**深 圳 大 学**

**实 验 报 告**

**课程名称****：并行计算**

**实验名称：归并的OpenMP并行程序**

**姓 名：刘俊楠**

**学 号：2017303010**

**班 级：计科一班**

**实验日期：第10周实验课**

## 一. 实验目的

1. 实现简单排序的OpenMP并行程序；

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. 软件环境：Microsoft Visual Studio 2013。

## 三. 实验内容

1. 实验要求：归并有序数组*a*和*b*：

* 数组大小*n*和线程数*p*都是可输入的参数。
* 数组*a*和*b*中的每个数都初始化为一个0到1之间的随机double型值（用rand()/double(RAND\_MAX)实现），然后调用sort函数分别排序数组*a*和*b*。
* 先将数组*a*等分为*p*段，然后选前*p*-1段的最后一个元素作为划分元，将数组b也分为*p*段，最后线程*i*归并数组*a*和*b*的第*i*段。
* 添加检测归并结果是否正确的代码。
* 添加计算归并时间的代码，注意不包含数组的初始化时间。

2. 程序代码和说明：

#include<iostream>

#include<algorithm>

#include <omp.h>

#include<vector>

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

//数组大小n和线程数p

int n**=** 100000000**,**p**;**

int num**;**

int**\*** u**,** **\*** v**;**

//将两个数组归并成一个数组的merge函数

void merge**(**double**\*** a\_start**,** double**\*** a\_end**,** double**\*** b\_start**,** double**\*** b\_end**,** double**\*** c\_start**)** **{**

double**\*** f **=** a\_start**;**

double**\*** g **=** b\_start**;**

**while** **(**f **!=** a\_end **&&** g **!=** b\_end**)** **{**

**if** **(\*(**f**)** **>** **\***g**)** **{**

**\***c\_start **=** **\***g**;**

g **=** g **+** 1**;**

**}**

**else** **{**

**\***c\_start **=** **\***f**;**

f **=** f **+** 1**;**

**}**

c\_start **=** c\_start **+** 1**;**

**}**

**while** **(**f **!=** a\_end**)** **{**

**\***c\_start **=** **\***f**;**

f **=** f **+** 1**;**

c\_start **=** c\_start **+** 1**;**

**}**

**while** **(**g **!=** b\_end**)** **{**

**\***c\_start **=** **\***g**;**

g **=** g **+** 1**;**

c\_start **=** c\_start **+** 1**;**

**}**

**}**

//判断是否有序的函数

bool judge**(**double**\*** a**,** double**\*** d**)** **{**

**for** **(**int i **=** 0**;** i **<** 2**\***n**;** i**++)** **{**

**if** **(**a**[**i**]** **!=** d**[**i**])** **{**

cout **<<** "error!" **<<** endl**;**

cout **<<** "i==" **<<** i **<<** endl**;**

cout **<<** "a[i]=" **<<** a**[**i**]** **<<** " " **<<** "d[i]==" **<<** d**[**i**]** **<<** endl**;**

**return** **false;**

**}**

**}**

**return** **true;**

**}**

//求秩的函数

int Rank**(**double **\***b1**,**double **\***b2**,**int uu**,**double **\***a**)** **{**

double**\*** q **=** a**+**uu**-**1**;**

int t **=** 0**;**

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

**if** **(\*(**b1 **+** i**)** **<** **\***q**)** **{**

t**++;**

**}**

**else**

**break;**

**}**

**return** t**;**

**}**

//打印函数

void printc**(**double **\***p**)** **{**

**for** **(**int i **=** 0**;** i **<** 2 **\*** n**;** i**++)** **{**

cout **<<** p**[**i**]** **<<** " "**;**

**if** **(**i **%** 10 **==** 0 **&&** i **!=** 0**)** **{**

cout **<<** endl**;**

**}**

**}**

**}**

//打印函数

void printab**(**double**\*** e**)** **{**

cout **<<**endl**<<** "---------------------------------" **<<** endl**;**

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

cout **<<** e**[**i**]** **<<** " "**;**

**if** **(**i **%** 10 **==** 0 **&&** i **!=** 0**)** cout **<<** endl**;**

**}**

cout **<<** endl **<<** "---------------------------------" **<<** endl**;**

**}**

int main**()** **{**

//线程初始时为1

p **=** 1**;**

cout **<<** "请输入数组大小:" **<<** endl**;**

cin **>>** n**;**

cout **<<** "请输入线程数目：" **<<** endl**;**

cin **>>** p**;**

//循环7次，每次线程\*2

//for (int l = 0; l < 7; l++) {

cout **<<** "--------------------------------------串行计算---------------------------------" **<<** endl**;**

double**\*** a**,** **\*** b**,** **\*** c**;**

//添加计算归并时间的代码。

clock\_t start**,** end**;**

double single\_time**;**

double multiple\_time**[**5**]** **=** **{** 0 **};**

c **=** **new** double**[**2 **\*** n **+** 1**];**

double**\*** c\_standard **=** **new** double**[**2 **\*** n **+** 1**];**

a **=** **new** double**[**n **+** 1**];**

b **=** **new** double**[**n **+** 1**];**

u **=** **new** int**[**p **+** 1**];**

v **=** **new** int**[**p **+** 1**];**

//数组a和b中的每个数都初始化为一个0到1之间的随机double型值

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

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

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

**}**

// 调用sort函数分别排序数组a和b。

sort**(**a**,** a **+** n**);**

sort**(**b**,** b **+** n**);**

//串行计算计时开始

start **=** clock**();**

num **=** n **/** p**;**

u**[**p**]** **=** n**;**

v**[**p**]** **=** n**;**

v**[**0**]** **=** 0**;**

u**[**0**]** **=** 0**;**

//先将数组a等分为p段，然后选前p - 1段的最后一个元素作为划分元，将数组b也分为p段

**for** **(**int i **=** 1**;** i **<** p**;** i**++)** **{**

u**[**i**]** **=** num **\*** i**;**

v**[**i**]** **=** Rank**(**b**,** b **+** n**,** u**[**i**],** a**);**

**}**

// 最后线程i归并数组a和b的第i段。

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

merge**(**a **+** u**[**i**],** a **+** u**[**i **+** 1**],** b **+** v**[**i**],** b **+** v**[**i **+** 1**],** c **+** u**[**i**]** **+** v**[**i**]);**

**}**

end **=** clock**();**

//串行计算计时结束

cout **<<** "+++++++++++++++++++++++++++++++线程数为" **<<** p **<<** "时：++++++++++++++++++++++++" **<<** endl**;**

//串行计算所用时间

single\_time **=** **(**end **-** start**)** **/** 1000.0**;**

cout **<<** "串行运行时间为：" **<<** single\_time **<<** "秒" **<<** endl**;**

int i**;**

//循环五次，测量平均值

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

//重新初始化数组，防止溢出

**delete[]**u**;**

**delete[]**v**;**

double**\*** c1 **=** **new** double**[**2 **\*** n **+** 1**];**

u **=** **new** int**[**p **+** 1**];**

v **=** **new** int**[**p **+** 1**];**

clock\_t start1**,** end1**;**

cout **<<** "---------------------第" **<<** t **+** 1 **<<** "次并行计算：----------------------" **<<** endl**;**

u**[**p**]** **=** n**;**

v**[**p**]** **=** n**;**

v**[**0**]** **=** 0**;**

u**[**0**]** **=** 0**;**

//并行计算开始计时

start1 **=** clock**();**

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

//设置并行域

#pragma omp parallel shared(a,b,c1,u,v,num,n,p) private(i)

**{**

//并行求秩v[i]，u[i]为分块

#pragma omp for schedule(static,1)

**for** **(**i **=** 1**;** i **<** p**;** i**++)** **{**

u**[**i**]** **=** num **\*** i**;**

v**[**i**]** **=** Rank**(**b**,** b **+** n**,** u**[**i**],** a**);**

**}**

#pragma omp barrier//等待所有线程执行完才执行下一步

//对所分得块进行并行归并

#pragma omp for schedule(static,1)

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

merge**(**a **+** u**[**i**],** a **+** u**[**i **+** 1**],** b **+** v**[**i**],** b **+** v**[**i **+** 1**],** c1 **+** u**[**i**]** **+** v**[**i**]);**

**}**

**}**

end1 **=** clock**();**

//结束计时并统计时间

multiple\_time**[**t**]** **=** **(**end1 **-** start1**)** **/** 1000.0**;**

cout **<<** "并行时间为" **<<** multiple\_time**[**t**]** **<<** "秒" **<<** endl**;**

//添加检测归并结果是否正确的代码。

judge**(**c**,** c1**);**

**delete[]**c1**;**

**}**

//计算并行平均时间

double average**=**0.0**;**

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

average **+=** multiple\_time**[**i**];**

**}**

cout **<<** "-----------------------------------------------------------------------" **<<** endl**;**

average **/=** 5**;**

cout **<<** "并行平均时间为：" **<<** average **<<** "秒" **<<** endl**;**

cout **<<** "平均加速比为" **<<** single\_time **/** average **<<** endl**;**

**delete[]**a**;**

**delete[]**b**;**

**delete[]**c**;**

//p \*= 2;

//}

**}**

3. 实验结果和分析：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，数组大小*n*固定为100000000，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 | 2.768 | 0.954 | 0.999 | 1.1 | 1.72 | 2.522 | 4.233 |
| 第2次 | 4.426 | 0.933 | 0.986 | 1.055 | 1.48 | 2.245 | 4.058 |
| 第3次 | 2.587 | 0.946 | 0.945 | 1.063 | 1.508 | 2.235 | 4.008 |
| 第4次 | 1.403 | 0.984 | 0.924 | 1.029 | 1.473 | 2.218 | 4.073 |
| 第5次 | 1.255 | 0.957 | 0.941 | 1.004 | 1.44 | 2.212 | 4.124 |
| 平均值 | 2.4878 | 0.9548 | 0.959 | 1.0502 | 1.5242 | 2.2864 | 4.0992 |
| 加速比 | 1.43621 | 1.07667 | 1.40042 | 1.67873 | 1.82194 | 2.10987 | 2.21897 |





