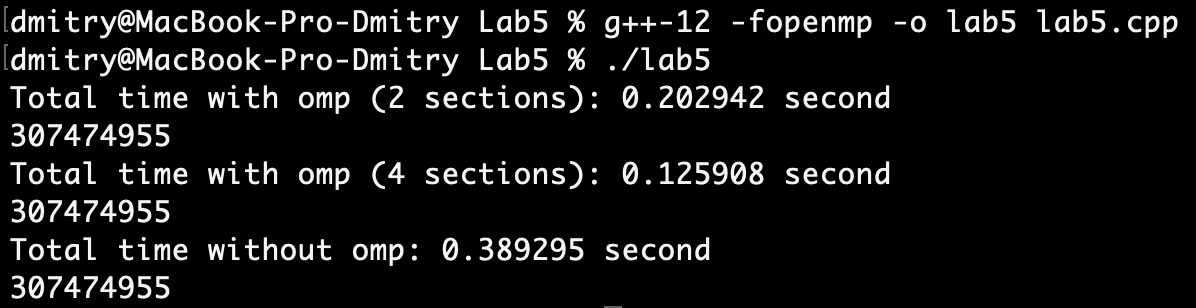
Технологии параллельного программирования

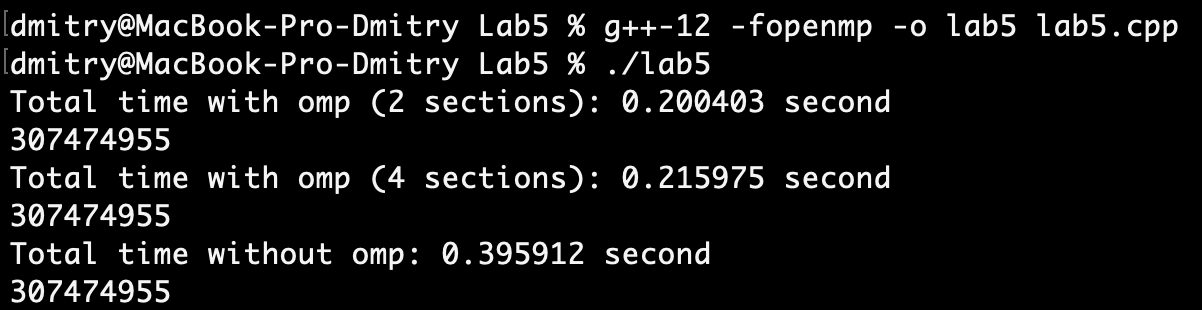
Отчёт к 5 лабораторной работе

Ивкучев Дмитрий ПИН-32

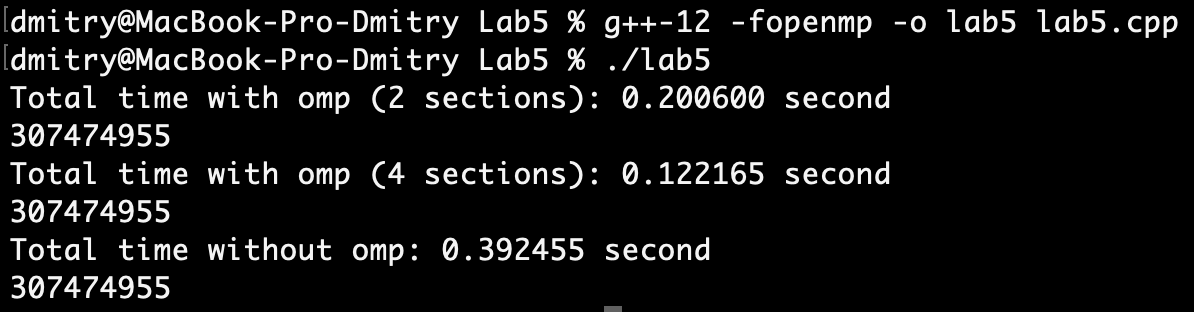
Вычисления со всеми потоками:



Вычисления с 2 потоками:



Вычисления с 4 потоками:



Код:

#include <iostream>

#include <omp.h>

using namespace std;

const int N = 50000000, LIMIT = 37000;

int main() {

int \*A = new int[N];

int \*B = new int[N];

int \*C = new int[N];

int i;

long long sum = 0;

for (i = 0; i < N; i++) {

A[i] = rand() % 10;

//cout << A[i] << "\t";

}

//cout << endl;

for (i = 0; i < N; i++) {

B[i] = rand() % 10;

//cout << B[i] << "\t";

}

//cout << endl;

double start, end;

start = omp\_get\_wtime();

#pragma omp parallel sections shared(A,B,C) private(i) reduction(+:sum) num\_threads(4)

{

#pragma omp section

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

if (A[i] > B[i]) C[i] = A[i];

else C[i] = B[i];

sum += C[i];

}

#pragma omp section

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

if (A[i] > B[i]) C[i] = A[i];

else C[i] = B[i];

sum += C[i];

}

}

end = omp\_get\_wtime();

printf("Total time with omp (2 sections): %f second\n", end-start);

printf("%lld\n",sum);

delete [] C;

C = new int[N];

sum = 0;

start = omp\_get\_wtime();

#pragma omp parallel sections shared(A,B,C) private(i) reduction(+:sum) num\_threads(4)

{

#pragma omp section

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

if (A[i] > B[i]) C[i] = A[i];

else C[i] = B[i];

sum += C[i];

}

#pragma omp section

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

if (A[i] > B[i]) C[i] = A[i];

else C[i] = B[i];

sum += C[i];

}

#pragma omp section

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

if (A[i] > B[i]) C[i] = A[i];

else C[i] = B[i];

sum += C[i];

}

#pragma omp section

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

if (A[i] > B[i]) C[i] = A[i];

else C[i] = B[i];

sum += C[i];

}

}

end = omp\_get\_wtime();

printf("Total time with omp (4 sections): %f second\n", end-start);

printf("%lld\n",sum);

delete [] C;

C = new int[N];

sum = 0;

start = omp\_get\_wtime();

for (i = 0; i < N; i++) {

if (A[i] > B[i]) C[i] = A[i];

else C[i] = B[i];

sum += C[i];

//printf("Прибавлена %d\n",C[i]);

}

end = omp\_get\_wtime();

printf("Total time without omp: %f second\n", end-start);

printf("%lld\n",sum);

return 0;

}