



**Курс: Системы и средства параллельного
программирования.**

Отчёт № 4.

**Параллельный алгоритм умножения матрицы на
вектор.**

Работу выполнил
Шахворостов Д. О.

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 – f (float)	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	0.00601258	0.000192	0.000096	0.000050	0.0000267	0.000015	0.0000150
1024	1024	0.0241893	0.000756	0.000382	0.000193	0.0000976	0.000050	0.0000508
2048	2048	0.0980117	0.003071	0.001536	0.000768	0.0003864	0.000195	0.0001957
4096	4096	0.40292	0.012583	0.006294	0.003149	0.0015826	0.000789	0.0007894
4096	1024	0.0967593	0.003031	0.001516	0.000759	0.0003822	0.000193	0.0001930
1024	4096	0.168498	0.010868	0.003635	0.001423	0.0007836	0.000463	0.0004630

ускорение								
Размеры		Количество процессов						
M	N	1	32	64	128	256	512	512 mapped
512	512	1	31,315521	62,631042	120,251600	225,190262	400,838667	400,838667
1024	1024	1	31,996429	63,322775	125,333161	247,841189	483,786000	476,167323
2048	2048	1	31,915239	63,809701	127,619401	253,653468	502,624103	500,826265
4096	4096	1	32,020981	64,016524	127,951731	254,593707	510,671736	510,412972
4096	1024	1	31,923227	63,825396	127,482609	253,164050	501,343523	501,343523
1024	4096	1	15,504049	46,354333	118,410401	215,030628	363,926566	363,926566

эффективность								
Размеры		Количество процессов						
М	N	1	32	64	128	256	512	512 mapped
512	512	1	0,978610	0,978610	0,939466	0,879649	0,782888	0,782888
1024	1024	1	0,999888	0,989418	0,979165	0,968130	0,944895	0,930014
2048	2048	1	0,997351	0,997027	0,997027	0,990834	0,981688	0,978176
4096	4096	1	1,000656	1,000258	0,999623	0,994507	0,997406	0,996900
4096	1024	1	0,997601	0,997272	0,995958	0,988922	0,979187	0,979187
1024	4096	1	0,484502	0,724286	0,925081	0,839963	0,710794	0,710794

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.