**МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ**

Федеральное ГОСУДАРСТВЕННОЕ бюджетное ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ

«ПОВОЛЖСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ»

Факультет информатики и вычислительной техники

Кафедра ИиСП

**Отчёт по первой лабораторной работе по дисциплине**

**"Параллельное программирование"**

Выполнил:

студент группы ПС-31

факультета Информатики

и Вычислительной Техники

специальности «Программная инженерия»

Колчин И. А.

Научный руководитель:

Филимонов

Александр

Александрович

г. Йошкар-Ола

2016

# Постановка задачи

Найти алгебраическое дополнение для каждого элемента матрицы.

Входные данные: произвольная матрица А размерности n х n. Заполнение начальной матрицы реализовать в главном потоке приложения.

1. Реализовать последовательный вариант программы для указанного варианта.
2. Реализовать параллельный вариант программы. Количество потоков выполнения должно являться входным параметром задачи.
3. После завершения программа должна выдавать время своей работы. Подобрать размеры матриц таким образом, чтобы время работы последовательного варианта составляло не менее одной секунды.
4. Посчитать параметры качества вашей параллельной программы и построить в Excel графики для количества потоков = [1;16] и количестве задействованных ядер = [1;4] (используйте функцию SetAffinityMask).
5. Время выполнения
6. Ускорение
7. Эффективность распараллеливания.

# Графики

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | Время | | | | | | | |
|  | Debug | | | | Release | | | |
|  | Количество процессоров | | | | | | | |
| Потоки | 1 | 2 | 3 | 4 | 1 | 2 | 3 | 4 |
| 1 | 51,24218 | 51,24218 | 51,24218 | 51,24218 | 14,038 | 14,038 | 14,038 | 14,038 |
| 2 | 51,1903 | 52,03 | 52,03 | 52,03 | 12,1935 | 9,08713 | 9,08713 | 9,08713 |
| 4 | 25,0047 | 21,653 | 17,093 | 13,21093 | 5,10923 | 3,1725 | 2,7533 | 1,8409 |
| 6 | 6,272 | 4,48 | 2,688 | 1,65104 | 3,10735 | 1,8593 | 1,23109 | 0,61198 |
| 8 | 8,158 | 3,978 | 2,134 | 1,61328 | 1,12453 | 0,61931 | 0,45927 | 0,328125 |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | Ускорение | | | | | | | |
|  | Debug | | | | Release | | | |
|  | Количество процессоров | | | | | | | |
| Потоки | 1 | 2 | 3 | 4 | 1 | 2 | 3 | 4 |
| 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 2 | 1,001013 | 0,984858 | 0,984858 | 0,984858 | 1,481341 | 2,018405 | 2,018405 | 2,018405 |
| 4 | 2,049302 | 2,366516 | 2,997846 | 3,878772 | 2,01229 | 2,092471 | 2,124572 | 2,14332 |
| 6 | 8,16999 | 11,43799 | 19,06331 | 31,0363 | 4,18056 | 3,308687 | 8,351339 | 16,79998 |
| 8 | 6,281218 | 12,88139 | 24,01227 | 31,76273 | 9,142709 | 16,60114 | 22,38607 | 31,33333 |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | Эффективность распараллеливания | | | | | | | |
|  | Debug | | | | Release | | | |
|  | Количество процессоров | | | | | | | |
| Потоки | 1 | 2 | 3 | 4 | 1 | 2 | 3 | 4 |
| 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 2 | 1 | 0,983861 | 0,983861 | 0,983861 | 1 | 1,341843 | 1,341843 | 1,341843 |
| 4 | 1 | 1,154791 | 1,462862 | 1,892728 | 1 | 1,610474 | 1,855675 | 2,775398 |
| 6 | 1 | 1,4 | 2,333333 | 3,798818 | 1 | 1,671247 | 2,524064 | 5,077535 |
| 8 | 1 | 2,050779 | 3,822868 | 5,056779 | 1 | 1,815779 | 2,448516 | 3,427139 |

# Вывод

Благодаря распараллеливанию программы удалось добиться значительного ускорения работы. При выполнении работы были выведены следующие закономерности:

1. чем больше потоков работает на одном процессоре тем быстрее обработка, но после определённого количества потоков(зависит от тактовой частоты процессора) производительность снижается, потому-то нужно определённое кол-во ресурсов процессора на синхронизацию потоков.
2. Чем больше задействовано процессоров тем быстрее обработка.

Эффективность распараллеливания сначала возрастает, затем падает, падение происходит потому что обрабатываемая матрица конечна и есть минимум данных, которые может обработать поток. Например потоку можно поручить заполнить часть матрицы, минимальное кол-во - одна ячейка.