Project1

> Author : 胡子昂
>
> 本次作业完成四种体系结构（openmp，mpi，pthread，ispc）对pi，矩阵乘法，矩阵转置的计算。

项目目录

- 项目目录如下：

```
$ tree -L 3 .
.
├── bin
|   ├── calpi
|   |   ├── calpi_ispc
|   |   ├── calpi_openmp
|   |   └── calpi_pthread
|   ├── multiplymatrix
|   |   ├── multiplymatrix_ispc
|   |   ├── multiplymatrix_openmp
|   |   └── multiplymatrix_pthread
|   └── transpositionmatrix
|       ├── transpositionmatrix_ispc
|       ├── transpositionmatrix_openmp
|       └── transpositionmatrix_pthread
├── inc
|   ├── calpi.h
|   ├── ldouble.h
|   ├── matrix.h
|   └── timer.h
└── src
    ├── ISPC
    |   ├── calpi
    |   ├── multiplymatrix
    |   └── transpositionmatrix
    ├── MPI
    |   ├── calpi.cpp
    |   ├── multiplymatrix.cpp
    |   └── transpositionmatrix.cpp
    ├── OpenMP
    |   ├── calpi.cpp
    |   ├── multiplymatrix.cpp
    |   └── transpositionmatrix.cpp
    └── pthread
        ├── calpi.cpp
```

```
        ├── multiplymatrix.cpp
        └── transpositionmatrix.cpp

13 directories, 22 files
```

（其中，bin是mac下的可执行程序，inc是头文件存放目录，src是源文件，按照四种体系结构放置）

项目编译运行环境

1. 所有程序都使用4个核处理。
2. 程序编译指令如下：（都加上O3 flag）

```
# pthread
$ g++ -std=c++11 -O3 -lpthread -o XXX XXX.cpp
# openmp
$ g++-8 -std=c++11 -O3 -fopenmp -o XXX XXX.cpp
# mpi
$ mpic++ -std=c++11 -O3 -o XXX XXX.cpp
$ mpirun -n 4 XXX
# ispc (见makefile)
$ make
```

3. 程序数据的大小将在运行程序后输入。

机器的详细配置

- 查看电脑配置如下：

```
硬件概览：

  型号名称：  MacBook Pro
  型号标识符：      MacBookPro14,1
  处理器名称：      Intel Core i5
  处理器速度：      2.3 GHz
  处理器数目：      1
  核总数：    2
  L2 缓存（每个核）：     256 KB
  L3 缓存：       4 MB
  内存：     8 GB
  Boot ROM 版本：    MBP141.0175.B00
  SMC 版本（系统）：     2.43f6
  序列号（系统）：    FVFW41A3HV2D
  硬件 UUID：    2A67EB9E-7112-52A5-8062-1DBA84638C27
```

程序运行结果

- 计算PI：

并行方法\数据大小	10000	100000	1000000	10000000
pthread	249us	401us	1237us	11128us
openmp	377us	500us	1314us	9378us
MPI	108us	495us	1208us	9475us
ISPC	15us	148us	1070us	9401us

- 矩阵乘法：

并行方法\数据大小	10*10	100*100	1000*1000	2000*2000
pthread	233us	556us	1720659us	16535552us
openmp	369us	825us	1512316us	16560456us
MPI	130us	412us	1663575us	18352551us
ISPC	4us	394us	1885006us	15101259us

- 矩阵转置：

并行方法\数据大小	10*10	100*100	1000*1000	10000*10000
pthread	228us	288us	6990us	470883us
openmp	418us	453us	6747us	471360us
MPI	102us	289us	8702us	Reduce时溢出
ISPC	1us	75us	7127us	799286us

实验结果

1. 在低规模时，四种并行模型的运行速度比较分别是ISPC > MPI > pthread > openmp
2. 在高规模时，四种并行模型的运行速度差距不太大，反而ISPC和MPI不够稳定（可能是代码编写问题）
3. 第一次运行时，除了ISPC之外，其他三种情况都出现了运行时间特别大的情况，猜测可能是装页。
4. MPI由于没有共享内存，因此在代码编写方面比较困难，需要注意的事情很多