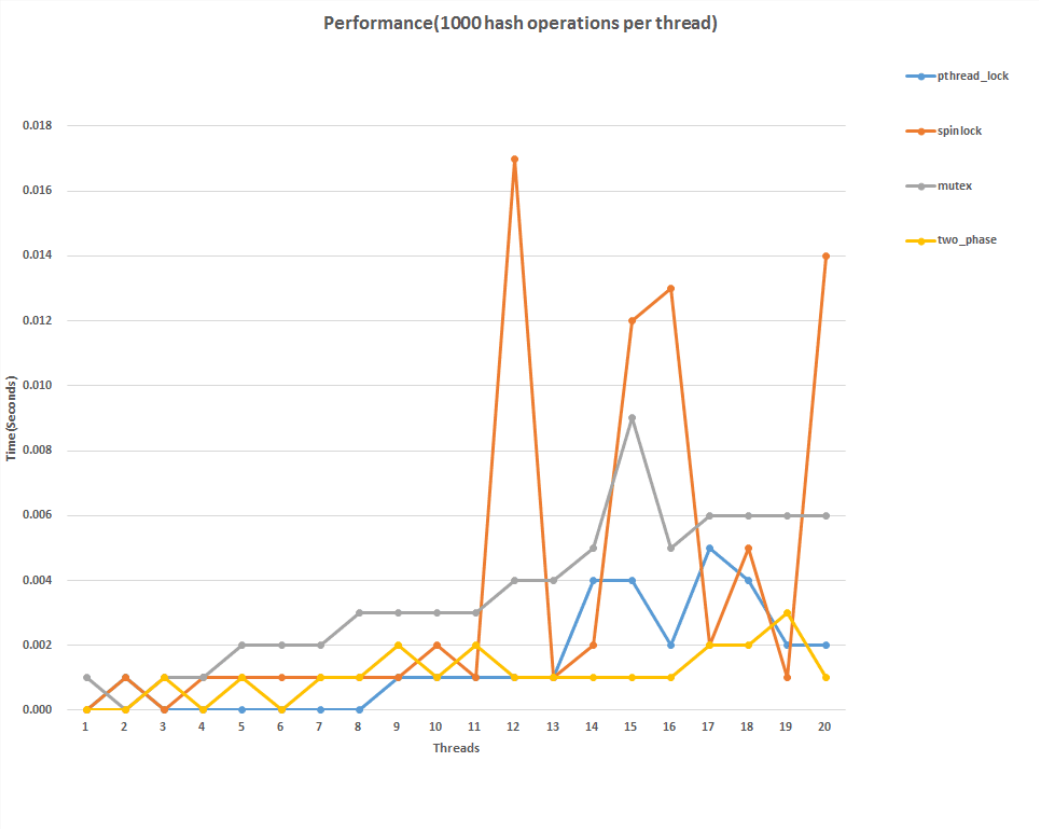

Performance(10000 list operation per thread)
pthread_lock
spin lock
mutex
two_phase
0.250
0.200
0.150
0.100
0.050
0.000
Time(Seconds)
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20
Threads

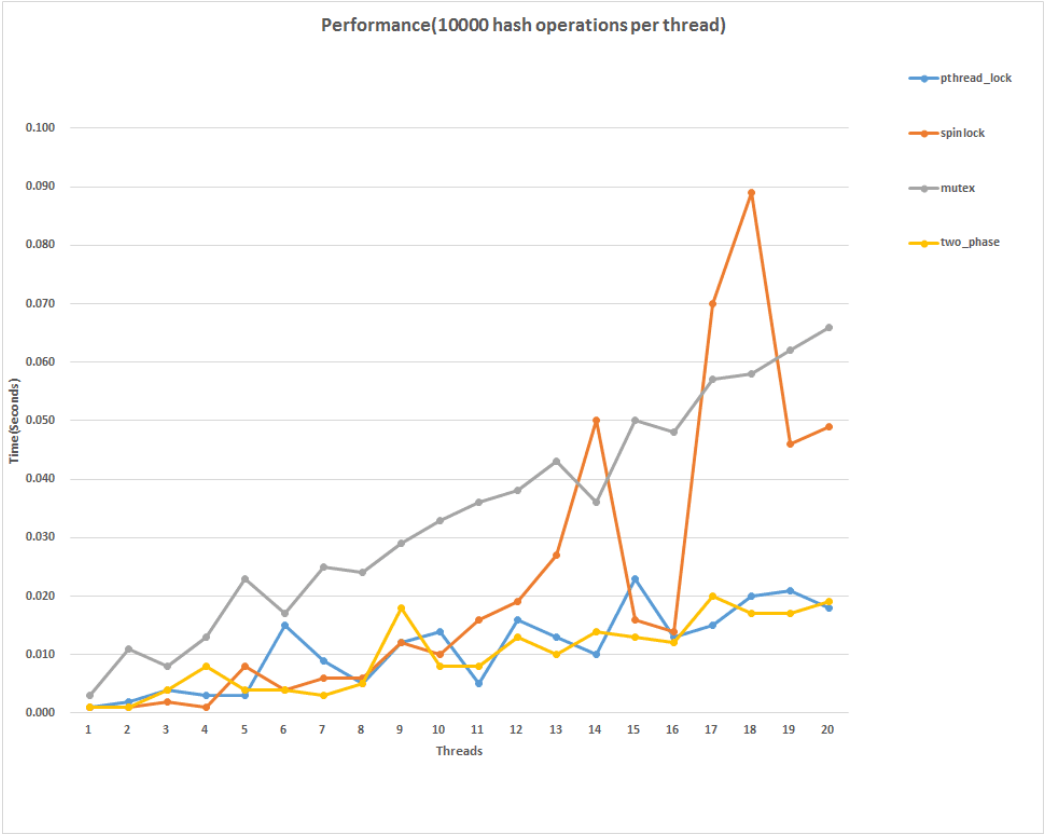

Performance(100000 list operation per thread)
pthread_lock
spinlock
mutex
two_phase
Time(Seconds)
3.000
2.500
2.000
1.500
1.000
0.500
0.000
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20
Threads

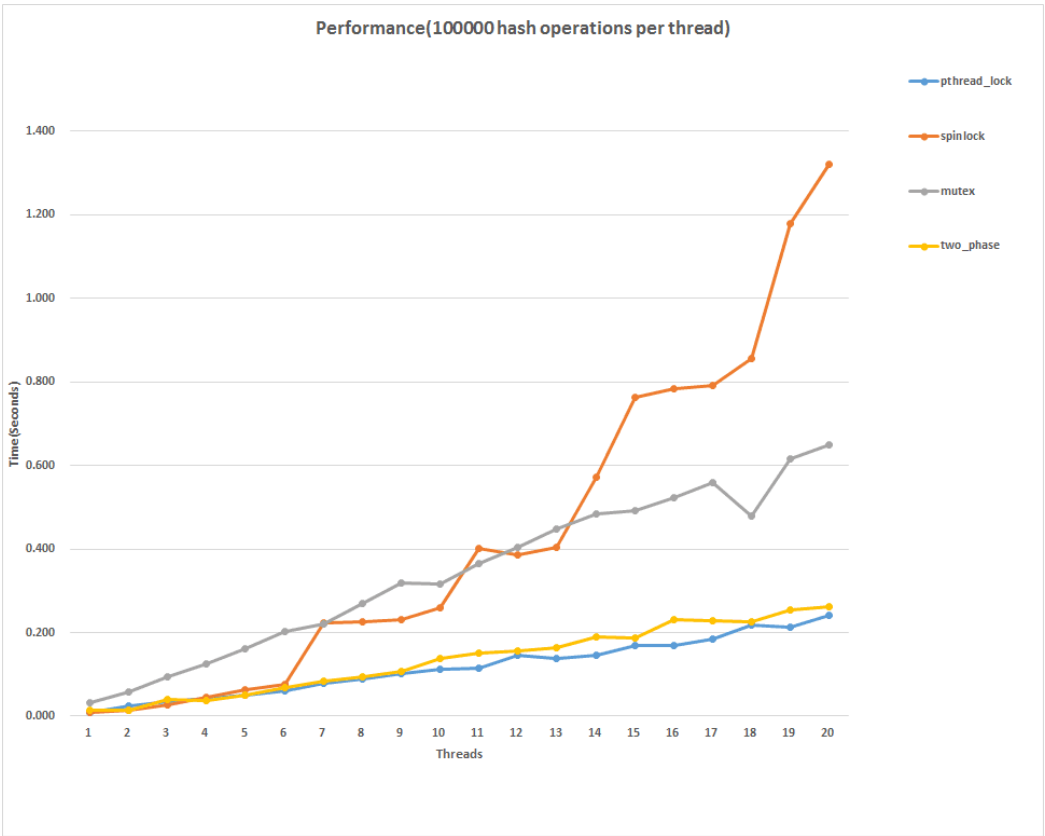

Performance(1000000 list operation per thread)
pthread_lock
spinlock
mutex
two_phase
Time(Seconds)
30.000
25.000
20.000
15.000
10.000
5.000
0.000
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20
Threads

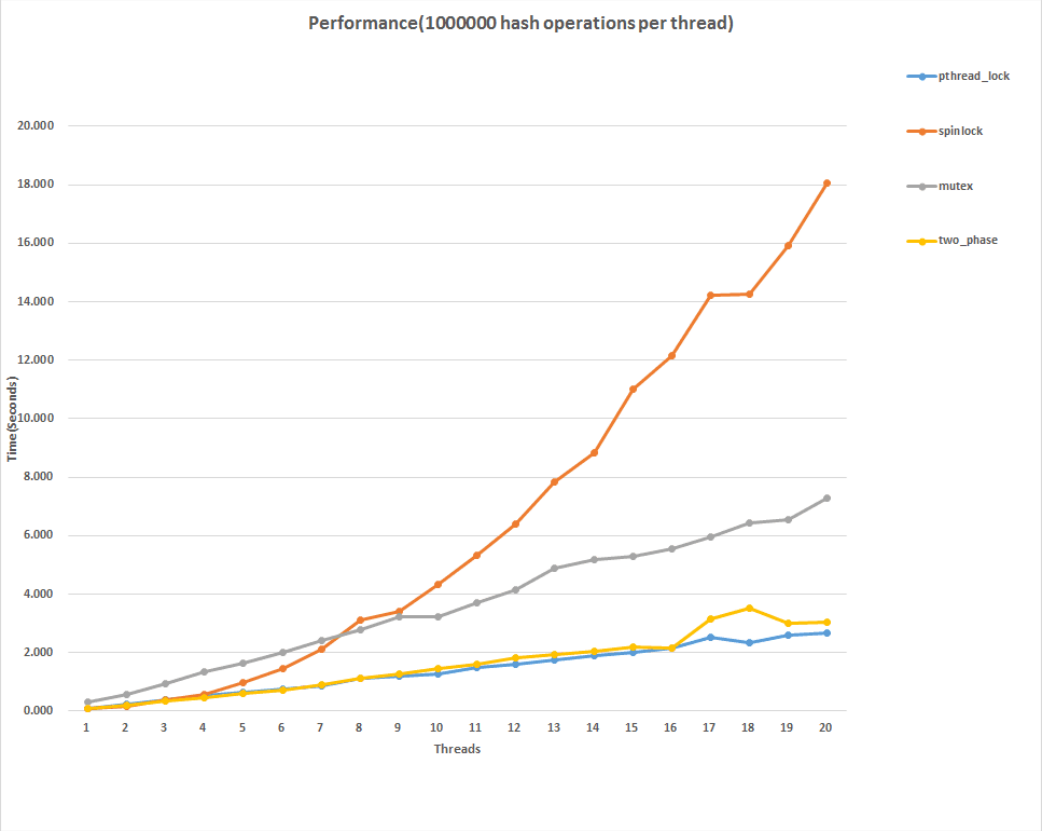
3. Hash

在 *hash* 操作数量较小时，四种锁的时间波动很大，但是总体上还是呈上升趋势。在 *hash* 操作数量很大时，时间趋于稳定。而且可以发现，如果线程数量较小，*mutex* 的时间是最长的，因为 *mutex* 需要不停的调用 *CPU* 。如果线程较多，四种锁的时间相对大小为 $spinlock > mutex > two\_phase > pthread\_lock$ 。




Performance(10000 hash operations per thread)
pthread_lock
spinlock
mutex
two_phase
Time(Seconds)
0.100
0.090
0.080
0.070
0.060
0.050
0.040
0.030
0.020
0.010
0.000
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20
Threads


Performance(100000 hash operations per thread)
pthread_lock
spinlock
mutex
two_phase
Time(Seconds)
1.400
1.200
1.000
0.800
0.600
0.400
0.200
0.000
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20
Threads



4. Conclusion

综合以上不同实现可以看出，当数据量足够大的时候，四种锁的时间 $spinlock > mutex > two\_phase > pthread\_lock$ 。但是当线程较少时，*mutex* 时间会较大，因为会大量调用 *CPU* 。

还可以看出一个奇怪的现象，数据量较大的，*spinlock* 的波动还是特别的明显。原因是因为我在测试的时候还开了其他进程，比如斗鱼直播伴侣。而这个进程特别耗 *CPU* ，所以导致了 *spinlock* 极度不稳定。