并行计算实验报告 – 3

# 串行实现

串行实现按照作业描述中的过程实现即可，代码在serial.cpp中。串行程序接受一个参数，用来指示矩阵的维度（设定矩阵为方阵）。第17-53行为若干数组申请空间，并从文件中读入矩阵A和向量b；56-59行初始化x0、r0、p0等；62-118行为主要迭代步骤，依次计算Ap、alpha、x、r、res、r、beta、p。

生成随机矩阵的程序使用了第一个作业的python程序，但有一些修改，追加了b数组的生成，代码见Random\_matrix\_modified.py。Python随机矩阵生成器将数据输出至input.dat文件。

由于矩阵具有随机性，为了避免长时间不能满足残差要求，所以设定最大迭代次数1000次。

# 并行改造

算法的核心就是一个迭代循环，第一想法就是将其并行化，但由于迭代步骤之间是有依赖的，所以不能直接将最外层循环并行化。所以将内部计算各个变量的次一级循环并行化。代码见omp.cpp。在第70/82/91/100/111/121行插入编译指导语句即完成并行化改造。

# 结果分析

矩阵规模为N=1000。并行运行效率如下图所示：

与第二个作业类似，小于等于16个线程时效率随着线程数的增长而上升，这是因为CPU硬件线程数量的限制。总体来说并行化改造带来的性能提升很明显的。