

Описание и результаты тестирования перемножения матриц

В своем решении отборочного задания я использовал 3 различных реализации перемножения:

- Алгоритм Штрассена
- Перемножение *за куб*, использующее транспонирование второй матрицы путем разбиения на небольшие подматрицы (таким образом, подматрицы транспонируются в кэше, доступ к которому в разы быстрее)
- Перемножение *за куб*, использующее транспонирование второй матрицы "по необходимости" (то есть транспонирование по одному столбцу, с которым в данный момент идет работа)

1 Результаты измерений

Результаты представлены для матриц различного размера типа `double`, которые генерировались с помощью `Math.random()`.

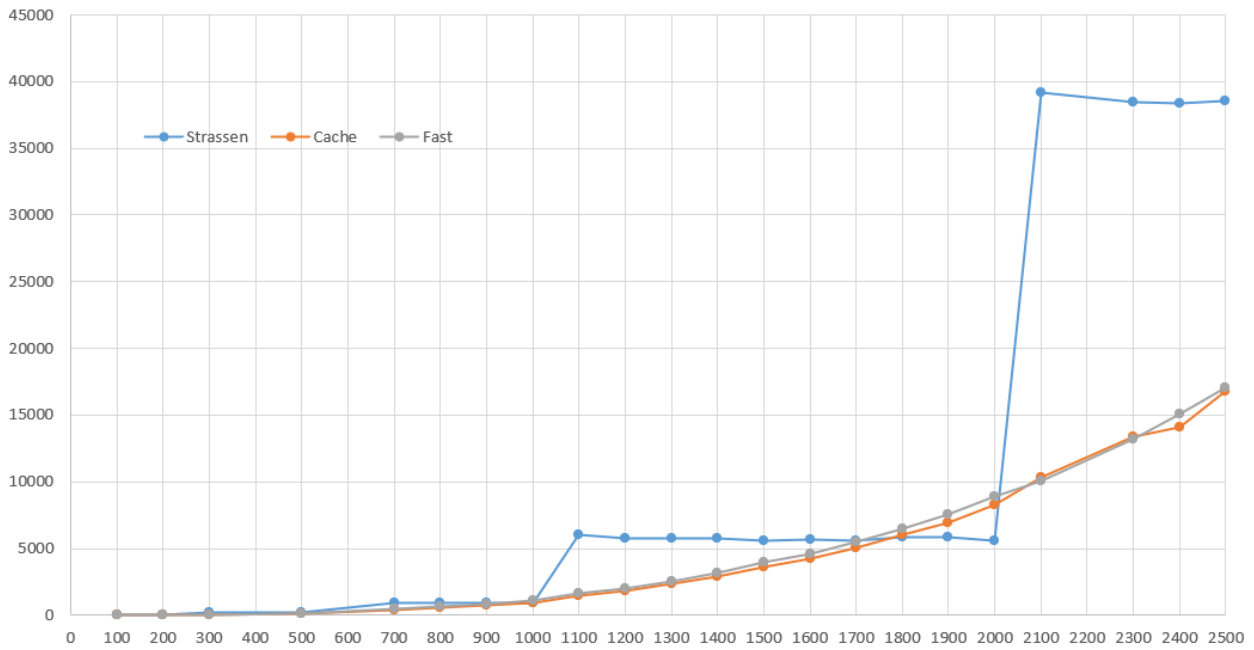
Приведены измерения для квадратных матриц, так как алгоритм Штрассена работает только для них. Если будет требоваться перемножение не квадратных матриц, будет выбран оптимальный метод из оставшихся двух

Замеры проводились на процессоре AMD Ryzen 5 3500U,
ОС Windows 10

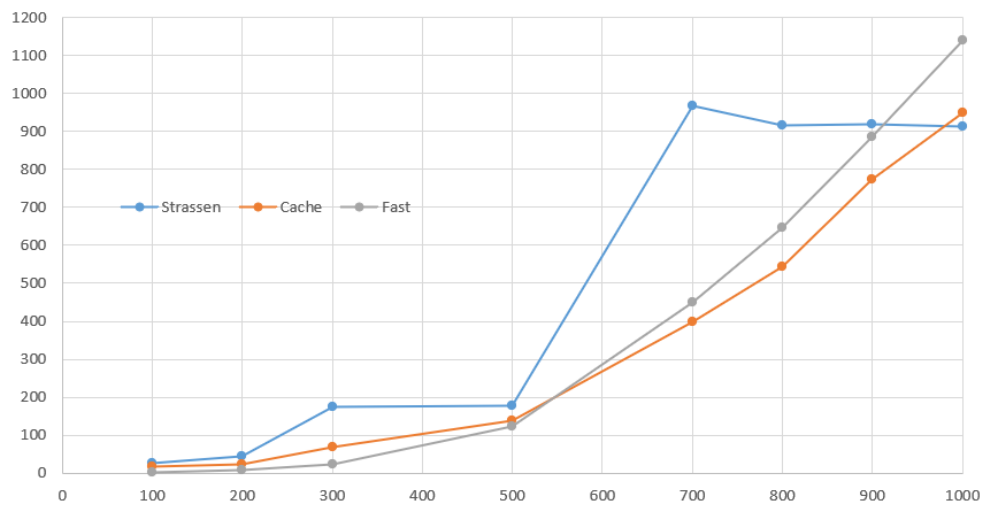
При этом, на ОС Linux особой разницы во времени не обнаружено, только реализация с кэшированным транспонированием работает на 5-10% быстрее.

Лучший показатель для матриц 1000x1000 составляет 800ms при работе *от розетки*

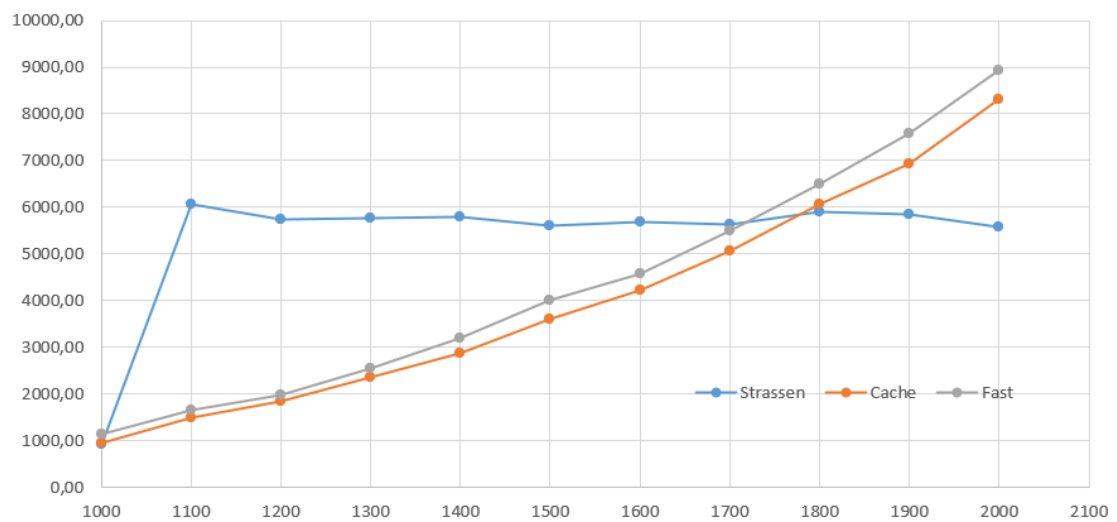
Диапазон 100x100 - 2500x2500



Диапазон 100x100 - 1000x1000



Диапазон 1000x1000 - 2000x2000



2 Выводы

Таким образом, видно, что свои лучшие цифры алгоритм Штрассена показывает при размерах матрицы, равных степеням двойки, а транспонирование с помощью кэша почти всегда опережает транспонирование по столбцам

Size	Strassen	Cache	Fast	Оптимальный
100	26	17	1	Fast
200	45	24	7	
300	176	69	23	
500	179	139	122	
700	968	400	450	Cache
800	916	543	647	
900	920,00	775	886	
1000	912,00	948,00	1141	Strassen
1100	6063	1491	1658	Cache
1200	5751	1852	1975	
1300	5766,00	2351	2545	
1400	5807	2876	3190	
1500	5616	3598	4016	
1600	5688	4211	4564	
1700	5626	5065	5487	
1800	5902	6051	6503	Strassen
1900	5835	6938	7572	
2000	5575	8305	8926	
2100	39149	10333	10107	Cache
2300	38460	13424	13244	
2400	38365	14136	15104	
2500	38575	16801	17051	