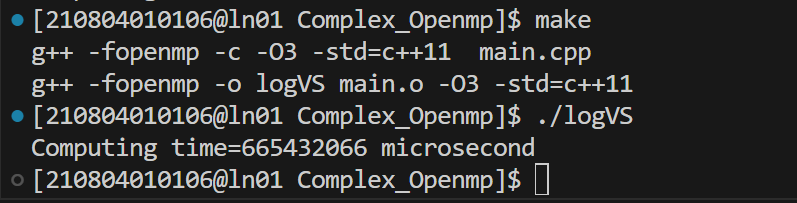
1. **实验准备**

MPI是一种用于并行计算的通信库，通常用于在多个处理器或节点上执行并行计算任务。在进行MPI优化之前，首先测量MPI程序的性能基线。记录串行执行的程序运行时间和性能数据，以便后续的优化可以与之对比。同时要分析MPI程序以确定哪些部分可以并行化，了解哪些任务可以在不同的进程之间并行执行，以充分利用多核或多节点系统的潜力。MPI提供了一组函数，常用的有六个，分别为：MPI\_Init、MPI\_Finalize、MPI\_Comm\_rank、MPI\_Comm\_size、MPI\_Send、MPI\_Recv，这些函数可以实现并行计算中的进程间通信和同步。需要对这些函数有认真详细的研究，通过组合使得MPI对程序的运行速度有较好的优化。

**二、实验步骤**

(1).经过精度降低、复数拆分和-O3优化后执行时间为：665432066 us

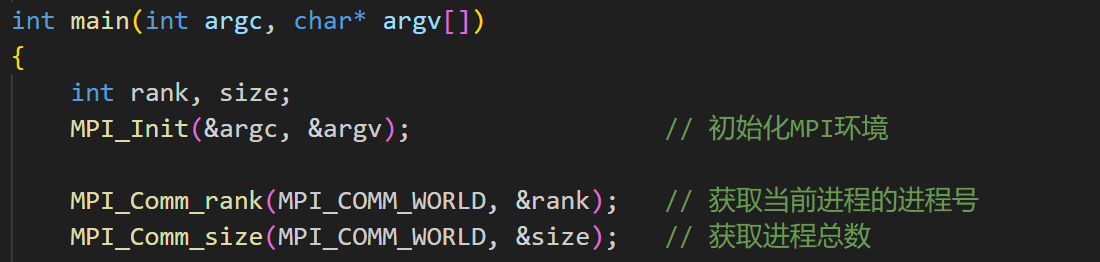


(2). MPI优化

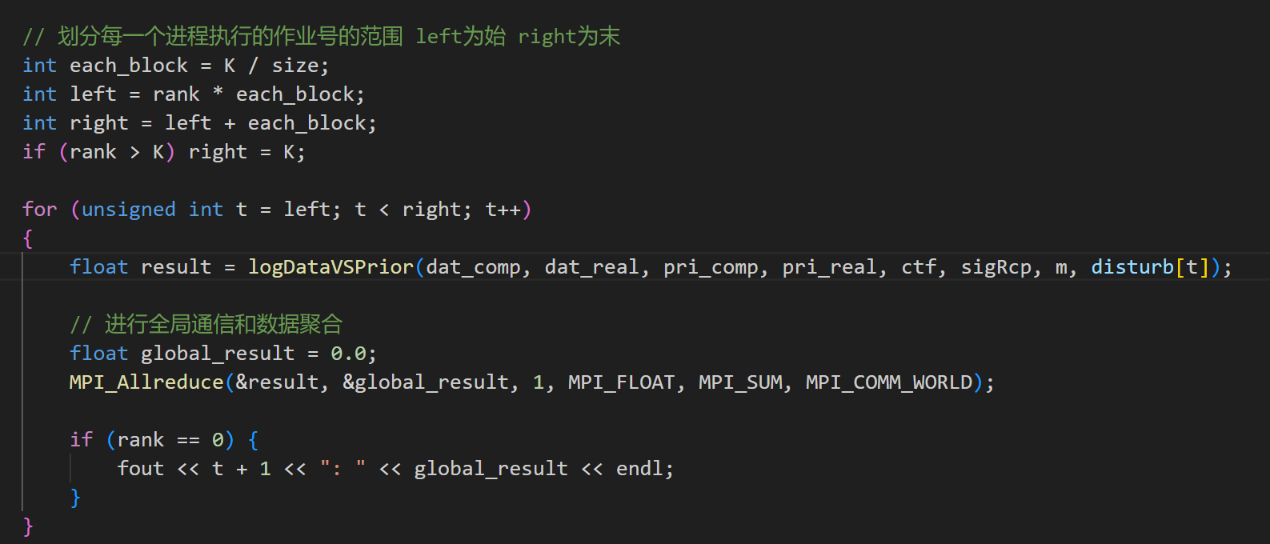
首先需要包含头文件

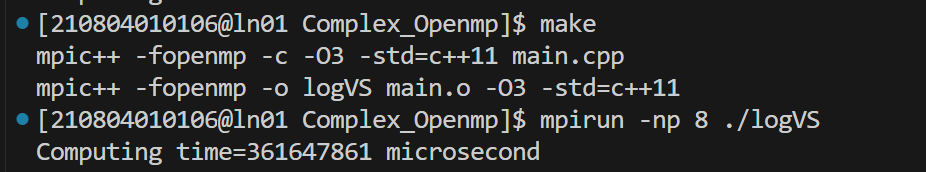
#include <mpi.h>

在程序一开始时就MPI\_Init函数初始化，进入MPI并行环境，并用rank标记当前的进程号



通过每个进程的rank值，分配每个进程所要处理的作业数的始末，然后在每个进程中，通过函数，完成线程间的数据通信和并行，实现优化。





加速比为：665432066 / 361647861= **1.84**

**总的加速比为：**

7884142619 / 361647861 = **21.80**

**三、实验总结**

本实验旨在优化MPI程序的性能，首先通过测量串行执行的程序性能基线，记录其运行时间和性能数据，为后续优化提供对比基准。在实验中，我们深入研究了MPI库的核心函数，包括MPI\_Init、MPI\_Finalize、MPI\_Comm\_rank、MPI\_Comm\_size、MPI\_Send、MPI\_Recv，以理解它们在并行计算中的关键作用。通过组合和合理运用这些函数，实现了MPI并行环境的初始化，并为每个进程分配作业任务，以便充分利用多核或多节点系统的潜力。

经过实验，通过精度降低、复数拆分和-O3优化后，我们成功将程序执行时间从7884142619.0微秒降低到361647861微秒，实现了显著的性能提升。这一过程中，MPI的并行计算机制发挥了关键作用，使不同进程之间能够高效地进行通信和同步，从而加速了程序的执行。

最终，实验结果表明，MPI优化后的程序加速比为21.80，进一步验证了MPI在并行计算中的重要性和效果。通过这次实验，我们深入了解了MPI库的功能和应用，为今后的并行计算工作提供了有力支持，使程序在多处理器或节点上能够更高效地运行，提高了计算效率。