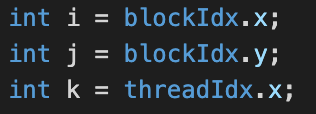
并行计算与分布式计算基础

第二次作业报告

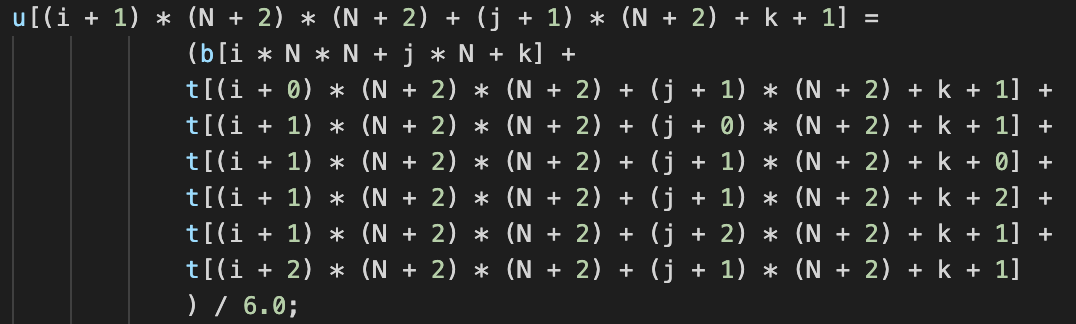
一、实验思路

本次作业中采用了cuda的方式进行加速，并且将Gauss- Seidel迭代退化成为了Jacobi迭代，成功降低了运行时间，具体的代码思路如下。主要的运算集中在Jacobi迭代时新的u向量的更新，和每次向量更新后残差的计算。

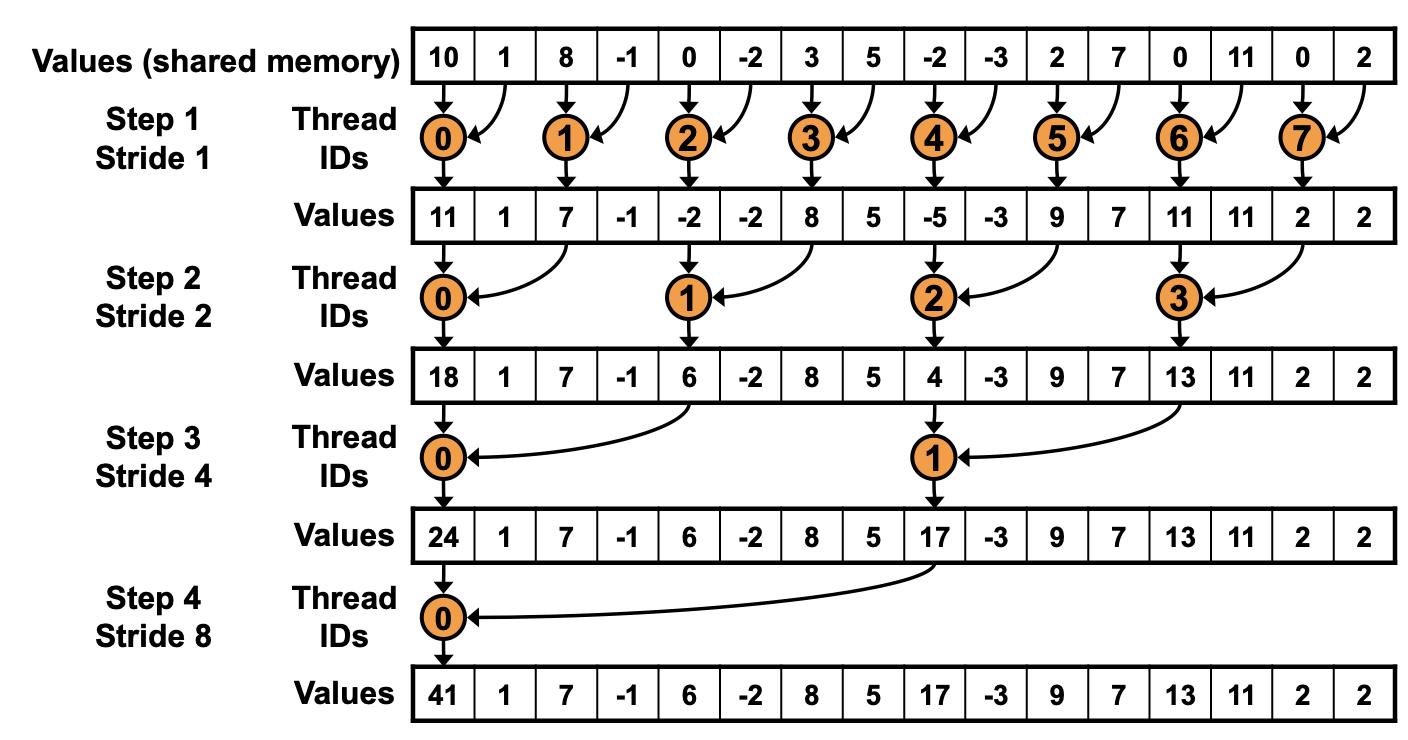
在Jacobi迭代的步骤中，我们将grid分为了（512，512，1）维，将block分为了（512，1，1）维，其中每一个线程会有三个不同的变量，如下所示：



其中i、j、k遍历了0-511中的每个数，且恰好与样例程序中的对应。也即每次Jacobi迭代更新后的数组（用u表示）被更新前的数组（用t表示）更新的运算如下：

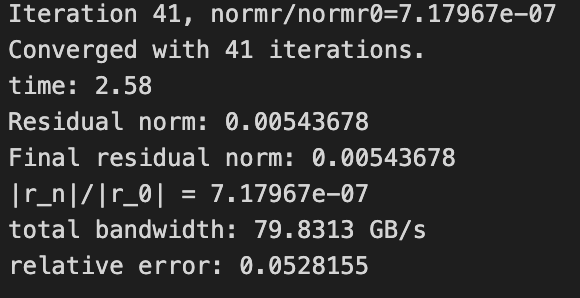


在残差的计算中，是一个大规模的数组规约操作。在这里我们使用了Nvidia官方文档中说明的规约操作，也即使用一个类似于二叉树的结构讲数据归到一个点，具体的操作流程示意图如下所示：



二、实验结果

最终，程序的运行结果在2.5s左右，加速比约为12，程序最终输出结果如下所示：



具体的代码在poison.cu中