**Лабораторная работа №5**

**Распределение вычислительной нагрузки**

**Цель:** изучить основные особенности распределения вычислительной нагрузки в OpenMP на примере использования в рамках языка С++.

**Лабораторные задания**

**Задание.** Модифицировать программы, составленные в Л.Р. №3, распределив вычислительную нагрузку по секциям следующим образом:

1. **разбиваем вычисления на 2 секции – в каждую секцию отправляем половину строк исходных матриц для вычисления половины строк результирующей матрицы;**
2. **разбиваем вычисления на 4 секции – в каждую секцию отправляем четверть строк исходных матриц для вычисления четверти строк результирующей матрицы.**

Измерять время работы программы для тех же значений параметров, что были использованы при выполнении Л.Р. №3. Дополнительно сравнить время работы с Л.Р. №3, запущенной на 2 и 4 потоках. **Результаты сравнить и занести в отчёт.**

**Таблица сравнения Л.Р. 3**

|  |  |  |
| --- | --- | --- |
| Размер массива n | Время вычисления без OpenMP, сек | Время вычисления с OpenMP, сек |
| 1 | 5.00004e-07 | 0.0119175 |
| 5 | 5.00004e-07 | 3.78999e-05 |
| 10 | 6.00005e-07 | 4.3e-05 |
| 100 | 2.10002e-06 | 4.08e-05 |
| 1000 | 1.52e-05 | 4.32001e-05 |
| 10000 | 7.71999e-05 | 7.61e-05 |
| 50000 | 0.000388 | 0.0002163 |
| 100000 | 0.0017256 | 0.0002687 |
| 1000000 | 0.0098707 | 0.0026271 |

**Таблица сравнения Л.Р. 5**

|  |  |  |
| --- | --- | --- |
| Размер массива n | Время вычисления с 2 секциями, сек | Время вычисления с 4 секциями, сек |
| 1 | 5.00004e-07 | 4.00003e-07 |
| 5 | 5.00004e-07 | 6.99889e-07 |
| 10 | 6.00005e-07 | 7.00005e-07 |
| 100 | 1.70001e-06 | 2.20002e-06 |
| 1000 | 1.41e-05 | 2.14e-05 |
| 10000 | 0.000209 | 0.0001023 |
| 50000 | 0.0003705 | 0.0004788 |
| 100000 | 0.0007364 | 0.0010016 |
| 1000000 | 0.00734 | 0.008221 |

/\*

На основе трёх равно размерных массивов A, B и C (длины N) функция возвращает произведение ненулевых значений,

полученных таким образом: если Ai четно: Bi/Ci , иначе Ai+Bi

\*/

#include <omp.h>

#include <iostream>

#include <ctime>

#include <iomanip>

// Возвращает псевдослучайное число из диапазона [min, max)

int Random(int min, int max) {

return min + rand() % (max - min);

}

//Создание рандомного массива

int\* generateArr(int n) {

//Автоматическая рандомизация

srand((unsigned int)time(NULL));

int\* arr = new int[n];

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

arr[i] = Random(1, 100);

}

return arr;

}

int sub\_omp\_2(int n, int\* A, int\* B, int\* C) {

int result = 1;

int i, j, k;

#pragma omp parallel sections for shared(A,B,C) private(i) reduction(\*:result)

{

#pragma omp section

for (i = 0; i < n / 2; i++) {

if (A[i] % 2 == 0) {

result \*= B[i] / C[i];

}

else {

result \*= (A[i] + B[i]);

}

}

#pragma omp section

for (i = n / 2; i < n; i++) {

if (A[i] % 2 == 0) {

result \*= B[i] / C[i];

}

else {

result \*= (A[i] + B[i]);

}

}

}

return result;

}

int sub\_omp\_4(int n, int\* A, int\* B, int\* C) {

int result = 1;

int i, j, k;

#pragma omp parallel sections for shared(A,B,C) private(i) reduction(\*:result)

{

#pragma omp section

for (i = 0; i < n / 4; i++) {

if (A[i] % 2 == 0) {

result \*= B[i] / C[i];

}

else {

result \*= (A[i] + B[i]);

}

}

#pragma omp section

for (i = n / 4; i < n / 2; i++) {

if (A[i] % 2 == 0) {

result \*= B[i] / C[i];

}

else {

result \*= (A[i] + B[i]);

}

}

#pragma omp section

for (i = n / 2; i < 3 \* n / 4; i++) {

if (A[i] % 2 == 0) {

result \*= B[i] / C[i];

}

else {

result \*= (A[i] + B[i]);

}

}

#pragma omp section

for (i = 3 \* n / 4; i < n; i++) {

if (A[i] % 2 == 0) {

result \*= B[i] / C[i];

}

else {

result \*= (A[i] + B[i]);

}

}

}

return result;

}

void test\_omp\_2(int n, int\* A, int\* B, int\* C) {

double t1, t2, result;

//Расчет произведения по данным условиям

t1 = omp\_get\_wtime();

result = sub\_omp\_2(n, A, B, C);

t2 = omp\_get\_wtime();

std::cout << "Тест c 2 секциями" << std::endl;

std::cout << "Произведение: " << result << std::endl;

std::cout << "Время исполнения основного вычислительного блока: " << t2 - t1 << std::endl;

}

void test\_omp\_4(int n, int\* A, int\* B, int\* C) {

double t1, t2, result;

//Расчет произведения по данным условиям

t1 = omp\_get\_wtime();

result = sub\_omp\_4(n, A, B, C);

t2 = omp\_get\_wtime();

std::cout << "Тест c 4 секциями" << std::endl;

std::cout << "Произведение: " << result << std::endl;

std::cout << "Время исполнения основного вычислительного блока: " << t2 - t1 << std::endl;

}

int main() {

system("chcp 1251");

int n;

while (true) {

//Ввод количества строк и столбцов массива

std::cout << "Введите размер массива n:" << std::endl;

std::cin >> n;

int\* A = generateArr(n);

int\* B = generateArr(n);

int\* C = generateArr(n);

test\_omp\_2(n, A, B, C);

test\_omp\_4(n, A, B, C);

}

return 0;

}