

Постановка задачи

В данной работе необходимо:

1. Написать программу блочного параллельного умножения двух матриц $C = A * B$.
2. Написать программу блочного параллельного умножения матриц с распределением данных по блокам.
3. Провести эксперименты по сравнению скорости работы для матриц разного размера и разного размера блоков, сделать выводы. Сравнивать нужно 3 программы: обычного умножения матриц, блочного и блочного с распределением данных по блокам.

Провести численные эксперименты и построить таблицу сравнений времени выполнения различных программных реализаций решения задачи. Определить лучшие реализации. Проверить корректность (правильность) программ.

Обычное умножение матриц

Нужно сгенерировать матрицы A и B со случайными элементами выбранного размера. Будем считать, что все матрицы квадратные размера N

```
for(i = 0; i < N; i++) {
  for(j = 0; j < N; j++) {
    for (k = 0; k < N; k++) {
      C[i][j] = C[i][j]+A[i][k]*B[k][j]
    }
  }
}
```

Блочное умножение матриц

Для того, чтобы те данные, которые мы в циклах читаем из двух матриц, чаще оказывались в кэше процессора, можно обходить элементы матрицы C (результата) не сразу по всей строке и столбцу, а по квадратным блокам какого-то размера.

Тогда вместо 3-х вложенных циклов будет 5:

```
for(ib = 0; ib < N/B; ib++) { // обходим строку блоков
  for(jb = 0; jb < N/B; jb++) { // обходим столбец блоков
    for(i = 0; i < B; i++) { // перемножаем два блока
      for(j = 0; j < B; j++) {
        for (k = 0; k < B; k++) {
          ...
        }
      }
    }
  }
}
```

```
}  
}}  
}
```

Эти циклы тоже можно выполнять параллельно. Выберите, какой цикл лучше распараллелить с помощью OpenMP.

Размещение данных по блокам

Матрицы изначально хранятся в памяти в непрерывном массиве, по строкам. Чтобы обращения к памяти в программе блочного умножения чаще случались ближе друг к другу, исходные матрицы можно разместить по-другому. Для этого нужно разместить в памяти последовательно данные всех блоков, а внутри каждого блока разместить элементы по строкам.

Напишите программу параллельного блочного умножения матриц с предварительным распределением данных по блокам. Сравните время работы при разных размерах блоков с остальными программами.