并行结算实验报告

赵东杰　P18206023　手机号：13121595665　邮箱：zdj8023first@163.com

开发前准备

开发配置：4 核 8 线程
开发环境：Ubuntu 18.04 LTS
开发平台：Clion
开发语言：c++

1 稀疏矩阵求解

通过配置项目的 CMakeLists.txt 来配置 openmp 环境

```
# 配置 openmp
FIND_PACKAGE( OpenMP REQUIRED)
if(OPENMP_FOUND)
    message("OPENMP FOUND")
    set(CMAKE_C_FLAGS "${CMAKE_C_FLAGS} ${OpenMP_C_FLAGS}")
    set(CMAKE_CXX_FLAGS "${CMAKE_CXX_FLAGS}
${OpenMP_CXX_FLAGS}")
endif()
```

不同情况下程序的运行时间如下表所示

	串行	1 核	2 核	4 核	8 核
时间：s	41.25	37.69	23.82	20.41	11.20

运行时间截图

```
program is running as serial
program time:41.2569s
```

串行

```
the prograom is running on 1 threads
program time 37.695466 s
```

并行化 1 核

```
the prograom is running on 2 threads
program time 23.823626 s
```

并行化 2 核

```
the prograom is running on 4 threads
program time 20.417332 s
```

并行化 4 核

```
the prograom is running on 8 threads
program time 11.203307 s
```

并行化 8 核

实验结果分析
从上述图标中可以看出，随着线程数目的增多，程序运行时间逐渐降低。
在解决矩阵运算这种大量 for 循环的问题，通过将 for 循环进行并行化处理，能够大大的降低程序的运行时间。

3 Random Reduction

环境配置：

下载 MPI 包 https://www.open-mpi.org/software/ompi/v3.0/

之后解压配置环境，这里不再赘述。
主要命令： mpic++, mpirun

通过 python 生成 23*23 的矩阵（我的学号以及詹姆斯粉丝）
通过 Reduction.cpp 生成 Alphas.mat、Betas.mat、Gammas.mat 文件
修改 Reduction.cpp 文件为 Reduction_mpi.cpp 文件并生成 AlphasMPI.mat、BetasMPI.mat、GammasMPI.mat 文件。

最后运行验证文件得到结果。验证结果截图如下：

```
zdj@zdj-pc:~/Documents/pc_homework/random_reduction$ ls
Alphas.mat     BetasMPI.mat   matrix            Reduction.cpp      Validation_mpi
AlphasMPI.mat  Gammas.mat     Random_matrix.py  Reduction_mpi      Validation_mpi.cpp
Betas.mat      GammasMPI.mat  Reduction         Reduction_mpi.cpp
zdj@zdj-pc:~/Documents/pc_homework/random_reduction$ ./Validation_mpi
VALID!
```

4 MPIio

实验要求较多这里不再赘述。
实验结果如图所示

```
zdj@zdj-pc:~/Documents/pc_homework/mpiio$ mpirun io_work 100
Number 1:5 9 10 1 7 10 4 2 7 3 2 6 9 2 5 3 3 6 4 10 6 5 4 6 7
Number 0:5 9 10 1 7 10 4 2 7 3 2 6 9 2 5 3 3 6 4 10 6 5 4 6 7
Number 2:5 9 10 1 7 10 4 2 7 3 2 6 9 2 5 3 3 6 4 10 6 5 4 6 7
Number 3:5 9 10 1 7 10 4 2 7 3 2 6 9 2 5 3 3 6 4 10 6 5 4 6 7
a:5 9 10 1 7 10 4 2 7 3 2 6 9 2 5 3 3 6 4 10 6 5 4 6 7 5 9 10 1 7 10 4 2 7 3 2 6 9 2 5 3 3 6
 4 10 6 5 4 6 7 5 9 10 1 7 10 4 2 7 3 2 6 9 2 5 3 3 6 4 10 6 5 4 6 7 5 9 10 1 7 10 4 2 7 3 2
 6 9 2 5 3 3 6 4 10 6 5 4 6 7
b:5 9 10 1 7 10 4 2 7 3 2 6 9 2 5 3 3 6 4 10 6 5 4 6 7 5 9 10 1 7 10 4 2 7 3 2 6 9 2 5 3 3 6
 4 10 6 5 4 6 7 5 9 10 1 7 10 4 2 7 3 2 6 9 2 5 3 3 6 4 10 6 5 4 6 7 5 9 10 1 7 10 4 2 7 3 2
 6 9 2 5 3 3 6 4 10 6 5 4 6 7
verify success
```