МИНОБРНАУКИ РОССИИ

Федеральное государственное автономное образовательное учреждение высшего образования «Южный федеральный университет»

Институт математики, механики и компьютерных наук им. И.И.Воровича Информатики и вычислительного эксперимента

Направление подготовки 02.04.02 – Фундаментальная информатика и информационные технологии

ОТЧЕТ ПО ТЕМЕ  
«Блочное перемножение матриц»

Выполнил:  
магистр 1 года 6 группы

Галицкий Константин Владимирович

**Введение**

Необходимо реализовать программу блочного умножения матриц, первая матрица симметричная, вторая верхне-треугольная, обе они хранятся в виде ономерного массива (в вытянутом виде). Программа будет реализована в трех исполнениях:  
**Последовательная  
Паралельная "Блоки"  
Паралельная "Вычисл Ядра"**

После чего проведём сравнение, обсуждение и построим диаграммы наглядно дающие понять о результатах.

**Реализация программы**

Реализация будет проводиться на языке программирования C++ версии v142, в среде разработки Visual Studio 19 версии 16.3.1, на операционной системе Windows 10 версии 1903. Вставлять сюда скриншоты программы не совсем корректно, так как программа объемная, и они будут не информативны, в общем программа будет во вложениях.  
Но обсудим основные тезисы этой программы. Для начала хочется сказать, что будем дробить матрицы на блоки, что бы в дальнейшем можно было проще распараллелить, а так же для облегчения проверки программы на верность (отладки). И так, в программе реализована система, которая подбирает размер блока относительно величины матрицы, единственное, что нам необходимо указать, это минимальный размер блока, от которого он будет расширяться. После определения оптимального размера блока для данной матрицы, мы дробим матрицу на блоки. И приступаем к перемножению. В дальнейшем эти блоки помогут нам с добавлением OpenMP. В данной программе, так как тестировалась она на похожих ПК, было подобранно оптимальное количество потоков, для обработки всех типов матриц, которые участвовали в тестах.

**Тесты**

**Компиляторы:**

Тесты проводились на машине 1:

Compiler: GСС v. 9.2, compiler option: -Ofast;

GNU OpenMP v. 9.2;

OS: Ubuntu 19.04 (Linux);

Visual Studio 19 v16.3.1

OS: Windows 10 v1903

Тесты проводились на машине 1:  
**Процессор: Intel core i7 2600k 4700 МГц  
 Кеш 1-го уровня(инструкции) 128 Кб  
 Кеш 1-го уровня(данные) 128 Кб  
 Кеш 2-го уровня 1 Мб  
 Кеш 3-го уровня 8 Мб  
 4 Ядра 8 Потока  
ОЗУ: DDR3 12 Гигабайта 1333 МГц  
Устройство памяти: HDD накопитель 4 Терабайта**

Тесты проводились на машине 1: Windows 10 v1903 Visual Studio 19 v16.3.1

|  |  |  |  |
| --- | --- | --- | --- |
| Размер блока | Последовательная | Паралельная "Вычисл Ядра" | Паралельная "Блоки" |
| 2 | 55,546 | 33,9543 | 30,4466 |
| 4 | 44,546 | 23,44534 | 19,9466 |
| 6 | 46,3864 | 21,91531 | 23,7466 |
| 8 | 44,4868 | 22,91531 | 19,4466 |
| 12 | 43,54864 | 22,41531 | 18,9466 |
| 16 | 43,14864 | 22,15315 | 18,5466 |
| 20 | 43,02486 | 23,92546 | 20,4666 |
| 24 | 44,1543 | 20,0564 | 19,5466 |
| 36 | 44,2348 | 23,15464 | 19,6466 |
| 48 | 45,64864 | 23,9453 | 20,4466 |
| 72 | 46,4864 | 24,9553 | 21,4466 |
| 80 | 49,5345 | 25,81907 | 22,31327 |
| 96 | 60,456 | 26,7194 | 23,21327 |
| 120 | 71,3775 | 28,61973 | 24,11327 |
| 144 | 73,456 | 29,52006 | 25,01327 |
| 160 | 80,486 | 31,08706 | 25,91327 |
| 180 | 84,215 | 32,48739 | 26,81327 |
| 240 | 88,76925 | 33,88772 | 27,71327 |
| 360 | 93,3235 | 35,28805 | 28,61327 |
| 480 | 97,87775 | 36,68838 | 29,51327 |
| 720 | 102,432 | 38,08871 | 30,41327 |

Заметим, что потоки заметно улучшают время работы программы, так как матрица типоразмера 2880х2880.

График с наглядным видом оптимального блока 24х24

Тесты проводились на машине 1: Ubuntu 19.04 (Linux); Compiler: GСС v. 9.2, compiler option: -Ofast; GNU OpenMP v. 9.2;

|  |  |  |  |
| --- | --- | --- | --- |
| Размер блока | Последовательная | Паралельная "Вычисл Ядра" | Паралельная "Блоки" |
| 2 | 49,46215 | 30,23535 | 27,11184 |
| 4 | 39,66696 | 20,87742 | 17,76189 |
| 6 | 41,30579 | 19,51497 | 21,14568 |
| 8 | 39,61425 | 20,40544 | 17,31665 |
| 12 | 38,77884 | 19,9602 | 16,87142 |
| 16 | 38,42265 | 19,72676 | 16,51523 |
| 20 | 38,31243 | 21,30495 | 18,22493 |
| 24 | 39,31817 | 17,85966 | 17,4057 |
| 36 | 39,38985 | 20,61856 | 17,49475 |
| 48 | 40,64883 | 21,32262 | 18,20712 |
| 72 | 41,39484 | 22,22199 | 19,0976 |
| 80 | 44,10908 | 22,99116 | 19,86934 |
| 96 | 53,83437 | 23,79288 | 20,67076 |
| 120 | 63,55966 | 25,48507 | 21,47219 |
| 144 | 65,41051 | 26,28679 | 22,27361 |
| 160 | 71,67053 | 27,68215 | 23,07504 |
| 180 | 74,9911 | 28,92911 | 23,87646 |
| 240 | 79,04653 | 30,17606 | 24,67789 |
| 360 | 83,10196 | 31,42302 | 25,47931 |
| 480 | 87,15739 | 32,66997 | 26,28074 |
| 720 | 91,21282 | 33,91693 | 27,08216 |

Заметим, что потоки заметно улучшают время работы программы, так как матрица типоразмера 2880х2880.

График с наглядным видом оптимального блока 24х24

**Вывод**

Каждый метод будет отлично работать в той ситуации которая подходит ему больше всего, не всегда потоки помогают улучшить работу программы сделав её быстрее, зачастую, как раз наоборот, потому, к созданию параллельного аналога программы надо подходить с полным осознанием логики и работы программы…