

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



Спецкурс: системы и средства параллельного программирования.

Отчёт № 5.

**Параллельный алгоритм умножения матрицы на вектор.
Разработка параллельной MPI программы и исследование ее
эффективности.**

Работу выполнил
Тони Кастильо Мартин

Москва 2018

Постановка задачи и формат данных.

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.

Задача: Результатом работы является количество простых чисел выведенное в командную строку и файл с самими числами в текстовом виде (сортировать не обязательно).

Формат командной строки: <имя файла матрицы A > <имя файла матрицы B > <имя файла матрицы C >

Формат файла-матрицы: Матрица представляются в виде бинарного файла следующего формата:

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

Элементы матрицы хранятся построчно.

Анализ времени выполнения: Для оценки времени выполнения программы использовалась функция:

MPI_Wtime()

Для повышения надёжности экспериментов опыты проводились несколько раз (10).

Верификация: Для проверки корректности работы программы использовались тестовые данные.

Результаты выполнения

Зависимость времени выполнения рабочих циклов: для каждого из заданных значений размеров матрицы (512x512, 1024x1024, 2048x2048, 4096x4096, 4096x1024, 1024x4096).

Время

N	M	Мэппинг	32	64	128	256	512
512	512		0.00544007	0.00546864	0.00568669	0.00625499	0.00719385
1024	1024		0.0218084	0.0218725	0.0230445	0.0248698	0.0259238
2048	2048		0.0917544	0.0925669	0.0966415	0.098145	0.1046961
4096	4096		0.415421	0.425566	0.43869	0.456379	0.4719605
4096	1024		0.0878395	0.0879987	0.0892775	0.090139	0.0922221
1024	4096		0.101405	0.101461	0.101869	0.102036	0.102869

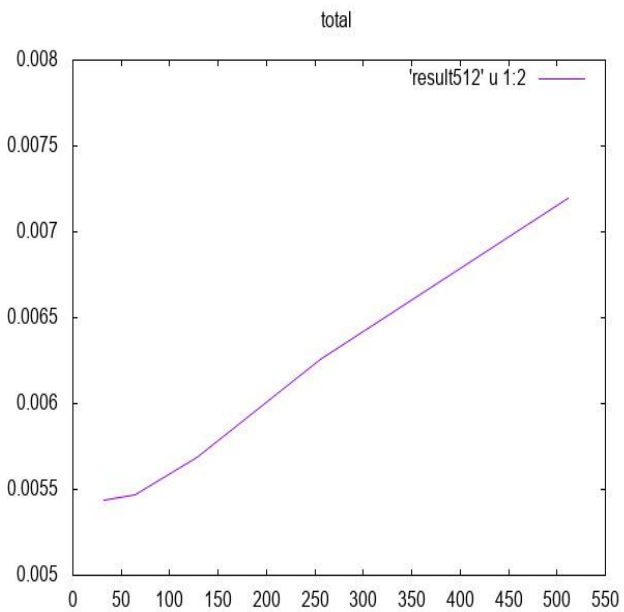
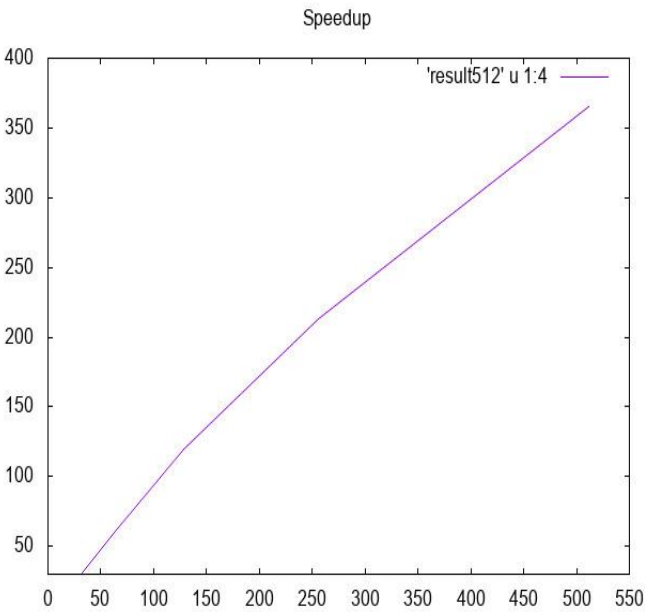
ускорения

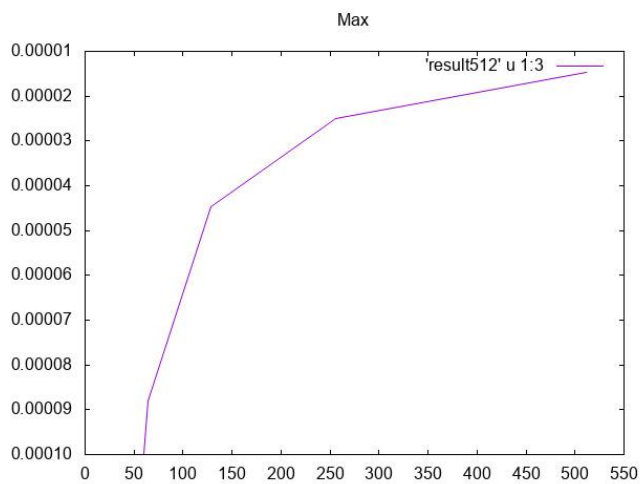
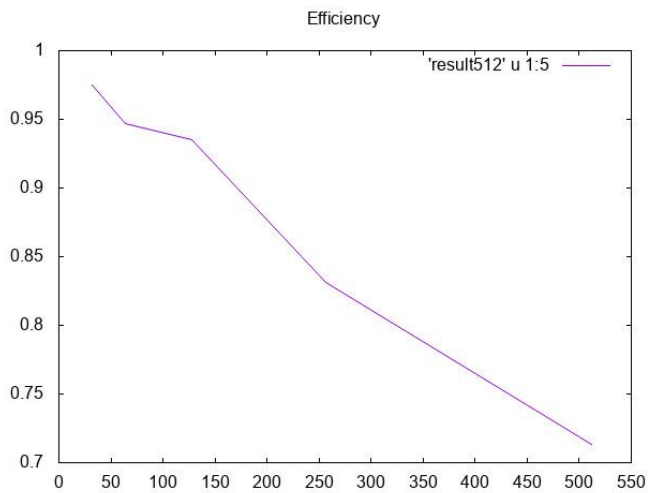
N	M	Мэппинг	32	64	128	256	512
512	512		31.1881	60.5994	119.733	212.846	365.327
1024	1024		31.2972	63.1837	122.702	240.793	450.375
2048	2048		31.8451	63.703	123.256	242.676	485.147
4096	4096		31.9175	63.8608	127.623	254.798	510.8
4096	1024		31.3544	63.7595	127.069	253.604	495.095
1024	4096		31.906	63.7162	127.039	252.532	499.176

эффективности.

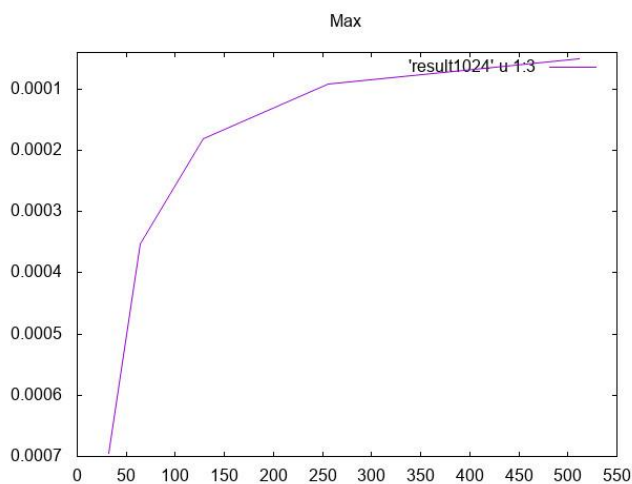
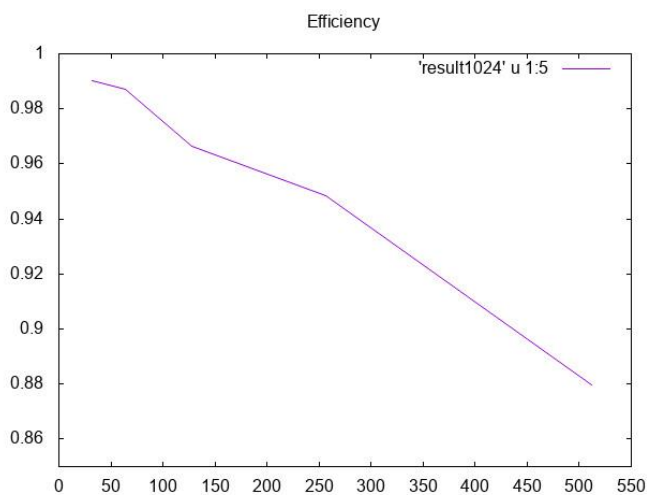
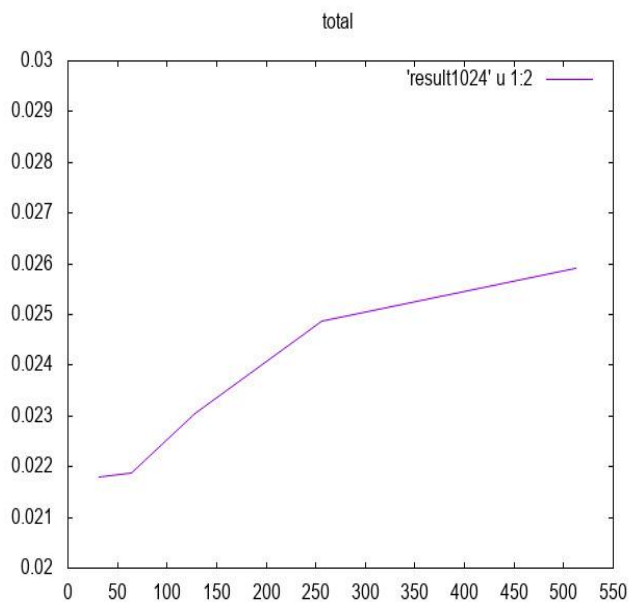
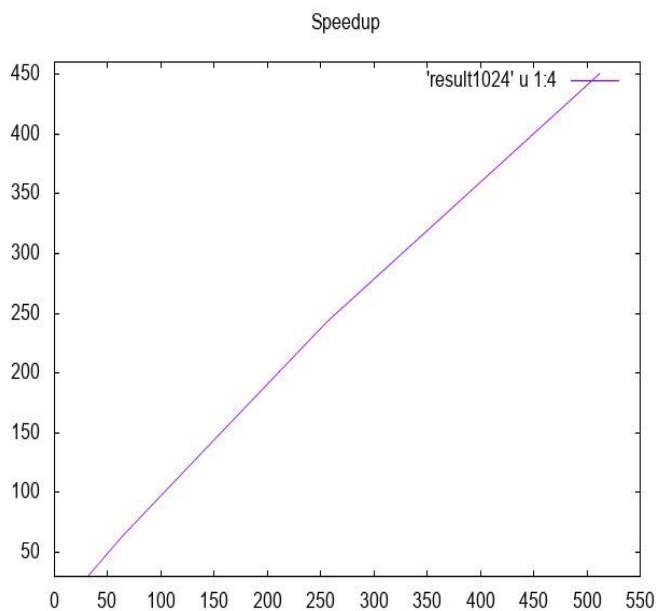
N	M	Мэппинг	32	64	128	256	512
512	512		0.974629	0.946865	0.935417	0.831431	0.713529
1024	1024		0.990304	0.987246	0.966423	0.94841	0.879639
2048	2048		0.995159	0.985359	0.965124	0.940141	0.927553
4096	4096		0.997423	0.997226	0.997053	0.995304	0.994304
4096	1024		0.9931107	0.9901987	0.987154	0.980064	0.966982
1024	4096		0.997062	0.995566	0.992492	0.986452	0.974953

512x512

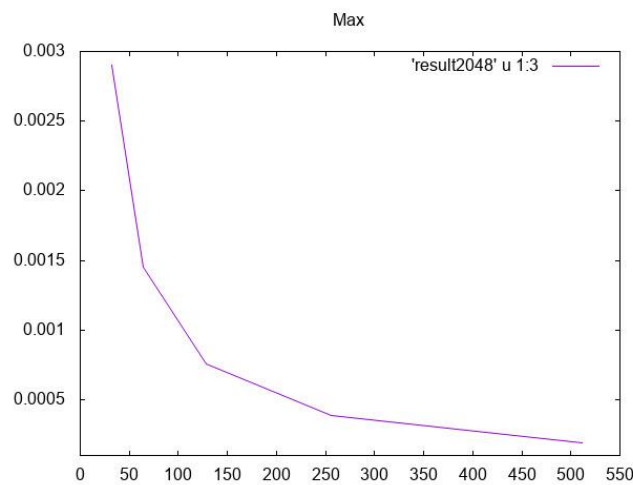
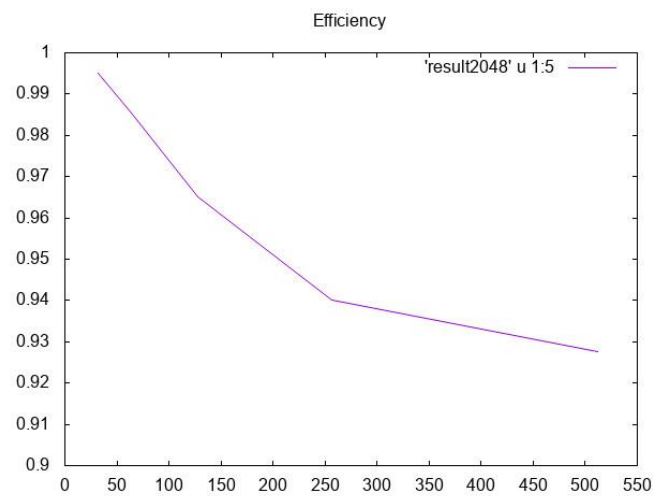
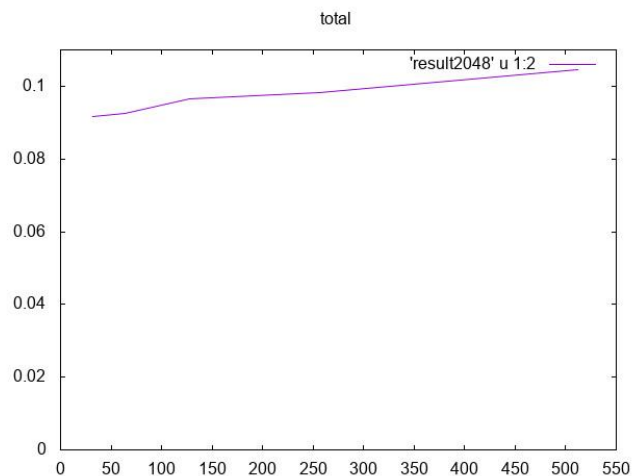
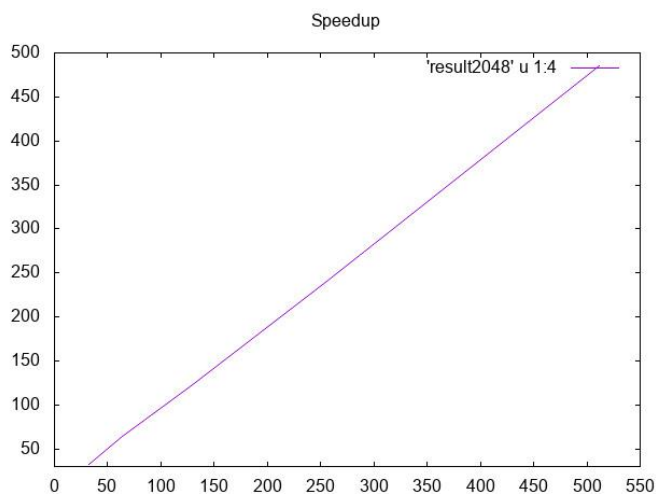




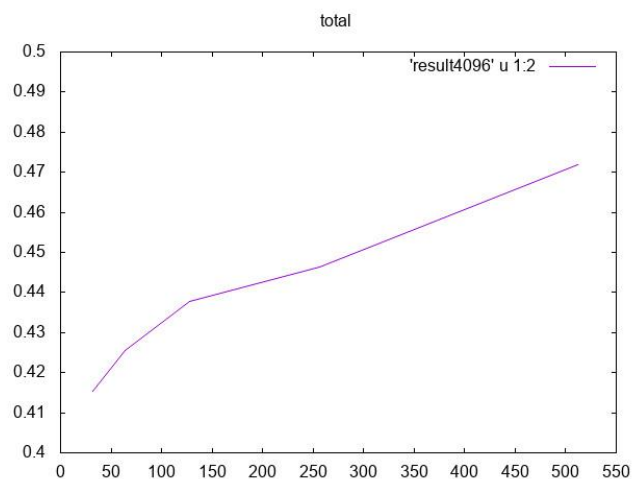
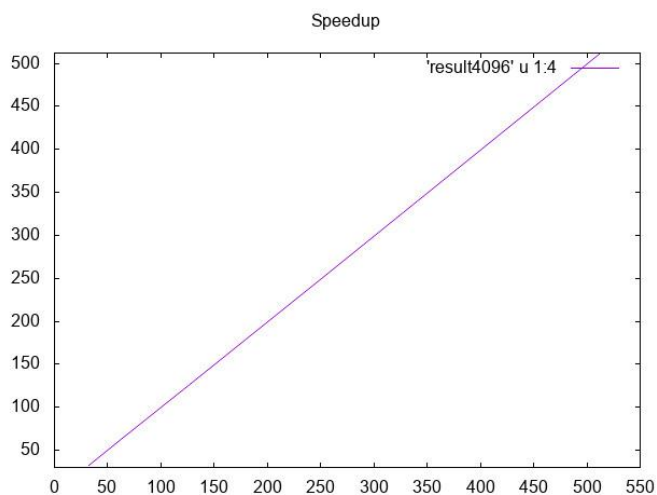
1024x1024

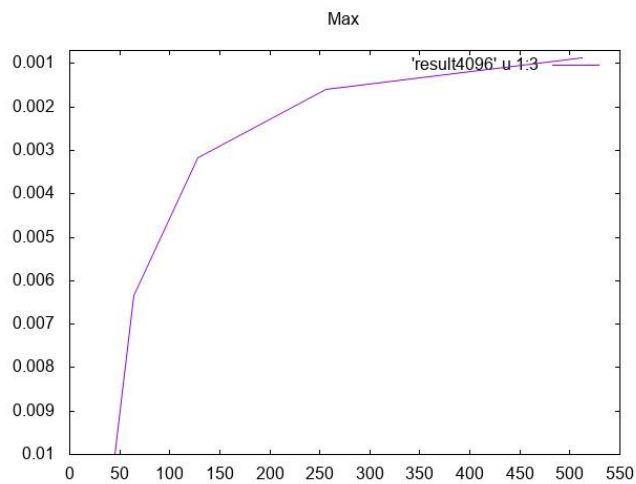
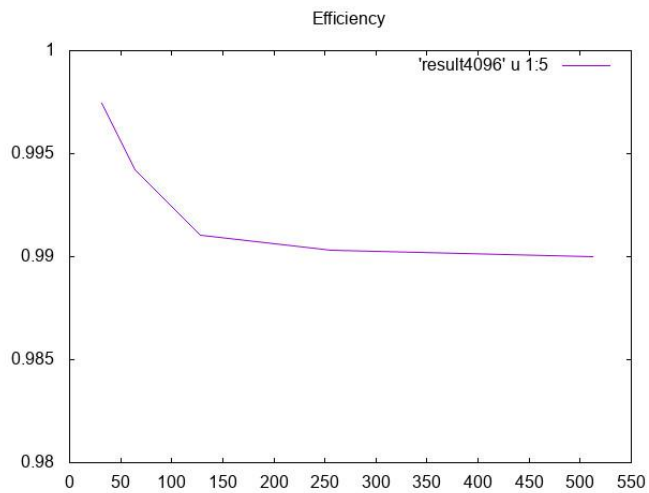


2048x2048



4096x4096





4096x1024

