

Московский Государственный Университет

им. М.В. Ломоносова

Факультет Вычислительной Математики и Кибернетики.
Кафедра Суперкомпьютеров и Квантовой Информатики.



Практикум на ЭВМ.

Отчет №2: Параллельная программа на MPI, которая
реализует однокубитное квантовое преобразование.

Шахворостов Дмитрий 323

2021

Постановка задачи

Реализовать параллельную программу на C++ с использованием MPI, которая выполняет однокубитное квантовое преобразование над вектором состояний длины 2^n , где n – количество кубитов, по указанному номеру кубита k .

Аргументы командной строки: $\langle n \rangle \langle k \rangle \langle mode \rangle$. Опции mode: 1 – чтение входного вектора из файла in.bin, 2 – случайная генерация.

Формат хранения данных: в бинарном файле (.bin) первое число (int) n — количество кубитов, следующие 2^n комплексных чисел — элементы вектора (комплексное число хранится в виде double Re, double Im).

Сборка: make

Пример запуска: `mpirun -np 8 main "25" "1" "2"`

Результаты

А) 1 позиция.

Количество кубитов	Количество потоков	Время работы программы (сек)	Ускорение
25	1	4,32	1,00
	2	2,33	1,86
	4	1,21	3,55
	8	0,66	6,56
26	1	9,34	1,00
	2	5,54	1,69
	4	2,80	3,33
	8	1,50	6,22
27	1	17,37	1,00
	2	9,40	1,85
	4	4,71	3,69
	8	2,40	7,23

Б) 18 позиция

Количество кубитов	Количество потоков	Максимальное время работы процесса(сек)	Ускорение
25	1	4,47	1,00
	2	2,28	1,96
	4	1,22	3,65
	8	0,62	7,20
26	1	8,67	1,00
	2	4,42	1,96
	4	2,24	3,87
	8	1,14	7,58
27	1	17,55	1,00
	2	8,82	1,99
	4	4,46	3,94
	8	2,34	7,51

В) Последняя позиция.

Количество кубитов	Количество потоков	Максимальное время работы процесса(сек)	Ускорение
25	1	4,32	1,00
	2	2,19	1,97
	4	1,13	3,82
	8	0,60	7,18
26	1	8,65	1,00
	2	4,42	1,96

	4	2,20	3,93
	8	1,15	7,52
27	1	17,29	1,00
	2	8,80	1,96
	4	4,42	3,91
	8	2,25	7,69

Результаты проверки на корректность (выходные векторы), полученные при моделирования 16 кубитов, лежат в папке test вместе с входным вектором in.txt/in.bin.

Основные выводы

Исследования показывают, что изменение количества запущенных процессов оказывает значительное влияние на максимальное время выполнения программы. Другими словами, алгоритм масштабируется хорошо, но не идеально из-за накладных расходов на поддержку параллельной программы. С увеличением номера кубита, к которому применяется преобразование, получаемое ускорение увеличивается.