HW9

522031910213 朱涵

May 8, 2024

1 实验结果

1.1 斐波那契第40个数

见下图。通过单线程递归和多线程递归跑出来的结果都是102334155。

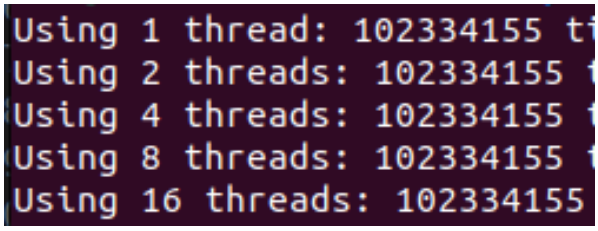

Figure 1: 程序运行结果

1.2 不同线程数下的性能

见下图。可以看到随着线程数的增多，性能也不断增优。

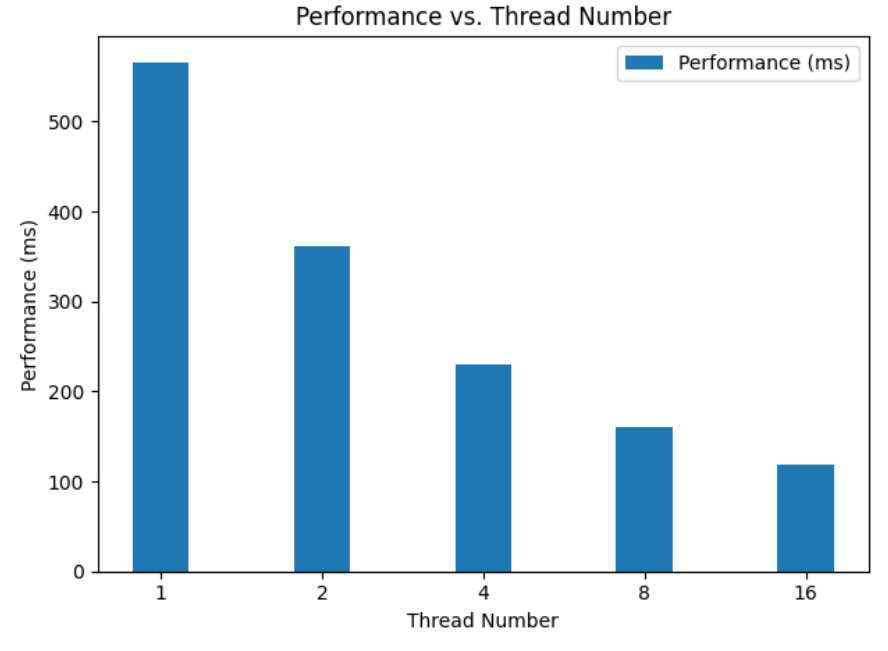

Figure 2: 不同线程数下斐波那契程序的性能柱状图

2 结果分析

多线程并发执行可以把原本串行的任务变为同时执行，从而减少耗时。对于递归版本的斐波那契计算来说，原本的单线程下每一项元素都需要**串行**的计算完前两项的元素，同样拿计算F(3)来举例，单线程下执行顺序是：**F(1)→F(2)→F(3)**（或者1和2交换顺序），而双线程就可以做到**F(1),F(2)同时计算**，省去了F(1)的计算时间。以此类推，线程数越多，也就可以省去越多的串行计算量（在硬件的限制允许内）。 不过，斐波那契的计算由于具有结果的依赖性，无法做到真正的并行（即把每一部分的工作分给不同线程执行，最终合并结果）。因此多线程也只能在**递归**的版本里进行性能的优化，如果是迭代计算的版本，是无需多线程并且仍能做到常数空间、线性时间复杂度的。