

并行程序设计 第八次作业

2100011047 工学院 陈国赐

文件结构

`source` 目录下主要文件(目录)及其功能如下表格所示:

文件(目录)名	文件(目录)功能
output/	存放输出log的文件夹
input.cpp	输入类实现
input.h	输入类声明及接口
mat_demo.cpp	矩阵类实现
mat_demo.h	矩阵类声明及接口
mytimer.cpp	计时器类实现
mytimer.h	计时器类声明及接口
main.cpp	主程序源码
Makefile	makefile

编译说明

`Makefile` 中提供了5个目标 `clean`, `mpi`, `serial`, `runmpi`, `runserial`.

运行 `make mpi` 会编译MPI版本 `hw8_mpi`.

运行 `make serial` 会编译串行版本 `hw8_serial`.

运行 `make runmpi` 会在编译MPI版本的基础上运行 `hw8_mpi`.

运行 `make runserial` 会在编译串行版本的基础上运行 `hw8_serial`.

运行 `make clean` 会清除生成的可执行文件.

其中, `runserial` 可指定 `INPUT_FILE_PATH` 作为输入文件路径.

`runmpi` 可指定 `INPUT_FILE_PATH` 作为输入文件路径.

`MPI_PROCESS_NUM` 为运行的进程个数,

`MPIRUN_FLAGS` 为 `mpirun` 运行时的附加参数, 如下:

```
make runserial INPUT_FILE_PATH=nani.in
make runmpi MPIRUN_FLAGS="--oversubscribe MPI_PROCESS_NUM=16"
```

(`--oversubscribe` 开关可以让运行进程个数大于机器核个数, 在自己电脑上测试需要此开关.)

运行说明

在命令行中执行

```
./main [FILE]
```

即可以 `[FILE]` 为输入文件运行程序.

如果在串行模式下执行, 会在 `result_print` 为1的情况下直接在屏幕上输出矩阵.

如果是在并行模式下执行, 会在 `print_mpi_log` 为1的情况下在对应的 `.log` 文件中输出分块的矩阵.

代码结构&函数功能

其他的文件和前几次作业差不多, 我们简析一下 `main.cpp`.

`main.cpp` 中主要有三个函数: `main()`, `mpi_dist()`, `mpi_solver()`;

其中 `mpi_dist()` 将某个一个矩阵的分块发送给另一个指定矩阵,

`mpi_solver()` 完成了并行模式下的一个进程的任务, 是一个较大的函数, 故从 `main()` 中独立出来.

`main` 函数由以下几部分组成:

1. 检查命令行参数, 创建文件夹.
2. (如果是并行模式) 获得 `rank` 和 `size`.
3. 主进程从文件中获得参数和矩阵.
4. 分类讨论, 执行 `mpi_solver()` 或是直接输出矩阵.
5. 输出计时器, 结束程序.

`mpi_solver()` 由以下组成:

1. 如果是主进程则发送矩阵和参数.
2. 如果是主进程则复制矩阵到某个小的切块矩阵, 否则接受矩阵和参数.
3. 如果指定 `print_mpi_log` 则做输出.