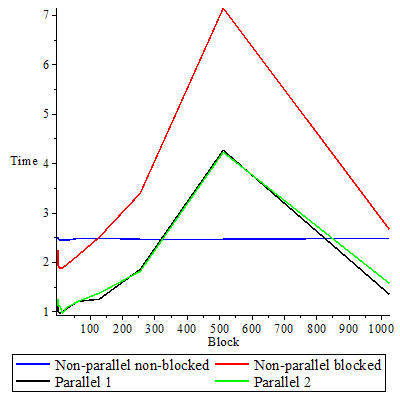
Таблица при 8 потоках:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| n | Block Size (elements) | Размер блока | Обычное умножение | Non-parallel time | Parallel time 1 | Parallel time 2 |
| 1024 | 4 | 2 | 2.5 | 2.23 | 1.24 | 1.26 |
| 1024 | 16 | 4 | 2.46 | 1.92 | 0.987 | 1.1 |
| 1024 | 64 | 8 | 2.44 | 1.87 | 0.967 | 1.1 |
| 1024 | 256 | 16 | 2.43 | 1.87 | 0.987 | 0.969 |
| 1024 | 1024 | 32 | 2.44 | 1.96 | 1.07 | 1.09 |
| 1024 | 4096 | 64 | 2.47 | 2.13 | 1.202 | 1.205 |
| 1024 | 16384 | 128 | 2.49 | 2.51 | 1.25 | 1.37 |
| 1024 | 65536 | 256 | 2.46 | 3.4 | 1.86 | 1.81 |
| 1024 | 262144 | 512 | 2.46 | 7.14 | 4.27 | 4.23 |
| 1024 | 1048576 | 1024 | 2.49 | 2.66 | 1.34 | 1.57 |

График при n = 1024:



Вывод:

Пусть блочный алгоритм и эффективнее обычного, с определенного размера блока начинает происходить очень много лишних умножений на нулевые элементы, и блочный алгоритм работает медленнее. При блоке = 512 их происходит очень много, из-за чего блочные алгоритмы сильно страдают, при размере блока = n этого не происходит т.к. вместо блочного алгоритма работает обычный.