Московский Государственный Университет им. М.В. Ломоносова
Факультет Вычислительной Математики и Кибернетики
Кафедра Суперкомпьютеров и Квантовой Информатики

Спецкурс: системы и средства параллельного программирования.

Отчёт № 4.
Параллельный алгоритм умножения матрицы на вектор.

Работу выполнил
Тимачев А. А.

Москва 2018

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.

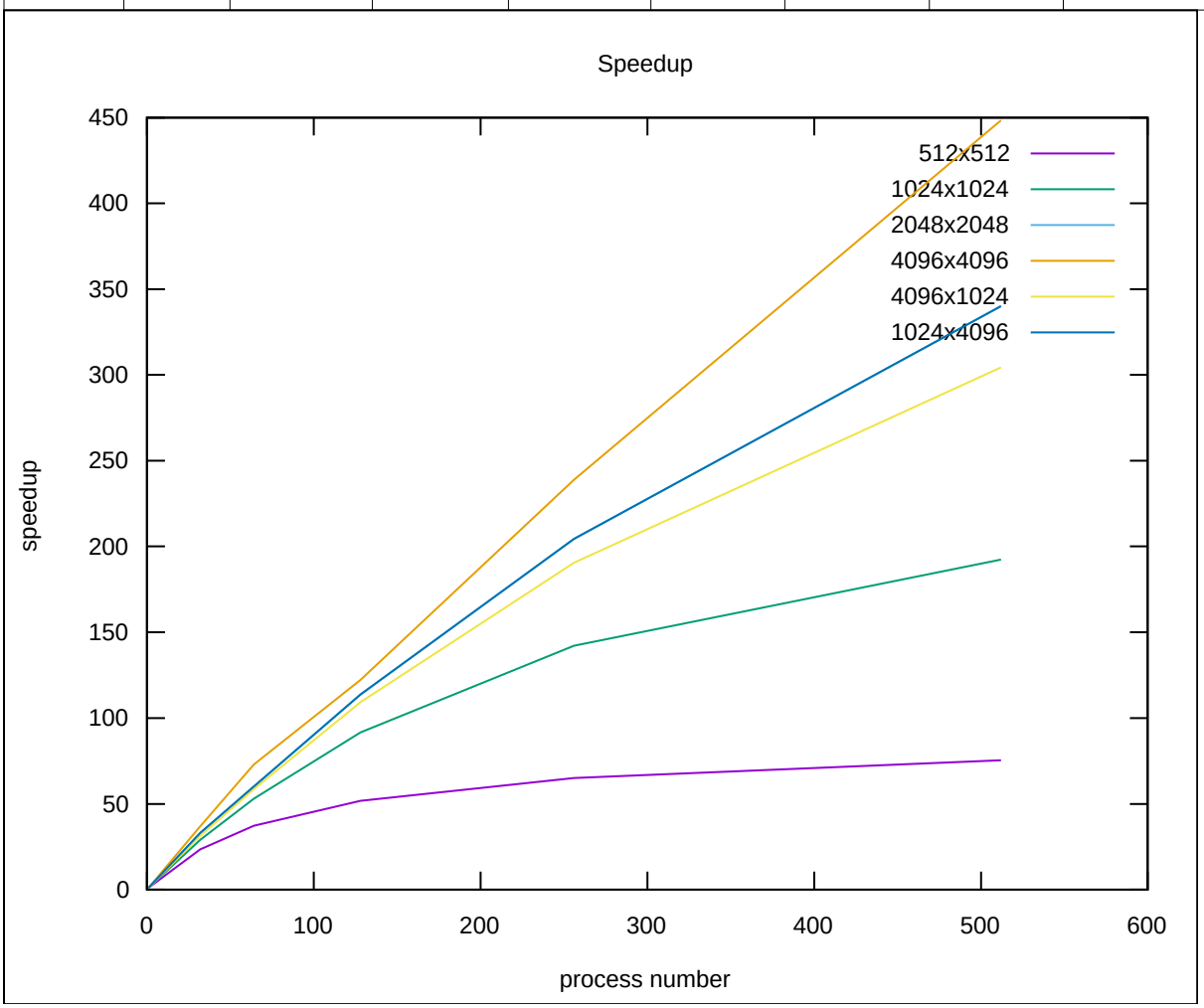
Результаты:

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

Ускорение								
Размеры		Количество процессов						
M	N	1	32	64	128	256	512	512 mapped
512	512	1.000000	23.541284	37.188406	51.838384	64.962025	75.470588	75.470588
1024	1024	1.000000	28.916667	52.917500	91.632035	142.06040	192.42727	194.19266
2048	2048	1.000000	33.031513	60.135424	113.67339	204.40589	340.16226	340.16226
4096	4096	1.000000	36.936776	72.856577	122.13092	238.85781	448.35437	448.35437
4096	1024	1.000000	31.327656	58.557785	109.32299	190.57657	304.37410	304.37410
1024	4096	1.000000	38.004880	71.500706	127.51259	207.89527	304.95481	304.95481

Эффективность								
Размеры		Количество процессов						
M	N	1	32	64	128	256	512	512 mapped
512	512	1.000000	0.735665	0.581069	0.404987	0.253758	0.147403	0.147403
1024	1024	1.000000	0.903646	0.826836	0.715875	0.554923	0.375835	0.379283
2048	2048	1.000000	1.032235	0.939616	0.888073	0.798461	0.664379	0.664379
4096	4096	1.000000	1.154274	1.138384	0.954148	0.933038	0.875692	0.875692
4096	1024	1.000000	0.978989	0.914965	0.854086	0.744440	0.594481	0.594481
1024	4096	1.000000	1.187652	1.117199	0.996192	0.812091	0.595615	0.595615



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.