

实验 1B: MPI 编程训练

图论中, 图 G 的一个连通分量是 G 的一个最大连同子图, 该子图中每一对节点中都存在至少一条路径。给定图 G , 如何找出其所有连通分量的问题称为连通分量问题。解决该问题的方法有三种:

- (1) 图搜索技术;
- (2) 通过图的布尔连接矩阵计算传递闭合路径;
- (3) 节点合并算法。

其中节点合并算法中, 起始图 G 中的 N 个节点被视为 N 个孤立的超节点 (super vertex)。算法执行过程中, 有边连通的超节点相继合并, 直到形成最后的连通分量。每个节点属于且仅属于一个超节点, 超节点中标号最小的被称为该超节点的根 (root)。

该算法的流程由一系列的循环组成。每次循环分为三步:

- (1) 发现每个节点的最小标号邻接超节点;
- (2) 把每个超节点的根连到最小标号邻接超节点的根上;
- (3) 所在第 (2) 步连接在一起的超节点合并成为一个更大的超节点。

算法中图 G 的节点总数为 N , 因为超节点的个数每次循环后至少减少一半, 所以把每个连通分量连接为单个超节点至多需要 $\log N$ 次循环。将节点 i 的超节点的根记为 $D(i)$, 初始时 $D(i)=i$ 。算法运行后, 所有处于同一个连通分量的节点具有相同的 $D(i)$ 。

为节点设置数值变量 D 和 C , 其中 $D(i)$ 是节点 i 所在的超节点号, $C(i)$ 为节点 i 或超节点 i 相连的超节点号。根据程序运行阶段的不同, 意义也会发生相应的变化。

算法主要分为 5 个步骤:

- (1) 各处理器并行为每个节点找出相应的 $C(i)$;
- (2) 各处理器并行为每个超节点找出最小邻接超节点, 编号放入 $C(i)$;
- (3) 修改所有 $D(i)=C(i)$;
- (4) 修改所有 $C(i)=C(C(i))$, 运行 $\log N$ 次;
- (5) 修改所有 $D(i) = \min(C(i), D(C(i)))$

代码参见 <http://cbb.sjtu.edu.cn/course/bi390/lab1/mmpi/connect.c>。

练习题

1. 给定一个 8×8 的邻接矩阵, 试着用程序找出连通分量:

0	0	0	0	0	0	0	1
0	0	0	0	0	1	1	0
0	0	0	0	0	0	0	0
0	0	0	0	0	1	0	1
0	0	0	0	0	0	1	1
0	1	0	1	0	0	0	0

0	1	0	0	1	0	0	0
1	0	0	1	1	0	0	0

2. 改写程序，使其能从文件中获取邻接矩阵。
3. 编写程序，在指定节点数的情况下，生成随机邻接矩阵，并计算其连通分量。
4. 针对不同节点数目，分别重复生成 1000 个随机邻接矩阵，指定不同的处理器数目，用上述程序处理。查看分析时间随节点数目、处理器数量的变化，并用图形表示。
5. 编写 PBS 脚本，用 qsub 提交脚本并运行。
6. 下载 mpiBLAST，阅读代码，并阐述其并行策略。用其自带的示例数据进行分析，计算加速比。