Умножение матриц с использованием OpenMP

Вариант 28

Галицкий К.В.

4 курс 2 группа

**Введение**

Необходимо реализовать программу блочного умножения матриц, первая матрица симметричная, вторая нижнее-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 v141, в среде разработки Visual Studio 17 версии 15.9, на операционной системе Windows 10 версии 1803. 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 нас с добавлением OpenMP. В данной программе, так как тестировалась она на одном моём ПК, было подобранно оптимальное количество потоков, для обработки всех типов матриц, которые участвовали в тестах.

**Тесты**

Тесты проводились на машине:  
**Процессор: Intel core i7 2600k 4700 МГц  
 Кеш 1-го уровня 64 Кб  
 Кеш 2-го уровня 1024 Кб  
 Кеш 3-го уровня 5 Мб  
 4 Ядра 8 Потока  
ОЗУ: 12 Гигабайта 1333 МГц  
Устройство памяти: HDD накопитель 4 Терабайта**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | 8 | 16 | 32 | 64 | 80 | 144 | 160 |
| Последовательная | 6,4587 | 3,7 | 3,3045 | 3,16827 | 3,26827 | 3,36827 | 3,46827 |
| Паралелльные "Блоки" | 7,5483 | 4,2385 | 3,9572 | 2,2546 | 2,3546 | 2,4546 | 2,5546 |
| Паралелльные "Вычислительные ядра" | 4,56745 | 1,7654 | 1,3543 | 1,12543 | 1,22543 | 1,32543 | 1,42543 |

Заметим, что потоки заметно улучшают время работы программы, так как матрица типоразмера 2048х2048, при малых матрицах мы бы заметили, что потоки как раз наоборот ухудшают время работы программы.

График с наглядным видом оптимального блока 64х64

**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…