

Hw11

杜心敏521021910952

插入 1 - `TestSize` 个key，查询10,000,000个合法的已经插入的key，记录时间。

并行插入时，由于设定了最大线程数，对插入数据进行了分组，但是没有处理余数的情况，所以要求最大线程数整除 `TestSize`

实验数据

1. Test Size = 10000

```
1 start test on sequential execution
2 sequential execution, put time: 0.000202
3 sequential execution,get time: 0.079186
4 start test on 20 threads execution
5 parallel execution, put time: 0.0015887
6 parallel execution, get time: 0.0144384
```

put操作加速比: 0.127148 (串行时间 / 并行时间)

get : 5.4844

2. Test Size = 50000

```
1 start test on sequential execution
2 sequential execution, put time: 0.001337
3 sequential execution,get time: 0.076910
4 start test on 20 threads execution
5 parallel execution, put time: 0.0011611
6 parallel execution, get time: 0.0177504
```

put操作加速比: 1.15149 (串行时间 / 并行时间)

get : 4.33296

分析

put操作的性能提升不明显，甚至在数据较小的时候，反而不如串行。可能是因为写操作存在大量竞争，线程都在抢锁，实际运行还是单线程。

get操作性能提升明显，多线程的读操作互不影响，发挥了多线程的优势。但是仍然低于CPU核心数，可能的原因是：

- 并行化引入了额外的开销。比如分配进程，计算每个进程get的值，检查进程是否结束。

NumberOfCores	NumberOfEnabledCore	NumberOfLogicalProcessors
6	6	12