

1. 文档说明：

(1) Mutex 中是使用 fmutex 实现的锁及三个相关库的代码

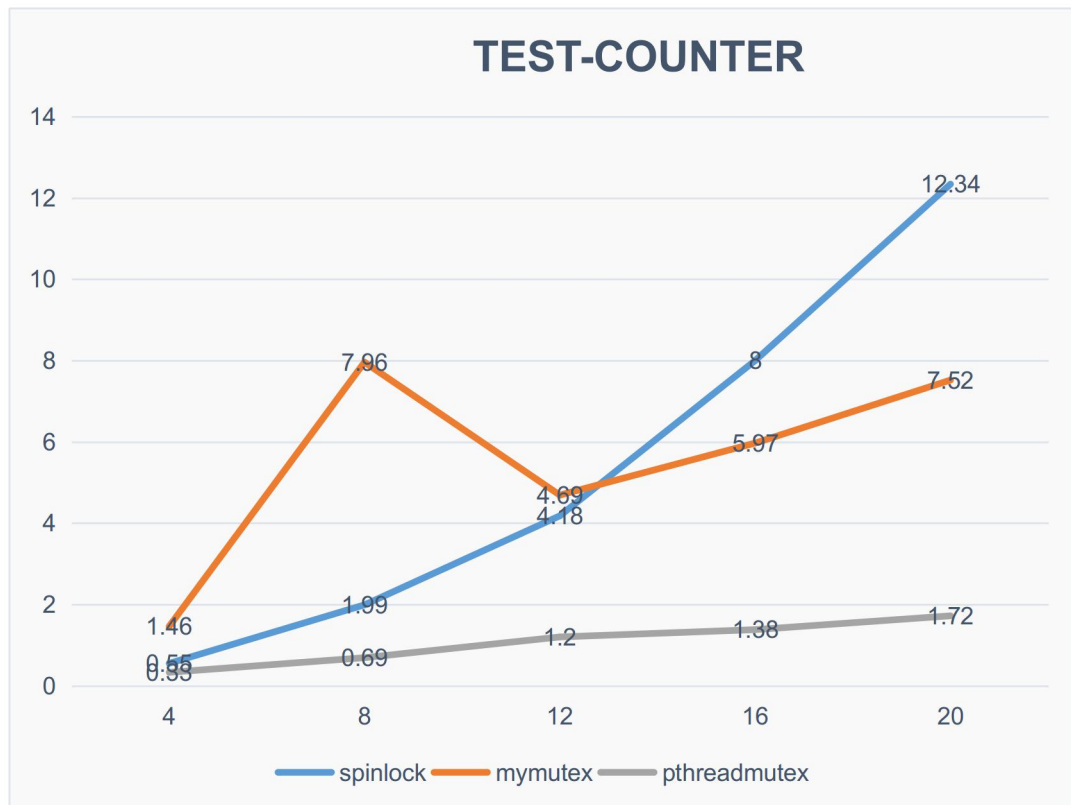
(2) Pthread\_lock 中是使用 pthread 自带的锁实现的

(3) Spinlock 是采用停等方式实现的锁

2. 实验环境：Ubuntu 16.04 4 核 4G 内存

3. 相关测试数据及图表

(1)

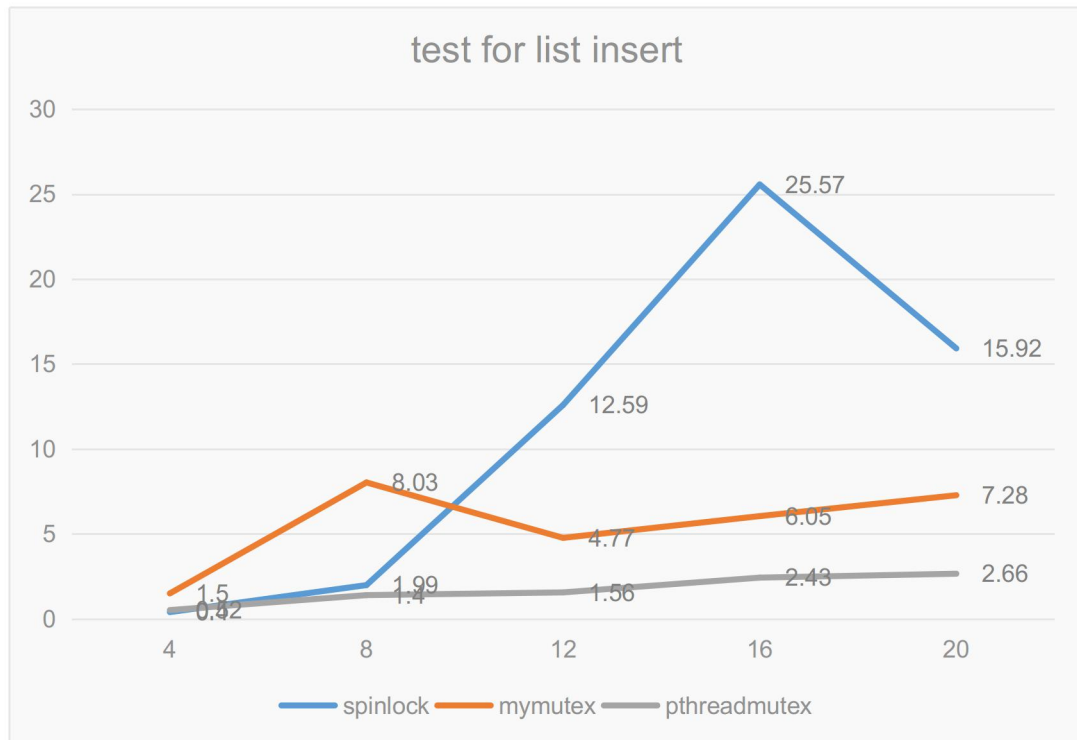


Each thread counter 1,000,000 times

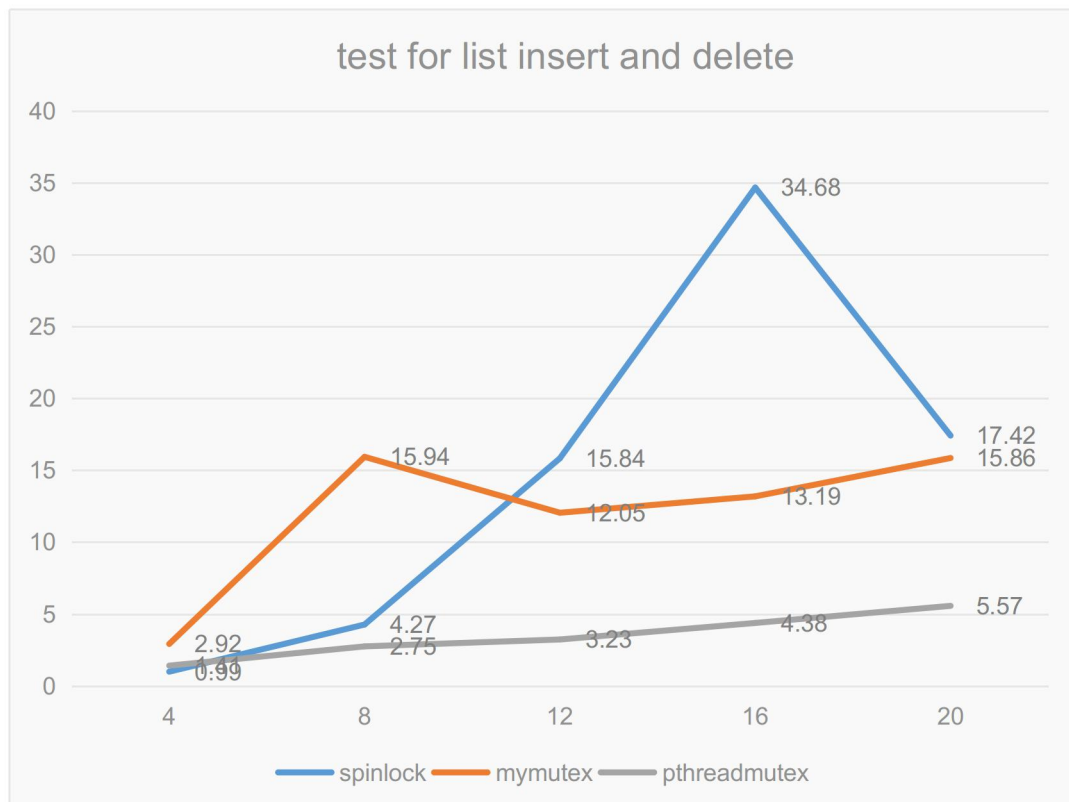
A for spinlock, B for mutex, C for pthread\_mutex

分析：当线程数少时，三者差距不大，但随着线程数增大，spinlock 的时间快速上升，mutex 的时间上升幅度较小，但在线程数为 8 时出现波动，原因不明。随着线程数增大，由于 mutex 引入了休眠制度，所以表现比忙等的好。

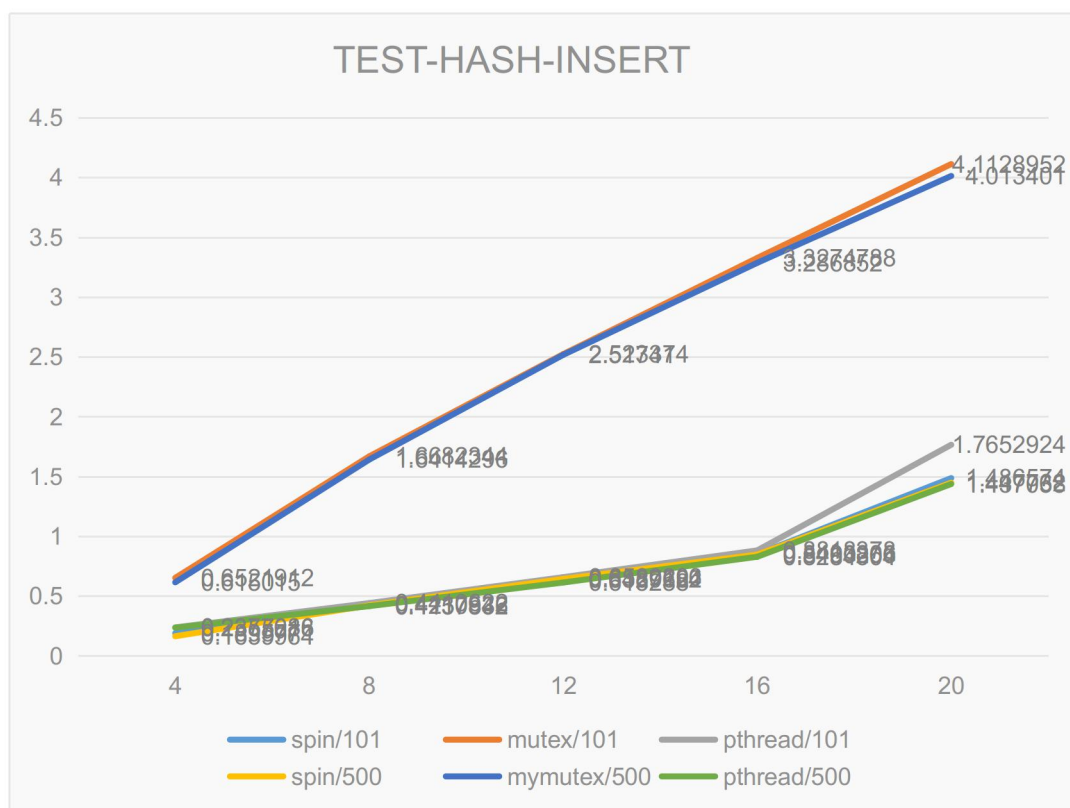
(2) test for list insert



(3) Test for list insert and delete

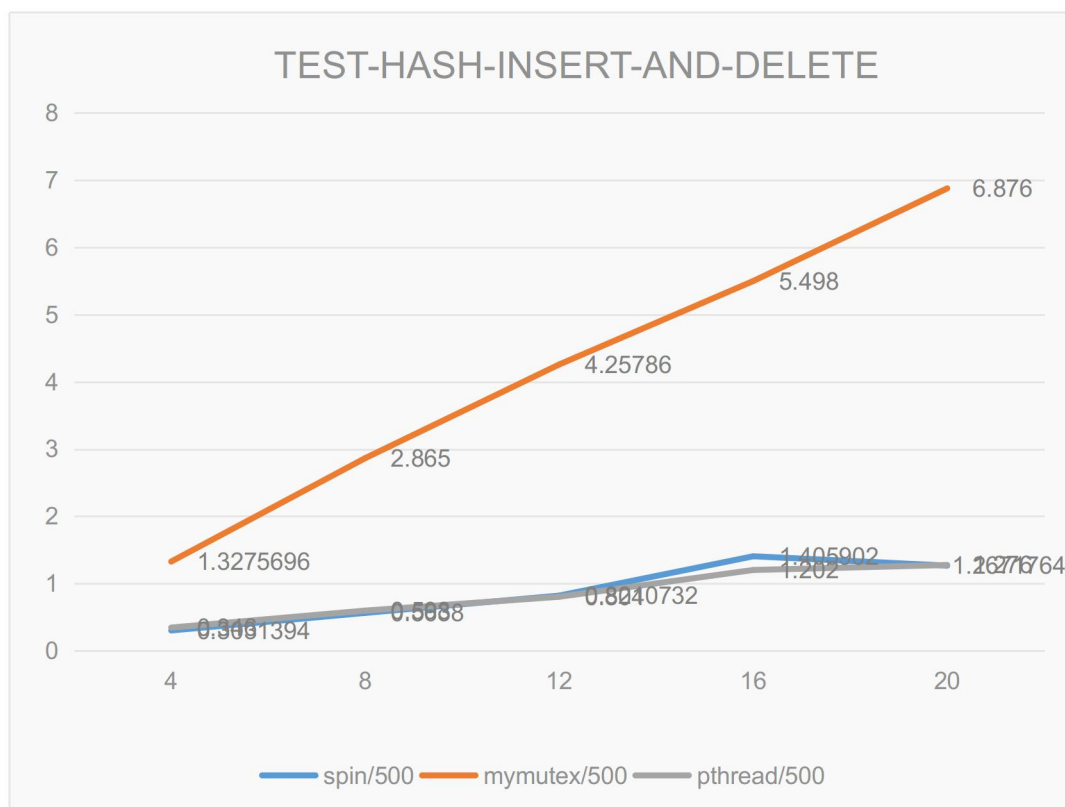


(4) Test for hash insert



插入的数据是每个线程的线程号，忙等情况少，所以由于少了维护休眠队列的开销，性能比 mutex 的好

#### (5) Test for hash insert and delete



Conclude : a. 可以看到 pthread 实现的锁的表现在三个数据结构中都很不错

Fairness :

运行过程时的 CPU 情况截图，可以推测线程间的任务分配比较均匀，此外，在测试 counter 时，在每个线程结束打印当前的 counter 的值，发现值都接近 1,000,000，所以 fairness 是比较合理的。

