



中山大學

SUN YAT-SEN UNIVERSITY

并行程序设计 with 算法实验

Lab8-并行多源最短路径搜索

姓 名 _____ 王志杰

学 号 _____ 22331095

学 院 _____ 计算机学院

专 业 _____ 计算机科学与技术

2025 年 5 月 14 日

1 实验目的

- 评估并行化最短路径算法的性能。
- 探索不同最短路径算法的并行适应性。

2 实验内容

实现并行的多源最短路径搜索，具体要求如下：

- 使用 OpenMP、Pthreads 或 MPI 中的任意一种并行编程模型来实现。

你选择的并行框架是：OpenMP

- 选用任意一种最短路径算法作为基础进行并行化，如 Bellman-Ford、Dijkstra（针对每个源点执行）、Floyd-Warshall 或 Johnson 算法。

你选用的算法是：Floyd-Warshall

- 简要描述所选算法的实现要点。

回答：

- **初始化：**构造 $n \times n$ 距离矩阵，使用权重或 ∞ 填充无边对角线为 0。
- **三重循环：**外层按中间顶点 k 迭代，内层双重循环更新所有 (i, j) 对应距离。
- **状态转移：**利用中间顶点 k 判断是否可通过 k 缩短 $i \rightarrow j$ 路径。
- **复杂度：**时间复杂度 $O(n^3)$ ，空间可优化为原地更新或交替使用两张矩阵。
- **负环检测：**若算法完成后存在 $d_{ii} < 0$ ，则图中含负权回路。

下面是我实现的外层并行 Floyd-Warshall 算法程序的部分核心代码

Listing 1: 并行 Floyd-Warshall

```
void floydWarshallOptimized() {
    for (int k = 0; k < n; k++) {
        // 只并行化外层循环，减少内存竞争
        #pragma omp parallel for schedule(static)
        for (int i = 0; i < n; i++) {
            if (dist[i][k] < INF) { // 提前检查避免无效计算
                for (int j = 0; j < n; j++) {
                    if (dist[k][j] < INF) {
                        double new_dist = dist[i][k] + dist[k][j];
                        if (new_dist < dist[i][j]) {
```


表 2: 数据集 2 (525 节点, 平均度数 27.98) 上 Floyd-Warshall 的并行性能 (串行时间 $T_s = 230$ 毫秒)

线程数量 (p)	运行时间 T_p (毫秒)	加速比 $S_p = T_s/T_p$	并行效率 $E_p = S_p/p$
1	230	1.00	1.00
2	158	1.46	0.73
4	82	2.80	0.70
8	56	4.11	0.51
16	26	8.85	0.55

3.2 并行性能分析

根据你的实验数据, 结合你选择的算法、并行框架、并行方式以及测试数据的特征 (节点数量、平均度数等), 分析程序的并行性能。可辅以图表 (如加速比曲线、效率曲线) 进行更直观的分析。

- 运行时间随线程数增加的变化趋势。

回答: 从表格和加速比曲线可以看出, 随着线程数 p 的增加, 运行时间 T_p 显著下降。对于数据集 1, T_p 从 138 ms 降至 56 ms; 对于数据集 2, T_p 从 230 ms 降至 26 ms。总体呈现快速下降到缓慢下降的趋势, 表明并行加速在低线程数时收益较大, 随后受内存带宽与同步开销影响, 降幅趋缓。

- 加速比的变化情况。

回答: 加速比 $S_p = T_s/T_p$ 随 p 增加而上升, 但并非线性。

数据集 1: $S_2 = 1.20, S_4 = 1.25, S_8 = 1.82, S_{16} = 2.46$;

数据集 2: $S_2 = 1.46, S_4 = 2.80, S_8 = 4.11, S_{16} = 8.85$ 。

数据集 2 的 S_p 更接近理想线性 ($S_p = p$), 因其图更密集, 计算量更大, 降低了线程间空闲与同步损耗。

- 并行效率的变化情况, 分析影响并行效率的主要因素。

回答: 并行效率 $E_p = S_p/p$ 随 p 增加而下降:

数据集 1: $E_2 = 0.60, E_4 = 0.31, E_8 = 0.23, E_{16} = 0.15$;

数据集 2: $E_2 = 0.73, E_4 = 0.70, E_8 = 0.51, E_{16} = 0.55$ 。

主要影响因素包括: 内存带宽瓶颈、线程创建与同步开销、负载不均衡。数据集 1 较稀疏, 单线程计算少, 开销占比高; 数据集 2 计算密集度高, 效率下降较慢。

- 不同数据特征 (如节点数量、平均度数) 对并行性能的影响。

回答:

- 节点数量：两数据集节点数比较，数据集 1 更大 (930 vs. 525)，但数据集 1 运行的更慢，说明节点数量非主要因素。
- 平均度数：数据集 2 平均度数 27.98 大于数据集 1 的 14.54，意味着更多的边更新操作，使得并行工作量更大，线程利用率更高，从而在相同 p 下获得更高的 S_p 与 E_p 。

注：实验报告格式参考本模板，可在此基础上进行修改；实验代码以 zip 格式另提交；最终提交内容包括实验报告 (pdf 格式) 和实验代码 (zip 压缩包格式)