Отчет

Индивидуальная работа

по курсу «Суперкомпьютеры»

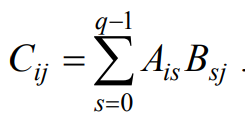
подготовил: Тихонов Алексей

вариант 46

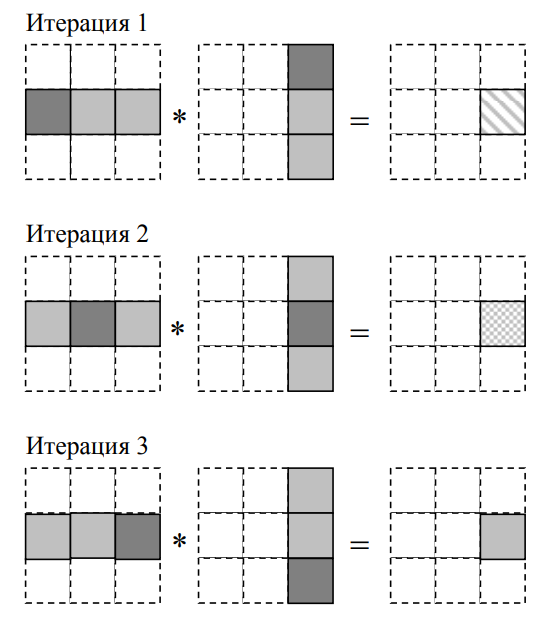
Разработка параллельного алгоритма матричного умножения.

Для разработке параллельного алгоритма матричного умножения воспользуемся блочной схемой представления матриц.

При блочном разбиение матрицы разделения данных исходные матрицы А, В и результирующая матрица С представляются в виде наборов блоков. Для более простого изложения следующего материала будем предполагать далее, что все матрицы являются квадратными размера n×n, количество блоков по горизонтали и вертикали являются одинаковым и равным q (т.е. размер всех блоков равен k×k, k=n/q). При таком представлении данных операция матричного умножения матриц А и B в блочном виде может быть представлена где, каждый блок Cij матрицы C определяется в соответствии



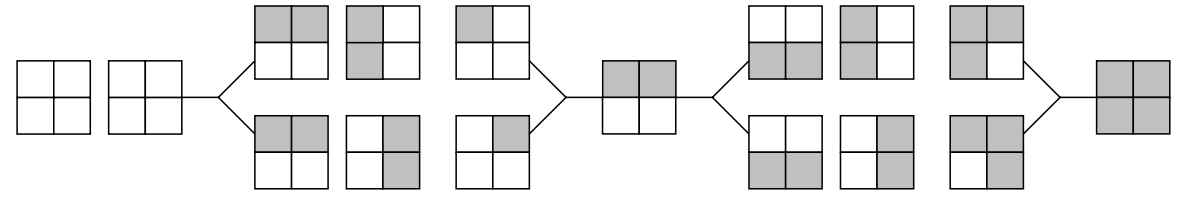
При блочном разделении данных принят подход, при котором базовые подзадачи отвечают за вычисления отдельных блоков матрицы C и при этом в подзадачи на каждой итерации расчетов обрабатывают только по одному блоку исходных матриц A и B. Как уже отмечалось выше, для вычисления блока результирующей матрицы поток должен выполнить умножение горизонтальной полосы матрицы А на вертикальную полосу матрицы В

. 

Для повышения эффективности использования кэш-памяти количество разбиений матриц должно быть таким, чтобы в кэш одновременно могли быть помещены три матричных блока – блоки матриц A, B и C. Если блоки матриц могут быть помещены в кэш полностью, то при вычислении результата умножения матричных блоков не происходит многократного чтения элементов блока в кэш, и, следовательно, затраты на загрузку данных из оперативной памяти существенно сокращаются.

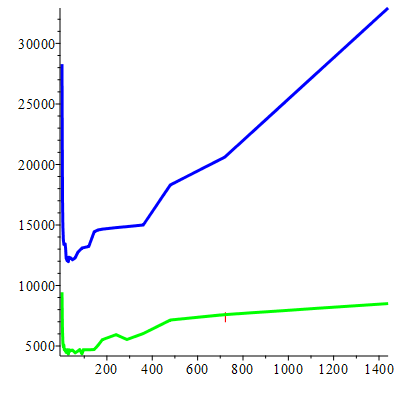
Реализация параллельного кэш-оптимизированного блочного алгоритма

Для распараллеливания представленного блочного алгоритма умножения матриц воспользуйтесь подходом, изложенным при рассмотрении блочного алгоритма. Пусть, как и ранее, поток отвечает за вычисление блока результирующей матрицы. Однако теперь, когда количество блоков определяется не количеством потоков, а объемом кэш-памяти, число блоков может существенно превосходить число доступных потоков. Поэтому каждый поток должен вычислять несколько матричных блоков. Распределите между потоками параллельной программы итерации второго цикла (цикла по переменной m). При таком распределении нагрузки на каждой итерации внешнего цикла поток последовательно выполняет поблочное умножение горизонтальной полосы матрицы A на несколько вертикальных полос матрицы B.



Результаты экспериментов

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Размер матрицы | Размер блока | Время выполнения | | |
| Последовательный | Параллельный  (размер блока 720)  для 4 потоков | Параллельный оптимизированный |
| 8000 | 2 | 28293 | 7375 | 9441 |
| 8000 | 4 | 17990 | 7375 | 6210 |
| 8000 | 6 | 14796 | 7375 | 5240 |
| 8000 | 8 | 13706 | 7375 | 4966 |
| 8000 | 10 | 13411 | 7375 | 5054 |
| 8000 | 12 | 12994 | 7375 | 4689 |
| 8000 | 16 | 12498 | 7375 | 4632 |
| 8000 | 18 | 13000 | 7375 | 4636 |
| 8000 | 20 | 12281 | 7375 | 4476 |
| 8000 | 24 | 12065 | 7375 | 4674 |
| 8000 | 30 | 11971 | 7375 | 4335 |
| 8000 | 32 | 12330 | 7375 | 4681 |
| 8000 | 36 | 12309 | 7375 | 4561 |
| 8000 | 40 | 12277 | 7375 | 4650 |
| 8000 | 48 | 12127 | 7375 | 4656 |
| 8000 | 60 | 12276 | 7375 | 4434 |
| 8000 | 72 | 12751 | 7375 | 4587 |
| 8000 | 80 | 12900 | 7375 | 4702 |
| 8000 | 90 | 13097 | 7375 | 4353 |
| 8000 | 96 | 13121 | 7375 | 4686 |
| 8000 | 120 | 13222 | 7375 | 4691 |
| 8000 | 144 | 14436 | 7375 | 4708 |
| 8000 | 160 | 14583 | 7375 | 5032 |
| 8000 | 180 | 14657 | 7375 | 5525 |
| 8000 | 240 | 14776 | 7375 | 5936 |
| 8000 | 288 | 14865 | 7375 | 5531 |
| 8000 | 360 | 14993 | 7375 | 6026 |
| 8000 | 480 | 18312 | 7375 | 7145 |
| 8000 | 720 | 20609 | 7375 | 7593 |
| 8000 | 1440 | 32928 | 7375 | 8503 |



Вывод

Для повышения эффективности использования кэш-памяти количество разбиений матриц должно быть таким, чтобы в кэш одновременно могли быть помещены три матричных блока – блоки матриц A, B и C. Если блоки матриц могут быть помещены в кэш полностью, то при вычислении результата умножения матричных блоков не происходит многократного чтения элементов блока в кэш, и, следовательно, затраты на загрузку данных из оперативной памяти существенно сокращаются.

Приложение

