链滴

有趣的优化问题

作者：tiangao
原文链接：https://ld246.com/article/1534134849164
来源网站：链滴
许可协议：署名-相同方式共享 4.0 国际 (CC BY-SA 4.0)

先上环境

```
$ java -version
java version "1.8.0_171"
Java(TM) SE Runtime Environment (build 1.8.0_171-b11)
Java HotSpot(TM) 64-Bit Server VM (build 25.171-b11, mixed mode)
```

测试代码

```
public class DemoTest {
    interface Op {
        int operate(int d);
    }

    static class M1 implements Op {
        @Override
        public int operate(int d) {
            return d + 1;
        }
    }

    static class M2 implements Op {
        @Override
        public int operate(int d) {
            return d + 1;
        }
    }

    public static void callMillion(Op op) {
        int d = 0;
        for (int i = 0; i < 1000000; i++) {
            op.operate(d);
        }
    }

    public static void main(String[] args) {
        CyclicBarrier cyclicBarrier = new CyclicBarrier(3);
        ExecutorService executorService = Executors.newFixedThreadPool(3);
        // 执行 100W 次 M1
        executorService.submit(() -> {
            try {
                cyclicBarrier.await();
            } catch (InterruptedException | BrokenBarrierException e) {
                e.printStackTrace();
            }
            long start = System.nanoTime();
            callMillion(new M1());
            long cost = System.nanoTime() - start;
            System.out.println("start:" + (start / 1000000) + ", test1: " + cost);
        });
        // 执行 100W 次 M2
        executorService.submit(() -> {
```

```
            try {
                cyclicBarrier.await();
            } catch (InterruptedException | BrokenBarrierException e) {
                e.printStackTrace();
            }
            long start = System.nanoTime();
            callMillion(new M2());
            long cost = System.nanoTime() - start;
            System.out.println("start:" + (start / 1000000) + ", test2: " + cost);
        });
        // 执行 100W 次真实逻辑
        executorService.submit(() -> {
            try {
                cyclicBarrier.await();
            } catch (InterruptedException | BrokenBarrierException e) {
                e.printStackTrace();
            }
            long start = System.nanoTime();
            int d = 0;
            for (int i = 0; i < 1000000; i++) {
                d += 1;
            }
            long cost = System.nanoTime() - start;
            System.out.println("start:" + (start / 1000000) + ", test3: " + cost);
        });
        executorService.shutdown();
    }
}
```

可以先心里想一下答案是什么，然后再验证一下，如果出了你的意料，欢迎讨论

原文链接：有趣的优化问题