



**Спецкурс: системы и средства параллельного
программирования.**

Отчёт № 4.

**Параллельный алгоритм умножения матрицы на
вектор.**

Работу выполнил
Тимачев А. А.

Постановка задачи и формат данных.

Задача: Разработать параллельную программу с использованием технологии MPI, реализующую алгоритм умножения плотной матрицы на вектор $Ab = c$. Тип данных – double. 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/вектора: Матрица представляются в виде бинарного файла следующего формата:

Тип	Значение	Описание
Число типа char	T – d (double)	Тип элементов
Число типа uint64_t	N – натуральное число	Число строк матрицы
Число типа uint64_t	M – натуральное число	Число столбцов матрицы
Массив чисел типа T	$N \times M$ элементов	Массив элементов матрицы

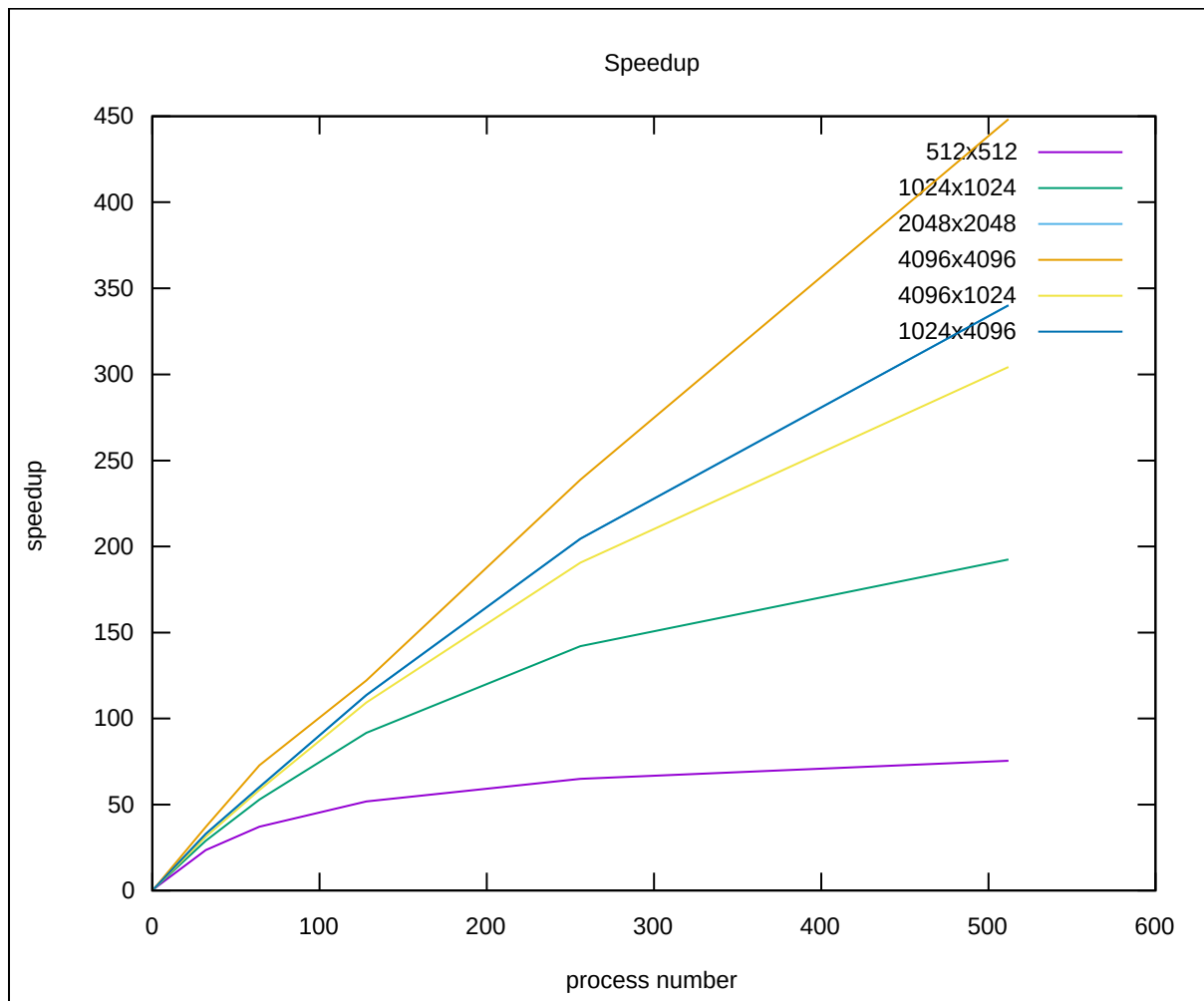
Элементы матрицы хранятся построчно.

Результаты выполнения.

Результаты:

Максимальное время								
Размеры		Количество процессов						
M	N	1	32	64	128	256	512	512 mapped
512	512	0.005132	0.000218	0.000138	0.000099	0.000079	0.000068	0.000068
1024	1024	0.021167	0.000732	0.000400	0.000231	0.000149	0.000110	0.000109
2048	2048	0.090143	0.002729	0.001499	0.000793	0.000441	0.000265	0.000265
4096	4096	0.404864	0.010961	0.005557	0.003315	0.001695	0.000903	0.000903
4096	1024	0.084616	0.002701	0.001445	0.000774	0.000444	0.000278	0.000278
1024	4096	0.101245	0.002664	0.001416	0.000794	0.000487	0.000332	0.000332

Суммарное время								
Размеры		Количество процессов						
M	N	1	32	64	128	256	512	512 mapped
512	512	0.005132	0.006966	0.008778	0.012573	0.020118	0.034509	0.034501
1024	1024	0.021167	0.023422	0.025549	0.029562	0.037945	0.056022	0.055751
2048	2048	0.090143	0.087318	0.095929	0.101444	0.112805	0.134360	0.134457
4096	4096	0.404864	0.350733	0.355603	0.424280	0.433656	0.460995	0.461343
4096	1024	0.084616	0.086407	0.092425	0.098998	0.113426	0.140731	0.141108
1024	4096	0.101245	0.085232	0.090589	0.101590	0.124475	0.169473	0.169389



Проводилось перемножение матриц с размерами 512x512, 1024x1024, 2048x2048, 4096x4096, 4096x1024 и 1024x4096 на соответствующий вектор. Зависимость ускорения от размеров и количества процессов представлена на графике.

Основные выводы.

Исследования показывают, что большие размеры матриц получают наибольшее ускорение при увеличении числа процессов. Однако для любых размеров матриц ускорение растет при увеличении числа процессов. Все это означает что данная задача хорошо распараллеливается.

Из таблиц можно увидеть, что рандомный мэппинг практически не оказывает на решение задачи никакого влияния. Это происходит из-за того, что данная задача не требует какой-либо особенной топологии, так как в ней происходит только общение сразу со всеми процессами при получении конечных результатов.