

Московский Государственный Университет

им. М.В. Ломоносова

Факультет Вычислительной Математики и Кибернетики.

Кафедра Суперкомпьютеров и Квантовой Информатики.




Практикум на ЭВМ.

Отчет №1: Однокубитные операции.

Ухин Сергей 323

2019

Постановка задачи.

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

Формат командной строки:

```
./main <n> <k> <mode (1-file \"in.bin\" , 2-random)>
```

Формат хранения данных:

Бинарный файл. Сначала $\text{int}(n)$ — количество кубитов, следующие 2^n комплексных чисел — элементы вектора. (комплексное число хранится в виде `double re`, `double im`).

Сборка:

```
make
```

Результаты.

А) 1 позиция.

Количество кубитов	Количество потоков	Максимальное время работы процесса(сек)	Ускорение
20	1	0.210123	1
	2	0.136369	1.540841394
	4	0.0625285	3.360435641
	8	0.0506086	4.151922796
24	1	3.40368	1
	2	1.7857	1.906076049
	4	0.972244	3.500849581
	8	0.576392	5.905147885
28	1	54.3602	1
	2	28.6249	1.899052922
	4	14.4598	3.759401928
	8	7.63258	7.122126463
30	1	223.645	1
	2	114.403	1.954887547

	4	57.8825	3.863775753
	8	31.9549	6.998770142

Б) 10 позиция

Количество кубитов	Количество потоков	Максимальное время работы процесса(сек)	Ускорение
20	1	0.209872	1
	2	0.128869	1.628568546
	4	0.0799433	2.625260653
	8	0.0484804	4.329007186
24	1	3.40218	1
	2	1.71461	1.98422965
	4	0.926418	3.672402738
	8	0.739642	4.599765833
28	1	54.1808	1
	2	27.2828	1.985895876
	4	13.825	3.919045208
	8	7.57354	7.153959707
30	1	218.18	1
	2	110.296	1.978131573
	4	55.8313	3.907843808
	8	29.038	7.513602865

В) Последняя позиция.

Количество кубитов	Количество потоков	Максимальное время работы процесса(сек)	Ускорение
20	1	0.211279	1
	2	0.106374	1.93263536
	4	0.0537416	3.061395077
	8	0.0371335	3.809163236
24	1	3.39085	1
	2	1.70833	1.905224084
	4	0.875055	2.66135006
	8	0.59072	4.0165847
28	1	54.3794	1
	2	27.5822	1.763952449
	4	14.022	3.661859402
	8	7.23951	4.186069676
30	1	217.682	1
	2	109.856	1.981521264
	4	55.6742	3.909925962
	8	29.0162	7.502085042

Основные выводы.

Исследования показывают, что изменение количества запущенных процессов оказывает значительное влияние на максимальное время выполнения программы. Другими словами алгоритм хорошо масштабируется, но не стоит забывать про накладные расходы на поддержку параллельной программы.