ПИН-33 Ткач Никита

ЛР6 Отчет

Репо <https://github.com/ChernyHleb/MyLabs/tree/main/Semester_5/OpenMPLabs/OpenMP_PIN-33_Tkach_Labs>

**Задание 1**

int64\_t Lab4::FuncWithLock(Matrix\* matrixA, Matrix\* matrixB)

{

int64\_t sum = 0;

int64\_t partitialSum;

int i, j, temp;

int dim = matrixA->N;

int\*\* A, \*\* B;

omp\_lock\_t lock;

omp\_init\_lock(&lock);

#pragma omp parallel shared(dim, matrixA, matrixB)

{

#pragma omp for private(i, j, temp, A, B, partitialSum)

for (i = 0; i < dim; i++)

{

partitialSum = 0;

A = matrixA->matrix;

B = matrixB->matrix;

for (j = 0; j < dim; j++)

{

int a = A[i][j];

int b = B[i][j];

temp = std::max(a + b, 4 \* a - b);

if (temp > 1) {

partitialSum += temp;

}

}

omp\_set\_lock(&lock);

sum += partitialSum;

omp\_unset\_lock(&lock);

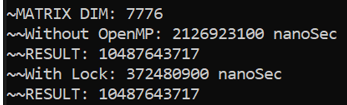
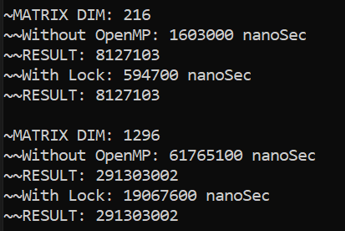
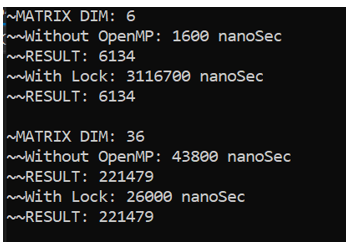
}

}

omp\_destroy\_lock(&lock);

return sum;

}

****

**Вывод: мы можем наблюдать выйгрыш по времени более чем в 3 раза при размерности матрицы 1296**

**Для матриц меньшей размерности выйгрыша по времени при использовании замка нет.**

**Задание 2**

int64\_t Lab4::FuncWithBarrier(Matrix\* matrixA, Matrix\* matrixB)

{

int64\_t sum = 0;

int64\_t partitialSum = 0;

int i, j, temp;

int dim = matrixA->N;

int\*\* A, \*\* B;

#pragma omp parallel shared(dim, matrixA, matrixB) private(partitialSum)

{

#pragma omp for private(i, j, temp, A, B)

for (i = 0; i < dim; i++)

{

partitialSum = 0;

A = matrixA->matrix;

B = matrixB->matrix;

for (j = 0; j < dim; j++)

{

int a = A[i][j];

int b = B[i][j];

temp = std::max(a + b, 4 \* a - b);

if (temp > 1) {

partitialSum += temp;

}

}

}

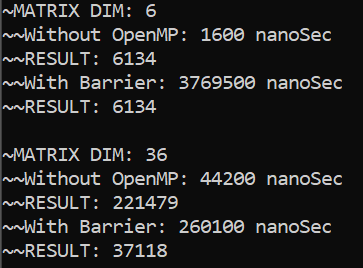
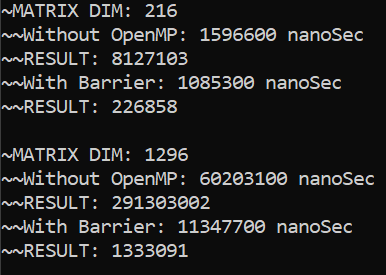
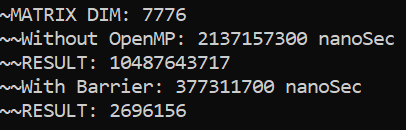
#pragma omp barrier

sum += partitialSum;

}

return sum;

}

**  **

**При использовании барьерной синхронизации мы можем видеть выйгрыш по времени при размерности матрицы от 216**