

# OpenMP 调度策略实验报告

容逸朗 2020010869

## 测量结果

- 在两个测例上分别应用 OpenMP 的 `static`、`dynamic` 和 `guided` 三种调度模式，测量结果如下：

| 调度模式    | 均匀长度分段用时/ms | 随机长度分段用时/ms |
|---------|-------------|-------------|
| static  | 68.3845     | 1414.37     |
| dynamic | 73.2568     | 1373.87     |
| guided  | 68.1761     | 1452.47     |

## 分析

- 对于排序任务而言，数据量越大，排序时间以  $n \log n$  的速度增长：
  - 均匀分布的情况下， $5 \times 10^7$  的数据被分为  $10^5$  组，每組约有 500 个数；
  - 对于随机长度的任务而言，程序利用正态分布生成 100 个位于区间  $[0, 5 \times 10^7]$  的数，再排序成为区间端点。一般情况下，第一个和最后一个数会位于  $1.3 \times 10^7$  及  $3.9 \times 10^7$  上，这导致了无论使用何种调度模式，程序也必需面对  $10^7$  量级的排序。
  - 在不考虑调度时间的情况下，执行一次  $n = 10^7$  排序的时间足以完成  $5 \times 10^4$  次规模为 500 的排序任务，因此两个任务的执行时间差距极大。
- 接下来考虑不同的调度模式对程序的影响，首先考虑均匀分布的测例：
  - `static` 和 `guided` 的时间相若，这是因为 `guided` 一开始分配的任务较多，以后减小，这和预先分配定量的 `static` 区别不大；
  - 而 `dynamic` 可能由于每次线程完成任务后都需要重新分配任务，故需要额外的调度时间；
- 对于随机长度的任务而言：
  - `dynamic` 每次重新分配可以使得每个进程完成的时间较集中，因此在数据量大的情况下速度较高；
  - `static` 方法只考虑任务的平均分配，因此仍然会对执行区间两侧的长时间任务的进程分配更多的数据，故速度较 `dynamic` 慢；
  - `guided` 算法一开始可能把多个任务量大的任务划分到同一进程中，故速度最慢。