Московский Государственный Университет им. М.В. Ломоносова

Факультет Вычислительной Математики и Кибернетики

Кафедра Суперкомпьютеров и Квантовой Информатики

Практикум на ЭВМ
Отчёт № 3
Параллельная программа на MPI и OpenMP, реализующая однокубитное квантовое преобразование с шумами

Работу выполнил

Сайбель Т. А.

Москва 2021

Постановка задачи и формат данных

1) Реализовать параллельную программу на C++ с использованием MPI и OpenMP, которая выполняет квантовое преобразование n-Адамар с зашумленными вентилями над вектором состояний длины 2 n , где n – количество кубитов. Использовать рекомендованную модель зашумления. Для работы с комплексными числами использовать стандартную библиотеку шаблонов.

2) Протестировать программу на системе Blue Gene/P.

Начальное состояние вектора генерируется случайным образом и нормируется (тоже параллельно).

Формат командной строки: <Число кубитов *n*> <Уровень шума e><Количество потоков> <имя файла исходного вектора> <имя файла полученного вектора> <>

Формат файла-вектора: Вектор представляется в виде бинарного файла следующего формата:

Тип	*Значение*	*Описание*
Число типа int	n – натуральное число	Число кубитов
Массив чисел типа complex<double>	2^n – комплексных чисел	Элементы вектора

Описание алгоритма

Однокубитная операция над комплексным входным вектором {a i } размерности 2 n задается двумя параметрами: комплексной матрицей {u ij } размера 2x2 (вентиль) и числом k от 1 до n (номер кубита, по которому проводится операция). Такая операция преобразует вектор $\{a_i\}$ в $\{b_i\}$ размерности 2^n , где все элементы вычисляются по следующей формуле:

$$b_{i_1 \ldots i_k \ldots i_n} = \sum_{j_k=0}^{1} u_{i_k j_k} a_{i_1 \ldots j_k \ldots i_n} = u_{i_k 0} a_{i_1 \ldots 0_k \ldots i_n} + u_{i_k 1} a_{i_1 \ldots 1_k \ldots i_n}$$

$$U = \begin{pmatrix} u_{00} & u_{01} \\ u_{10} & u_{11} \end{pmatrix}$$

Зашумленный вентиль Адамара H_e определяется следующими формулами:

$$H = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}, H_e = HU(\Theta), U(\Theta) = \begin{pmatrix} \cos\Theta & \sin\Theta \\ -\sin\Theta & \cos\Theta \end{pmatrix}, \Theta = e\xi, \xi \sim N(0,1),$$

где e – это уровень шума.
Преобразование n-Адамар – это преобразование Адамара, выполненное последовательно n раз над вектором состояний, при этом кубит, по которому проводится преобразование изменяется от 1 до n. В качестве меры потери точности используется 1-F, где F – мера точности (вероятность совпадения между идеальным и зашумленным векторами состояний). Мера точности вычисляется как квадрат модуля скалярного произведения соответствующих векторов.

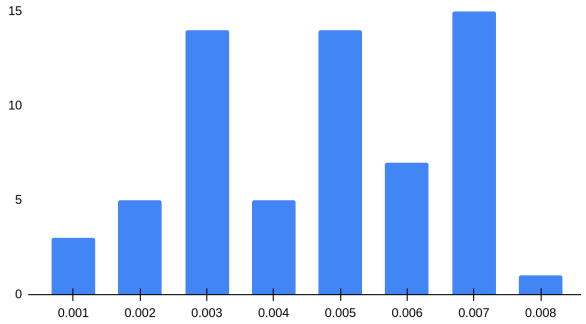
Аппаратное обеспечение:Исследования проводились на вычислительном комплексе Blue Gene/P.

Анализ времени выполнения: Для оценки времени выполнения программы использовалась функция MPI_Wtime().

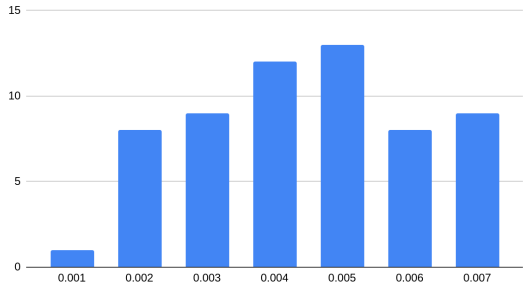
Результаты выполнения

Количество кубитов (n)	Количество вычислительных узлов	Количество используемых ядер в узле	Время работы (сек)	Ускорение
28	1	1	41,08	1
		2	21,398794	1,919687203
		4	11,536339	3,560834247
	2	1	43,816157	0,9375306693
		2	21,732912	1,890174267
		4	11,880787	3,457598474
	4	1	33,14957	1,239201323
		2	11,10364	3,699596799
		4	6,893902	5,958743104
	8	1	30,764215	1,335284876
		2	17,637573	2,32906143
		4	11,075619	3,708956673
	16	1	15,71953	2,613245498
		2	9,181529	4,474090427
		4	5,915609	6,9441694
	32	1	8,206907	5,005416901
		2	4,951889	8,295620318
		4	3,323409	12,36049821
	64	1	4,455869	9,219075112
		2	2,833516	14,49753275
		4	2,024185	20,29408923
	128	1	2,40456	17,08378705
		2	1,596655	25,7281573
		4	1,194315	34,3954409

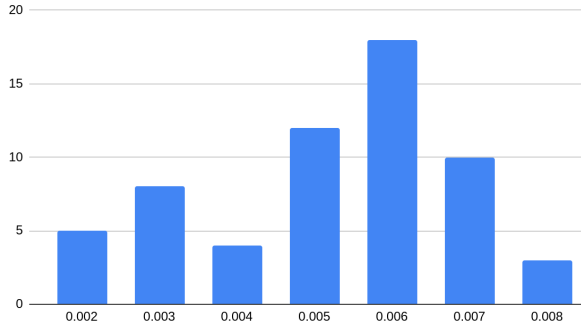
n=24; e=0.01



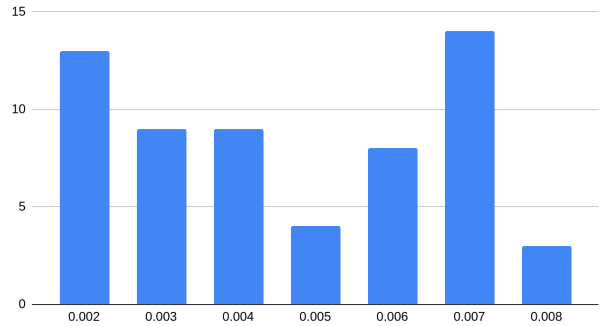
n=25; e=0.01



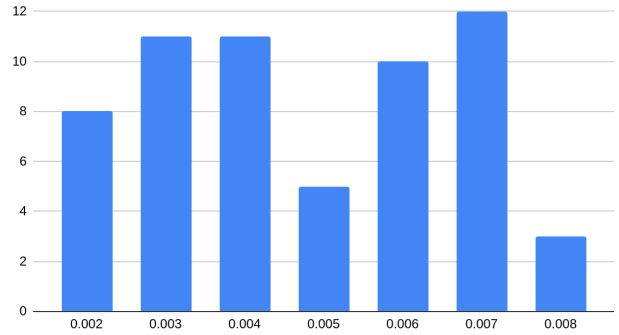
n=26; e=0.01



n=27; e=0.01

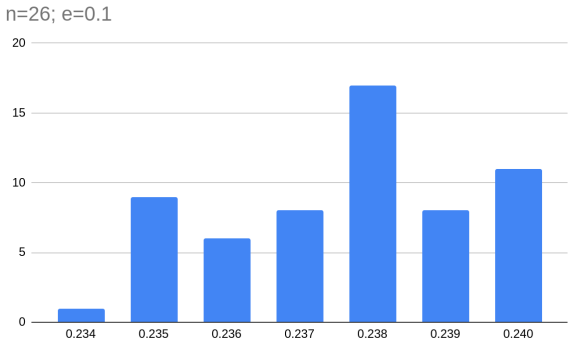


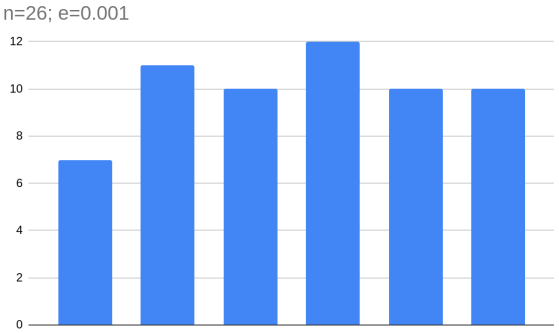
n=28; e=0.01



Количество кубитов	Среднее значение потери точности
24	0.00264463
25	0.00236435
26	0.00265423
27	0.00269367
28	0,00286371

e	Среднее значение потери точности
0.1	0.2352893
0.01	0.0026542
0.001	0,0000257





Основные выводы

Распараллеливание ускоряет выполнение программы, но с увеличением числа процессов эффективность снижается из-за роста количества пересылок и накладных расходов на организацию параллелизма.

Зашумление приводит к потерям точности. Увеличение количества кубитов в векторе состояния приводит к росту потерь точности в связи с ростом числа операций.