

Лабораторная работа №1.

Задание 10.

Программа для определения обратной матрицы для квадратной матрицы с использованием технологии OpenMP.

Работу выполнила:
Лаврова А.К., 19.Б05-пу

Постановка задачи:

Для квадратной матрицы A необходимо найти такую обратную A^{-1} , что $A \cdot A^{-1}$ будет равна единичной матрице.

Необходимо решить задачу с помощью технологии параллельного программирования OpenMP.

Ход работы:

Для создания программы был выбран язык C++.

В общем случае алгоритм построения обратной матрицы методом Гаусса выглядит следующим образом:

1. Присоединяем к исходной матрице единичную. Все элементарные преобразования, которые мы будем делать с исходной матрицей, мы будем делать и с единичной.

2. Приводим исходную матрицу к единичной с помощью элементарных преобразований. Для этого используем метод Гаусса: вычитаем строку с индексом i из всех остальных с таким коэффициентом, чтобы в i -ом столбце было только одно число, не равно нулю, и находилось в i -ой строке. Если элемент на позиции $[i, i]$ равен нулю, меняем i -ую строку с местами со строкой, в которой i -ый элемент не нулевой. Если такой строки нет, то определитель матрицы равен нулю и обратной матрицы не существует.

3. Когда получили диагональную матрицу, каждую строку делим на единственный ненулевой элемент в ней, получая единичную матрицу. При этом присоединенная матрица в результате этих преобразований стала обратной к исходной.

Однако так как задачу нужно было разбить на потоке, был выбран следующий алгоритм:

Задачу вычитания строки, умноженной на коэффициент, из остальных строк можно распределить на потоки (OpenMP). Для этого у каждого процесса/потока должен быть доступ к своей части матрицы. Добавляем директивы

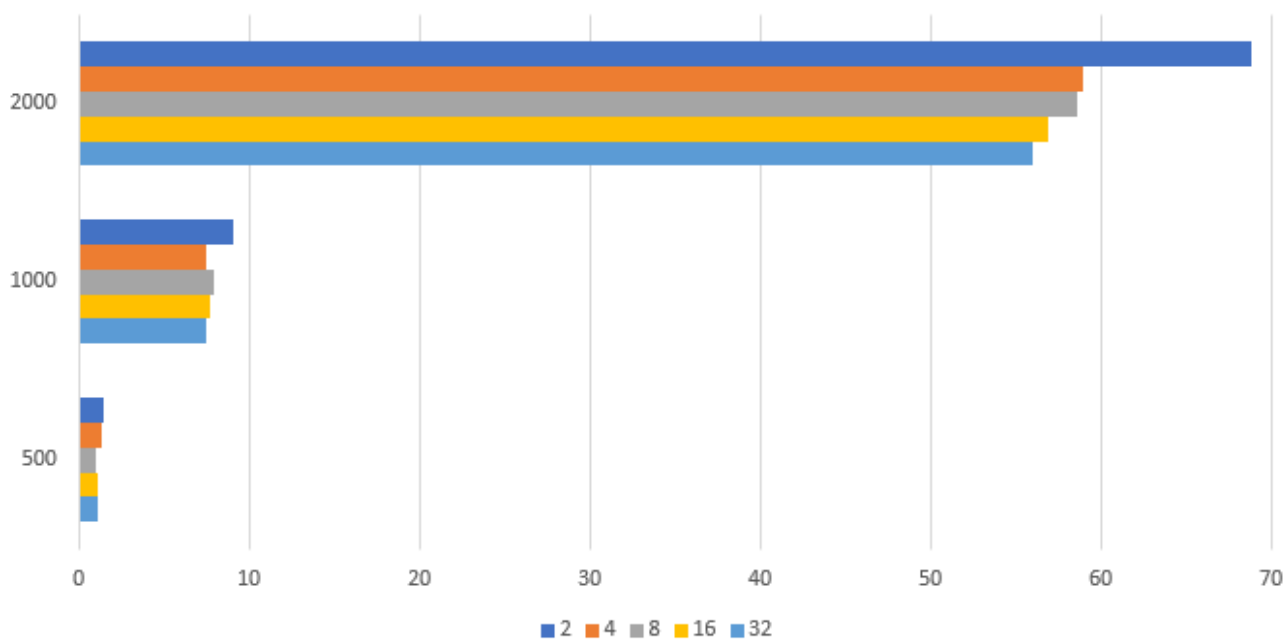
```
omp parallel for shared(m, I) private(i, j, k).
```

Исходный код программы может быть найден здесь: [github](#)

Запустим нашу программу на матрицах размером 500, 1000 и 2000 с количеством потоков от 2 до 32. Значения элементов в матрице выбираются случайно от 0 до 100. Результаты представлены в таблице (усредненное значение после 3х запусков):

Количество потоков	Время (в сек)
Матрица 500x500	
2	1,495
4	1,375
8	0,955
16	1,119
32	1,118
Матрица 1000x1000	
2	9,095
4	7,483
8	7,924
16	7,765
32	7,49
Матрица 2000x2000	
2	68,823
4	58,924
8	58,641
16	56,93
32	55,981

Построение в зависимости от времени



Вывод

В ходе работы был реализован алгоритм вычисления обратной матрицы с использованием распараллеливания программы с помощью библиотеки OpenMP. В результате использования программы на квадратной матрице большого размера можно заметить существенное сокращение времени подсчета (почти на 20% для $n=2000$). Однако для некоторых матриц, полученных случайным путем, увеличение количества потоков приводило к увеличению времени. Это особенно чувствовалось в экспериментах на маленьких матрицах (на 500 и 1000 элементов), где инициализация потоков и их локальных переменных, а также обращение к общим переменным и массивам, требовало больше времени, чем вычисление исходной задачи.