**深 圳 大 学**

**实 验 报 告**

**课程名称****：并行计算**

**实验名称：简单矩阵乘法的OpenMP并行程序**

**姓 名：刘俊楠**

**学 号：2017303010**

**班 级：计科01**

**实验日期：第4周实验课**

## 一. 实验目的

1. 学会编写简单的OpenMP程序；

2. 掌握for编译制导语句；

3. 对并行程序进行简单的性能分析；

## 二. 实验环境

1. 硬件环境：64核CPU、128GB内存的SMP并行计算平台；

2 软件环境：Microsoft Visual Studio 2013；

3. 登录方式：通过远程桌面（mstsc）连接172.31.226.180或hpc.szu.edu.cn:805，用户名和初始密码都是自己的学号。

## 三. 实验内容

1. 用OpenMP编写两个*n*阶方阵*a*和*b*的并行相乘程序，结果存放在方阵*c*中，其中矩阵乘法部分用for编译制导语句实现并行化操作。为了验证计算结果的正确性，将矩阵乘法的串行计算结果存放在方阵*d*中，并比较是否与*c*相等。在下面写出完整的程序代码，并添加必要的注释。

#include <omp.h>

#include <stdlib.h>

#include <time.h>

#include <iostream>

**using** **namespace** std**;**

const int n **=** 1000**;**

double a**[**n**][**n**],** b**[**n**][**n**],** c**[**n**][**n**],** d**[**n**][**n**];**

double chuanxing**[**5**],** bingxing**[**5**];**//记录串行执行时间与并行执行时间

void main**()**

**{**

**for** **(**int i **=** 0**;** i **<** n**;** i**++)**

**for** **(**int j **=** 0**;** j **<** n**;** j**++)**

**{**

a**[**i**][**j**]** **=** rand**()** **\*** 1.0 **/** RAND\_MAX**;**

b**[**i**][**j**]** **=** rand**()** **\*** 1.0 **/** RAND\_MAX**;**

**}**

int hthreads**=**1**;**//线程数变量

**for** **(**int pi **=** 0**;**pi **<** 7**;**pi**++)** **{**//每次线程数\*2，循环七次

printf**(**"线程数为%d时：\n"**,** hthreads**);**

**for** **(**int h **=** 0**;**h **<** 5**;**h**++)** **{**//每次循环五次

clock\_t start1 **=** clock**();**//记录串行执行时间起始

**for** **(**int i **=** 0**;**i **<** n**;**i**++)** **{**//串行矩阵运算

**for** **(**int j **=** 0**;**j **<** n**;**j**++)** **{**

**for** **(**int k **=** 0**;**k **<** n**;**k**++)**

d**[**i**][**j**]** **+=** a**[**i**][**k**]** **\*** b**[**k**][**j**];**

**}**

**}**

clock\_t end1 **=** clock**();**

omp\_set\_num\_threads**(**hthreads**);**//设置并行线程数

clock\_t start **=** clock**();**//记录并行执行起始时间

int i**,** j**,** k**;**

//以下是并行域，共享变量为a,b,c数组，私有变量是i,j,k

#pragma omp parallel shared(a,b,c) private(i,j,k)

**{**

#pragma omp for //对以下for循环进行并行线程分配

**for** **(**i **=** 0**;**i **<** n**;**i**++)** **{**

**for** **(**j **=** 0**;**j **<** n**;**j**++)** **{**

**for** **(**k **=** 0**;**k **<** n**;**k**++)** **{**

c**[**i**][**j**]** **+=** a**[**i**][**k**]** **\*** b**[**k**][**j**];**

**}**

**}**

**}**

**}**

clock\_t end **=** clock**();**

//打印结果，并计算所记录的时间、加速比

printf**(**"---------------------第%d次：----------------------\n"**,** h**+**1**);**

bingxing**[**h**]** **=** **(**end **-** start**)** **/** 1000.0**;**

chuanxing**[**h**]** **=** **(**end1 **-** start1**)** **/** 1000.0**;**

printf**(**"串行消耗时间为：%f秒\n"**,** chuanxing**[**h**]);**

printf**(**"并行消耗时间为：%f秒\n"**,** bingxing**[**h**]);**

printf**(**"加速比为%f\n"**,** chuanxing**[**h**]** **/** bingxing**[**h**]);**

//遍历查看是否有不一致计算结果

**for** **(**int i **=** 0**;**i **<** n**;**i**++)** **{**

**for** **(**int j **=** 0**;**j **<** n**;**j**++)** **{**

**if** **(**c**[**i**][**j**]** **!=** d**[**i**][**j**])**

**{**

printf**(**"c[%d][%d]=%f,d[%d][%d]=%f "**,** i**,** j**,** c**[**i**][**j**],** i**,** j**,** d**[**i**][**j**]);**

printf**(**"不相同\n"**);**

exit**(**0**);**

**}**

**}**

**}**

**}**

//线程数\*2

hthreads **\*=** 2**;**

//遍历并计算串行、并行执行时间以及加速比的平均值

double average**[**2**]** **=** **{** 0**,**0 **};**

**for** **(**int i **=** 0**;**i **<** 5**;**i**++)** **{**

average**[**0**]** **+=** chuanxing**[**i**];**

average**[**1**]** **+=** bingxing**[**i**];**

**}**

average**[**0**]** **/=** 5.0**;**

average**[**1**]** **/=** 5.0**;**

printf**(**"-------------------------------------------------\n"**);**

printf**(**"串行平均时间为：%f秒\n"**,** average**[**0**]);**

printf**(**"并行平均时间为：%f秒\n"**,** average**[**1**]);**

printf**(**"平均加速比为%f\n"**,** average**[**0**]** **/** average**[**1**]);**

printf**(**"-------------------------------------------------\n"**);**

**}**

**}**

2. 测试并行程序在不同线程数下的执行时间和加速比（串行执行时间/并行执行时间），并分析实验结果。其中，*n*固定为1000，线程数分别取1、2、4、8、16、32、64时，为减少误差，每项实验进行5次，取平均值作为实验结果。

表1 并行程序在不同线程数下的执行时间（秒）和加速比

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 线程数  执行时间 | 1 | 2 | 4 | 8 | 16 | 32 | 64 |
| 第1次 | 4.438 | 2.609 | 2.472 | 2.333 | 2.281 | 2.269 | 2.331 |
| 第2次 | 4.420 | 2.718 | 2.408 | 2.326 | 2.306 | 2.288 | 2.270 |
| 第3次 | 4.393 | 3.095 | 2.410 | 2.326 | 2.293 | 2.271 | 2.259 |
| 第4次 | 4.448 | 2.703 | 2.355 | 2.286 | 2.293 | 2.305 | 2.260 |
| 第5次 | 4.431 | 2.727 | 2.447 | 2.341 | 2.303 | 2.262 | 2.264 |
| 平均值 | 4.426 | 2.770 | 2.418 | 2.322 | 2.295 | 2.279 | 2.276 |
| 加速比 | 0.852 | 1.332 | 1.523 | 1.580 | 1.650 | 1.609 | 1.616 |

