



**Спецкурс: системы и средства параллельного  
программирования**

**Отчёт № 4**

**Параллельный алгоритм умножения матрицы на вектор**

**Разработка параллельной MPI программы и  
исследование ее эффективности**

Работу выполнил  
**Чепурнов А. В.**

## Постановка задачи и формат данных

**Задача:** Разработать параллельную программу с использованием технологии MPI, реализующую алгоритм умножения плотной матрицы на вектор  $Ab = c$ . Провести исследование эффективности разработанной программы на системе Blue Gene/P.

**Формат командной строки:** <имя файла матрицы A> <имя файла вектора b> <имя файла вектора c>

**Формат файла-матрицы:** Матрица представляется в виде бинарного файла следующего формата:

Тип	Значение	Описание
Число типа char	'd' – тип double	Тип элементов матрицы
Число типа int	N – натуральное число	Число строк матрицы
Число типа int	M – натуральное число	Число столбцов матрицы
Массив чисел типа double	$N \times M$ элементов	Массив элементов матрицы

**Формат файла-вектора:** Вектор представляется как матрица размера  $N \times 1$ .

## Описание алгоритма

Если  $N > M$ , элементы матрицы равномерно распределяются по процессам блоками строк, если  $N \leq M$  – блоками столбцов. Каждый процесс перемножает свой участок на вектор  $b$ . Вектор  $c$  находится как сумма векторов, полученных каждым процессом.

**Аппаратное обеспечение:** Исследования проводились на вычислительном комплексе IBM Blue Gene/P.

**Анализ времени выполнения:** Для оценки времени выполнения программы использовалась функция MPI\_Wtime().

**Анализ ускорения:** Ускорение, получаемое при использовании параллельного алгоритма для  $p$  процессоров, высчитывалось как отношение времени выполнения задачи на одном процессоре к времени параллельного выполнения задачи при использовании  $p$  процессоров.

### Основные функции:

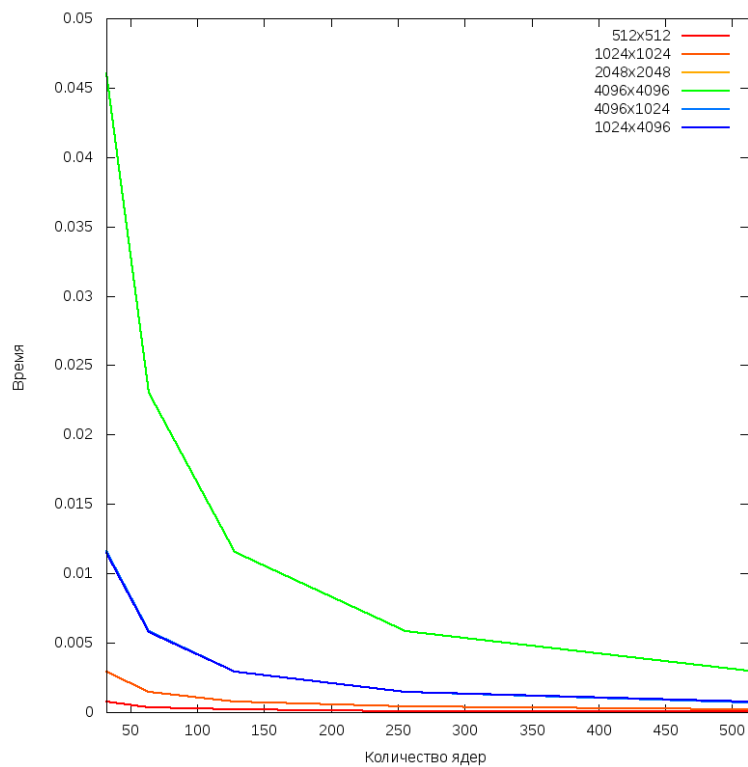
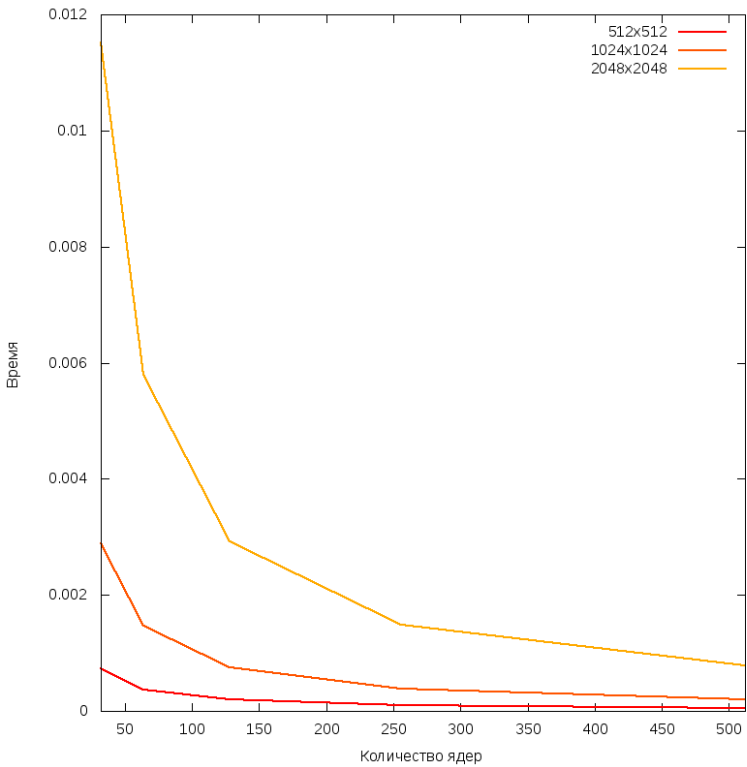
- **Разбор командной строки.** В рамках функции осуществляется анализ и разбор командной строки.
- **Параллельное чтение из файла.** В рамках функции осуществляется параллельное чтение элементов матрицы из файла с помощью средств MPI, а также анализ совместимости матрицы и вектора.
- **Умножение матрицы на вектор.** В рамках функции осуществляется умножение матрицы на вектор в зависимости от соотношения N и M.

## Результаты выполнения

Проводились умножения матриц  $512 \times 512$ ,  $1024 \times 1024$ ,  $2048 \times 2048$ ,  $4096 \times 4096$ ,  $4096 \times 1024$ ,  $1024 \times 4096$ . Алгоритм запускался на 32, 64, 128, 256 и 512 ядрах. Альтернативный вариант мэппинга для 512 ядер выбирался случайно.

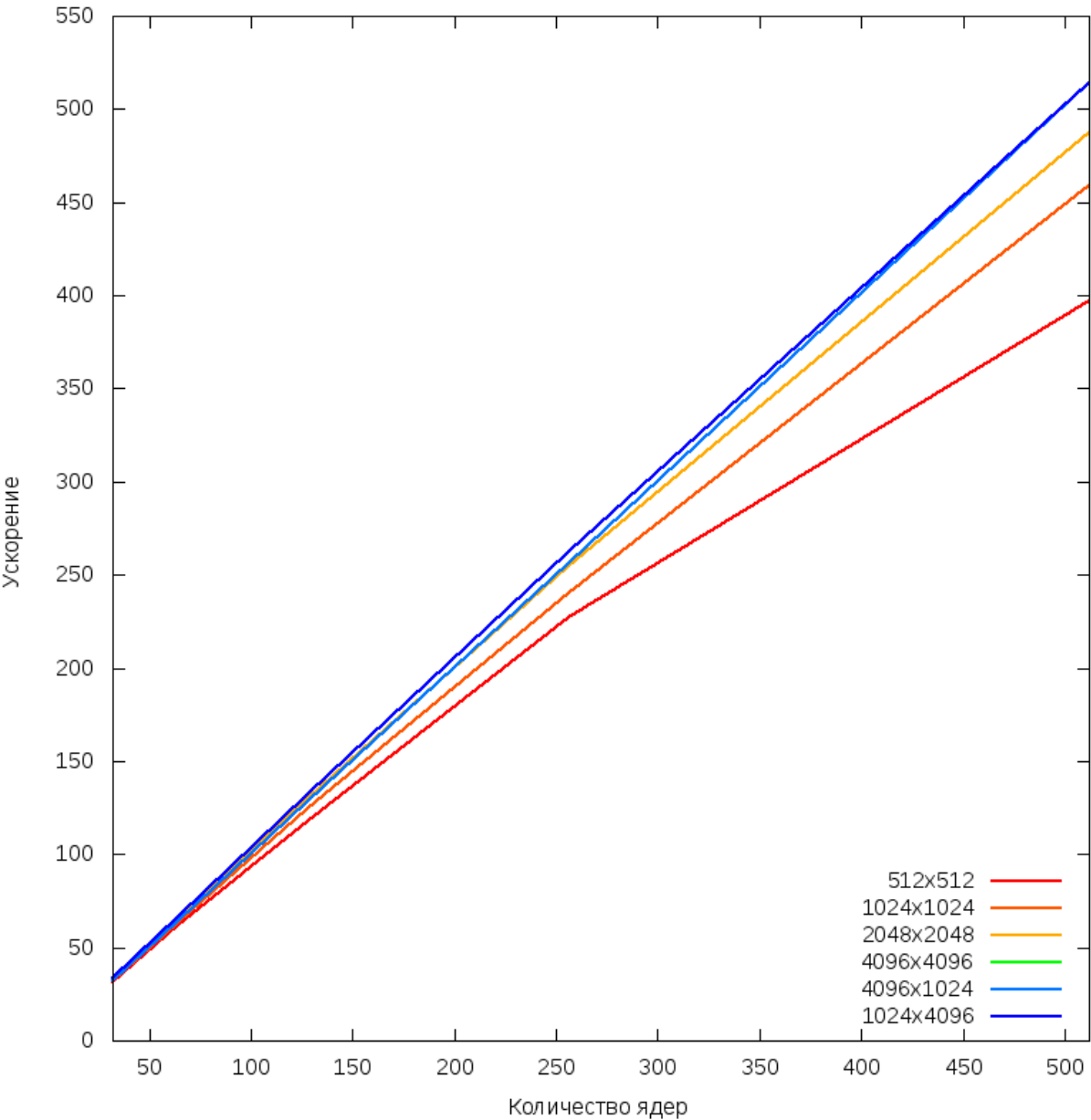
Среднее время выполнения процесса (в секундах):

n	m	мэппинг	32	64	128	256	512
512	512	XYZT	0.000731782	0.000374082	0.000195124	0.000101424	5.80804e-05
512	512	ZXYT					5.80831e-05
1024	1024	XYZT	0.0028919	0.00146165	0.000745832	0.000388001	0.00020282
1024	1024	YXZT					0.000202819
2048	2048	XYZT	0.0115113	0.00578307	0.00292195	0.00148927	0.000777478
2048	2048	TYXZ					0.000777479
4096	4096	XYZT	0.0460203	0.0230297	0.0115639	0.00584081	0.00297988
4096	4096	YZXT					0.00297988
4096	1024	XYZT	0.0116153	0.00581321	0.00290454	0.00145258	0.00072441
4096	1024	ZXYT					0.000724408
1024	4096	XYZT	0.0114962	0.00575566	0.00289182	0.00146036	0.00074562
1024	4096	ZYXT					0.000745617



Ускорение работы алгоритма:

п	м	мэппинг	32	64	128	256	512
512	512	XYZT	31.5015	61.6234	118.141	227.285	396.902
512	512	ZXYT					396.883
1024	1024	XYZT	32.1814	63.6715	124.781	239.859	458.858
1024	1024	YXZT					458.86
2048	2048	XYZT	32.9309	65.5494	129.734	254.539	487.573
2048	2048	TYXZ					487.572
4096	4096	XYZT	33.2955	66.5345	132.505	262.339	514.205
4096	4096	YZXT					514.205
4096	1024	XYZT	32.0495	64.0378	128.167	256.278	513.887
4096	1024	ZXYT					513.889
1024	4096	XYZT	33.3214	66.5552	132.466	262.311	513.759
1024	4096	ZYXT					513.761



## Основные выводы

Исследования показывают, что с увеличением количества ядер среднее время выполнения процесса уменьшается, а следовательно, ускоряется вся параллельная программа. Это связано с тем, что каждому процессу необходимо произвести меньше вычислений. Наибольшее ускорение наблюдается при распределении элементов матрицы по процессам блоками столбцов (при  $N \leq M$ ). Мэппинг данной параллельной программы не влияет на время её работы.