

并行程序设计与算法实验

5-基于OpenMP的并行矩阵乘法

提交格式说明

按照实验报告模板填写报告，需要提供源代码及代码描述至<https://easyhpc.net/course/222>。实验报告模板使用PDF格式，命名方式为“并行程序设计_学号_姓名”。如有疑问，请在企业微信课程群提问。

以下题目二选一。

1. OpenMP通用矩阵乘法

使用OpenMP实现并行通用矩阵乘法，并通过实验分析不同进程数量、矩阵规模、调度机制时该实现的性能。

输入： m, n, k 三个整数，每个整数的取值范围均为 $[128, 2048]$

问题描述：随机生成 $m \times n$ 的矩阵 A 及 $n \times k$ 的矩阵 B ，并对这两个矩阵进行矩阵乘法运算，得到矩阵 C 。

输出： A, B, C 三个矩阵，及矩阵计算所消耗的时间 t 。

要求：使用OpenMP多线程实现并行矩阵乘法，设置不同线程数量（1-16）、矩阵规模（128-2048）、调度模式（默认、静态、动态调度），通过实验分析程序的并行性能。

2. 构造基于Pthreads的并行for循环分解、分配、执行机制

模仿OpenMP的omp_parallel_for构造基于Pthreads的并行for循环分解、分配及执行机制。

问题描述：实现parallel_for函数，该函数创建多个Pthreads线程，并行执行parallel_for函数的参数所指定的内容。

函数参数： parallel_for函数的参数应当指明被并行循环的索引信息，循环中所需要执行的内容，并行构造等。以下为parallel_for函数的基础定义，实验实现应包括但不限于以下内容：

```
parallel_for(int start, int end, int inc,
             void *(*functor)( int,void*), void *arg, int
num_threads)
```

- start, end, inc分别为循环的开始、结束及索引自增量；
- functor为函数指针，定义了每次循环所执行的内容；
- arg为functor的参数指针，给出了functor执行所需的数据；
- num_threads为期望产生的线程数量。
- 选做：除上述内容外，还可以考虑调度方式等额外参数。

示例： 给定functor及参数如下：

```
struct functor_args {
    float *A, *B, *C;
};

void *functor(int idx, void* args){
    functor_args *args_data = (functor_args*) args;
    args_data->C[idx] = args_data->A[idx] + args_data->B[idx];
}
```

调用方式如下：

```
functor_args args = {A, B, C};
parallel_for(0, 10, 1, functor, (void*)&args, 2)
```

该调用方式应当能产生两个线程，并行执行functor完成数组求和 ($C_i = A_i + B_i$)。当不考虑调度方式时，可由前一个线程执行任务 {0,1,2,3,4}，后一个线程执行任务 {5,6,7,8,9}。也可以实现对调度方式的定义。

要求： 完成parallel_for函数实现，并以矩阵乘法（或其他）为例，测试其实现的正确性及效率。