МГУ им. М. В. Ломоносова, факультет ВМК

Задание 1 Проблемы масштабируемости в OpenMP. Лишние барьеры для синхронизации работы нитей.

Арбузов Николай Романович
группа 423

Проблема

При реализации параллелизма в решении задач добавление лишней синхронизации в виде барьеров OpenMP повышает время выполнения задачи без влияния на качество выполнения программы.

Алгоритм

В качестве показательного алгоритма для этой проблемы был выбран параллельный алгоритм перемножения матриц:
Проблемный код:

```
Matrix matmul(Matrix &A, Matrix &B)
{
    int size = A.size;
    Matrix C(size);
    C.generate();
    for (int i = 0; i < size; i++)
    {
        for (int k = 0; k < size; k++)
        {
#pragma omp parallel for
            for (int j = 0; j < size; j++)
            {
                C.data[i * size + j] += A.data[i * size + k] * B.data[k * size + j];
            }
#pragma omp barrier
        }
    }
    return C;
}
```

Код без проблемы:

```
Matrix matmul(Matrix &A, Matrix &B)
{
    int size = A.size;
    Matrix C(size);
    C.generate();
    for (int i = 0; i < size; i++)
    {
        for (int k = 0; k < size; k++)
        {
#pragma omp parallel for
            for (int j = 0; j < size; j++)
            {
                C.data[i * size + j] += A.data[i * size + k] * B.data[k * size + j];
            }
        }
    }
    return C;
}
```

Компиляция и запуск

Все вычисления производились на машине Polus.

Сама программа написана на языке C++ и состоит из файлов:

- main_barriers.cpp / main_no_barriers.cpp (в зависимости от того, как мы хотим запустить с или без барьеров соответственно)
- matrix.cpp (Реализация структуры Matrix из практикума прошлого семестра)
- matrix.h

Компилировалась с использованием Makefile:

```
all: main

main_no_barriers: *.cpp *.h
    g++ main_no_barriers.cpp matrix.cpp -o prog -std=c++17 -fopenmp

main_barriers: *.cpp *.h
    g++ main_barriers.cpp matrix.cpp -o prog -std=c++17 -fopenmp

omp_polus_no_barriers: *.cpp *.h
    xlc++ main_no_barriers.cpp matrix.cpp -o prog -Wall -std=c++11 -qsmp=omp -fopenmp

omp_polus_barriers: *.cpp *.h
    xlc++ main_barriers.cpp matrix.cpp -o prog -Wall -std=c++11 -qsmp=omp -fopenmp

clean:
    rm -rf ./prog
```

Запуск производился постановкой в очередь с помощью lsf-файлов вида:
Где **i** – количество потоков, на которых будет запускаться программа, **M** – количество ядер

Для 1 и 2 потоков:

```
#BSUB -n 1
#BSUB -W 00:30
#BSUB -o \"./out_files/j/i.out\"
#BSUB -e \"./err_files/j/i.err\"
#BSUB -R \"span[hosts=1]\"
OMP_NUM_THREADS=i ./prog j
```

Для 4, 8, 16 и 32 потоков:

```
#BSUB -W 00:15
#BSUB -o "./out_files/j/i.out"
#BSUB -e "./err_files/j/i.err"
#BSUB -R "affinity[core(M)]"
OMP_NUM_THREADS=i
/polusfs/lsf/openmp/launchOpenMP.py ./prog j
```

(**M = i / 2**), **j** – размер матрицы, на которой будут производиться вычисления.

Оценка результативности изменений проводилась по средством измерения времени с помощью функции omp_get_wtime().

Результаты

	Nthreads	MatrixSize	Time_With	Time_Without	diff
0	1	1000	5.478742	1.739826	3.7389159999999997
1	1	4000	359.641975	72.92872	286.713255
2	1	6000	221.3996666666667	279.4656	-58.065933333333305
3	2	1000	5.9555125	2.4347079999999997	3.5208045000000006
4	2	4000	230.573325	70.09836	160.474965
5	2	6000	186.81866666666667	201.80599999999998	-14.98733333333331
6	4	1000	6.38507	2.329362	4.055707999999999
7	4	4000	209.1225	63.18244	145.94006000000002
8	4	6000	539.39025	174.53640000000001	364.85385
9	8	1000	5.8430025	2.4442166666666667	3.398785833333333
10	8	4000	143.08925	54.946133333333336	88.14311666666666
11	8	6000	338.792	140.2372	198.55479999999997
12	16	1000	2.9131625	2.8663600000000002	0.046802499999999636
13	16	4000	50.44655	51.066024999999996	-0.6194749999999942
14	16	6000	127.66199999999999	125.5815	2.0804999999999865
15	32	1000	4.91411	5.484427500000001	-0.5703175000000007
16	32	4000	88.41575	98.04797500000001	-9.632225000000005
17	32	6000	188.20925	190.77349999999998	-2.564249999999987

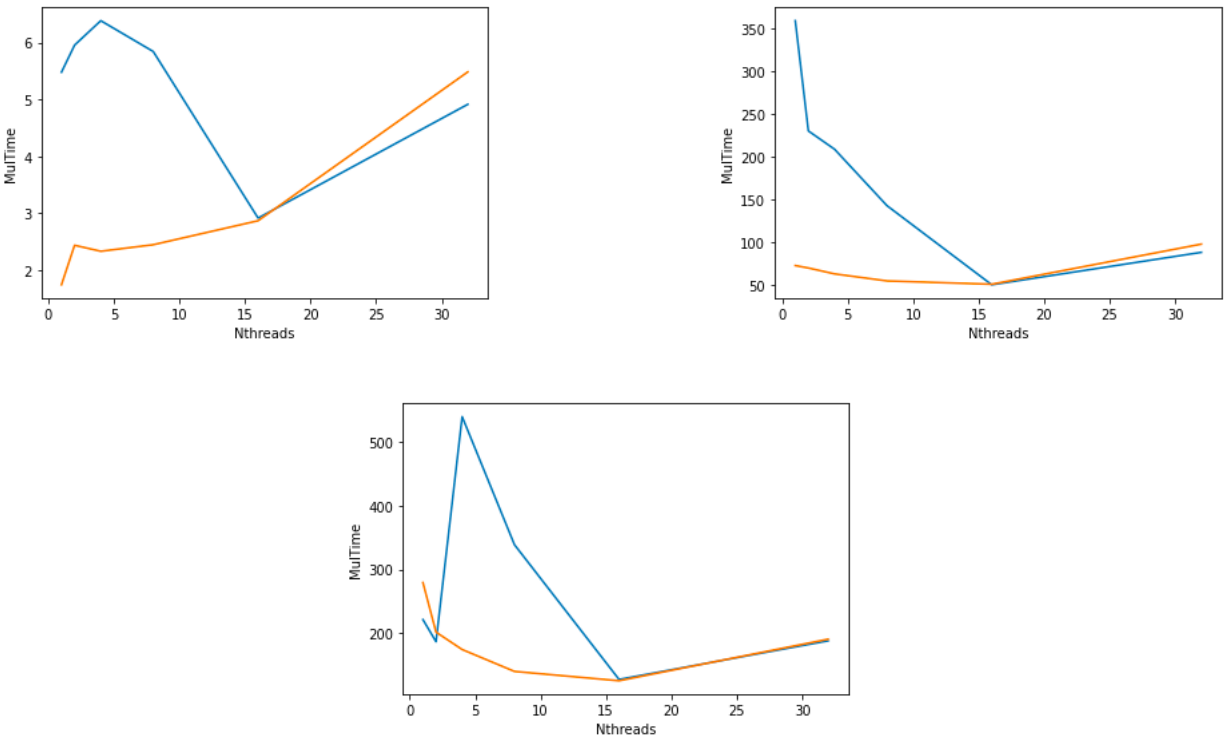

Рисунок 1. Графики времени выполнения программы: Синим цветом с "ошибочным" использованием барьеров, оранжевым цветом без использования барьеров. Графика по порядку для матриц размером 1000, 4000, 6000.