

并行与分布式计算导论 作业 2

PDC 2023s Homework 2

截止期限 2023 年 4 月 9 日 23:59

DDL: 2023 Apr. 9 23:59 (GMT+8)

利用 **OpenMP** 并行编程,

本次作业需要利用 OpenMP 加速常见的图算法: 单源最短路径算法 (SSSP)。本次作业需要你自行设计并实现 **Dijkstra** 与 **Bellman-Ford** 两种 SSSP 算法的并行版本。

请实现以下三个任务, 并撰写书面报告, 简述你的实现思路, 测试结果和分析。将书面报告和代码源码共同打包为压缩包上传至教学网。

任务 1

完成 OpenMP 环境配置。将附件中压缩包上传至你要使用的机器, 解压后, 进入 /hw2/sssp 目录, 使用命令 `make`, 编译生成可执行文件 `sssp`。本次作业需要使用的数据集可以自行下载 (http://cs149.stanford.edu/cs149asstdata/all_graphs.tgz, 约 3GB)。可以在 hw2/sssp 目录中, 使用命令 `./sssp <path_to_graph>` 测试可执行文件 `sssp` 是否可以正确运行, 我们已经编写了串行版本的 Dijkstra 与 Bellman-Ford 算法可供参考, 你应该可以看到类似下图的运行结果。

```
-----
Your Code: Timing Summary
Threads  Par-Dijkstra          Par-Bellman-Ford
1:        0.000 (1.00x)        0.000 (1.00x)
2:        0.000 (14.57x)        0.000 (1.12x)
4:        0.000 (7.29x)         0.000 (1.12x)
8:        0.000 (8.50x)         0.000 (0.90x)
16:       0.000 (7.29x)         0.000 (1.12x)
32:       0.000 (7.29x)         0.000 (1.00x)
64:       0.000 (7.29x)         0.000 (1.00x)
96:       0.000 (7.29x)         0.000 (1.00x)
-----
Correctness:
Parallel Dijkstra is not Correct
Parallel Bellman-Ford is not Correct
```

任务 2

并行加速 Dijkstra 算法。修改 hw2/sssp/sssp.cpp, 利用 OpenMP 对函数 `dijkstra_parallel` 进行编写, 实现可并行加速版本的 Dijkstra 算法。你可以参考 hw2/common/graph.h 和 hw2/sssp/sssp.h 以熟悉程序中存储图的数据结构和接口。(我们保证读入的图数据不存在负权重的边)

任务 3

并行加速 Bellman-Ford 算法。修改 `hw2/sssp/sssp.cpp`，利用 OpenMP 对函数 `bellmanford_parallel` 进行编写，实现可并行加速版本的 Bellman-Ford 算法。你可以参考 `hw2/common/graph.h` 和 `hw2/sssp/sssp.h` 以熟悉程序中存储图的数据结构和接口。

提示：数据集中存在较大规模的图可能导致无法在有限时间内完成算法的求解。你可以选择对算法进行进一步的优化，或选择跳过这一部分数据。

撰写书面报告：在完成上述任务后，撰写报告阐述你的并行算法设计思路、编程思路、测试结果和分析，报告要求尽量**简洁**。**不要**粘贴代码截图。最终在教学网将**实验报告与源代码**打包提交。