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

Системы и средства параллельного программирования

Отчёт № 4
Параллельный алгоритм умножения матрицы на вектор

Работу выполнила
Зайденварг Елизавета

Москва 2020

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.

Требуется:

1. Разработать параллельную программу с использованием технологии MPI. Предусмотреть равномерное распределение элементов матрицы блоками строк или столбцов, в зависимости от соотношения m и n. Вектора b и c распределены по процессам равномерно.
2. Исследовать эффективность разработанной программы в зависимости от размеров матрицы и количества используемых процессов. Построить графики времени работы, ускорения и эффективности разработанной программы. Время на ввод/вывод данных не включать.
3. Исследовать влияние мэппинга параллельной программы на время работы программы.
4. Построить таблицы: времени, ускорения, эффективности.

m	**n**	**мэппинг**	**32**	**64**	**128**	**256**	**512**

Для 512 процессоров рассмотреть два варианта мэппинга – стандартный, принятый по умолчанию и произвольный. Для произвольного мэппинга предусмотреть генерацию строк файла для задания случайного значения XYZT.
Ускорение (speedup), получаемое при использовании параллельного алгоритма для p процессоров, определяется величиной:
Speedup(n) = T1(n)/Tp(n),

где T1(n)- время выполнения задачи на одном процессоре.
Tp(n)- время параллельного выполнения задачи при использовании p процессоров.

5. Построить графики – для каждого из заданных значений размеров матрицы (512x512, 1024x1024, 2048x2048, 4096x4096, 4096x1024, 1024x4096).

Формат командной строки

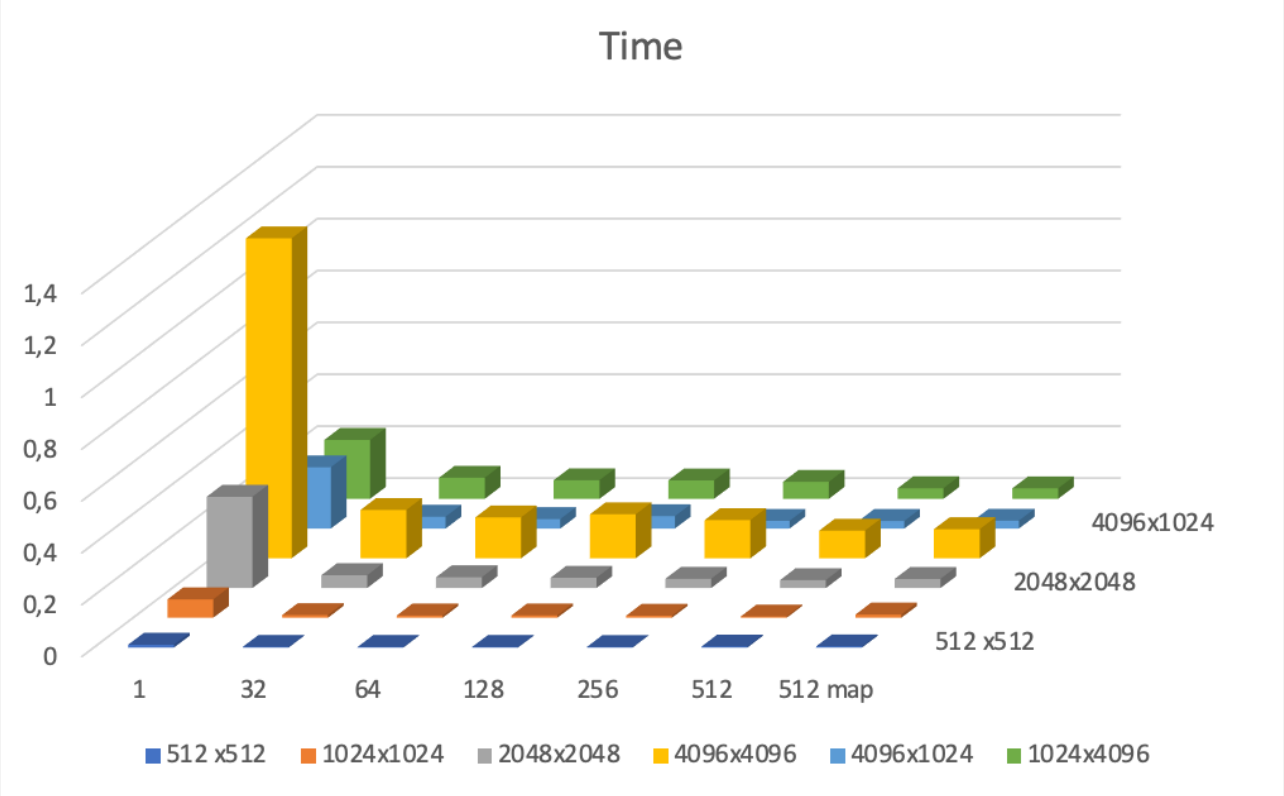
- имя файла – матрица A размером m x n
- имя файла - вектор в
- имя файла – результат, вектор c

Формат задания матрицы A – как в первом задании.

Результаты

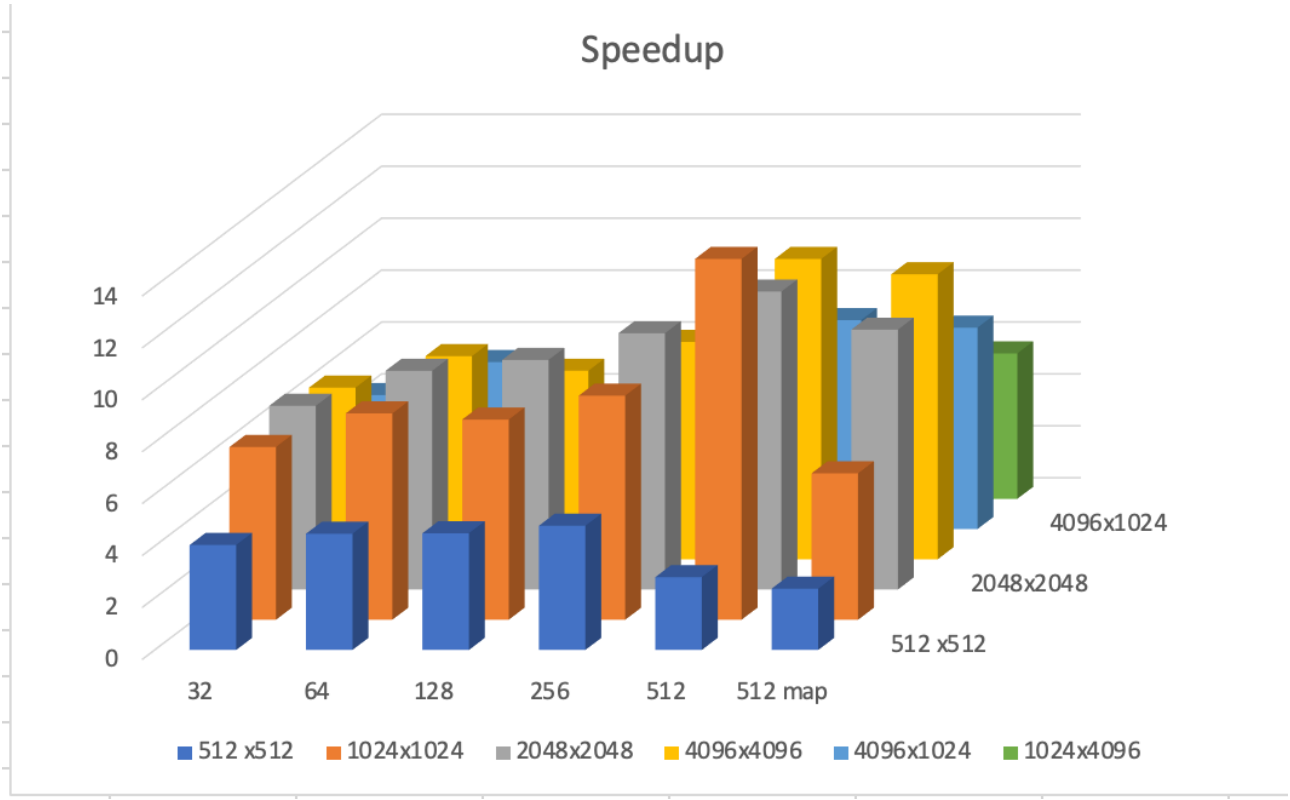
Время выполнения

	512 x512	1024x1024	2048x2048	4096x4096	4096x1024	1024x4096
1	0,0110203	0,07124313	0,351918	1,23384021	0,236392	0,227343
32	0,00272534	0,01070898	0,04972847	0,18656258	0,04578827	0,08152663
64	0,00246134	0,00896303	0,04176783	0,15762873	0,03672624	0,07173687
128	0,00245158	0,00924017	0,03978272	0,16987391	0,04782672	0,0708392
256	0,00230368	0,00825485	0,03565123	0,14736824	0,02954832	0,06572372
512	0,00392944	0,00512552	0,03066261	0,10662342	0,02936873	0,04076872
512 map	0,00467121	0,01263767	0,03516271	0,11236101	0,03047101	0,0405611



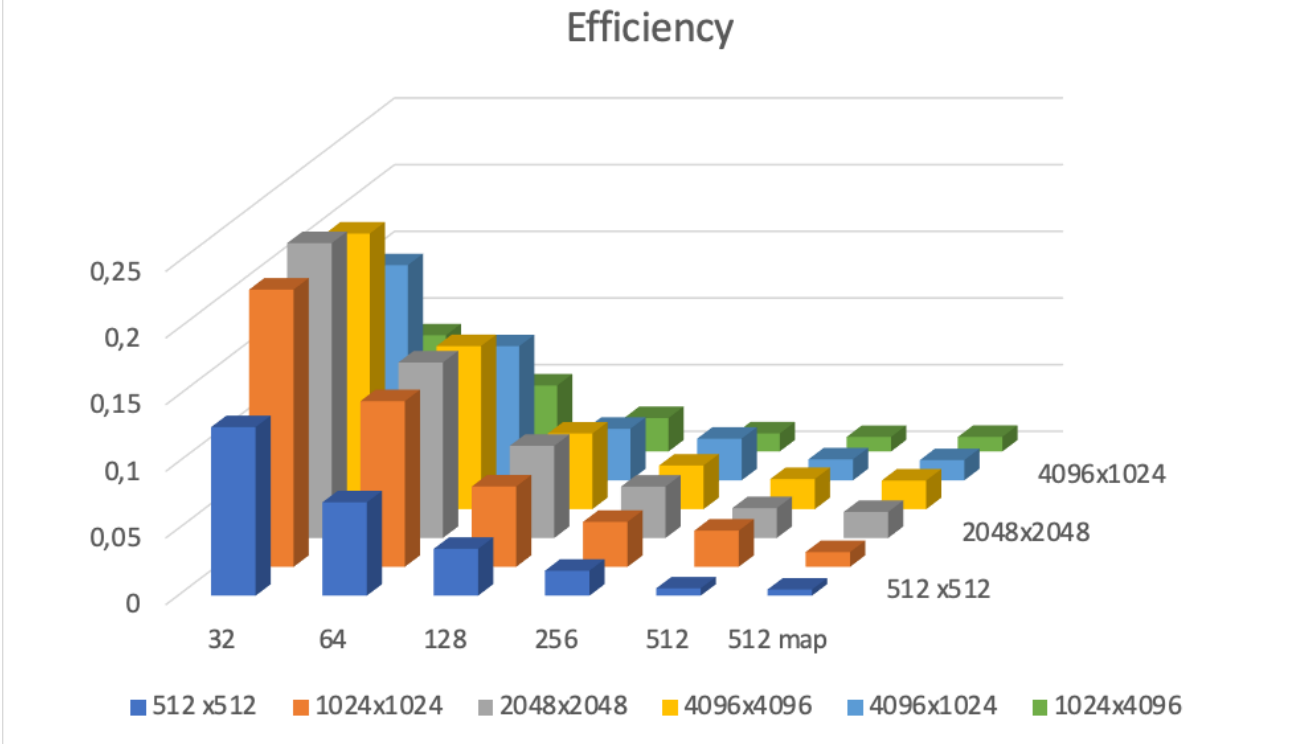
Ускорение

	512 x512	1024x1024	2048x2048	4096x4096	4096x1024	1024x4096
32	4,043642261	6,652653194	7,076791222	6,613546028	5,162719622	2,7885735
64	4,477357862	7,948554228	8,425575377	7,827508412	6,436596831	3,1691235
128	4,495182699	7,710153601	8,84600148	7,263270799	4,942676395	3,2092824
256	4,783780733	8,630457246	9,871132076	8,372497425	8,000184105	3,4590708
512	2,804547213	13,89968823	11,47710518	11,57194367	8,04910529	5,5764076
512 map	2,359196011	5,637362742	10,00827297	10,98103524	7,757931227	5,6049515



Эффективность

	512 x512	1024x1024	2048x2048	4096x4096	4096x1024	1024x4096
32	0,126363821	0,207895412	0,221149726	0,206673313	0,161334988	0,08714292
64	0,069958717	0,12419616	0,131649615	0,122304819	0,100571825	0,04951755
128	0,035118615	0,060235575	0,069109387	0,056744303	0,038614659	0,02507252
256	0,018686643	0,033712724	0,03855911	0,032705068	0,031250719	0,013512
512	0,005477631	0,027147829	0,022416221	0,022601452	0,015720909	0,01089142
512 map	0,004607805	0,011010474	0,019547408	0,021447334	0,015152209	0,01094717



Выводы

Параллелизм дает хорошие результаты с ростом объема данных. Ускорение программы растет до определенного предела и затем перестает увеличиваться, что связано с ростом накладных расходов на организацию параллелизма и на пересылки данных.